

Package: niflowr (via r-universe)

May 28, 2026

Title Spec-Driven Neuroimaging Command-Line Tool Wrappers

Version 0.1.0

Description A lightweight, declarative framework for wrapping neuroimaging command-line tools (FSL, AFNI, ANTs, FreeSurfer) in R. Tool interfaces are defined as JSON specifications that describe inputs, outputs, CLI rendering rules, and validation constraints. The package builds safe argument vectors (never shell strings), executes via 'processx', captures provenance, and integrates with 'targets' for pipeline orchestration.

License MIT + file LICENSE

URL <https://github.com/bbuchtsbaum/niflowr>

BugReports <https://github.com/bbuchtsbaum/niflowr/issues>

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Imports cli, digest, fs, glue, jsonlite, jsonvalidate, processx, yaml, utils

Suggests bidsappr, bidser, crew, fmriprefixer, knitr, neuroim2, neurotransform, openneuroR, rmarkdown, targets, tarchetypes, tibble, testthat (>= 3.0.0), withr

VignetteBuilder knitr

Config/testthat/edition 3

Config/pak/sysreqs cmake make libuv1-dev libssl-dev libnode-dev

Repository <https://bbuchtsbaum.r-universe.dev>

Date/Publication 2026-02-27 01:01:31 UTC

RemoteUrl <https://github.com/bbuchtsbaum/niflowr>

RemoteRef HEAD

RemoteSha 760c3d2dfad6776905a6724d5bbc59ac616ff7df

Contents

ni_afni_a_boverlap	9
ni_afni_afn_ito_nifti	10
ni_afni_align_epi_anat_py	11
ni_afni_allineate	12
ni_afni_auto_tcorrelate	16
ni_afni_auto_tlrc	17
ni_afni_autobox	19
ni_afni_automask	20
ni_afni_axialize	21
ni_afni_bandpass	22
ni_afni_blur_in_mask	24
ni_afni_blur_to_fwhm	25
ni_afni_brick_stat	26
ni_afni_bucket	27
ni_afni_calc	29
ni_afni_cat	30
ni_afni_cat_matvec	31
ni_afni_center_mass	32
ni_afni_clip_level	33
ni_afni_convert_dset	34
ni_afni_copy	35
ni_afni_deconvolve	36
ni_afni_degree centrality	40
ni_afni_despike	41
ni_afni_detrend	42
ni_afni_dot	43
ni_afni_ecm	44
ni_afni_edge3	46
ni_afni_eval	47
ni_afni_fim	48
ni_afni_fourier	49
ni_afni_fwh_mx	50
ni_afni_gcor	52
ni_afni_hist	53
ni_afni_lfcd	54
ni_afni_local_bistat	55
ni_afni_localstat	56
ni_afni_mask_tool	59
ni_afni_maskave	60
ni_afni_means	61
ni_afni_merge	62
ni_afni_net_corr	63
ni_afni_notes	66
ni_afni_nwarp_adjust	67
ni_afni_nwarp_apply	68
ni_afni_nwarp_cat	69

ni_afni_one_d_tool_py	70
ni_afni_outlier_count	72
ni_afni_quality_index	73
ni_afni_qwarp	75
ni_afni_qwarp_plus_minus	82
ni_afni_re_ho	90
ni_afni_refit	91
ni_afni_remlfit	93
ni_afni_resample	96
ni_afni_retroicor	98
ni_afni_roi_stats	99
ni_afni_seg	101
ni_afni_skull_strip	102
ni_afni_svm_test	103
ni_afni_svm_train	104
ni_afni_synthesize	106
ni_afni_t_cat	107
ni_afni_t_cat_sub_brick	108
ni_afni_t_corr_map	109
ni_afni_t_corr1_d	111
ni_afni_t_correlate	112
ni_afni_t_norm	113
ni_afni_t_project	114
ni_afni_t_shift	117
ni_afni_t_smooth	118
ni_afni_t_stat	119
ni_afni_to3_d	120
ni_afni_undump	122
ni_afni_unifize	123
ni_afni_volreg	125
ni_afni_warp	126
ni_afni_z_cut_up	128
ni_afni_zcat	129
ni_afni_zeropad	130
ni_ants_affine_initializer	131
ni_ants_ai	133
ni_ants_ants	134
ni_ants_ants_introduction	136
ni_ants_apply_transforms	137
ni_ants_apply_transforms_to_points	139
ni_ants_atropos	140
ni_ants_average_affine_transform	142
ni_ants_average_images	143
ni_ants_brain_extraction	144
ni_ants_buildtemplateparallel	145
ni_ants_compose_multi_transform	147
ni_ants_composite_transform_util	148
ni_ants_convert_scalar_image_to_rgb	149

ni_ants_cortical_thickness	150
ni_ants_create_jacobian_determinant_image	153
ni_ants_create_tiled_mosaic	154
ni_ants_denoise_image	156
ni_ants_gen_warp_fields	157
ni_ants_image_math	158
ni_ants_joint_fusion	160
ni_ants_kelly_kapowski	162
ni_ants_label_geometry	164
ni_ants_laplacian_thickness	165
ni_ants_measure_image_similarity	166
ni_ants_multiply_images	167
ni_ants_n4_bias_field_correction	168
ni_ants_registration	170
ni_ants_registration_syn_quick	172
ni_ants_resample_image_by_spacing	174
ni_ants_threshold_image	175
ni_ants_warp_image_multi_transform	176
ni_ants_warp_time_series_image_multi_transform	177
ni_bids_app	179
ni_bids_inputs	180
ni_call	180
ni_cmd	181
ni_config	182
ni_constraints	183
ni_dcm2niix_convert	183
ni_deriv_path	185
ni_doctor	186
ni_dry_run	187
ni_fastsurfer_run	187
ni_fastsurfer_segment	189
ni_fmriprep_confounds	190
ni_fmriprep_derivatives	191
ni_fmriprep_preproc	192
ni_freesurfer_add_x_form_to_header	193
ni_freesurfer_aparc2_aseg	194
ni_freesurfer_apas2_aseg	196
ni_freesurfer_apply_mask	197
ni_freesurfer_apply_vol_transform	198
ni_freesurfer_bb_register	200
ni_freesurfer_binarize	202
ni_freesurfer_ca_label	204
ni_freesurfer_ca_normalize	205
ni_freesurfer_ca_register	206
ni_freesurfer_check_talairach_alignment	208
ni_freesurfer_concatenate	209
ni_freesurfer_concatenate_lta	210
ni_freesurfer_contrast	212

ni_freesurfer_curvature	213
ni_freesurfer_curvature_stats	214
ni_freesurfer_dicom_convert	215
ni_freesurfer_edit_w_mwith_aseg	216
ni_freesurfer_em_register	217
ni_freesurfer_euler_number	218
ni_freesurfer_extract_main_component	219
ni_freesurfer_fit_ms_params	220
ni_freesurfer_fix_topology	221
ni_freesurfer_fuse_segmentations	222
ni_freesurfer_glm_fit	223
ni_freesurfer_gtm_seg	227
ni_freesurfer_gtmpvc	228
ni_freesurfer_image_info	232
ni_freesurfer_jacobian	233
ni_freesurfer_label2_annot	234
ni_freesurfer_label2_label	235
ni_freesurfer_label2_vol	236
ni_freesurfer_logan	238
ni_freesurfer_lta_convert	242
ni_freesurfer_make_average_subject	243
ni_freesurfer_make_surfaces	244
ni_freesurfer_mni_bias_correction	246
ni_freesurfer_mp_rto_mni305	247
ni_freesurfer_mr_is_ca_label	248
ni_freesurfer_mr_is_calc	249
ni_freesurfer_mr_is_combine	251
ni_freesurfer_mr_is_convert	252
ni_freesurfer_mr_is_expand	254
ni_freesurfer_mr_is_inflate	255
ni_freesurfer_mri_convert	256
ni_freesurfer_mri_coreg	261
ni_freesurfer_mri_fill	264
ni_freesurfer_mri_marching_cubes	265
ni_freesurfer_mri_preteess	266
ni_freesurfer_mri_tessellate	267
ni_freesurfer_mris_preproc	268
ni_freesurfer_mris_preproc_recon_all	270
ni_freesurfer_mrtm1	272
ni_freesurfer_mrtm2	275
ni_freesurfer_ms_lda	279
ni_freesurfer_normalize	280
ni_freesurfer_one_sample_t_test	281
ni_freesurfer_paint	285
ni_freesurfer_parcellation_stats	286
ni_freesurfer_parse_dicom_dir	288
ni_freesurfer_recon_all	289
ni_freesurfer_register	291

ni_freesurfer_register_av_ito_talairach	292
ni_freesurfer_relabel_hypointensities	293
ni_freesurfer_remove_intersection	294
ni_freesurfer_remove_neck	295
ni_freesurfer_resample	296
ni_freesurfer_robust_register	297
ni_freesurfer_robust_template	299
ni_freesurfer_sample_to_surface	301
ni_freesurfer_seg_stats	303
ni_freesurfer_seg_stats_recon_all	306
ni_freesurfer_segment_cc	309
ni_freesurfer_segment_wm	310
ni_freesurfer_smooth	311
ni_freesurfer_smooth_tessellation	312
ni_freesurfer_sphere	314
ni_freesurfer_spherical_average	315
ni_freesurfer_surface_smooth	316
ni_freesurfer_surface_snapshots	317
ni_freesurfer_surface_transform	319
ni_freesurfer_surface2_vol_transform	321
ni_freesurfer_synthesize_flash	322
ni_freesurfer_talairach_avi	323
ni_freesurfer_talairach_qc	324
ni_freesurfer_tkregister2	325
ni_freesurfer_unpack_sdicom_dir	327
ni_freesurfer_volume_mask	328
ni_freesurfer_watershed_skull_strip	329
ni_from_openneuro	330
ni_fsl_apply_mask	332
ni_fsl_apply_topup	333
ni_fsl_apply_warp	334
ni_fsl_apply_xfm	336
ni_fsl_ar1_image	339
ni_fsl_av_scale	340
ni_fsl_b0_calc	341
ni_fsl_bedpostx5	342
ni_fsl_bet	344
ni_fsl_binary_maths	346
ni_fsl_change_data_type	347
ni_fsl_cluster	348
ni_fsl_complex	350
ni_fsl_contrast_mgr	352
ni_fsl_convert_warp	353
ni_fsl_convert_xfm	355
ni_fsl_copy_geom	356
ni_fsl_dilate_image	357
ni_fsl_distance_map	358
ni_fsl_dti_fit	359

ni_fsl_dual_regression	361
ni_fsl_eddy	362
ni_fsl_eddy_correct	366
ni_fsl_eddy_quad	367
ni_fsl_epi_de_warp	368
ni_fsl_epi_reg	369
ni_fsl_erode_image	371
ni_fsl_extract_roi	372
ni_fsl_fast	373
ni_fsl_feat	375
ni_fsl_feat_model	376
ni_fsl_filmgls	377
ni_fsl_filter_regressor	379
ni_fsl_find_the_biggest	380
ni_fsl_first	381
ni_fsl_flameo	382
ni_fsl_flirt	384
ni_fsl_fnirt	387
ni_fsl_fugue	390
ni_fsl_glm	392
ni_fsl_ica_aroma	394
ni_fsl_image_maths	396
ni_fsl_image_meants	397
ni_fsl_image_stats	398
ni_fsl_inv_warp	399
ni_fsl_isotropic_smooth	401
ni_fsl_make_dyadic_vectors	402
ni_fsl_maths_command	403
ni_fsl_max_image	404
ni_fsl_maxn_image	405
ni_fsl_mcflirt	406
ni_fsl_mean_image	408
ni_fsl_median_image	409
ni_fsl_melodic	410
ni_fsl_merge	413
ni_fsl_min_image	414
ni_fsl_motion_outliers	416
ni_fsl_multi_image_maths	417
ni_fsl_overlay	418
ni_fsl_percentile_image	420
ni_fsl_plot_motion_params	421
ni_fsl_plot_time_series	422
ni_fsl_power_spectrum	423
ni_fsl_prelude	424
ni_fsl_prepare_fieldmap	426
ni_fsl_prob_track_x	427
ni_fsl_prob_track_x2	430
ni_fsl_proj_thresh	433

ni_fsl_randomise	434
ni_fsl_reorient2_std	436
ni_fsl_robust_fov	437
ni_fsl_sig_loss	438
ni_fsl_slice	439
ni_fsl_slice_timer	440
ni_fsl_slicer	441
ni_fsl_smm	443
ni_fsl_smooth	444
ni_fsl_smooth_estimate	445
ni_fsl_spatial_filter	446
ni_fsl_split	447
ni_fsl_std_image	448
ni_fsl_susan	449
ni_fsl_swap_dimensions	450
ni_fsl_temporal_filter	451
ni_fsl_text2_vest	452
ni_fsl_threshold	453
ni_fsl_topup	454
ni_fsl_tract_skeleton	457
ni_fsl_unary_maths	458
ni_fsl_vec_reg	459
ni_fsl_vest2_text	460
ni_fsl_warp_points	461
ni_fsl_warp_points_from_std	462
ni_fsl_warp_points_to_std	463
ni_fsl_warp_utils	464
ni_fsl_x_fibres5	466
ni_golden_cmdline_generate	468
ni_inputs	469
ni_lint_specs	469
ni_lock_read	470
ni_lock_validate	470
ni_outputs	471
ni_pin	471
ni_provenance	472
ni_provenance_read	472
ni_provenance_write	473
ni_read_output	473
ni_read_transform	474
ni_result	474
ni_run	475
ni_run_files	476
ni_spec_list	476
ni_spec_read	477
tar_ni_step	477

 ni_afni_a_boverlap *AFNI ABoverlap*

Description

Output (to screen) is a count of various things about how

Usage

```
ni_afni_a_boverlap(
  in_file_a,
  in_file_b,
  args = NULL,
  no_automask = NULL,
  out_file = NULL,
  quiet = NULL,
  verb = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file_a	Character; file path. input file A Required.
in_file_b	Character; file path. input file B Required.
args	Character. Additional parameters to the command
no_automask	Logical. consider input datasets as masks
out_file	Character; file path. collect output to a file
quiet	Logical. be as quiet as possible (without being entirely mute)
verb	Logical. print out some progress reports (to stderr)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

 ni_afni_afn_ito_nifti *AFNI AFNItoNIFTI*

Description

Converts AFNI format files to NIFTI format. This can also convert 2D or

Usage

```
ni_afni_afn_ito_nifti(
  in_file,
  args = NULL,
  denote = NULL,
  newid = NULL,
  oldid = NULL,
  out_file = NULL,
  pure = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. input file to 3dAFNItoNIFTI Required.
args	Character. Additional parameters to the command
denote	Logical. When writing the AFNI extension field, remove text notes that might contain subject identifying information.
newid	Logical. Give the new dataset a new AFNI ID code, to distinguish it from the input dataset.
oldid	Logical. Give the new dataset the input datasets AFNI ID code.
out_file	Character; file path. output image file name
pure	Logical. Do NOT write an AFNI extension field into the output file. Only use this option if needed. You can also use the 'nifti_tool' program to strip extensions from a file.
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

```
ni_afni_align_epi_anat_py
    AFNI AlignEpiAnatPy
```

Description

Align EPI to anatomical datasets or vice versa.

Usage

```
ni_afni_align_epi_anat_py(
    anat,
    epi_base,
    in_file,
    anat2epi = NULL,
    args = NULL,
    epi2anat = NULL,
    epi_strip = NULL,
    save_skullstrip = NULL,
    suffix = "_al",
    tshift = "on",
    volreg = "on",
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)
```

Arguments

anat	Character; file path. name of structural dataset Required.
epi_base	Character or numeric vector. the epi base used in alignments should be one of (0/mean/median/max/subbrick#) Required.
in_file	Character; file path. EPI dataset to align Required.
anat2epi	Logical. align anatomical to EPI dataset (default)
args	Character. Additional parameters to the command
epi2anat	Logical. align EPI to anatomical dataset
epi_strip	Character; one of: "3dSkullStrip", "3dAutomask", "None". method to mask brain in EPI data should be one of [3dSkullStrip/3dAutomask/None)

save_skullstrip	Logical. save skull-stripped (not aligned)
suffix	Character. append suffix to the original anat/epi dataset to use in the resulting dataset names (default is "_al")
tshift	Character; one of: "on", "off". do time shifting of EPI dataset before alignment should be 'on' or 'off', defaults to 'on'
volreg	Character; one of: "on", "off". do volume registration on EPI dataset before alignment should be 'on' or 'off', defaults to 'on'
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_allineate *AFNI Allineate*

Description

Program to align one dataset (the 'source') to a base dataset

Usage

```
ni_afni_allineate(
  in_file,
  allcostx = NULL,
  args = NULL,
  autobox = NULL,
  automask = NULL,
  autoweight = NULL,
  center_of_mass = NULL,
  check = NULL,
  convergence = NULL,
  cost = NULL,
  epi = NULL,
  final_interpolation = NULL,
  fine_blur = NULL,
  in_matrix = NULL,
  in_param_file = NULL,
  interpolation = NULL,
```

```

master = NULL,
maxrot = NULL,
maxscl = NULL,
maxshf = NULL,
maxshr = NULL,
newgrid = NULL,
nmatch = NULL,
no_pad = NULL,
nomask = NULL,
nwarp = NULL,
nwarp_fixdep = NULL,
nwarp_fixmot = NULL,
one_pass = NULL,
out_file = NULL,
out_matrix = NULL,
out_param_file = NULL,
out_weight_file = NULL,
overwrite = NULL,
quiet = NULL,
reference = NULL,
replacebase = NULL,
replacemeth = NULL,
source_automask = NULL,
source_mask = NULL,
two_best = NULL,
two_blur = NULL,
two_first = NULL,
two_pass = NULL,
usetemp = NULL,
verbose = NULL,
warp_type = NULL,
warpfreeze = NULL,
weight = NULL,
weight_file = NULL,
zclip = NULL,
.cwd = NULL,
.env = NULL,
.engine = NULL,
.profile = NULL,
dry_run = FALSE,
echo = interactive()
)

```

Arguments

in_file	Character; file path. input file to 3dAllineate Required.
allcostx	Character; file path. Compute and print ALL available cost functionals for the un-warped inputs AND THEN QUIT. If you use this option none of the other

	expected outputs will be produced
args	Character. Additional parameters to the command
autobox	Logical. Expand the -automask function to enclose a rectangular box that holds the irregular mask.
automask	Integer. Compute a mask function, set a value for dilation or 0.
autoweight	Character. Compute a weight function using the 3dAutomask algorithm plus some blurring of the base image.
center_of_mass	Character. Use the center-of-mass calculation to bracket the shifts.
check	Character or numeric vector. After cost functional optimization is done, start at the final parameters and RE-optimize using this new cost functions. If the results are too different, a warning message will be printed. However, the final parameters from the original optimization will be used to create the output dataset.
convergence	Numeric. Convergence test in millimeters (default 0.05mm).
cost	Character; one of: "leastsq", "ls", "mutualinfo", "mi", "corratio_mul", "crM", "norm_mutualinfo", "nmi", "hellinger", "hel", "corratio_add", "crA", "corratio_uns", "crU". Defines the 'cost' function that defines the matching between the source and the base
epi	Logical. Treat the source dataset as being composed of warped EPI slices, and the base as comprising anatomically 'true' images. Only phase-encoding direction image shearing and scaling will be allowed with this option.
final_interpolation	Character; one of: "nearestneighbour", "linear", "cubic", "quintic", "wsinc5". Defines interpolation method used to create the output dataset
fine_blur	Numeric. Set the blurring radius to use in the fine resolution pass to 'x' mm. A small amount (1-2 mm?) of blurring at the fine step may help with convergence, if there is some problem, especially if the base volume is very noisy. [Default == 0 mm = no blurring at the final alignment pass]
in_matrix	Character; file path. matrix to align input file
in_param_file	Character; file path. Read warp parameters from file and apply them to the source dataset, and produce a new dataset
interpolation	Character; one of: "nearestneighbour", "linear", "cubic", "quintic". Defines interpolation method to use during matching
master	Character; file path. Write the output dataset on the same grid as this file.
maxrot	Numeric. Maximum allowed rotation in degrees.
maxscl	Numeric. Maximum allowed scaling factor.
maxshf	Numeric. Maximum allowed shift in mm.
maxshr	Numeric. Maximum allowed shearing factor.
newgrid	Numeric. Write the output dataset using isotropic grid spacing in mm.
nmatch	Integer. Use at most n scattered points to match the datasets.
no_pad	Logical. Do not use zero-padding on the base image.

nomask	Logical. Don't compute the autoweight/mask; if -weight is not also used, then every voxel will be counted equally.
nwarp	Character; one of: "bilinear", "cubic", "quintic", "heptic", "nonic", "poly3", "poly5", "poly7", "poly9". Experimental nonlinear warping: bilinear or legendre poly.
nwarp_fixdep	Character or numeric vector. To fix non-linear warp dependency along directions.
nwarp_fixmot	Character or numeric vector. To fix motion along directions.
one_pass	Logical. Use only the refining pass – do not try a coarse resolution pass first. Useful if you know that only small amounts of image alignment are needed.
out_file	Character; file path. output file from 3dAllineate
out_matrix	Character; file path. Save the transformation matrix for each volume.
out_param_file	Character; file path. Save the warp parameters in ASCII (.1D) format.
out_weight_file	Character; file path. Write the weight volume to disk as a dataset
overwrite	Logical. overwrite output file if it already exists
quiet	Logical. Don't print out verbose progress reports.
reference	Character; file path. file to be used as reference, the first volume will be used if not given the reference will be the first volume of in_file.
replacebase	Logical. If the source has more than one volume, then after the first volume is aligned to the base.
replacemeth	Character; one of: "leastsq", "ls", "mutualinfo", "mi", "corratio_mul", "crM", "norm_mutualinfo", "nmi", "hellinger", "hel", "corratio_add", "crA", "corratio_uns", "crU". After first volume is aligned, switch method for later volumes. For use with '-replacebase'.
source_automask	Integer. Automatically mask the source dataset with dilation or 0.
source_mask	Character; file path. mask the input dataset
two_best	Integer. In the coarse pass, use the best 'bb' set of initialpoints to search for the starting point for the finepass. If bb==0, then no search is made for the beststarting point, and the identity transformation is used as the starting point. [Default=5; min=0 max=11]
two_blur	Numeric. Set the blurring radius for the first pass in mm.
two_first	Logical. Use -twopass on the first image to be registered, and then on all subsequent images from the source dataset, use results from the first image's coarse pass to start the fine pass.
two_pass	Logical. Use a two pass alignment strategy for all volumes, searching for a large rotation+shift and then refining the alignment.
usetemp	Logical. temporary file use
verbose	Logical. Print out verbose progress reports.
warp_type	Character; one of: "shift_only", "shift_rotate", "shift_rotate_scale", "affine_general". Set the warp type.

warpfreeze	Logical. Freeze the non-rigid body parameters after first volume.
weight	Character or numeric vector. Set the weighting for each voxel in the base dataset; larger weights mean that voxel count more in the cost function. If an image file is given, the volume must be defined on the same grid as the base dataset
weight_file	Character; file path. Set the weighting for each voxel in the base dataset; larger weights mean that voxel count more in the cost function. Must be defined on the same grid as the base dataset
zclip	Logical. Replace negative values in the input datasets (source & base) with zero.
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_auto_tcorrelate
AFNI AutoTcorrelate

Description

Computes the correlation coefficient between the time series of each

Usage

```
ni_afni_auto_tcorrelate(
  in_file,
  args = NULL,
  eta2 = NULL,
  mask = NULL,
  mask_only_targets = NULL,
  mask_source = NULL,
  out_file = NULL,
  polort = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. timeseries x space (volume or surface) file Required.
args	Character. Additional parameters to the command
eta2	Logical. eta^2 similarity
mask	Character; file path. mask of voxels
mask_only_targets	Logical. use mask only on targets voxels
mask_source	Character; file path. mask for source voxels
out_file	Character; file path. output image file name
polort	Integer. Remove polynomial trend of order m or -1 for no detrending
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_auto_tlrc	<i>AFNI AutoTLRC</i>
-------------------	----------------------

Description

A minimal wrapper for the AutoTLRC script

Usage

```
ni_afni_auto_tlrc(
  base,
  in_file,
  args = NULL,
  no_ss = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

base	<p>Character. Reference anatomical volume. Usually this volume is in some standard space like TLRC or MNI space and with afni dataset view of (+tlrc). Preferably, this reference volume should have had the skull removed but that is not mandatory. AFNI's distribution contains several templates. For a longer list, use "whereami -show_templates" TT_N27+tlrc -> Single subject, skull stripped volume. This volume is also known as N27_SurfVol_NoSkull+tlrc elsewhere in AFNI and SUMA land. (www.loni.ucla.edu, www.bic.mni.mcgill.ca) This template has a full set of FreeSurfer (surfer.nmr.mgh.harvard.edu) surface models that can be used in SUMA. For details, see Talairach-related link: https://afni.nimh.nih.gov/afni/suma TT_icbm452+tlrc -> Average volume of 452 normal brains. Skull Stripped. (www.loni.ucla.edu) TT_avg152T1+tlrc -> Average volume of 152 normal brains. Skull Stripped. (www.bic.mni.mcgill.ca) TT_EPI+tlrc -> EPI template from spm2, masked as TT_avg152T1 TT_avg152 and TT_EPI volume sources are from SPM's distribution. (www.fil.ion.ucl.ac.uk/spm/) If you do not specify a path for the template, the script will attempt to locate the template AFNI's binaries directory. NOTE: These datasets have been slightly modified from their original size to match the standard TLRC dimensions (Jean Talairach and Pierre Tournoux Co-Planar Stereotaxic Atlas of the Human Brain Thieme Medical Publishers, New York, 1988). That was done for internal consistency in AFNI. You may use the original form of these volumes if you choose but your TLRC coordinates will not be consistent with AFNI's TLRC database (San Antonio Talairach Daemon database), for example. Required.</p>																														
in_file	<p>Character; file path. Original anatomical volume (+orig).The skull is removed by this script unless instructed otherwise (-no_ss). Required.</p>																														
args	<p>Character. Additional parameters to the command</p>																														
no_ss	<p>Logical. Do not strip skull of input data set (because skull has already been removed or because template still has the skull) NOTE: The -no_ss option is not all that optional. Here is a table of when you should and should not use -no_ss</p> <table border="1"> <thead> <tr> <th>Dataset</th> <th>Template</th> <th>w/ skull</th> <th>wo/ skull</th> <th>WITH skull</th> <th>-no_ss</th> </tr> </thead> <tbody> <tr> <td>xxx</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table> <p>Template means: Your template of choice Dset. means: Your anatomical dataset -no_ss means: Skull stripping should not be attempted on Dset xxx means: Don't put anything, the script will strip Dset No Cigar means: Don't try that combination, it makes no sense.</p>	Dataset	Template	w/ skull	wo/ skull	WITH skull	-no_ss	xxx																							
Dataset	Template	w/ skull	wo/ skull	WITH skull	-no_ss																										
xxx																															
.cwd	<p>Working directory override.</p>																														
.env	<p>Named character vector of environment variables.</p>																														
.engine	<p>Execution engine override.</p>																														
.profile	<p>Runtime profile override.</p>																														
dry_run	<p>Logical; preview command without executing.</p>																														
echo	<p>Logical; echo stdout/stderr in real time.</p>																														

Value

An ni_result object.

ni_afni_autobox	<i>AFNI Autobox</i>
-----------------	---------------------

Description

Computes size of a box that fits around the volume.

Usage

```
ni_afni_autobox(
  in_file,
  args = NULL,
  no_clustering = NULL,
  out_file = NULL,
  padding = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. input file Required .
args	Character. Additional parameters to the command
no_clustering	Logical. Don't do any clustering to find box. Any non-zero voxel will be preserved in the cropped volume. The default method uses some clustering to find the cropping box, and will clip off small isolated blobs.
out_file	Character; file path
padding	Integer. Number of extra voxels to pad on each side of box
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_automask	<i>AFNI Automask</i>
------------------	----------------------

Description

Create a brain-only mask of the image using AFNI 3dAutomask command

Usage

```
ni_afni_automask(
  in_file,
  args = NULL,
  brain_file = NULL,
  clfrac = NULL,
  dilate = NULL,
  erode = NULL,
  out_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. input file to 3dAutomask Required.
args	Character. Additional parameters to the command
brain_file	Character; file path. output file from 3dAutomask
clfrac	Numeric. sets the clip level fraction (must be 0.1-0.9). A small value will tend to make the mask larger [default = 0.5].
dilate	Integer. dilate the mask outwards
erode	Integer. erode the mask inwards
out_file	Character; file path. output image file name
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

 ni_afni_axialize *AFNI Axialize*

Description

Read in a dataset and write it out as a new dataset

Usage

```
ni_afni_axialize(
  in_file,
  args = NULL,
  axial = NULL,
  coronal = NULL,
  orientation = NULL,
  out_file = NULL,
  sagittal = NULL,
  verb = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. input file to 3daxialize Required.
args	Character. Additional parameters to the command
axial	Logical. Do axial slice order [-orient RAI]This is the default AFNI axial order, andis the one currently required by thevolume rendering plugin; this is alsothe default orientation output by thisprogram (hence the program's name).
coronal	Logical. Do coronal slice order [-orient RSA]
orientation	Character. new orientation code
out_file	Character; file path. output image file name
sagittal	Logical. Do sagittal slice order [-orient ASL]
verb	Logical. Print out a progerss report
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_bandpass	<i>AFNI Bandpass</i>
------------------	----------------------

Description

Program to lowpass and/or highpass each voxel time series in a

Usage

```
ni_afni_bandpass(
  highpass,
  in_file,
  lowpass,
  args = NULL,
  automask = NULL,
  blur = NULL,
  despike = NULL,
  localPV = NULL,
  mask = NULL,
  nfft = NULL,
  no_detrend = NULL,
  normalize = NULL,
  notrans = NULL,
  orthogonalize_dset = NULL,
  orthogonalize_file = NULL,
  out_file = NULL,
  tr = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

highpass	Numeric. highpass Required.
in_file	Character; file path. input file to 3dBandpass Required.
lowpass	Numeric. lowpass Required.
args	Character. Additional parameters to the command
automask	Logical. Create a mask from the input dataset.

blur	Numeric. Blur (inside the mask only) with a filter width (FWHM) of 'fff' millimeters.
despike	Logical. Despike each time series before other processing. Hopefully, you don't actually need to do this, which is why it is optional.
localPV	Numeric. Replace each vector by the local Principal Vector (AKA first singular vector) from a neighborhood of radius 'rrr' millimeters. Note that the PV time series is L2 normalized. This option is mostly for Bob Cox to have fun with.
mask	Character; file path. mask file
nfft	Integer. Set the FFT length [must be a legal value].
no_detrend	Logical. Skip the quadratic detrending of the input that occurs before the FFT-based bandpassing. You would only want to do this if the dataset had been detrended already in some other program.
normalize	Logical. Make all output time series have L2 norm = 1 (i.e., sum of squares = 1).
notrans	Logical. Don't check for initial positive transients in the data. The test is a little slow, so skipping it is OK, if you KNOW the data time series are transient-free.
orthogonalize_dset	Character; file path. Orthogonalize each voxel to the corresponding voxel time series in dataset 'fset', which must have the same spatial and temporal grid structure as the main input dataset. At present, only one '-dsort' option is allowed.
orthogonalize_file	Character or numeric vector. Also orthogonalize input to columns in f.1D. Multiple '-ort' options are allowed.
out_file	Character; file path. output file from 3dBandpass
tr	Numeric. Set time step (TR) in sec [default=from dataset header].
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

 ni_afni_blur_in_mask *AFNI BlurInMask*

Description

Blurs a dataset spatially inside a mask. That's all. Experimental.

Usage

```
ni_afni_blur_in_mask(
  fwhm,
  in_file,
  args = NULL,
  automask = NULL,
  float_out = NULL,
  mask = NULL,
  multimask = NULL,
  options = NULL,
  out_file = NULL,
  preserve = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

fwhm	Numeric. fwhm kernel size Required.
in_file	Character; file path. input file to 3dSkullStrip Required.
args	Character. Additional parameters to the command
automask	Logical. Create an automask from the input dataset.
float_out	Logical. Save dataset as floats, no matter what the input data type is.
mask	Character; file path. Mask dataset, if desired. Blurring will occur only within the mask. Voxels NOT in the mask will be set to zero in the output.
multimask	Character; file path. Multi-mask dataset – each distinct nonzero value in dataset will be treated as a separate mask for blurring purposes.
options	Character. options
out_file	Character; file path. output to the file
preserve	Logical. Normally, voxels not in the mask will be set to zero in the output. If you want the original values in the dataset to be preserved in the output, use this option.

.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

ni_afni_blur_to_fwhm *AFNI BlurToFWHM*

Description

Blurs a 'master' dataset until it reaches a specified FWHM smoothness

Usage

```
ni_afni_blur_to_fwhm(
  in_file,
  args = NULL,
  automask = NULL,
  blurmaster = NULL,
  fwhm = NULL,
  fwhmxy = NULL,
  mask = NULL,
  out_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

<code>in_file</code>	Character; file path. The dataset that will be smoothed Required .
<code>args</code>	Character. Additional parameters to the command
<code>automask</code>	Logical. Create an automask from the input dataset.
<code>blurmaster</code>	Character; file path. The dataset whose smoothness controls the process.
<code>fwhm</code>	Numeric. Blur until the 3D FWHM reaches this value (in mm)

fwhmxy	Numeric. Blur until the 2D (x,y)-plane FWHM reaches this value (in mm)
mask	Character; file path. Mask dataset, if desired. Voxels NOT in mask will be set to zero in output.
out_file	Character; file path. output image file name
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_brick_stat	<i>AFNI BrickStat</i>
--------------------	-----------------------

Description

Computes maximum and/or minimum voxel values of an input dataset.

Usage

```
ni_afni_brick_stat(
  in_file,
  args = NULL,
  mask = NULL,
  max = NULL,
  mean = NULL,
  min = NULL,
  percentile = NULL,
  slow = NULL,
  sum = NULL,
  var = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. input file to 3dmaskave Required.
args	Character. Additional parameters to the command
mask	Character; file path. -mask dset = use dset as mask to include/exclude voxels
max	Logical. print the maximum value in the dataset
mean	Logical. print the mean value in the dataset
min	Logical. print the minimum value in dataset
percentile	Character or numeric vector. p0 ps p1 write the percentile values starting at p0% and ending at p1% at a step of ps%. only one sub-brick is accepted.
slow	Logical. read the whole dataset to find the min and max values
sum	Logical. print the sum of values in the dataset
var	Logical. print the variance in the dataset
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_bucket	<i>AFNI Bucket</i>
----------------	--------------------

Description

Concatenate sub-bricks from input datasets into one big

Usage

```
ni_afni_bucket(
  in_file,
  args = NULL,
  out_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character or numeric vector. List of tuples of input datasets and subbrick selection strings as described in more detail in the following afni help string Input dataset specified using one of these forms: prefix+view, prefix+view.HEAD, or prefix+view.BRIK. You can also add a sub-brick selection list after the end of the dataset name. This allows only a subset of the sub-bricks to be included into the output (by default, all of the input dataset is copied into the output). A sub-brick selection list looks like one of the following forms:: fred+orig[5] ==> use only sub-brick #5 fred+orig[5,9,17] ==> use #5, #9, and #17 fred+orig[5..8] or [5-8] ==> use #5, #6, #7, and #8 fred+orig[5..13(2)] or [5-13(2)] ==> use #5, #7, #9, #11, and #13 Sub-brick indexes start at 0. You can use the character '\$' to indicate the last sub-brick in a dataset; for example, you can select every third sub-brick by using the selection list fred+orig\[0..\$(3)\] N.B.: The sub-bricks are output in the order specified, which may not be the order in the original datasets. For example, using fred+orig\[0..\$(2),1..\$(2)\] will cause the sub-bricks in fred+orig to be output into the new dataset in an interleaved fashion. Using fred+orig\[\$.0\] will reverse the order of the sub-bricks in the output. N.B.: Bucket datasets have multiple sub-bricks, but do NOT have a time dimension. You can input sub-bricks from a 3D+time dataset into a bucket dataset. You can use the '3dinfo' program to see how many sub-bricks a 3D+time or a bucket dataset contains. N.B.: In non-bucket functional datasets (like the 'fico' datasets output by FIM, or the 'fitt' datasets output by 3dtttest), sub-brick \[0\] is the 'intensity' and sub-brick [1] is the statistical parameter used as a threshold. Thus, to create a bucket dataset using the intensity from dataset A and the threshold from dataset B, and calling the output dataset C, you would type:: 3dbucket -prefix C -fbuc 'A+orig[0]' -fbuc 'B+orig[1] Required.
args	Character. Additional parameters to the command
out_file	Character; file path
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_calc

AFNI Calc

Description

This program does voxel-by-voxel arithmetic on 3D datasets.

Usage

```
ni_afni_calc(
  expr,
  in_file_a,
  args = NULL,
  in_file_b = NULL,
  in_file_c = NULL,
  out_file = NULL,
  overwrite = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

expr	Character. expr Required.
in_file_a	Character; file path. input file to 3dcalc Required.
args	Character. Additional parameters to the command
in_file_b	Character; file path. operand file to 3dcalc
in_file_c	Character; file path. operand file to 3dcalc
out_file	Character; file path. output image file name
overwrite	Logical. overwrite output
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

 ni_afni_cat

 AFNI Cat

Description

lcat takes as input one or more 1D files, and writes out a 1D file

Usage

```
ni_afni_cat(
  in_files,
  out_file,
  args = NULL,
  keepfree = NULL,
  omitconst = NULL,
  out_double = NULL,
  out_fint = NULL,
  out_format = NULL,
  out_int = NULL,
  out_nice = NULL,
  sel = NULL,
  stack = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_files	Character or numeric vector Required.
out_file	Character; file path. output (concatenated) file name Required.
args	Character. Additional parameters to the command
keepfree	Logical. Keep only columns that are marked as 'free' in the 3dAllineate header from '-1Dparam_save'. If there is no such header, all columns are kept.
omitconst	Logical. Omit columns that are identically constant from output.
out_double	Logical. specify double data type for output
out_fint	Logical. specify int, rounded down, data type for output
out_format	Character; one of: "int", "nice", "double", "fint", "cint". specify data type for output.
out_int	Logical. specify int data type for output
out_nice	Logical. specify nice data type for output

sel	Character. Apply the same column/row selection string to all filenames on the command line.
stack	Logical. Stack the columns of the resultant matrix in the output.
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_cat_matvec	<i>AFNI CatMatvec</i>
--------------------	-----------------------

Description

Catenates 3D rotation+shift matrix+vector transformations.

Usage

```
ni_afni_cat_matvec(
  in_file,
  out_file,
  args = NULL,
  fourxfour = NULL,
  matrix = NULL,
  oneline = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character or numeric vector. list of tuples of mfiles and associated opkeys Required.
out_file	Character; file path. File to write concattenated matvecs to Required.
args	Character. Additional parameters to the command

fourxfour	Logical. Output matrix in augmented form (last row is 0 0 0 1) This option does not work with -MATRIX or -ONELINE
matrix	Logical. indicates that the resulting matrix will be written to outfile in the 'MATRIX(...)' format (FORM 3). This feature could be used, with clever scripting, to input a matrix directly on the command line to program 3dWarp.
oneline	Logical. indicates that the resulting matrix will simply be written as 12 numbers on one line.
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_center_mass *AFNI CenterMass*

Description

Computes center of mass using 3dCM command

Usage

```
ni_afni_center_mass(
  in_file,
  all_rois = NULL,
  args = NULL,
  automask = NULL,
  cm_file = NULL,
  local_ijk = NULL,
  mask_file = NULL,
  roi_vals = NULL,
  set_cm = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. input file to 3dCM Required.
all_rois	Logical. Don't bother listing the values of ROIs you want: The program will find all of them and produce a full list
args	Character. Additional parameters to the command
automask	Logical. Generate the mask automatically
cm_file	Character; file path. File to write center of mass to
local_ijk	Logical. Output values as (i,j,k) in local orientation
mask_file	Character; file path. Only voxels with nonzero values in the provided mask will be averaged.
roi_vals	Character or numeric vector. Compute center of mass for each blob with voxel value of v0, v1, v2, etc. This option is handy for getting ROI centers of mass.
set_cm	Character or numeric vector. After computing the center of mass, set the origin fields in the header so that the center of mass will be at (x,y,z) in DICOM coords.
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_clip_level *AFNI ClipLevel*

Description

Estimates the value at which to clip the anatomical dataset so

Usage

```
ni_afni_clip_level(
  in_file,
  args = NULL,
  doall = NULL,
  grad = NULL,
  mfrac = NULL,
  .cwd = NULL,
  .env = NULL,
```

```

    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

in_file	Character; file path. input file to 3dClipLevel Required.
args	Character. Additional parameters to the command
doall	Logical. Apply the algorithm to each sub-brick separately.
grad	Character; file path. Also compute a 'gradual' clip level as a function of voxel position, and output that to a dataset.
mfrac	Numeric. Use the number ff instead of 0.50 in the algorithm
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_convert_dset *AFNI ConvertDset*

Description

Converts a surface dataset from one format to another.

Usage

```

ni_afni_convert_dset(
  in_file,
  out_file,
  out_type,
  args = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)

```

Arguments

in_file	Character; file path. input file to ConvertDset Required.
out_file	Character; file path. output file for ConvertDset Required.
out_type	Character; one of: "niml", "niml_asc", "niml_bi", "1D", "1Dp", "1Dpt", "gii", "gii_asc", "gii_b64", "gii_b64gz". output type Required.
args	Character. Additional parameters to the command
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_copy

AFNI Copy

Description

Copies an image of one type to an image of the same

Usage

```
ni_afni_copy(
  in_file,
  args = NULL,
  out_file = NULL,
  verbose = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. input file to 3dcopy Required.
args	Character. Additional parameters to the command
out_file	Character; file path. output image file name
verbose	Logical. print progress reports
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_deconvolve *AFNI Deconvolve*

Description

Performs OLS regression given a 4D neuroimage file and stimulus timings

Usage

```
ni_afni_deconvolve(
  STATmask = NULL,
  TR_1D = NULL,
  allzero_OK = NULL,
  args = NULL,
  automask = NULL,
  cbucket = NULL,
  censor = NULL,
  dmbase = NULL,
  dname = NULL,
  force_TR = NULL,
  fout = NULL,
  global_times = NULL,
  glt_label = NULL,
  gltsym = NULL,
  goforit = NULL,
  in_files = NULL,
  input1D = NULL,
  legendre = NULL,
```

```

local_times = NULL,
mask = NULL,
noblock = NULL,
nocond = NULL,
nodmbase = NULL,
nofdr = NULL,
nolegendre = NULL,
nosvd = NULL,
num_glt = NULL,
num_stimts = NULL,
num_threads = NULL,
ortvec = NULL,
out_file = NULL,
polort = NULL,
rmsmin = NULL,
rout = NULL,
sat = NULL,
singvals = NULL,
stim_label = NULL,
stim_times = NULL,
stim_times_subtract = NULL,
svd = NULL,
tout = NULL,
trans = NULL,
vout = NULL,
x1D = NULL,
x1D_stop = NULL,
.cwd = NULL,
.env = NULL,
.engine = NULL,
.profile = NULL,
dry_run = FALSE,
echo = interactive()
)

```

Arguments

STATmask	Character; file path. build a mask from provided file, and use this mask for the purpose of reporting truncation-to float issues AND for computing the FDR curves. The actual results ARE not masked with this option (only with 'mask' or 'automask' options).
TR_1D	Numeric. TR to use with 'input1D'. This option has no effect if you do not also use 'input1D'.
allzero_OK	Logical. don't consider all zero matrix columns to be the type of error that 'gotforit' is needed to ignore.
args	Character. Additional parameters to the command
automask	Logical. build a mask automatically from input data (will be slow for long time series datasets)

cbucket	Character. Name for dataset in which to save the regression coefficients (no statistics). This dataset will be used in a -xrestore run [not yet implemented] instead of the bucket dataset, if possible.
cancel	Character; file path. filename of cancel .1D time series. This is a file of 1s and 0s, indicating which time points are to be included (1) and which are to be excluded (0).
dmbase	Logical. de-mean baseline time series (default if 'polort' >= 0)
dname	Character or numeric vector. set environmental variable to provided value
force_TR	Numeric. use this value instead of the TR in the 'input' dataset. (It's better to fix the input using Refit.)
fout	Logical. output F-statistic for each stimulus
global_times	Logical. use global timing for stimulus timing files
glt_label	Character or numeric vector. general linear test (i.e., contrast) labels
gltsym	Character or numeric vector. general linear tests (i.e., contrasts) using symbolic conventions (e.g., '+Label1 -Label2')
goforit	Integer. use this to proceed even if the matrix has bad problems (e.g., duplicate columns, large condition number, etc.).
in_files	Character or numeric vector. filenames of 3D+time input datasets. More than one filename can be given and the datasets will be auto-catenated in time. You can input a 1D time series file here, but the time axis should run along the ROW direction, not the COLUMN direction as in the 'input1D' option.
input1D	Character; file path. filename of single (fMRI) .1D time series where time runs down the column.
legendre	Logical. use Legendre polynomials for null hypothesis (baseline model)
local_times	Logical. use local timing for stimulus timing files
mask	Character; file path. filename of 3D mask dataset; only data time series from within the mask will be analyzed; results for voxels outside the mask will be set to zero.
noblock	Logical. normally, if you input multiple datasets with 'input', then the separate datasets are taken to be separate image runs that get separate baseline models. Use this options if you want to have the program consider these to be all one big run.* If any of the input dataset has only 1 sub-brick, then this option is automatically invoked!* If the auto-catenation feature isn't used, then this option has no effect, no how, no way.
nocond	Logical. DON'T calculate matrix condition number
nodmbase	Logical. don't de-mean baseline time series
nofdr	Logical. Don't compute the statistic-vs-FDR curves for the bucket dataset.
nolegendre	Logical. use power polynomials for null hypotheses. Don't do this unless you are crazy!
nosvd	Logical. use Gaussian elimination instead of SVD
num_glt	Integer. number of general linear tests (i.e., contrasts)
num_stimts	Integer. number of stimulus timing files

num_threads	Integer. run the program with provided number of sub-processes
ortvec	Character or numeric vector. this option lets you input a rectangular array of 1 or more baseline vectors from a file. This method is a fast way to include a lot of baseline regressors in one step.
out_file	Character; file path. output statistics file
polort	Integer. degree of polynomial corresponding to the null hypothesis [default: 1]
rmsmin	Numeric. minimum rms error to reject reduced model (default = 0; don't use this option normally!)
rout	Logical. output the R ² statistic for each stimulus
sat	Logical. check the dataset time series for initial saturation transients, which should normally have been excised before data analysis.
singvals	Logical. print out the matrix singular values
stim_label	Character or numeric vector. label for kth input stimulus (e.g., Label1)
stim_times	Character or numeric vector. generate a response model from a set of stimulus times given in file.
stim_times_subtract	Numeric. this option means to subtract specified seconds from each time encountered in any 'stim_times' option. The purpose of this option is to make it simple to adjust timing files for the removal of images from the start of each imaging run.
svd	Logical. use SVD instead of Gaussian elimination (default)
tout	Logical. output the T-statistic for each stimulus
trans	Logical. check the dataset time series for initial saturation transients, which should normally have been excised before data analysis.
vout	Logical. output the sample variance (MSE) for each stimulus
x1D	Character; file path. specify name for saved X matrix
x1D_stop	Logical. stop running after writing .xmat.1D file
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

```
ni_afni_degree_centrality
    AFNI DegreeCentrality
```

Description

Performs degree centrality on a dataset using a given maskfile

Usage

```
ni_afni_degree_centrality(
    in_file,
    args = NULL,
    autoclip = NULL,
    automask = NULL,
    mask = NULL,
    oned_file = NULL,
    out_file = NULL,
    polort = NULL,
    sparsity = NULL,
    thresh = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)
```

Arguments

<code>in_file</code>	Character; file path. input file to 3dDegreeCentrality Required.
<code>args</code>	Character. Additional parameters to the command
<code>autoclip</code>	Logical. Clip off low-intensity regions in the dataset
<code>automask</code>	Logical. Mask the dataset to target brain-only voxels
<code>mask</code>	Character; file path. mask file to mask input data
<code>oned_file</code>	Character. output filepath to text dump of correlation matrix
<code>out_file</code>	Character; file path. output image file name
<code>polort</code>	Integer
<code>sparsity</code>	Numeric. only take the top percent of connections
<code>thresh</code>	Numeric. threshold to exclude connections where corr <= thresh
<code>.cwd</code>	Working directory override.
<code>.env</code>	Named character vector of environment variables.

.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_despike	<i>AFNI Despike</i>
-----------------	---------------------

Description

Removes 'spikes' from the 3D+time input dataset

Usage

```
ni_afni_despike(
  in_file,
  args = NULL,
  out_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. input file to 3dDespike Required.
args	Character. Additional parameters to the command
out_file	Character; file path. output image file name
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_detrend *AFNI Detrend*

Description

This program removes components from voxel time series using

Usage

```
ni_afni_detrend(  
  in_file,  
  args = NULL,  
  out_file = NULL,  
  .cwd = NULL,  
  .env = NULL,  
  .engine = NULL,  
  .profile = NULL,  
  dry_run = FALSE,  
  echo = interactive()  
)
```

Arguments

in_file	Character; file path. input file to 3dDetrend Required.
args	Character. Additional parameters to the command
out_file	Character; file path. output image file name
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_dot

AFNI Dot

Description

Correlation coefficient between sub-brick pairs.

Usage

```
ni_afni_dot(
  args = NULL,
  demean = NULL,
  docoef = NULL,
  docor = NULL,
  dodice = NULL,
  dodot = NULL,
  doeta2 = NULL,
  dosums = NULL,
  full = NULL,
  in_files = NULL,
  mask = NULL,
  mrange = NULL,
  out_file = NULL,
  show_labels = NULL,
  upper = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

args	Character. Additional parameters to the command
demean	Logical. Remove the mean from each volume prior to computing the correlation
docoef	Logical. Return the least square fit coefficients $\{ \{a,b\} \}$ so that $dset2$ is approximately $a + b \cdot dset1$
docor	Logical. Return the correlation coefficient (default).
dodice	Logical. Return the Dice coefficient (the Sorensen-Dice index).
dodot	Logical. Return the dot product (unscaled).
doeta2	Logical. Return eta-squared (Cohen, NeuroImage 2008).
dosums	Logical. Return the 6 numbers $xbar= \langle x-xbar \rangle^2$ $ybar= \langle y-ybar \rangle^2$ $\langle (x-xbar)(y-ybar) \rangle$ and the correlation coefficient.

full	Logical. Compute the whole matrix. A waste of time, but handy for parsing.
in_files	Character or numeric vector. list of input files, possibly with subbrick selectors
mask	Character; file path. Use this dataset as a mask
mrangle	Character or numeric vector. Means to further restrict the voxels from 'mset' so that only those mask values within this range (inclusive) will be used.
out_file	Character; file path. collect output to a file
show_labels	Logical. Print sub-brick labels to help identify what is being correlated. This option is useful when you have more than 2 sub-bricks at input.
upper	Logical. Compute upper triangular matrix
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

ni_afni_ecm

AFNI ECM

Description

Performs degree centrality on a dataset using a given maskfile

Usage

```
ni_afni_ecm(
  in_file,
  args = NULL,
  autoclip = NULL,
  automask = NULL,
  eps = NULL,
  fecm = NULL,
  full = NULL,
  mask = NULL,
  max_iter = NULL,
  memory = NULL,
  out_file = NULL,
  polort = NULL,
  scale = NULL,
  shift = NULL,
```

```

    sparsity = NULL,
    thresh = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

in_file	Character; file path. input file to 3dECM Required.
args	Character. Additional parameters to the command
autoclip	Logical. Clip off low-intensity regions in the dataset
automask	Logical. Mask the dataset to target brain-only voxels
eps	Numeric. sets the stopping criterion for the power iteration; $ v_{old} - v_{new} < \text{eps}$ default = 0.001
fecm	Logical. Fast centrality method; substantial speed increase but cannot accommodate thresholding; automatically selected if -thresh or -sparsity are not set
full	Logical. Full power method; enables thresholding; automatically selected if -thresh or -sparsity are set
mask	Character; file path. mask file to mask input data
max_iter	Integer. sets the maximum number of iterations to use in the power iteration; default = 1000
memory	Numeric. Limit memory consumption on system by setting the amount of GB to limit the algorithm to; default = 2GB
out_file	Character; file path. output image file name
polort	Integer
scale	Numeric. scale correlation coefficients in similarity matrix to after shifting, $x \geq 0.0$; default = 1.0 for -full, 0.5 for -fecm
shift	Numeric. shift correlation coefficients in similarity matrix to enforce non-negativity, $s \geq 0.0$; default = 0.0 for -full, 1.0 for -fecm
sparsity	Numeric. only take the top percent of connections
thresh	Numeric. threshold to exclude connections where $\text{corr} \leq \text{thresh}$
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

 ni_afni_edge3

 AFNI Edge3

Description

Does 3D Edge detection using the library 3DEdge

Usage

```
ni_afni_edge3(
  in_file,
  args = NULL,
  datum = NULL,
  fscale = NULL,
  gscale = NULL,
  nscale = NULL,
  out_file = NULL,
  scale_floats = NULL,
  verbose = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. input file to 3dedge3 Required.
args	Character. Additional parameters to the command
datum	Character; one of: "byte", "short", "float". specify data type for output. Valid types are 'byte', 'short' and 'float'.
fscale	Logical. Force scaling of the output to the maximum integer range.
gscale	Logical. Same as '-fscale', but also forces each output sub-brick to to get the same scaling factor.
nscale	Logical. Don't do any scaling on output to byte or short datasets.
out_file	Character; file path. output image file name
scale_floats	Numeric. Multiply input by VAL, but only if the input datum is float. This is needed when the input dataset has a small range, like 0 to 2.0 for instance. With such a range, very few edges are detected due to what I suspect to be truncation problems. Multiplying such a dataset by 10000 fixes the problem and the scaling is undone at the output.
verbose	Logical. Print out some information along the way.

.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_eval	<i>AFNI Eval</i>
--------------	------------------

Description

Evaluates an expression that may include columns of data from one or

Usage

```
ni_afni_eval(
  expr,
  in_file_a,
  args = NULL,
  in_file_b = NULL,
  in_file_c = NULL,
  out1D = NULL,
  out_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

expr	Character. expr Required.
in_file_a	Character; file path. input file to ldeval Required.
args	Character. Additional parameters to the command
in_file_b	Character; file path. operand file to ldeval
in_file_c	Character; file path. operand file to ldeval
out1D	Logical. output in 1D

out_file	Character; file path. output image file name
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_fim	<i>AFNI Fim</i>
-------------	-----------------

Description

Program to calculate the cross-correlation of an ideal reference

Usage

```
ni_afni_fim(
  ideal_file,
  in_file,
  args = NULL,
  fim_thr = NULL,
  out = NULL,
  out_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

ideal_file	Character; file path. ideal time series file name Required.
in_file	Character; file path. input file to 3dfim+ Required.
args	Character. Additional parameters to the command
fim_thr	Numeric. fim internal mask threshold value
out	Character. Flag to output the specified parameter
out_file	Character; file path. output image file name

.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_fourier	<i>AFNI Fourier</i>
-----------------	---------------------

Description

Program to lowpass and/or highpass each voxel time series in a

Usage

```
ni_afni_fourier(
  highpass,
  in_file,
  lowpass,
  args = NULL,
  out_file = NULL,
  retrend = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

highpass	Numeric. highpass Required.
in_file	Character; file path. input file to 3dFourier Required.
lowpass	Numeric. lowpass Required.
args	Character. Additional parameters to the command
out_file	Character; file path. output image file name
retrend	Logical. Any mean and linear trend are removed before filtering. This will restore the trend after filtering.

.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_fwh_mx	<i>AFNI FWHMx</i>
----------------	-------------------

Description

Unlike the older 3dFWHM, this program computes FWHMs for all sub-bricks

Usage

```
ni_afni_fwh_mx(
  in_file,
  acf = FALSE,
  args = NULL,
  arith = NULL,
  automask = FALSE,
  combine = NULL,
  compat = NULL,
  demed = NULL,
  detrend = FALSE,
  geom = NULL,
  mask = NULL,
  out_detrend = NULL,
  out_file = NULL,
  out_subbricks = NULL,
  unif = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. input dataset Required.
acf	Character. computes the spatial autocorrelation
args	Character. Additional parameters to the command
arith	Logical. if in_file has more than one sub-brick, compute the final estimate as the arithmetic mean of the individual sub-brick FWHM estimates
automask	Logical. compute a mask from THIS dataset, a la 3dAutomask
combine	Logical. combine the final measurements along each axis
compat	Logical. be compatible with the older 3dFWHM
demed	Logical. If the input dataset has more than one sub-brick (e.g., has a time axis), then subtract the median of each voxel's time series before processing FWHM. This will tend to remove intrinsic spatial structure and leave behind the noise.
detrend	Character. instead of demed (0th order detrending), detrend to the specified order. If order is not given, the program picks $q=NT/30$. -detrend disables -demed, and includes -unif.
geom	Logical. if in_file has more than one sub-brick, compute the final estimate as the geometric mean of the individual sub-brick FWHM estimates
mask	Character; file path. use only voxels that are nonzero in mask
out_detrend	Character; file path. Save the detrended file into a dataset
out_file	Character; file path. output file
out_subbricks	Character; file path. output file listing the subbricks FWHM
unif	Logical. If the input dataset has more than one sub-brick, then normalize each voxel's time series to have the same MAD before processing FWHM.
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

 ni_afni_gcor

 AFNI GCOR

Description

Computes the average correlation between every voxel

Usage

```
ni_afni_gcor(
  in_file,
  args = NULL,
  mask = NULL,
  nfirst = NULL,
  no_demean = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. input dataset to compute the GCOR over Required .
args	Character. Additional parameters to the command
mask	Character; file path. mask dataset, for restricting the computation
nfirst	Integer. specify number of initial TRs to ignore
no_demean	Logical. do not (need to) demean as first step
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_hist

AFNI Hist

Description

Computes average of all voxels in the input dataset

Usage

```
ni_afni_hist(
  in_file,
  args = NULL,
  bin_width = NULL,
  mask = NULL,
  max_value = NULL,
  min_value = NULL,
  nbin = NULL,
  out_file = NULL,
  out_show = NULL,
  showhist = FALSE,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. input file to 3dHist Required.
args	Character. Additional parameters to the command
bin_width	Numeric. bin width
mask	Character; file path. matrix to align input file
max_value	Numeric. maximum intensity value
min_value	Numeric. minimum intensity value
nbin	Integer. number of bins
out_file	Character; file path. Write histogram to nimgl file with this prefix
out_show	Character; file path. output image file name
showhist	Logical. write a text visual histogram
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.

.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_lfcd	<i>AFNI LFCD</i>
--------------	------------------

Description

Performs degree centrality on a dataset using a given maskfile

Usage

```
ni_afni_lfcd(
  in_file,
  args = NULL,
  autoclip = NULL,
  automask = NULL,
  mask = NULL,
  out_file = NULL,
  polort = NULL,
  thresh = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. input file to 3dLFCD Required.
args	Character. Additional parameters to the command
autoclip	Logical. Clip off low-intensity regions in the dataset
automask	Logical. Mask the dataset to target brain-only voxels
mask	Character; file path. mask file to mask input data
out_file	Character; file path. output image file name
polort	Integer
thresh	Numeric. threshold to exclude connections where corr <= thresh

.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_local_bistat *AFNI LocalBistat*

Description

3dLocalBistat - computes statistics between 2 datasets, at each voxel,

Usage

```
ni_afni_local_bistat(
  in_file1,
  in_file2,
  neighborhood,
  stat,
  args = NULL,
  automask = NULL,
  mask_file = NULL,
  out_file = NULL,
  weight_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file1	Character; file path. Filename of the first image Required.
in_file2	Character; file path. Filename of the second image Required.
neighborhood	Character or numeric vector. The region around each voxel that will be extracted for the statistics calculation. Possible regions are: 'SPHERE', 'RHDD' (rhombic dodecahedron), 'TOHD' (truncated octahedron) with a given radius in mm or 'RECT' (rectangular block) with dimensions to specify in mm. Required.

stat	Character or numeric vector. Statistics to compute. Possible names are: * pearson = Pearson correlation coefficient * spearman = Spearman correlation coefficient * quadrant = Quadrant correlation coefficient * mutinfo = Mutual Information * normuti = Normalized Mutual Information * jointent = Joint entropy * hellinger= Hellinger metric * crU = Correlation ratio (Unsymmetric) * crM = Correlation ratio (symmetrized by Multiplication) * crA = Correlation ratio (symmetrized by Addition) * L2slope = slope of least-squares (L2) linear regression of the data from dataset1 vs. the dataset2 (i.e., $d2 = a + b*d1$ ==> this is 'b') * L1slope = slope of least-absolute-sum (L1) linear regression of the data from dataset1 vs. the dataset2 * num = number of the values in the region: with the use of -mask or -automask, the size of the region around any given voxel will vary; this option lets you map that size. * ALL = all of the above, in that order More than one option can be used. Required.
args	Character. Additional parameters to the command
automask	Logical. Compute the mask as in program 3dAutomask.
mask_file	Character; file path. mask image file name. Voxels NOT in the mask will not be used in the neighborhood of any voxel. Also, a voxel NOT in the mask will have its statistic(s) computed as zero (0).
out_file	Character; file path. Output dataset.
weight_file	Character; file path. File name of an image to use as a weight. Only applies to 'pearson' statistics.
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_localstat	<i>AFNI Localstat</i>
-------------------	-----------------------

Description

3dLocalstat - computes statistics at each voxel,

Usage

```

ni_afni_localstat(
  in_file,
  neighborhood,
  stat,
  args = NULL,
  automask = NULL,
  grid_rmode = NULL,
  mask_file = NULL,
  nonmask = NULL,
  out_file = NULL,
  overwrite = NULL,
  quiet = NULL,
  reduce_grid = NULL,
  reduce_max_vox = NULL,
  reduce_restore_grid = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)

```

Arguments

in_file	Character; file path. input dataset Required.
neighborhood	Character or numeric vector. The region around each voxel that will be extracted for the statistics calculation. Possible regions are: 'SPHERE', 'RHDD' (rhombic dodecahedron), 'TOHD' (truncated octahedron) with a given radius in mm or 'RECT' (rectangular block) with dimensions to specify in mm. Required.
stat	Character or numeric vector. statistics to compute. Possible names are: * mean = average of the values * stdev = standard deviation * var = variance (stdev\stdev) * cvar = coefficient of variation = stdev/fabs(mean) * median = median of the values * MAD = median absolute deviation * min = minimum * max = maximum * absmax = maximum of the absolute values * num = number of the values in the region: with the use of -mask or -automask, the size of the region around any given voxel will vary; this option lets you map that size. It may be useful if you plan to compute a t-statistic (say) from the mean and stdev outputs. * sum = sum of the values in the region * FWHM = compute (like 3dFWHM) image smoothness inside each voxel's neighborhood. Results are in 3 sub-bricks: FWHMx, FWHMy, and FWHMz. Places where an output is -1 are locations where the FWHM value could not be computed (e.g., outside the mask). * FWHMbar = Compute just the average of the 3 FWHM values (normally would NOT do this with FWHM also). * perc:P0:P1:Pstep = Compute percentiles between P0 and P1 with a step of Pstep. Default P1 is equal to P0 and default P2 = 1 * rank = rank of the voxel's intensity * frank = rank / number of voxels in neighborhood * P2skew = Pearson's second skewness coefficient $3 \sqrt{(\text{mean} - \text{median}) /$

stdev * ALL = all of the above, in that order (except for FWHMbar and perc). *
 mMP2s = Exactly the same output as: median, MAD, P2skew, but a little faster
 * mmMP2s = Exactly the same output as: mean, median, MAD, P2skew More than one option can be used. **Required.**

args	Character. Additional parameters to the command
automask	Logical. Compute the mask as in program 3dAutomask.
grid_rmode	Character; one of: "NN", "Li", "Cu", "Bk". Interpolant to use when resampling the output with the reduce_restore_grid option. The resampling method string RESAM should come from the set {'NN', 'Li', 'Cu', 'Bk'}. These stand for 'Nearest Neighbor', 'Linear', 'Cubic', and 'Blocky' interpolation, respectively.
mask_file	Character; file path. Mask image file name. Voxels NOT in the mask will not be used in the neighborhood of any voxel. Also, a voxel NOT in the mask will have its statistic(s) computed as zero (0) unless the parameter 'nonmask' is set to true.
nonmask	Logical. Voxels not in the mask WILL have their local statistics computed from all voxels in their neighborhood that ARE in the mask. For instance, this option can be used to compute the average local white matter time series, even at non-WM voxels.
out_file	Character; file path. Output dataset.
overwrite	Logical. overwrite output file if it already exists
quiet	Logical. Stop the highly informative progress reports.
reduce_grid	Character or numeric vector. Compute output on a grid that is reduced by the specified factors. If a single value is passed, output is resampled to the specified isotropic grid. Otherwise, the 3 inputs describe the reduction in the X, Y, and Z directions. This option speeds up computations at the expense of resolution. It should only be used when the nbhd is quite large with respect to the input's resolution, and the resultant stats are expected to be smooth.
reduce_max_vox	Numeric. Like reduce_restore_grid, but automatically set Rx Ry Rz so that the computation grid is at a resolution of nbhd/MAX_VOX voxels.
reduce_restore_grid	Character or numeric vector. Like reduce_grid, but also resample output back to input grid.
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

 ni_afni_mask_tool *AFNI MaskTool*

Description

3dmask_tool - for combining/dilating/eroding/filling masks

Usage

```
ni_afni_mask_tool(
  in_file,
  args = NULL,
  count = NULL,
  datum = NULL,
  dilate_inputs = NULL,
  dilate_results = NULL,
  fill_dirs = NULL,
  fill_holes = NULL,
  frac = NULL,
  inter = NULL,
  out_file = NULL,
  union = NULL,
  verbose = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character or numeric vector. input file or files to 3dmask_tool Required.
args	Character. Additional parameters to the command
count	Logical. Instead of created a binary 0/1 mask dataset, create one with counts of voxel overlap, i.e., each voxel will contain the number of masks that it is set in.
datum	Character; one of: "byte", "short", "float". specify data type for output.
dilate_inputs	Character. Use this option to dilate and/or erode datasets as they are read. ex. '5 -5' to dilate and erode 5 times
dilate_results	Character. dilate and/or erode combined mask at the given levels.
fill_dirs	Character. fill holes only in the given directions. This option is for use with -fill holes. should be a single string that specifies 1-3 of the axes using {x,y,z} labels (i.e. dataset axis order), or using the labels in {R,L,A,P,I,S}.

fill_holes	Logical. This option can be used to fill holes in the resulting mask, i.e. after all other processing has been done.
frac	Numeric. When combining masks (across datasets and sub-bricks), use this option to restrict the result to a certain fraction of the set of volumes
inter	Logical. intersection, this means -frac 1.0
out_file	Character; file path. output image file name
union	Logical. union, this means -frac 0
verbose	Integer. specify verbosity level, for 0 to 3
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_maskave	<i>AFNI Maskave</i>
-----------------	---------------------

Description

Computes average of all voxels in the input dataset

Usage

```
ni_afni_maskave(
  in_file,
  args = NULL,
  mask = NULL,
  out_file = NULL,
  quiet = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. input file to 3dmaskave Required.
args	Character. Additional parameters to the command
mask	Character; file path. matrix to align input file
out_file	Character; file path. output image file name
quiet	Logical. matrix to align input file
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_means	<i>AFNI Means</i>
---------------	-------------------

Description

Takes the voxel-by-voxel mean of all input datasets using 3dMean

Usage

```
ni_afni_means(
  in_file_a,
  args = NULL,
  count = NULL,
  datum = NULL,
  in_file_b = NULL,
  mask_inter = NULL,
  mask_union = NULL,
  non_zero = NULL,
  out_file = NULL,
  scale = NULL,
  sqr = NULL,
  std_dev = NULL,
  summ = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
```

```

    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

in_file_a	Character; file path. input file to 3dMean Required.
args	Character. Additional parameters to the command
count	Logical. compute count of non-zero voxels
datum	Character. Sets the data type of the output dataset
in_file_b	Character; file path. another input file to 3dMean
mask_inter	Logical. create intersection mask
mask_union	Logical. create union mask
non_zero	Logical. use only non-zero values
out_file	Character; file path. output image file name
scale	Character. scaling of output
sqr	Logical. mean square instead of value
std_dev	Logical. calculate std dev
summ	Logical. take sum, (not average)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_merge

AFNI Merge

Description

Merge or edit volumes using AFNI 3dmerge command

Usage

```

ni_afni_merge(
  in_files,
  args = NULL,
  blurfwhm = NULL,
  doall = NULL,
  out_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)

```

Arguments

in_files	Character or numeric vector Required.
args	Character. Additional parameters to the command
blurfwhm	Integer. FWHM blur value (mm)
doall	Logical. apply options to all sub-bricks in dataset
out_file	Character; file path. output image file name
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_net_corr	<i>AFNI NetCorr</i>
------------------	---------------------

Description

Calculate correlation matrix of a set of ROIs (using mean time series of

Usage

```

ni_afni_net_corr(
  in_file,
  in_rois,
  args = NULL,
  fish_z = NULL,
  ignore_LT = NULL,
  mask = NULL,
  nifti = NULL,
  out_file = NULL,
  output_mask_nonnull = NULL,
  part_corr = NULL,
  push_thru_many_zeros = NULL,
  ts_indiv = NULL,
  ts_label = NULL,
  ts_out = NULL,
  ts_wb_Z = NULL,
  ts_wb_corr = NULL,
  ts_wb_strlabel = NULL,
  weight_ts = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)

```

Arguments

in_file	Character; file path. input time series file (4D data set) Required.
in_rois	Character; file path. input set of ROIs, each labelled with distinct integers Required.
args	Character. Additional parameters to the command
fish_z	Logical. switch to also output a matrix of Fisher Z-transform values for the corr coefs (r): $Z = \text{atanh}(r)$, (with Z=4 being output along matrix diagonals where $r=1$, as the r-to-Z conversion is ceilinged at $Z = \text{atanh}(r=0.999329) = 4$, which is still <i>quite</i> a high Pearson-r value)
ignore_LT	Logical. switch to ignore any label table labels in the '-in_rois' file, if there are any labels attached
mask	Character; file path. can include a whole brain mask within which to calculate correlation. Otherwise, data should be masked already
nifti	Logical. output any correlation map files as NIFTI files (default is BRIK/HEAD). Only useful if using '-ts_wb_corr' and/or '-ts_wb_Z'
out_file	Character; file path. output file name part

output_mask_nonnull	Logical. internally, this program checks for where there are nonnull time series, because we don't like those, in general. With this flag, the user can output the determined mask of non-null time series.
part_corr	Logical. output the partial correlation matrix
push_thru_many_zeros	Logical. by default, this program will grind to a halt and refuse to calculate if any ROI contains >10 percent of voxels with null times series (i.e., each point is 0), as of April, 2017. This is because it seems most likely that hidden badness is responsible. However, if the user still wants to carry on the calculation anyways, then this option will allow one to push on through. However, if any ROI <i>only</i> has null time series, then the program will not calculate and the user will really, really, really need to address their masking
ts_indiv	Logical. switch to create a directory for each network that contains the average time series for each ROI in individual files (each file has one line). The directories are labelled PREFIX_000_INDIV/, PREFIX_001_INDIV/, etc. (one per network). Within each directory, the files are labelled ROI_001.netts, ROI_002.netts, etc., with the numbers given by the actual ROI integer labels
ts_label	Logical. additional switch when using '-ts_out'. Using this option will insert the integer ROI label at the start of each line of the *.netts file created. Thus, for a time series of length N, each line will have N+1 numbers, where the first is the integer ROI label and the subsequent N are scientific notation values
ts_out	Logical. switch to output the mean time series of the ROIs that have been used to generate the correlation matrices. Output filenames mirror those of the correlation matrix files, with a '.netts' postfix
ts_wb_Z	Logical. same as above in '-ts_wb_corr', except that the maps have been Fisher transformed to Z-scores the relation: $Z = \text{atanh}(r)$. To avoid infinities in the transform, Pearson values are effectively capped at $ r = 0.999329$ (where $ Z = 4.0$). Files are labelled WB_Z_ROI_001+orig, etc
ts_wb_corr	Logical. switch to create a set of whole brain correlation maps. Performs whole brain correlation for each ROI's average time series; this will automatically create a directory for each network that contains the set of whole brain correlation maps (Pearson 'r's). The directories are labelled as above for '-ts_indiv' Within each directory, the files are labelled WB_CORR_ROI_001+orig, WB_CORR_ROI_002+orig, etc., with the numbers given by the actual ROI integer labels
ts_wb_strlabel	Logical. by default, '-ts_wb_{corr,Z}' output files are named using the int number of a given ROI, such as: WB_Z_ROI_001+orig. With this option, one can replace the int (such as '001') with the string label (such as 'L-thalamus') <i>if</i> one has a labeltable attached to the file
weight_ts	Character; file path. input a 1D file WTS of weights that will be applied multiplicatively to each ROI's average time series. WTS can be a column- or row-file of values, but it must have the same length as the input time series volume. If the initial average time series was $A[n]$ for $n=0, \dots, (N-1)$ time points, then applying a set of weights $W[n]$ of the same length from WTS would produce a new time series: $B[n] = A[n] * W[n]$
.cwd	Working directory override.

.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_notes	<i>AFNI Notes</i>
---------------	-------------------

Description

A program to add, delete, and show notes for AFNI datasets.

Usage

```
ni_afni_notes(
  in_file,
  add = NULL,
  add_history = NULL,
  args = NULL,
  delete = NULL,
  out_file = NULL,
  rep_history = NULL,
  ses = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. input file to 3dNotes Required.
add	Character. note to add
add_history	Character. note to add to history
args	Character. Additional parameters to the command
delete	Integer. delete note number num
out_file	Character; file path. output image file name

rep_history	Character. note with which to replace history
ses	Logical. print to stdout the expanded notes
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_nwarp_adjust *AFNI NwarpAdjust*

Description

This program takes as input a bunch of 3D warps, averages them,

Usage

```
ni_afni_nwarp_adjust(
  warps,
  args = NULL,
  in_files = NULL,
  out_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

warps	Character or numeric vector. List of input 3D warp datasets Required.
args	Character. Additional parameters to the command
in_files	Character or numeric vector. List of input 3D datasets to be warped by the adjusted warp datasets. There must be exactly as many of these datasets as there are input warps.
out_file	Character; file path. Output mean dataset, only needed if in_files are also given. The output dataset will be on the common grid shared by the source datasets.

.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_nwarp_apply *AFNI NwarpApply*

Description

Program to apply a nonlinear 3D warp saved from 3dQwarp

Usage

```
ni_afni_nwarp_apply(
  in_file,
  warp,
  ainterp = NULL,
  args = NULL,
  interp = "wsinc5",
  inv_warp = NULL,
  master = NULL,
  out_file = NULL,
  quiet = NULL,
  short = NULL,
  verb = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character or numeric vector. the name of the dataset to be warped can be multiple datasets Required.
warp	Character. the name of the warp dataset. multiple warps can be concatenated (make sure they exist) Required.

ainterp	Character; one of: "NN", "nearestneighbour", "nearestneighbor", "linear", "trilinear", "cubic", "tricubic", "quintic", "triquintic", "wsinc5". specify a different interpolation method than might be used for the warp
args	Character. Additional parameters to the command
interp	Character; one of: "wsinc5", "NN", "nearestneighbour", "nearestneighbor", "linear", "trilinear", "cubic", "tricubic", "quintic", "triquintic". defines interpolation method to use during warp
inv_warp	Logical. After the warp specified in '-nwarp' is computed, invert it
master	Character; file path. the name of the master dataset, which defines the output grid
out_file	Character; file path. output image file name
quiet	Logical. don't be verbose :(
short	Logical. Write output dataset using 16-bit short integers, rather than the usual 32-bit floats.
verb	Logical. be extra verbose :)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

ni_afni_nwarp_cat	<i>AFNI NwarpCat</i>
-------------------	----------------------

Description

Catenates (composes) 3D warps defined on a grid, OR via a matrix.

Usage

```
ni_afni_nwarp_cat(
  in_files,
  args = NULL,
  expad = NULL,
  interp = "wsinc5",
  inv_warp = NULL,
  out_file = NULL,
  space = NULL,
```

```

    verb = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

in_files	Character or numeric vector. list of tuples of 3D warps and associated functions Required.
args	Character. Additional parameters to the command
expad	Integer. Pad the nonlinear warps by the given number of voxels in all directions. The warp displacements are extended by linear extrapolation from the faces of the input grid..
interp	Character; one of: "wsinc5", "linear", "quintic". specify a different interpolation method than might be used for the warp
inv_warp	Logical. invert the final warp before output
out_file	Character; file path. output image file name
space	Character. string to attach to the output dataset as its atlas space marker.
verb	Logical. be verbose
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_one_d_tool_py *AFNI OneDToolPy*

Description

This program is meant to read/manipulate/write/diagnose 1D datasets.

Usage

```

ni_afni_one_d_tool_py(
  in_file,
  args = NULL,
  censor_motion = NULL,
  censor_prev_TR = NULL,
  demean = NULL,
  derivative = NULL,
  out_file = NULL,
  set_nruns = NULL,
  show_censor_count = NULL,
  show_cormat_warnings = NULL,
  show_indices_interest = NULL,
  show_trs_run = NULL,
  show_trs_uncensored = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)

```

Arguments

<code>in_file</code>	Character; file path. input file to OneDTool Required.
<code>args</code>	Character. Additional parameters to the command
<code>censor_motion</code>	Character or numeric vector. Tuple of motion limit and outfile prefix. need to also set <code>set_nruns -r set_run_lengths</code>
<code>censor_prev_TR</code>	Logical. for each censored TR, also censor previous
<code>demean</code>	Logical. demean each run (new mean of each run = 0.0)
<code>derivative</code>	Logical. take the temporal derivative of each vector (done as first backward difference)
<code>out_file</code>	Character; file path. write the current 1D data to FILE
<code>set_nruns</code>	Integer. treat the input data as if it has nruns
<code>show_censor_count</code>	Logical. display the total number of censored TRs Note : if input is a valid xmat.1D dataset, then the count will come from the header. Otherwise the input is assumed to be a binary censorfile, and zeros are simply counted.
<code>show_cormat_warnings</code>	Character; file path. Write cormat warnings to a file
<code>show_indices_interest</code>	Logical. display column indices for regs of interest
<code>show_trs_run</code>	Integer. restrict <code>-show_trs_[un]censored</code> to the given 1-based run

show_trs_uncensored	Character; one of: "comma", "space", "encoded", "verbose". display a list of TRs which were not censored in the specified style
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_outlier_count *AFNI OutlierCount*

Description

Calculates number of 'outliers' at each time point of a

Usage

```
ni_afni_outlier_count(
  in_file,
  args = NULL,
  autoclip = FALSE,
  automask = FALSE,
  fraction = FALSE,
  interval = FALSE,
  legendre = FALSE,
  mask = NULL,
  outliers_file = NULL,
  polort = NULL,
  qthr = 0.001,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. input dataset Required.
args	Character. Additional parameters to the command
autoclip	Logical. clip off small voxels
automask	Logical. clip off small voxels
fraction	Logical. write out the fraction of masked voxels which are outliers at each timepoint
interval	Logical. write out the median + 3.5 MAD of outlier count with each timepoint
legendre	Logical. use Legendre polynomials
mask	Character; file path. only count voxels within the given mask
outliers_file	Character; file path. output image file name
polort	Integer. detrend each voxel timeseries with polynomials
qthr	Character. indicate a value for q to compute alpha
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_quality_index *AFNI QualityIndex*

Description

Computes a quality index for each sub-brick in a 3D+time dataset.

Usage

```
ni_afni_quality_index(
  in_file,
  args = NULL,
  autoclip = FALSE,
  automask = FALSE,
  clip = NULL,
  interval = FALSE,
  mask = NULL,
  out_file = NULL,
```

```

quadrant = FALSE,
spearman = FALSE,
.cwd = NULL,
.env = NULL,
.engine = NULL,
.profile = NULL,
dry_run = FALSE,
echo = interactive()
)

```

Arguments

in_file	Character; file path. input dataset Required.
args	Character. Additional parameters to the command
autoclip	Logical. clip off small voxels
automask	Logical. clip off small voxels
clip	Numeric. clip off values below
interval	Logical. write out the median + 3.5 MAD of outlier count with each timepoint
mask	Character; file path. compute correlation only across masked voxels
out_file	Character; file path. capture standard output
quadrant	Logical. Similar to -spearman, but using 1 minus the quadrant correlation coefficient as the quality index.
spearman	Logical. Quality index is 1 minus the Spearman (rank) correlation coefficient of each sub-brick with the median sub-brick. (default).
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_qwarp	<i>AFNI Qwarp</i>
---------------	-------------------

Description

Allineate your images prior to passing them to this workflow.

Usage

```
ni_afni_qwarp(  
  base_file,  
  in_file,  
  Qfinal = NULL,  
  Qonly = NULL,  
  allineate = NULL,  
  allineate_opts = NULL,  
  allsave = NULL,  
  args = NULL,  
  ballopt = NULL,  
  bandpass = NULL,  
  baxopt = NULL,  
  blur = NULL,  
  duplo = NULL,  
  emask = NULL,  
  expad = NULL,  
  gridlist = NULL,  
  hel = NULL,  
  inilev = NULL,  
  iniwarp = NULL,  
  iwarp = NULL,  
  lpa = NULL,  
  lpc = NULL,  
  maxlev = NULL,  
  mi = NULL,  
  minpatch = NULL,  
  nmi = NULL,  
  noXdis = NULL,  
  noYdis = NULL,  
  noZdis = NULL,  
  noneg = NULL,  
  nopad = NULL,  
  nopadWARP = NULL,  
  nopenalty = NULL,  
  nowarp = NULL,  
  noweight = NULL,  
  out_file = NULL,  
  out_weight_file = NULL,
```

```

overwrite = NULL,
pblur = NULL,
pear = NULL,
penfac = NULL,
plusminus = NULL,
quiet = NULL,
resample = NULL,
verb = NULL,
wball = NULL,
weight = NULL,
wmask = NULL,
workhard = NULL,
.cwd = NULL,
.env = NULL,
.engine = NULL,
.profile = NULL,
dry_run = FALSE,
echo = interactive()
)

```

Arguments

base_file	Character; file path. Base image (opposite phase encoding direction than source image). Required.
in_file	Character; file path. Source image (opposite phase encoding direction than base image). Required.
Qfinal	Logical. At the finest patch size (the final level), use Hermite quintic polynomials for the warp instead of cubic polynomials. * In a 3D 'patch', there are $2 \times 2 \times 2 \times 3 = 24$ cubic polynomial basis function parameters over which to optimize (2 polynomials dependent on each of the x,y,z directions, and 3 different directions of displacement). * There are $3 \times 3 \times 3 \times 3 = 81$ quintic polynomial parameters per patch. * With -Qfinal, the final level will have more detail in the allowed warps, at the cost of yet more CPU time. * However, no patch below $7 \times 7 \times 7$ in size will be done with quintic polynomials. * This option is also not usually needed, and is experimental.
Qonly	Logical. Use Hermite quintic polynomials at all levels. * Very slow (about 4 times longer). Also experimental. * Will produce a (discrete representation of a) C2 warp.
allineate	Logical. This option will make 3dQwarp run 3dAllineate first, to align the source dataset to the base with an affine transformation. It will then use that alignment as a starting point for the nonlinear warping.
allineate_opts	Character. add extra options to the 3dAllineate command to be run by 3dQwarp.
allsave	Logical. This option lets you save the output warps from each level" of the refinement process. Mostly used for experimenting." Will only save all the outputs if the program terminates" normally – if it crashes, or freezes, then all these" warps are lost.
args	Character. Additional parameters to the command

ballopt	Logical. Normally, the incremental warp parameters are optimized inside a rectangular 'box' (24 dimensional for cubic patches, 81 for quantic patches), whose limits define the amount of distortion allowed at each step. Using '-ballopt' switches these limits to be applied to a 'ball' (interior of a hypersphere), which can allow for larger incremental displacements. Use this option if you think things need to be able to move farther.
bandpass	Character or numeric vector
baxopt	Logical. Use the 'box' optimization limits instead of the 'ball' [this is the default at present]. Note that if '-workhard' is used, then ball and box optimization are alternated in the different iterations at each level, so these two options have no effect in that case.
blur	Character or numeric vector. Gaussian blur the input images by 'bb' (FWHM) voxels before doing the alignment (the output dataset will not be blurred). The default is 2.345 (for no good reason). * Optionally, you can provide 2 values for 'bb', and then the first one is applied to the base volume, the second to the source volume. e.g., '-blur 0 3' to skip blurring the base image (if the base is a blurry template, for example). * A negative blur radius means to use 3D median filtering, rather than Gaussian blurring. This type of filtering will better preserve edges, which can be important in alignment. * If the base is a template volume that is already blurry, you probably don't want to blur it again, but blurring the source volume a little is probably a good idea, to help the program avoid trying to match tiny features. * Note that -duplo will blur the volumes some extra amount for the initial small-scale warping, to make that phase of the program converge more rapidly.
duplo	Logical. Start off with 1/2 scale versions of the volumes, "for getting a speedy coarse first alignment." * Then scales back up to register the full volumes." The goal is greater speed, and it seems to help this "positively piggy program to be more expeditious." * However, accuracy is somewhat lower with '-duplo'," for reasons that currently elude Zhark; for this reason, "the Emperor does not usually use '-duplo'.
emask	Character; file path. Here, 'ee' is a dataset to specify a mask of voxels to EXCLUDE from the analysis – all voxels in 'ee' that are NONZERO will not be used in the alignment. The base image always automasked – the emask is extra, to indicate voxels you definitely DON'T want included in the matching process, even if they are inside the brain.
expad	Integer. This option instructs the program to pad the warp by an extra 'EE' voxels (and then 3dQwarp starts optimizing it). This option is seldom needed, but can be useful if you might later concatenate the nonlinear warp – via 3dNwarpCat – with an affine transformation that contains a large shift. Under that circumstance, the nonlinear warp might be shifted partially outside its original grid, so expanding that grid can avoid this problem. Note that this option perforce turns off '-nopadWARP'.
gridlist	Character; file path. This option provides an alternate way to specify the patch grid sizes used in the warp optimization process. 'gl' is a 1D file with a list of patches to use – in most cases, you will want to use it in the following form: -gridlist '1D: 0 151 101 75 51' * Here, a 0 patch size means the global domain. Patch sizes otherwise should be odd integers >= 5. * If you use the

	'0' patch size again after the first position, you will actually get an iteration at the size of the default patch level 1, where the patch sizes are 75% of the volume dimension. There is no way to force the program to literally repeat the sui generis step of lev=0.
hel	Logical. Hellinger distance: a matching function for the adventurousThis option has NOT be extensively tested for usefulnessand should be considered experimental at this infundibulum.
inilev	Integer. The initial refinement 'level' at which to start. * Usually used with -iniwarp; CANNOT be used with -duplo. * The combination of -inilev and -iniwarp lets you take the results of a previous 3dQwarp run and refine them further: Note that the source dataset in the second run is the SAME as in the first run. If you don't see why this is necessary, then you probably need to seek help from an AFNI guru.
iniwarp	Character or numeric vector. A dataset with an initial nonlinear warp to use. * If this option is not used, the initial warp is the identity. * You can specify a catenation of warps (in quotes) here, as in program 3dNwarpApply. * As a special case, if you just input an affine matrix in a .1D file, that will work also – it is treated as giving the initial warp via the string "IDENT(base_dataset matrix_file.aff12.1D)". * You CANNOT use this option with -duplo !! * -iniwarp is usually used with -inilev to re-start 3dQwarp from a previous stopping point.
iwarp	Logical. Do compute and save the _WARPINV file.
lpa	Logical. Local Pearson maximization. This option has not be extensively tested
lpc	Logical. Local Pearson minimization (i.e., EPI-T1 registration)This option has not be extensively testedIf you use '-lpc', then '-maxlev 0' is automatically set.If you want to go to more refined levels, you can set '-maxlev'This should be set up to have lpc as the second to last argumentand maxlev as the second to last argument, as needed by AFNIUsing maxlev > 1 is not recommended for EPI-T1 alignment.
maxlev	Integer. The initial refinement 'level' at which to start. * Usually used with -iniwarp; CANNOT be used with -duplo. * The combination of -inilev and -iniwarp lets you take the results of a previous 3dQwarp run and refine them further: Note that the source dataset in the second run is the SAME as in the first run. If you don't see why this is necessary, then you probably need to seek help from an AFNI guru.
mi	Logical. Mutual Information: a matching function for the adventurousThis option has NOT be extensively tested for usefulnessand should be considered experimental at this infundibulum.
minpatch	Integer. The value of mm should be an odd integer. * The default value of mm is 25. * For more accurate results than mm=25, try 19 or 13. * The smallest allowed patch size is 5. * You may want stop at a larger patch size (say 7 or 9) and use the -Qfinal option to run that final level with quintic warps, which might run faster and provide the same degree of warp detail. * Trying to make two different brain volumes match in fine detail is usually a waste of time, especially in humans. There is too much variability in anatomy to match gyrus to gyrus accurately. For this reason, the default minimum patch size is 25 voxels. Using a smaller '-minpatch' might try to force the warp to match features that do not

	match, and the result can be useless image distortions – another reason to LOOK AT THE RESULTS.
nmi	Logical. Normalized Mutual Information: a matching function for the adventurous. This option has NOT been extensively tested for usefulness and should be considered experimental at this infundibulum.
noXdis	Logical. Warp will not displace in x direction
noYdis	Logical. Warp will not displace in y direction
noZdis	Logical. Warp will not displace in z direction
noneg	Logical. Replace negative values in either input volume with 0. * If there ARE negative input values, and you do NOT use -noneg, then strict Pearson correlation will be used, since the 'clipped' method only is implemented for non-negative volumes. * '-noneg' is not the default, since there might be situations where you want to align datasets with positive and negative values mixed. * But, in many cases, the negative values in a dataset are just the result of interpolation artifacts (or other peculiarities), and so they should be ignored. That is what '-noneg' is for.
nopad	Logical. Do NOT use zero-padding on the 3D base and source images. [Default == zero-pad, if needed] * The underlying model for deformations goes to zero at the edge of the volume being warped. However, if there is significant data near an edge of the volume, then it won't get displaced much, and so the results might not be good. * Zero padding is designed as a way to work around this potential problem. You should NOT need the '-nopad' option for any reason that Zhark can think of, but it is here to be symmetrical with 3dAllineate. * Note that the output (warped from source) dataset will be on the base dataset grid whether or not zero-padding is allowed. However, unless you use the following option, allowing zero-padding (i.e., the default operation) will make the output WARP dataset(s) be on a larger grid (also see '-expad' below).
nopadWARP	Logical. If for some reason you require the warp volume to match the base volume, then use this option to have the output WARP dataset(s) truncated.
nopenalty	Logical. Replace negative values in either input volume with 0. * If there ARE negative input values, and you do NOT use -noneg, then strict Pearson correlation will be used, since the 'clipped' method only is implemented for non-negative volumes. * '-noneg' is not the default, since there might be situations where you want to align datasets with positive and negative values mixed. * But, in many cases, the negative values in a dataset are just the result of interpolation artifacts (or other peculiarities), and so they should be ignored. That is what '-noneg' is for.
nowarp	Logical. Do not save the _WARP file.
noweight	Logical. If you want a binary weight (the old default), use this option. That is, each voxel in the base volume automask will be weighted the same in the computation of the cost functional.
out_file	Character; file path. Sets the prefix/suffix for the output datasets. * The source dataset is warped to match the base and gets prefix 'ppp'. (Except if '-plusminus' is used * The final interpolation to this output dataset is done using the 'wsinc5' method. See the output of 3dAllineate -HELP (in the "Modifying '-final wsinc5'" section) for the lengthy technical details. * The 3D warp used

is saved in a dataset with prefix 'ppp_WARP' – this dataset can be used with 3dNwarpApply and 3dNwarpCat, for example. * To be clear, this is the warp from source dataset coordinates to base dataset coordinates, where the values at each base grid point are the xyz displacements needed to move that grid point's xyz values to the corresponding xyz values in the source dataset: $\text{base}(x,y,z) + \text{WARP}(x,y,z)$ matches $\text{source}(x,y,z)$. Another way to think of this warp is that it 'pulls' values back from source space to base space. * 3dNwarpApply would use 'ppp_WARP' to transform datasets aligned with the source dataset to be aligned with the base dataset. **If you do NOT want this warp saved, use the option '-nowarp'**. (However, this warp is usually the most valuable possible output!) * If you want to calculate and save the inverse 3D warp, use the option '-iwarp'. This inverse warp will then be saved in a dataset with prefix 'ppp_WARPINV'. * This inverse warp could be used to transform data from base space to source space, if you need to do such an operation. * You can easily compute the inverse later, say by a command like 3dNwarpCat -prefix Z_WARPINV 'INV(Z_WARP+tlrc)' or the inverse can be computed as needed in 3dNwarpApply, like 3dNwarpApply -nwarp 'INV(Z_WARP+tlrc)' -source Dataset.nii ...

out_weight_file	Character; file path. Write the weight volume to disk as a dataset
overwrite	Logical. Overwrite outputs
pblur	Character or numeric vector. Use progressive blurring; that is, for larger patch sizes, the amount of blurring is larger. The general idea is to avoid trying to match finer details when the patch size and incremental warps are coarse. When '-blur' is used as well, it sets a minimum amount of blurring that will be used. [06 Aug 2014 – '-pblur' may become the default someday]. * You can optionally give the fraction of the patch size that is used for the progressive blur by providing a value between 0 and 0.25 after '-pblur'. If you provide TWO values, the the first fraction is used for progressively blurring the base image and the second for the source image. The default parameters when just '-pblur' is given is the same as giving the options as '-pblur 0.09 0.09'. * '-pblur' is useful when trying to match 2 volumes with high amounts of detail; e.g, warping one subject's brain image to match another's, or trying to warp to match a detailed template. * Note that using negative values with '-blur' means that the progressive blurring will be done with median filters, rather than Gaussian linear blurring. Note: The combination of the -allineate and -pblur options will make the results of using 3dQwarp to align to a template somewhat less sensitive to initial head position and scaling.
pear	Logical. Use strict Pearson correlation for matching. Not usually recommended, since the 'clipped Pearson' method used by default will reduce the impact of outlier values.
penfac	Numeric. Use this value to weight the penalty. The default value is 1. Larger values mean the penalty counts more, reducing grid distortions, insha'Allah; '-nopenalty' is the same as '-penfac 0'. In 23 Sep 2013 Zhark increased the default value of the penalty by a factor of 5, and also made it get progressively larger with each level of refinement. Thus, warping results will vary from earlier instances of 3dQwarp. * The progressive increase in the penalty at higher levels

means that the 'cost function' can actually look like the alignment is getting worse when the levels change. * IF you wish to turn off this progression, for whatever reason (e.g., to keep compatibility with older results), use the option '-penold'. To be completely compatible with the older 3dQwarp, you'll also have to use '-penfac 0.2'.

- plusminus Logical. Normally, the warp displacements $dis(x)$ are defined to match $base(x)$ to $source(x+dis(x))$. With this option, the match is between $base(x-dis(x))$ and $source(x+dis(x))$ – the two images 'meet in the middle'. * One goal is to mimic the warping done to MRI EPI data by field inhomogeneities, when registering between a 'blip up' and a 'blip down' down volume, which will have opposite distortions. * Define $Wp(x) = x+dis(x)$ and $Wm(x) = x-dis(x)$. Then since $base(Wm(x))$ matches $source(Wp(x))$, by substituting $INV(Wm(x))$ wherever we see x , we have $base(x)$ matches $source(Wp(INV(Wm(x))))$; that is, the warp $V(x)$ that one would get from the 'usual' way of running 3dQwarp is $V(x) = Wp(INV(Wm(x)))$. * Conversely, we can calculate $Wp(x)$ in terms of $V(x)$ as follows: If $V(x) = x + dv(x)$, define $Vh(x) = x + dv(x)/2$; then $Wp(x) = V(INV(Vh(x)))$. * With the above formulas, it is possible to compute $Wp(x)$ from $V(x)$ and vice-versa, using program 3dNwarpCalc. The requisite commands are left as an exercise for the aspiring AFNI Jedi Master. * You can use the semi-secret '-pmBASE' option to get the $V(x)$ warp and the source dataset warped to base space, in addition to the $Wp(x)$ '_PLUS' and $Wm(x)$ '_MINUS' warps. * Alas: -plusminus does not work with -duplo or -allineate :-(* However, you can use -iniwarp with -plusminus :-) * The outputs have _PLUS (from the source dataset) and _MINUS (from the base dataset) in their filenames, in addition to the prefix. The -iwarp option, if present, will be ignored.
- quiet Logical. Cut out most of the fun fun fun progress messages :-)
- resample Logical. This option simply resamples the source dataset to match the base dataset grid. You can use this if the two datasets overlap well (as seen in the AFNI GUI), but are not on the same 3D grid. * If they don't overlap well, allineate them first * The reampling here is done with the 'wsinc5' method, which has very little blurring artifact. * If the base and source datasets ARE on the same 3D grid, then the -resample option will be ignored. * You CAN use -resample with these 3dQwarp options: -plusminus -inilev -iniwarp -duplo
- verb Logical. more detailed description of the process
- wball Character or numeric vector. "-wball x y z r f Enhance automatic weight from '-useweight' by a factor of $1+f \cdot \text{Gaussian}(FWHM=r)$ centered in the base image at DICOM coordinates (x,y,z) and with radius 'r'. The goal of this option is to try and make the alignment better in a specific part of the brain. Example: -wball 0 14 6 30 40 to emphasize the thalamic area (in MNI/Talairach space). * The 'r' parameter must be positive! * The 'f' parameter must be between 1 and 100 (inclusive). * '-wball' does nothing if you input your own weight with the '-weight' option. * '-wball' does change the binary weight created by the '-noweight' option. * You can only use '-wball' once in a run of 3dQwarp. **The effect of '-wball' is not dramatic.** The example above makes the average brain image across a collection of subjects a little sharper in the thalamic area, which might have some small value. If you care enough about alignment to use '-wball', then you should examine the results from 3dQwarp for each subject, to see if the alignments are good enough for your purposes.

weight	Character; file path. Instead of computing the weight from the base dataset, directly input the weight volume from dataset 'www'. Useful if you know what over parts of the base image you want to emphasize or de-emphasize the matching functional.
wmask	Character or numeric vector. Similar to '-wball', but here, you provide a dataset 'ws' that indicates where to increase the weight. * The 'ws' dataset must be on the same 3D grid as the base dataset. * 'ws' is treated as a mask – it only matters where it is nonzero – otherwise, the values inside are not used. * After 'ws' comes the factor 'f' by which to increase the automatically computed weight. Where 'ws' is nonzero, the weighting will be multiplied by (1+f). * As with '-wball', the factor 'f' should be between 1 and 100.
workhard	Logical. Iterate more times, which can help when the volumes are hard to align at all, or when you hope to get a more precise alignment. * Slows the program down (possibly a lot), of course. * When you combine '-workhard' with '-duplo', only the full size volumes get the extra iterations. * For finer control over which refinement levels work hard, you can use this option in the form (for example) -workhard: 4:7 which implies the extra iterations will be done at levels 4, 5, 6, and 7, but not otherwise. * You can also use '-superhard' to iterate even more, but this extra option will REALLY slow things down. * Under most circumstances, you should not need to use either -workhard or -superhard. * The fastest way to register to a template image is via the -duplo option, and without the -workhard or -superhard options. * If you use this option in the form '-Workhard' (first letter in upper case), then the second iteration at each level is done with quintic polynomial warps.
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_qwarp_plus_minus

AFNI QwarpPlusMinus

Description

A version of 3dQwarp for performing field susceptibility correction

Usage

```
ni_afni_qwarp_plus_minus(  
  base_file,  
  in_file,  
  Qfinal = NULL,  
  Qonly = NULL,  
  allineate = NULL,  
  allineate_opts = NULL,  
  allsave = NULL,  
  args = NULL,  
  ballopt = NULL,  
  bandpass = NULL,  
  baxopt = NULL,  
  blur = NULL,  
  duplo = NULL,  
  emask = NULL,  
  expad = NULL,  
  gridlist = NULL,  
  hel = NULL,  
  inilev = NULL,  
  iniwarp = NULL,  
  iwarp = NULL,  
  lpa = NULL,  
  lpc = NULL,  
  maxlev = NULL,  
  mi = NULL,  
  minpatch = NULL,  
  nmi = NULL,  
  noXdis = NULL,  
  noYdis = NULL,  
  noZdis = NULL,  
  noneg = NULL,  
  nopad = NULL,  
  nopadWARP = NULL,  
  nopenalty = NULL,  
  nowarp = NULL,  
  noweight = NULL,  
  out_file = "Qwarp.nii.gz",  
  out_weight_file = NULL,  
  overwrite = NULL,  
  pblur = NULL,  
  pear = NULL,  
  penfac = NULL,  
  plusminus = TRUE,  
  quiet = NULL,  
  resample = NULL,  
  source_file = NULL,  
  verb = NULL,
```

```

wball = NULL,
weight = NULL,
wmask = NULL,
workhard = NULL,
.cwd = NULL,
.env = NULL,
.engine = NULL,
.profile = NULL,
dry_run = FALSE,
echo = interactive()
)

```

Arguments

base_file	Character; file path. Base image (opposite phase encoding direction than source image). Required.
in_file	Character; file path. Source image (opposite phase encoding direction than base image). Required.
Qfinal	Logical. At the finest patch size (the final level), use Hermite quintic polynomials for the warp instead of cubic polynomials. * In a 3D 'patch', there are $2 \times 2 \times 2 \times 3 = 24$ cubic polynomial basis function parameters over which to optimize (2 polynomials dependent on each of the x,y,z directions, and 3 different directions of displacement). * There are $3 \times 3 \times 3 \times 3 = 81$ quintic polynomial parameters per patch. * With -Qfinal, the final level will have more detail in the allowed warps, at the cost of yet more CPU time. * However, no patch below $7 \times 7 \times 7$ in size will be done with quintic polynomials. * This option is also not usually needed, and is experimental.
Qonly	Logical. Use Hermite quintic polynomials at all levels. * Very slow (about 4 times longer). Also experimental. * Will produce a (discrete representation of a) C2 warp.
allineate	Logical. This option will make 3dQwarp run 3dAllineate first, to align the source dataset to the base with an affine transformation. It will then use that alignment as a starting point for the nonlinear warping.
allineate_opts	Character. add extra options to the 3dAllineate command to be run by 3dQwarp.
allsave	Logical. This option lets you save the output warps from each level" of the refinement process. Mostly used for experimenting." Will only save all the outputs if the program terminates" normally – if it crashes, or freezes, then all these" warps are lost.
args	Character. Additional parameters to the command
ballopt	Logical. Normally, the incremental warp parameters are optimized inside a rectangular 'box' (24 dimensional for cubic patches, 81 for quintic patches), whose limits define the amount of distortion allowed at each step. Using '-ballopt' switches these limits to be applied to a 'ball' (interior of a hypersphere), which can allow for larger incremental displacements. Use this option if you think things need to be able to move farther.
bandpass	Character or numeric vector

baxopt	Logical. Use the 'box' optimization limits instead of the 'ball' [this is the default at present]. Note that if '-workhard' is used, then ball and box optimization are alternated in the different iterations at each level, so these two options have no effect in that case.
blur	Character or numeric vector. Gaussian blur the input images by 'bb' (FWHM) voxels before doing the alignment (the output dataset will not be blurred). The default is 2.345 (for no good reason). * Optionally, you can provide 2 values for 'bb', and then the first one is applied to the base volume, the second to the source volume. e.g., '-blur 0 3' to skip blurring the base image (if the base is a blurry template, for example). * A negative blur radius means to use 3D median filtering, rather than Gaussian blurring. This type of filtering will better preserve edges, which can be important in alignment. * If the base is a template volume that is already blurry, you probably don't want to blur it again, but blurring the source volume a little is probably a good idea, to help the program avoid trying to match tiny features. * Note that -duplo will blur the volumes some extra amount for the initial small-scale warping, to make that phase of the program converge more rapidly.
duplo	Logical. Start off with 1/2 scale versions of the volumes, "for getting a speedy coarse first alignment." * Then scales back up to register the full volumes." The goal is greater speed, and it seems to help this "positively piggish program to be more expeditious." * However, accuracy is somewhat lower with '-duplo'," for reasons that currently elude Zhark; for this reason, "the Emperor does not usually use '-duplo'.
emask	Character; file path. Here, 'ee' is a dataset to specify a mask of voxels to EXCLUDE from the analysis – all voxels in 'ee' that are NONZERO will not be used in the alignment. The base image always automasked – the emask is extra, to indicate voxels you definitely DON'T want included in the matching process, even if they are inside the brain.
expad	Integer. This option instructs the program to pad the warp by an extra 'EE' voxels (and then 3dQwarp starts optimizing it). This option is seldom needed, but can be useful if you might later concatenate the nonlinear warp – via 3dNwarpCat – with an affine transformation that contains a large shift. Under that circumstance, the nonlinear warp might be shifted partially outside its original grid, so expanding that grid can avoid this problem. Note that this option forces turns off '-nopadWARP'.
gridlist	Character; file path. This option provides an alternate way to specify the patch grid sizes used in the warp optimization process. 'gl' is a 1D file with a list of patches to use – in most cases, you will want to use it in the following form: -gridlist '1D: 0 151 101 75 51' * Here, a 0 patch size means the global domain. Patch sizes otherwise should be odd integers >= 5. * If you use the '0' patch size again after the first position, you will actually get an iteration at the size of the default patch level 1, where the patch sizes are 75% of the volume dimension. There is no way to force the program to literally repeat the sui generis step of lev=0.
hel	Logical. Hellinger distance: a matching function for the adventurous. This option has NOT been extensively tested for usefulness and should be considered experimental at this infundibulum.

inilev	Integer. The initial refinement 'level' at which to start. * Usually used with -iniwarp; CANNOT be used with -duplo. * The combination of -inilev and -iniwarp lets you take the results of a previous 3dQwarp run and refine them further: Note that the source dataset in the second run is the SAME as in the first run. If you don't see why this is necessary, then you probably need to seek help from an AFNI guru.
iniwarp	Character or numeric vector. A dataset with an initial nonlinear warp to use. * If this option is not used, the initial warp is the identity. * You can specify a catenation of warps (in quotes) here, as in program 3dNwarpApply. * As a special case, if you just input an affine matrix in a .1D file, that will work also – it is treated as giving the initial warp via the string "IDENT(base_dataset)matrix_file.aff12.1D". * You CANNOT use this option with -duplo !! * -iniwarp is usually used with -inilev to re-start 3dQwarp from a previous stopping point.
iwarp	Logical. Do compute and save the _WARPINV file.
lpa	Logical. Local Pearson maximization. This option has not be extensively tested
lpc	Logical. Local Pearson minimization (i.e., EPI-T1 registration)This option has not be extensively testedIf you use '-lpc', then '-maxlev 0' is automatically set.If you want to go to more refined levels, you can set '-maxlev'This should be set up to have lpc as the second to last argumentand maxlev as the second to last argument, as needed by AFNIUsing maxlev > 1 is not recommended for EPI-T1 alignment.
maxlev	Integer. The initial refinement 'level' at which to start. * Usually used with -iniwarp; CANNOT be used with -duplo. * The combination of -inilev and -iniwarp lets you take the results of a previous 3dQwarp run and refine them further: Note that the source dataset in the second run is the SAME as in the first run. If you don't see why this is necessary, then you probably need to seek help from an AFNI guru.
mi	Logical. Mutual Information: a matching function for the adventurousThis option has NOT be extensively tested for usefulnessand should be considered experimental at this infundibulum.
minpatch	Integer. The value of mm should be an odd integer. * The default value of mm is 25. * For more accurate results than mm=25, try 19 or 13. * The smallest allowed patch size is 5. * You may want stop at a larger patch size (say 7 or 9) and use the -Qfinal option to run that final level with quintic warps, which might run faster and provide the same degree of warp detail. * Trying to make two different brain volumes match in fine detail is usually a waste of time, especially in humans. There is too much variability in anatomy to match gyrus to gyrus accurately. For this reason, the default minimum patch size is 25 voxels. Using a smaller '-minpatch' might try to force the warp to match features that do not match, and the result can be useless image distortions – another reason to LOOK AT THE RESULTS.
nmi	Logical. Normalized Mutual Information: a matching function for the adventurousThis option has NOT been extensively tested for usefulnessand should be considered experimental at this infundibulum.
noXdis	Logical. Warp will not displace in x direction
noYdis	Logical. Warp will not displace in y direction

noZdis	Logical. Warp will not displace in z direction
noneg	Logical. Replace negative values in either input volume with 0. * If there ARE negative input values, and you do NOT use -noneg, then strict Pearson correlation will be used, since the 'clipped' method only is implemented for non-negative volumes. * '-noneg' is not the default, since there might be situations where you want to align datasets with positive and negative values mixed. * But, in many cases, the negative values in a dataset are just the result of interpolation artifacts (or other peculiarities), and so they should be ignored. That is what '-noneg' is for.
nopad	Logical. Do NOT use zero-padding on the 3D base and source images. [Default == zero-pad, if needed] * The underlying model for deformations goes to zero at the edge of the volume being warped. However, if there is significant data near an edge of the volume, then it won't get displaced much, and so the results might not be good. * Zero padding is designed as a way to work around this potential problem. You should NOT need the '-nopad' option for any reason that Zhark can think of, but it is here to be symmetrical with 3dAllineate. * Note that the output (warped from source) dataset will be on the base dataset grid whether or not zero-padding is allowed. However, unless you use the following option, allowing zero-padding (i.e., the default operation) will make the output WARP dataset(s) be on a larger grid (also see '-expad' below).
nopadWARP	Logical. If for some reason you require the warp volume to match the base volume, then use this option to have the output WARP dataset(s) truncated.
nopenalty	Logical. Replace negative values in either input volume with 0. * If there ARE negative input values, and you do NOT use -noneg, then strict Pearson correlation will be used, since the 'clipped' method only is implemented for non-negative volumes. * '-noneg' is not the default, since there might be situations where you want to align datasets with positive and negative values mixed. * But, in many cases, the negative values in a dataset are just the result of interpolation artifacts (or other peculiarities), and so they should be ignored. That is what '-noneg' is for.
nowarp	Logical. Do not save the _WARP file.
noweight	Logical. If you want a binary weight (the old default), use this option. That is, each voxel in the base volume automask will be weighted the same in the computation of the cost functional.
out_file	Character; file path. Output file
out_weight_file	Character; file path. Write the weight volume to disk as a dataset
overwrite	Logical. Overwrite outputs
pblur	Character or numeric vector. Use progressive blurring; that is, for larger patch sizes, the amount of blurring is larger. The general idea is to avoid trying to match finer details when the patch size and incremental warps are coarse. When '-blur' is used as well, it sets a minimum amount of blurring that will be used. [06 Aug 2014 – '-pblur' may become the default someday]. * You can optionally give the fraction of the patch size that is used for the progressive blur by providing a value between 0 and 0.25 after '-pblur'. If you provide TWO values, the the first fraction is used for progressively blurring the base image and

the second for the source image. The default parameters when just '-pblur' is given is the same as giving the options as '-pblur 0.09 0.09'. * '-pblur' is useful when trying to match 2 volumes with high amounts of detail; e.g, warping one subject's brain image to match another's, or trying to warp to match a detailed template. * Note that using negative values with '-blur' means that the progressive blurring will be done with median filters, rather than Gaussian linear blurring. Note: The combination of the -allineate and -pblur options will make the results of using 3dQwarp to align to a template somewhat less sensitive to initial head position and scaling.

pear	Logical. Use strict Pearson correlation for matching. Not usually recommended, since the 'clipped Pearson' method used by default will reduce the impact of outlier values.
penfac	Numeric. Use this value to weight the penalty. The default value is 1. Larger values mean the penalty counts more, reducing grid distortions, insha'Allah; '-nopenalty' is the same as '-penfac 0'. In 23 Sep 2013 Zhark increased the default value of the penalty by a factor of 5, and also made it get progressively larger with each level of refinement. Thus, warping results will vary from earlier instances of 3dQwarp. * The progressive increase in the penalty at higher levels means that the 'cost function' can actually look like the alignment is getting worse when the levels change. * IF you wish to turn off this progression, for whatever reason (e.g., to keep compatibility with older results), use the option '-penold'. To be completely compatible with the older 3dQwarp, you'll also have to use '-penfac 0.2'.
plusminus	Logical. Normally, the warp displacements $dis(x)$ are defined to match $base(x)$ to $source(x+dis(x))$. With this option, the match is between $base(x-dis(x))$ and $source(x+dis(x))$ – the two images 'meet in the middle'. For more info, view Qwarp' interface
quiet	Logical. Cut out most of the fun fun fun progress messages :-)
resample	Logical. This option simply resamples the source dataset to match the base dataset grid. You can use this if the two datasets overlap well (as seen in the AFNI GUI), but are not on the same 3D grid. * If they don't overlap well, allineate them first * The resampling here is done with the 'wsinc5' method, which has very little blurring artifact. * If the base and source datasets ARE on the same 3D grid, then the -resample option will be ignored. * You CAN use -resample with these 3dQwarp options: -plusminus -inilev -iniwarp -duplo
source_file	Character; file path. Source image (opposite phase encoding direction than base image)
verb	Logical. more detailed description of the process
wball	Character or numeric vector. "-wball x y z r f Enhance automatic weight from '-useweight' by a factor of $1+f \cdot Gaussian(FWHM=r)$ centered in the base image at DICOM coordinates (x,y,z) and with radius 'r'. The goal of this option is to try and make the alignment better in a specific part of the brain. Example: -wball 0 14 6 30 40 to emphasize the thalamic area (in MNI/Talairach space). * The 'r' parameter must be positive! * The 'f' parameter must be between 1 and 100 (inclusive). * '-wball' does nothing if you input your own weight with the '-weight' option. * '-wball' does change the binary weight created by

the '-noweight' option. * You can only use '-wball' once in a run of 3dQwarp. **The effect of '-wball' is not dramatic.** The example above makes the average brain image across a collection of subjects a little sharper in the thalamic area, which might have some small value. If you care enough about alignment to use '-wball', then you should examine the results from 3dQwarp for each subject, to see if the alignments are good enough for your purposes.

weight	Character; file path. Instead of computing the weight from the base dataset, directly input the weight volume from dataset 'www'. Useful if you know what over parts of the base image you want to emphasize or de-emphasize the matching functional.
wmask	Character or numeric vector. Similar to '-wball', but here, you provide a dataset 'ws' that indicates where to increase the weight. * The 'ws' dataset must be on the same 3D grid as the base dataset. * 'ws' is treated as a mask – it only matters where it is nonzero – otherwise, the values inside are not used. * After 'ws' comes the factor 'f' by which to increase the automatically computed weight. Where 'ws' is nonzero, the weighting will be multiplied by (1+f). * As with '-wball', the factor 'f' should be between 1 and 100.
workhard	Logical. Iterate more times, which can help when the volumes are hard to align at all, or when you hope to get a more precise alignment. * Slows the program down (possibly a lot), of course. * When you combine '-workhard' with '-duplo', only the full size volumes get the extra iterations. * For finer control over which refinement levels work hard, you can use this option in the form (for example) -workhard:4:7 which implies the extra iterations will be done at levels 4, 5, 6, and 7, but not otherwise. * You can also use '-superhard' to iterate even more, but this extra option will REALLY slow things down. * Under most circumstances, you should not need to use either -workhard or -superhard. * The fastest way to register to a template image is via the -duplo option, and without the -workhard or -superhard options. * If you use this option in the form '-Workhard' (first letter in upper case), then the second iteration at each level is done with quintic polynomial warps.
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_re_ho

*AFNI ReHo***Description**

Compute regional homogeneity for a given neighbourhood.

Usage

```
ni_afni_re_ho(
  in_file,
  args = NULL,
  chi_sq = NULL,
  ellipsoid = NULL,
  label_set = NULL,
  mask_file = NULL,
  neighborhood = NULL,
  out_file = NULL,
  overwrite = NULL,
  sphere = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

<code>in_file</code>	Character; file path. input dataset Required .
<code>args</code>	Character. Additional parameters to the command
<code>chi_sq</code>	Logical. Output the Friedman chi-squared value in addition to the Kendall's W. This option is currently compatible only with the AFNI (BRIK/HEAD) output type; the chi-squared value will be the second sub-brick of the output dataset.
<code>ellipsoid</code>	Character or numeric vector. \ Tuple indicating the x, y, and z radius of an ellipsoid defining the neighbourhood of each voxel. The 'hood is then made according to the following relation: $(i/A)^2 + (j/B)^2 + (k/C)^2 \leq 1$. which will have approx. $V=4 \pi \frac{A B C}{3}$. The impetus for this freedom was for use with data having anisotropic voxel edge lengths.
<code>label_set</code>	Character; file path. a set of ROIs, each labelled with distinct integers. ReHo will then be calculated per ROI.
<code>mask_file</code>	Character; file path. Mask within which ReHo should be calculated voxelwise
<code>neighborhood</code>	Character; one of: "faces", "edges", "vertices". voxels in neighborhood. can be: faces (for voxel and 6 facewise neighbors, only), edges (for voxel and 18

	face- and edge-wise neighbors), vertices (for voxel and 26 face-, edge-, and node-wise neighbors).
out_file	Character; file path. Output dataset.
overwrite	Logical. overwrite output file if it already exists
sphere	Numeric. \ For additional voxelwise neighborhood control, the radius R of a desired neighborhood can be put in; R is a floating point number, and must be >1. Examples of the numbers of voxels in a given radius are as follows (you can roughly approximate with the ol' $4\pi R^3/3$ thing): * R=2.0 -> V=33 * R=2.3 -> V=57, * R=2.9 -> V=93, * R=3.1 -> V=123, * R=3.9 -> V=251, * R=4.5 -> V=389, * R=6.1 -> V=949, but you can choose most any value.
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_refit	<i>AFNI Refit</i>
---------------	-------------------

Description

Changes some of the information inside a 3D dataset's header

Usage

```
ni_afni_refit(
  in_file,
  args = NULL,
  atrcopy = NULL,
  atrfloat = NULL,
  atrint = NULL,
  atrstring = NULL,
  deoblique = NULL,
  duporigin_file = NULL,
  nosaveatr = NULL,
  saveatr = NULL,
  space = NULL,
  xdel = NULL,
```

```

xorigin = NULL,
xyzscale = NULL,
ydel = NULL,
yorigin = NULL,
zdel = NULL,
zorigin = NULL,
.cwd = NULL,
.env = NULL,
.engine = NULL,
.profile = NULL,
dry_run = FALSE,
echo = interactive()
)

```

Arguments

in_file	Character; file path. input file to 3drefit Required.
args	Character. Additional parameters to the command
atrcopy	Character or numeric vector. Copy AFNI header attribute from the given file into the header of the dataset(s) being modified. For more information on AFNI header attributes, see documentation file README.attributes. More than one '-atrcopy' option can be used. For AFNI advanced users only. Do NOT use -atrcopy or -atrstring with other modification options. See also -copyaux.
atrfloat	Character or numeric vector. Create or modify floating point attributes. The input values may be specified as a single string in quotes or as a 1D filename or string, example '1 0.2 0 0 -0.2 1 0 0 0 0 1 0' or flipZ.1D or '1D:1,0.2,2@0,-0.2,1,2@0,2@0,1,0'
atrint	Character or numeric vector. Create or modify integer attributes. The input values may be specified as a single string in quotes or as a 1D filename or string, example '1 0 0 0 0 1 0 0 0 0 1 0' or flipZ.1D or '1D:1,0,2@0,-0,1,2@0,2@0,1,0'
atrstring	Character or numeric vector. Copy the last given string into the dataset(s) being modified, giving it the attribute name given by the last string. To be safe, the last string should be in quotes.
deoblique	Logical. replace current transformation matrix with cardinal matrix
duporigin_file	Character; file path. Copies the xorigin, yorigin, and zorigin values from the header of the given dataset
nosaveatr	Logical. Opposite of -saveatr
saveatr	Logical. (default) Copy the attributes that are known to AFNI into the dataset->dblk structure thereby forcing changes to known attributes to be present in the output. This option only makes sense with -atrcopy.
space	Character; one of: "TLRC", "MNI", "ORIG". Associates the dataset with a specific template type, e.g. TLRC, MNI, ORIG
xdel	Numeric. new x voxel dimension in mm
xorigin	Character. x distance for edge voxel offset
xyzscale	Numeric. Scale the size of the dataset voxels by the given factor

ydel	Numeric. new y voxel dimension in mm
yorigin	Character. y distance for edge voxel offset
zdel	Numeric. new z voxel dimension in mm
zorigin	Character. z distance for edge voxel offset
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_remlfit	<i>AFNI Remlfit</i>
-----------------	---------------------

Description

Performs Generalized least squares time series fit with Restricted

Usage

```
ni_afni_remlfit(
  in_files,
  matrix,
  STATmask = NULL,
  addbase = NULL,
  args = NULL,
  automask = FALSE,
  dsort = NULL,
  dsort_nods = NULL,
  errts_file = NULL,
  fitts_file = NULL,
  fout = NULL,
  glt_file = NULL,
  gltsym = NULL,
  goforit = NULL,
  mask = NULL,
  matim = NULL,
  nobout = NULL,
  nodmbase = NULL,
  nofdr = NULL,
```

```

obeta = NULL,
obuck = NULL,
oerrts = NULL,
ofitts = NULL,
oglt = NULL,
out_file = NULL,
ovar = NULL,
polort = NULL,
quiet = NULL,
rbeta_file = NULL,
rout = NULL,
slibase = NULL,
slibase_sm = NULL,
tout = NULL,
usetemp = NULL,
var_file = NULL,
verb = NULL,
wherr_file = NULL,
.cwd = NULL,
.env = NULL,
.engine = NULL,
.profile = NULL,
dry_run = FALSE,
echo = interactive()
)

```

Arguments

in_files	Character or numeric vector. Read time series dataset Required.
matrix	Character; file path. the design matrix file, which should have been output from Deconvolve via the 'x1D' option Required.
STATmask	Character; file path. filename of 3D mask dataset to be used for the purpose of reporting truncation-to float issues AND for computing the FDR curves. The actual results ARE not masked with this option (only with 'mask' or 'automask' options).
adddbase	Character or numeric vector. file(s) to add baseline model columns to the matrix with this option. Each column in the specified file(s) will be appended to the matrix. File(s) must have at least as many rows as the matrix does.
args	Character. Additional parameters to the command
automask	Logical. build a mask automatically from input data (will be slow for long time series datasets)
dsort	Character; file path. 4D dataset to be used as voxelwise baseline regressor
dsort_nods	Logical. if 'dsort' option is used, this command will output additional results files excluding the 'dsort' file
errts_file	Character; file path. output dataset for REML residuals = data - fitted model
fitts_file	Character; file path. output dataset for REML fitted model

fout	Logical. output F-statistic for each stimulus
glt_file	Character; file path. output dataset for beta + statistics from the REML estimation, but ONLY for the GLTs added on the REMLfit command line itself via 'gltsym'; GLTs from Deconvolve's command line will NOT be included.
gltsym	Character or numeric vector. read a symbolic GLT from input file and associate it with a label. As in Deconvolve, you can also use the 'SYM:' method to provide the definition of the GLT directly as a string (e.g., with 'SYM: +Label1-Label2'). Unlike Deconvolve, you MUST specify 'SYM: ' if providing the GLT directly as a string instead of from a file
goforit	Logical. With potential issues flagged in the design matrix, an attempt will nevertheless be made to fit the model
mask	Character; file path. filename of 3D mask dataset; only data time series from within the mask will be analyzed; results for voxels outside the mask will be set to zero.
matim	Character; file path. read a standard file as the matrix. You can use only Col as a name in GLTs with these nonstandard matrix input methods, since the other names come from the 'matrix' file. These mutually exclusive options are ignored if 'matrix' is used.
nobout	Logical. do NOT add baseline (null hypothesis) regressor betas to the 'rbeta_file' and/or 'obeta_file' output datasets.
nodmbase	Logical. by default, baseline columns added to the matrix via 'addbase' or 'slibase' or 'dsort' will each have their mean removed (as is done in Deconvolve); this option turns this centering off
nofdr	Logical. do NOT add FDR curve data to bucket datasets; FDR curves can take a long time if 'tout' is used
obeta	Character; file path. dataset for beta weights from the OLSQ estimation
obuck	Character; file path. dataset for beta + statistics from the OLSQ estimation
oerrts	Character; file path. dataset for OLSQ residuals (data - fitted model)
ofitts	Character; file path. dataset for OLSQ fitted model
oglt	Character; file path. dataset for beta + statistics from 'gltsym' options
out_file	Character; file path. output dataset for beta + statistics from the REML estimation; also contains the results of any GLT analysis requested in the Deconvolve setup, similar to the 'bucket' output from Deconvolve. This dataset does NOT get the betas (or statistics) of those regressors marked as 'baseline' in the matrix file.
ovar	Character; file path. dataset for OLSQ st.dev. parameter (kind of boring)
polort	Integer. if no 'matrix' option is given, AND no 'matim' option, create a matrix with Legendre polynomial regressors up to the specified order. The default value is 0, which produces a matrix with a single column of all ones
quiet	Logical. turn off most progress messages
rbeta_file	Character; file path. output dataset for beta weights from the REML estimation, similar to the 'bucket' output from Deconvolve. This dataset will contain all the beta weights, for baseline and stimulus regressors alike, unless the '-nobout' option is given – in that case, this dataset will only get the betas for the stimulus regressors.

rout	Logical. output the R ² statistic for each stimulus
slibase	Character or numeric vector. similar to 'addbase' in concept, BUT each specified file must have an integer multiple of the number of slices in the input dataset(s); then, separate regression matrices are generated for each slice, with the first column of the file appended to the matrix for the first slice of the dataset, the second column of the file appended to the matrix for the first slice of the dataset, and so on. Intended to help model physiological noise in FMRI, or other effects you want to regress out that might change significantly in the inter-slice time intervals. This will slow the program down, and make it use a lot more memory (to hold all the matrix stuff).
slibase_sm	Character or numeric vector. similar to 'slibase', BUT each file must be in slice major order (i.e. all slice0 columns come first, then all slice1 columns, etc).
tout	Logical. output the T-statistic for each stimulus; if you use 'out_file' and do not give any of 'fout', 'tout', or 'rout', then the program assumes 'fout' is activated.
usetemp	Logical. write intermediate stuff to disk, to economize on RAM. Using this option might be necessary to run with 'slibase' and with 'Grid' values above the default, since the program has to store a large number of matrices for such a problem: two for every slice and for every (a,b) pair in the ARMA parameter grid. Temporary files are written to the directory given in environment variable TMPDIR, or in /tmp, or in ./ (preference is in that order)
var_file	Character; file path. output dataset for REML variance parameters
verb	Logical. turns on more progress messages, including memory usage progress reports at various stages
wherr_file	Character; file path. dataset for REML residual, whitened using the estimated ARMA(1,1) correlation matrix of the noise
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_resample	<i>AFNI Resample</i>
------------------	----------------------

Description

Resample or reorient an image using AFNI 3dresample command

Usage

```

ni_afni_resample(
  in_file,
  args = NULL,
  master = NULL,
  orientation = NULL,
  out_file = NULL,
  resample_mode = NULL,
  voxel_size = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)

```

Arguments

<code>in_file</code>	Character; file path. input file to 3dresample Required.
<code>args</code>	Character. Additional parameters to the command
<code>master</code>	Character; file path. align dataset grid to a reference file
<code>orientation</code>	Character. new orientation code
<code>out_file</code>	Character; file path. output image file name
<code>resample_mode</code>	Character; one of: "NN", "Li", "Cu", "Bk". resampling method from set {"NN", "Li", "Cu", "Bk"}. These are for "Nearest Neighbor", "Linear", "Cubic" and "Blocky" interpolation, respectively. Default is NN.
<code>voxel_size</code>	Character or numeric vector. resample to new dx, dy and dz
<code>.cwd</code>	Working directory override.
<code>.env</code>	Named character vector of environment variables.
<code>.engine</code>	Execution engine override.
<code>.profile</code>	Runtime profile override.
<code>dry_run</code>	Logical; preview command without executing.
<code>echo</code>	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

 ni_afni_retroicor *AFNI Retroicor*

Description

Performs Retrospective Image Correction for physiological

Usage

```
ni_afni_retroicor(
  in_file,
  args = NULL,
  card = NULL,
  cardphase = NULL,
  order = NULL,
  out_file = NULL,
  resp = NULL,
  respphase = NULL,
  threshold = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. input file to 3dretroicor Required.
args	Character. Additional parameters to the command
card	Character; file path. 1D cardiac data file for cardiac correction
cardphase	Character; file path. Filename for 1D cardiac phase output
order	Integer. The order of the correction (2 is typical)
out_file	Character; file path. output image file name
resp	Character; file path. 1D respiratory waveform data for correction
respphase	Character; file path. Filename for 1D resp phase output
threshold	Integer. Threshold for detection of R-wave peaks in input (Make sure it is above the background noise level, Try 3/4 or 4/5 times range plus minimum)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_roi_stats	<i>AFNI ROIStats</i>
-------------------	----------------------

Description

Display statistics over masked regions

Usage

```
ni_afni_roi_stats(
  in_file,
  args = NULL,
  debug = NULL,
  format1D = NULL,
  format1DR = NULL,
  mask = NULL,
  mask_f2short = NULL,
  mask_file = NULL,
  nobriklab = NULL,
  nomeanout = NULL,
  num_roi = NULL,
  out_file = NULL,
  quiet = NULL,
  roisel = NULL,
  stat = NULL,
  zerofill = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. input dataset Required.
args	Character. Additional parameters to the command
debug	Logical. print debug information
format1D	Logical. Output results in a 1D format that includes commented labels
format1DR	Logical. Output results in a 1D format that includes uncommented labels. May not work optimally with typical 1D functions, but is useful for R functions.

mask	Character; file path. input mask
mask_f2short	Logical. Tells the program to convert a float mask to short integers, by simple rounding.
mask_file	Character; file path. input mask
nobriklab	Logical. Do not print the sub-brick label next to its index
nomeanout	Logical. Do not include the (zero-inclusive) mean among computed stats
num_roi	Integer. Forces the assumption that the mask dataset's ROIs are denoted by 1 to n inclusive. Normally, the program figures out the ROIs on its own. This option is useful if a) you are certain that the mask dataset has no values outside the range [0 n], b) there may be some ROIs missing between [1 n] in the mask data-set and c) you want those columns in the output any-way so the output lines up with the output from other invocations of 3dROIstats.
out_file	Character; file path. output file
quiet	Logical. execute quietly
roisel	Character; file path. Only considers ROIs denoted by values found in the specified file. Note that the order of the ROIs as specified in the file is not preserved. So an SEL.1D of '2 8 20' produces the same output as '8 20 2'
stat	Character or numeric vector. Statistics to compute. Options include: * mean = Compute the mean using only non_zero voxels. Implies the opposite for the mean computed by default. * median = Compute the median of nonzero voxels * mode = Compute the mode of nonzero voxels. (integral valued sets only) * minmax = Compute the min/max of nonzero voxels * sum = Compute the sum using only nonzero voxels. * voxels = Compute the number of nonzero voxels * sigma = Compute the standard deviation of nonzero voxels Statistics that include zero-valued voxels: * zerominmax = Compute the min/max of all voxels. * zerosigma = Compute the standard deviation of all voxels. * zeromedian = Compute the median of all voxels. * zeromode = Compute the mode of all voxels. * summary = Only output a summary line with the grand mean across all briks in the input dataset. This option cannot be used with nomeanout. More than one option can be specified.
zerofill	Character. For ROI labels not found, use the provided string instead of a '0' in the output file. Only active if num_roi is enabled.
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_seg

AFNI Seg

Description

3dSeg segments brain volumes into tissue classes. The program allows

Usage

```
ni_afni_seg(
  in_file,
  mask,
  args = NULL,
  bias_classes = NULL,
  bias_fwhm = NULL,
  blur_meth = NULL,
  bmrif = NULL,
  classes = NULL,
  main_N = NULL,
  mixfloor = NULL,
  mixfrac = NULL,
  prefix = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. ANAT is the volume to segment Required.
mask	Character or numeric vector. only non-zero voxels in mask are analyzed. mask can either be a dataset or the string "AUTO" which would use AFNI's automask function to create the mask. Required.
args	Character. Additional parameters to the command
bias_classes	Character. A semicolon delimited string of classes that contribute to the estimation of the bias field
bias_fwhm	Numeric. The amount of blurring used when estimating the field bias with the Wells method
blur_meth	Character; one of: "BFT", "BIM". set the blurring method for bias field estimation
bmrif	Numeric. Weighting factor controlling spatial homogeneity of the classifications
classes	Character. CLASS_STRING is a semicolon delimited string of class labels

main_N	Integer. Number of iterations to perform.
mixfloor	Numeric. Set the minimum value for any class's mixing fraction
mixfrac	Character. MIXFRAC sets up the volume-wide (within mask) tissue fractions while initializing the segmentation (see IGNORE for exception)
prefix	Character. the prefix for the output folder containing all output volumes
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_skull_strip *AFNI SkullStrip*

Description

A program to extract the brain from surrounding tissue from MRI

Usage

```
ni_afni_skull_strip(
  in_file,
  args = NULL,
  out_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. input file to 3dSkullStrip Required.
args	Character. Additional parameters to the command
out_file	Character; file path. output image file name
.cwd	Working directory override.

.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_svm_test	<i>AFNI SVMTest</i>
------------------	---------------------

Description

Temporally predictive modeling with the support vector machine

Usage

```
ni_afni_svm_test(
  in_file,
  model,
  args = NULL,
  classout = NULL,
  multiclass = NULL,
  nodetrend = NULL,
  nopredcensord = NULL,
  options = NULL,
  out_file = NULL,
  testlabels = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. A 3D or 3D+t AFNI brick dataset to be used for testing. Required.
model	Character. modname is the basename for the brick containing the SVM model Required.
args	Character. Additional parameters to the command

classout	Logical. Flag to specify that pname files should be integer-valued, corresponding to class category decisions.
multiclass	Logical. Specifies multiclass algorithm for classification
nodetrend	Logical. Flag to specify that pname files should not be linearly detrended
nopredcensord	Logical. Flag to prevent writing predicted values for censored time-points
options	Character. additional options for SVM-light
out_file	Character; file path. filename for .1D prediction file(s).
testlabels	Character; file path. <i>true</i> class category .1D labels for the test dataset. It is used to calculate the prediction accuracy performance
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_svm_train	<i>AFNI SVMTrain</i>
-------------------	----------------------

Description

Temporally predictive modeling with the support vector machine

Usage

```
ni_afni_svm_train(
  in_file,
  ttype,
  alphas = NULL,
  args = NULL,
  censor = NULL,
  kernel = NULL,
  mask = NULL,
  max_iterations = NULL,
  model = NULL,
  nomodelmask = NULL,
  options = NULL,
  out_file = NULL,
  trainlabels = NULL,
  w_out = NULL,
```

```

    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

in_file	Character; file path. A 3D+t AFNI brick dataset to be used for training. Required.
ttype	Character. tname: classification or regression Required.
alphas	Character; file path. output alphas file name
args	Character. Additional parameters to the command
sensor	Character; file path. .1D sensor file that allows the user to ignore certain samples in the training data.
kernel	Character. string specifying type of kernel function: linear, polynomial, rbf, sigmoid
mask	Character; file path. byte-format brick file used to mask voxels in the analysis
max_iterations	Integer. Specify the maximum number of iterations for the optimization.
model	Character; file path. basename for the brick containing the SVM model
nomodelmask	Logical. Flag to enable the omission of a mask file
options	Character. additional options for SVM-light
out_file	Character; file path. output sum of weighted linear support vectors file name
trainlabels	Character; file path. .1D labels corresponding to the stimulus paradigm for the training data.
w_out	Logical. output sum of weighted linear support vectors
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

 ni_afni_synthesize *AFNI Synthesize*

Description

Reads a '-cbucket' dataset and a '.xmat.1D' matrix from 3dDeconvolve,

Usage

```
ni_afni_synthesize(
  cbucket,
  matrix,
  select,
  TR = NULL,
  args = NULL,
  cenfill = NULL,
  dry_run_arg = NULL,
  out_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

cbucket	Character; file path. Read the dataset output from 3dDeconvolve via the '-cbucket' option. Required.
matrix	Character; file path. Read the matrix output from 3dDeconvolve via the '-x1D' option. Required.
select	Character or numeric vector. A list of selected columns from the matrix (and the corresponding coefficient sub-bricks from the cbucket). Valid types include 'baseline', 'polort', 'allfunc', 'allstim', 'all', Can also provide 'something' where something matches a stim_label from 3dDeconvolve, and 'digits' where digits are the numbers of the select matrix columns by numbers (starting at 0), or number ranges of the form '3..7' and '3-7'. Required.
TR	Numeric. TR to set in the output. The default value of TR is read from the header of the matrix file.
args	Character. Additional parameters to the command
cenfill	Character; one of: "zero", "nbhr", "none". Determines how censored time points from the 3dDeconvolve run will be filled. Valid types are 'zero', 'nbhr' and 'none'.
dry_run_arg	Logical. Don't compute the output, just check the inputs. (spec input: dry_run)

out_file	Character; file path. output dataset prefix name (default 'syn')
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_t_cat	<i>AFNI TCat</i>
---------------	------------------

Description

Concatenate sub-bricks from input datasets into one big 3D+time dataset.

Usage

```
ni_afni_t_cat(
  in_files,
  args = NULL,
  out_file = NULL,
  rlt = NULL,
  verbose = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_files	Character or numeric vector. input file to 3dTcat Required.
args	Character. Additional parameters to the command
out_file	Character; file path. output image file name
rlt	Character; one of: "", "+", "++". Remove linear trends in each voxel time series loaded from each input dataset, SEPARATELY. Option -rlt removes the least squares fit of 'a+b*t' to each voxel time series. Option -rlt+ adds dataset mean back in. Option -rlt++ adds overall mean of all dataset timeseries back in.

verbose	Logical. Print out some verbose output as the program
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_t_cat_sub_brick
AFNI TCatSubBrick

Description

Hopefully a temporary function to allow sub-brick selection until

Usage

```
ni_afni_t_cat_sub_brick(
  in_files,
  args = NULL,
  out_file = NULL,
  rlt = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_files	Character or numeric vector. List of tuples of file names and subbrick selectors as strings. Don't forget to protect the single quotes in the subbrick selector so the contents are protected from the command line interpreter. Required.
args	Character. Additional parameters to the command
out_file	Character; file path. output image file name
rlt	Character; one of: "", "+", "++". Remove linear trends in each voxel time series loaded from each input dataset, SEPARATELY. Option -rlt removes the least squares fit of 'a+b*t' to each voxel time series. Option -rlt+ adds dataset mean back in. Option -rlt++ adds overall mean of all dataset timeseries back in.

.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

ni_afni_t_corr_map	<i>AFNI TCorrMap</i>
--------------------	----------------------

Description

For each voxel time series, computes the correlation between it

Usage

```
ni_afni_t_corr_map(
  in_file,
  absolute_threshold = NULL,
  args = NULL,
  automask = NULL,
  average_expr = NULL,
  average_expr_nonzero = NULL,
  bandpass = NULL,
  blur_fwhm = NULL,
  correlation_maps = NULL,
  correlation_maps_masked = NULL,
  histogram = NULL,
  mask = NULL,
  mean_file = NULL,
  out_file = NULL,
  pmean = NULL,
  polort = NULL,
  qmean = NULL,
  regress_out_timeseries = NULL,
  seeds = NULL,
  seeds_width = NULL,
  sum_expr = NULL,
  var_absolute_threshold = NULL,
  var_absolute_threshold_normalize = NULL,
  zmean = NULL,
  .cwd = NULL,
```

```

.env = NULL,
.engine = NULL,
.profile = NULL,
.dry_run = FALSE,
.echo = interactive()
)

```

Arguments

in_file	Character; file path Required.
absolute_threshold	Character; file path
args	Character. Additional parameters to the command
automask	Logical
average_expr	Character; file path
average_expr_nonzero	Character; file path
bandpass	Character or numeric vector
blur_fwhm	Numeric
correlation_maps	Character; file path
correlation_maps_masked	Character; file path
histogram	Character; file path
mask	Character; file path
mean_file	Character; file path
out_file	Character; file path. output image file name
pmean	Character; file path
polort	Integer
qmean	Character; file path
regress_out_timeseries	Character; file path
seeds	Character; file path
seeds_width	Numeric
sum_expr	Character; file path
var_absolute_threshold	Character; file path
var_absolute_threshold_normalize	Character; file path
zmean	Character; file path
.cwd	Working directory override.
.env	Named character vector of environment variables.

.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_t_corr1_d	<i>AFNI TCorrID</i>
-------------------	---------------------

Description

Computes the correlation coefficient between each voxel time series

Usage

```
ni_afni_t_corr1_d(
  xset,
  y_1d,
  args = NULL,
  ktaub = NULL,
  out_file = NULL,
  pearson = NULL,
  quadrant = NULL,
  spearman = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

xset	Character; file path. 3d+time dataset input Required.
y_1d	Character; file path. 1D time series file input Required.
args	Character. Additional parameters to the command
ktaub	Logical. Correlation is the Kendall's tau_b correlation coefficient
out_file	Character; file path. output filename prefix
pearson	Logical. Correlation is the normal Pearson correlation coefficient
quadrant	Logical. Correlation is the quadrant correlation coefficient

spearman	Logical. Correlation is the Spearman (rank) correlation coefficient
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_t_correlate *AFNI TCorrelate*

Description

Computes the correlation coefficient between corresponding voxel

Usage

```
ni_afni_t_correlate(
  xset,
  yset,
  args = NULL,
  out_file = NULL,
  pearson = NULL,
  polort = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

xset	Character; file path. input xset Required.
yset	Character; file path. input yset Required.
args	Character. Additional parameters to the command
out_file	Character; file path. output image file name
pearson	Logical. Correlation is the normal Pearson correlation coefficient
polort	Integer. Remove polynomial trend of order m

.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_t_norm	<i>AFNI TNorm</i>
----------------	-------------------

Description

Shifts voxel time series from input so that separate slices are aligned

Usage

```
ni_afni_t_norm(
  in_file,
  L1fit = NULL,
  args = NULL,
  norm1 = NULL,
  norm2 = NULL,
  normR = NULL,
  normx = NULL,
  out_file = NULL,
  polort = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. input file to 3dTNorm Required.
L1fit	Logical. Detrend with L1 regression (L2 is the default) This option is here just for the hell of it
args	Character. Additional parameters to the command
norm1	Logical. L1 normalize (sum of absolute values = 1)

norm2	Logical. L2 normalize (sum of squares = 1) [DEFAULT]
normR	Logical. normalize so sum of squares = number of time points $\sqrt{*}$ e.g., so RMS = 1.
normx	Logical. Scale so max absolute value = 1 (L_infinity norm)
out_file	Character; file path. output image file name
polort	Integer. Detrend with polynomials of order p before normalizing [DEFAULT = don't do this]. Use '-polort 0' to remove the mean, for example
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_t_project	<i>AFNI TProject</i>
-------------------	----------------------

Description

This program projects (detrends) out various 'nuisance' time series from

Usage

```
ni_afni_t_project(
  in_file,
  TR = NULL,
  args = NULL,
  automask = NULL,
  bandpass = NULL,
  blur = NULL,
  cenmode = NULL,
  censor = NULL,
  censortr = NULL,
  concat = NULL,
  dsort = NULL,
  mask = NULL,
  noblock = NULL,
  norm = NULL,
  ort = NULL,
  out_file = NULL,
```

```

    polort = NULL,
    stopband = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

in_file	Character; file path. input file to 3dTproject Required.
TR	Numeric. Use time step dd for the frequency calculations, rather than the value stored in the dataset header.
args	Character. Additional parameters to the command
automask	Logical. Generate a mask automatically
bandpass	Character or numeric vector. Remove all frequencies EXCEPT those in the range
blur	Numeric. Blur (inside the mask only) with a filter that has width (FWHM) of fff millimeters. Spatial blurring (if done) is after the time series filtering.
cenmode	Character; one of: "KILL", "ZERO", "NTRP". Specifies how censored time points are treated in the output dataset: * mode = ZERO – put zero values in their place; output dataset is same length as input * mode = KILL – remove those time points; output dataset is shorter than input * mode = NTRP – censored values are replaced by interpolated neighboring (in time) non-censored values, BEFORE any projections, and then the analysis proceeds without actual removal of any time points – this feature is to keep the Spanish Inquisition happy. * The default mode is KILL !!!
cancel	Character; file path. Filename of censor .1D time series. This is a file of 1s and 0s, indicating which time points are to be included (1) and which are to be excluded (0).
canceltr	Character or numeric vector. List of strings that specify time indexes to be removed from the analysis. Each string is of one of the following forms: * 37 => remove global time index #37 * 2:37 => remove time index #37 in run #2 * 37..47 => remove global time indexes #37-47 * 37-47 => same as above * 2:37..47 => remove time indexes #37-47 in run #2 * *:0-2 => remove time indexes #0-2 in all runs * Time indexes within each run start at 0. * Run indexes start at 1 (just be to confusing). * N.B.: 2:37,47 means index #37 in run #2 and global time index 47; it does NOT mean index #37 in run #2 AND index #47 in run #2.
concat	Character; file path. The catenation file, as in 3dDeconvolve, containing the TR indexes of the start points for each contiguous run within the input dataset (the first entry should be 0). * Also as in 3dDeconvolve, if the input dataset is automatically catenated from a collection of datasets, then the run start indexes are determined directly, and '-concat' is not needed (and will be ignored). * Each

run must have at least 9 time points AFTER censoring, or the program will not work! * The only use made of this input is in setting up the bandpass/stopband regressors. * '-ort' and '-dsort' regressors run through all time points, as read in. If you want separate projections in each run, then you must either break these ort files into appropriate components, OR you must run 3dTproject for each run separately, using the appropriate pieces from the ort files via the \{ . . \} selector for the 1D files and the \[. . \] selector for the datasets.

dsort	Character or numeric vector. Remove the 3D+time time series in dataset fset. * That is, 'fset' contains a different nuisance time series for each voxel (e.g., from AnatICOR). * Multiple -dsort options are allowed.
mask	Character; file path. Only operate on voxels nonzero in the mset dataset. * Voxels outside the mask will be filled with zeros. * If no masking option is given, then all voxels will be processed.
noblock	Logical. Also as in 3dDeconvolve, if you want the program to treat an auto-catenated dataset as one long run, use this option. However, '-noblock' will not affect catenation if you use the '-concat' option.
norm	Logical. Normalize each output time series to have sum of squares = 1. This is the LAST operation.
ort	Character; file path. Remove each column in file. Each column will have its mean removed.
out_file	Character; file path. output image file name
polort	Integer. Remove polynomials up to and including degree pp. * Default value is 2. * It makes no sense to use a value of pp greater than 2, if you are bandpassing out the lower frequencies! * For catenated datasets, each run gets a separate set set of pp+1 Legendre polynomial regressors. * Use of -polort -1 is not advised (if data mean != 0), even if -ort contains constant terms, as all means are removed.
stopband	Character or numeric vector. Remove all frequencies in the range
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_t_shift	<i>AFNI TShift</i>
-----------------	--------------------

Description

Shifts voxel time series from input so that separate slices are aligned

Usage

```
ni_afni_t_shift(
  in_file,
  args = NULL,
  ignore = NULL,
  interp = NULL,
  out_file = NULL,
  rlt = NULL,
  rltplus = NULL,
  slice_timing = NULL,
  tpattern = NULL,
  tr = NULL,
  tslice = NULL,
  tzero = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

<code>in_file</code>	Character; file path. input file to 3dTshift Required.
<code>args</code>	Character. Additional parameters to the command
<code>ignore</code>	Integer. ignore the first set of points specified
<code>interp</code>	Character; one of: "Fourier", "linear", "cubic", "quintic", "heptic". different interpolation methods (see 3dTshift for details) default = Fourier
<code>out_file</code>	Character; file path. output image file name
<code>rlt</code>	Logical. Before shifting, remove the mean and linear trend
<code>rltplus</code>	Logical. Before shifting, remove the mean and linear trend and later put back the mean
<code>slice_timing</code>	Character or numeric vector. time offsets from the volume acquisition onset for each slice
<code>tpattern</code>	Character or numeric vector. use specified slice time pattern rather than one in header

tr	Character. manually set the TR. You can attach suffix "s" for seconds or "ms" for milliseconds.
tslice	Integer. align each slice to time offset of given slice
tzero	Numeric. align each slice to given time offset
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_t_smooth	<i>AFNI TSmooth</i>
------------------	---------------------

Description

Smooths each voxel time series in a 3D+time dataset and produces

Usage

```
ni_afni_t_smooth(
  in_file,
  adaptive = NULL,
  args = NULL,
  blackman = NULL,
  custom = NULL,
  datum = NULL,
  hamming = NULL,
  lin = NULL,
  lin3 = NULL,
  med = NULL,
  osf = NULL,
  out_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. input file to 3dTSmooth Required.
adaptive	Integer. use adaptive mean filtering of width N (where N must be odd and bigger than 3).
args	Character. Additional parameters to the command
blackman	Integer. Use N point Blackman windows. (N must be odd and bigger than 1.)
custom	Character; file path. odd # of coefficients must be in a single column in ASCII file
datum	Character. Sets the data type of the output dataset
hamming	Integer. Use N point Hamming windows. (N must be odd and bigger than 1.)
lin	Logical. 3 point linear filter: $0.15a + 0.70b + 0.15c$ [This is the default smoother]
lin3	Integer. 3 point linear filter: $0.5(1-m)a + m b + 0.5(1-m)c$. Here, 'm' is a number strictly between 0 and 1.
med	Logical. 3 point median filter: median(a,b,c)
osf	Logical. 3 point order statistics filter: $0.15\min(a,b,c) + 0.70\text{median}(a,b,c) + 0.15\max(a,b,c)$
out_file	Character; file path. output file from 3dTSmooth
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_t_stat

AFNI TStat

Description

Compute voxel-wise statistics using AFNI 3dTstat command

Usage

```

ni_afni_t_stat(
  in_file,
  args = NULL,
  mask = NULL,
  options = NULL,
  out_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)

```

Arguments

in_file	Character; file path. input file to 3dTstat Required.
args	Character. Additional parameters to the command
mask	Character; file path. mask file
options	Character. selected statistical output
out_file	Character; file path. output image file name
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_to3_d

AFNI To3D

Description

Create a 3D dataset from 2D image files using AFNI to3d command

Usage

```

ni_afni_to3_d(
  in_folder,
  args = NULL,
  assumemosaic = NULL,
  datatype = NULL,
  filetype = NULL,
  funcparams = NULL,
  out_file = NULL,
  skipoutliers = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)

```

Arguments

in_folder	Character; directory path. folder with DICOM images to convert Required.
args	Character. Additional parameters to the command
assumemosaic	Logical. assume that Siemens image is mosaic
datatype	Character; one of: "short", "float", "byte", "complex". set output file datatype
filetype	Character; one of: "spgr", "fse", "epan", "anat", "ct", "spct", "pet", "mra", "bmap", "diff", "omri", "abuc", "fim", "fith", "fico", "fitt", "fift", "fizt", "fict", "fibt", "fibn", "figt", "fipt", "fbuc". type of datafile being converted
funcparams	Character. parameters for functional data
out_file	Character; file path. output image file name
skipoutliers	Logical. skip the outliers check
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

 ni_afni_undump

 AFNI Undump

Description

3dUndump - Assembles a 3D dataset from an ASCII list of coordinates and

Usage

```
ni_afni_undump(
  in_file,
  args = NULL,
  coordinates_specification = NULL,
  datatype = NULL,
  default_value = NULL,
  fill_value = NULL,
  head_only = NULL,
  mask_file = NULL,
  orient = NULL,
  out_file = NULL,
  srad = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. input file to 3dUndump, whose geometry will determinethe geometry of the output Required.
args	Character. Additional parameters to the command
coordinates_specification	Character; one of: "ijk", "xyz". Coordinates in the input file as index triples (i, j, k) or spatial coordinates (x, y, z) in mm
datatype	Character; one of: "short", "float", "byte". set output file datatype
default_value	Numeric. default value stored in each input voxel that does not have a value supplied in the input file
fill_value	Numeric. value, used for each voxel in the output dataset that is NOT listed in the input file
head_only	Logical. create only the .HEAD file which gets exploited by the AFNI matlab library function New_HEAD.m

mask_file	Character; file path. mask image file name. Only voxels that are nonzero in the mask can be set.
orient	Character or numeric vector. Specifies the coordinate order used by -xyz. The code must be 3 letters, one each from the pairs {R,L} {A,P} {I,S}. The first letter gives the orientation of the x-axis, the second the orientation of the y-axis, the third the z-axis: R = right-to-left L = left-to-right A = anterior-to-posterior P = posterior-to-anterior I = inferior-to-superior S = superior-to-inferior If -orient isn't used, then the coordinate order of the -master (in_file) dataset is used to interpret (x,y,z) inputs.
out_file	Character; file path. output image file name
srad	Numeric. radius in mm of the sphere that will be filled about each input (x,y,z) or (i,j,k) voxel. If the radius is not given, or is 0, then each input data line sets the value in only one voxel.
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_unifize	<i>AFNI Unifize</i>
-----------------	---------------------

Description

3dUnifize - for uniformizing image intensity

Usage

```
ni_afni_unifize(
  in_file,
  args = NULL,
  cl_frac = NULL,
  epi = NULL,
  gm = NULL,
  no_duplo = NULL,
  out_file = NULL,
  quiet = NULL,
  rbt = NULL,
  scale_file = NULL,
```

```

t2 = NULL,
t2_up = NULL,
urad = NULL,
.cwd = NULL,
.env = NULL,
.engine = NULL,
.profile = NULL,
dry_run = FALSE,
echo = interactive()
)

```

Arguments

in_file	Character; file path. input file to 3dUnifize Required.
args	Character. Additional parameters to the command
cl_frac	Numeric. Option for AFNI experts only. Set the automask 'clip level fraction'. Must be between 0.1 and 0.9. A small fraction means to make the initial threshold for clipping (a la 3dClipLevel) smaller, which will tend to make the mask larger. [default=0.1]
epi	Logical. Assume the input dataset is a T2 (or T2*) weighted EPI time series. After computing the scaling, apply it to ALL volumes (TRs) in the input dataset. That is, a given voxel will be scaled by the same factor at each TR. This option also implies '-noduplo' and '-T2'. This option turns off '-GM' if you turned it on.
gm	Logical. Also scale to unifize 'gray matter' = lower intensity voxels (to aid in registering images from different scanners).
no_duplo	Logical. Do NOT use the 'duplo down' step; this can be useful for lower resolution datasets.
out_file	Character; file path. output image file name
quiet	Logical. Don't print the progress messages.
rbt	Character or numeric vector. Option for AFNI experts only. Specify the 3 parameters for the algorithm: R = radius; same as given by option '-Urad', [default=18.3] b = bottom percentile of normalizing data range, [default=70.0] r = top percentile of normalizing data range, [default=80.0]
scale_file	Character; file path. output file name to save the scale factor used at each voxel
t2	Logical. Treat the input as if it were T2-weighted, rather than T1-weighted. This processing is done simply by inverting the image contrast, processing it as if that result were T1-weighted, and then re-inverting the results counts of voxel overlap, i.e., each voxel will contain the number of masks that it is set in.
t2_up	Numeric. Option for AFNI experts only. Set the upper percentile point used for T2-T1 inversion. Allowed to be anything between 90 and 100 (inclusive), with default to 98.5 (for no good reason).
urad	Numeric. Sets the radius (in voxels) of the ball used for the sneaky trick. Default value is 18.3, and should be changed proportionally if the dataset voxel size differs significantly from 1 mm.

.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

ni_afni_volreg	<i>AFNI Volreg</i>
----------------	--------------------

Description

Register input volumes to a base volume using AFNI 3dvolreg command

Usage

```
ni_afni_volreg(
  in_file,
  args = NULL,
  basefile = NULL,
  copyorigin = NULL,
  in_weight_volume = NULL,
  interp = NULL,
  md1d_file = NULL,
  oned_file = NULL,
  oned_matrix_save = NULL,
  out_file = NULL,
  timeshift = NULL,
  verbose = NULL,
  zpad = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. input file to 3dvolreg Required.
args	Character. Additional parameters to the command
basefile	Character; file path. base file for registration
copyorigin	Logical. copy base file origin coords to output
in_weight_volume	Character or numeric vector. weights for each voxel specified by a file with an optional volume number (defaults to 0)
interp	Character; one of: "Fourier", "cubic", "heptic", "quintic", "linear". spatial interpolation methods [default = heptic]
md1d_file	Character; file path. max displacement output file
oned_file	Character; file path. 1D movement parameters output file
oned_matrix_save	Character; file path. Save the matrix transformation
out_file	Character; file path. output image file name
timeshift	Logical. time shift to mean slice time offset
verbose	Logical. more detailed description of the process
zpad	Integer. Zeropad around the edges by 'n' voxels during rotations
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_warp

AFNI Warp

Description

Use 3dWarp for spatially transforming a dataset.

Usage

```

ni_afni_warp(
  in_file,
  args = NULL,
  deoblique = NULL,
  gridset = NULL,
  interp = NULL,
  matparent = NULL,
  mni2tta = NULL,
  newgrid = NULL,
  oblique_parent = NULL,
  out_file = NULL,
  tta2mni = NULL,
  verbose = NULL,
  zpad = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)

```

Arguments

<code>in_file</code>	Character; file path. input file to 3dWarp Required.
<code>args</code>	Character. Additional parameters to the command
<code>deoblique</code>	Logical. transform dataset from oblique to cardinal
<code>gridset</code>	Character; file path. copy grid of specified dataset
<code>interp</code>	Character; one of: "linear", "cubic", "NN", "quintic". spatial interpolation methods [default = linear]
<code>matparent</code>	Character; file path. apply transformation from 3dWarpDrive
<code>mni2tta</code>	Logical. transform dataset from MNI152 to Talairach
<code>newgrid</code>	Numeric. specify grid of this size (mm)
<code>oblique_parent</code>	Character; file path. Read in the oblique transformation matrix from an oblique dataset and make cardinal dataset oblique to match
<code>out_file</code>	Character; file path. output image file name
<code>tta2mni</code>	Logical. transform dataset from Talairach to MNI152
<code>verbose</code>	Logical. Print out some information along the way.
<code>zpad</code>	Integer. pad input dataset with N planes of zero on all sides.
<code>.cwd</code>	Working directory override.
<code>.env</code>	Named character vector of environment variables.
<code>.engine</code>	Execution engine override.

.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_z_cut_up	<i>AFNI ZCutUp</i>
------------------	--------------------

Description

Cut z-slices from a volume using AFNI 3dZcutup command

Usage

```
ni_afni_z_cut_up(
  in_file,
  args = NULL,
  keep = NULL,
  out_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. input file to 3dZcutup Required.
args	Character. Additional parameters to the command
keep	Character. slice range to keep in output
out_file	Character; file path. output image file name
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_zcat

AFNI Zcat

Description

Copies an image of one type to an image of the same

Usage

```
ni_afni_zcat(
  in_files,
  args = NULL,
  datum = NULL,
  fscale = NULL,
  nscale = NULL,
  out_file = NULL,
  verb = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_files	Character or numeric vector Required.
args	Character. Additional parameters to the command
datum	Character; one of: "byte", "short", "float". specify data type for output. Valid types are 'byte', 'short' and 'float'.
fscale	Logical. Force scaling of the output to the maximum integer range. This only has effect if the output datum is byte or short (either forced or defaulted). This option is sometimes necessary to eliminate unpleasant truncation artifacts.
nscale	Logical. Don't do any scaling on output to byte or short datasets. This may be especially useful when operating on mask datasets whose output values are only 0's and 1's.
out_file	Character; file path. output dataset prefix name (default 'zcat')
verb	Logical. print out some verbosity as the program proceeds.
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_afni_zeropad	<i>AFNI Zeropad</i>
-----------------	---------------------

Description

Adds planes of zeros to a dataset (i.e., pads it out).

Usage

```
ni_afni_zeropad(
  in_files,
  A = NULL,
  AP = NULL,
  I = NULL,
  IS = NULL,
  L = NULL,
  P = NULL,
  R = NULL,
  RL = NULL,
  S = NULL,
  args = NULL,
  master = NULL,
  mm = NULL,
  out_file = NULL,
  z = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_files	Character; file path. input dataset Required.
A	Integer. adds 'n' planes of zero at the Anterior edge
AP	Integer. specify that planes should be added or cut symmetrically to make the resulting volume haveN slices in the anterior-posterior direction
I	Integer. adds 'n' planes of zero at the Inferior edge
IS	Integer. specify that planes should be added or cut symmetrically to make the resulting volume haveN slices in the inferior-superior direction

L	Integer. adds 'n' planes of zero at the Left edge
P	Integer. adds 'n' planes of zero at the Posterior edge
R	Integer. adds 'n' planes of zero at the Right edge
RL	Integer. specify that planes should be added or cut symmetrically to make the resulting volume have N slices in the right-left direction
S	Integer. adds 'n' planes of zero at the Superior edge
args	Character. Additional parameters to the command
master	Character; file path. match the volume described in dataset 'mset', where mset must have the same orientation and grid spacing as dataset to be padded. the goal of -master is to make the output dataset from 3dZeropad match the spatial 'extents' of mset by adding or subtracting slices as needed. You can't use -I,-S,..., or -mm with -master
mm	Logical. pad counts 'n' are in mm instead of slices, where each 'n' is an integer and at least 'n' mm of slices will be added/removed; e.g., n = 3 and slice thickness = 2.5 mm ==> 2 slices added
out_file	Character; file path. output dataset prefix name (default 'zeropad')
z	Integer. adds 'n' planes of zero on EACH of the dataset z-axis (slice-direction) faces
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_ants_affine_initializer
ANTS AffineInitializer

Description

Initialize an affine transform (as in antsBrainExtraction.sh)

Usage

```
ni_ants_affine_initializer(
  fixed_image,
  moving_image,
  args = NULL,
  dimension = 3,
  local_search = 10,
  out_file = "transform.mat",
  principal_axes = FALSE,
  radian_fraction = 0.1,
  search_factor = 15,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

fixed_image	Character; file path. reference image Required.
moving_image	Character; file path. moving image Required.
args	Character. Additional parameters to the command
dimension	Character; one of: "3", "2". dimension
local_search	Integer. determines if a local optimization is run at each search point for the set number of iterations
out_file	Character; file path. output transform file
principal_axes	Logical. whether the rotation is searched around an initial principal axis alignment.
radian_fraction	Character. search this arc +/- principal axes
search_factor	Numeric. increments (degrees) for affine search
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_ants_ai	ANTS AI
------------	---------

Description

Calculate the optimal linear transform parameters for aligning two images.

Usage

```
ni_ants_ai(
  fixed_image,
  metric,
  moving_image,
  args = NULL,
  convergence = c(10, 1e-06, 10),
  dimension = 3,
  fixed_image_mask = NULL,
  output_transform = "initialization.mat",
  principal_axes = FALSE,
  search_factor = c(20, 0.12),
  search_grid = NULL,
  transform = c("Affine", "0.1"),
  verbose = FALSE,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

fixed_image	Character; file path. Image to which the moving_image should be transformed Required.
metric	Character or numeric vector. the metric(s) to use. Required.
moving_image	Character; file path. Image that will be transformed to fixed_image Required.
args	Character. Additional parameters to the command
convergence	Character or numeric vector. convergence
dimension	Character; one of: "3", "2". dimension of output image
fixed_image_mask	Character; file path. fixed mage mask
output_transform	Character; file path. output file name
principal_axes	Logical. align using principal axes

search_factor	Character or numeric vector. search factor
search_grid	Character or numeric vector. Translation search grid in mm
transform	Character or numeric vector. Several transform options are available
verbose	Logical. enable verbosity
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

ni_ants_ants	<i>ANTS ANTS</i>
--------------	------------------

Description

ANTS wrapper for registration of images

Usage

```
ni_ants_ants(
  fixed_image,
  metric,
  metric_weight,
  moving_image,
  output_transform_prefix,
  radius,
  transformation_model,
  affine_gradient_descent_option = NULL,
  args = NULL,
  dimension = NULL,
  mi_option = NULL,
  number_of_affine_iterations = NULL,
  number_of_iterations = NULL,
  regularization = NULL,
  smoothing_sigmas = NULL,
  subsampling_factors = NULL,
  use_histogram_matching = TRUE,
  .cwd = NULL,
  .env = NULL,
```

```

    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

fixed_image	Character or numeric vector. image to which the moving image is warped Required.
metric	Character or numeric vector Required.
metric_weight	Character or numeric vector. the metric weight(s) for each stage. The weights must sum to 1 per stage. Required.
moving_image	Character or numeric vector. image to apply transformation to (generally a coregisteredfunctional) Required.
output_transform_prefix	Character Required.
radius	Character or numeric vector. radius of the region (i.e. number of layers around a voxel/pixel) that is used for computing cross correlation Required.
transformation_model	Character; one of: "Diff", "Elast", "Exp", "Greedy Exp", "SyN" Required.
affine_gradient_descent_option	Character or numeric vector
args	Character. Additional parameters to the command
dimension	Character; one of: "3", "2". image dimension (2 or 3)
mi_option	Character or numeric vector
number_of_affine_iterations	Character or numeric vector
number_of_iterations	Character or numeric vector
regularization	Character; one of: "Gauss", "DMFFD"
smoothing_sigmas	Character or numeric vector
subsampling_factors	Character or numeric vector
use_histogram_matching	Logical
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_ants_ants_introduction
ANTS antsIntroduction

Description

Uses ANTS to generate matrices to warp data from one space to another.

Usage

```
ni_ants_ants_introduction(
  input_image,
  reference_image,
  args = NULL,
  bias_field_correction = NULL,
  dimension = 3,
  force_proceed = NULL,
  inverse_warp_template_labels = NULL,
  max_iterations = NULL,
  out_prefix = "ants_",
  quality_check = NULL,
  similarity_metric = NULL,
  transformation_model = "GR",
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

input_image	Character; file path. input image to warp to template Required.
reference_image	Character; file path. template file to warp to Required.
args	Character. Additional parameters to the command
bias_field_correction	Logical. Applies bias field correction to moving image
dimension	Character; one of: "3", "2". image dimension (2 or 3)
force_proceed	Logical. force script to proceed even if headers may be incompatible

inverse_warp_template_labels	Logical. Applies inverse warp to the template labels to estimate label positions in target space (use for template-based segmentation)
max_iterations	Character or numeric vector. maximum number of iterations (must be list of integers in the form [J,K,L...]: J = coarsest resolution iterations, K = middle resolution iterations, L = fine resolution iterations)
out_prefix	Character. Prefix that is prepended to all output files (default = ants_)
quality_check	Logical. Perform a quality check of the result
similarity_metric	Character; one of: "PR", "CC", "MI", "MSQ". Type of similarity metric used for registration (CC = cross correlation, MI = mutual information, PR = probability mapping, MSQ = mean square difference)
transformation_model	Character; one of: "GR", "EL", "SY", "S2", "EX", "DD", "RI", "RA". Type of transformation model used for registration (EL = elastic transformation model, SY = SyN with time, arbitrary number of time points, S2 = SyN with time optimized for 2 time points, GR = greedy SyN, EX = exponential, DD = diffeomorphic demons style exponential mapping, RI = purely rigid, RA = affine rigid)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

ni_ants_apply_transforms

ANTS ApplyTransforms

Description

ApplyTransforms, applied to an input image, transforms it according to a

Usage

```

ni_ants_apply_transforms(
  input_image,
  reference_image,
  transforms,
  args = NULL,
  default_value = 0,
  dimension = NULL,
  float = FALSE,
  input_image_type = NULL,
  interpolation = "Linear",
  output_image = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)

```

Arguments

input_image	Character; file path. image to apply transformation to (generally a coregistered functional) Required.
reference_image	Character; file path. reference image space that you wish to warp INTO Required.
transforms	Character or numeric vector. transform files: will be applied in reverse order. For example, the last specified transform will be applied first. Required.
args	Character. Additional parameters to the command
default_value	Numeric
dimension	Character; one of: "2", "3", "4". This option forces the image to be treated as a specified-dimensional image. If not specified, antsWarp tries to infer the dimensionality from the input image.
float	Logical. Use float instead of double for computations.
input_image_type	Character; one of: "0", "1", "2", "3". Option specifying the input image type of scalar (default), vector, tensor, or time series.
interpolation	Character; one of: "Linear", "NearestNeighbor", "CosineWindowedSinc", "WelchWindowedSinc", "HammingWindowedSinc", "LanczosWindowedSinc", "MultiLabel", "Gaussian", "BSpline", "GenericLabel"
output_image	Character. output file name
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.

.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_ants_apply_transforms_to_points
ANTS ApplyTransformsToPoints

Description

ApplyTransformsToPoints, applied to an CSV file, transforms coordinates

Usage

```
ni_ants_apply_transforms_to_points(
  input_file,
  transforms,
  args = NULL,
  dimension = NULL,
  output_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

input_file	Character; file path. Currently, the only input supported is a csv file with columns including x,y (2D), x,y,z (3D) or x,y,z,t,label (4D) column headers. The points should be defined in physical space. If in doubt how to convert coordinates from your files to the space required by antsApplyTransformsToPoints try creating/drawing a simple label volume with only one voxel set to 1 and all others set to 0. Write down the voxel coordinates. Then use ImageMaths LabelStats to find out what coordinates for this voxel antsApplyTransformsToPoints is expecting. Required.
transforms	Character or numeric vector. transforms that will be applied to the points Required.
args	Character. Additional parameters to the command

dimension	Character; one of: "2", "3", "4". This option forces the image to be treated as a specified-dimensional image. If not specified, antsWarp tries to infer the dimensionality from the input image.
output_file	Character. Name of the output CSV file
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_ants_atropos	<i>ANTS Atropos</i>
-----------------	---------------------

Description

A multivariate n-class segmentation algorithm.

Usage

```
ni_ants_atropos(
  initialization,
  intensity_images,
  mask_image,
  number_of_tissue_classes,
  args = NULL,
  dimension = 3,
  icm_use_synchronous_update = NULL,
  likelihood_model = NULL,
  mrf_smoothing_factor = NULL,
  n_iterations = NULL,
  out_classified_image_name = NULL,
  posterior_formulation = NULL,
  use_random_seed = TRUE,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

initialization	Character; one of: "Random", "Otsu", "KMeans", "PriorProbabilityImages", "PriorLabelImage" Required.
intensity_images	Character or numeric vector Required.
mask_image	Character; file path Required.
number_of_tissue_classes	Integer Required.
args	Character. Additional parameters to the command
dimension	Character; one of: "3", "2", "4". image dimension (2, 3, or 4)
icm_use_synchronous_update	Logical
likelihood_model	Character
mrf_smoothing_factor	Numeric
n_iterations	Integer
out_classified_image_name	Character; file path
posterior_formulation	Character
use_random_seed	Logical. use random seed value over constant
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

```
ni_ants_average_affine_transform
    ANTS AverageAffineTransform
```

Description

Examples

Usage

```
ni_ants_average_affine_transform(
    dimension,
    output_affine_transform,
    transforms,
    args = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)
```

Arguments

dimension	Character; one of: "3", "2". image dimension (2 or 3) Required.
output_affine_transform	Character; file path. Outputname.txt: the name of the resulting transform. Required.
transforms	Character or numeric vector. transforms to average Required.
args	Character. Additional parameters to the command
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_ants_average_images
ANTS AverageImages

Description

Examples

Usage

```
ni_ants_average_images(
  dimension,
  images,
  normalize,
  args = NULL,
  output_average_image = "average.nii",
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

dimension	Character; one of: "3", "2". image dimension (2 or 3) Required.
images	Character or numeric vector. image to apply transformation to (generally a coregistered functional) Required.
normalize	Logical. Normalize: if true, the 2nd image is divided by its mean. This will select the largest image to average into. Required.
args	Character. Additional parameters to the command
output_average_image	Character; file path. the name of the resulting image.
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

 ni_ants_brain_extraction

ANTS BrainExtraction

Description

Atlas-based brain extraction.

Usage

```
ni_ants_brain_extraction(
  anatomical_image,
  brain_probability_mask,
  brain_template,
  args = NULL,
  debug = NULL,
  dimension = 3,
  extraction_registration_mask = NULL,
  image_suffix = "nii.gz",
  keep_temporary_files = NULL,
  out_prefix = "highres001_",
  use_floatingpoint_precision = NULL,
  use_random_seeding = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

`anatomical_image`

Character; file path. Structural image, typically T1. If more than one anatomical image is specified, subsequently specified images are used during the segmentation process. However, only the first image is used in the registration of priors. Our suggestion would be to specify the T1 as the first image. Anatomical template created using e.g. LPBA40 data set with `buildtemplateparallel.sh` in ANTs. **Required.**

`brain_probability_mask`

Character; file path. Brain probability mask created using e.g. LPBA40 data set which have brain masks defined, and warped to anatomical template and averaged resulting in a probability image. **Required.**

`brain_template`

Character; file path. Anatomical template created using e.g. LPBA40 data set with `buildtemplateparallel.sh` in ANTs. **Required.**

args	Character. Additional parameters to the command
debug	Logical. If > 0, runs a faster version of the script. Only for testing. Implies -u 0. Requires single thread computation for complete reproducibility.
dimension	Character; one of: "3", "2". image dimension (2 or 3)
extraction_registration_mask	Character; file path. Mask (defined in the template space) used during registration for brain extraction. To limit the metric computation to a specific region.
image_suffix	Character. any of standard ITK formats, nii.gz is default
keep_temporary_files	Integer. Keep brain extraction/segmentation warps, etc (default = 0).
out_prefix	Character. Prefix that is prepended to all output files
use_floatingpoint_precision	Character; one of: "0", "1". Use floating point precision in registrations (default = 0)
use_random_seeding	Character; one of: "0", "1". Use random number generated from system clock in Atropos (default = 1)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

```
ni_ants_buildtemplateparallel
      ANTS buildtemplateparallel
```

Description

Generate a optimal average template

Usage

```
ni_ants_buildtemplateparallel(
  in_files,
  args = NULL,
  bias_field_correction = NULL,
  dimension = 3,
```

```

gradient_step_size = NULL,
iteration_limit = 4,
max_iterations = NULL,
num_cores = NULL,
out_prefix = "antsTMPL_",
parallelization = 0,
rigid_body_registration = NULL,
similarity_metric = NULL,
transformation_model = "GR",
.cwd = NULL,
.env = NULL,
.engine = NULL,
.profile = NULL,
dry_run = FALSE,
echo = interactive()
)

```

Arguments

<code>in_files</code>	Character or numeric vector. list of images to generate template from Required .
<code>args</code>	Character. Additional parameters to the command
<code>bias_field_correction</code>	Logical. Applies bias field correction to moving image
<code>dimension</code>	Character; one of: "3", "2", "4". image dimension (2, 3 or 4)
<code>gradient_step_size</code>	Numeric. smaller magnitude results in more cautious steps (default = .25)
<code>iteration_limit</code>	Integer. iterations of template construction
<code>max_iterations</code>	Character or numeric vector. maximum number of iterations (must be list of integers in the form [J,K,L...]: J = coarsest resolution iterations, K = middle resolution iterations, L = fine resolution iterations)
<code>num_cores</code>	Integer. Requires parallelization = 2 (PEXEC). Sets number of cpu cores to use
<code>out_prefix</code>	Character. Prefix that is prepended to all output files (default = antsTMPL_)
<code>parallelization</code>	Character; one of: "0", "1", "2". control for parallel processing (0 = serial, 1 = use PBS, 2 = use PEXEC, 3 = use Apple XGrid)
<code>rigid_body_registration</code>	Logical. registers inputs before creating template (useful if no initial template available)
<code>similarity_metric</code>	Character; one of: "PR", "CC", "MI", "MSQ". Type of similarity metric used for registration (CC = cross correlation, MI = mutual information, PR = probability mapping, MSQ = mean square difference)
<code>transformation_model</code>	Character; one of: "GR", "EL", "SY", "S2", "EX", "DD". Type of transformation model used for registration (EL = elastic transformation model, SY = SyN)

	with time, arbitrary number of time points, S2 = SyN with time optimized for 2 time points, GR = greedy SyN, EX = exponential, DD = diffeomorphic demons style exponential mapping
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

```
ni_ants_compose_multi_transform
      ANTS ComposeMultiTransform
```

Description

Take a set of transformations and convert them to a single transformation matrix/warfield.

Usage

```
ni_ants_compose_multi_transform(
  transforms,
  args = NULL,
  dimension = 3,
  output_transform = NULL,
  reference_image = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

transforms	Character or numeric vector. transforms to average Required.
args	Character. Additional parameters to the command
dimension	Character; one of: "3", "2". image dimension (2 or 3)
output_transform	Character; file path. the name of the resulting transform.

reference_image	Character; file path. Reference image (only necessary when output is warpfield)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_ants_composite_transform_util
ANTS CompositeTransformUtil

Description

ANTs utility which can combine or break apart transform files into their individual

Usage

```
ni_ants_composite_transform_util(
  in_file,
  args = NULL,
  out_file = NULL,
  output_prefix = "transform",
  process = "assemble",
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character or numeric vector. Input transform file(s) Required.
args	Character. Additional parameters to the command
out_file	Character; file path. Output file path (only used for disassembly).
output_prefix	Character. A prefix that is prepended to all output files (only used for assembly).
process	Character; one of: "assemble", "disassemble". What to do with the transform inputs (assemble or disassemble)

.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_ants_convert_scalar_image_to_rgb
ANTS ConvertScalarImageToRGB

Description

Convert scalar images to RGB.

Usage

```
ni_ants_convert_scalar_image_to_rgb(
  colormap,
  dimension,
  input_image,
  maximum_input,
  minimum_input,
  args = NULL,
  custom_color_map_file = "none",
  mask_image = "none",
  maximum_RGB_output = 255,
  minimum_RGB_output = 0,
  output_image = "rgb.nii.gz",
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

colormap	Character; one of: "grey", "red", "green", "blue", "copper", "jet", "hsv", "spring", "summer", "autumn", "winter", "hot", "cool", "overunder", "custom". Select a colormap Required.
dimension	Character; one of: "3", "2". image dimension (2 or 3) Required.
input_image	Character; file path. Main input is a 3-D grayscale image. Required.
maximum_input	Integer. maximum input Required.
minimum_input	Integer. minimum input Required.
args	Character. Additional parameters to the command
custom_color_map_file	Character. custom color map file
mask_image	Character or numeric vector. mask image
maximum_RGB_output	Integer
minimum_RGB_output	Integer
output_image	Character. rgb output image
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_ants_cortical_thickness
ANTS CorticalThickness

Description

Examples

Usage

```

ni_ants_cortical_thickness(
  anatomical_image,
  brain_probability_mask,
  brain_template,
  segmentation_priors,
  t1_registration_template,
  args = NULL,
  b_spline_smoothing = NULL,
  debug = NULL,
  dimension = 3,
  extraction_registration_mask = NULL,
  image_suffix = "nii.gz",
  keep_temporary_files = NULL,
  label_propagation = NULL,
  max_iterations = NULL,
  out_prefix = "antsCT_",
  posterior_formulation = NULL,
  prior_segmentation_weight = NULL,
  quick_registration = NULL,
  segmentation_iterations = NULL,
  use_floatingpoint_precision = NULL,
  use_random_seeding = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)

```

Arguments

- `anatomical_image` Character; file path. Structural *intensity* image, typically T1. If more than one anatomical image is specified, subsequently specified images are used during the segmentation process. However, only the first image is used in the registration of priors. Our suggestion would be to specify the T1 as the first image. **Required.**
- `brain_probability_mask` Character; file path. brain probability mask in template space **Required.**
- `brain_template` Character; file path. Anatomical *intensity* template (possibly created using a population data set with `buildtemplateparallel.sh` in ANTs). This template is *not* skull-stripped. **Required.**
- `segmentation_priors` Character or numeric vector **Required.**
- `t1_registration_template` Character; file path. Anatomical *intensity* template (assumed to be skull-stripped).

	A common case would be where this would be the same template as specified in the <code>-e</code> option which is not skull stripped. Required.
<code>args</code>	Character. Additional parameters to the command
<code>b_spline_smoothing</code>	Logical. Use B-spline SyN for registrations and B-spline exponential mapping in DiReCT.
<code>debug</code>	Logical. If > 0 , runs a faster version of the script. Only for testing. Implies <code>-u 0</code> . Requires single thread computation for complete reproducibility.
<code>dimension</code>	Character; one of: "3", "2". image dimension (2 or 3)
<code>extraction_registration_mask</code>	Character; file path. Mask (defined in the template space) used during registration for brain extraction.
<code>image_suffix</code>	Character. any of standard ITK formats, <code>nii.gz</code> is default
<code>keep_temporary_files</code>	Integer. Keep brain extraction/segmentation warps, etc (default = 0).
<code>label_propagation</code>	Character. Incorporate a distance prior one the posterior formulation. Should be of the form 'label[lambda,boundaryProbability]' where label is a value of 1,2,3,... denoting label ID. The label probability for anything outside the current label = $\text{boundaryProbability} * \exp(-\text{lambda} * \text{distanceFromBoundary})$ Intuitively, smaller lambda values will increase the spatial capture range of the distance prior. To apply to all label values, simply omit specifying the label, i.e. <code>-l [lambda,boundaryProbability]</code> .
<code>max_iterations</code>	Integer. ANTS registration max iterations (default = 100x100x70x20)
<code>out_prefix</code>	Character. Prefix that is prepended to all output files
<code>posterior_formulation</code>	Character. Atropos posterior formulation and whether or not to use mixture model proportions. e.g 'Socrates[1]' (default) or 'Aristotle[1]'. Choose the latter if you want use the distance priors (see also the <code>-l</code> option for label propagation control).
<code>prior_segmentation_weight</code>	Numeric. Atropos spatial prior <i>probability</i> weight for the segmentation
<code>quick_registration</code>	Logical. If = 1, use <code>antsRegistrationSyNQuick.sh</code> as the basis for registration during brain extraction, brain segmentation, and (optional) normalization to a template. Otherwise use <code>antsRegistrationSyN.sh</code> (default = 0).
<code>segmentation_iterations</code>	Integer. N4 -> Atropos -> N4 iterations during segmentation (default = 3)
<code>use_floatingpoint_precision</code>	Character; one of: "0", "1". Use floating point precision in registrations (default = 0)
<code>use_random_seeding</code>	Character; one of: "0", "1". Use random number generated from system clock in Atropos (default = 1)

.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

```
ni_ants_create_jacobian_determinant_image
      ANTS CreateJacobianDeterminantImage
```

Description

Examples

Usage

```
ni_ants_create_jacobian_determinant_image(
  deformationField,
  imageDimension,
  outputImage,
  args = NULL,
  doLogJacobian = NULL,
  useGeometric = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

deformationField Character; file path. deformation transformation file **Required.**

imageDimension Character; one of: "3", "2". image dimension (2 or 3) **Required.**

outputImage Character; file path. output filename **Required.**

args Character. Additional parameters to the command

doLogJacobian Character; one of: "0", "1". return the log jacobian

useGeometric	Character; one of: "0", "1". return the geometric jacobian
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_ants_create_tiled_mosaic
ANTS CreateTiledMosaic

Description

The program CreateTiledMosaic in conjunction with ConvertScalarImageToRGB

Usage

```
ni_ants_create_tiled_mosaic(
  input_image,
  rgb_image,
  alpha_value = NULL,
  args = NULL,
  direction = NULL,
  flip_slice = NULL,
  mask_image = NULL,
  output_image = "output.png",
  pad_or_crop = NULL,
  permute_axes = NULL,
  slices = NULL,
  tile_geometry = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

input_image	Character; file path. Main input is a 3-D grayscale image. Required.
rgb_image	Character; file path. An optional Rgb image can be added as an overlay. It must have the same image geometry as the input grayscale image. Required.
alpha_value	Numeric. If an Rgb image is provided, render the overlay using the specified alpha parameter.
args	Character. Additional parameters to the command
direction	Integer. Specifies the direction of the slices. If no direction is specified, the direction with the coarsest spacing is chosen.
flip_slice	Character. flipXxflipY
mask_image	Character; file path. Specifies the ROI of the RGB voxels used.
output_image	Character. The output consists of the tiled mosaic image.
pad_or_crop	Character. argument passed to -p flag: [padVoxelWidth,<constantValue=0>][lowerPadding[0]xlowerPaddi user can specify whether to pad or crop a specified voxel-width boundary of each individual slice. For this program, cropping is simply padding with negative voxel-widths. If one pads (+), the user can also specify a constant pad value (default = 0). If a mask is specified, the user can use the mask to define the region, by using the keyword "mask" plus an offset, e.g. "-p mask+3".
permute_axes	Logical. doPermute
slices	Character. Number of slices to increment Slice1xSlice2xSlice3[numberOfSlicesToIncrement,<minSlice=
tile_geometry	Character. The tile geometry specifies the number of rows and columns in the output image. For example, if the user specifies "5x10", then 5 rows by 10 columns of slices are rendered. If R < 0 and C > 0 (or vice versa), the negative value is selected based on direction.
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

 ni_ants_denoise_image *ANTS DenoiseImage*

Description

Examples

Usage

```
ni_ants_denoise_image(
  input_image,
  save_noise,
  args = NULL,
  dimension = NULL,
  noise_model = "Gaussian",
  output_image = NULL,
  shrink_factor = 1,
  verbose = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

input_image	Character; file path. A scalar image is expected as input for noise correction. Required.
save_noise	Logical. True if the estimated noise should be saved to file. Required.
args	Character. Additional parameters to the command
dimension	Character; one of: "2", "3", "4". This option forces the image to be treated as a specified-dimensional image. If not specified, the program tries to infer the dimensionality from the input image.
noise_model	Character; one of: "Gaussian", "Rician". Employ a Rician or Gaussian noise model.
output_image	Character; file path. The output consists of the noise corrected version of the input image.
shrink_factor	Integer. Running noise correction on large images can be time consuming. To lessen computation time, the input image can be resampled. The shrink factor, specified as a single integer, describes this resampling. Shrink factor = 1 is the default.
verbose	Logical. Verbose output.
.cwd	Working directory override.

.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

ni_ants_gen_warp_fields
ANTS GenWarpFields

Description

Uses ANTS to generate matrices to warp data from one space to another.

Usage

```
ni_ants_gen_warp_fields(
  input_image,
  reference_image,
  args = NULL,
  bias_field_correction = NULL,
  dimension = 3,
  force_proceed = NULL,
  inverse_warp_template_labels = NULL,
  max_iterations = NULL,
  out_prefix = "ants_",
  quality_check = NULL,
  similarity_metric = NULL,
  transformation_model = "GR",
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

`input_image` Character; file path. input image to warp to template **Required.**
`reference_image` Character; file path. template file to warp to **Required.**

args	Character. Additional parameters to the command
bias_field_correction	Logical. Applies bias field correction to moving image
dimension	Character; one of: "3", "2". image dimension (2 or 3)
force_proceed	Logical. force script to proceed even if headers may be incompatible
inverse_warp_template_labels	Logical. Applies inverse warp to the template labels to estimate label positions in target space (use for template-based segmentation)
max_iterations	Character or numeric vector. maximum number of iterations (must be list of integers in the form [J,K,L...]: J = coarsest resolution iterations, K = middle resolution iterations, L = fine resolution iterations)
out_prefix	Character. Prefix that is prepended to all output files (default = ants_)
quality_check	Logical. Perform a quality check of the result
similarity_metric	Character; one of: "PR", "CC", "MI", "MSQ". Type of similarity metric used for registration (CC = cross correlation, MI = mutual information, PR = probability mapping, MSQ = mean square difference)
transformation_model	Character; one of: "GR", "EL", "SY", "S2", "EX", "DD", "RI", "RA". Type of transformation model used for registration (EL = elastic transformation model, SY = SyN with time, arbitrary number of time points, S2 = SyN with time optimized for 2 time points, GR = greedy SyN, EX = exponential, DD = diffeomorphic demons style exponential mapping, RI = purely rigid, RA = affine rigid)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

ni_ants_image_math	<i>ANTS ImageMath</i>
--------------------	-----------------------

Description

Operations over images.

Usage

```

ni_ants_image_math(
  op1,
  operation,
  args = NULL,
  dimension = 3,
  op2 = NULL,
  output_image = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)

```

Arguments

op1	Character; file path. first operator Required.
operation	Character; one of: "m", "vm", "+", "v+", "-", "v-", "/", "^", "max", "exp", "addtozero", "overadd", "abs", "total", "mean", "vtotal", "Decision", "Neg", "Project", "G", "MD", "ME", "MO", "MC", "GD", "GE", "GO", "GC", "Extract-Contours", "Translate", "4DTensorTo3DTensor", "ExtractVectorComponent", "TensorColor", "TensorFA", "TensorFADenominator", "TensorFANumerator", "TensorMeanDiffusion", "TensorRadialDiffusion", "TensorAxialDiffusion", "TensorEigenvalue", "TensorToVector", "TensorToVectorComponent", "TensorMask", "Byte", "CorruptImage", "D", "MaurerDistance", "ExtractSlice", "FillHoles", "Convolve", "Finite", "FlattenImage", "GetLargestComponent", "Grad", "RescaleImage", "WindowImage", "NeighborhoodStats", "ReplicateDisplacement", "ReplicateImage", "LabelStats", "Laplacian", "Canny", "Lipschitz", "MTR", "Normalize", "PadImage", "SigmoidImage", "Sharpen", "UnsharpMask", "PValueImage", "ReplaceVoxelValue", "SetTimeSpacing", "SetTimeSpacingWarp", "stack", "ThresholdAtMean", "TriPlanarView", "TruncateImageIntensity". mathematical operations Required.
args	Character. Additional parameters to the command
dimension	Integer. dimension of output image
op2	Character or numeric vector. second operator
output_image	Character; file path. output image file
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_ants_joint_fusion *ANTs JointFusion*

Description

An image fusion algorithm.

Usage

```
ni_ants_joint_fusion(
  atlas_image,
  atlas_segmentation_image,
  target_image,
  alpha = 0.1,
  args = NULL,
  beta = 2,
  constrain_nonnegative = FALSE,
  dimension = NULL,
  exclusion_image_label = NULL,
  mask_image = NULL,
  out_label_fusion = NULL,
  patch_metric = NULL,
  patch_radius = NULL,
  retain_atlas_voting_images = FALSE,
  retain_label_posterior_images = FALSE,
  search_radius = c(3, 3, 3),
  verbose = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

atlas_image Character or numeric vector. The atlas image (or multimodal atlas images) assumed to be aligned to a common image domain. **Required.**

atlas_segmentation_image
Character or numeric vector. The atlas segmentation images. For performing label fusion the number of specified segmentations should be identical to the number of atlas image sets. **Required.**

target_image	Character or numeric vector. The target image (or multimodal target images) assumed to be aligned to a common image domain. Required.
alpha	Numeric. Regularization term added to matrix Mx for calculating the inverse. Default = 0.1
args	Character. Additional parameters to the command
beta	Numeric. Exponent for mapping intensity difference to the joint error. Default = 2.0
constrain_nonnegative	Logical. Constrain solution to non-negative weights.
dimension	Character; one of: "3", "2", "4". This option forces the image to be treated as a specified-dimensional image. If not specified, the program tries to infer the dimensionality from the input image.
exclusion_image_label	Character or numeric vector. Specify a label for the exclusion region.
mask_image	Character; file path. If a mask image is specified, fusion is only performed in the mask region.
out_label_fusion	Character; file path. The output label fusion image.
patch_metric	Character; one of: "PC", "MSQ". Metric to be used in determining the most similar neighborhood patch. Options include Pearson's correlation (PC) and mean squares (MSQ). Default = PC (Pearson correlation).
patch_radius	Character or numeric vector. Patch radius for similarity measures. Default: 2x2x2
retain_atlas_voting_images	Logical. Retain atlas voting images. Default = false
retain_label_posterior_images	Logical. Retain label posterior probability images. Requires atlas segmentations to be specified. Default = false
search_radius	Character or numeric vector. Search radius for similarity measures. Default = 3x3x3. One can also specify an image where the value at the voxel specifies the isotropic search radius at that voxel.
verbose	Logical. Verbose output.
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

```
ni_ants_kelly_kapowski
    ANTS KellyKapowski
```

Description

Nipype Interface to ANTs' KellyKapowski, also known as DiReCT.

Usage

```
ni_ants_kelly_kapowski(
    segmentation_image,
    args = NULL,
    convergence = "[50,0.001,10]",
    cortical_thickness = NULL,
    dimension = 3,
    gradient_step = 0.025,
    gray_matter_prob_image = NULL,
    max_invert_displacement_field_iters = 20,
    number_integration_points = 10,
    smoothing_variance = 1,
    smoothing_velocity_field = 1.5,
    thickness_prior_estimate = 10,
    thickness_prior_image = NULL,
    use_bspline_smoothing = NULL,
    white_matter_prob_image = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)
```

Arguments

segmentation_image	Character; file path. A segmentation image must be supplied labeling the gray and white matters. Default values = 2 and 3, respectively. Required.
args	Character. Additional parameters to the command
convergence	Character. Convergence is determined by fitting a line to the normalized energy profile of the last N iterations (where N is specified by the window size) and determining the slope which is then compared with the convergence threshold.
cortical_thickness	Character; file path. Filename for the cortical thickness.
dimension	Character; one of: "3", "2". image dimension (2 or 3)

gradient_step	Numeric. Gradient step size for the optimization.
gray_matter_prob_image	Character; file path. In addition to the segmentation image, a gray matter probability image can be used. If no such image is supplied, one is created using the segmentation image and a variance of 1.0 mm.
max_invert_displacement_field_iters	Integer. Maximum number of iterations for estimating the invertdisplacement field.
number_integration_points	Integer. Number of compositions of the diffeomorphism per iteration.
smoothing_variance	Numeric. Defines the Gaussian smoothing of the hit and total images.
smoothing_velocity_field	Numeric. Defines the Gaussian smoothing of the velocity field (default = 1.5). If the b-spline smoothing option is chosen, then this defines the isotropic mesh spacing for the smoothing spline (default = 15).
thickness_prior_estimate	Numeric. Provides a prior constraint on the final thickness measurement in mm.
thickness_prior_image	Character; file path. An image containing spatially varying prior thickness values.
use_bspline_smoothing	Logical. Sets the option for B-spline smoothing of the velocity field.
white_matter_prob_image	Character; file path. In addition to the segmentation image, a white matter probability image can be used. If no such image is supplied, one is created using the segmentation image and a variance of 1.0 mm.
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

ni_ants_label_geometry
ANTS LabelGeometry

Description

Extracts geometry measures using a label file and an optional image file

Usage

```
ni_ants_label_geometry(  
  intensity_image,  
  label_image,  
  args = NULL,  
  dimension = 3,  
  output_file = NULL,  
  .cwd = NULL,  
  .env = NULL,  
  .engine = NULL,  
  .profile = NULL,  
  dry_run = FALSE,  
  echo = interactive()  
)
```

Arguments

intensity_image	Character; file path. Intensity image to extract values from. This is an optional input Required .
label_image	Character; file path. label image to use for extracting geometry measures Required .
args	Character. Additional parameters to the command
dimension	Character; one of: "3", "2". image dimension (2 or 3)
output_file	Character. name of output file
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

```
ni_ants_laplacian_thickness
    ANTS LaplacianThickness
```

Description

Calculates the cortical thickness from an anatomical image

Usage

```
ni_ants_laplacian_thickness(
    input_gm,
    input_wm,
    args = NULL,
    dT = NULL,
    output_image = NULL,
    prior_thickness = NULL,
    smooth_param = NULL,
    sulcus_prior = NULL,
    tolerance = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)
```

Arguments

input_gm	Character; file path. gray matter segmentation image Required.
input_wm	Character; file path. white matter segmentation image Required.
args	Character. Additional parameters to the command
dT	Numeric. Time delta used during integration (defaults to 0.01)
output_image	Character. name of output file
prior_thickness	Numeric. Prior thickness (defaults to 500)
smooth_param	Numeric. Sigma of the Laplacian Recursive Image Filter (defaults to 1)
sulcus_prior	Numeric. Positive floating point number for sulcus prior. Authors said that 0.15 might be a reasonable value
tolerance	Numeric. Tolerance to reach during optimization (defaults to 0.001)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.

.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_ants_measure_image_similarity
ANTS MeasureImageSimilarity

Description

Examples

Usage

```
ni_ants_measure_image_similarity(
    fixed_image,
    metric,
    moving_image,
    radius_or_number_of_bins,
    sampling_percentage,
    args = NULL,
    dimension = NULL,
    fixed_image_mask = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)
```

Arguments

fixed_image	Character; file path. Image to which the moving image is warped Required.
metric	Character; one of: "CC", "MI", "Mattes", "MeanSquares", "Demons", "GC" Required.
moving_image	Character; file path. Image to apply transformation to (generally a coregistered functional) Required.
radius_or_number_of_bins	Integer. The number of bins in each stage for the MI and Mattes metric, or the radius for other metrics Required.

sampling_percentage	Character. Percentage of points accessible to the sampling strategy over which to optimize the metric. Required.
args	Character. Additional parameters to the command
dimension	Character; one of: "2", "3", "4". Dimensionality of the fixed/moving image pair
fixed_image_mask	Character; file path. mask used to limit metric sampling region of the fixed image
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_ants_multiply_images
ANTS MultiplyImages

Description

Examples

Usage

```
ni_ants_multiply_images(
  dimension,
  first_input,
  output_product_image,
  second_input,
  args = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

dimension	Character; one of: "3", "2". image dimension (2 or 3) Required.
first_input	Character; file path. image 1 Required.
output_product_image	Character; file path. Outputfname.nii.gz: the name of the resulting image. Required.
second_input	Character or numeric vector. image 2 or multiplication weight Required.
args	Character. Additional parameters to the command
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_ants_n4_bias_field_correction
ANTS N4BiasFieldCorrection

Description

Bias field correction.

Usage

```
ni_ants_n4_bias_field_correction(
  copy_header,
  input_image,
  save_bias,
  args = NULL,
  bspline_fitting_distance = NULL,
  dimension = 3,
  histogram_sharpening = NULL,
  mask_image = NULL,
  n_iterations = NULL,
  output_image = NULL,
  rescale_intensities = FALSE,
  shrink_factor = NULL,
  weight_image = NULL,
  .cwd = NULL,
```

```

    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

copy_header	Logical. copy headers of the original image into the output (corrected) file Required.
input_image	Character; file path. input for bias correction. Negative values or values close to zero should be processed prior to correction Required.
save_bias	Logical. True if the estimated bias should be saved to file. Required.
args	Character. Additional parameters to the command
bspline_fitting_distance	Numeric
dimension	Character; one of: "3", "2", "4". image dimension (2, 3 or 4)
histogram_sharpening	Character or numeric vector. Three-values tuple of histogram sharpening parameters (FWHM, wienerNose, numberOfHistogramBins). These options describe the histogram sharpening parameters, i.e. the deconvolution step parameters described in the original N3 algorithm. The default values have been shown to work fairly well.
mask_image	Character; file path. image to specify region to perform final bias correction in
n_iterations	Character or numeric vector
output_image	Character. output file name
rescale_intensities	Logical. [NOTE: Only ANTs>=2.1.0] At each iteration, a new intensity mapping is calculated and applied but there is nothing which constrains the new intensity range to be within certain values. The result is that the range can "drift" from the original at each iteration. This option rescales to the [min,max] range of the original image intensities within the user-specified mask.
shrink_factor	Integer
weight_image	Character; file path. image for relative weighting (e.g. probability map of the white matter) of voxels during the B-spline fitting.
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_ants_registration *ANTS Registration*

Description

ANTs Registration command for registration of images

Usage

```
ni_ants_registration(  
    fixed_image,  
    metric,  
    metric_weight,  
    moving_image,  
    shrink_factors,  
    smoothing_sigmas,  
    transforms,  
    args = NULL,  
    collapse_output_transforms = TRUE,  
    dimension = 3,  
    fixed_image_mask = NULL,  
    float = NULL,  
    initial_moving_transform = NULL,  
    initial_moving_transform_com = NULL,  
    initialize_transforms_per_stage = FALSE,  
    interpolation = "Linear",  
    output_transform_prefix = "transform",  
    random_seed = NULL,  
    restore_state = NULL,  
    save_state = NULL,  
    verbose = FALSE,  
    winsorize_lower_quantile = 0,  
    winsorize_upper_quantile = 1,  
    write_composite_transform = FALSE,  
    .cwd = NULL,  
    .env = NULL,  
    .engine = NULL,  
    .profile = NULL,  
    dry_run = FALSE,  
    echo = interactive()  
)
```

Arguments

`fixed_image` Character or numeric vector. Image to which the `moving_image` should be transformed(usually a structural image) **Required.**

metric	Character or numeric vector. the metric(s) to use for each stage. Note that multiple metrics per stage are not supported in ANTS 1.9.1 and earlier. Required.
metric_weight	Character or numeric vector. the metric weight(s) for each stage. The weights must sum to 1 per stage. Required.
moving_image	Character or numeric vector. Image that will be registered to the space of fixed_image. This is the image on which the transformations will be applied to Required.
shrink_factors	Character or numeric vector Required.
smoothing_sigmas	Character or numeric vector Required.
transforms	Character or numeric vector Required.
args	Character. Additional parameters to the command
collapse_output_transforms	Logical. Collapse output transforms. Specifically, enabling this option combines all adjacent linear transforms and composes all adjacent displacement field transforms before writing the results to disk.
dimension	Character; one of: "3", "2". image dimension (2 or 3)
fixed_image_mask	Character; file path. Mask used to limit metric sampling region of the fixed image in all stages
float	Logical. Use float instead of double for computations.
initial_moving_transform	Character or numeric vector. A transform or a list of transforms that should be applied before the registration begins. Note that, when a list is given, the transformations are applied in reverse order.
initial_moving_transform_com	Character; one of: "0", "1", "2". Align the moving_image and fixed_image before registration using the geometric center of the images (=0), the image intensities (=1), or the origin of the images (=2).
initialize_transforms_per_stage	Logical. Initialize linear transforms from the previous stage. By enabling this option, the current linear stage transform is directly initialized from the previous stages linear transform; this allows multiple linear stages to be run where each stage directly updates the estimated linear transform from the previous stage. (e.g. Translation -> Rigid -> Affine).
interpolation	Character; one of: "Linear", "NearestNeighbor", "CosineWindowedSinc", "WelchWindowedSinc", "HammingWindowedSinc", "LanczosWindowedSinc", "BSpline", "MultiLabel", "Gaussian", "GenericLabel"
output_transform_prefix	Character
random_seed	Integer. Fixed seed for random number generation
restore_state	Character; file path. Filename for restoring the internal restorable state of the registration

save_state	Character; file path. Filename for saving the internal restorable state of the registration
verbose	Logical
winsorize_lower_quantile	Character. The Lower quantile to clip image ranges
winsorize_upper_quantile	Character. The Upper quantile to clip image ranges
write_composite_transform	Logical
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_ants_registration_syn_quick
ANTS RegistrationSynQuick

Description

Registration using a symmetric image normalization method (SyN).

Usage

```
ni_ants_registration_syn_quick(
  fixed_image,
  moving_image,
  args = NULL,
  dimension = 3,
  histogram_bins = 32,
  num_threads = 1,
  output_prefix = "transform",
  precision_type = "double",
  random_seed = NULL,
  spline_distance = 26,
  transform_type = "s",
  use_histogram_matching = NULL,
  .cwd = NULL,
```

```

    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

fixed_image	Character or numeric vector. Fixed image or source image or reference image Required.
moving_image	Character or numeric vector. Moving image or target image Required.
args	Character. Additional parameters to the command
dimension	Character; one of: "3", "2". image dimension (2 or 3)
histogram_bins	Integer. histogram bins for mutual information in SyN stage (default = 32)
num_threads	Integer. Number of threads (default = 1)
output_prefix	Character. A prefix that is prepended to all output files
precision_type	Character; one of: "double", "float". precision type (default = double)
random_seed	Integer. fixed random seed
spline_distance	Integer. spline distance for deformable B-spline SyN transform (default = 26)
transform_type	Character; one of: "s", "t", "r", "a", "sr", "b", "br". Transform type * t: translation * r: rigid * a: rigid + affine * s: rigid + affine + deformable syn (default) * sr: rigid + deformable syn * b: rigid + affine + deformable b-spline syn * br: rigid + deformable b-spline syn
use_histogram_matching	Logical. use histogram matching
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

```
ni_ants_resample_image_by_spacing
    ANTS ResampleImageBySpacing
```

Description

Resample an image with a given spacing.

Usage

```
ni_ants_resample_image_by_spacing(
    input_image,
    out_spacing,
    addvox = NULL,
    apply_smoothing = NULL,
    args = NULL,
    dimension = 3,
    nn_interp = NULL,
    output_image = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)
```

Arguments

input_image	Character; file path. input image file Required.
out_spacing	Character or numeric vector. output spacing Required.
addvox	Integer. addvox pads each dimension by addvox
apply_smoothing	Logical. smooth before resampling
args	Character. Additional parameters to the command
dimension	Integer. dimension of output image
nn_interp	Logical. nn interpolation
output_image	Character; file path. output image file
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_ants_threshold_image

ANTS ThresholdImage

Description

Apply thresholds on images.

Usage

```
ni_ants_threshold_image(
  copy_header,
  input_image,
  args = NULL,
  dimension = 3,
  input_mask = NULL,
  inside_value = NULL,
  mode = NULL,
  num_thresholds = NULL,
  output_image = NULL,
  outside_value = NULL,
  th_high = NULL,
  th_low = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

copy_header	Logical. copy headers of the original image into the output (corrected) file Required.
input_image	Character; file path. input image file Required.
args	Character. Additional parameters to the command
dimension	Integer. dimension of output image
input_mask	Character; file path. input mask for Otsu, Kmeans
inside_value	Numeric. inside value
mode	Character; one of: "Otsu", "Kmeans". whether to run Otsu / Kmeans thresholding

num_thresholds	Integer. number of thresholds
output_image	Character; file path. output image file
outside_value	Numeric. outside value
th_high	Numeric. upper threshold
th_low	Numeric. lower threshold
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

ni_ants_warp_image_multi_transform
ANTS WarpImageMultiTransform

Description

Warpes an image from one space to another

Usage

```
ni_ants_warp_image_multi_transform(
  input_image,
  transformation_series,
  args = NULL,
  dimension = 3,
  output_image = NULL,
  reference_image = NULL,
  reslice_by_header = NULL,
  tightest_box = NULL,
  use_bspline = NULL,
  use_nearest = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

input_image	Character; file path. image to apply transformation to (generally a coregistered functional) Required.
transformation_series	Character or numeric vector. transformation file(s) to be applied Required.
args	Character. Additional parameters to the command
dimension	Character; one of: "3", "2". image dimension (2 or 3)
output_image	Character; file path. name of the output warped image
reference_image	Character; file path. reference image space that you wish to warp INTO
reslice_by_header	Logical. Uses orientation matrix and origin encoded in reference image file header. Not typically used with additional transforms
tightest_box	Logical. computes tightest bounding box (overridden by reference_image if given)
use_bspline	Logical. Use 3rd order B-Spline interpolation
use_nearest	Logical. Use nearest neighbor interpolation
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

ni_ants_warp_time_series_image_multi_transform
ANTS WarpTimeSeriesImageMultiTransform

Description

Warp a time-series from one space to another

Usage

```

ni_ants_warp_time_series_image_multi_transform(
  input_image,
  transformation_series,
  args = NULL,
  dimension = 4,
  out_postfix = "_wtsimt",
  reference_image = NULL,
  reslice_by_header = NULL,
  tightest_box = NULL,
  use_bspline = NULL,
  use_nearest = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)

```

Arguments

input_image	Character; file path. image to apply transformation to (generally a coregistered functional) Required.
transformation_series	Character or numeric vector. transformation file(s) to be applied Required.
args	Character. Additional parameters to the command
dimension	Character; one of: "4", "3". image dimension (3 or 4)
out_postfix	Character. Postfix that is prepended to all output files (default = _wtsimt)
reference_image	Character; file path. reference image space that you wish to warp INTO
reslice_by_header	Logical. Uses orientation matrix and origin encoded in reference image file header. Not typically used with additional transforms
tightest_box	Logical. computes tightest bounding box (overridden by reference_image if given)
use_bspline	Logical. Use 3rd order B-Spline interpolation
use_nearest	Logical. Use nearest neighbor interpolation
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

<code>ni_bids_app</code>	<i>Create a bidsappr-based BIDS App pre-configured with niflowr</i>
--------------------------	---

Description

Scaffolds a BIDS App using `bidsappr::new_app()` and creates a template `app.R` with `niflowr` integration. The generated app includes example usage of `niflowr` functions and can be customized with tool-specific examples.

Usage

```
ni_bids_app(path, app_name = basename(path), tools = NULL)
```

Arguments

<code>path</code>	Directory path where the app will be created.
<code>app_name</code>	Name of the BIDS App. Default: <code>basename</code> of <code>path</code> .
<code>tools</code>	Optional character vector of spec IDs (e.g., <code>c("fsl.bet", "ants.registration")</code>) to include as commented-out examples in the generated app.

Value

Invisibly returns the path to the created app directory.

Examples

```
## Not run:
# Create a basic BIDS App
ni_bids_app("~/my_bids_app", app_name = "MyApp")

# Create an app with tool examples
ni_bids_app("~/my_bids_app", tools = c("fsl.bet", "ants.registration"))

## End(Not run)
```

ni_bids_inputs	<i>Query a BIDS project for inputs matching a spec</i>
----------------	--

Description

Uses `bidser::bids_project` to discover files matching a spec's input requirements.

Usage

```
ni_bids_inputs(
  project,
  spec_id,
  subid = NULL,
  task = NULL,
  session = NULL,
  run = NULL,
  modality = NULL
)
```

Arguments

project	A <code>bids_project</code> object (from <code>bidser::bids_project()</code>).
spec_id	Spec ID or <code>ni_spec</code> object.
subid	Subject ID filter.
task	Task filter.
session	Session filter.
run	Run filter.
modality	Image modality to search for (e.g. "T1w", "bold").

Value

A `data.frame` with a `path` column containing matched file paths.

ni_call	<i>Construct an ni_call: bind parameter values to a spec</i>
---------	--

Description

Creates a validated call object that captures a spec plus concrete parameter values, ready for execution or dry-run preview.

Usage

```
ni_call(
  spec_id,
  ...,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  .validate = TRUE
)
```

Arguments

spec_id	A spec ID (e.g. "fsl.bet"), path to a JSON file, or an ni_spec object.
...	Named parameter values.
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override. One of "auto", "native", "docker", "apptainer".
.profile	Runtime profile override (maps to profiles entry in niflowr.yml).
.validate	Logical; validate inputs at construction time. Default TRUE.

Value

An ni_call object (S3 list).

ni_cmd	<i>Get the command + args that would be executed</i>
--------	--

Description

Get the command + args that would be executed

Usage

```
ni_cmd(call)
```

Arguments

call	An ni_call object.
------	--------------------

Value

A list with command, args, wd, env, stdout, stderr.

ni_config	<i>Get or update niflowr runtime configuration</i>
-----------	--

Description

By default, `ni_config()` returns the effective configuration computed as:

Usage

```
ni_config(
  derivatives_dir = NULL,
  config = NULL,
  config_file = NULL,
  auto_read = NULL,
  .reset = FALSE
)
```

Arguments

<code>derivatives_dir</code>	Base directory used for inferred BIDS-derivatives outputs. Default: "derivatives/niflowr".
<code>config</code>	Optional named list of configuration overrides to merge into in-memory overrides.
<code>config_file</code>	Path to project config file. Default: "niflowr.yml".
<code>auto_read</code>	Logical; if TRUE, read <code>config_file</code> automatically when resolving config.
<code>.reset</code>	Logical; if TRUE, reset in-memory overrides and behavior.

Details

```
defaults <- niflowr defaults
file <- niflowr.yml (if present and auto_read = TRUE)
effective <- merge(defaults, file, in_memory_overrides)
```

Value

Named list with current effective configuration.

ni_constraints	<i>List spec constraints as a table</i>
----------------	---

Description

List spec constraints as a table

Usage

```
ni_constraints(spec_id)
```

Arguments

spec_id Spec ID, spec path, or ni_spec object.

Value

A data frame (or tibble, if available) with one row per constraint.

ni_dcm2niix_convert	<i>dcm2niix Convert</i>
---------------------	-------------------------

Description

Convert DICOM folders to NIFTI (and optional BIDS sidecars) using dcm2niix.

Usage

```
ni_dcm2niix_convert(
  in_folder,
  args = NULL,
  compression_level = NULL,
  adjacent = NULL,
  bids_sidecar = NULL,
  anon_bids = NULL,
  bids_subject = NULL,
  bids_session = NULL,
  comment = NULL,
  depth = NULL,
  export_format = NULL,
  filename = NULL,
  ignore_derived = NULL,
  lossless_scale = NULL,
  merge_2d_slices = NULL,
  series_crc = NULL,
  out_dir = NULL,
```

```

philips_float = NULL,
search_only = NULL,
rename = NULL,
single_file = NULL,
verbose = NULL,
write_conflicts = NULL,
crop = NULL,
compression = NULL,
big_endian = NULL,
progress = NULL,
ignore_trigger_times = NULL,
terse = NULL,
.cwd = NULL,
.env = NULL,
.engine = NULL,
.profile = NULL,
dry_run = FALSE,
echo = interactive()
)

```

Arguments

in_folder	Character; directory path. Folder with DICOM images to convert. Required.
args	Character. Additional parameters to the command
compression_level	Integer. Gzip compression level (1=fastest..9=smallest).
adjacent	Character; one of: "n", "y". Assume adjacent DICOMs from a series are stored in the same folder (n/y).
bids_sidecar	Character; one of: "y", "n", "o". BIDS sidecar mode (y/n/o; o=JSON only, no NIfTI).
anon_bids	Character; one of: "y", "n". Anonymize BIDS sidecars (y/n).
bids_subject	Character. Set BIDS subject label (equivalent to -bi).
bids_session	Character. Set BIDS session label (equivalent to -bv).
comment	Character. Comment stored in NIfTI aux_file (up to 24 characters).
depth	Integer. Directory search depth for recursive discovery (0..9).
export_format	Character; one of: "n", "y", "o", "j", "b". Export format (n=NIfTI, y=NRRD, o=MGH, j=JNIfTI, b=BJNIfTI).
filename	Character. Output filename template (e.g. %p_%t_%s).
ignore_derived	Character; one of: "y", "n". Ignore derived/localizer/2D images (y/n).
lossless_scale	Character; one of: "y", "n", "o". Lossless scaling for 16-bit integers (y/n/o).
merge_2d_slices	Character; one of: "0", "1", "2", "n", "y". Merge slices from same series despite differing properties (0/1/2 or n/y).
series_crc	Character or numeric vector. Only convert specific series CRC numbers; repeat-able.

out_dir	Character; directory path. Output directory (omit to write into input directory).
philips_float	Character; one of: "y", "n". Use Philips precise floating-point scaling (y/n).
search_only	Character; one of: "y", "l", "n". Directory search mode (y=count, l=list, n=off).
rename	Character; one of: "y", "n". Rename DICOMs instead of converting (y/n).
single_file	Character; one of: "y", "n". Single-file mode: convert only one file in folder (y/n).
verbose	Character; one of: "0", "1", "2", "n", "y". Verbosity (0/1/2 or n/y).
write_conflicts	Character; one of: "0", "1", "2". Name conflict behavior (0=skip, 1=overwrite, 2=add suffix).
crop	Character; one of: "y", "n", "i". Crop 3D acquisitions (y/n/i where i=ignore crop and rotate).
compression	Character; one of: "y", "o", "i", "n", "3". Compression mode (y/o/i/n/3).
big_endian	Character; one of: "y", "n", "o". Output byte order (y=big-endian, n=little-endian, o=native/optimal).
progress	Character; one of: "y", "n". Progress reporting (y/n).
ignore_trigger_times	Logical. Ignore trigger time DICOM fields (0018,1060 and 0020,9153).
terse	Logical. Omit filename postfixes (can increase overwrite risk).
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_deriv_path	<i>Generate a BIDS-derivatives output path</i>
---------------	--

Description

Constructs an output path following BIDS-derivatives conventions.

Usage

```
ni_deriv_path(
  input_path,
  derivatives_dir = "derivatives/niflowr",
  desc = NULL,
  space = NULL,
  suffix = NULL,
  extension = ".nii.gz"
)
```

Arguments

input_path	The input file path (a BIDS-formatted path).
derivatives_dir	Base derivatives directory. Default: "derivatives/niflowr".
desc	Description entity (e.g. "brain", "preproc").
space	Space entity (e.g. "MNI152NLin2009cAsym").
suffix	Override the BIDS suffix (e.g. "T1w", "bold").
extension	File extension. Default: ".nii.gz".

Value

A character string with the derivatives output path.

ni_doctor	<i>Run runtime diagnostics</i>
-----------	--------------------------------

Description

Performs quick checks for runtime binaries, mount roots, profile shape, and lockfile availability/consistency.

Usage

```
ni_doctor(cfg = NULL, strict = FALSE, check_lock = TRUE)
```

Arguments

cfg	Optional resolved config list. Defaults to current effective config.
strict	Logical; if TRUE, abort on any failed checks.
check_lock	Logical; include lockfile checks.

Value

Data frame with check, status, and message columns.

ni_dry_run	<i>Run a spec as a dry run</i>
------------	--------------------------------

Description

Shorthand for `ni_run(ni_call(spec, ...), dry_run = TRUE)`.

Usage

```
ni_dry_run(spec_id, ...)
```

Arguments

spec_id	A spec ID (e.g. "fsl.bet"), path to a JSON file, or an <code>ni_spec</code> object.
...	Named parameter values.

ni_fastsurfer_run	<i>FastSurfer Run</i>
-------------------	-----------------------

Description

FastSurfer: GPU-accelerated brain segmentation and cortical surface reconstruction

Usage

```
ni_fastsurfer_run(
  t1,
  sid,
  sd,
  fs_license = NULL,
  t2 = NULL,
  seg_only = NULL,
  surf_only = NULL,
  device = NULL,
  viewagg_device = NULL,
  threads = NULL,
  vox_size = NULL,
  three_t = NULL,
  no_asegdkf = NULL,
  no_cereb = NULL,
  no_hypothal = NULL,
  no_cc = NULL,
  no_biasfield = NULL,
  no_surfreq = NULL,
  no_fs_t1 = NULL,
```

```

    py = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

t1	Character; file path. Input T1-weighted image Required.
sid	Character. Subject ID Required.
sd	Character; directory path. Subjects directory (output root) Required.
fs_license	Character; file path. FreeSurfer license file
t2	Character; file path. Optional T2-weighted image for hypothalamus segmentation
seg_only	Logical. Run segmentation only (skip surface reconstruction, ~5 min)
surf_only	Logical. Run surface reconstruction only (requires prior segmentation)
device	Character; one of: "auto", "cpu", "cuda", "mps". Device for DNN inference
viewagg_device	Character; one of: "cpu", "cuda". Device for view aggregation (use cpu to reduce VRAM usage)
threads	Integer. Number of parallel threads
vox_size	Character. Voxel size for processing (0.7-1.0 or 'min')
three_t	Logical. Use the 3T atlas for talairach registration
no_asegdk	Logical. Skip whole-brain DKT segmentation
no_cereb	Logical. Skip cerebellum segmentation
no_hypothal	Logical. Skip hypothalamus segmentation
no_cc	Logical. Skip corpus callosum segmentation
no_biasfield	Logical. Skip bias field correction
no_surfreg	Logical. Skip surface registration (spherical registration)
no_fs_t1	Logical. Skip T1.mgz generation from conformed image
py	Character. Python command to use (override default)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

 ni_fastsurfer_segment *FastSurfer Segment*

Description

FastSurfer segmentation-only mode: fast brain segmentation (~5 min) without surface reconstruction

Usage

```
ni_fastsurfer_segment(
  t1,
  sid,
  sd,
  device = NULL,
  threads = NULL,
  vox_size = NULL,
  no_asegdkit = NULL,
  no_cereb = NULL,
  no_hypothal = NULL,
  no_cc = NULL,
  no_biasfield = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

t1	Character; file path. Input T1-weighted image Required.
sid	Character. Subject ID Required.
sd	Character; directory path. Subjects directory (output root) Required.
device	Character; one of: "auto", "cpu", "cuda", "mps". Device for DNN inference
threads	Integer. Number of parallel threads
vox_size	Character. Voxel size for processing (0.7-1.0 or 'min')
no_asegdkit	Logical. Skip whole-brain DKT segmentation
no_cereb	Logical. Skip cerebellum segmentation
no_hypothal	Logical. Skip hypothalamus segmentation
no_cc	Logical. Skip corpus callosum segmentation
no_biasfield	Logical. Skip bias field correction
.cwd	Working directory override.

.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

`ni_fmriprep_confounds` *Read fMRIPrep confounds timeseries*

Description

Reads an fMRIPrep confounds TSV file for a given subject, task, and optional session and run.

Usage

```
ni_fmriprep_confounds(
  deriv_dir,
  subid,
  task,
  session = NULL,
  run = NULL,
  select = NULL
)
```

Arguments

<code>deriv_dir</code>	Path to the fMRIPrep derivatives directory.
<code>subid</code>	Subject ID (without "sub-" prefix).
<code>task</code>	Task name (without "task-" prefix).
<code>session</code>	Session ID (without "ses-" prefix). Default: NULL.
<code>run</code>	Run number (without "run-" prefix). Default: NULL.
<code>select</code>	Character vector of column names to select. Default: NULL (all columns).

Details

fMRIPrep confounds files follow the pattern: `{deriv_dir}/sub-{subid}/[ses-{session}]/func/sub-{subid}_{ses-}`

Value

A `data.frame` with confound timeseries.

Examples

```
## Not run:
conf <- ni_fmriprep_confounds(
  deriv_dir = "derivatives/fmriprep",
  subid = "01",
  task = "rest"
)

## End(Not run)
```

ni_fmriprep_derivatives

List fMRIPrep derivatives overview

Description

Summarizes what's available in an fMRIPrep derivatives directory by subject.

Usage

```
ni_fmriprep_derivatives(deriv_dir, subid = NULL)
```

Arguments

deriv_dir Path to the fMRIPrep derivatives directory.
subid Subject ID filter (without "sub-" prefix). Default: NULL (all subjects).

Value

A data.frame with columns: subject, has_anat, has_func, n_anat_files, n_func_files.

Examples

```
## Not run:
overview <- ni_fmriprep_derivatives("derivatives/fmriprep")
print(overview)

## End(Not run)
```

ni_fmriprep_preproc *Locate fMRIPrep preprocessed files*

Description

Searches for preprocessed files in an fMRIPrep derivatives directory and returns a data.frame with file paths and parsed BIDS entities.

Usage

```
ni_fmriprep_preproc(
  deriv_dir,
  subid = NULL,
  task = NULL,
  session = NULL,
  run = NULL,
  space = NULL,
  suffix = NULL,
  extension = ".nii.gz"
)
```

Arguments

deriv_dir	Path to the fMRIPrep derivatives directory.
subid	Subject ID (without "sub-" prefix). Default: NULL (all subjects).
task	Task name filter (without "task-" prefix). Default: NULL.
session	Session ID filter (without "ses-" prefix). Default: NULL.
run	Run number filter (without "run-" prefix). Default: NULL.
space	Space filter (without "space-" prefix, e.g. "MNI152NLin2009cAsym"). Default: NULL.
suffix	Suffix filter (e.g. "bold", "T1w", "mask"). Default: NULL.
extension	File extension filter. Default: ".nii.gz".

Details

fMRIPrep outputs follow the pattern: {deriv_dir}/sub-{subid}/[ses-{session}]/[anat|func]/sub-{subid}_[enti

Value

A data.frame with columns: path, subject, session, task, run, space, desc, suffix, datatype.

Examples

```
## Not run:
# Find all preprocessed bold files
bold <- ni_fmriprep_preproc(
  deriv_dir = "derivatives/fmriprep",
  suffix = "bold"
)

# Find specific subject/task in MNI space
preproc <- ni_fmriprep_preproc(
  deriv_dir = "derivatives/fmriprep",
  subid = "01",
  task = "rest",
  space = "MNI152NLin2009cAsym"
)

## End(Not run)
```

ni_freesurfer_add_x_form_to_header

FREESURFER AddXFormToHeader

Description

Just adds specified xform to the volume header.

Usage

```
ni_freesurfer_add_x_form_to_header(
  in_file,
  transform,
  args = NULL,
  copy_name = NULL,
  out_file = "output.mgz",
  verbose = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. input volume Required.
transform	Character; file path. xfm file Required.

args	Character. Additional parameters to the command
copy_name	Logical. do not try to load the xfmfile, just copy name
out_file	Character; file path. output volume
verbose	Logical. be verbose
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_aparc2_aseg
FREESURFER Aparc2Aseg

Description

Maps the cortical labels from the automatic cortical parcellation

Usage

```
ni_freesurfer_aparc2_aseg(
  lh_annotation,
  lh_pial,
  lh_ribbon,
  lh_white,
  out_file,
  rh_annotation,
  rh_pial,
  rh_ribbon,
  rh_white,
  ribbon,
  subject_id,
  a2009s = NULL,
  args = NULL,
  aseg = NULL,
  ctxseg = NULL,
  hypo_wm = NULL,
  label_wm = NULL,
  rip_unknown = NULL,
```

```

    volmask = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

lh_annotation	Character; file path. Input file must be <subject_id>/label/lh.aparc.annot Required.
lh_pial	Character; file path. Input file must be <subject_id>/surf/lh.pial Required.
lh_ribbon	Character; file path. Input file must be <subject_id>/mri/lh.ribbon.mgz Required.
lh_white	Character; file path. Input file must be <subject_id>/surf/lh.white Required.
out_file	Character; file path. Full path of file to save the output segmentation in Required.
rh_annotation	Character; file path. Input file must be <subject_id>/label/rh.aparc.annot Required.
rh_pial	Character; file path. Input file must be <subject_id>/surf/rh.pial Required.
rh_ribbon	Character; file path. Input file must be <subject_id>/mri/rh.ribbon.mgz Required.
rh_white	Character; file path. Input file must be <subject_id>/surf/rh.white Required.
ribbon	Character; file path. Input file must be <subject_id>/mri/ribbon.mgz Required.
subject_id	Character. Subject being processed Required.
a2009s	Logical. Using the a2009s atlas
args	Character. Additional parameters to the command
aseg	Character; file path. Input aseg file
ctxseg	Character; file path
hypo_wm	Logical. Label hypointensities as WM
label_wm	Logical. For each voxel labeled as white matter in the aseg, re-assign its label to be that of the closest cortical point if its distance is less than dmaxctx.
rip_unknown	Logical. Do not label WM based on 'unknown' cortical label
volmask	Logical. Volume mask flag
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_apas2_aseg
FREESURFER Apas2Aseg

Description

Converts aparc+aseg.mgz into something like aseg.mgz by replacing the

Usage

```
ni_freesurfer_apas2_aseg(  
  in_file,  
  out_file,  
  args = NULL,  
  .cwd = NULL,  
  .env = NULL,  
  .engine = NULL,  
  .profile = NULL,  
  dry_run = FALSE,  
  echo = interactive()  
)
```

Arguments

in_file	Character; file path. Input aparc+aseg.mgz Required.
out_file	Character; file path. Output aseg file Required.
args	Character. Additional parameters to the command
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

```
ni_freesurfer_apply_mask
    FREESURFER ApplyMask
```

Description

Use Freesurfer's mri_mask to apply a mask to an image.

Usage

```
ni_freesurfer_apply_mask(
    in_file,
    mask_file,
    args = NULL,
    invert_xfm = NULL,
    keep_mask_deletion_edits = NULL,
    mask_thresh = NULL,
    out_file = NULL,
    transfer = NULL,
    use_abs = NULL,
    xfm_file = NULL,
    xfm_source = NULL,
    xfm_target = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)
```

Arguments

in_file	Character; file path. input image (will be masked) Required.
mask_file	Character; file path. image defining mask space Required.
args	Character. Additional parameters to the command
invert_xfm	Logical. invert transformation
keep_mask_deletion_edits	Logical. transfer voxel-deletion edits (voxels=1) from mask to out vol
mask_thresh	Numeric. threshold mask before applying
out_file	Character; file path. final image to write
transfer	Integer. transfer only voxel value # from mask to out
use_abs	Logical. take absolute value of mask before applying
xfm_file	Character; file path. LTA-format transformation matrix to align mask with input

xfm_source	Character; file path. image defining transform source space
xfm_target	Character; file path. image defining transform target space
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_apply_vol_transform

FREESURFER ApplyVolTransform

Description

Use FreeSurfer mri_vol2vol to apply a transform.

Usage

```
ni_freesurfer_apply_vol_transform(
  fs_target,
  fsl_reg_file,
  lta_file,
  lta_inv_file,
  mni_152_reg,
  reg_file,
  reg_header,
  source_file,
  subject,
  tal,
  target_file,
  xfm_reg_file,
  args = NULL,
  interp = NULL,
  inverse = NULL,
  invert_morph = NULL,
  m3z_file = NULL,
  no_ded_m3z_path = NULL,
  no_resample = NULL,
  tal_resolution = NULL,
  transformed_file = NULL,
```

```

    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

fs_target	Logical. use orig.mgz from subject in regfile as target Required.
fsl_reg_file	Character; file path. fslRAS-to-fslRAS matrix (FSL format) Required.
lta_file	Character; file path. Linear Transform Array file Required.
lta_inv_file	Character; file path. LTA, invert Required.
mni_152_reg	Logical. target MNI152 space Required.
reg_file	Character; file path. tkRAS-to-tkRAS matrix (tkregister2 format) Required.
reg_header	Logical. ScannerRAS-to-ScannerRAS matrix = identity Required.
source_file	Character; file path. Input volume you wish to transform Required.
subject	Character. set matrix = identity and use subject for any templates Required.
tal	Logical. map to a sub FOV of MNI305 (with -reg only) Required.
target_file	Character; file path. Output template volume Required.
xfm_reg_file	Character; file path. ScannerRAS-to-ScannerRAS matrix (MNI format) Required.
args	Character. Additional parameters to the command
interp	Character; one of: "trilin", "nearest", "cubic". Interpolation method (or nearest)
inverse	Logical. sample from target to source
invert_morph	Logical. Compute and use the inverse of the non-linear morph to resample the input volume. To be used by -m3z.
m3z_file	Character; file path. This is the morph to be applied to the volume. Unless the morph is in mri/transforms (eg.: for talairach.m3z computed by reconall), you will need to specify the full path to this morph and use the -noDefM3zPath flag.
no_ded_m3z_path	Logical. To be used with the m3z flag. Instructs the code not to look for them3z morph in the default location (SUBJECTS_DIR/subj/mri/transforms), but instead just use the path indicated in -m3z.
no_resample	Logical. Do not resample; just change vox2ras matrix
tal_resolution	Numeric. Resolution to sample when using tal
transformed_file	Character; file path. Output volume
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.

.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_bb_register
FREESURFER BBRegister

Description

Use FreeSurfer bbregister to register a volume to the Freesurfer anatomical.

Usage

```
ni_freesurfer_bb_register(
    contrast_type,
    source_file,
    subject_id,
    args = NULL,
    dof = NULL,
    epi_mask = NULL,
    fsldof = NULL,
    init = NULL,
    init_cost_file = NULL,
    init_reg_file = NULL,
    intermediate_file = NULL,
    out_fsl_file = NULL,
    out_lta_file = NULL,
    out_reg_file = NULL,
    reg_frame = NULL,
    reg_middle_frame = NULL,
    registered_file = NULL,
    spm_nifti = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)
```

Arguments

contrast_type	Character; one of: "t1", "t2", "bold", "dti". contrast type of image Required.
source_file	Character; file path. source file to be registered Required.
subject_id	Character. freesurfer subject id Required.
args	Character. Additional parameters to the command
dof	Character; one of: "6", "9", "12". number of transform degrees of freedom
epi_mask	Logical. mask out B0 regions in stages 1 and 2
fsl_dof	Integer. degrees of freedom for initial registration (FSL)
init	Character; one of: "coreg", "rr", "spm", "fsl", "header", "best". initialize registration with mri_coreg, spm, fsl, or header
init_cost_file	Character or numeric vector. output initial registration cost file
init_reg_file	Character; file path. existing registration file
intermediate_file	Character; file path. Intermediate image, e.g. in case of partial FOV
out_fsl_file	Character or numeric vector. write the transformation matrix in FSL FLIRT format
out_lta_file	Character or numeric vector. write the transformation matrix in LTA format
out_reg_file	Character; file path. output registration file
reg_frame	Integer. 0-based frame index for 4D source file
reg_middle_frame	Logical. Register middle frame of 4D source file
registered_file	Character or numeric vector. output warped sourcefile either True or filename
spm_nifti	Logical. force use of nifti rather than analyze with SPM
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

ni_freesurfer_binarize

FREESURFER Binarize

Description

Use FreeSurfer mri_binarize to threshold an input volume

Usage

```
ni_freesurfer_binarize(  
  in_file,  
  abs = NULL,  
  args = NULL,  
  bin_col_num = NULL,  
  bin_val = NULL,  
  bin_val_not = NULL,  
  binary_file = NULL,  
  count_file = NULL,  
  dilate = NULL,  
  erode = NULL,  
  erode2d = NULL,  
  frame_no = NULL,  
  invert = NULL,  
  mask_file = NULL,  
  mask_thresh = NULL,  
  match = NULL,  
  max = NULL,  
  merge_file = NULL,  
  min = NULL,  
  rmax = NULL,  
  rmin = NULL,  
  ventricles = NULL,  
  wm = NULL,  
  wm_ven_csf = NULL,  
  zero_edges = NULL,  
  zero_slice_edge = NULL,  
  .cwd = NULL,  
  .env = NULL,  
  .engine = NULL,  
  .profile = NULL,  
  dry_run = FALSE,  
  echo = interactive()  
)
```

Arguments

in_file Character; file path. input volume **Required.**

abs	Logical. take abs of invol first (ie, make unsigned)
args	Character. Additional parameters to the command
bin_col_num	Logical. set binarized voxel value to its column number
bin_val	Integer. set vox within thresh to val (default is 1)
bin_val_not	Integer. set vox outside range to val (default is 0)
binary_file	Character; file path. binary output volume
count_file	Character or numeric vector. save number of hits in ascii file (hits, ntotvox, pct)
dilate	Integer. niters: dilate binarization in 3D
erode	Integer. nerode: erode binarization in 3D (after any dilation)
erode2d	Integer. nerode2d: erode binarization in 2D (after any 3D erosion)
frame_no	Integer. use 0-based frame of input (default is 0)
invert	Logical. set binval=0, binvalnot=1
mask_file	Character; file path. must be within mask
mask_thresh	Numeric. set thresh for mask
match	Character or numeric vector. match instead of threshold
max	Numeric. max thresh
merge_file	Character; file path. merge with mergevol
min	Numeric. min thresh
rmax	Numeric. compute max based on rmax*globalmean
rmin	Numeric. compute min based on rmin*globalmean
ventricles	Logical. set match vals those for aseg ventricles+choroid (not 4th)
wm	Logical. set match vals to 2 and 41 (aseg for cerebral WM)
wm_ven_csf	Logical. WM and ventricular CSF, including choroid (not 4th)
zero_edges	Logical. zero the edge voxels
zero_slice_edge	Logical. zero the edge slice voxels
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

 ni_freesurfer_ca_label

FREESURFER CALabel

Description

Label subcortical structures based in GCA model.

Usage

```
ni_freesurfer_ca_label(
  in_file,
  out_file,
  template,
  transform,
  align = NULL,
  args = NULL,
  aseg = NULL,
  in_vol = NULL,
  intensities = NULL,
  label = NULL,
  no_big_ventricles = NULL,
  prior = NULL,
  relabel_unlikely = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. Input volume for CALabel Required.
out_file	Character; file path. Output file for CALabel Required.
template	Character; file path. Input template for CALabel Required.
transform	Character; file path. Input transform for CALabel Required.
align	Logical. Align CALabel
args	Character. Additional parameters to the command
aseg	Character; file path. Undocumented flag. Autorecon3 uses ../mri/aseg.presurf.mgz as input file
in_vol	Character; file path. set input volume
intensities	Character; file path. input label intensities file(used in longitudinal processing)

label	Character; file path. Undocumented flag. Autorecon3 uses ../label/{hemisphere}.cortex.label as input file
no_big_ventricles	Logical. No big ventricles
prior	Numeric. Prior for CALabel
relabel_unlikely	Character or numeric vector. Reclassify voxels at least some std devs from the mean using some size Gaussian window
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_ca_normalize

FREESURFER CANormalize

Description

This program creates a normalized volume using the brain volume and an

Usage

```
ni_freesurfer_ca_normalize(
  atlas,
  in_file,
  transform,
  args = NULL,
  control_points = NULL,
  long_file = NULL,
  mask = NULL,
  out_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

atlas	Character; file path. The atlas file in gca format Required.
in_file	Character; file path. The input file for CANormalize Required.
transform	Character; file path. The transform file in lta format Required.
args	Character. Additional parameters to the command
control_points	Character; file path. File name for the output control points
long_file	Character; file path. undocumented flag used in longitudinal processing
mask	Character; file path. Specifies volume to use as mask
out_file	Character; file path. The output file for CANormalize
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_ca_register
FREESURFER CARegister

Description

Generates a multi-dimensional talairach transform from a gca file and talairach.lta file

Usage

```
ni_freesurfer_ca_register(
  in_file,
  A = NULL,
  align = NULL,
  args = NULL,
  invert_and_save = NULL,
  l_files = NULL,
  levels = NULL,
  mask = NULL,
  no_big_ventricles = NULL,
  out_file = NULL,
  template = NULL,
```

```

    transform = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

in_file	Character; file path. The input volume for CARegister Required.
A	Integer. undocumented flag used in longitudinal processing
align	Character. Specifies when to perform alignment
args	Character. Additional parameters to the command
invert_and_save	Logical. Invert and save the .m3z multi-dimensional talarach transform to x, y, and z .mgz files
l_files	Character or numeric vector. undocumented flag used in longitudinal processing
levels	Integer. defines how many surrounding voxels will be used in interpolations, default is 6
mask	Character; file path. Specifies volume to use as mask
no_big_ventricles	Logical. No big ventricles
out_file	Character; file path. The output volume for CARegister
template	Character; file path. The template file in gca format
transform	Character; file path. Specifies transform in lta format
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

```
ni_freesurfer_check_talairach_alignment
    FREESURFER CheckTalairachAlignment
```

Description

This program detects Talairach alignment failures

Usage

```
ni_freesurfer_check_talairach_alignment(
    in_file,
    subject,
    args = NULL,
    threshold = 0.01,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)
```

Arguments

in_file	Character; file path. specify the talairach.xfm file to check Required.
subject	Character. specify subject's name Required.
args	Character. Additional parameters to the command
threshold	Numeric. Talairach transforms for subjects with p-values $\leq T$ are considered as very unlikely default=0.010
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_concatenate
FREESURFER Concatenate

Description

Use Freesurfer mri_concat to combine several input volumes

Usage

```
ni_freesurfer_concatenate(
  in_files,
  add_val = NULL,
  args = NULL,
  combine = NULL,
  concatenated_file = NULL,
  gmean = NULL,
  keep_dtype = NULL,
  mask_file = NULL,
  max_bonfcor = NULL,
  max_index = NULL,
  mean_div_n = NULL,
  multiply_by = NULL,
  multiply_matrix_file = NULL,
  paired_stats = NULL,
  sign = NULL,
  sort = NULL,
  stats = NULL,
  vote = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_files	Character or numeric vector. Individual volumes to be concatenated Required.
add_val	Numeric. Add some amount to the input volume
args	Character. Additional parameters to the command
combine	Logical. Combine non-zero values into single frame volume
concatenated_file	Character; file path. Output volume
gmean	Integer. create matrix to average Ng groups, Nper=Ntot/Ng

keep_dtype	Logical. Keep voxelwise precision type (default is float)
mask_file	Character; file path. Mask input with a volume
max_bonfcor	Logical. Compute max and bonferroni correct (assumes -log10(ps))
max_index	Logical. Compute the index of max voxel in concatenated volumes
mean_div_n	Logical. compute mean/nframes (good for var)
multiply_by	Numeric. Multiply input volume by some amount
multiply_matrix_file	Character; file path. Multiply input by an ascii matrix in file
paired_stats	Character; one of: "sum", "avg", "diff", "diff-norm", "diff-norm1", "diff-norm2". Compute paired sum, avg, or diff
sign	Character; one of: "abs", "pos", "neg". Take only pos or neg voxles from input, or take abs
sort	Logical. Sort each voxel by ascending frame value
stats	Character; one of: "sum", "var", "std", "max", "min", "mean". Compute the sum, var, std, max, min or mean of the input volumes
vote	Logical. Most frequent value at each voxel and fraction of occurrences
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_concatenate_lta
FREESURFER ConcatenateLTA

Description

Concatenates two consecutive LTA transformations into one overall

Usage

```

ni_freesurfer_concatenate_lta(
  in_lta1,
  in_lta2,
  args = NULL,
  invert_1 = NULL,
  invert_2 = NULL,
  invert_out = NULL,
  out_file = NULL,
  out_type = NULL,
  subject = NULL,
  tal_source_file = NULL,
  tal_template_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)

```

Arguments

in_lta1	Character; file path. maps some src1 to dst1 Required.
in_lta2	Character or numeric vector. maps dst1(src2) to dst2 Required.
args	Character. Additional parameters to the command
invert_1	Logical. invert in_lta1 before applying it
invert_2	Logical. invert in_lta2 before applying it
invert_out	Logical. invert output LTA
out_file	Character; file path. the combined LTA maps: src1 to dst2 = LTA2*LTA1
out_type	Character; one of: "VOX2VOX", "RAS2RAS". set final LTA type
subject	Character. set subject in output LTA
tal_source_file	Character; file path. if in_lta2 is talairach.xfm, specify source for talairach
tal_template_file	Character; file path. if in_lta2 is talairach.xfm, specify template for talairach
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_contrast

FREESURFER Contrast

Description

Compute surface-wise gray/white contrast

Usage

```
ni_freesurfer_contrast(
  annotation,
  cortex,
  hemisphere,
  orig,
  rawavg,
  subject_id,
  thickness,
  white,
  args = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

annotation	Character; file path. Input annotation file must be <subject_id>/label/.aparc.annot Required.
cortex	Character; file path. Input cortex label must be <subject_id>/label/.cortex.label Required.
hemisphere	Character; one of: "lh", "rh". Hemisphere being processed Required.
orig	Character; file path. Implicit input file mri/orig.mgz Required.
rawavg	Character; file path. Implicit input file mri/rawavg.mgz Required.
subject_id	Character. Subject being processed Required.
thickness	Character; file path. Input file must be <subject_id>/surf/?h.thickness Required.
white	Character; file path. Input file must be <subject_id>/surf/.white Required.
args	Character. Additional parameters to the command

.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_curvature
FREESURFER Curvature

Description

This program will compute the second fundamental form of a cortical

Usage

```
ni_freesurfer_curvature(
  in_file,
  args = NULL,
  averages = NULL,
  distances = NULL,
  n = NULL,
  save = NULL,
  threshold = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. Input file for Curvature Required.
args	Character. Additional parameters to the command
averages	Integer. Perform this number iterative averages of curvature measure before saving
distances	Character or numeric vector. Undocumented input integer distances
n	Logical. Undocumented boolean flag

save	Logical. Save curvature files (will only generate screen output without this option)
threshold	Numeric. Undocumented input threshold
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_curvature_stats
FREESURFER CurvatureStats

Description

In its simplest usage, 'mris_curvature_stats' will compute a set

Usage

```
ni_freesurfer_curvature_stats(
  curvfile1,
  curvfile2,
  hemisphere,
  subject_id,
  args = NULL,
  min_max = NULL,
  out_file = NULL,
  surface = NULL,
  values = NULL,
  write = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

curvfile1	Character; file path. Input file for CurvatureStats Required.
curvfile2	Character; file path. Input file for CurvatureStats Required.
hemisphere	Character; one of: "lh", "rh". Hemisphere being processed Required.
subject_id	Character. Subject being processed Required.
args	Character. Additional parameters to the command
min_max	Logical. Output min / max information for the processed curvature.
out_file	Character; file path. Output curvature stats file
surface	Character; file path. Specify surface file for CurvatureStats
values	Logical. Triggers a series of derived curvature values
write	Logical. Write curvature files
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_dicom_convert
FREESURFER DICOMConvert

Description

use fs mri_convert to convert dicom files

Usage

```
ni_freesurfer_dicom_convert(
  base_output_dir,
  dicom_dir,
  args = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

base_output_dir	Character; directory path. directory in which subject directories are created Required.
dicom_dir	Character; directory path. dicom directory from which to convert dicom files Required.
args	Character. Additional parameters to the command
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_edit_w_mwith_aseg
FREESURFER EditWMwithAseg

Description

Edits a wm file using a segmentation

Usage

```
ni_freesurfer_edit_w_mwith_aseg(
  brain_file,
  in_file,
  out_file,
  seg_file,
  args = NULL,
  keep_in = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

brain_file	Character; file path. Input brain/T1 file Required.
in_file	Character; file path. Input white matter segmentation file Required.
out_file	Character; file path. File to be written as output Required.
seg_file	Character; file path. Input presurf segmentation file Required.
args	Character. Additional parameters to the command
keep_in	Logical. Keep edits as found in input volume
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_em_register
FREESURFER EMRegister

Description

This program creates a transform in lta format

Usage

```
ni_freesurfer_em_register(
  in_file,
  template,
  args = NULL,
  mask = NULL,
  nbrspacing = NULL,
  out_file = NULL,
  skull = NULL,
  transform = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. in brain volume Required.
template	Character; file path. template gca Required.
args	Character. Additional parameters to the command
mask	Character; file path. use volume as a mask
nbrspacing	Integer. align to atlas containing skull setting unknown_nbr_spacing = nbrspacing
out_file	Character; file path. output transform
skull	Logical. align to atlas containing skull (uns=5)
transform	Character; file path. Previously computed transform
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_euler_number
FREESURFER EulerNumber

Description

This program computes EulerNumber for a cortical surface

Usage

```
ni_freesurfer_euler_number(
  in_file,
  args = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. Input file for EulerNumber Required.
args	Character. Additional parameters to the command
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_extract_main_component
FREESURFER ExtractMainComponent

Description

Extract the main component of a tessellated surface

Usage

```
ni_freesurfer_extract_main_component(
  in_file,
  args = NULL,
  out_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. input surface file Required.
args	Character. Additional parameters to the command
out_file	Character; file path. surface containing main component
.cwd	Working directory override.
.env	Named character vector of environment variables.

.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_fit_ms_params
FREESURFER FitMSParams

Description

Estimate tissue parameters from a set of FLASH images.

Usage

```
ni_freesurfer_fit_ms_params(
  in_files,
  args = NULL,
  out_dir = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_files	Character or numeric vector. list of FLASH images (must be in mgh format) Required.
args	Character. Additional parameters to the command
out_dir	Character; directory path. directory to store output in
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

```
ni_freesurfer_fix_topology
    FREESURFER FixTopology
```

Description

This program computes a mapping from the unit sphere onto the surface

Usage

```
ni_freesurfer_fix_topology(
    copy_inputs,
    hemisphere,
    in_brain,
    in_inflated,
    in_orig,
    in_wm,
    subject_id,
    args = NULL,
    ga = NULL,
    mgz = NULL,
    seed = NULL,
    sphere = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)
```

Arguments

copy_inputs	Logical. If running as a node, set this to True otherwise, the topology fixing will be done in place. Required.
hemisphere	Character. Hemisphere being processed Required.
in_brain	Character; file path. Implicit input brain.mgz Required.
in_inflated	Character; file path. Undocumented input file .inflated Required.
in_orig	Character; file path. Undocumented input file .orig Required.
in_wm	Character; file path. Implicit input wm.mgz Required.
subject_id	Character. Subject being processed Required.
args	Character. Additional parameters to the command
ga	Logical. No documentation. Direct questions to analysis-bugs@nmr.mgh.harvard.edu
mgz	Logical. No documentation. Direct questions to analysis-bugs@nmr.mgh.harvard.edu

seed	Integer. Seed for setting random number generator
sphere	Character; file path. Sphere input file
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_fuse_segmentations
FREESURFER FuseSegmentations

Description

fuse segmentations together from multiple timepoints

Usage

```
ni_freesurfer_fuse_segmentations(
  in_norms,
  in_segmentations,
  in_segmentations_noCC,
  out_file,
  timepoints,
  args = NULL,
  subject_id = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_norms Character or numeric vector. -n - name of norm file to use (default: norm.mgs) must include the corresponding norm file for all given timepoints as well as for the current subject **Required.**

in_segmentations	Character or numeric vector. name of aseg file to use (default: aseg.mgz) must include the aseg files for all the given timepoints Required.
in_segmentations_noCC	Character or numeric vector. name of aseg file w/o CC labels (default: aseg.auto_noCCseg.mgz) must include the corresponding file for all the given timepoints Required.
out_file	Character; file path. output fused segmentation file Required.
timepoints	Character or numeric vector. subject_ids or timepoints to be processed Required.
args	Character. Additional parameters to the command
subject_id	Character. subject_id being processed
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_glm_fit *FREESURFER GLMFit*

Description

Use FreeSurfer's mri_glmfit to specify and estimate a general linear model.

Usage

```
ni_freesurfer_glm_fit(
  in_file,
  allow_ill_cond = NULL,
  allow_repeated_subjects = NULL,
  args = NULL,
  bp_clip_max = NULL,
  bp_clip_neg = NULL,
  calc_AR1 = NULL,
  check_opts = NULL,
  compute_log_y = NULL,
  contrast = NULL,
  cortex = NULL,
  debug = NULL,
```

```
design = NULL,  
diag = NULL,  
diag_cluster = NULL,  
fixed_fx_dof = NULL,  
fixed_fx_dof_file = NULL,  
fixed_fx_var = NULL,  
force_perm = NULL,  
fsgd = NULL,  
fwhm = NULL,  
glm_dir = NULL,  
invert_mask = NULL,  
label_file = NULL,  
logan = NULL,  
mask_file = NULL,  
mrtm1 = NULL,  
mrtm2 = NULL,  
nii = NULL,  
nii_gz = NULL,  
no_contrast_ok = NULL,  
no_est_fwhm = NULL,  
no_mask_smooth = NULL,  
no_prune = NULL,  
one_sample = NULL,  
pca = NULL,  
per_voxel_reg = NULL,  
profile = NULL,  
prune = NULL,  
prune_thresh = NULL,  
resynth_test = NULL,  
save_cond = NULL,  
save_estimate = NULL,  
save_res_corr_mtx = NULL,  
save_residual = NULL,  
seed = NULL,  
self_reg = NULL,  
sim_done_file = NULL,  
sim_sign = NULL,  
simulation = NULL,  
surf = NULL,  
synth = NULL,  
uniform = NULL,  
var_fwhm = NULL,  
vox_dump = NULL,  
weight_inv = NULL,  
weight_sqrt = NULL,  
weighted_ls = NULL,  
.cwd = NULL,  
.env = NULL,
```

```

    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

in_file	Character; file path. input 4D file Required.
allow_ill_cond	Logical. allow ill-conditioned design matrices
allow_repeated_subjects	Logical. allow subject names to repeat in the fsgd file (must appear before -fsgd)
args	Character. Additional parameters to the command
bp_clip_max	Numeric. set BP voxels above max to max
bp_clip_neg	Logical. set negative BP voxels to zero
calc_AR1	Logical. compute and save temporal AR1 of residual
check_opts	Logical. don't run anything, just check options and exit
compute_log_y	Logical. compute natural log of y prior to analysis
contrast	Character or numeric vector. contrast file
cortex	Logical. use subjects ?h.cortex.label as label
debug	Logical. turn on debugging
design	Character; file path. design matrix file
diag	Integer. Gdiag_no : set diagnostic level
diag_cluster	Logical. save sig volume and exit from first sim loop
fixed_fx_dof	Integer. dof for fixed effects analysis
fixed_fx_dof_file	Character; file path. text file with dof for fixed effects analysis
fixed_fx_var	Character; file path. for fixed effects analysis
force_perm	Logical. force permutation test, even when design matrix is not orthog
fsgd	Character or numeric vector. freesurfer descriptor file
fwhm	Character. smooth input by fwhm
glm_dir	Character. save outputs to dir
invert_mask	Logical. invert mask
label_file	Character; file path. use label as mask, surfaces only
logan	Character or numeric vector. RefTac TimeSec tstar : perform Logan kinetic modeling
mask_file	Character; file path. binary mask
mrtm1	Character or numeric vector. RefTac TimeSec : perform MRTM1 kinetic modeling
mrtm2	Character or numeric vector. RefTac TimeSec k2prime : perform MRTM2 kinetic modeling

nii	Logical. save outputs as nii
nii_gz	Logical. save outputs as nii.gz
no_contrast_ok	Logical. do not fail if no contrasts specified
no_est_fwhm	Logical. turn off FWHM output estimation
no_mask_smooth	Logical. do not mask when smoothing
no_prune	Logical. do not prune
one_sample	Logical. construct X and C as a one-sample group mean
pca	Logical. perform pca/svd analysis on residual
per_voxel_reg	Character or numeric vector. per-voxel regressors
profile	Integer. niters : test speed
prune	Logical. remove voxels that do not have a non-zero value at each frame (def)
prune_thresh	Numeric. prune threshold. Default is FLT_MIN
resynth_test	Integer. test GLM by resynthesis
save_cond	Logical. flag to save design matrix condition at each voxel
save_estimate	Logical. save signal estimate (yhat)
save_res_corr_mtx	Logical. save residual error spatial correlation matrix (eres.scm). Big!
save_residual	Logical. save residual error (eres)
seed	Integer. used for synthesizing noise
self_reg	Character or numeric vector. self-regressor from index col row slice
sim_done_file	Character; file path. create file when simulation finished
sim_sign	Character; one of: "abs", "pos", "neg". abs, pos, or neg
simulation	Character or numeric vector. nulltype nsim thresh csdbasename
surf	Logical. analysis is on a surface mesh
synth	Logical. replace input with gaussian
uniform	Character or numeric vector. use uniform distribution instead of gaussian
var_fwhm	Character. smooth variance by fwhm
vox_dump	Character or numeric vector. dump voxel GLM and exit
weight_inv	Logical. invert weights
weight_sqrt	Logical. sqrt of weights
weighted_ls	Character; file path. weighted least squares
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

 ni_freesurfer_gtm_seg *FREESURFER GTMSeg*

Description

create an anatomical segmentation for the geometric transfer matrix (GTM).

Usage

```
ni_freesurfer_gtm_seg(
  subject_id,
  args = NULL,
  colortable = NULL,
  ctx_annot = NULL,
  dmax = NULL,
  head = NULL,
  keep_cc = NULL,
  keep_hypo = NULL,
  no_pons = NULL,
  no_seg_stats = NULL,
  no_vermis = NULL,
  out_file = "gtmseg.mgz",
  output_upsampling_factor = NULL,
  subseg_cblum_wm = NULL,
  subsegwm = NULL,
  upsampling_factor = NULL,
  wm_annot = NULL,
  xcarseg = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

subject_id	Character. subject id Required.
args	Character. Additional parameters to the command
colortable	Character; file path. colortable
ctx_annot	Character or numeric vector. annot lhbase rhbase : annotation to use for cortical segmentation (default is aparc 1000 2000)
dmax	Numeric. distance threshold to use when subsegmenting WM (default is 5)
head	Character. use headseg instead of apas+head.mgz

keep_cc	Logical. do not relabel corpus callosum as WM
keep_hypo	Logical. do not relabel hypointensities as WM when subsegmenting WM
no_pons	Logical. do not add pons segmentation when doing <code>—xcerseg</code>
no_seg_stats	Logical. do not compute segmentation stats
no_vermis	Logical. do not add vermis segmentation when doing <code>—xcerseg</code>
out_file	Character; file path. output volume relative to subject/mri
output_upsampling_factor	Integer. set output USF different than USF, mostly for debugging
subseg_cblum_wm	Logical. subsegment cerebellum WM into core and gyri
subsegwm	Logical. subsegment WM into lobes (default)
upsampling_factor	Integer. upsampling factor (default is 2)
wm_annot	Character or numeric vector. annot lhbase rhbase : annotation to use for WM segmentation (with <code>—subsegwm</code> , default is lobes 3200 4200)
xcerseg	Logical. run xcerebralseg on this subject to create apas+head.mgz
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

ni_freesurfer_gtmpvc *FREESURFER GTMPVC*

Description

Perform Partial Volume Correction (PVC) to PET Data.

Usage

```
ni_freesurfer_gtmpvc(
  in_file,
  reg_file,
  reg_identity,
  regheader,
  segmentation,
```

```
X = NULL,  
X0 = NULL,  
args = NULL,  
auto_mask = NULL,  
beta = NULL,  
color_table_file = NULL,  
contrast = NULL,  
default_color_table = NULL,  
default_seg_merge = NULL,  
frame = NULL,  
km_hb = NULL,  
km_ref = NULL,  
lat = NULL,  
mask_file = NULL,  
merge_cblum_wm_gyri = NULL,  
merge_hypos = NULL,  
mg = NULL,  
mg_ref_cerebral_wm = NULL,  
mg_ref_lobes_wm = NULL,  
mgx = NULL,  
no_pvc = NULL,  
no_reduce_fov = NULL,  
no_rescale = NULL,  
no_tfe = NULL,  
num_threads = NULL,  
opt_brain = NULL,  
opt_seg_merge = NULL,  
opt_tol = NULL,  
optimization_schema = NULL,  
psf = NULL,  
psf_col = NULL,  
psf_row = NULL,  
psf_slice = NULL,  
pvc_dir = NULL,  
rbv = NULL,  
rbv_res = NULL,  
reduce_fox_eqodd = NULL,  
replace = NULL,  
rescale = NULL,  
save_eres = NULL,  
save_input = NULL,  
save_yhat = NULL,  
save_yhat0 = NULL,  
save_yhat_full_fov = NULL,  
save_yhat_with_noise = NULL,  
scale_refval = NULL,  
steady_state_params = NULL,  
tissue_fraction_resolution = NULL,
```

```

tt_reduce = NULL,
tt_update = NULL,
y = NULL,
.cwd = NULL,
.env = NULL,
.engine = NULL,
.profile = NULL,
dry_run = FALSE,
echo = interactive()
)

```

Arguments

in_file	Character; file path. input volume - source data to pvc Required.
reg_file	Character; file path. LTA registration file that maps PET to anatomical Required.
reg_identity	Logical. assume that input is in anatomical space Required.
regheader	Logical. assume input and seg share scanner space Required.
segmentation	Character; file path. segfile : anatomical segmentation to define regions for GTM Required.
X	Logical. save X matrix in matlab4 format as X.mat (it will be big)
X0	Logical. save X0 matrix in matlab4 format as X0.mat (it will be big)
args	Character. Additional parameters to the command
auto_mask	Character or numeric vector. FWHM thresh : automatically compute mask
beta	Logical. save beta matrix in matlab4 format as beta.mat
color_table_file	Character; file path. color table file with seg id names
contrast	Character or numeric vector. contrast file
default_color_table	Logical. use \$FREESURFER_HOME/FreeSurferColorLUT.txt
default_seg_merge	Logical. default schema for merging ROIs
frame	Integer. only process 0-based frame F from inputvol
km_hb	Character or numeric vector. RefId1 RefId2 ... : compute HiBinding TAC for KM as mean of given RefIds
km_ref	Character or numeric vector. RefId1 RefId2 ... : compute reference TAC for KM as mean of given RefIds
lat	Logical. lateralize tissue types
mask_file	Character; file path. ignore areas outside of the mask (in input vol space)
merge_cblum_wm_gyri	Logical. cerebellum WM gyri back into cerebellum WM
merge_hypos	Logical. merge left and right hypointensities into ROI

mg	Character or numeric vector. gmthresh RefId1 RefId2 ...: perform Mueller-Gaertner PVC, gmthresh is min gm pvf bet 0 and 1
mg_ref_cerebral_wm	Logical. set MG RefIds to 2 and 41
mg_ref_lobes_wm	Logical. set MG RefIds to those for lobes when using wm subseg
mgx	Numeric. gmxtresh : GLM-based Mueller-Gaertner PVC, gmxtresh is min gm pvf bet 0 and 1
no_pvc	Logical. turns off PVC entirely (both PSF and TFE)
no_reduce_fov	Logical. do not reduce FoV to encompass mask
no_rescale	Logical. do not global rescale such that mean of reference region is scaleref
no_tfe	Logical. do not correct for tissue fraction effect (with -psf 0 turns off PVC entirely)
num_threads	Integer. threads : number of threads to use
opt_brain	Logical. apply adaptive GTM
opt_seg_merge	Logical. optimal schema for merging ROIs when applying adaptive GTM
opt_tol	Character or numeric vector. n_iters_max ftol lin_min_tol : optimization parameters for adaptive gtm using fminsearch
optimization_schema	Character; one of: "3D", "2D", "1D", "3D_MB", "2D_MB", "1D_MB", "MBZ", "MB3". opt : optimization schema for applying adaptive GTM
psf	Numeric. scanner PSF FWHM in mm
psf_col	Numeric. xFWHM : full-width-half-maximum in the x-direction
psf_row	Numeric. yFWHM : full-width-half-maximum in the y-direction
psf_slice	Numeric. zFWHM : full-width-half-maximum in the z-direction
pvc_dir	Character. save outputs to dir
rbv	Logical. perform Region-based Voxelwise (RBV) PVC
rbv_res	Numeric. voxsize : set RBV voxel resolution (good for when standard res takes too much memory)
reduce_fov_eqodd	Logical. reduce FoV to encompass mask but force nc=nr and ns to be odd
replace	Character or numeric vector. Id1 Id2 : replace seg Id1 with seg Id2
rescale	Character or numeric vector. Id1 <Id2...> : specify reference region(s) used to rescale (default is pons)
save_eres	Logical. saves residual error
save_input	Logical. saves rescaled input as input.rescaled.nii.gz
save_yhat	Logical. save signal estimate (yhat) smoothed with the PSF
save_yhat0	Logical. save signal estimate (yhat)
save_yhat_full_fov	Logical. save signal estimate (yhat)

save_yhat_with_noise	Character or numeric vector. seed nreps : save signal estimate (yhat) with noise
scale_refval	Numeric. refval : scale such that mean in reference region is refval
steady_state_params	Character or numeric vector. bpc scale dcf : steady-state analysis spec blood plasma concentration, unit scale and decay correction factor. You must also spec -km-ref. Turns off rescaling
tissue_fraction_resolution	Numeric. set the tissue fraction resolution parameter (def is 0.5)
tt_reduce	Logical. reduce segmentation to that of a tissue type
tt_update	Logical. changes tissue type of VentralDC, BrainStem, and Pons to be SubcortGM
y	Logical. save y matrix in matlab4 format as y.mat
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_image_info
FREESURFER ImageInfo

Description

General support for FreeSurfer commands.

Usage

```
ni_freesurfer_image_info(  
  args = NULL,  
  in_file = NULL,  
  .cwd = NULL,  
  .env = NULL,  
  .engine = NULL,  
  .profile = NULL,  
  dry_run = FALSE,  
  echo = interactive()  
)
```

Arguments

args	Character. Additional parameters to the command
in_file	Character; file path. image to query
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_jacobian
FREESURFER Jacobian

Description

This program computes the Jacobian of a surface mapping.

Usage

```
ni_freesurfer_jacobian(
  in_mappedsurf,
  in_origsurf,
  args = NULL,
  out_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_mappedsurf	Character; file path. Mapped surface Required.
in_origsurf	Character; file path. Original surface Required.
args	Character. Additional parameters to the command
out_file	Character; file path. Output Jacobian of the surface mapping

.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_label2_annot
FREESURFER Label2Annot

Description

Converts a set of surface labels to an annotation file

Usage

```
ni_freesurfer_label2_annot(
  hemisphere,
  in_labels,
  orig,
  out_annot,
  subject_id,
  args = NULL,
  color_table = NULL,
  keep_max = NULL,
  verbose_off = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

hemisphere	Character; one of: "lh", "rh". Input hemisphere Required.
in_labels	Character or numeric vector. List of input label files Required.
orig	Character; file path. implicit {hemisphere}.orig Required.
out_annot	Character. Name of the annotation to create Required.

subject_id	Character. Subject name/ID Required.
args	Character. Additional parameters to the command
color_table	Character; file path. File that defines the structure names, their indices, and their color
keep_max	Logical. Keep label with highest 'stat' value
verbose_off	Logical. Turn off overlap and stat override messages
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_label2_label

FREESURFER Label2Label

Description

Converts a label in one subject's space to a label

Usage

```
ni_freesurfer_label2_label(
  hemisphere,
  source_label,
  source_sphere_reg,
  source_subject,
  source_white,
  sphere_reg,
  subject_id,
  white,
  args = NULL,
  out_file = NULL,
  registration_method = "surface",
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

hemisphere	Character; one of: "lh", "rh". Input hemisphere Required.
source_label	Character; file path. Source label Required.
source_sphere_reg	Character; file path. Implicit input .sphere.reg Required.
source_subject	Character. Source subject name Required.
source_white	Character; file path. Implicit input .white Required.
sphere_reg	Character; file path. Implicit input .sphere.reg Required.
subject_id	Character. Target subject Required.
white	Character; file path. Implicit input .white Required.
args	Character. Additional parameters to the command
out_file	Character; file path. Target label
registration_method	Character; one of: "surface", "volume". Registration method
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_label2_vol

FREESURFER Label2Vol

Description

Make a binary volume from a Freesurfer label

Usage

```
ni_freesurfer_label2_vol(
  annot_file,
  aparc_aseg,
  label_file,
  seg_file,
  template_file,
  args = NULL,
```

```

    fill_thresh = NULL,
    hemi = NULL,
    identity = NULL,
    invert_mtx = NULL,
    label_hit_file = NULL,
    label_voxel_volume = NULL,
    map_label_stat = NULL,
    native_vox2ras = NULL,
    proj = NULL,
    reg_file = NULL,
    reg_header = NULL,
    subject_id = NULL,
    surface = NULL,
    vol_label_file = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

annot_file	Character; file path. surface annotation file Required.
aparc_aseg	Logical. use aparc+aseg.mgz in subjectdir as seg Required.
label_file	Character or numeric vector. list of label files Required.
seg_file	Character; file path. segmentation file Required.
template_file	Character; file path. output template volume Required.
args	Character. Additional parameters to the command
fill_thresh	Character. thresh : between 0 and 1
hemi	Character; one of: "lh", "rh". hemisphere to use lh or rh
identity	Logical. set R=I
invert_mtx	Logical. Invert the registration matrix
label_hit_file	Character; file path. file with each frame is nhits for a label
label_voxel_volume	Numeric. volume of each label point (def 1mm3)
map_label_stat	Character; file path. map the label stats field into the vol
native_vox2ras	Logical. use native vox2ras xform instead of tkregister-style
proj	Character or numeric vector. project along surface normal
reg_file	Character; file path. tkregister style matrix VolXYZ = R*LabelXYZ
reg_header	Character; file path. label template volume
subject_id	Character. subject id

surface	Character. use surface instead of white
vol_label_file	Character; file path. output volume
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_logan *FREESURFER Logan*

Description

Perform Logan kinetic modeling.

Usage

```
ni_freesurfer_logan(
  in_file,
  logan,
  allow_ill_cond = NULL,
  allow_repeated_subjects = NULL,
  args = NULL,
  bp_clip_max = NULL,
  bp_clip_neg = NULL,
  calc_AR1 = NULL,
  check_opts = NULL,
  compute_log_y = NULL,
  contrast = NULL,
  cortex = NULL,
  debug = NULL,
  design = NULL,
  diag = NULL,
  diag_cluster = NULL,
  fixed_fx_dof = NULL,
  fixed_fx_dof_file = NULL,
  fixed_fx_var = NULL,
  force_perm = NULL,
  fsgd = NULL,
  fwhm = NULL,
```

```

    glm_dir = NULL,
    invert_mask = NULL,
    label_file = NULL,
    mask_file = NULL,
    mrtm1 = NULL,
    mrtm2 = NULL,
    nii = NULL,
    nii_gz = NULL,
    no_contrast_ok = NULL,
    no_est_fwhm = NULL,
    no_mask_smooth = NULL,
    no_prune = NULL,
    one_sample = NULL,
    pca = NULL,
    per_voxel_reg = NULL,
    profile = NULL,
    prune = NULL,
    prune_thresh = NULL,
    resynth_test = NULL,
    save_cond = NULL,
    save_estimate = NULL,
    save_res_corr_mtx = NULL,
    save_residual = NULL,
    seed = NULL,
    self_reg = NULL,
    sim_done_file = NULL,
    sim_sign = NULL,
    simulation = NULL,
    surf = NULL,
    synth = NULL,
    uniform = NULL,
    var_fwhm = NULL,
    vox_dump = NULL,
    weight_inv = NULL,
    weight_sqrt = NULL,
    weighted_ls = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

in_file	Character; file path. input 4D file Required.
logan	Character or numeric vector. RefTac TimeSec tstar : perform Logan kinetic

modeling **Required.**

allow_ill_cond	Logical. allow ill-conditioned design matrices
allow_repeated_subjects	Logical. allow subject names to repeat in the fsgd file (must appear before -fsgd)
args	Character. Additional parameters to the command
bp_clip_max	Numeric. set BP voxels above max to max
bp_clip_neg	Logical. set negative BP voxels to zero
calc_AR1	Logical. compute and save temporal AR1 of residual
check_opts	Logical. don't run anything, just check options and exit
compute_log_y	Logical. compute natural log of y prior to analysis
contrast	Character or numeric vector. contrast file
cortex	Logical. use subjects ?h.cortex.label as label
debug	Logical. turn on debugging
design	Character; file path. design matrix file
diag	Integer. Gdiag_no : set diagnostic level
diag_cluster	Logical. save sig volume and exit from first sim loop
fixed_fx_dof	Integer. dof for fixed effects analysis
fixed_fx_dof_file	Character; file path. text file with dof for fixed effects analysis
fixed_fx_var	Character; file path. for fixed effects analysis
force_perm	Logical. force permutation test, even when design matrix is not orthog
fsgd	Character or numeric vector. freesurfer descriptor file
fwhm	Character. smooth input by fwhm
glm_dir	Character. save outputs to dir
invert_mask	Logical. invert mask
label_file	Character; file path. use label as mask, surfaces only
mask_file	Character; file path. binary mask
mrtm1	Character or numeric vector. RefTac TimeSec : perform MRTM1 kinetic modeling
mrtm2	Character or numeric vector. RefTac TimeSec k2prime : perform MRTM2 kinetic modeling
nii	Logical. save outputs as nii
nii_gz	Logical. save outputs as nii.gz
no_contrast_ok	Logical. do not fail if no contrasts specified
no_est_fwhm	Logical. turn off FWHM output estimation
no_mask_smooth	Logical. do not mask when smoothing
no_prune	Logical. do not prune
one_sample	Logical. construct X and C as a one-sample group mean

pca	Logical. perform pca/svd analysis on residual
per_voxel_reg	Character or numeric vector. per-voxel regressors
profile	Integer. niters : test speed
prune	Logical. remove voxels that do not have a non-zero value at each frame (def)
prune_thresh	Numeric. prune threshold. Default is FLT_MIN
resynth_test	Integer. test GLM by resynthesis
save_cond	Logical. flag to save design matrix condition at each voxel
save_estimate	Logical. save signal estimate (yhat)
save_res_corr_mtx	Logical. save residual error spatial correlation matrix (eres.scm). Big!
save_residual	Logical. save residual error (eres)
seed	Integer. used for synthesizing noise
self_reg	Character or numeric vector. self-regressor from index col row slice
sim_done_file	Character; file path. create file when simulation finished
sim_sign	Character; one of: "abs", "pos", "neg". abs, pos, or neg
simulation	Character or numeric vector. nulltype nsim thresh csdbasename
surf	Logical. analysis is on a surface mesh
synth	Logical. replace input with gaussian
uniform	Character or numeric vector. use uniform distribution instead of gaussian
var_fwhm	Character. smooth variance by fwhm
vox_dump	Character or numeric vector. dump voxel GLM and exit
weight_inv	Logical. invert weights
weight_sqrt	Logical. sqrt of weights
weighted_ls	Character; file path. weighted least squares
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

```
ni_freesurfer_lta_convert
    FREESURFER LTAConvert
```

Description

Convert different transformation formats.

Usage

```
ni_freesurfer_lta_convert(
    in_fsl,
    in_itk,
    in_lta,
    in_mni,
    in_niftyreg,
    in_reg,
    args = NULL,
    invert = NULL,
    ltavox2vox = NULL,
    out_fsl = NULL,
    out_itk = NULL,
    out_lta = NULL,
    out_mni = NULL,
    out_reg = NULL,
    source_file = NULL,
    target_conform = NULL,
    target_file = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)
```

Arguments

in_fsl	Character; file path. input transform of FSL type Required.
in_itk	Character; file path. input transform of ITK type Required.
in_lta	Character or numeric vector. input transform of LTA type Required.
in_mni	Character; file path. input transform of MNI/XFM type Required.
in_niftyreg	Character; file path. input transform of Nifty Reg type (inverse RAS2RAS) Required.
in_reg	Character; file path. input transform of TK REG type (deprecated format) Required.

args	Character. Additional parameters to the command
invert	Logical
ltavox2vox	Logical
out_fsl	Character or numeric vector. output transform in FSL format
out_itk	Character or numeric vector. output transform in ITK format
out_lta	Character or numeric vector. output linear transform (LTA Freesurfer format)
out_mni	Character or numeric vector. output transform in MNI/XFM format
out_reg	Character or numeric vector. output transform in reg dat format
source_file	Character; file path
target_conform	Logical
target_file	Character; file path
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_make_average_subject
FREESURFER MakeAverageSubject

Description

Make an average freesurfer subject

Usage

```
ni_freesurfer_make_average_subject(
  subjects_ids,
  args = NULL,
  out_name = "average",
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

subjects_ids	Character or numeric vector. freesurfer subjects ids to average Required.
args	Character. Additional parameters to the command
out_name	Character; file path. name for the average subject
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_make_surfaces
FREESURFER MakeSurfaces

Description

This program positions the tessellation of the cortical surface at the

Usage

```
ni_freesurfer_make_surfaces(
  hemisphere,
  in_filled,
  in_orig,
  in_wm,
  subject_id,
  args = NULL,
  fix_mtl = NULL,
  in_T1 = NULL,
  in_aseg = NULL,
  longitudinal = NULL,
  maximum = NULL,
  mgz = NULL,
  no_white = NULL,
  noaparc = NULL,
  orig_pial = NULL,
  orig_white = NULL,
  white = NULL,
  white_only = NULL,
```

```

    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

hemisphere	Character; one of: "lh", "rh". Hemisphere being processed Required.
in_filled	Character; file path. Implicit input file filled.mgz Required.
in_orig	Character; file path. Implicit input file .orig Required.
in_wm	Character; file path. Implicit input file wm.mgz Required.
subject_id	Character. Subject being processed Required.
args	Character. Additional parameters to the command
fix_mtl	Logical. Undocumented flag
in_T1	Character; file path. Input brain or T1 file
in_aseg	Character; file path. Input segmentation file
longitudinal	Logical. No documentation (used for longitudinal processing)
maximum	Numeric. No documentation (used for longitudinal processing)
mgz	Logical. No documentation. Direct questions to analysis-bugs@nmr.mgh.harvard.edu
no_white	Logical. Undocumented flag
noaparc	Logical. No documentation. Direct questions to analysis-bugs@nmr.mgh.harvard.edu
orig_pial	Character; file path. Specify a pial surface to start with
orig_white	Character; file path. Specify a white surface to start with
white	Character. White surface name
white_only	Logical. Undocumented flag
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

 ni_freesurfer_mni_bias_correction

FREESURFER MNIBiasCorrection

Description

Wrapper for nu_correct, a program from the Montreal Neurological Institute (MNI)

Usage

```
ni_freesurfer_mni_bias_correction(
  in_file,
  args = NULL,
  distance = NULL,
  iterations = 4,
  mask = NULL,
  no_rescale = NULL,
  out_file = NULL,
  protocol_iterations = NULL,
  shrink = NULL,
  stop = NULL,
  transform = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. input volume. Input can be any format accepted by mri_convert. Required.
args	Character. Additional parameters to the command
distance	Integer. N3 -distance option
iterations	Integer. Number of iterations to run nu_correct. Default is 4. This is the number of times that nu_correct is repeated (ie, using the output from the previous run as the input for the next). This is different than the -iterations option to nu_correct.
mask	Character; file path. brainmask volume. Input can be any format accepted by mri_convert.
no_rescale	Logical. do not rescale so that global mean of output == input global mean
out_file	Character; file path. output volume. Output can be any format accepted by mri_convert. If the output format is COR, then the directory must exist.

protocol_iterations	Integer. Passes Np as argument of the -iterations flag of nu_correct. This is different than the -n flag above. Default is not to pass nu_correct the -iterations flag.
shrink	Integer. Shrink parameter for finer sampling (default is 4)
stop	Numeric. Convergence threshold below which iteration stops (suggest 0.01 to 0.0001)
transform	Character; file path. tal.xfm. Use mri_make_uchar instead of conforming
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_mp_rto_mni305
FREESURFER MPRtoMNI305

Description

For complete details, see FreeSurfer documentation

Usage

```
ni_freesurfer_mp_rto_mni305(
  reference_dir,
  target,
  args = NULL,
  in_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

reference_dir	Character; directory path. Path to FreeSurfer's MNI305 reference directory (commonly \$FREESURFER_HOME/average). Required.
target	Character. input atlas file Required.
args	Character. Additional parameters to the command
in_file	Character; file path. the input file prefix for MPRtoMNI305
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

`ni_freesurfer_mr_is_ca_label`
FREESURFER MRIsCALabel

Description

For a single subject, produces an annotation file, in which each

Usage

```
ni_freesurfer_mr_is_ca_label(
  canonsurf,
  classifier,
  curv,
  hemisphere,
  smoothwm,
  subject_id,
  sulc,
  args = NULL,
  aseg = NULL,
  label = NULL,
  out_file = NULL,
  seed = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
```

```

    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

canonsurf	Character; file path. Input canonical surface file Required.
classifier	Character; file path. Classifier array input file Required.
curv	Character; file path. implicit input {hemisphere}.curv Required.
hemisphere	Character; one of: "lh", "rh". Hemisphere ('lh' or 'rh') Required.
smoothwm	Character; file path. implicit input {hemisphere}.smoothwm Required.
subject_id	Character. Subject name or ID Required.
sulc	Character; file path. implicit input {hemisphere}.sulc Required.
args	Character. Additional parameters to the command
aseg	Character; file path. Undocumented flag. Autorecon3 uses ../mri/aseg.presurf.mgz as input file
label	Character; file path. Undocumented flag. Autorecon3 uses ../label/{hemisphere}.cortex.label as input file
out_file	Character; file path. Annotated surface output file
seed	Integer
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_mr_is_calc

FREESURFER MRIsCalc

Description

'mris_calc' is a simple calculator that operates on FreeSurfer

Usage

```
ni_freesurfer_mr_is_calc(  
  action,  
  in_file1,  
  out_file,  
  args = NULL,  
  in_file2 = NULL,  
  in_float = NULL,  
  in_int = NULL,  
  .cwd = NULL,  
  .env = NULL,  
  .engine = NULL,  
  .profile = NULL,  
  dry_run = FALSE,  
  echo = interactive()  
)
```

Arguments

action	Character. Action to perform on input file(s) Required.
in_file1	Character; file path. Input file 1 Required.
out_file	Character; file path. Output file after calculation Required.
args	Character. Additional parameters to the command
in_file2	Character; file path. Input file 2
in_float	Numeric. Input float
in_int	Integer. Input integer
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

 ni_freesurfer_mr_is_combine

FREESURFER MRIsCombine

Description

Uses Freesurfer's `mrconvert` to combine two surface files into one.

Usage

```
ni_freesurfer_mr_is_combine(
  in_files,
  out_file,
  args = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

<code>in_files</code>	Character or numeric vector. Two surfaces to be combined. Required.
<code>out_file</code>	Character; file path. Output filename. Combined surfaces from <code>in_files</code> . Required.
<code>args</code>	Character. Additional parameters to the command
<code>.cwd</code>	Working directory override.
<code>.env</code>	Named character vector of environment variables.
<code>.engine</code>	Execution engine override.
<code>.profile</code>	Runtime profile override.
<code>dry_run</code>	Logical; preview command without executing.
<code>echo</code>	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

```
ni_freesurfer_mr_is_convert
    FREESURFER MRIsConvert
```

Description

Uses Freesurfer's `mr_is_convert` to convert surface files to various formats

Usage

```
ni_freesurfer_mr_is_convert(
    in_file,
    out_datatype,
    out_file,
    annot_file = NULL,
    args = NULL,
    dataarray_num = NULL,
    functional_file = NULL,
    label_file = NULL,
    labelstats_outfile = NULL,
    normal = NULL,
    origname = NULL,
    parcstats_file = NULL,
    patch = NULL,
    rescale = NULL,
    scalarcurv_file = NULL,
    scale = NULL,
    talairachxfm_subjid = NULL,
    to_scanner = NULL,
    to_tkr = NULL,
    vertex = NULL,
    xyz_ascii = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)
```

Arguments

<code>in_file</code>	Character; file path. File to read/convert Required.
<code>out_datatype</code>	Character; one of: "asc", "ico", "tri", "stl", "vtk", "gii", "mgh", "mgz". These file formats are supported: ASCII: .asc ICO: .ico .tri GEO: .geo STL: .stl VTK: .vtk GIFTI: .gii MGH surface-encoded 'volume': .mgh, .mgz Required.

out_file	Character; file path. output filename or True to generate one Required.
annot_file	Character; file path. input is annotation or gifti label data
args	Character. Additional parameters to the command
dataarray_num	Integer. if input is gifti, 'num' specifies which data array to use
functional_file	Character; file path. input is functional time-series or other multi-frame data (must specify surface)
label_file	Character; file path. infile is .label file, label is name of this label
labelstats_outfile	Character; file path. outfile is name of gifti file to which label stats will be written
normal	Logical. output is an ascii file where vertex data
origname	Character. read orig positions
parcstats_file	Character; file path. infile is name of text file containing label/val pairs
patch	Logical. input is a patch, not a full surface
rescale	Logical. rescale vertex xyz so total area is same as group average
scalarcurv_file	Character; file path. input is scalar curv overlay file (must still specify surface)
scale	Numeric. scale vertex xyz by scale
talairachxfm_subjid	Character. apply talairach xfm of subject to vertex xyz
to_scanner	Logical. convert coordinates from native FS (tkr) coords to scanner coords
to_tkr	Logical. convert coordinates from scanner coords to native FS (tkr) coords
vertex	Logical. Writes out neighbors of a vertex in each row
xyz_ascii	Logical. Print only surface xyz to ascii file
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

```
ni_freesurfer_mr_is_expand
    FREESURFER MRIsExpand
```

Description

Expands a surface (typically ?h.white) outwards while maintaining

Usage

```
ni_freesurfer_mr_is_expand(
    distance,
    in_file,
    args = NULL,
    dt = NULL,
    nsurfaces = NULL,
    out_name = "expanded",
    pial = NULL,
    smooth_averages = NULL,
    spring = NULL,
    thickness = NULL,
    thickness_name = NULL,
    write_iterations = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)
```

Arguments

distance	Numeric. Distance in mm or fraction of cortical thickness Required.
in_file	Character; file path. Surface to expand Required.
args	Character. Additional parameters to the command
dt	Numeric. dt (implicit: 0.25)
nsurfaces	Integer. Number of surfaces to write during expansion
out_name	Character. Output surface file. If no path, uses directory of in_file. If no path AND missing "lh." or "rh.", derive from in_file
pial	Character. Name of pial file (implicit: "pial") If no path, uses directory of in_file If no path AND missing "lh." or "rh.", derive from in_file
smooth_averages	Integer. Smooth surface with N iterations after expansion
spring	Numeric. Spring term (implicit: 0.05)

thickness	Logical. Expand by fraction of cortical thickness, not mm
thickness_name	Character. Name of thickness file (implicit: "thickness") If no path, uses directory of in_file If no path AND missing "lh." or "rh.", derive from in_file
write_iterations	Integer. Write snapshots of expansion every N iterations
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_mr_is_inflate
FREESURFER MRIsInflate

Description

This program will inflate a cortical surface.

Usage

```
ni_freesurfer_mr_is_inflate(
  in_file,
  args = NULL,
  no_save_sulc = NULL,
  out_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. Input file for MRIsInflate Required.
args	Character. Additional parameters to the command
no_save_sulc	Logical. Do not save sulc file as output

out_file	Character; file path. Output file for MRIsInflate
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_mri_convert
FREESURFER MRIConvert

Description

use fs mri_convert to manipulate files

Usage

```
ni_freesurfer_mri_convert(
  in_file,
  apply_inv_transform = NULL,
  apply_transform = NULL,
  args = NULL,
  ascii = NULL,
  autoalign_matrix = NULL,
  color_file = NULL,
  conform = NULL,
  conform_min = NULL,
  conform_size = NULL,
  crop_center = NULL,
  crop_gdf = NULL,
  crop_size = NULL,
  cut_ends = NULL,
  cw256 = NULL,
  devolve_transform = NULL,
  drop_n = NULL,
  fill_parcellation = NULL,
  force_ras = NULL,
  frame = NULL,
  frame_subsample = NULL,
  fwhm = NULL,
```

```
in_center = NULL,  
in_i_dir = NULL,  
in_i_size = NULL,  
in_info = NULL,  
in_j_dir = NULL,  
in_j_size = NULL,  
in_k_dir = NULL,  
in_k_size = NULL,  
in_like = NULL,  
in_matrix = NULL,  
in_orientation = NULL,  
in_scale = NULL,  
in_stats = NULL,  
in_type = NULL,  
invert_contrast = NULL,  
midframe = NULL,  
no_change = NULL,  
no_scale = NULL,  
no_translate = NULL,  
no_write = NULL,  
out_center = NULL,  
out_datatype = NULL,  
out_file = NULL,  
out_i_count = NULL,  
out_i_dir = NULL,  
out_i_size = NULL,  
out_info = NULL,  
out_j_count = NULL,  
out_j_dir = NULL,  
out_j_size = NULL,  
out_k_count = NULL,  
out_k_dir = NULL,  
out_k_size = NULL,  
out_matrix = NULL,  
out_orientation = NULL,  
out_scale = NULL,  
out_stats = NULL,  
out_type = NULL,  
parse_only = NULL,  
read_only = NULL,  
reorder = NULL,  
resample_type = NULL,  
reslice_like = NULL,  
sdcm_list = NULL,  
skip_n = NULL,  
slice_bias = NULL,  
slice_crop = NULL,  
slice_reverse = NULL,
```

```

smooth_parcellation = NULL,
sphinx = NULL,
split = NULL,
status_file = NULL,
subject_name = NULL,
te = NULL,
template_info = NULL,
template_type = NULL,
ti = NULL,
tr = NULL,
unwarp_gradient = NULL,
vox_size = NULL,
zero_ge_z_offset = NULL,
zero_outlines = NULL,
.cwd = NULL,
.env = NULL,
.engine = NULL,
.profile = NULL,
dry_run = FALSE,
echo = interactive()
)

```

Arguments

in_file	Character; file path. File to read/convert Required.
apply_inv_transform	Character; file path. apply inverse transformation xfm file
apply_transform	Character; file path. apply xfm file
args	Character. Additional parameters to the command
ascii	Logical. save output as ascii col>row>slice>frame
autoalign_matrix	Character; file path. text file with autoalign matrix
color_file	Character; file path. color file
conform	Logical. conform to 1mm voxel size in coronal slice direction with 256^3 or more
conform_min	Logical. conform to smallest size
conform_size	Numeric. conform to size_in_mm
crop_center	Character or numeric vector. crop to 256 around center (x, y, z)
crop_gdf	Logical. apply GDF cropping
crop_size	Character or numeric vector. crop to size <dx, dy, dz>
cut_ends	Integer. remove ncut slices from the ends
cw256	Logical. conform to dimensions of 256^3
devolve_transform	Character. subject id

drop_n	Integer. drop the last n frames
fill_parcellation	Logical. fill parcellation
force_ras	Logical. use default when orientation info absent
frame	Integer. keep only 0-based frame number
frame_subsample	Character or numeric vector. start delta end : frame subsampling (end = -1 for end)
fwhm	Numeric. smooth input volume by fwhm mm
in_center	Character or numeric vector.
in_i_dir	Character or numeric vector.
in_i_size	Integer. input i size
in_info	Logical. display input info
in_j_dir	Character or numeric vector.
in_j_size	Integer. input j size
in_k_dir	Character or numeric vector.
in_k_size	Integer. input k size
in_like	Character; file path. input looks like
in_matrix	Logical. display input matrix
in_orientation	Character; one of: "LAI", "LIA", "ALI", "AIL", "ILA", "IAL", "LAS", "LSA", "ALS", "ASL", "SLA", "SAL", "LPI", "LIP", "PLI", "PIL", "ILP", "IPL", "LPS", "LSP", "PLS", "PSL", "SLP", "SPL", "RAI", "RIA", "ARI", "AIR", "IRA", "IAR", "RAS", "RSA", "ARS", "ASR", "SRA", "SAR", "RPI", "RIP", "PRI", "PIR", "IRP", "IPR", "RPS", "RSP", "PRS", "PSR", "SRP", "SPR". specify the input orientation
in_scale	Numeric. input intensity scale factor
in_stats	Logical. display input stats
in_type	Character; one of: "cor", "mgh", "mgz", "minc", "analyze", "analyze4d", "spm", "afni", "brik", "bshort", "bfloat", "sdt", "outline", "otl", "gdf", "nifti1", "nii", "niigz", "ge", "gelx", "lx", "ximg", "siemens", "dicom", "siemens_dicom". input file type
invert_contrast	Numeric. threshold for inversting contrast
midframe	Logical. keep only the middle frame
no_change	Logical. don't change type of input to that of template
no_scale	Logical. dont rescale values for COR
no_translate	Logical. ???
no_write	Logical. do not write output
out_center	Character or numeric vector.
out_datatype	Character; one of: "uchar", "short", "int", "float". output data type <uchar short int float>

out_file	Character; file path. output filename or True to generate one
out_i_count	Integer. some count ?? in i direction
out_i_dir	Character or numeric vector.
out_i_size	Integer. output i size
out_info	Logical. display output info
out_j_count	Integer. some count ?? in j direction
out_j_dir	Character or numeric vector.
out_j_size	Integer. output j size
out_k_count	Integer. some count ?? in k direction
out_k_dir	Character or numeric vector.
out_k_size	Integer. output k size
out_matrix	Logical. display output matrix
out_orientation	Character; one of: "LAI", "LIA", "ALI", "AIL", "ILA", "IAL", "LAS", "LSA", "ALS", "ASL", "SLA", "SAL", "LPI", "LIP", "PLI", "PIL", "ILP", "IPL", "LPS", "LSP", "PLS", "PSL", "SLP", "SPL", "RAI", "RIA", "ARI", "AIR", "IRA", "IAR", "RAS", "RSA", "ARS", "ASR", "SRA", "SAR", "RPI", "RIP", "PRI", "PIR", "IRP", "IPR", "RPS", "RSP", "PRS", "PSR", "SRP", "SPR". specify the output orientation
out_scale	Numeric. output intensity scale factor
out_stats	Logical. display output stats
out_type	Character; one of: "cor", "mgh", "mgz", "minc", "analyze", "analyze4d", "spm", "afni", "brik", "bshort", "bfloat", "sdt", "outline", "otl", "gdf", "nifti1", "nii", "niigz". output file type
parse_only	Logical. parse input only
read_only	Logical. read the input volume
reorder	Character or numeric vector. olddim1 olddim2 olddim3
resample_type	Character; one of: "interpolate", "weighted", "nearest", "sinc", "cubic". <interpolatelweightedlnearestlsinlcubic> (default is interpolate)
reslice_like	Character; file path. reslice output to match file
sdc_m_list	Character; file path. list of DICOM files for conversion
skip_n	Integer. skip the first n frames
slice_bias	Numeric. apply half-cosine bias field
slice_crop	Character or numeric vector. s_start s_end : keep slices s_start to s_end
slice_reverse	Logical. reverse order of slices, update vox2ras
smooth_parcellation	Logical. smooth parcellation
sphinx	Logical. change orientation info to sphinx
split	Logical. split output frames into separate output files.
status_file	Character; file path. status file for DICOM conversion

subject_name	Character. subject name ???
te	Integer. TE in msec
template_info	Logical. dump info about template
template_type	Character; one of: "cor", "mgh", "mgz", "minc", "analyze", "analyze4d", "spm", "afni", "brik", "bshort", "bfloat", "sdt", "outline", "otl", "gdf", "nifti1", "nii", "niigz", "ge", "gelx", "lx", "ximg", "siemens", "dicom", "siemens_dicom". template file type
ti	Integer. TI in msec (note upper case flag)
tr	Integer. TR in msec
unwarp_gradient	Logical. unwarp gradient nonlinearity
vox_size	Character or numeric vector. <size_x> <size_y> <size_z> specify the size (mm) - useful for upsampling or downsampling
zero_ge_z_offset	Logical. zero ge z offset ???
zero_outlines	Logical. zero outlines
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_mri_coreg

FREESURFER MRICoreg

Description

This program registers one volume to another

Usage

```
ni_freesurfer_mri_coreg(
  reference_file,
  source_file,
  subject_id,
  args = NULL,
  brute_force_limit = NULL,
```

```

brute_force_samples = NULL,
conform_reference = NULL,
dof = NULL,
ftol = NULL,
initial_rotation = NULL,
initial_scale = NULL,
initial_shear = NULL,
initial_translation = NULL,
linmintol = NULL,
max_iters = NULL,
no_brute_force = NULL,
no_coord_dithering = NULL,
no_cras0 = NULL,
no_intensity_dithering = NULL,
no_smooth = NULL,
num_threads = NULL,
out_lta_file = TRUE,
out_params_file = NULL,
out_reg_file = NULL,
ref_fwhm = NULL,
reference_mask = NULL,
saturation_threshold = NULL,
sep = NULL,
source_mask = NULL,
source_oob = NULL,
subjects_dir = NULL,
.cwd = NULL,
.env = NULL,
.engine = NULL,
.profile = NULL,
dry_run = FALSE,
echo = interactive()
)

```

Arguments

reference_file	Character; file path. reference (target) file Required.
source_file	Character; file path. source file to be registered Required.
subject_id	Character. freesurfer subject ID (implies reference_mask == aparc+aseg.mgz unless otherwise specified) Required.
args	Character. Additional parameters to the command
brute_force_limit	Numeric. constrain brute force search to +/- lim
brute_force_samples	Integer. number of samples in brute force search
conform_reference	Logical. conform reference without rescaling

dof	Character; one of: "6", "9", "12". number of transform degrees of freedom
ftol	Numeric. floating-point tolerance (default=1e-7)
initial_rotation	Character or numeric vector. initial rotation in degrees
initial_scale	Character or numeric vector. initial scale
initial_shear	Character or numeric vector. initial shear (Hxy, Hxz, Hyz)
initial_translation	Character or numeric vector. initial translation in mm (implies no_cras0)
linmintol	Numeric
max_iters	Character. maximum iterations (default: 4)
no_brute_force	Logical. do not brute force search
no_coord_dithering	Logical. turn off coordinate dithering
no_cras0	Logical. do not set translation parameters to align centers of source and reference files
no_intensity_dithering	Logical. turn off intensity dithering
no_smooth	Logical. do not apply smoothing to either reference or source file
num_threads	Integer. number of OpenMP threads
out_lta_file	Character or numeric vector. output registration file (LTA format)
out_params_file	Character or numeric vector. output parameters file
out_reg_file	Character or numeric vector. output registration file (REG format)
ref_fwhm	Numeric. apply smoothing to reference file
reference_mask	Character or numeric vector. mask reference volume with given mask, or None if False
saturation_threshold	Character. saturation threshold (default=9.999)
sep	Character or numeric vector. set spatial scales, in voxels (default [2, 4])
source_mask	Character. mask source file with given mask
source_oob	Logical. count source voxels that are out-of-bounds as 0
subjects_dir	Character; directory path. FreeSurfer SUBJECTS_DIR
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

 ni_freesurfer_mri_fill

FREESURFER MRIFill

Description

This program creates hemispheric cutting planes and fills white matter

Usage

```
ni_freesurfer_mri_fill(
  in_file,
  out_file,
  args = NULL,
  log_file = NULL,
  segmentation = NULL,
  transform = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. Input white matter file Required.
out_file	Character; file path. Output filled volume file name for MRIFill Required.
args	Character. Additional parameters to the command
log_file	Character; file path. Output log file for MRIFill
segmentation	Character; file path. Input segmentation file for MRIFill
transform	Character; file path. Input transform file for MRIFill
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

```
ni_freesurfer_mri_marching_cubes
    FREESURFER MRIMarchingCubes
```

Description

Uses Freesurfer's mri_mc to create surfaces by tessellating a given input volume

Usage

```
ni_freesurfer_mri_marching_cubes(
    in_file,
    label_value,
    args = NULL,
    connectivity_value = 1,
    out_file = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)
```

Arguments

in_file	Character; file path. Input volume to tessellate voxels from. Required.
label_value	Integer. Label value which to tessellate from the input volume. (integer, if input is "filled.mgz" volume, 127 is rh, 255 is lh) Required.
args	Character. Additional parameters to the command
connectivity_value	Integer. Alter the marching cubes connectivity: 1=6+,2=18,3=6,4=26 (default=1)
out_file	Character; file path. output filename or True to generate one
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

 ni_freesurfer_mri_preteess

FREESURFER MRIPreteess

Description

Uses Freesurfer's mri_preteess to prepare volumes to be tessellated.

Usage

```
ni_freesurfer_mri_preteess(
  in_filled,
  in_norm,
  label,
  args = NULL,
  keep = NULL,
  nocorners = NULL,
  out_file = NULL,
  test = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_filled	Character; file path. filled volume, usually wm.mgz Required.
in_norm	Character; file path. the normalized, brain-extracted T1w image. Usually norm.mgz Required.
label	Character or numeric vector. label to be picked up, can be a Freesurfer's string like 'wm' or a label value (e.g. 127 for rh or 255 for lh) Required.
args	Character. Additional parameters to the command
keep	Logical. keep WM edits
nocorners	Logical. do not remove corner configurations in addition to edge ones.
out_file	Character; file path. the output file after mri_preteess.
test	Logical. adds a voxel that should be removed by mri_preteess. The value of the voxel is set to that of an ON-edited WM, so it should be kept with -keep. The output will NOT be saved.
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.

.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_mri_tessellate
FREESURFER MRITessellate

Description

Uses Freesurfer's mri_tessellate to create surfaces by tessellating a given input volume

Usage

```
ni_freesurfer_mri_tessellate(
    in_file,
    label_value,
    args = NULL,
    out_file = NULL,
    tessellate_all_voxels = NULL,
    use_real_RAS_coordinates = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)
```

Arguments

in_file	Character; file path. Input volume to tessellate voxels from. Required.
label_value	Integer. Label value which to tessellate from the input volume. (integer, if input is "filled.mgz" volume, 127 is rh, 255 is lh) Required.
args	Character. Additional parameters to the command
out_file	Character; file path. output filename or True to generate one
tessellate_all_voxels	Logical. Tessellate the surface of all voxels with different labels
use_real_RAS_coordinates	Logical. Saves surface with real RAS coordinates where c_(r,a,s) != 0
.cwd	Working directory override.

.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_mris_preproc
FREESURFER MRISPreproc

Description

Use FreeSurfer mris_preproc to prepare a group of contrasts for

Usage

```
ni_freesurfer_mris_preproc(
  hemi,
  target,
  args = NULL,
  fsgd_file = NULL,
  fwhm = NULL,
  fwhm_source = NULL,
  num_iters = NULL,
  num_iters_source = NULL,
  out_file = NULL,
  proj_frac = NULL,
  smooth_cortex_only = NULL,
  source_format = NULL,
  subject_file = NULL,
  subjects = NULL,
  surf_area = NULL,
  surf_dir = NULL,
  surf_measure = NULL,
  surf_measure_file = NULL,
  vol_measure_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

hemi	Character; one of: "lh", "rh". hemisphere for source and target Required.
target	Character. target subject name Required.
args	Character. Additional parameters to the command
fsgd_file	Character; file path. specify subjects using fsgd file
fwhm	Numeric. smooth by fwhm mm on the target surface
fwhm_source	Numeric. smooth by fwhm mm on the source surface
num_iters	Integer. niters : smooth by niters on the target surface
num_iters_source	Integer. niters : smooth by niters on the source surface
out_file	Character; file path. output filename
proj_frac	Numeric. projection fraction for vol2surf
smooth_cortex_only	Logical. only smooth cortex (ie, exclude medial wall)
source_format	Character. source format
subject_file	Character; file path. file specifying subjects separated by white space
subjects	Character or numeric vector. subjects from who measures are calculated
surf_area	Character. Extract vertex area from subject/surf/hemi.surfname to use as input.
surf_dir	Character. alternative directory (instead of surf)
surf_measure	Character. Use subject/surf/hemi.surf_measure as input
surf_measure_file	Character or numeric vector. file alternative to surfmeas, still requires list of subjects
vol_measure_file	Character or numeric vector. list of volume measure and reg file tuples
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

 ni_freesurfer_mris_preproc_recon_all

FREESURFER MRISPreprocReconAll

Description

Extends MRISPreproc to allow it to be used in a recon-all workflow

Usage

```
ni_freesurfer_mris_preproc_recon_all(
  hemi,
  target,
  args = NULL,
  fsgd_file = NULL,
  fwhm = NULL,
  fwhm_source = NULL,
  num_iters = NULL,
  num_iters_source = NULL,
  out_file = NULL,
  proj_frac = NULL,
  smooth_cortex_only = NULL,
  source_format = NULL,
  subject_file = NULL,
  subject_id = "subject_id",
  subjects = NULL,
  surf_area = NULL,
  surf_dir = NULL,
  surf_measure = NULL,
  surf_measure_file = NULL,
  surfreg_files = NULL,
  vol_measure_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

hemi	Character; one of: "lh", "rh". hemisphere for source and target Required.
target	Character. target subject name Required.
args	Character. Additional parameters to the command
fsgd_file	Character; file path. specify subjects using fsgd file

fwhm	Numeric. smooth by fwhm mm on the target surface
fwhm_source	Numeric. smooth by fwhm mm on the source surface
num_iters	Integer. niters : smooth by niters on the target surface
num_iters_source	Integer. niters : smooth by niters on the source surface
out_file	Character; file path. output filename
proj_frac	Numeric. projection fraction for vol2surf
smooth_cortex_only	Logical. only smooth cortex (ie, exclude medial wall)
source_format	Character. source format
subject_file	Character; file path. file specifying subjects separated by white space
subject_id	Character. subject from whom measures are calculated
subjects	Character or numeric vector. subjects from who measures are calculated
surf_area	Character. Extract vertex area from subject/surf/hemi.surfname to use as input.
surf_dir	Character. alternative directory (instead of surf)
surf_measure	Character. Use subject/surf/hemi.surf_measure as input
surf_measure_file	Character; file path. file necessary for surfmeas
surfreg_files	Character or numeric vector. lh and rh input surface registration files
vol_measure_file	Character or numeric vector. list of volume measure and reg file tuples
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_mrtm1 *FREESURFER MRTM1*

Description

Perform MRTM1 kinetic modeling.

Usage

```
ni_freesurfer_mrtm1(  
  in_file,  
  mrtm1,  
  allow_ill_cond = NULL,  
  allow_repeated_subjects = NULL,  
  args = NULL,  
  bp_clip_max = NULL,  
  bp_clip_neg = NULL,  
  calc_AR1 = NULL,  
  check_opts = NULL,  
  compute_log_y = NULL,  
  contrast = NULL,  
  cortex = NULL,  
  debug = NULL,  
  design = NULL,  
  diag = NULL,  
  diag_cluster = NULL,  
  fixed_fx_dof = NULL,  
  fixed_fx_dof_file = NULL,  
  fixed_fx_var = NULL,  
  force_perm = NULL,  
  fsgd = NULL,  
  fwhm = NULL,  
  glm_dir = NULL,  
  invert_mask = NULL,  
  label_file = NULL,  
  logan = NULL,  
  mask_file = NULL,  
  mrtm2 = NULL,  
  nii = NULL,  
  nii_gz = NULL,  
  no_contrast_ok = NULL,  
  no_est_fwhm = NULL,  
  no_mask_smooth = NULL,  
  no_prune = NULL,  
  one_sample = NULL,  
  pca = NULL,  
  per_voxel_reg = NULL,
```

```

profile = NULL,
prune = NULL,
prune_thresh = NULL,
resynth_test = NULL,
save_cond = NULL,
save_estimate = NULL,
save_res_corr_mtx = NULL,
save_residual = NULL,
seed = NULL,
self_reg = NULL,
sim_done_file = NULL,
sim_sign = NULL,
simulation = NULL,
surf = NULL,
synth = NULL,
uniform = NULL,
var_fwhm = NULL,
vox_dump = NULL,
weight_inv = NULL,
weight_sqrt = NULL,
weighted_ls = NULL,
.cwd = NULL,
.env = NULL,
.engine = NULL,
.profile = NULL,
dry_run = FALSE,
echo = interactive()
)

```

Arguments

in_file	Character; file path. input 4D file Required.
mrtm1	Character or numeric vector. RefTac TimeSec : perform MRTM1 kinetic modeling Required.
allow_ill_cond	Logical. allow ill-conditioned design matrices
allow_repeated_subjects	Logical. allow subject names to repeat in the fsgd file (must appear before -fsgd)
args	Character. Additional parameters to the command
bp_clip_max	Numeric. set BP voxels above max to max
bp_clip_neg	Logical. set negative BP voxels to zero
calc_AR1	Logical. compute and save temporal AR1 of residual
check_opts	Logical. don't run anything, just check options and exit
compute_log_y	Logical. compute natural log of y prior to analysis
contrast	Character or numeric vector. contrast file
cortex	Logical. use subjects ?h.cortex.label as label

debug	Logical. turn on debugging
design	Character; file path. design matrix file
diag	Integer. Gdiag_no : set diagnostic level
diag_cluster	Logical. save sig volume and exit from first sim loop
fixed_fx_dof	Integer. dof for fixed effects analysis
fixed_fx_dof_file	Character; file path. text file with dof for fixed effects analysis
fixed_fx_var	Character; file path. for fixed effects analysis
force_perm	Logical. force permutation test, even when design matrix is not orthog
fsgd	Character or numeric vector. freesurfer descriptor file
fwhm	Character. smooth input by fwhm
glm_dir	Character. save outputs to dir
invert_mask	Logical. invert mask
label_file	Character; file path. use label as mask, surfaces only
logan	Character or numeric vector. RefTac TimeSec tstar : perform Logan kinetic modeling
mask_file	Character; file path. binary mask
mrtm2	Character or numeric vector. RefTac TimeSec k2prime : perform MRTM2 kinetic modeling
nii	Logical. save outputs as nii
nii_gz	Logical. save outputs as nii.gz
no_contrast_ok	Logical. do not fail if no contrasts specified
no_est_fwhm	Logical. turn off FWHM output estimation
no_mask_smooth	Logical. do not mask when smoothing
no_prune	Logical. do not prune
one_sample	Logical. construct X and C as a one-sample group mean
pca	Logical. perform pca/svd analysis on residual
per_voxel_reg	Character or numeric vector. per-voxel regressors
profile	Integer. niters : test speed
prune	Logical. remove voxels that do not have a non-zero value at each frame (def)
prune_thresh	Numeric. prune threshold. Default is FLT_MIN
resynth_test	Integer. test GLM by resynthesis
save_cond	Logical. flag to save design matrix condition at each voxel
save_estimate	Logical. save signal estimate (yhat)
save_res_corr_mtx	Logical. save residual error spatial correlation matrix (eres.scm). Big!
save_residual	Logical. save residual error (eres)
seed	Integer. used for synthesizing noise

self_reg	Character or numeric vector. self-regressor from index col row slice
sim_done_file	Character; file path. create file when simulation finished
sim_sign	Character; one of: "abs", "pos", "neg". abs, pos, or neg
simulation	Character or numeric vector. nulltype nsim thresh csdbasename
surf	Logical. analysis is on a surface mesh
synth	Logical. replace input with gaussian
uniform	Character or numeric vector. use uniform distribution instead of gaussian
var_fwhm	Character. smooth variance by fwhm
vox_dump	Character or numeric vector. dump voxel GLM and exit
weight_inv	Logical. invert weights
weight_sqrt	Logical. sqrt of weights
weighted_ls	Character; file path. weighted least squares
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_mrtm2 *FREESURFER MRTM2*

Description

Perform MRTM2 kinetic modeling.

Usage

```
ni_freesurfer_mrtm2(
  in_file,
  mrtm2,
  allow_ill_cond = NULL,
  allow_repeated_subjects = NULL,
  args = NULL,
  bp_clip_max = NULL,
  bp_clip_neg = NULL,
  calc_AR1 = NULL,
  check_opts = NULL,
```

```
compute_log_y = NULL,  
contrast = NULL,  
cortex = NULL,  
debug = NULL,  
design = NULL,  
diag = NULL,  
diag_cluster = NULL,  
fixed_fx_dof = NULL,  
fixed_fx_dof_file = NULL,  
fixed_fx_var = NULL,  
force_perm = NULL,  
fsgd = NULL,  
fwhm = NULL,  
glm_dir = NULL,  
invert_mask = NULL,  
label_file = NULL,  
logan = NULL,  
mask_file = NULL,  
mrtm1 = NULL,  
nii = NULL,  
nii_gz = NULL,  
no_contrast_ok = NULL,  
no_est_fwhm = NULL,  
no_mask_smooth = NULL,  
no_prune = NULL,  
one_sample = NULL,  
pca = NULL,  
per_voxel_reg = NULL,  
profile = NULL,  
prune = NULL,  
prune_thresh = NULL,  
resynth_test = NULL,  
save_cond = NULL,  
save_estimate = NULL,  
save_res_corr_mtx = NULL,  
save_residual = NULL,  
seed = NULL,  
self_reg = NULL,  
sim_done_file = NULL,  
sim_sign = NULL,  
simulation = NULL,  
surf = NULL,  
synth = NULL,  
uniform = NULL,  
var_fwhm = NULL,  
vox_dump = NULL,  
weight_inv = NULL,  
weight_sqrt = NULL,
```

```

    weighted_ls = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

in_file	Character; file path. input 4D file Required.
mrtm2	Character or numeric vector. RefTac TimeSec k2prime : perform MRTM2 kinetic modeling Required.
allow_ill_cond	Logical. allow ill-conditioned design matrices
allow_repeated_subjects	Logical. allow subject names to repeat in the fsgd file (must appear before -fsgd)
args	Character. Additional parameters to the command
bp_clip_max	Numeric. set BP voxels above max to max
bp_clip_neg	Logical. set negative BP voxels to zero
calc_AR1	Logical. compute and save temporal AR1 of residual
check_opts	Logical. don't run anything, just check options and exit
compute_log_y	Logical. compute natural log of y prior to analysis
contrast	Character or numeric vector. contrast file
cortex	Logical. use subjects ?h.cortex.label as label
debug	Logical. turn on debugging
design	Character; file path. design matrix file
diag	Integer. Gdiag_no : set diagnostic level
diag_cluster	Logical. save sig volume and exit from first sim loop
fixed_fx_dof	Integer. dof for fixed effects analysis
fixed_fx_dof_file	Character; file path. text file with dof for fixed effects analysis
fixed_fx_var	Character; file path. for fixed effects analysis
force_perm	Logical. force permutation test, even when design matrix is not orthog
fsgd	Character or numeric vector. freesurfer descriptor file
fwhm	Character. smooth input by fwhm
glm_dir	Character. save outputs to dir
invert_mask	Logical. invert mask
label_file	Character; file path. use label as mask, surfaces only
logan	Character or numeric vector. RefTac TimeSec tstar : perform Logan kinetic modeling

mask_file	Character; file path. binary mask
mrtm1	Character or numeric vector. RefTac TimeSec : perform MRTM1 kinetic modeling
nii	Logical. save outputs as nii
nii_gz	Logical. save outputs as nii.gz
no_contrast_ok	Logical. do not fail if no contrasts specified
no_est_fwhm	Logical. turn off FWHM output estimation
no_mask_smooth	Logical. do not mask when smoothing
no_prune	Logical. do not prune
one_sample	Logical. construct X and C as a one-sample group mean
pca	Logical. perform pca/svd analysis on residual
per_voxel_reg	Character or numeric vector. per-voxel regressors
profile	Integer. niters : test speed
prune	Logical. remove voxels that do not have a non-zero value at each frame (def)
prune_thresh	Numeric. prune threshold. Default is FLT_MIN
resynth_test	Integer. test GLM by resynthesis
save_cond	Logical. flag to save design matrix condition at each voxel
save_estimate	Logical. save signal estimate (yhat)
save_res_corr_mtx	Logical. save residual error spatial correlation matrix (eres.scm). Big!
save_residual	Logical. save residual error (eres)
seed	Integer. used for synthesizing noise
self_reg	Character or numeric vector. self-regressor from index col row slice
sim_done_file	Character; file path. create file when simulation finished
sim_sign	Character; one of: "abs", "pos", "neg". abs, pos, or neg
simulation	Character or numeric vector. nulltype nsim thresh csdbasename
surf	Logical. analysis is on a surface mesh
synth	Logical. replace input with gaussian
uniform	Character or numeric vector. use uniform distribution instead of gaussian
var_fwhm	Character. smooth variance by fwhm
vox_dump	Character or numeric vector. dump voxel GLM and exit
weight_inv	Logical. invert weights
weight_sqrt	Logical. sqrt of weights
weighted_ls	Character; file path. weighted least squares
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_ms_lda *FREESURFER MS_LDA*

Description

Perform LDA reduction on the intensity space of an arbitrary # of FLASH images

Usage

```
ni_freesurfer_ms_lda(
  images,
  lda_labels,
  vol_synth_file,
  weight_file,
  args = NULL,
  conform = NULL,
  label_file = NULL,
  mask_file = NULL,
  shift = NULL,
  use_weights = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

images	Character or numeric vector. list of input FLASH images Required.
lda_labels	Character or numeric vector. pair of class labels to optimize Required.
vol_synth_file	Character; file path. filename for the synthesized output volume Required.
weight_file	Character; file path. filename for the LDA weights (input or output) Required.
args	Character. Additional parameters to the command
conform	Logical. Conform the input volumes (brain mask typically already conformed)
label_file	Character; file path. filename of the label volume
mask_file	Character; file path. filename of the brain mask volume
shift	Integer. shift all values equal to the given value to zero
use_weights	Logical. Use the weights from a previously generated weight file
.cwd	Working directory override.

.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_normalize
FREESURFER Normalize

Description

Normalize the white-matter, optionally based on control points. The

Usage

```
ni_freesurfer_normalize(
  in_file,
  args = NULL,
  gradient = NULL,
  mask = NULL,
  out_file = NULL,
  segmentation = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. The input file for Normalize Required.
args	Character. Additional parameters to the command
gradient	Integer. use max intensity/mm gradient g (default=1)
mask	Character; file path. The input mask file for Normalize
out_file	Character; file path. The output file for Normalize
segmentation	Character; file path. The input segmentation for Normalize
.cwd	Working directory override.

.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

ni_freesurfer_one_sample_t_test
FREESURFER OneSampleTTest

Description

Use FreeSurfer's `mri_glmfit` to specify and estimate a general linear model.

Usage

```
ni_freesurfer_one_sample_t_test(
  in_file,
  allow_ill_cond = NULL,
  allow_repeated_subjects = NULL,
  args = NULL,
  bp_clip_max = NULL,
  bp_clip_neg = NULL,
  calc_AR1 = NULL,
  check_opts = NULL,
  compute_log_y = NULL,
  contrast = NULL,
  cortex = NULL,
  debug = NULL,
  design = NULL,
  diag = NULL,
  diag_cluster = NULL,
  fixed_fx_dof = NULL,
  fixed_fx_dof_file = NULL,
  fixed_fx_var = NULL,
  force_perm = NULL,
  fsgd = NULL,
  fwhm = NULL,
  glm_dir = NULL,
  invert_mask = NULL,
  label_file = NULL,
  logan = NULL,
```

```

mask_file = NULL,
mrtm1 = NULL,
mrtm2 = NULL,
nii = NULL,
nii_gz = NULL,
no_contrast_ok = NULL,
no_est_fwhm = NULL,
no_mask_smooth = NULL,
no_prune = NULL,
one_sample = TRUE,
pca = NULL,
per_voxel_reg = NULL,
profile = NULL,
prune = NULL,
prune_thresh = NULL,
resynth_test = NULL,
save_cond = NULL,
save_estimate = NULL,
save_res_corr_mtx = NULL,
save_residual = NULL,
seed = NULL,
self_reg = NULL,
sim_done_file = NULL,
sim_sign = NULL,
simulation = NULL,
surf = NULL,
synth = NULL,
uniform = NULL,
var_fwhm = NULL,
vox_dump = NULL,
weight_inv = NULL,
weight_sqrt = NULL,
weighted_ls = NULL,
.cwd = NULL,
.env = NULL,
.engine = NULL,
.profile = NULL,
dry_run = FALSE,
echo = interactive()
)

```

Arguments

in_file	Character; file path. input 4D file Required.
allow_ill_cond	Logical. allow ill-conditioned design matrices
allow_repeated_subjects	Logical. allow subject names to repeat in the fsgd file (must appear before -fsgd)
args	Character. Additional parameters to the command

bp_clip_max	Numeric. set BP voxels above max to max
bp_clip_neg	Logical. set negative BP voxels to zero
calc_AR1	Logical. compute and save temporal AR1 of residual
check_opts	Logical. don't run anything, just check options and exit
compute_log_y	Logical. compute natural log of y prior to analysis
contrast	Character or numeric vector. contrast file
cortex	Logical. use subjects ?h.cortex.label as label
debug	Logical. turn on debugging
design	Character; file path. design matrix file
diag	Integer. Gdiag_no : set diagnostic level
diag_cluster	Logical. save sig volume and exit from first sim loop
fixed_fx_dof	Integer. dof for fixed effects analysis
fixed_fx_dof_file	Character; file path. text file with dof for fixed effects analysis
fixed_fx_var	Character; file path. for fixed effects analysis
force_perm	Logical. force permutation test, even when design matrix is not orthog
fsgd	Character or numeric vector. freesurfer descriptor file
fwhm	Character. smooth input by fwhm
glm_dir	Character. save outputs to dir
invert_mask	Logical. invert mask
label_file	Character; file path. use label as mask, surfaces only
logan	Character or numeric vector. RefTac TimeSec tstar : perform Logan kinetic modeling
mask_file	Character; file path. binary mask
mrtm1	Character or numeric vector. RefTac TimeSec : perform MRTM1 kinetic modeling
mrtm2	Character or numeric vector. RefTac TimeSec k2prime : perform MRTM2 kinetic modeling
nii	Logical. save outputs as nii
nii_gz	Logical. save outputs as nii.gz
no_contrast_ok	Logical. do not fail if no contrasts specified
no_est_fwhm	Logical. turn off FWHM output estimation
no_mask_smooth	Logical. do not mask when smoothing
no_prune	Logical. do not prune
one_sample	Logical. construct X and C as a one-sample group mean
pca	Logical. perform pca/svd analysis on residual
per_voxel_reg	Character or numeric vector. per-voxel regressors
profile	Integer. niters : test speed

prune	Logical. remove voxels that do not have a non-zero value at each frame (def)
prune_thresh	Numeric. prune threshold. Default is FLT_MIN
resynth_test	Integer. test GLM by resynthesis
save_cond	Logical. flag to save design matrix condition at each voxel
save_estimate	Logical. save signal estimate (yhat)
save_res_corr_mtx	Logical. save residual error spatial correlation matrix (eres.scm). Big!
save_residual	Logical. save residual error (eres)
seed	Integer. used for synthesizing noise
self_reg	Character or numeric vector. self-regressor from index col row slice
sim_done_file	Character; file path. create file when simulation finished
sim_sign	Character; one of: "abs", "pos", "neg". abs, pos, or neg
simulation	Character or numeric vector. nulltype nsim thresh csdbasename
surf	Logical. analysis is on a surface mesh
synth	Logical. replace input with gaussian
uniform	Character or numeric vector. use uniform distribution instead of gaussian
var_fwhm	Character. smooth variance by fwhm
vox_dump	Character or numeric vector. dump voxel GLM and exit
weight_inv	Logical. invert weights
weight_sqrt	Logical. sqrt of weights
weighted_ls	Character; file path. weighted least squares
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

 ni_freesurfer_paint *FREESURFER Paint*

Description

This program is useful for extracting one of the arrays ("a variable")

Usage

```
ni_freesurfer_paint(
    in_surf,
    template,
    args = NULL,
    averages = NULL,
    out_file = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)
```

Arguments

in_surf	Character; file path. Surface file with grid (vertices) onto which the template data is to be sampled or 'painted' Required.
template	Character; file path. Template file Required.
args	Character. Additional parameters to the command
averages	Integer. Average curvature patterns
out_file	Character; file path. File containing a surface-worth of per-vertex values, saved in 'curvature' format.
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

```
ni_freesurfer_parcellation_stats
    FREESURFER ParcellationStats
```

Description

This program computes a number of anatomical properties.

Usage

```
ni_freesurfer_parcellation_stats(
    aseg,
    brainmask,
    hemisphere,
    lh_pial,
    lh_white,
    rh_pial,
    rh_white,
    ribbon,
    subject_id,
    thickness,
    transform,
    wm,
    args = NULL,
    in_annotation = NULL,
    in_cortex = NULL,
    in_label = NULL,
    mgz = NULL,
    out_color = NULL,
    out_table = NULL,
    surface = NULL,
    tabular_output = NULL,
    th3 = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)
```

Arguments

aseg	Character; file path. Input file must be <subject_id>/mri/aseg.presurf.mgz Required.
brainmask	Character; file path. Input file must be <subject_id>/mri/brainmask.mgz Required.

hemisphere	Character; one of: "lh", "rh". Hemisphere being processed Required.
lh_pial	Character; file path. Input file must be <subject_id>/surf/lh.pial Required.
lh_white	Character; file path. Input file must be <subject_id>/surf/lh.white Required.
rh_pial	Character; file path. Input file must be <subject_id>/surf/rh.pial Required.
rh_white	Character; file path. Input file must be <subject_id>/surf/rh.white Required.
ribbon	Character; file path. Input file must be <subject_id>/mri/ribbon.mgz Required.
subject_id	Character. Subject being processed Required.
thickness	Character; file path. Input file must be <subject_id>/surf/?h.thickness Required.
transform	Character; file path. Input file must be <subject_id>/mri/transforms/talairach.xfm Required.
wm	Character; file path. Input file must be <subject_id>/mri/wm.mgz Required.
args	Character. Additional parameters to the command
in_annotation	Character; file path. compute properties for each label in the annotation file separately
in_cortex	Character; file path. Input cortex label
in_label	Character; file path. limit calculations to specified label
mgz	Logical. Look for mgz files
out_color	Character; file path. Output annotation files's colortable to text file
out_table	Character; file path. Table output to tablefile
surface	Character. Input surface (e.g. 'white')
tabular_output	Logical. Tabular output
th3	Logical. turns on new vertex-wise volume calc for mris_anat_stats
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

```
ni_freesurfer_parse_dicom_dir
    FREESURFER ParseDICOMDir
```

Description

Uses mri_parse_sdcmdir to get information from dicom directories

Usage

```
ni_freesurfer_parse_dicom_dir(
    dicom_dir,
    args = NULL,
    dicom_info_file = "dicominfo.txt",
    sortbyrun = NULL,
    summarize = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)
```

Arguments

dicom_dir	Character; directory path. path to siemens dicom directory Required.
args	Character. Additional parameters to the command
dicom_info_file	Character; file path. file to which results are written
sortbyrun	Logical. assign run numbers
summarize	Logical. only print out info for run leaders
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

`ni_freesurfer_recon_all`*FREESURFER ReconAll*

Description

Uses recon-all to generate surfaces and parcellations of structural data

Usage

```
ni_freesurfer_recon_all(  
  FLAIR_file = NULL,  
  T1_files = NULL,  
  T2_file = NULL,  
  args = NULL,  
  base_template_id = NULL,  
  base_timepoint_ids = NULL,  
  big_ventricles = NULL,  
  brainstem = NULL,  
  directive = "all",  
  expert = NULL,  
  flags = NULL,  
  hemi = NULL,  
  hippocampal_subfields_T1 = NULL,  
  hippocampal_subfields_T2 = NULL,  
  hires = NULL,  
  longitudinal_template_id = NULL,  
  longitudinal_timepoint_id = NULL,  
  mprage = NULL,  
  openmp = NULL,  
  parallel = NULL,  
  subject_id = NULL,  
  subjects_dir = NULL,  
  use_FLAIR = NULL,  
  use_T2 = NULL,  
  xopts = NULL,  
  .cwd = NULL,  
  .env = NULL,  
  .engine = NULL,  
  .profile = NULL,  
  dry_run = FALSE,  
  echo = interactive()  
)
```

Arguments

FLAIR_file Character; file path. Convert FLAIR image to orig directory

T1_files	Character or numeric vector. name of T1 file to process
T2_file	Character; file path. Convert T2 image to orig directory
args	Character. Additional parameters to the command
base_template_id	Character. base template id
base_timepoint_ids	Character or numeric vector. processed timepoint to use in template
big_ventricles	Logical. For use in subjects with enlarged ventricles
brainstem	Logical. Segment brainstem structures
directive	Character; one of: "all", "autorecon1", "autorecon2", "autorecon2-volonly", "autorecon2-perhemi", "autorecon2-inflate1", "autorecon2-cp", "autorecon2-wm", "autorecon3", "autorecon3-T2pial", "autorecon-pial", "autorecon-hemi", "localGI", "qcache". process directive
expert	Character; file path. Set parameters using expert file
flags	Character or numeric vector. additional parameters
hemi	Character; one of: "lh", "rh". hemisphere to process
hippocampal_subfields_T1	Logical. segment hippocampal subfields using input T1 scan
hippocampal_subfields_T2	Character or numeric vector. segment hippocampal subfields using T2 scan, identified by ID (may be combined with hippocampal_subfields_T1)
hires	Logical. Conform to minimum voxel size (for voxels < 1mm)
longitudinal_template_id	Character. longitudinal base template id
longitudinal_timepoint_id	Character. longitudinal session/timepoint id
mprage	Logical. Assume scan parameters are MGH MP-RAGE protocol, which produces darker gray matter
openmp	Integer. Number of processors to use in parallel
parallel	Logical. Enable parallel execution
subject_id	Character. subject name
subjects_dir	Character; directory path. path to subjects directory
use_FLAIR	Logical. Use FLAIR image to refine the pial surface
use_T2	Logical. Use T2 image to refine the pial surface
xopts	Character; one of: "use", "clean", "overwrite". Use, delete or overwrite existing expert options file
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_register

FREESURFER Register

Description

This program registers a surface to an average surface template.

Usage

```
ni_freesurfer_register(
    in_sulc,
    in_surf,
    target,
    args = NULL,
    curv = NULL,
    out_file = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)
```

Arguments

in_sulc	Character; file path. Undocumented mandatory input file <code>\${SUBJECTS_DIR}/surf/{hemisphere}.sulc</code> Required.
in_surf	Character; file path. Surface to register, often <code>{hemi}.sphere</code> Required.
target	Character; file path. The data to register to. In normal recon-all usage, this is a template file for average surface. Required.
args	Character. Additional parameters to the command
curv	Logical. Use smoothwm curvature for final alignment
out_file	Character; file path. Output surface file to capture registration
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

```
ni_freesurfer_register_av_ito_talairach
    FREESURFER RegisterAVItoTalairach
```

Description

converts the vox2vox from talairach_avi to a talairach.xfm file

Usage

```
ni_freesurfer_register_av_ito_talairach(
    in_file,
    target,
    vox2vox,
    args = NULL,
    out_file = "talairach.auto.xfm",
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)
```

Arguments

in_file	Character; file path. The input file Required .
target	Character; file path. The target file Required .
vox2vox	Character; file path. The vox2vox file Required .
args	Character. Additional parameters to the command
out_file	Character; file path. The transform output
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

```
ni_freesurfer_relabel_hypointensities
    FREESURFER RelabelHypointensities
```

Description

Relabel Hypointensities

Usage

```
ni_freesurfer_relabel_hypointensities(
  aseg,
  lh_white,
  rh_white,
  args = NULL,
  out_file = NULL,
  surf_directory = ".",
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

aseg	Character; file path. Input aseg file Required.
lh_white	Character; file path. Implicit input file must be lh.white Required.
rh_white	Character; file path. Implicit input file must be rh.white Required.
args	Character. Additional parameters to the command
out_file	Character; file path. Output aseg file
surf_directory	Character; directory path. Directory containing lh.white and rh.white
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

`ni_freesurfer_remove_intersection`*FREESURFER RemoveIntersection*

Description

This program removes the intersection of the given MRI

Usage

```
ni_freesurfer_remove_intersection(  
    in_file,  
    args = NULL,  
    out_file = NULL,  
    .cwd = NULL,  
    .env = NULL,  
    .engine = NULL,  
    .profile = NULL,  
    dry_run = FALSE,  
    echo = interactive()  
)
```

Arguments

<code>in_file</code>	Character; file path. Input file for RemoveIntersection Required.
<code>args</code>	Character. Additional parameters to the command
<code>out_file</code>	Character; file path. Output file for RemoveIntersection
<code>.cwd</code>	Working directory override.
<code>.env</code>	Named character vector of environment variables.
<code>.engine</code>	Execution engine override.
<code>.profile</code>	Runtime profile override.
<code>dry_run</code>	Logical; preview command without executing.
<code>echo</code>	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

```
ni_freesurfer_remove_neck
    FREESURFER RemoveNeck
```

Description

Crops the neck out of the mri image

Usage

```
ni_freesurfer_remove_neck(
    in_file,
    template,
    transform,
    args = NULL,
    out_file = NULL,
    radius = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)
```

Arguments

in_file	Character; file path. Input file for RemoveNeck Required.
template	Character; file path. Input template file for RemoveNeck Required.
transform	Character; file path. Input transform file for RemoveNeck Required.
args	Character. Additional parameters to the command
out_file	Character; file path. Output file for RemoveNeck
radius	Integer. Radius
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

 ni_freesurfer_resample

FREESURFER Resample

Description

Use FreeSurfer mri_convert to up or down-sample image files

Usage

```
ni_freesurfer_resample(
  in_file,
  voxel_size,
  args = NULL,
  resampled_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. file to resample Required.
voxel_size	Character or numeric vector. triplet of output voxel sizes Required.
args	Character. Additional parameters to the command
resampled_file	Character; file path. output filename
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_robust_register
FREESURFER RobustRegister

Description

Perform intramodal linear registration (translation and rotation) using

Usage

```
ni_freesurfer_robust_register(  
    auto_sens,  
    outlier_sens,  
    source_file,  
    target_file,  
    args = NULL,  
    est_int_scale = NULL,  
    force_double = NULL,  
    force_float = NULL,  
    half_source = NULL,  
    half_source_xfm = NULL,  
    half_targ = NULL,  
    half_targ_xfm = NULL,  
    half_weights = NULL,  
    high_iterations = NULL,  
    in_xfm_file = NULL,  
    init_orient = NULL,  
    iteration_thresh = NULL,  
    least_squares = NULL,  
    mask_source = NULL,  
    mask_target = NULL,  
    max_iterations = NULL,  
    no_init = NULL,  
    no_multi = NULL,  
    out_reg_file = TRUE,  
    outlier_limit = NULL,  
    registered_file = NULL,  
    subsample_thresh = NULL,  
    trans_only = NULL,  
    weights_file = NULL,  
    write_vo2vox = NULL,  
    .cwd = NULL,  
    .env = NULL,  
    .engine = NULL,  
    .profile = NULL,  
    dry_run = FALSE,  
    echo = interactive())
```

)

Arguments

auto_sens	Logical. auto-detect good sensitivity Required.
outlier_sens	Numeric. set outlier sensitivity explicitly Required.
source_file	Character; file path. volume to be registered Required.
target_file	Character; file path. target volume for the registration Required.
args	Character. Additional parameters to the command
est_int_scale	Logical. estimate intensity scale (recommended for unnormalized images)
force_double	Logical. use double-precision intensities
force_float	Logical. use float intensities
half_source	Character or numeric vector. write source volume mapped to halfway space
half_source_xfm	Character or numeric vector. write transform from source to halfway space
half_targ	Character or numeric vector. write target volume mapped to halfway space
half_targ_xfm	Character or numeric vector. write transform from target to halfway space
half_weights	Character or numeric vector. write weights volume mapped to halfway space
high_iterations	Integer. max # of times on highest resolution
in_xfm_file	Character; file path. use initial transform on source
init_orient	Logical. use moments for initial orient (recommended for stripped brains)
iteration_thresh	Numeric. stop iterations when below threshold
least_squares	Logical. use least squares instead of robust estimator
mask_source	Character; file path. image to mask source volume with
mask_target	Character; file path. image to mask target volume with
max_iterations	Integer. maximum # of times on each resolution
no_init	Logical. skip transform init
no_multi	Logical. work on highest resolution
out_reg_file	Character or numeric vector. registration file; either True or filename
outlier_limit	Numeric. set maximal outlier limit in satit
registered_file	Character or numeric vector. registered image; either True or filename
subsample_thresh	Integer. subsample if dimension is above threshold size
trans_only	Logical. find 3 parameter translation only
weights_file	Character or numeric vector. weights image to write; either True or filename
write_vo2vox	Logical. output vox2vox matrix (default is RAS2RAS)
.cwd	Working directory override.

.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

```
ni_freesurfer_robust_template
      FREESURFER RobustTemplate
```

Description

construct an unbiased robust template for longitudinal volumes

Usage

```
ni_freesurfer_robust_template(
  auto_detect_sensitivity,
  in_files,
  out_file,
  outlier_sensitivity,
  args = NULL,
  average_metric = NULL,
  fixed_timepoint = NULL,
  in_intensity_scales = NULL,
  initial_timepoint = NULL,
  initial_transforms = NULL,
  intensity_scaling = NULL,
  no_iteration = NULL,
  scaled_intensity_outputs = NULL,
  subsample_threshold = NULL,
  transform_outputs = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

auto_detect_sensitivity	Logical. auto-detect good sensitivity (recommended for head or full brain scans) Required.
in_files	Character or numeric vector. input movable volumes to be aligned to common mean/median template Required.
out_file	Character; file path. output template volume (final mean/median image) Required.
outlier_sensitivity	Numeric. set outlier sensitivity manually (e.g. "-sat 4.685"). Higher values mean less sensitivity. Required.
args	Character. Additional parameters to the command
average_metric	Character; one of: "median", "mean". construct template from: 0 Mean, 1 Median (default)
fixed_timepoint	Logical. map everything to init TP# (init TP is not resampled)
in_intensity_scales	Character or numeric vector. use initial intensity scales
initial_timepoint	Integer. use TP# for special init (default random), 0: no init
initial_transforms	Character or numeric vector. use initial transforms (lta) on source
intensity_scaling	Logical. allow also intensity scaling (default off)
no_iteration	Logical. do not iterate, just create first template
scaled_intensity_outputs	Character or numeric vector. final intensity scales (will activate -iscale)
subsample_threshold	Integer. subsample if dim > # on all axes (default no subs.)
transform_outputs	Character or numeric vector. output xforms to template (for each input)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_sample_to_surface
FREESURFER SampleToSurface

Description

Sample a volume to the cortical surface using Freesurfer's mri_vol2surf.

Usage

```
ni_freesurfer_sample_to_surface(  
    hemi,  
    mni152reg,  
    projection_stem,  
    reg_file,  
    reg_header,  
    sampling_method,  
    source_file,  
    apply_rot = NULL,  
    apply_trans = NULL,  
    args = NULL,  
    cortex_mask = NULL,  
    fix_tk_reg = NULL,  
    float2int_method = NULL,  
    frame = NULL,  
    hits_file = NULL,  
    hits_type = NULL,  
    ico_order = NULL,  
    interp_method = NULL,  
    mask_label = NULL,  
    no_reshape = NULL,  
    out_file = NULL,  
    out_type = NULL,  
    override_reg_subj = NULL,  
    reference_file = NULL,  
    reshape = NULL,  
    reshape_slices = NULL,  
    scale_input = NULL,  
    smooth_surf = NULL,  
    smooth_vol = NULL,  
    surf_reg = NULL,  
    surface = NULL,  
    target_subject = NULL,  
    vox_file = NULL,  
    .cwd = NULL,  
    .env = NULL,  
    .engine = NULL,
```

```

    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

hemi	Character; one of: "lh", "rh". target hemisphere Required.
mni152reg	Logical. source volume is in MNI152 space Required.
projection_stem	Character. stem for precomputed linear estimates and volume fractions Required.
reg_file	Character; file path. source-to-reference registration file Required.
reg_header	Logical. register based on header geometry Required.
sampling_method	Character; one of: "point", "max", "average". how to sample – at a point or at the max or average over a range Required.
source_file	Character; file path. volume to sample values from Required.
apply_rot	Character or numeric vector. rotation angles (in degrees) to apply to reg matrix
apply_trans	Character or numeric vector. translation (in mm) to apply to reg matrix
args	Character. Additional parameters to the command
cortex_mask	Logical. mask the target surface with hemi.cortex.label
fix_tk_reg	Logical. make reg matrix round-compatible
float2int_method	Character; one of: "round", "tkregister". method to convert reg matrix values (default is round)
frame	Integer. save only one frame (0-based)
hits_file	Character or numeric vector. save image with number of hits at each voxel
hits_type	Character; one of: "cor", "mgh", "mgz", "minc", "analyze", "analyze4d", "spm", "afni", "brik", "bshort", "bfloat", "sdt", "outline", "otl", "gdf", "nifti1", "nii", "niigz". hits file type
ico_order	Integer. icosahedron order when target_subject is 'ico'
interp_method	Character; one of: "nearest", "trilinear". interpolation method
mask_label	Character; file path. label file to mask output with
no_reshape	Logical. do not reshape surface vector (default)
out_file	Character; file path. surface file to write
out_type	Character; one of: "cor", "mgh", "mgz", "minc", "analyze", "analyze4d", "spm", "afni", "brik", "bshort", "bfloat", "sdt", "outline", "otl", "gdf", "nifti1", "nii", "niigz", "gii". output file type
override_reg_subj	Logical. override the subject in the reg file header
reference_file	Character; file path. reference volume (default is orig.mgz)

reshape	Logical. reshape surface vector to fit in non-mgh format
reshape_slices	Integer. number of 'slices' for reshaping
scale_input	Numeric. multiple all intensities by scale factor
smooth_surf	Numeric. smooth output surface (mm fwhm)
smooth_vol	Numeric. smooth input volume (mm fwhm)
surf_reg	Character or numeric vector. use surface registration to target subject
surface	Character. target surface (default is white)
target_subject	Character. sample to surface of different subject than source
vox_file	Character or numeric vector. text file with the number of voxels intersecting the surface
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_seg_stats

FREESURFER SegStats

Description

Use FreeSurfer mri_segstats for ROI analysis

Usage

```
ni_freesurfer_seg_stats(
  annot,
  segmentation_file,
  surf_label,
  args = NULL,
  avgwf_file = NULL,
  avgwf_txt_file = NULL,
  brain_vol = NULL,
  brainmask_file = NULL,
  calc_power = NULL,
  calc_snr = NULL,
  color_table_file = NULL,
```

```

cortex_vol_from_surf = NULL,
default_color_table = NULL,
empty = NULL,
etiv = NULL,
euler = NULL,
exclude_ctx_gm_wm = NULL,
exclude_id = NULL,
frame = NULL,
gca_color_table = NULL,
in_file = NULL,
in_intensity = NULL,
intensity_units = NULL,
mask_erode = NULL,
mask_file = NULL,
mask_invert = NULL,
mask_thresh = NULL,
multiply = NULL,
non_empty_only = NULL,
partial_volume_file = NULL,
segment_id = NULL,
sf_avg_file = NULL,
subcort_gm = NULL,
summary_file = NULL,
supratent = NULL,
total_gray = NULL,
vox = NULL,
wm_vol_from_surf = NULL,
.cwd = NULL,
.env = NULL,
.engine = NULL,
.profile = NULL,
dry_run = FALSE,
echo = interactive()
)

```

Arguments

annot	Character or numeric vector. subject hemi parc : use surface parcellation Required.
segmentation_file	Character; file path. segmentation volume path Required.
surf_label	Character or numeric vector. subject hemi label : use surface label Required.
args	Character. Additional parameters to the command
avgwf_file	Character or numeric vector. Save as binary volume (bool or filename)
avgwf_txt_file	Character or numeric vector. Save average waveform into file (bool or filename)
brain_vol	Character; one of: "brain-vol-from-seg", "brainmask". Compute brain volume either with brainmask or brain-vol-from-seg

brainmask_file	Character; file path. Load brain mask and compute the volume of the brain as the non-zero voxels in this volume
calc_power	Character; one of: "sqr", "sqrt". Compute either the sqr or the sqrt of the input
calc_snr	Logical. save mean/std as extra column in output table
color_table_file	Character; file path. color table file with seg id names
cortex_vol_from_surf	Logical. Compute cortex volume from surf
default_color_table	Logical. use \$FREESURFER_HOME/FreeSurferColorLUT.txt
empty	Logical. Report on segmentations listed in the color table
etiv	Logical. Compute ICV from talairach transform
euler	Logical. Write out number of defect holes in orig.nofix based on the euler number
exclude_ctx_gm_wm	Logical. exclude cortical gray and white matter
exclude_id	Integer. Exclude seg id from report
frame	Integer. Report stats on nth frame of input volume
gca_color_table	Character; file path. get color table from GCA (CMA)
in_file	Character; file path. Use the segmentation to report stats on this volume
in_intensity	Character; file path. Undocumented input norm.mgz file
intensity_units	Character; one of: "MR". Intensity units
mask_erode	Integer. Erode mask by some amount
mask_file	Character; file path. Mask volume (same size as seg
mask_invert	Logical. Invert binarized mask volume
mask_thresh	Numeric. binarize mask with this threshold <0.5>
multiply	Numeric. multiply input by val
non_empty_only	Logical. Only report nonempty segmentations
partial_volume_file	Character; file path. Compensate for partial voluming
segment_id	Character or numeric vector. Manually specify segmentation ids
sf_avg_file	Character or numeric vector. Save mean across space and time
subcort_gm	Logical. Compute volume of subcortical gray matter
summary_file	Character; file path. Segmentation stats summary table file
supratent	Logical. Undocumented input flag
total_gray	Logical. Compute volume of total gray matter
vox	Character or numeric vector. Replace seg with all 0s except at C R S (three int inputs)

wm_vol_from_surf	Logical. Compute wm volume from surf
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_seg_stats_recon_all
FREESURFER SegStatsReconAll

Description

This class inherits SegStats and modifies it for use in a recon-all workflow.

Usage

```
ni_freesurfer_seg_stats_recon_all(
  annot,
  lh_orig_nofix,
  lh_pial,
  lh_white,
  rh_orig_nofix,
  rh_pial,
  rh_white,
  ribbon,
  segmentation_file,
  subject_id,
  surf_label,
  transform,
  args = NULL,
  avgwf_file = NULL,
  avgwf_txt_file = NULL,
  brain_vol = NULL,
  brainmask_file = NULL,
  calc_power = NULL,
  calc_snr = NULL,
  color_table_file = NULL,
  cortex_vol_from_surf = NULL,
```

```

default_color_table = NULL,
empty = NULL,
etiv = NULL,
euler = NULL,
exclude_ctx_gm_wm = NULL,
exclude_id = NULL,
frame = NULL,
gca_color_table = NULL,
in_file = NULL,
in_intensity = NULL,
intensity_units = NULL,
mask_erode = NULL,
mask_file = NULL,
mask_invert = NULL,
mask_thresh = NULL,
multiply = NULL,
non_empty_only = NULL,
partial_volume_file = NULL,
segment_id = NULL,
sf_avg_file = NULL,
subcort_gm = NULL,
summary_file = NULL,
supratent = NULL,
total_gray = NULL,
vox = NULL,
wm_vol_from_surf = NULL,
.cwd = NULL,
.env = NULL,
.engine = NULL,
.profile = NULL,
dry_run = FALSE,
echo = interactive()
)

```

Arguments

annot	Character or numeric vector. subject hemi parc : use surface parcellation Required.
lh_orig_nofix	Character; file path. Input lh.orig.nofix Required.
lh_pial	Character; file path. Input file must be <subject_id>/surf/lh.pial Required.
lh_white	Character; file path. Input file must be <subject_id>/surf/lh.white Required.
rh_orig_nofix	Character; file path. Input rh.orig.nofix Required.
rh_pial	Character; file path. Input file must be <subject_id>/surf/rh.pial Required.
rh_white	Character; file path. Input file must be <subject_id>/surf/rh.white Required.
ribbon	Character; file path. Input file mri/ribbon.mgz Required.
segmentation_file	Character; file path. segmentation volume path Required.

subject_id	Character. Subject id being processed Required.
surf_label	Character or numeric vector. subject hemi label : use surface label Required.
transform	Character; file path. Input transform file Required.
args	Character. Additional parameters to the command
avgwf_file	Character or numeric vector. Save as binary volume (bool or filename)
avgwf_txt_file	Character or numeric vector. Save average waveform into file (bool or filename)
brain_vol	Character; one of: "brain-vol-from-seg", "brainmask". Compute brain volume either with brainmask or brain-vol-from-seg
brainmask_file	Character; file path. Load brain mask and compute the volume of the brain as the non-zero voxels in this volume
calc_power	Character; one of: "sqr", "sqrt". Compute either the sqr or the sqrt of the input
calc_snr	Logical. save mean/std as extra column in output table
color_table_file	Character; file path. color table file with seg id names
cortex_vol_from_surf	Logical. Compute cortex volume from surf
default_color_table	Logical. use \$FREESURFER_HOME/FreeSurferColorLUT.txt
empty	Logical. Report on segmentations listed in the color table
etiv	Logical. Compute ICV from talairach transform
euler	Logical. Write out number of defect holes in orig.nofix based on the euler number
exclude_ctx_gm_wm	Logical. exclude cortical gray and white matter
exclude_id	Integer. Exclude seg id from report
frame	Integer. Report stats on nth frame of input volume
gca_color_table	Character; file path. get color table from GCA (CMA)
in_file	Character; file path. Use the segmentation to report stats on this volume
in_intensity	Character; file path. Undocumented input norm.mgz file
intensity_units	Character; one of: "MR". Intensity units
mask_erode	Integer. Erode mask by some amount
mask_file	Character; file path. Mask volume (same size as seg)
mask_invert	Logical. Invert binarized mask volume
mask_thresh	Numeric. binarize mask with this threshold <0.5>
multiply	Numeric. multiply input by val
non_empty_only	Logical. Only report nonempty segmentations
partial_volume_file	Character; file path. Compensate for partial voluming

segment_id	Character or numeric vector. Manually specify segmentation ids
sf_avg_file	Character or numeric vector. Save mean across space and time
subcort_gm	Logical. Compute volume of subcortical gray matter
summary_file	Character; file path. Segmentation stats summary table file
supratent	Logical. Undocumented input flag
total_gray	Logical. Compute volume of total gray matter
vox	Character or numeric vector. Replace seg with all 0s except at C R S (three int inputs)
wm_vol_from_surf	Logical. Compute wm volume from surf
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_segment_cc
FREESURFER SegmentCC

Description

This program segments the corpus callosum into five separate labels in

Usage

```
ni_freesurfer_segment_cc(
  in_file,
  in_norm,
  out_rotation,
  subject_id,
  args = NULL,
  out_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. Input aseg file to read from subjects directory Required.
in_norm	Character; file path. Required undocumented input {subject}/mri/norm.mgz Required.
out_rotation	Character; file path. Global filepath for writing rotation lta Required.
subject_id	Character. Subject name Required.
args	Character. Additional parameters to the command
out_file	Character; file path. Filename to write aseg including CC
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_segment_wm
FREESURFER SegmentWM

Description

This program segments white matter from the input volume. The input

Usage

```
ni_freesurfer_segment_wm(
  in_file,
  out_file,
  args = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. Input file for SegmentWM Required .
out_file	Character; file path. File to be written as output for SegmentWM Required .
args	Character. Additional parameters to the command
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_smooth *FREESURFER Smooth*

Description

Use FreeSurfer mris_volsmooth to smooth a volume

Usage

```
ni_freesurfer_smooth(
  in_file,
  num_iters,
  reg_file,
  surface_fwhm,
  args = NULL,
  proj_frac = NULL,
  proj_frac_avg = NULL,
  smoothed_file = NULL,
  vol_fwhm = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. source volume Required.
num_iters	Character. number of iterations instead of fwhm Required.
reg_file	Character; file path. registers volume to surface anatomical Required.
surface_fwhm	Character. surface FWHM in mm Required.
args	Character. Additional parameters to the command
proj_frac	Numeric. project frac of thickness a long surface normal
proj_frac_avg	Character or numeric vector. average a long normal min max delta
smoothed_file	Character; file path. output volume
vol_fwhm	Character. volume smoothing outside of surface
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_smooth_tessellation
FREESURFER SmoothTessellation

Description

Smooth a tessellated surface.

Usage

```
ni_freesurfer_smooth_tessellation(
  in_file,
  args = NULL,
  curvature_averaging_iterations = NULL,
  disable_estimates = NULL,
  gaussian_curvature_norm_steps = NULL,
  gaussian_curvature_smoothing_steps = NULL,
  normalize_area = NULL,
  out_area_file = NULL,
  out_curvature_file = NULL,
  out_file = NULL,
```

```

seed = NULL,
smoothing_iterations = NULL,
snapshot_writing_iterations = NULL,
use_gaussian_curvature_smoothing = NULL,
use_momentum = NULL,
.cwd = NULL,
.env = NULL,
.engine = NULL,
.profile = NULL,
dry_run = FALSE,
echo = interactive()
)

```

Arguments

in_file	Character; file path. Input volume to tessellate voxels from. Required.
args	Character. Additional parameters to the command
curvature_averaging_iterations	Integer. Number of curvature averaging iterations (default=10)
disable_estimates	Logical. Disables the writing of curvature and area estimates
gaussian_curvature_norm_steps	Integer. Use Gaussian curvature smoothing
gaussian_curvature_smoothing_steps	Integer. Use Gaussian curvature smoothing
normalize_area	Logical. Normalizes the area after smoothing
out_area_file	Character; file path. Write area to ?h. areaname (default "area")
out_curvature_file	Character; file path. Write curvature to ?h. curvname (default "curv")
out_file	Character; file path. output filename or True to generate one
seed	Integer. Seed for setting random number generator
smoothing_iterations	Integer. Number of smoothing iterations (default=10)
snapshot_writing_iterations	Integer. Write snapshot every <i>n</i> iterations
use_gaussian_curvature_smoothing	Logical. Use Gaussian curvature smoothing
use_momentum	Logical. Uses momentum
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_sphere *FREESURFER Sphere*

Description

This program will add a template into an average surface

Usage

```
ni_freesurfer_sphere(
  in_file,
  args = NULL,
  magic = NULL,
  out_file = NULL,
  seed = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. Input file for Sphere Required .
args	Character. Additional parameters to the command
magic	Logical. No documentation. Direct questions to analysis-bugs@nmr.mgh.harvard.edu
out_file	Character; file path. Output file for Sphere
seed	Integer. Seed for setting random number generator
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

```
ni_freesurfer_spherical_average
    FREESURFER SphericalAverage
```

Description

This program will add a template into an average surface.

Usage

```
ni_freesurfer_spherical_average(
    fname,
    hemisphere,
    in_surf,
    subject_id,
    which,
    args = NULL,
    erode = NULL,
    in_average = NULL,
    in_orig = NULL,
    out_file = NULL,
    threshold = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)
```

Arguments

fname	Character. Filename from the average subject directory. Example: to use rh.entorhinal.label as the input label filename, set fname to 'rh.entorhinal' and which to 'label'. The program will then search for <in_average>/label/rh.entorhinal.label Required.
hemisphere	Character; one of: "lh", "rh". Input hemisphere Required.
in_surf	Character; file path. Input surface file Required.
subject_id	Character. Output subject id Required.
which	Character; one of: "coords", "label", "vals", "curv", "area". No documentation Required.
args	Character. Additional parameters to the command
erode	Integer. Undocumented
in_average	Character; directory path. Average subject

in_orig	Character; file path. Original surface filename
out_file	Character; file path. Output filename
threshold	Numeric. Undocumented
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_surface_smooth
FREESURFER SurfaceSmooth

Description

Smooth a surface image with mri_surf2surf.

Usage

```
ni_freesurfer_surface_smooth(
  hemi,
  in_file,
  subject_id,
  args = NULL,
  cortex = TRUE,
  fwhm = NULL,
  out_file = NULL,
  reshape = NULL,
  smooth_iters = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

hemi	Character; one of: "lh", "rh". hemisphere to operate on Required.
in_file	Character; file path. source surface file Required.
subject_id	Character. subject id of surface file Required.
args	Character. Additional parameters to the command
cortex	Logical. only smooth within \$hemi.cortex.label
fwhm	Numeric. effective FWHM of the smoothing process
out_file	Character; file path. surface file to write
reshape	Logical. reshape surface vector to fit in non-mgh format
smooth_iters	Integer. iterations of the smoothing process
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_surface_snapshots
FREESURFER SurfaceSnapshots

Description

Use Tksurfer to save pictures of the cortical surface.

Usage

```
ni_freesurfer_surface_snapshots(  
  hemi,  
  subject_id,  
  surface,  
  annot_file = NULL,  
  annot_name = NULL,  
  args = NULL,  
  colortable = NULL,  
  demean_overlay = NULL,  
  identity_reg = NULL,  
  invert_overlay = NULL,
```

```

label_file = NULL,
label_name = NULL,
label_outline = NULL,
label_under = NULL,
mni152_reg = NULL,
orig_suffix = NULL,
overlay = NULL,
overlay_range = NULL,
overlay_range_offset = NULL,
overlay_reg = NULL,
patch_file = NULL,
reverse_overlay = NULL,
show_color_scale = NULL,
show_color_text = NULL,
show_curv = NULL,
show_gray_curv = NULL,
sphere_suffix = NULL,
tcl_script = NULL,
truncate_overlay = NULL,
.cwd = NULL,
.env = NULL,
.engine = NULL,
.profile = NULL,
dry_run = FALSE,
echo = interactive()
)

```

Arguments

hemi	Character; one of: "lh", "rh". hemisphere to visualize Required.
subject_id	Character. subject to visualize Required.
surface	Character. surface to visualize Required.
annot_file	Character; file path. path to annotation file to display
annot_name	Character. name of annotation to display (must be in \$subject/label directory)
args	Character. Additional parameters to the command
colortable	Character; file path. load colortable file
demean_overlay	Logical. remove mean from overlay
identity_reg	Logical. use the identity matrix to register the overlay to the surface
invert_overlay	Logical. invert the overlay display
label_file	Character; file path. path to label file to display
label_name	Character. name of label to display (must be in \$subject/label directory)
label_outline	Logical. draw label/annotation as outline
label_under	Logical. draw label/annotation under overlay
mni152_reg	Logical. use to display a volume in MNI152 space on the average subject

orig_suffix	Character. set the orig surface suffix string
overlay	Character; file path. load an overlay volume/surface
overlay_range	Character or numeric vector. overlay range—either min, (min, max) or (min, mid, max)
overlay_range_offset	Numeric. overlay range will be symmetric around offset value
overlay_reg	Character; file path. registration matrix file to register overlay to surface
patch_file	Character; file path. load a patch
reverse_overlay	Logical. reverse the overlay display
show_color_scale	Logical. display the color scale bar
show_color_text	Logical. display text in the color scale bar
show_curv	Logical. show curvature
show_gray_curv	Logical. show curvature in gray
sphere_suffix	Character. set the sphere.reg suffix string
tcl_script	Character; file path. override default screenshot script
truncate_overlay	Logical. truncate the overlay display
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_surface_transform
FREESURFER SurfaceTransform

Description

Transform a surface file from one subject to another via a spherical registration.

Usage

```

ni_freesurfer_surface_transform(
  hemi,
  source_annot_file,
  source_file,
  source_subject,
  target_subject,
  args = NULL,
  out_file = NULL,
  reshape = NULL,
  reshape_factor = NULL,
  source_type = NULL,
  target_ico_order = NULL,
  target_type = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)

```

Arguments

hemi	Character; one of: "lh", "rh". hemisphere to transform Required.
source_annot_file	Character; file path. surface annotation file Required.
source_file	Character; file path. surface file with source values Required.
source_subject	Character. subject id for source surface Required.
target_subject	Character. subject id of target surface Required.
args	Character. Additional parameters to the command
out_file	Character; file path. surface file to write
reshape	Logical. reshape output surface to conform with Nifti
reshape_factor	Integer. number of slices in reshaped image
source_type	Character; one of: "cor", "mgh", "mgz", "minc", "analyze", "analyze4d", "spm", "afni", "brik", "bshort", "bfloat", "sdt", "outline", "otl", "gdf", "nifti1", "nii", "niigz". source file format
target_ico_order	Character; one of: "1", "2", "3", "4", "5", "6", "7". order of the icosahedron if target_subject is 'ico'
target_type	Character; one of: "cor", "mgh", "mgz", "minc", "analyze", "analyze4d", "spm", "afni", "brik", "bshort", "bfloat", "sdt", "outline", "otl", "gdf", "nifti1", "nii", "niigz", "gii". output format
.cwd	Working directory override.

.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_surface2_vol_transform

FREESURFER Surface2VolTransform

Description

Use FreeSurfer mri_surf2vol to apply a transform.

Usage

```
ni_freesurfer_surface2_vol_transform(
  hemi,
  reg_file,
  source_file,
  args = NULL,
  mkmask = NULL,
  projfrac = NULL,
  subject_id = NULL,
  subjects_dir = NULL,
  surf_name = NULL,
  template_file = NULL,
  transformed_file = NULL,
  vertexvol_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

hemi	Character. hemisphere of data Required.
reg_file	Character; file path. tkRAS-to-tkRAS matrix (tkregister2 format) Required.

source_file	Character; file path. This is the source of the surface values Required.
args	Character. Additional parameters to the command
mkmask	Logical. make a mask instead of loading surface values
projfrac	Numeric. thickness fraction
subject_id	Character. subject id
subjects_dir	Character. freesurfer subjects directory defaults to \$SUBJECTS_DIR
surf_name	Character. surfname (default is white)
template_file	Character; file path. Output template volume
transformed_file	Character; file path. Output volume
vertexvol_file	Character; file path. Path name of the vertex output volume, which is the same as output volume except that the value of each voxel is the vertex-id that is mapped to that voxel.
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

```
ni_freesurfer_synthesize_flash
      FREESURFER SynthesizeFLASH
```

Description

Synthesize a FLASH acquisition from T1 and proton density maps.

Usage

```
ni_freesurfer_synthesize_flash(
    flip_angle,
    pd_image,
    t1_image,
    te,
    tr,
    args = NULL,
    fixed_weighting = NULL,
    out_file = NULL,
```

```

    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

flip_angle	Numeric. flip angle (in degrees) Required.
pd_image	Character; file path. image of proton density values Required.
t1_image	Character; file path. image of T1 values Required.
te	Numeric. echo time (in msec) Required.
tr	Numeric. repetition time (in msec) Required.
args	Character. Additional parameters to the command
fixed_weighting	Logical. use a fixed weighting to generate optimal gray/white contrast
out_file	Character; file path. image to write
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_talairach_avi

FREESURFER TalairachAVI

Description

Front-end for Avi Snyders image registration tool. Computes the

Usage

```

ni_freesurfer_talairach_avi(
  in_file,
  out_file,
  args = NULL,
  atlas = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)

```

Arguments

in_file	Character; file path. input volume Required.
out_file	Character; file path. output xfm file Required.
args	Character. Additional parameters to the command
atlas	Character. alternate target atlas (in freesurfer/average dir)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_talairach_qc

FREESURFER TalairachQC

Description

Examples

Usage

```
ni_freesurfer_talairach_qc(
  log_file,
  args = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

log_file	Character; file path. The log file for TalairachQC Required .
args	Character. Additional parameters to the command
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_tkregister2

FREESURFER Tkregister2

Description

Examples

Usage

```
ni_freesurfer_tkregister2(
  moving_image,
  reg_file,
  args = NULL,
  fsl_in_matrix = NULL,
  fsl_out = NULL,
  fstal = NULL,
  fstarg = NULL,
```

```

invert_lta_out = NULL,
lta_in = NULL,
lta_out = NULL,
movscale = NULL,
noedit = TRUE,
reg_header = NULL,
subject_id = NULL,
target_image = NULL,
xfm = NULL,
.cwd = NULL,
.env = NULL,
.engine = NULL,
.profile = NULL,
dry_run = FALSE,
echo = interactive()
)

```

Arguments

moving_image	Character; file path. moving volume Required.
reg_file	Character; file path. freesurfer-style registration file Required.
args	Character. Additional parameters to the command
fsl_in_matrix	Character; file path. fsl-style registration input matrix
fsl_out	Character or numeric vector. compute an FSL-compatible resgitation matrix
fstal	Logical. set mov to be tal and reg to be tal xfm
fstarg	Logical. use subject's T1 as reference
invert_lta_out	Logical. Invert input LTA before applying
lta_in	Character; file path. use a matrix in MNI coordinates as initial registration
lta_out	Character or numeric vector. output registration file (LTA format)
movscale	Numeric. adjust registration matrix to scale mov
noedit	Logical. do not open edit window (exit)
reg_header	Logical. compute registration from headers
subject_id	Character. freesurfer subject ID
target_image	Character; file path. target volume
xfm	Character; file path. use a matrix in MNI coordinates as initial registration
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

```
ni_freesurfer_unpack_sdicom_dir
    FREESURFER UnpackSDICOMDir
```

Description

Use ununpacksdcmDir to convert dicom files

Usage

```
ni_freesurfer_unpack_sdicom_dir(
    config,
    run_info,
    seq_config,
    source_dir,
    args = NULL,
    dir_structure = NULL,
    log_file = NULL,
    no_info_dump = NULL,
    no_unpack_err = NULL,
    output_dir = NULL,
    scan_only = NULL,
    spm_zeropad = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)
```

Arguments

config	Character; file path. specify unpacking rules in file Required.
run_info	Character or numeric vector. runno subdir format name : spec unpacking rules on cmdline Required.
seq_config	Character; file path. specify unpacking rules based on sequence Required.
source_dir	Character; directory path. directory with the DICOM files Required.
args	Character. Additional parameters to the command
dir_structure	Character; one of: "fsfast", "generic". unpack to specified directory structures
log_file	Character; file path. explicitly set log file
no_info_dump	Logical. do not create infodump file
no_unpack_err	Logical. do not try to unpack runs with errors
output_dir	Character; directory path. top directory into which the files will be unpacked

scan_only	Character; file path. only scan the directory and put result in file
spm_zeropad	Integer. set frame number zero padding width for SPM
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_volume_mask
FREESURFER VolumeMask

Description

Computes a volume mask, at the same resolution as the

Usage

```
ni_freesurfer_volume_mask(
  left_ribbonlabel,
  left_whitelabel,
  lh_pial,
  lh_white,
  rh_pial,
  rh_white,
  right_ribbonlabel,
  right_whitelabel,
  subject_id,
  args = NULL,
  in_aseg = NULL,
  save_ribbon = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

left_ribbonlabel	Integer. Left cortical ribbon label Required.
left_whitelabel	Integer. Left white matter label Required.
lh_pial	Character; file path. Implicit input left pial surface Required.
lh_white	Character; file path. Implicit input left white matter surface Required.
rh_pial	Character; file path. Implicit input right pial surface Required.
rh_white	Character; file path. Implicit input right white matter surface Required.
right_ribbonlabel	Integer. Right cortical ribbon label Required.
right_whitelabel	Integer. Right white matter label Required.
subject_id	Character. Subject being processed Required.
args	Character. Additional parameters to the command
in_aseg	Character; file path. Input aseg file for VolumeMask
save_ribbon	Logical. option to save just the ribbon for the hemispheres in the format ?h.ribbon.mgz
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_freesurfer_watershed_skull_strip
FREESURFER WatershedSkullStrip

Description

This program strips skull and other outer non-brain tissue and

Usage

```

ni_freesurfer_watershed_skull_strip(
  in_file,
  out_file,
  args = NULL,
  brain_atlas = NULL,
  t1 = NULL,
  transform = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)

```

Arguments

in_file	Character; file path. input volume Required.
out_file	Character; file path. output volume Required.
args	Character. Additional parameters to the command
brain_atlas	Character; file path
t1	Logical. specify T1 input volume (T1 grey value = 110)
transform	Character; file path. undocumented
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_from_openneuro	<i>Fetch and prepare OpenNeuro dataset</i>
-------------------	--

Description

Downloads an OpenNeuro dataset and optionally queries it for BIDS inputs matching a spec. This is a convenience wrapper around the openneuroR workflow.

Usage

```

ni_from_openneuro(
  dataset_id,
  tag = NULL,
  subjects = NULL,
  spec_id = NULL,
  modality = NULL,
  subid = NULL,
  task = NULL,
  session = NULL,
  run = NULL,
  derivatives = FALSE,
  prep_dir = "derivatives/fmriprep",
  quiet = FALSE,
  force = FALSE,
  ...
)

```

Arguments

dataset_id	OpenNeuro dataset identifier (e.g. "ds000114").
tag	Optional dataset version tag (e.g. "1.0.0").
subjects	Optional subject IDs to download (default: all subjects).
spec_id	Spec ID or ni_spec object to match inputs against (optional).
modality	Image modality to search for (e.g. "T1w", "bold").
subid	Subject ID filter for input query.
task	Task filter for input query.
session	Session filter for input query.
run	Run filter for input query.
derivatives	Include derivatives in the BIDS project.
prep_dir	Path to preprocessed derivatives directory.
quiet	Suppress progress messages.
force	Force re-download of cached data.
...	Additional arguments passed to openneuroR::on_fetch().

Value

If spec_id or modality is provided, returns a data.frame from ni_bids_inputs(). Otherwise, returns a bids_project object.

Examples

```

## Not run:
# Get the full BIDS project
proj <- ni_from_openneuro("ds000114")

```

```
# Get specific inputs for a spec
inputs <- ni_from_openneuro("ds000114", spec_id = "fsl_feat", modality = "bold")

## End(Not run)
```

ni_fsl_apply_mask	<i>FSL ApplyMask</i>
-------------------	----------------------

Description

Use fslmaths to apply a binary mask to another image.

Usage

```
ni_fsl_apply_mask(
  in_file,
  mask_file,
  args = NULL,
  internal_datatype = NULL,
  nan2zeros = NULL,
  out_file = NULL,
  output_datatype = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. image to operate on Required.
mask_file	Character; file path. binary image defining mask space Required.
args	Character. Additional parameters to the command
internal_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for calculations (default is float)
nan2zeros	Logical. change NaNs to zeros before doing anything
out_file	Character; file path. image to write
output_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for output (default uses input type)
.cwd	Working directory override.

.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_apply_topup	<i>FSL ApplyTOPUP</i>
--------------------	-----------------------

Description

Interface for FSL topup, a tool for estimating and correcting

Usage

```
ni_fsl_apply_topup(
  encoding_file,
  in_files,
  args = NULL,
  datatype = NULL,
  in_index = NULL,
  in_topup_fieldcoef = NULL,
  interp = NULL,
  method = NULL,
  out_corrected = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

encoding_file	Character; file path. name of text file with PE directions/times Required.
in_files	Character or numeric vector. name of file with images Required.
args	Character. Additional parameters to the command
datatype	Character; one of: "char", "short", "int", "float", "double". force output data type

in_index	Character or numeric vector. comma separated list of indices corresponding to -datain
in_topup_fieldcoef	Character; file path. topup file containing the field coefficients
interp	Character; one of: "trilinear", "spline". interpolation method
method	Character; one of: "jac", "lsr". use jacobian modulation (jac) or least-squares resampling (lsr)
out_corrected	Character; file path. output (warped) image
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_apply_warp	<i>FSL ApplyWarp</i>
-------------------	----------------------

Description

FSL's applywarp wrapper to apply the results of a FNIRT registration

Usage

```
ni_fsl_apply_warp(
  in_file,
  ref_file,
  abswarp = NULL,
  args = NULL,
  datatype = NULL,
  field_file = NULL,
  interp = NULL,
  mask_file = NULL,
  out_file = NULL,
  postmat = NULL,
  premat = NULL,
  relwarp = NULL,
  superlevel = NULL,
  supersample = NULL,
  .cwd = NULL,
```

```

    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

<code>in_file</code>	Character; file path. image to be warped Required.
<code>ref_file</code>	Character; file path. reference image Required.
<code>abswarp</code>	Logical. treat warp field as absolute: $x' = w(x)$
<code>args</code>	Character. Additional parameters to the command
<code>datatype</code>	Character; one of: "char", "short", "int", "float", "double". Force output data type [char short int float double].
<code>field_file</code>	Character; file path. file containing warp field
<code>interp</code>	Character; one of: "nn", "trilinear", "sinc", "spline". interpolation method
<code>mask_file</code>	Character; file path. filename for mask image (in reference space)
<code>out_file</code>	Character; file path. output filename
<code>postmat</code>	Character; file path. filename for post-transform (affine matrix)
<code>premat</code>	Character; file path. filename for pre-transform (affine matrix)
<code>relwarp</code>	Logical. treat warp field as relative: $x' = x + w(x)$
<code>superlevel</code>	Character or numeric vector. level of intermediary supersampling, a for 'automatic' or integer level. Default = 2
<code>supersample</code>	Logical. intermediary supersampling of output, default is off
<code>.cwd</code>	Working directory override.
<code>.env</code>	Named character vector of environment variables.
<code>.engine</code>	Execution engine override.
<code>.profile</code>	Runtime profile override.
<code>dry_run</code>	Logical; preview command without executing.
<code>echo</code>	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

ni_fsl_apply_xfm	<i>FSL ApplyXFM</i>
------------------	---------------------

Description

Currently just a light wrapper around FLIRT,

Usage

```
ni_fsl_apply_xfm(  
    in_file,  
    reference,  
    angle_rep = NULL,  
    apply_isoxfm = NULL,  
    apply_xfm = TRUE,  
    args = NULL,  
    bbrslope = NULL,  
    bbrtype = NULL,  
    bgvalue = NULL,  
    bins = NULL,  
    coarse_search = NULL,  
    cost = NULL,  
    cost_func = NULL,  
    datatype = NULL,  
    display_init = NULL,  
    dof = NULL,  
    echospacing = NULL,  
    fieldmap = NULL,  
    fieldmapmask = NULL,  
    fine_search = NULL,  
    force_scaling = NULL,  
    in_matrix_file = NULL,  
    in_weight = NULL,  
    interp = NULL,  
    min_sampling = NULL,  
    no_clamp = NULL,  
    no_resample = NULL,  
    no_resample_blur = NULL,  
    no_search = NULL,  
    out_file = NULL,  
    out_matrix_file = NULL,  
    padding_size = NULL,  
    pedir = NULL,  
    ref_weight = NULL,  
    rigid2D = NULL,  
    schedule = NULL,  
    searchr_x = NULL,
```

```

    searchr_y = NULL,
    searchr_z = NULL,
    sinc_width = NULL,
    sinc_window = NULL,
    uses_qform = NULL,
    verbose = NULL,
    wm_seg = NULL,
    wmcoords = NULL,
    wmnorms = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

in_file	Character; file path. input file Required.
reference	Character; file path. reference file Required.
angle_rep	Character; one of: "quaternion", "euler". representation of rotation angles
apply_isoxfm	Numeric. as applyxfm but forces isotropic resampling
apply_xfm	Logical. apply transformation supplied by in_matrix_file or uses_qform to use the affine matrix stored in the reference header
args	Character. Additional parameters to the command
bbrslope	Numeric. value of bbr slope
bbrtype	Character; one of: "signed", "global_abs", "local_abs". type of bbr cost function: signed [default], global_abs, local_abs
bgvalue	Numeric. use specified background value for points outside FOV
bins	Integer. number of histogram bins
coarse_search	Integer. coarse search delta angle
cost	Character; one of: "mutualinfo", "corratio", "normcorr", "normmi", "leastsq", "labeldiff", "bbr". cost function
cost_func	Character; one of: "mutualinfo", "corratio", "normcorr", "normmi", "leastsq", "labeldiff", "bbr". cost function
datatype	Character; one of: "char", "short", "int", "float", "double". force output data type
display_init	Logical. display initial matrix
dof	Integer. number of transform degrees of freedom
echospadding	Numeric. value of EPI echo spacing - units of seconds
fieldmap	Character; file path. fieldmap image in rads/s - must be already registered to the reference image

fieldmapmask	Character; file path. mask for fieldmap image
fine_search	Integer. fine search delta angle
force_scaling	Logical. force rescaling even for low-res images
in_matrix_file	Character; file path. input 4x4 affine matrix
in_weight	Character; file path. File for input weighting volume
interp	Character; one of: "trilinear", "nearestneighbour", "sinc", "spline". final interpolation method used in reslicing
min_sampling	Numeric. set minimum voxel dimension for sampling
no_clamp	Logical. do not use intensity clamping
no_resample	Logical. do not change input sampling
no_resample_blur	Logical. do not use blurring on downsampling
no_search	Logical. set all angular searches to ranges 0 to 0
out_file	Character; file path. registered output file
out_matrix_file	Character; file path. output affine matrix in 4x4 asciii format
padding_size	Integer. for applyxfm: interpolates outside image by size
pedir	Integer. phase encode direction of EPI - 1/2/3=x/y/z & -1/-2/-3=-x/-y/-z
ref_weight	Character; file path. File for reference weighting volume
rigid2D	Logical. use 2D rigid body mode - ignores dof
schedule	Character; file path. replaces default schedule
searchr_x	Character or numeric vector. search angles along x-axis, in degrees
searchr_y	Character or numeric vector. search angles along y-axis, in degrees
searchr_z	Character or numeric vector. search angles along z-axis, in degrees
sinc_width	Integer. full-width in voxels
sinc_window	Character; one of: "rectangular", "hanning", "blackman". sinc window
uses_qform	Logical. initialize using sform or qform
verbose	Integer. verbose mode, 0 is least
wm_seg	Character; file path. white matter segmentation volume needed by BBR cost function
wmcoords	Character; file path. white matter boundary coordinates for BBR cost function
wmnorms	Character; file path. white matter boundary normals for BBR cost function
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

ni_fsl_ar1_image	<i>FSLAR1Image</i>
------------------	--------------------

Description

Use fslmaths to generate an AR1 coefficient image across a

Usage

```
ni_fsl_ar1_image(
  in_file,
  args = NULL,
  dimension = "T",
  internal_datatype = NULL,
  nan2zeros = NULL,
  out_file = NULL,
  output_datatype = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. image to operate on Required.
args	Character. Additional parameters to the command
dimension	Character; one of: "T", "X", "Y", "Z". dimension to find AR(1) coefficient across
internal_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for calculations (default is float)
nan2zeros	Logical. change NaNs to zeros before doing anything
out_file	Character; file path. image to write
output_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for output (default uses input type)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_av_scale	<i>FSL AvScale</i>
-----------------	--------------------

Description

Use FSL avscale command to extract info from mat file output of FLIRT

Usage

```
ni_fsl_av_scale(
  all_param = NULL,
  args = NULL,
  mat_file = NULL,
  ref_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

all_param	Logical
args	Character. Additional parameters to the command
mat_file	Character; file path. mat file to read
ref_file	Character; file path. reference file to get center of rotation
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_b0_calc	<i>FSL B0Calc</i>
----------------	-------------------

Description

B0 inhomogeneities occur at interfaces of materials with different magnetic susceptibilities,

Usage

```
ni_fsl_b0_calc(
  in_file,
  args = NULL,
  chi_air = 4e-07,
  compute_xyz = FALSE,
  delta = -9.45e-06,
  directconv = FALSE,
  extendboundary = 1,
  out_file = NULL,
  x_b0 = 0,
  x_grad = 0,
  xyz_b0 = NULL,
  y_b0 = 0,
  y_grad = 0,
  z_b0 = 1,
  z_grad = 0,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. filename of input image (usually a tissue/air segmentation) Required.
args	Character. Additional parameters to the command
chi_air	Numeric. susceptibility of air
compute_xyz	Logical. calculate and save all 3 field components (i.e. x,y,z)
delta	Numeric. Delta value (chi_tissue - chi_air)
directconv	Logical. use direct (image space) convolution, not FFT
extendboundary	Numeric. Relative proportion to extend voxels at boundary
out_file	Character; file path. filename of B0 output volume
x_b0	Numeric. Value for zeroth-order b0 field (x-component), in Tesla

x_grad	Numeric. Value for zeroth-order x-gradient field (per mm)
xyz_b0	Character or numeric vector. Zeroth-order B0 field in Tesla
y_b0	Numeric. Value for zeroth-order b0 field (y-component), in Tesla
y_grad	Numeric. Value for zeroth-order y-gradient field (per mm)
z_b0	Numeric. Value for zeroth-order b0 field (z-component), in Tesla
z_grad	Numeric. Value for zeroth-order z-gradient field (per mm)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

ni_fsl_bedpostx5	<i>FSL BEDPOSTX5</i>
------------------	----------------------

Description

BEDPOSTX stands for Bayesian Estimation of Diffusion Parameters Obtained

Usage

```
ni_fsl_bedpostx5(
  bvals,
  bvecs,
  dwi,
  mask,
  n_fibres,
  out_dir,
  all_ard = NULL,
  args = NULL,
  burn_in = 0,
  burn_in_no_ard = 0,
  cilinear = NULL,
  f0_ard = NULL,
  f0_noard = NULL,
  force_dir = TRUE,
  fudge = NULL,
  gradnonlin = NULL,
  logdir = NULL,
```

```

model = NULL,
n_jumps = 5000,
no_ard = NULL,
no_spat = NULL,
non_linear = NULL,
rician = NULL,
sample_every = 1,
seed = NULL,
update_proposal_every = 40,
.cwd = NULL,
.env = NULL,
.engine = NULL,
.profile = NULL,
dry_run = FALSE,
echo = interactive()
)

```

Arguments

bvals	Character; file path. b values file Required.
bvecs	Character; file path. b vectors file Required.
dwi	Character; file path. diffusion weighted image data file Required.
mask	Character; file path. bet binary mask file Required.
n_fibres	Character. Maximum number of fibres to fit in each voxel Required.
out_dir	Character; directory path. output directory Required.
all_ard	Logical. Turn ARD on on all fibres
args	Character. Additional parameters to the command
burn_in	Character. Total num of jumps at start of MCMC to be discarded
burn_in_no_ard	Character. num of burnin jumps before the ard is imposed
cnlinear	Logical. Initialise with constrained nonlinear fitting
f0_ard	Logical. Noise floor model: add to the model an unattenuated signal compartment f0
f0_noard	Logical. Noise floor model: add to the model an unattenuated signal compartment f0
force_dir	Logical. use the actual directory name given (do not add + to make a new directory)
fudge	Integer. ARD fudge factor
gradnonlin	Logical. consider gradient nonlinearities, default off
logdir	Character; directory path
model	Character; one of: "1", "2", "3". use monoexponential (1, default, required for single-shell) or multiexponential (2, multi-shell) model
n_jumps	Integer. Num of jumps to be made by MCMC
no_ard	Logical. Turn ARD off on all fibres

no_spat	Logical. Initialise with tensor, not spatially
non_linear	Logical. Initialise with nonlinear fitting
rician	Logical. use Rician noise modeling
sample_every	Character. Num of jumps for each sample (MCMC)
seed	Integer. seed for pseudo random number generator
update_proposal_every	Character. Num of jumps for each update to the proposal density std (MCMC)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_bet

FSL BET

Description

FSL BET wrapper for skull stripping

Usage

```
ni_fsl_bet(
  in_file,
  args = NULL,
  center = NULL,
  frac = NULL,
  functional = NULL,
  mask = NULL,
  mesh = NULL,
  no_output = NULL,
  out_file = NULL,
  outline = NULL,
  padding = NULL,
  radius = NULL,
  reduce_bias = NULL,
  remove_eyes = NULL,
  robust = NULL,
  skull = NULL,
```

```

surfaces = NULL,
t2_guided = NULL,
threshold = NULL,
vertical_gradient = NULL,
.cwd = NULL,
.env = NULL,
.engine = NULL,
.profile = NULL,
dry_run = FALSE,
echo = interactive()
)

```

Arguments

in_file	Character; file path. input file to skull strip Required.
args	Character. Additional parameters to the command
center	Character or numeric vector. center of gravity in voxels
frac	Numeric. fractional intensity threshold
functional	Logical. apply to 4D fMRI data
mask	Logical. create binary mask image
mesh	Logical. generate a vtk mesh brain surface
no_output	Logical. Don't generate segmented output
out_file	Character; file path. name of output skull stripped image
outline	Logical. create surface outline image
padding	Logical. improve BET if FOV is very small in Z (by temporarily padding end slices)
radius	Integer. head radius
reduce_bias	Logical. bias field and neck cleanup
remove_eyes	Logical. eye & optic nerve cleanup (can be useful in SIENA)
robust	Logical. robust brain centre estimation (iterates BET several times)
skull	Logical. create skull image
surfaces	Logical. run bet2 and then betsurf to get additional skull and scalp surfaces (includes registrations)
t2_guided	Character; file path. as with creating surfaces, when also feeding in non-brain-extracted T2 (includes registrations)
threshold	Logical. apply thresholding to segmented brain image and mask
vertical_gradient	Numeric. vertical gradient in fractional intensity threshold (-1, 1)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_binary_maths	<i>FSL BinaryMaths</i>
---------------------	------------------------

Description

Use fslmaths to perform mathematical operations using a second image or

Usage

```
ni_fsl_binary_maths(
  in_file,
  operand_file,
  operand_value,
  operation,
  args = NULL,
  internal_datatype = NULL,
  nan2zeros = NULL,
  out_file = NULL,
  output_datatype = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. image to operate on Required.
operand_file	Character; file path. second image to perform operation with Required.
operand_value	Numeric. value to perform operation with Required.
operation	Character; one of: "add", "sub", "mul", "div", "rem", "max", "min". operation to perform Required.
args	Character. Additional parameters to the command
internal_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for calculations (default is float)
nan2zeros	Logical. change NaNs to zeros before doing anything
out_file	Character; file path. image to write

output_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for output (default uses input type)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_change_data_type
FSL ChangeDataType

Description

Use fslmaths to change the datatype of an image.

Usage

```
ni_fsl_change_data_type(
  in_file,
  output_datatype,
  args = NULL,
  internal_datatype = NULL,
  nan2zeros = NULL,
  out_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. image to operate on Required.
output_datatype	Character; one of: "float", "char", "int", "short", "double", "input". output data type Required.

args	Character. Additional parameters to the command
internal_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for calculations (default is float)
nan2zeros	Logical. change NaNs to zeros before doing anything
out_file	Character; file path. image to write
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_cluster	<i>FSL Cluster</i>
----------------	--------------------

Description

Uses FSL cluster to perform clustering on statistical output

Usage

```
ni_fsl_cluster(
  in_file,
  threshold,
  args = NULL,
  connectivity = NULL,
  cope_file = NULL,
  dlh = NULL,
  find_min = FALSE,
  fractional = FALSE,
  minclustersize = FALSE,
  no_table = FALSE,
  num_maxima = NULL,
  out_index_file = NULL,
  out_localmax_txt_file = NULL,
  out_localmax_vol_file = NULL,
  out_max_file = NULL,
  out_mean_file = NULL,
  out_pval_file = NULL,
  out_size_file = NULL,
```

```

    out_threshold_file = NULL,
    peak_distance = NULL,
    pthreshold = NULL,
    std_space_file = NULL,
    use_mm = FALSE,
    volume = NULL,
    warpfield_file = NULL,
    xfm_file = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

in_file	Character; file path. input volume Required.
threshold	Numeric. threshold for input volume Required.
args	Character. Additional parameters to the command
connectivity	Integer. the connectivity of voxels (default 26)
cope_file	Character; file path. cope volume
dlh	Numeric. smoothness estimate = $\sqrt{\det(\text{Lambda})}$
find_min	Logical. find minima instead of maxima
fractional	Logical. interprets the threshold as a fraction of the robust range
minclustersize	Logical. prints out minimum significant cluster size
no_table	Logical. suppresses printing of the table info
num_maxima	Integer. no of local maxima to report
out_index_file	Character or numeric vector. output of cluster index (in size order)
out_localmax_txt_file	Character or numeric vector. local maxima text file
out_localmax_vol_file	Character or numeric vector. output of local maxima volume
out_max_file	Character or numeric vector. filename for output of max image
out_mean_file	Character or numeric vector. filename for output of mean image
out_pval_file	Character or numeric vector. filename for image output of log pvals
out_size_file	Character or numeric vector. filename for output of size image
out_threshold_file	Character or numeric vector. thresholded image
peak_distance	Numeric. minimum distance between local maxima/minima, in mm (default 0)
pthreshold	Numeric. p-threshold for clusters
std_space_file	Character; file path. filename for standard-space volume

use_mm	Logical. use mm, not voxel, coordinates
volume	Integer. number of voxels in the mask
warpfield_file	Character; file path. file containing warpfield
xfm_file	Character; file path. filename for Linear: input->standard-space transform. Non-linear: input->highres transform
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_complex	<i>FSL Complex</i>
----------------	--------------------

Description

fslcomplex is a tool for converting complex data

Usage

```
ni_fsl_complex(
  args = NULL,
  complex_cartesian = NULL,
  complex_in_file = NULL,
  complex_in_file2 = NULL,
  complex_merge = NULL,
  complex_out_file = NULL,
  complex_polar = NULL,
  complex_split = NULL,
  end_vol = NULL,
  imaginary_in_file = NULL,
  imaginary_out_file = NULL,
  magnitude_in_file = NULL,
  magnitude_out_file = NULL,
  phase_in_file = NULL,
  phase_out_file = NULL,
  real_cartesian = NULL,
  real_in_file = NULL,
  real_out_file = NULL,
```

```

    real_polar = NULL,
    start_vol = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

args	Character. Additional parameters to the command
complex_cartesian	Logical
complex_in_file	Character; file path
complex_in_file2	Character; file path
complex_merge	Logical
complex_out_file	Character; file path
complex_polar	Logical
complex_split	Logical
end_vol	Integer
imaginary_in_file	Character; file path
imaginary_out_file	Character; file path
magnitude_in_file	Character; file path
magnitude_out_file	Character; file path
phase_in_file	Character; file path
phase_out_file	Character; file path
real_cartesian	Logical
real_in_file	Character; file path
real_out_file	Character; file path
real_polar	Logical
start_vol	Integer
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.

.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_contrast_mgr *FSL ContrastMgr*

Description

Use FSL contrast_mgr command to evaluate contrasts

Usage

```
ni_fsl_contrast_mgr(
  corrections,
  dof_file,
  param_estimates,
  sigmasquareds,
  tcon_file,
  args = NULL,
  contrast_num = NULL,
  fcon_file = NULL,
  suffix = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

corrections	Character; file path. statistical corrections used within FILM modelling Required.
dof_file	Character; file path. degrees of freedom Required.
param_estimates	Character or numeric vector. Parameter estimates for each column of the design matrix Required.
sigmasquareds	Character; file path. summary of residuals, See Woolrich, et. al., 2001 Required.
tcon_file	Character; file path. contrast file containing T-contrasts Required.

args	Character. Additional parameters to the command
contrast_num	Character. contrast number to start labeling copes from
fcon_file	Character; file path. contrast file containing F-contrasts
suffix	Character. suffix to put on the end of the cope filename before the contrast number, default is nothing
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_convert_warp *FSL ConvertWarp*

Description

Use FSL convertwarp <http://fsl.fmrib.ox.ac.uk/fsl/fsl-4.1.9/fnirt/warp_utils.html>_

Usage

```
ni_fsl_convert_warp(
  reference,
  abs warp = NULL,
  args = NULL,
  cons_jacobian = NULL,
  jacobian_max = NULL,
  jacobian_min = NULL,
  midmat = NULL,
  out_abs warp = NULL,
  out_file = NULL,
  out_rel warp = NULL,
  postmat = NULL,
  premat = NULL,
  rel warp = NULL,
  shift_direction = NULL,
  shift_in_file = NULL,
  warp1 = NULL,
  warp2 = NULL,
  .cwd = NULL,
```

```

    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

reference	Character; file path. Name of a file in target space of the full transform. Required.
abswarp	Logical. If set it indicates that the warps in <code>-warp1</code> and <code>-warp2</code> should be interpreted as absolute. I.e. the values in <code>-warp1/2</code> are the coordinates in the next space, rather than displacements. This flag is ignored if <code>-warp1/2</code> was created by <code>fnirt</code> , which always creates relative displacements.
args	Character. Additional parameters to the command
cons_jacobian	Logical. Constrain the Jacobian of the warpfield to lie within specified min/max limits.
jacobian_max	Numeric. Maximum acceptable Jacobian value for constraint (default 100.0)
jacobian_min	Numeric. Minimum acceptable Jacobian value for constraint (default 0.01)
midmat	Character; file path. Name of file containing mid-warp-affine transform
out_abswarp	Logical. If set it indicates that the warps in <code>-out</code> should be absolute, i.e. the values in <code>-out</code> are displacements from the coordinates in <code>-ref</code> .
out_file	Character; file path. Name of output file, containing warps that are the combination of all those given as arguments. The format of this will be a field-file (rather than spline coefficients) with any affine components included.
out_relwarp	Logical. If set it indicates that the warps in <code>-out</code> should be relative, i.e. the values in <code>-out</code> are displacements from the coordinates in <code>-ref</code> .
postmat	Character; file path. Name of file containing an affine transform (applied last). It could e.g. be an affine transform that maps the MNI152-space into a better approximation to the Talairach-space (if indeed there is one).
premat	Character; file path. filename for pre-transform (affine matrix)
relwarp	Logical. If set it indicates that the warps in <code>-warp1/2</code> should be interpreted as relative. I.e. the values in <code>-warp1/2</code> are displacements from the coordinates in the next space.
shift_direction	Character; one of: "y-", "y", "x", "x-", "z", "z-". Indicates the direction that the distortions from <code>-shiftmap</code> goes. It depends on the direction and polarity of the phase-encoding in the EPI sequence.
shift_in_file	Character; file path. Name of file containing a "shiftmap", a non-linear transform with displacements only in one direction (applied first, before <code>premat</code>). This would typically be a fieldmap that has been pre-processed using <code>fugue</code> that maps a subjects functional (EPI) data onto an undistorted space (i.e. a space that corresponds to his/her true anatomy).

warp1	Character; file path. Name of file containing initial warp-fields/coefficients (follows premat). This could e.g. be a fnirt-transform from a subjects structural scan to an average of a group of subjects.
warp2	Character; file path. Name of file containing secondary warp-fields/coefficients (after warp1/midmat but before postmat). This could e.g. be a fnirt-transform from the average of a group of subjects to some standard space (e.g. MNI152).
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

ni_fsl_convert_xfm	<i>FSL ConvertXFM</i>
--------------------	-----------------------

Description

Use the FSL utility `convert_xfm` to modify FLIRT transformation matrices.

Usage

```
ni_fsl_convert_xfm(
  in_file,
  args = NULL,
  concat_xfm = NULL,
  fix_scale_skew = NULL,
  in_file2 = NULL,
  invert_xfm = NULL,
  out_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. input transformation matrix Required.
args	Character. Additional parameters to the command
concat_xfm	Logical. write joint transformation of two input matrices
fix_scale_skew	Logical. use secondary matrix to fix scale and skew
in_file2	Character; file path. second input matrix (for use with fix_scale_skew or concat_xfm)
invert_xfm	Logical. invert input transformation
out_file	Character; file path. final transformation matrix
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_copy_geom	<i>FSL CopyGeom</i>
------------------	---------------------

Description

Use fslcpgeom to copy the header geometry information to another image.

Usage

```
ni_fsl_copy_geom(
  dest_file,
  in_file,
  args = NULL,
  ignore_dims = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

dest_file	Character; file path. destination image Required.
in_file	Character; file path. source image Required.
args	Character. Additional parameters to the command
ignore_dims	Logical. Do not copy image dimensions
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_dilate_image	<i>FSL DilateImage</i>
---------------------	------------------------

Description

Use fslmaths to perform a spatial dilation of an image.

Usage

```
ni_fsl_dilate_image(
  in_file,
  operation,
  args = NULL,
  internal_datatype = NULL,
  kernel_file = NULL,
  kernel_shape = NULL,
  kernel_size = NULL,
  nan2zeros = NULL,
  out_file = NULL,
  output_datatype = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. image to operate on Required.
operation	Character; one of: "mean", "modal", "max". filtering operation to perform in dilation Required.
args	Character. Additional parameters to the command
internal_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for calculations (default is float)
kernel_file	Character; file path. use external file for kernel
kernel_shape	Character; one of: "3D", "2D", "box", "boxv", "gauss", "sphere", "file". kernel shape to use
kernel_size	Numeric. kernel size - voxels for box/boxv, mm for sphere, mm sigma for gauss
nan2zeros	Logical. change NaNs to zeros before doing anything
out_file	Character; file path. image to write
output_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for output (default uses input type)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_distance_map *FSL DistanceMap*

Description

Use FSL's distancemap to generate a map of the distance to the nearest

Usage

```
ni_fsl_distance_map(
  in_file,
  args = NULL,
  distance_map = NULL,
  invert_input = NULL,
  local_max_file = NULL,
  mask_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. image to calculate distance values for Required .
args	Character. Additional parameters to the command
distance_map	Character; file path. distance map to write
invert_input	Logical. invert input image
local_max_file	Character or numeric vector. write an image of the local maxima
mask_file	Character; file path. binary mask to constrain calculations
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

ni_fsl_dti_fit

FSL DTIFit

Description

Use FSL dtifit command for fitting a diffusion tensor model at each

Usage

```

ni_fsl_dti_fit(
  bvals,
  bvecs,
  dwi,
  mask,
  args = NULL,
  base_name = "dtifit_",
  cni = NULL,
  gradnonlin = NULL,
  little_bit = NULL,
  max_x = NULL,
  max_y = NULL,
  max_z = NULL,
  min_x = NULL,
  min_y = NULL,
  min_z = NULL,
  save_tensor = NULL,
  sse = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)

```

Arguments

bvals	Character; file path. b values file Required.
bvecs	Character; file path. b vectors file Required.
dwi	Character; file path. diffusion weighted image data file Required.
mask	Character; file path. bet binary mask file Required.
args	Character. Additional parameters to the command
base_name	Character. base_name that all output files will start with
cni	Character; file path. input counfound regressors
gradnonlin	Character; file path. gradient non linearities
little_bit	Logical. only process small area of brain
max_x	Integer. max x
max_y	Integer. max y
max_z	Integer. max z
min_x	Integer. min x
min_y	Integer. min y
min_z	Integer. min z

save_tensor	Logical. save the elements of the tensor
sse	Logical. output sum of squared errors
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_dual_regression

FSL DualRegression

Description

Wrapper Script for Dual Regression Workflow

Usage

```
ni_fsl_dual_regression(  
  group_IC_maps_4D,  
  in_files,  
  n_perm,  
  args = NULL,  
  con_file = NULL,  
  des_norm = TRUE,  
  design_file = NULL,  
  one_sample_group_mean = NULL,  
  out_dir = "output",  
  .cwd = NULL,  
  .env = NULL,  
  .engine = NULL,  
  .profile = NULL,  
  dry_run = FALSE,  
  echo = interactive()  
)
```

Arguments

group_IC_maps_4D	Character; file path. 4D image containing spatial IC maps (melodic_IC) from the whole-group ICA analysis Required.
in_files	Character or numeric vector. List all subjects' preprocessed, standard-space 4D datasets Required.
n_perm	Integer. Number of permutations for randomise; set to 1 for just raw tstat output, set to 0 to not run randomise at all. Required.
args	Character. Additional parameters to the command
con_file	Character; file path. Design contrasts for final cross-subject modelling with randomise
des_norm	Logical. Whether to variance-normalise the timecourses used as the stage-2 regressors; True is default and recommended
design_file	Character; file path. Design matrix for final cross-subject modelling with randomise
one_sample_group_mean	Logical. perform 1-sample group-mean test instead of generic permutation test
out_dir	Character; directory path. This directory will be created to hold all output and logfiles
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

ni_fsl_eddy

FSL Eddy

Description

Interface for FSL eddy, a tool for estimating and correcting eddy

Usage

```
ni_fsl_eddy(  
  in_acqp,  
  in_bval,  
  in_bvec,  
  in_file,  
  in_index,  
  in_mask,  
  args = NULL,  
  cnr_maps = NULL,  
  dont_peas = NULL,  
  dont_sep_offs_move = NULL,  
  estimate_move_by_susceptibility = NULL,  
  fep = NULL,  
  field = NULL,  
  field_mat = NULL,  
  flm = "quadratic",  
  fudge_factor = 10,  
  fwhm = NULL,  
  in_topup_fieldcoef = NULL,  
  initrand = NULL,  
  interp = "spline",  
  is_shelled = NULL,  
  json = NULL,  
  mbs_ksp = NULL,  
  mbs_lambda = NULL,  
  mbs_niter = NULL,  
  method = "jac",  
  mporder = NULL,  
  multiband_factor = NULL,  
  multiband_offset = NULL,  
  niter = 5,  
  nvoxhp = 1000,  
  out_base = "eddy_corrected",  
  outlier_nstd = NULL,  
  outlier_nvox = NULL,  
  outlier_pos = NULL,  
  outlier_sqr = NULL,  
  outlier_type = NULL,  
  repol = NULL,  
  residuals = NULL,  
  session = NULL,  
  slice2vol_interp = NULL,  
  slice2vol_lambda = NULL,  
  slice2vol_niter = NULL,  
  slice_order = NULL,  
  slm = "none",  
  .cwd = NULL,  
)
```

```

    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

in_acqp	Character; file path. File containing acquisition parameters Required.
in_bval	Character; file path. File containing the b-values for all volumes in <code>-imain</code> Required.
in_bvec	Character; file path. File containing the b-vectors for all volumes in <code>-imain</code> Required.
in_file	Character; file path. File containing all the images to estimate distortions for Required.
in_index	Character; file path. File containing indices for all volumes in <code>-imain</code> into <code>-acqp</code> and <code>-topup</code> Required.
in_mask	Character; file path. Mask to indicate brain Required.
args	Character. Additional parameters to the command
cnr_maps	Logical. Output CNR-Maps
dont_peas	Logical. Do NOT perform a post-eddy alignment of shells
dont_sep_offs_move	Logical. Do NOT attempt to separate field offset from subject movement
estimate_move_by_susceptibility	Logical. Estimate how susceptibility field changes with subject movement
fep	Logical. Fill empty planes in x- or y-directions
field	Character; file path. Non-topup derived fieldmap scaled in Hz
field_mat	Character; file path. Matrix specifying the relative positions of the fieldmap, <code>-field</code> , and the first volume of the input file, <code>-imain</code>
flm	Character; one of: "quadratic", "linear", "cubic". First level EC model
fudge_factor	Numeric. Fudge factor for hyperparameter error variance
fwhm	Numeric. FWHM for conditioning filter when estimating the parameters
in_topup_fieldcoef	Character; file path. Topup results file containing the field coefficients
initransd	Logical. Resets rand for when selecting voxels
interp	Character; one of: "spline", "trilinear". Interpolation model for estimation step
is_shelled	Logical. Override internal check to ensure that data are acquired on a set of b-value shells
json	Character; file path. Name of .json text file with information about slice timing
mbs_ksp	Integer. Knot-spacing for MBS field estimation
mbs_lambda	Integer. Weighting of regularisation for MBS estimation

mbs_niter	Integer. Number of iterations for MBS estimation
method	Character; one of: "jac", "lsr". Final resampling method (jacobian/least squares)
mporder	Integer. Order of slice-to-vol movement model
multiband_factor	Integer. Multi-band factor
multiband_offset	Character; one of: "0", "1", "-1". Multi-band offset (-1 if bottom slice removed, 1 if top slice removed)
niter	Integer. Number of iterations
nvoxhp	Integer. # of voxels used to estimate the hyperparameters
out_base	Character. Basename for output image
outlier_nstd	Integer. Number of std off to qualify as outlier
outlier_nvox	Integer. Min # of voxels in a slice for inclusion in outlier detection
outlier_pos	Logical. Consider both positive and negative outliers if set
outlier_sqr	Logical. Consider outliers among sums-of-squared differences if set
outlier_type	Character; one of: "sw", "gw", "both". Type of outliers, slicewise (sw), group-wise (gw) or both (both)
repol	Logical. Detect and replace outlier slices
residuals	Logical. Output Residuals
session	Character; file path. File containing session indices for all volumes in -imain
slice2vol_interp	Character; one of: "trilinear", "spline". Slice-to-vol interpolation model for estimation step
slice2vol_lambda	Integer. Regularisation weight for slice-to-vol movement (reasonable range 1-10)
slice2vol_niter	Integer. Number of iterations for slice-to-vol
slice_order	Character; file path. Name of text file completely specifying slice/group acquisition
slm	Character; one of: "none", "linear", "quadratic". Second level EC model
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_eddy_correct *FSL EddyCorrect*

Description

.. warning:: Deprecated in FSL. Please use

Usage

```
ni_fsl_eddy_correct(
  in_file,
  ref_num,
  args = NULL,
  out_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. 4D input file Required.
ref_num	Integer. reference number Required.
args	Character. Additional parameters to the command
out_file	Character; file path. 4D output file
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_eddy_quad	<i>FSL EddyQuad</i>
------------------	---------------------

Description

Interface for FSL eddy_quad, a tool for generating single subject reports

Usage

```
ni_fsl_eddy_quad(
    bval_file,
    idx_file,
    mask_file,
    param_file,
    args = NULL,
    base_name = "eddy_corrected",
    bvec_file = NULL,
    field = NULL,
    output_dir = NULL,
    slice_spec = NULL,
    verbose = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)
```

Arguments

bval_file	Character; file path. b-values file Required.
idx_file	Character; file path. File containing indices for all volumes into acquisition parameters Required.
mask_file	Character; file path. Binary mask file Required.
param_file	Character; file path. File containing acquisition parameters Required.
args	Character. Additional parameters to the command
base_name	Character. Basename (including path) for EDDY output files, i.e., corrected images and QC files
bvec_file	Character; file path. b-vectors file - only used when <base_name>.eddy_residuals file is present
field	Character; file path. TOPUP estimated field (in Hz)
output_dir	Character. Output directory - default = '<base_name>.qc'
slice_spec	Character; file path. Text file specifying slice/group acquisition

verbose	Logical. Display debug messages
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

ni_fsl_epi_de_warp	<i>FSL EPIDeWarp</i>
--------------------	----------------------

Description

Wraps the unwarping script 'epidewarp.fsl

Usage

```
ni_fsl_epi_de_warp(
  dph_file,
  mag_file,
  args = NULL,
  cleanup = NULL,
  epi_file = NULL,
  epidw = NULL,
  esp = 0.58,
  exf_file = NULL,
  exfdw = NULL,
  nocleanup = TRUE,
  sigma = 2,
  tediff = 2.46,
  tmpdir = NULL,
  vsm = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

dph_file	Character; file path. Phase file assumed to be scaled from 0 to 4095 Required.
mag_file	Character; file path. Magnitude file Required.
args	Character. Additional parameters to the command
cleanup	Logical. cleanup
epi_file	Character; file path. EPI volume to unwarp
epidw	Character. dewarped epi volume
esp	Numeric. EPI echo spacing
exf_file	Character; file path. example func volume (or use epi)
exfdw	Character. dewarped example func volume
nocleanup	Logical. no cleanup
sigma	Integer. 2D spatial gaussian smoothing stdev (default = 2mm)
tediff	Numeric. difference in B0 field map TEs
tmpdir	Character. tmpdir
vsm	Character. voxel shift map
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_epi_reg

FSL EpiReg

Description

Runs FSL epi_reg script for simultaneous coregistration and fieldmap

Usage

```

ni_fsl_epi_reg(
    epi,
    t1_brain,
    t1_head,
    args = NULL,
    echospacing = NULL,
    fmap = NULL,
    fmapmag = NULL,
    fmapmagbrain = NULL,
    no_clean = TRUE,
    no_fmapreg = NULL,
    out_base = "epi2struct",
    pedir = NULL,
    weight_image = NULL,
    wmseg = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

epi	Character; file path. EPI image Required .
t1_brain	Character; file path. brain extracted T1 image Required .
t1_head	Character; file path. wholehead T1 image Required .
args	Character. Additional parameters to the command
echospacing	Numeric. Effective EPI echo spacing (sometimes called dwell time) - in seconds
fmap	Character; file path. fieldmap image (in rad/s)
fmapmag	Character; file path. fieldmap magnitude image - wholehead
fmapmagbrain	Character; file path. fieldmap magnitude image - brain extracted
no_clean	Logical. do not clean up intermediate files
no_fmapreg	Logical. do not perform registration of fmap to T1 (use if fmap already registered)
out_base	Character. output base name
pedir	Character; one of: "x", "y", "z", "-x", "-y", "-z". phase encoding direction, dir = x/y/z/-x/-y/-z
weight_image	Character; file path. weighting image (in T1 space)
wmseg	Character; file path. white matter segmentation of T1 image, has to be named like the t1brain and end on _wmseg
.cwd	Working directory override.

.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_erode_image	<i>FSL ErodeImage</i>
--------------------	-----------------------

Description

Use fslmaths to perform a spatial erosion of an image.

Usage

```
ni_fsl_erode_image(
  in_file,
  args = NULL,
  internal_datatype = NULL,
  kernel_file = NULL,
  kernel_shape = NULL,
  kernel_size = NULL,
  minimum_filter = FALSE,
  nan2zeros = NULL,
  out_file = NULL,
  output_datatype = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. image to operate on Required.
args	Character. Additional parameters to the command
internal_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for calculations (default is float)

kernel_file	Character; file path. use external file for kernel
kernel_shape	Character; one of: "3D", "2D", "box", "boxv", "gauss", "sphere", "file". kernel shape to use
kernel_size	Numeric. kernel size - voxels for box/boxv, mm for sphere, mm sigma for gauss
minimum_filter	Logical. if true, minimum filter rather than erosion by zeroing-out
nan2zeros	Logical. change NaNs to zeros before doing anything
out_file	Character; file path. image to write
output_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for output (default uses input type)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_extract_roi	<i>FSL ExtractROI</i>
--------------------	-----------------------

Description

Uses FSL Fslroi command to extract region of interest (ROI)

Usage

```
ni_fsl_extract_roi(
  in_file,
  args = NULL,
  crop_list = NULL,
  roi_file = NULL,
  t_min = NULL,
  t_size = NULL,
  x_min = NULL,
  x_size = NULL,
  y_min = NULL,
  y_size = NULL,
  z_min = NULL,
  z_size = NULL,
  .cwd = NULL,
```

```

    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

in_file	Character; file path. input file Required.
args	Character. Additional parameters to the command
crop_list	Character or numeric vector. list of two tuples specifying crop options
roi_file	Character; file path. output file
t_min	Integer
t_size	Integer
x_min	Integer
x_size	Integer
y_min	Integer
y_size	Integer
z_min	Integer
z_size	Integer
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_fast

FSL FAST

Description

FSL FAST wrapper for segmentation and bias correction

Usage

```

ni_fsl_fast(
  in_files,
  args = NULL,
  bias_iters = NULL,
  bias_lowpass = NULL,
  hyper = NULL,
  img_type = NULL,
  init_seg_smooth = NULL,
  init_transform = NULL,
  iters_afterbias = NULL,
  manual_seg = NULL,
  mixel_smooth = NULL,
  no_bias = NULL,
  no_pve = NULL,
  number_classes = NULL,
  other_priors = NULL,
  out_basename = NULL,
  output_biascorrected = NULL,
  output_biasfield = NULL,
  probability_maps = NULL,
  segment_iters = NULL,
  segments = NULL,
  use_priors = NULL,
  verbose = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)

```

Arguments

<code>in_files</code>	Character or numeric vector. image, or multi-channel set of images, to be segmented Required.
<code>args</code>	Character. Additional parameters to the command
<code>bias_iters</code>	Character. number of main-loop iterations during bias-field removal
<code>bias_lowpass</code>	Character. bias field smoothing extent (FWHM) in mm
<code>hyper</code>	Character. segmentation spatial smoothness
<code>img_type</code>	Character; one of: "1", "2", "3". int specifying type of image: (1 = T1, 2 = T2, 3 = PD)
<code>init_seg_smooth</code>	Character. initial segmentation spatial smoothness (during bias field estimation)
<code>init_transform</code>	Character; file path. <standard2input.mat> initialise using priors

iters_afterbias	Character. number of main-loop iterations after bias-field removal
manual_seg	Character; file path. Filename containing intensities
mixel_smooth	Character. spatial smoothness for mixeltype
no_bias	Logical. do not remove bias field
no_pve	Logical. turn off PVE (partial volume estimation)
number_classes	Character. number of tissue-type classes
other_priors	Character or numeric vector. alternative prior images
out_basename	Character; file path. base name of output files
output_biascorrected	Logical. output restored image (bias-corrected image)
output_biasfield	Logical. output estimated bias field
probability_maps	Logical. outputs individual probability maps
segment_iters	Character. number of segmentation-initialisation iterations
segments	Logical. outputs a separate binary image for each tissue type
use_priors	Logical. use priors throughout
verbose	Logical. switch on diagnostic messages
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

ni_fsl_feat

FSL FEAT

Description

Uses FSL feat to calculate first level stats

Usage

```
ni_fsl_feat(
    fsf_file,
    args = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)
```

Arguments

fsf_file	Character; file path. File specifying the feat design spec file Required .
args	Character. Additional parameters to the command
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_feat_model	<i>FSL FEATModel</i>
-------------------	----------------------

Description

Uses FSL feat_model to generate design.mat files

Usage

```
ni_fsl_feat_model(
    ev_files,
    fsf_file,
    args = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)
```

Arguments

ev_files	Character or numeric vector. Event spec files generated by level1design Required.
fsf_file	Character; file path. File specifying the feat design spec file Required.
args	Character. Additional parameters to the command
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_filmgls	<i>FSL FILMGLS</i>
----------------	--------------------

Description

Use FSL film_gls command to fit a design matrix to voxel timeseries

Usage

```
ni_fsl_filmgls(
  in_file,
  args = NULL,
  autocorr_estimate_only = NULL,
  autocorr_noestimate = NULL,
  brightness_threshold = NULL,
  design_file = NULL,
  fit_armodel = NULL,
  full_data = NULL,
  mask_size = NULL,
  multitaper_product = NULL,
  output_pwdata = NULL,
  results_dir = "results",
  smooth_autocorr = NULL,
  threshold = 1000,
  tukey_window = NULL,
  use_pava = NULL,
  .cwd = NULL,
  .env = NULL,
```

```

    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

<code>in_file</code>	Character; file path. input data file Required.
<code>args</code>	Character. Additional parameters to the command
<code>autocorr_estimate_only</code>	Logical. perform autocorrelation estimation only
<code>autocorr_noestimate</code>	Logical. do not estimate autocorrs
<code>brightness_threshold</code>	Character. susan brightness threshold, otherwise it is estimated
<code>design_file</code>	Character; file path. design matrix file
<code>fit_armodel</code>	Logical. fits autoregressive model - default is to use tukey with $M=\sqrt{\text{numvols}}$
<code>full_data</code>	Logical. output full data
<code>mask_size</code>	Integer. susan mask size
<code>multitaper_product</code>	Integer. multitapering with slepian tapers and num is the time-bandwidth product
<code>output_pwdata</code>	Logical. output prewhitened data and average design matrix
<code>results_dir</code>	Character; directory path. directory to store results in
<code>smooth_autocorr</code>	Logical. Smooth auto corr estimates
<code>threshold</code>	Character. threshold
<code>tukey_window</code>	Integer. tukey window size to estimate autocorr
<code>use_pava</code>	Logical. estimates autocorr using PAVA
<code>.cwd</code>	Working directory override.
<code>.env</code>	Named character vector of environment variables.
<code>.engine</code>	Execution engine override.
<code>.profile</code>	Runtime profile override.
<code>dry_run</code>	Logical; preview command without executing.
<code>echo</code>	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

```
ni_fsl_filter_regressor
    FSL FilterRegressor
```

Description

Data de-noising by regressing out part of a design matrix

Usage

```
ni_fsl_filter_regressor(
    design_file,
    filter_all,
    filter_columns,
    in_file,
    args = NULL,
    mask = NULL,
    out_file = NULL,
    out_vnscales = NULL,
    var_norm = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)
```

Arguments

design_file	Character; file path. name of the matrix with time courses (e.g. GLM design or MELODIC mixing matrix) Required.
filter_all	Logical. use all columns in the design file in denoising Required.
filter_columns	Character or numeric vector. (1-based) column indices to filter out of the data Required.
in_file	Character; file path. input file name (4D image) Required.
args	Character. Additional parameters to the command
mask	Character; file path. mask image file name
out_file	Character; file path. output file name for the filtered data
out_vnscales	Logical. output scaling factors for variance normalization
var_norm	Logical. perform variance-normalization on data
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.

.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_find_the_biggest
FSL FindTheBiggest

Description

Use FSL find_the_biggest for performing hard segmentation on

Usage

```
ni_fsl_find_the_biggest(
  in_files,
  args = NULL,
  out_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_files	Character or numeric vector. a list of input volumes or a singleMatrixFile Required.
args	Character. Additional parameters to the command
out_file	Character; file path. file with the resulting segmentation
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_first	<i>FSL FIRST</i>
--------------	------------------

Description

FSL run_first_all wrapper for segmentation of subcortical volumes

Usage

```
ni_fsl_first(
  in_file,
  out_file,
  affine_file = NULL,
  args = NULL,
  brain_extracted = NULL,
  list_of_specific_structures = NULL,
  method = "auto",
  method_as_numerical_threshold = NULL,
  no_cleanup = NULL,
  verbose = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. input data file Required .
out_file	Character; file path. output data file Required .
affine_file	Character; file path. Affine matrix to use (e.g. img2std.mat) (does not re-run registration)
args	Character. Additional parameters to the command
brain_extracted	Logical. Input structural image is already brain-extracted
list_of_specific_structures	Character or numeric vector. Runs only on the specified structures (e.g. L_Hipp, R_HippL_Accu, R_Accu, L_Amyg, R_AmygL_Caud, R_Caud, L_Pall, R_PallL_Puta, R_Puta, L_Thal, R_Thal, BrStem)
method	Character; one of: "auto", "fast", "none". Method must be one of auto, fast, none, or it can be entered using the 'method_as_numerical_threshold' input
method_as_numerical_threshold	Numeric. Specify a numerical threshold value or use the 'method' input to choose auto, fast, or none

no_cleanup	Logical. Input structural image is already brain-extracted
verbose	Logical. Use verbose logging.
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

ni_fsl_flameo	<i>FSL FLAMEO</i>
---------------	-------------------

Description

Use FSL flameo command to perform higher level model fits

Usage

```
ni_fsl_flameo(
  cope_file,
  cov_split_file,
  design_file,
  mask_file,
  run_mode,
  t_con_file,
  args = NULL,
  burnin = NULL,
  dof_var_cope_file = NULL,
  f_con_file = NULL,
  fix_mean = NULL,
  infer_outliers = NULL,
  log_dir = "stats",
  n_jumps = NULL,
  no_pe_outputs = NULL,
  outlier_iter = NULL,
  sample_every = NULL,
  sigma_dofs = NULL,
  var_cope_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
```

```

    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

cope_file	Character; file path. cope regressor data file Required.
cov_split_file	Character; file path. ascii matrix specifying the groups the covariance is split into Required.
design_file	Character; file path. design matrix file Required.
mask_file	Character; file path. mask file Required.
run_mode	Character; one of: "fe", "ols", "flame1", "flame12". inference to perform Required.
t_con_file	Character; file path. ascii matrix specifying t-contrasts Required.
args	Character. Additional parameters to the command
burnin	Integer. number of jumps at start of mcmc to be discarded
dof_var_cope_file	Character; file path. dof data file for varcope data
f_con_file	Character; file path. ascii matrix specifying f-contrasts
fix_mean	Logical. fix mean for tfit
infer_outliers	Logical. infer outliers - not for fe
log_dir	Character; directory path
n_jumps	Integer. number of jumps made by mcmc
no_pe_outputs	Logical. do not output pe files
outlier_iter	Integer. Number of max iterations to use when inferring outliers. Default is 12.
sample_every	Integer. number of jumps for each sample
sigma_dofs	Integer. sigma (in mm) to use for Gaussian smoothing the DOFs in FLAME 2. Default is 1mm, -1 indicates no smoothing
var_cope_file	Character; file path. varcope weightings data file
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

`ni_fsl_flirt`*FSL FLIRT*

Description

FSL FLIRT wrapper for coregistration

Usage

```
ni_fsl_flirt(  
  in_file,  
  reference,  
  angle_rep = NULL,  
  apply_isoxfm = NULL,  
  apply_xfm = NULL,  
  args = NULL,  
  bbrslope = NULL,  
  bbrtype = NULL,  
  bgvalue = NULL,  
  bins = NULL,  
  coarse_search = NULL,  
  cost = NULL,  
  cost_func = NULL,  
  datatype = NULL,  
  display_init = NULL,  
  dof = NULL,  
  echospacing = NULL,  
  fieldmap = NULL,  
  fieldmapmask = NULL,  
  fine_search = NULL,  
  force_scaling = NULL,  
  in_matrix_file = NULL,  
  in_weight = NULL,  
  interp = NULL,  
  min_sampling = NULL,  
  no_clamp = NULL,  
  no_resample = NULL,  
  no_resample_blur = NULL,  
  no_search = NULL,  
  out_file = NULL,  
  out_matrix_file = NULL,  
  padding_size = NULL,  
  pedir = NULL,  
  ref_weight = NULL,  
  rigid2D = NULL,  
  schedule = NULL,  
  searchr_x = NULL,
```

```

    searchr_y = NULL,
    searchr_z = NULL,
    sinc_width = NULL,
    sinc_window = NULL,
    uses_qform = NULL,
    verbose = NULL,
    wm_seg = NULL,
    wmcoords = NULL,
    wmnorms = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

in_file	Character; file path. input file Required .
reference	Character; file path. reference file Required .
angle_rep	Character; one of: "quaternion", "euler". representation of rotation angles
apply_isoxfm	Numeric. as applyxfm but forces isotropic resampling
apply_xfm	Logical. apply transformation supplied by in_matrix_file or uses_qform to use the affine matrix stored in the reference header
args	Character. Additional parameters to the command
bbrslope	Numeric. value of bbr slope
bbrtype	Character; one of: "signed", "global_abs", "local_abs". type of bbr cost function: signed [default], global_abs, local_abs
bgvalue	Numeric. use specified background value for points outside FOV
bins	Integer. number of histogram bins
coarse_search	Integer. coarse search delta angle
cost	Character; one of: "mutualinfo", "corratio", "normcorr", "normmi", "leastsq", "labeldiff", "bbr". cost function
cost_func	Character; one of: "mutualinfo", "corratio", "normcorr", "normmi", "leastsq", "labeldiff", "bbr". cost function
datatype	Character; one of: "char", "short", "int", "float", "double". force output data type
display_init	Logical. display initial matrix
dof	Integer. number of transform degrees of freedom
echospadding	Numeric. value of EPI echo spacing - units of seconds
fieldmap	Character; file path. fieldmap image in rads/s - must be already registered to the reference image

fieldmapmask	Character; file path. mask for fieldmap image
fine_search	Integer. fine search delta angle
force_scaling	Logical. force rescaling even for low-res images
in_matrix_file	Character; file path. input 4x4 affine matrix
in_weight	Character; file path. File for input weighting volume
interp	Character; one of: "trilinear", "nearestneighbour", "sinc", "spline". final interpolation method used in reslicing
min_sampling	Numeric. set minimum voxel dimension for sampling
no_clamp	Logical. do not use intensity clamping
no_resample	Logical. do not change input sampling
no_resample_blur	Logical. do not use blurring on downsampling
no_search	Logical. set all angular searches to ranges 0 to 0
out_file	Character; file path. registered output file
out_matrix_file	Character; file path. output affine matrix in 4x4 asciii format
padding_size	Integer. for applyxfm: interpolates outside image by size
pedir	Integer. phase encode direction of EPI - 1/2/3=x/y/z & -1/-2/-3=-x/-y/-z
ref_weight	Character; file path. File for reference weighting volume
rigid2D	Logical. use 2D rigid body mode - ignores dof
schedule	Character; file path. replaces default schedule
searchr_x	Character or numeric vector. search angles along x-axis, in degrees
searchr_y	Character or numeric vector. search angles along y-axis, in degrees
searchr_z	Character or numeric vector. search angles along z-axis, in degrees
sinc_width	Integer. full-width in voxels
sinc_window	Character; one of: "rectangular", "hanning", "blackman". sinc window
uses_qform	Logical. initialize using sform or qform
verbose	Integer. verbose mode, 0 is least
wm_seg	Character; file path. white matter segmentation volume needed by BBR cost function
wmcoords	Character; file path. white matter boundary coordinates for BBR cost function
wmnorms	Character; file path. white matter boundary normals for BBR cost function
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

`ni_fsl_fnirt`*FSL FNIRT*

Description

FSL FNIRT wrapper for non-linear registration

Usage

```
ni_fsl_fnirt(  
    in_file,  
    ref_file,  
    affine_file = NULL,  
    apply_inmask = NULL,  
    apply_intensity_mapping = NULL,  
    apply_refmask = NULL,  
    args = NULL,  
    bias_regularization_lambda = NULL,  
    biasfield_resolution = NULL,  
    config_file = NULL,  
    derive_from_ref = NULL,  
    field_file = NULL,  
    fieldcoeff_file = NULL,  
    hessian_precision = NULL,  
    in_fwhm = NULL,  
    in_intensitymap_file = NULL,  
    inmask_file = NULL,  
    inmask_val = NULL,  
    intensity_mapping_model = NULL,  
    intensity_mapping_order = NULL,  
    inwarp_file = NULL,  
    jacobian_file = NULL,  
    jacobian_range = NULL,  
    log_file = NULL,  
    max_nonlin_iter = NULL,  
    modulatedref_file = NULL,  
    out_intensitymap_file = NULL,  
    ref_fwhm = NULL,  
    refmask_file = NULL,  
    refmask_val = NULL,  
    regularization_lambda = NULL,  
    regularization_model = NULL,  
    skip_implicit_in_masking = NULL,  
    skip_implicit_ref_masking = NULL,  
    skip_inmask = NULL,  
    skip_intensity_mapping = NULL,  
    skip_lambda_ssq = NULL,
```

```

skip_refmask = NULL,
spline_order = NULL,
subsampling_scheme = NULL,
warp_resolution = NULL,
warped_file = NULL,
.cwd = NULL,
.env = NULL,
.engine = NULL,
.profile = NULL,
dry_run = FALSE,
echo = interactive()
)

```

Arguments

<code>in_file</code>	Character; file path. name of input image Required.
<code>ref_file</code>	Character; file path. name of reference image Required.
<code>affine_file</code>	Character; file path. name of file containing affine transform
<code>apply_inmask</code>	Character or numeric vector. list of iterations to use input mask on (1 to use, 0 to skip)
<code>apply_intensity_mapping</code>	Character or numeric vector. List of subsampling levels to apply intensity mapping for (0 to skip, 1 to apply)
<code>apply_refmask</code>	Character or numeric vector. list of iterations to use reference mask on (1 to use, 0 to skip)
<code>args</code>	Character. Additional parameters to the command
<code>bias_regularization_lambda</code>	Numeric. Weight of regularisation for bias-field, default 10000
<code>biasfield_resolution</code>	Character or numeric vector. Resolution (in mm) of bias-field modelling local intensities, default 50, 50, 50
<code>config_file</code>	Character or numeric vector. Name of config file specifying command line arguments
<code>derive_from_ref</code>	Logical. If true, ref image is used to calculate derivatives. Default false
<code>field_file</code>	Character or numeric vector. name of output file with field or true
<code>fieldcoeff_file</code>	Character or numeric vector. name of output file with field coefficients or true
<code>hessian_precision</code>	Character; one of: "double", "float". Precision for representing Hessian, double or float. Default double
<code>in_fwhm</code>	Character or numeric vector. FWHM (in mm) of gaussian smoothing kernel for input volume, default [6, 4, 2, 2]
<code>in_intensitymap_file</code>	Character or numeric vector. name of file/files containing initial intensity mapping usually generated by previous fnirt run

inmask_file	Character; file path. name of file with mask in input image space
inmask_val	Numeric. Value to mask out in -in image. Default =0.0
intensity_mapping_model	Character; one of: "none", "global_linear", "global_non_linear", "local_linear", "global_non_linear_with_bias", "local_non_linear". Model for intensity-mapping
intensity_mapping_order	Integer. Order of polynomial for mapping intensities, default 5
inwarp_file	Character; file path. name of file containing initial non-linear warps
jacobian_file	Character or numeric vector. name of file for writing out the Jacobian of the field (for diagnostic or VBM purposes)
jacobian_range	Character or numeric vector. Allowed range of Jacobian determinants, default 0.01, 100.0
log_file	Character; file path. Name of log-file
max_nonlin_iter	Character or numeric vector. Max # of non-linear iterations list, default [5, 5, 5, 5]
modulatedref_file	Character or numeric vector. name of file for writing out intensity modulated -ref (for diagnostic purposes)
out_intensitymap_file	Character or numeric vector. name of files for writing information pertaining to intensity mapping
ref_fwhm	Character or numeric vector. FWHM (in mm) of gaussian smoothing kernel for ref volume, default [4, 2, 0, 0]
refmask_file	Character; file path. name of file with mask in reference space
refmask_val	Numeric. Value to mask out in -ref image. Default =0.0
regularization_lambda	Character or numeric vector. Weight of regularisation, default depending on -ssqlambda and -regmod switches. See user documentation.
regularization_model	Character; one of: "membrane_energy", "bending_energy". Model for regularisation of warp-field [membrane_energy bending_energy], default bending_energy
skip_implicit_in_masking	Logical. skip implicit masking based on value in -in image. Default = 0
skip_implicit_ref_masking	Logical. skip implicit masking based on value in -ref image. Default = 0
skip_inmask	Logical. skip specified inmask if set, default false
skip_intensity_mapping	Logical. Skip estimate intensity-mapping default false
skip_lambda_ssq	Logical. If true, lambda is not weighted by current ssq, default false
skip_refmask	Logical. Skip specified refmask if set, default false
spline_order	Integer. Order of spline, 2->Quadratic spline, 3->Cubic spline. Default=3

subsampling_scheme	Character or numeric vector. sub-sampling scheme, list, default [4, 2, 1, 1]
warp_resolution	Character or numeric vector. (approximate) resolution (in mm) of warp basis in x-, y- and z-direction, default 10, 10, 10
warped_file	Character; file path. name of output image
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_fugue

FSL FUGUE

Description

FSL FUGUE set of tools for EPI distortion correction

Usage

```
ni_fsl_fugue(
  args = NULL,
  asym_se_time = NULL,
  despiking_2dfilter = NULL,
  despiking_threshold = NULL,
  dwell_time = NULL,
  dwell_to_asym_ratio = NULL,
  fmap_in_file = NULL,
  fmap_out_file = NULL,
  fourier_order = NULL,
  icorr = NULL,
  icorr_only = NULL,
  in_file = NULL,
  mask_file = NULL,
  median_2dfilter = NULL,
  no_extend = NULL,
  no_gap_fill = NULL,
  nokspace = NULL,
  pava = NULL,
```

```

    phase_conjugate = NULL,
    phasemap_in_file = NULL,
    poly_order = NULL,
    save_unmasked_fmap = NULL,
    save_unmasked_shift = NULL,
    shift_in_file = NULL,
    shift_out_file = NULL,
    smooth2d = NULL,
    smooth3d = NULL,
    unwarp_direction = NULL,
    unwarped_file = NULL,
    warped_file = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

args	Character. Additional parameters to the command
asym_se_time	Numeric. set the fieldmap asymmetric spin echo time (sec)
despike_2dfilter	Logical. apply a 2D de-spiking filter
despike_threshold	Numeric. specify the threshold for de-spiking (default=3.0)
dwel_time	Numeric. set the EPI dwell time per phase-encode line - same as echo spacing - (sec)
dwel_to_asym_ratio	Numeric. set the dwell to asym time ratio
fmap_in_file	Character; file path. filename for loading fieldmap (rad/s)
fmap_out_file	Character; file path. filename for saving fieldmap (rad/s)
fourier_order	Integer. apply Fourier (sinusoidal) fitting of order N
icorr	Logical. apply intensity correction to unwarping (pixel shift method only)
icorr_only	Logical. apply intensity correction only
in_file	Character; file path. filename of input volume
mask_file	Character; file path. filename for loading valid mask
median_2dfilter	Logical. apply 2D median filtering
no_extend	Logical. do not apply rigid-body extrapolation to the fieldmap
no_gap_fill	Logical. do not apply gap-filling measure to the fieldmap
nospace	Logical. do not use k-space forward warping

pava	Logical. apply monotonic enforcement via PAVA
phase_conjugate	Logical. apply phase conjugate method of unwarping
phasemap_in_file	Character; file path. filename for input phase image
poly_order	Integer. apply polynomial fitting of order N
save_unmasked_fmap	Logical. saves the unmasked fieldmap when using <code>--savefmap</code>
save_unmasked_shift	Logical. saves the unmasked shiftmap when using <code>--saveshift</code>
shift_in_file	Character; file path. filename for reading pixel shift volume
shift_out_file	Character; file path. filename for saving pixel shift volume
smooth2d	Numeric. apply 2D Gaussian smoothing of sigma N (in mm)
smooth3d	Numeric. apply 3D Gaussian smoothing of sigma N (in mm)
unwarp_direction	Character; one of: "x", "y", "z", "x-", "y-", "z-". specifies direction of warping (default y)
unwarped_file	Character; file path. apply unwarping and save as filename
warped_file	Character; file path. apply forward warping and save as filename
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

ni_fsl_glm

FSL GLM

Description

FSL GLM:

Usage

```

ni_fsl_glm(
  design,
  in_file,
  args = NULL,
  contrasts = NULL,
  dat_norm = NULL,
  demean = NULL,
  des_norm = NULL,
  dof = NULL,
  mask = NULL,
  out_cope = NULL,
  out_data_name = NULL,
  out_f_name = NULL,
  out_file = NULL,
  out_p_name = NULL,
  out_pf_name = NULL,
  out_res_name = NULL,
  out_sigsq_name = NULL,
  out_t_name = NULL,
  out_varcb_name = NULL,
  out_vnscales_name = NULL,
  out_z_name = NULL,
  var_norm = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)

```

Arguments

design	Character; file path. file name of the GLM design matrix (text time courses for temporal regression or an image file for spatial regression) Required.
in_file	Character; file path. input file name (text matrix or 3D/4D image file) Required.
args	Character. Additional parameters to the command
contrasts	Character; file path. matrix of t-statics contrasts
dat_norm	Logical. switch on normalization of the data time series to unit std deviation
demean	Logical. switch on demeaning of design and data
des_norm	Logical. switch on normalization of the design matrix columns to unit std deviation
dof	Integer. set degrees of freedom explicitly
mask	Character; file path. mask image file name if input is image
out_cope	Character; file path. output file name for COPE (either as txt or image)

out_data_name	Character; file path. output file name for pre-processed data
out_f_name	Character; file path. output file name for F-value of full model fit
out_file	Character; file path. filename for GLM parameter estimates (GLM betas)
out_p_name	Character; file path. output file name for p-values of Z-stats (either as text file or image)
out_pf_name	Character; file path. output file name for p-value for full model fit
out_res_name	Character; file path. output file name for residuals
out_sigsq_name	Character; file path. output file name for residual noise variance sigma-square
out_t_name	Character; file path. output file name for t-stats (either as txt or image)
out_varcb_name	Character; file path. output file name for variance of COPEs
out_vnscales_name	Character; file path. output file name for scaling factors for variance normalisation
out_z_name	Character; file path. output file name for Z-stats (either as txt or image)
var_norm	Logical. perform MELODIC variance-normalisation on data
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

ni_fsl_ica_aroma	<i>FSL ICA_AROMA</i>
------------------	----------------------

Description

Interface for the ICA_AROMA.py script.

Usage

```
ni_fsl_ica_aroma(
  denoise_type,
  feat_dir,
  in_file,
  motion_parameters,
  out_dir,
  TR = NULL,
```

```

    args = NULL,
    dim = NULL,
    fnirt_warp_file = NULL,
    mask = NULL,
    mat_file = NULL,
    melodic_dir = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

denoise_type	Character; one of: "nonaggr", "aggr", "both", "no". Type of denoising strategy: -no: only classification, no denoising -nonaggr (default): non-aggressive denoising, i.e. partial component regression -aggr: aggressive denoising, i.e. full component regression -both: both aggressive and non-aggressive denoising (two outputs) Required.
feat_dir	Character; directory path. If a feat directory exists and temporal filtering has not been run yet, ICA_AROMA can use the files in this directory. Required.
in_file	Character; file path. volume to be denoised Required.
motion_parameters	Character; file path. motion parameters file Required.
out_dir	Character; directory path. output directory Required.
TR	Numeric. TR in seconds. If this is not specified the TR will be extracted from the header of the fMRI nifti file.
args	Character. Additional parameters to the command
dim	Integer. Dimensionality reduction when running MELODIC (default is automatic estimation)
fnirt_warp_file	Character; file path. File name of the warp-file describing the non-linear registration (e.g. FSL FNIRT) of the structural data to MNI152 space (.nii.gz)
mask	Character; file path. path/name volume mask
mat_file	Character; file path. path/name of the mat-file describing the affine registration (e.g. FSL FLIRT) of the functional data to structural space (.mat file)
melodic_dir	Character; directory path. path to MELODIC directory if MELODIC has already been run
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_image_maths	<i>FSL ImageMaths</i>
--------------------	-----------------------

Description

Use FSL fslmaths command to allow mathematical manipulation of images

Usage

```
ni_fsl_image_maths(
  in_file,
  args = NULL,
  in_file2 = NULL,
  mask_file = NULL,
  op_string = NULL,
  out_data_type = NULL,
  out_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path Required.
args	Character. Additional parameters to the command
in_file2	Character; file path
mask_file	Character; file path. use (following image>0) to mask current image
op_string	Character. string defining the operation, i. e. -add
out_data_type	Character; one of: "char", "short", "int", "float", "double", "input". output datatype, one of (char, short, int, float, double, input)
out_file	Character; file path
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_image_meants *FSL ImageMeants*

Description

Use fslmeants for printing the average timeseries (intensities) to

Usage

```
ni_fsl_image_meants(
  in_file,
  args = NULL,
  eig = NULL,
  mask = NULL,
  nobin = NULL,
  order = 1,
  out_file = NULL,
  show_all = NULL,
  spatial_coord = NULL,
  transpose = NULL,
  use_mm = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. input file for computing the average timeseries Required.
args	Character. Additional parameters to the command
eig	Logical. calculate Eigenvariate(s) instead of mean (output will have 0 mean)
mask	Character; file path. input 3D mask
nobin	Logical. do not binarise the mask for calculation of Eigenvariates
order	Integer. select number of Eigenvariates
out_file	Character; file path. name of output text matrix
show_all	Logical. show all voxel time series (within mask) instead of averaging
spatial_coord	Character or numeric vector. requested spatial coordinate (instead of mask)
transpose	Logical. output results in transpose format (one row per voxel/mean)

use_mm	Logical. use mm instead of voxel coordinates (for -c option)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_image_stats	<i>FSL ImageStats</i>
--------------------	-----------------------

Description

Use FSL fslstats command to calculate stats from images

Usage

```
ni_fsl_image_stats(
  in_file,
  op_string,
  args = NULL,
  index_mask_file = NULL,
  split_4d = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. input file to generate stats of Required .
op_string	Character. string defining the operation, options are applied in order, e.g. -M -l 10 -M will report the non-zero mean, apply a threshold and then report the nonzero mean Required .
args	Character. Additional parameters to the command
index_mask_file	Character; file path. generate separate n submasks from indexMask, for index-values 1..n where n is the maximum index value in indexMask, and generate statistics for each submask

split_4d	Logical. give a separate output line for each 3D volume of a 4D timeseries
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

ni_fsl_inv_warp	<i>FSL InvWarp</i>
-----------------	--------------------

Description

Use FSL Invwarp to invert a FNIRT warp

Usage

```
ni_fsl_inv_warp(
  reference,
  warp,
  absolute = NULL,
  args = NULL,
  inverse_warp = NULL,
  jacobian_max = NULL,
  jacobian_min = NULL,
  niter = NULL,
  noconstraint = NULL,
  regularise = NULL,
  relative = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

reference	Character; file path. Name of a file in target space. Note that the target space is now different from the target space that was used to create the <code>-warp</code> file. It would typically be the file that was specified with the <code>-in</code> argument when running <code>fnirt</code> . Required.
warp	Character; file path. Name of file containing warp-coefficients/fields. This would typically be the output from the <code>-cout</code> switch of <code>fnirt</code> (but can also use fields, like the output from <code>-fout</code>). Required.
absolute	Logical. If set it indicates that the warps in <code>-warp</code> should be interpreted as absolute, provided that it is not created by <code>fnirt</code> (which always uses relative warps). If set it also indicates that the output <code>-out</code> should be absolute.
args	Character. Additional parameters to the command
inverse_warp	Character; file path. Name of output file, containing warps that are the "reverse" of those in <code>-warp</code> . This will be a field-file (rather than a file of spline coefficients), and it will have any affine component included as part of the displacements.
jacobian_max	Numeric. Maximum acceptable Jacobian value for constraint (default 100.0)
jacobian_min	Numeric. Minimum acceptable Jacobian value for constraint (default 0.01)
niter	Integer. Determines how many iterations of the gradient-descent search that should be run.
noconstraint	Logical. Do not apply Jacobian constraint
regularise	Numeric. Regularization strength (default=1.0).
relative	Logical. If set it indicates that the warps in <code>-warp</code> should be interpreted as relative. I.e. the values in <code>-warp</code> are displacements from the coordinates in the <code>-ref</code> space. If set it also indicates that the output <code>-out</code> should be relative.
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

```
ni_fsl_isotropic_smooth
    FSL IsotropicSmooth
```

Description

Use fslmaths to spatially smooth an image with a gaussian kernel.

Usage

```
ni_fsl_isotropic_smooth(
    fwhm,
    in_file,
    sigma,
    args = NULL,
    internal_datatype = NULL,
    nan2zeros = NULL,
    out_file = NULL,
    output_datatype = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)
```

Arguments

fwhm	Numeric. fwhm of smoothing kernel [mm] Required.
in_file	Character; file path. image to operate on Required.
sigma	Numeric. sigma of smoothing kernel [mm] Required.
args	Character. Additional parameters to the command
internal_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for calculations (default is float)
nan2zeros	Logical. change NaNs to zeros before doing anything
out_file	Character; file path. image to write
output_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for output (default uses input type)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.

.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

```
ni_fsl_make_dyadic_vectors
    FSL MakeDyadicVectors
```

Description

Create vector volume representing mean principal diffusion direction

Usage

```
ni_fsl_make_dyadic_vectors(
    phi_vol,
    theta_vol,
    args = NULL,
    mask = NULL,
    output = "dyads",
    perc = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)
```

Arguments

phi_vol	Character; file path Required.
theta_vol	Character; file path Required.
args	Character. Additional parameters to the command
mask	Character; file path
output	Character; file path
perc	Numeric. the {perc}% angle of the output cone of uncertainty (output will be in degrees)
.cwd	Working directory override.
.env	Named character vector of environment variables.

.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_maths_command *FSL MathsCommand*

Description

Base support for FSL commands.

Usage

```
ni_fsl_maths_command(
    in_file,
    args = NULL,
    internal_datatype = NULL,
    nan2zeros = NULL,
    out_file = NULL,
    output_datatype = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)
```

Arguments

in_file	Character; file path. image to operate on Required.
args	Character. Additional parameters to the command
internal_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for calculations (default is float)
nan2zeros	Logical. change NaNs to zeros before doing anything
out_file	Character; file path. image to write
output_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for output (default uses input type)

.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_max_image	<i>FSL MaxImage</i>
------------------	---------------------

Description

Use fslmaths to generate a max image across a given dimension.

Usage

```
ni_fsl_max_image(
  in_file,
  args = NULL,
  dimension = "T",
  internal_datatype = NULL,
  nan2zeros = NULL,
  out_file = NULL,
  output_datatype = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. image to operate on Required.
args	Character. Additional parameters to the command
dimension	Character; one of: "T", "X", "Y", "Z". dimension to max across
internal_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for calculations (default is float)
nan2zeros	Logical. change NaNs to zeros before doing anything

out_file	Character; file path. image to write
output_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for output (default uses input type)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_maxn_image	<i>FSL MaxnImage</i>
-------------------	----------------------

Description

Use fslmaths to generate an image of index of max across

Usage

```
ni_fsl_maxn_image(
  in_file,
  args = NULL,
  dimension = "T",
  internal_datatype = NULL,
  nan2zeros = NULL,
  out_file = NULL,
  output_datatype = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. image to operate on Required.
args	Character. Additional parameters to the command
dimension	Character; one of: "T", "X", "Y", "Z". dimension to index max across
internal_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for calculations (default is float)
nan2zeros	Logical. change NaNs to zeros before doing anything
out_file	Character; file path. image to write
output_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for output (default uses input type)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_mcflirt	<i>FSL MCFLIRT</i>
----------------	--------------------

Description

FSL MCFLIRT wrapper for within-modality motion correction

Usage

```
ni_fsl_mcflirt(
  in_file,
  args = NULL,
  bins = NULL,
  cost = NULL,
  dof = NULL,
  init = NULL,
  interpolation = NULL,
  mean_vol = NULL,
  out_file = NULL,
  ref_file = NULL,
```

```

    ref_vol = NULL,
    rotation = NULL,
    save_mats = NULL,
    save_plots = NULL,
    save_rms = NULL,
    scaling = NULL,
    smooth = NULL,
    stages = NULL,
    stats_imgs = NULL,
    use_contour = NULL,
    use_gradient = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

in_file	Character; file path. timeseries to motion-correct Required.
args	Character. Additional parameters to the command
bins	Integer. number of histogram bins
cost	Character; one of: "mutualinfo", "woods", "corratio", "normcorr", "normmi", "leastquares". cost function to optimize
dof	Integer. degrees of freedom for the transformation
init	Character; file path. initial transformation matrix
interpolation	Character; one of: "spline", "nn", "sinc". interpolation method for transformation
mean_vol	Logical. register to mean volume
out_file	Character; file path. file to write
ref_file	Character; file path. target image for motion correction
ref_vol	Integer. volume to align frames to
rotation	Integer. scaling factor for rotation tolerances
save_mats	Logical. save transformation matrices
save_plots	Logical. save transformation parameters
save_rms	Logical. save rms displacement parameters
scaling	Numeric. scaling factor to use
smooth	Numeric. smoothing factor for the cost function
stages	Integer. stages (if 4, perform final search with sinc interpolation)
stats_imgs	Logical. produce variance and std. dev. images
use_contour	Logical. run search on contour images

use_gradient	Logical. run search on gradient images
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_mean_image	<i>FSL MeanImage</i>
-------------------	----------------------

Description

Use fslmaths to generate a mean image across a given dimension.

Usage

```
ni_fsl_mean_image(
  in_file,
  args = NULL,
  dimension = "T",
  internal_datatype = NULL,
  nan2zeros = NULL,
  out_file = NULL,
  output_datatype = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. image to operate on Required .
args	Character. Additional parameters to the command
dimension	Character; one of: "T", "X", "Y", "Z". dimension to mean across
internal_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for calculations (default is float)

nan2zeros	Logical. change NaNs to zeros before doing anything
out_file	Character; file path. image to write
output_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for output (default uses input type)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_median_image	<i>FSL MedianImage</i>
---------------------	------------------------

Description

Use fslmaths to generate a median image across a given dimension.

Usage

```
ni_fsl_median_image(
  in_file,
  args = NULL,
  dimension = "T",
  internal_datatype = NULL,
  nan2zeros = NULL,
  out_file = NULL,
  output_datatype = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. image to operate on Required.
args	Character. Additional parameters to the command
dimension	Character; one of: "T", "X", "Y", "Z". dimension to median across
internal_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for calculations (default is float)
nan2zeros	Logical. change NaNs to zeros before doing anything
out_file	Character; file path. image to write
output_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for output (default uses input type)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_melodic	<i>FSL MELODIC</i>
----------------	--------------------

Description

Multivariate Exploratory Linear Optimised Decomposition into Independent

Usage

```
ni_fsl_melodic(
  in_files,
  ICs = NULL,
  approach = NULL,
  args = NULL,
  bg_image = NULL,
  bg_threshold = NULL,
  cov_weight = NULL,
  dim = NULL,
  dim_est = NULL,
  epsilon = NULL,
```

```
epsilonS = NULL,  
log_power = NULL,  
mask = NULL,  
max_restart = NULL,  
maxit = NULL,  
mignp = NULL,  
mignN = NULL,  
mignp_factor = NULL,  
mignp_shuffle = NULL,  
mix = NULL,  
mm_thresh = NULL,  
no_bet = NULL,  
no_mask = NULL,  
no_mm = NULL,  
non_linearity = NULL,  
num_ICs = NULL,  
out_all = NULL,  
out_dir = NULL,  
out_mean = NULL,  
out_orig = NULL,  
out_pca = NULL,  
out_stats = NULL,  
out_unmix = NULL,  
out_white = NULL,  
pbsc = NULL,  
rem_cmp = NULL,  
remove_deriv = NULL,  
report = NULL,  
report_maps = NULL,  
s_con = NULL,  
s_des = NULL,  
sep_vn = NULL,  
sep_whiten = NULL,  
smode = NULL,  
t_con = NULL,  
t_des = NULL,  
tr_sec = NULL,  
update_mask = NULL,  
var_norm = NULL,  
.cwd = NULL,  
.env = NULL,  
.engine = NULL,  
.profile = NULL,  
dry_run = FALSE,  
echo = interactive()  
)
```

Arguments

in_files	Character or numeric vector. input file names (either single file name or a list) Required.
ICs	Character; file path. filename of the IC components file for mixture modelling
approach	Character. approach for decomposition, 2D: defl, symm (default), 3D: tica (default), concat
args	Character. Additional parameters to the command
bg_image	Character; file path. specify background image for report (default: mean image)
bg_threshold	Numeric. brain/non-brain threshold used to mask non-brain voxels, as a percentage (only if <code>-nobet</code> selected)
cov_weight	Numeric. voxel-wise weights for the covariance matrix (e.g. segmentation information)
dim	Integer. dimensionality reduction into #num dimensions (default: automatic estimation)
dim_est	Character. use specific dim. estimation technique: lap, bic, mdl, aic, mean (default: lap)
epsilon	Numeric. minimum error change
epsilonS	Numeric. minimum error change for rank-1 approximation in TICA
log_power	Logical. calculate log of power for frequency spectrum
mask	Character; file path. file name of mask for thresholding
max_restart	Integer. maximum number of restarts
maxit	Integer. maximum number of iterations before restart
migp	Logical. switch on MIGP data reduction
migpN	Integer. number of internal Eigenmaps
migp_factor	Integer. Internal Factor of mem-threshold relative to number of Eigenmaps (default: 2)
migp_shuffle	Logical. randomise MIGP file order (default: TRUE)
mix	Character; file path. mixing matrix for mixture modelling / filtering
mm_thresh	Numeric. threshold for Mixture Model based inference
no_bet	Logical. switch off BET
no_mask	Logical. switch off masking
no_mm	Logical. switch off mixture modelling on IC maps
non_linearity	Character. nonlinearity: gauss, tanh, pow3, pow4
num_ICs	Integer. number of IC's to extract (for deflation approach)
out_all	Logical. output everything
out_dir	Character; directory path. output directory name
out_mean	Logical. output mean volume
out_orig	Logical. output the original ICs
out_pca	Logical. output PCA results

out_stats	Logical. output thresholded maps and probability maps
out_unmix	Logical. output unmixing matrix
out_white	Logical. output whitening/dewhitening matrices
pbsc	Logical. switch off conversion to percent BOLD signal change
rem_cmp	Character or numeric vector. component numbers to remove
remove_deriv	Logical. removes every second entry in paradigm file (EV derivatives)
report	Logical. generate Melodic web report
report_maps	Character. control string for spatial map images (see slicer)
s_con	Character; file path. t-contrast matrix across subject-domain
s_des	Character; file path. design matrix across subject-domain
sep_vn	Logical. switch off joined variance normalization
sep_whiten	Logical. switch on separate whitening
smode	Character; file path. matrix of session modes for report generation
t_con	Character; file path. t-contrast matrix across time-domain
t_des	Character; file path. design matrix across time-domain
tr_sec	Numeric. TR in seconds
update_mask	Logical. switch off mask updating
var_norm	Logical. switch off variance normalization
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

ni_fsl_merge

FSL Merge

Description

Use `fslmerge` to concatenate images

Usage

```

ni_fsl_merge(
  dimension,
  in_files,
  args = NULL,
  merged_file = NULL,
  tr = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)

```

Arguments

dimension	Character; one of: "t", "x", "y", "z", "a". dimension along which to merge, optionally set tr input when dimension is t Required.
in_files	Character or numeric vector Required.
args	Character. Additional parameters to the command
merged_file	Character; file path
tr	Numeric. use to specify TR in seconds (default is 1.00 sec), overrides dimension and sets it to tr
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_min_image	<i>FSL MinImage</i>
------------------	---------------------

Description

Use fslmaths to generate a minimum image across a given dimension.

Usage

```

ni_fsl_min_image(
  in_file,
  args = NULL,
  dimension = "T",
  internal_datatype = NULL,
  nan2zeros = NULL,
  out_file = NULL,
  output_datatype = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)

```

Arguments

<code>in_file</code>	Character; file path. image to operate on Required .
<code>args</code>	Character. Additional parameters to the command
<code>dimension</code>	Character; one of: "T", "X", "Y", "Z". dimension to min across
<code>internal_datatype</code>	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for calculations (default is float)
<code>nan2zeros</code>	Logical. change NaNs to zeros before doing anything
<code>out_file</code>	Character; file path. image to write
<code>output_datatype</code>	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for output (default uses input type)
<code>.cwd</code>	Working directory override.
<code>.env</code>	Named character vector of environment variables.
<code>.engine</code>	Execution engine override.
<code>.profile</code>	Runtime profile override.
<code>dry_run</code>	Logical; preview command without executing.
<code>echo</code>	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

 ni_fsl_motion_outliers

FSL MotionOutliers

Description

Use FSL `fsl_motion_outliers` http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FSLMotionOutliers_ to find outliers in timeseries (4d) data.

Usage

```
ni_fsl_motion_outliers(
  in_file,
  args = NULL,
  dummy = NULL,
  mask = NULL,
  metric = NULL,
  no_motion_correction = NULL,
  out_file = NULL,
  out_metric_plot = NULL,
  out_metric_values = NULL,
  threshold = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

<code>in_file</code>	Character; file path. unfiltered 4D image Required .
<code>args</code>	Character. Additional parameters to the command
<code>dummy</code>	Integer. number of dummy scans to delete (before running anything and creating EVs)
<code>mask</code>	Character; file path. mask image for calculating metric
<code>metric</code>	Character; one of: "refrms", "dvars", "refmse", "fd", "fdrms". metrics: refrms - RMS intensity difference to reference volume as metric [default metric], refmse - Mean Square Error version of refrms (used in original version of <code>fsl_motion_outliers</code>), dvars - DVARS, fd - frame displacement, fdrms - FD with RMS matrix calculation
<code>no_motion_correction</code>	Logical. do not run motion correction (assumed already done)
<code>out_file</code>	Character; file path. output outlier file name

out_metric_plot	Character; file path. output metric values plot (DVARs etc.) file name
out_metric_values	Character; file path. output metric values (DVARs etc.) file name
threshold	Numeric. specify absolute threshold value (otherwise use box-plot cutoff = P75 + 1.5*IQR)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

`ni_fsl_multi_image_maths`

FSL MultiImageMaths

Description

Use `fslmaths` to perform a sequence of mathematical operations.

Usage

```
ni_fsl_multi_image_maths(
  in_file,
  op_string,
  operand_files,
  args = NULL,
  internal_datatype = NULL,
  nan2zeros = NULL,
  out_file = NULL,
  output_datatype = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. image to operate on Required.
op_string	Character. python formatted string of operations to perform Required.
operand_files	Character or numeric vector. list of file names to plug into op string Required.
args	Character. Additional parameters to the command
internal_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for calculations (default is float)
nan2zeros	Logical. change NaNs to zeros before doing anything
out_file	Character; file path. image to write
output_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for output (default uses input type)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_overlay	<i>FSL Overlay</i>
----------------	--------------------

Description

Use FSL's overlay command to combine background and statistical images

Usage

```
ni_fsl_overlay(
  auto_thresh_bg,
  background_image,
  bg_thresh,
  full_bg_range,
  stat_image,
  stat_thresh,
  args = NULL,
  out_file = NULL,
  out_type = "float",
```

```

    show_negative_stats = NULL,
    stat_image2 = NULL,
    stat_thresh2 = NULL,
    transparency = TRUE,
    use_checkerboard = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

auto_thresh_bg	Logical. automatically threshold the background image Required.
background_image	Character; file path. image to use as background Required.
bg_thresh	Character or numeric vector. min and max values for background intensity Required.
full_bg_range	Logical. use full range of background image Required.
stat_image	Character; file path. statistical image to overlay in color Required.
stat_thresh	Character or numeric vector. min and max values for the statistical overlay Required.
args	Character. Additional parameters to the command
out_file	Character; file path. combined image volume
out_type	Character; one of: "float", "int". write output with float or int
show_negative_stats	Logical. display negative statistics in overlay
stat_image2	Character; file path. second statistical image to overlay in color
stat_thresh2	Character or numeric vector. min and max values for second statistical overlay
transparency	Logical. make overlay colors semi-transparent
use_checkerboard	Logical. use checkerboard mask for overlay
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

```
ni_fsl_percentile_image
    FSL PercentileImage
```

Description

Use fslmaths to generate a percentile image across a given dimension.

Usage

```
ni_fsl_percentile_image(
    in_file,
    args = NULL,
    dimension = "T",
    internal_datatype = NULL,
    nan2zeros = NULL,
    out_file = NULL,
    output_datatype = NULL,
    perc = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)
```

Arguments

in_file	Character; file path. image to operate on Required.
args	Character. Additional parameters to the command
dimension	Character; one of: "T", "X", "Y", "Z". dimension to percentile across
internal_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for calculations (default is float)
nan2zeros	Logical. change NaNs to zeros before doing anything
out_file	Character; file path. image to write
output_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for output (default uses input type)
perc	Character. nth percentile (0-100) of FULL RANGE across dimension
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.

.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_plot_motion_params
FSL PlotMotionParams

Description

Use fsl_tsplot to plot the estimated motion parameters from a

Usage

```
ni_fsl_plot_motion_params(
  in_file,
  in_source,
  plot_type,
  args = NULL,
  out_file = NULL,
  plot_size = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character or numeric vector. file with motion parameters Required.
in_source	Character; one of: "spm", "fsl". which program generated the motion parameter file - fsl, spm Required.
plot_type	Character; one of: "rotations", "translations", "displacement". which motion type to plot - rotations, translations, displacement Required.
args	Character. Additional parameters to the command
out_file	Character; file path. image to write
plot_size	Character or numeric vector. plot image height and width
.cwd	Working directory override.
.env	Named character vector of environment variables.

.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_plot_time_series
FSL PlotTimeSeries

Description

Use fsl_tsplot to create images of time course plots.

Usage

```
ni_fsl_plot_time_series(
  in_file,
  args = NULL,
  labels = NULL,
  legend_file = NULL,
  out_file = NULL,
  plot_finish = NULL,
  plot_range = NULL,
  plot_size = NULL,
  plot_start = NULL,
  sci_notation = NULL,
  title = NULL,
  x_precision = NULL,
  x_units = 1,
  y_max = NULL,
  y_min = NULL,
  y_range = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character or numeric vector. file or list of files with columns of timecourse information Required.
args	Character. Additional parameters to the command
labels	Character or numeric vector. label or list of labels
legend_file	Character; file path. legend file
out_file	Character; file path. image to write
plot_finish	Integer. final column from in-file to plot
plot_range	Character or numeric vector. first and last columns from the in-file to plot
plot_size	Character or numeric vector. plot image height and width
plot_start	Integer. first column from in-file to plot
sci_notation	Logical. switch on scientific notation
title	Character. plot title
x_precision	Integer. precision of x-axis labels
x_units	Integer. scaling units for x-axis (between 1 and length of in file)
y_max	Numeric. maximum y value
y_min	Numeric. minimum y value
y_range	Character or numeric vector. min and max y axis values
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

`ni_fsl_power_spectrum` *FSL PowerSpectrum*

Description

Use FSL PowerSpectrum command for power spectrum estimation.

Usage

```
ni_fsl_power_spectrum(
  in_file,
  args = NULL,
  out_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. input 4D file to estimate the power spectrum Required.
args	Character. Additional parameters to the command
out_file	Character; file path. name of output 4D file for power spectrum
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_prelude	<i>FSL PRELUDE</i>
----------------	--------------------

Description

FSL prelude wrapper for phase unwrapping

Usage

```
ni_fsl_prelude(
  complex_phase_file,
  magnitude_file,
  phase_file,
  args = NULL,
  end = NULL,
```

```

label_file = NULL,
labelprocess2d = NULL,
mask_file = NULL,
num_partitions = NULL,
process2d = NULL,
process3d = NULL,
rawphase_file = NULL,
removeramps = NULL,
savemask_file = NULL,
start = NULL,
threshold = NULL,
unwrapped_phase_file = NULL,
.cwd = NULL,
.env = NULL,
.engine = NULL,
.profile = NULL,
dry_run = FALSE,
echo = interactive()
)

```

Arguments

complex_phase_file	Character; file path. complex phase input volume Required.
magnitude_file	Character; file path. file containing magnitude image Required.
phase_file	Character; file path. raw phase file Required.
args	Character. Additional parameters to the command
end	Integer. final image number to process (default Inf)
label_file	Character; file path. saving the area labels output
labelprocess2d	Logical. does label processing in 2D (slice at a time)
mask_file	Character; file path. filename of mask input volume
num_partitions	Integer. number of phase partitions to use
process2d	Logical. does all processing in 2D (slice at a time)
process3d	Logical. forces all processing to be full 3D
rawphase_file	Character; file path. saving the raw phase output
removeramps	Logical. remove phase ramps during unwrapping
savemask_file	Character; file path. saving the mask volume
start	Integer. first image number to process (default 0)
threshold	Numeric. intensity threshold for masking
unwrapped_phase_file	Character; file path. file containing unwrapped phase
.cwd	Working directory override.
.env	Named character vector of environment variables.

.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

```
ni_fsl_prepare_fieldmap
    FSL PrepareFieldmap
```

Description

Interface for the fsl_prepare_fieldmap script (FSL 5.0)

Usage

```
ni_fsl_prepare_fieldmap(
    delta_TE,
    in_magnitude,
    in_phase,
    args = NULL,
    nocheck = FALSE,
    out_fieldmap = NULL,
    scanner = "SIEMENS",
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)
```

Arguments

delta_TE	Numeric. echo time difference of the fieldmap sequence in ms. (usually 2.46ms in Siemens) Required.
in_magnitude	Character; file path. Magnitude difference map, brain extracted Required.
in_phase	Character; file path. Phase difference map, in SIEMENS format range from 0-4096 or 0-8192) Required.
args	Character. Additional parameters to the command
nocheck	Logical. do not perform sanity checks for image size/range/dimensions
out_fieldmap	Character; file path. output name for prepared fieldmap

scanner	Character. must be SIEMENS
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_prob_track_x	<i>FSL ProbTrackX</i>
---------------------	-----------------------

Description

Use FSL probtrackx for tractography on bedpostx results

Usage

```
ni_fsl_prob_track_x(
  fsamples,
  mask,
  phsamples,
  seed,
  thsamples,
  args = NULL,
  avoid_mp = NULL,
  c_thresh = NULL,
  correct_path_distribution = NULL,
  dist_thresh = NULL,
  fibst = NULL,
  force_dir = TRUE,
  inv_xfm = NULL,
  loop_check = NULL,
  mask2 = NULL,
  mesh = NULL,
  mod_euler = NULL,
  mode = NULL,
  n_samples = 5000,
  n_steps = NULL,
  network = NULL,
  opd = TRUE,
  os2t = NULL,
```

```

out_dir = NULL,
rand_fib = NULL,
random_seed = NULL,
s2tastext = NULL,
sample_random_points = NULL,
samples_base_name = "merged",
seed_ref = NULL,
step_length = NULL,
stop_mask = NULL,
target_masks = NULL,
use_anisotropy = NULL,
verbose = NULL,
waypoints = NULL,
xfm = NULL,
.cwd = NULL,
.env = NULL,
.engine = NULL,
.profile = NULL,
dry_run = FALSE,
echo = interactive()
)

```

Arguments

fsamples	Character or numeric vector Required.
mask	Character; file path. bet binary mask file in diffusion space Required.
phsamples	Character or numeric vector Required.
seed	Character or numeric vector. seed volume(s), or voxel(s) or freesurfer label file Required.
thsamples	Character or numeric vector Required.
args	Character. Additional parameters to the command
avoid_mp	Character; file path. reject pathways passing through locations given by this mask
c_thresh	Numeric. curvature threshold - default=0.2
correct_path_distribution	Logical. correct path distribution for the length of the pathways
dist_thresh	Numeric. discards samples shorter than this threshold (in mm - default=0)
fibst	Integer. force a starting fibre for tracking - default=1, i.e. first fibre orientation. Only works if randfib==0
force_dir	Logical. use the actual directory name given - i.e. do not add + to make a new directory
inv_xfm	Character; file path. transformation matrix taking DTI space to seed space (compulsory when using a warp_field for seeds_to_dti)
loop_check	Logical. perform loop_checks on paths - slower, but allows lower curvature threshold

mask2	Character; file path. second bet binary mask (in diffusion space) in twomask_symm mode
mesh	Character; file path. Freesurfer-type surface descriptor (in ascii format)
mod_euler	Logical. use modified euler streamlining
mode	Character; one of: "simple", "two_mask_symm", "seedmask". options: simple (single seed voxel), seedmask (mask of seed voxels), twomask_symm (two bet binary masks)
n_samples	Integer. number of samples - default=5000
n_steps	Integer. number of steps per sample - default=2000
network	Logical. activate network mode - only keep paths going through at least one seed mask (required if multiple seed masks)
opd	Logical. outputs path distributions
os2t	Logical. Outputs seeds to targets
out_dir	Character; directory path. directory to put the final volumes in
rand_fib	Character; one of: "0", "1", "2", "3". options: 0 - default, 1 - to randomly sample initial fibres (with $f > \text{fibthresh}$), 2 - to sample in proportion fibres (with $f > \text{fibthresh}$) to f , 3 - to sample ALL populations at random (even if $f < \text{fibthresh}$)
random_seed	Integer. random seed
s2tastext	Logical. output seed-to-target counts as a text file (useful when seeding from a mesh)
sample_random_points	Numeric. sample random points within seed voxels
samples_base_name	Character. the rootname/base_name for samples files
seed_ref	Character; file path. reference vol to define seed space in simple mode - diffusion space assumed if absent
step_length	Numeric. step_length in mm - default=0.5
stop_mask	Character; file path. stop tracking at locations given by this mask file
target_masks	Character or numeric vector. list of target masks - required for seeds_to_targets classification
use_anisotropy	Logical. use anisotropy to constrain tracking
verbose	Character; one of: "0", "1", "2". Verbose level, [0-2]. Level 2 is required to output particle files.
waypoints	Character; file path. waypoint mask or ascii list of waypoint masks - only keep paths going through ALL the masks
xfm	Character; file path. transformation matrix taking seed space to DTI space (either FLIRT matrix or FNIRT warp_field) - default is identity
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_prob_track_x2 *FSL ProbTrackX2*

Description

Use FSL probtrackx2 for tractography on bedpostx results

Usage

```
ni_fsl_prob_track_x2(  
  fsamples,  
  mask,  
  phsamples,  
  seed,  
  thsamples,  
  args = NULL,  
  avoid_mp = NULL,  
  c_thresh = NULL,  
  colmask4 = NULL,  
  correct_path_distribution = NULL,  
  dist_thresh = NULL,  
  distthresh1 = NULL,  
  distthresh3 = NULL,  
  fibst = NULL,  
  fopd = NULL,  
  force_dir = TRUE,  
  inv_xfm = NULL,  
  loop_check = NULL,  
  lrtarget3 = NULL,  
  meshspace = NULL,  
  mod_euler = NULL,  
  n_samples = 5000,  
  n_steps = NULL,  
  network = NULL,  
  omatrix1 = NULL,  
  omatrix2 = NULL,  
  omatrix3 = NULL,  
  omatrix4 = NULL,  
  onewaycondition = NULL,  
  opd = TRUE,  
  os2t = NULL,  
  out_dir = NULL,  
  rand_fib = NULL,
```

```

random_seed = NULL,
s2tastext = NULL,
sample_random_points = NULL,
samples_base_name = "merged",
seed_ref = NULL,
simple = NULL,
step_length = NULL,
stop_mask = NULL,
target2 = NULL,
target3 = NULL,
target4 = NULL,
target_masks = NULL,
use_anisotropy = NULL,
verbose = NULL,
waycond = NULL,
wayorder = NULL,
waypoints = NULL,
xfm = NULL,
.cwd = NULL,
.env = NULL,
.engine = NULL,
.profile = NULL,
dry_run = FALSE,
echo = interactive()
)

```

Arguments

fsamples	Character or numeric vector Required.
mask	Character; file path. bet binary mask file in diffusion space Required.
phsamples	Character or numeric vector Required.
seed	Character or numeric vector. seed volume(s), or voxel(s) or freesurfer label file Required.
thsamples	Character or numeric vector Required.
args	Character. Additional parameters to the command
avoid_mp	Character; file path. reject pathways passing through locations given by this mask
c_thresh	Numeric. curvature threshold - default=0.2
colmask4	Character; file path. Mask for columns of matrix4 (default=seed mask)
correct_path_distribution	Logical. correct path distribution for the length of the pathways
dist_thresh	Numeric. discards samples shorter than this threshold (in mm - default=0)
distthresh1	Numeric. Discards samples (in matrix1) shorter than this threshold (in mm - default=0)
distthresh3	Numeric. Discards samples (in matrix3) shorter than this threshold (in mm - default=0)

fibst	Integer. force a starting fibre for tracking - default=1, i.e. first fibre orientation. Only works if randfib==0
fopd	Character; file path. Other mask for binning tract distribution
force_dir	Logical. use the actual directory name given - i.e. do not add + to make a new directory
inv_xfm	Character; file path. transformation matrix taking DTI space to seed space (compulsory when using a warp_field for seeds_to_dti)
loop_check	Logical. perform loop_checks on paths - slower, but allows lower curvature threshold
lrtarget3	Character; file path. Column-space mask used for NxN connectivity matrix
meshspace	Character; one of: "caret", "freesurfer", "first", "vox". Mesh reference space - either "caret" (default) or "freesurfer" or "first" or "vox"
mod_euler	Logical. use modified euler streamlining
n_samples	Integer. number of samples - default=5000
n_steps	Integer. number of steps per sample - default=2000
network	Logical. activate network mode - only keep paths going through at least one seed mask (required if multiple seed masks)
omatrix1	Logical. Output matrix1 - SeedToSeed Connectivity
omatrix2	Logical. Output matrix2 - SeedToLowResMask
omatrix3	Logical. Output matrix3 (NxN connectivity matrix)
omatrix4	Logical. Output matrix4 - DtiMaskToSeed (special Oxford Sparse Format)
onewaycondition	Logical. Apply waypoint conditions to each half tract separately
opd	Logical. outputs path distributions
os2t	Logical. Outputs seeds to targets
out_dir	Character; directory path. directory to put the final volumes in
rand_fib	Character; one of: "0", "1", "2", "3". options: 0 - default, 1 - to randomly sample initial fibres (with f > fibthresh), 2 - to sample in proportion fibres (with f>fibthresh) to f, 3 - to sample ALL populations at random (even if f<fibthresh)
random_seed	Integer. random seed
s2tastext	Logical. output seed-to-target counts as a text file (useful when seeding from a mesh)
sample_random_points	Numeric. sample random points within seed voxels
samples_base_name	Character. the rootname/base_name for samples files
seed_ref	Character; file path. reference vol to define seed space in simple mode - diffusion space assumed if absent
simple	Logical. rack from a list of voxels (seed must be a ASCII list of coordinates)
step_length	Numeric. step_length in mm - default=0.5

stop_mask	Character; file path. stop tracking at locations given by this mask file
target2	Character; file path. Low resolution binary brain mask for storing connectivity distribution in matrix2 mode
target3	Character; file path. Mask used for NxN connectivity matrix (or Nxn if lrtarget3 is set)
target4	Character; file path. Brain mask in DTI space
target_masks	Character or numeric vector. list of target masks - required for seeds_to_targets classification
use_anisotropy	Logical. use anisotropy to constrain tracking
verbose	Character; one of: "0", "1", "2". Verbose level, [0-2]. Level 2 is required to output particle files.
waycond	Character; one of: "OR", "AND". Waypoint condition. Either "AND" (default) or "OR"
wayorder	Logical. Reject streamlines that do not hit waypoints in given order. Only valid if waycond=AND
waypoints	Character; file path. waypoint mask or ascii list of waypoint masks - only keep paths going through ALL the masks
xfm	Character; file path. transformation matrix taking seed space to DTI space (either FLIRT matrix or FNIRT warp_field) - default is identity
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_proj_thresh	<i>FSL ProjThresh</i>
--------------------	-----------------------

Description

Use FSL proj_thresh for thresholding some outputs of probtrack

Usage

```
ni_fsl_proj_thresh(
  in_files,
  threshold,
  args = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_files	Character or numeric vector. a list of input volumes Required.
threshold	Integer. threshold indicating minimum number of seed voxels entering this mask region Required.
args	Character. Additional parameters to the command
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_randomise	<i>FSL Randomise</i>
------------------	----------------------

Description

FSL Randomise: feeds the 4D projected FA data into GLM

Usage

```
ni_fsl_randomise(
  in_file,
  args = NULL,
  base_name = "randomise",
  c_thresh = NULL,
```

```

cm_thresh = NULL,
demean = NULL,
design_mat = NULL,
f_c_thresh = NULL,
f_cm_thresh = NULL,
f_only = NULL,
fcon = NULL,
mask = NULL,
num_perm = NULL,
one_sample_group_mean = NULL,
p_vec_n_dist_files = NULL,
raw_stats_imgs = NULL,
seed = NULL,
show_info_parallel_mode = NULL,
show_total_perms = NULL,
tcon = NULL,
tfce = NULL,
tfce2D = NULL,
tfce_C = NULL,
tfce_E = NULL,
tfce_H = NULL,
var_smooth = NULL,
vox_p_values = NULL,
x_block_labels = NULL,
.cwd = NULL,
.env = NULL,
.engine = NULL,
.profile = NULL,
dry_run = FALSE,
echo = interactive()
)

```

Arguments

in_file	Character; file path. 4D input file Required.
args	Character. Additional parameters to the command
base_name	Character. the rootname that all generated files will have
c_thresh	Numeric. carry out cluster-based thresholding
cm_thresh	Numeric. carry out cluster-mass-based thresholding
demean	Logical. demean data temporally before model fitting
design_mat	Character; file path. design matrix file
f_c_thresh	Numeric. carry out f cluster thresholding
f_cm_thresh	Numeric. carry out f cluster-mass thresholding
f_only	Logical. calculate f-statistics only
fcon	Character; file path. f contrasts file
mask	Character; file path. mask image

num_perm	Integer. number of permutations (default 5000, set to 0 for exhaustive)
one_sample_group_mean	Logical. perform 1-sample group-mean test instead of generic permutation test
p_vec_n_dist_files	Logical. output permutation vector and null distribution text files
raw_stats_imgs	Logical. output raw (unpermuted) statistic images
seed	Integer. specific integer seed for random number generator
show_info_parallel_mode	Logical. print out information required for parallel mode and exit
show_total_perms	Logical. print out how many unique permutations would be generated and exit
tcon	Character; file path. t contrasts file
tfce	Logical. carry out Threshold-Free Cluster Enhancement
tfce2D	Logical. carry out Threshold-Free Cluster Enhancement with 2D optimisation
tfce_C	Numeric. TFCE connectivity (6 or 26; default=6)
tfce_E	Numeric. TFCE extent parameter (default=0.5)
tfce_H	Numeric. TFCE height parameter (default=2)
var_smooth	Integer. use variance smoothing (std is in mm)
vox_p_values	Logical. output voxelwise (corrected and uncorrected) p-value images
x_block_labels	Character; file path. exchangeability block labels file
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

`ni_fsl_reorient2_std` *FSL Reorient2Std*

Description

`fslreorient2std` is a tool for reorienting the image to match the

Usage

```
ni_fsl_reorient2_std(
  in_file,
  args = NULL,
  out_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path Required.
args	Character. Additional parameters to the command
out_file	Character; file path
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_robust_fov	<i>FSL RobustFOV</i>
-------------------	----------------------

Description

Automatically crops an image removing lower head and neck.

Usage

```
ni_fsl_robust_fov(
  in_file,
  args = NULL,
  brainsize = NULL,
  out_roi = NULL,
  out_transform = NULL,
```

```

    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

in_file	Character; file path. input filename Required.
args	Character. Additional parameters to the command
brainsize	Integer. size of brain in z-dimension (default 170mm/150mm)
out_roi	Character; file path. ROI volume output name
out_transform	Character; file path. Transformation matrix in_file to out_roi output name
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_sig_loss	<i>FSL SigLoss</i>
-----------------	--------------------

Description

Estimates signal loss from a field map (in rad/s)

Usage

```

ni_fsl_sig_loss(
  in_file,
  args = NULL,
  echo_time = NULL,
  mask_file = NULL,
  out_file = NULL,
  slice_direction = NULL,
  .cwd = NULL,
  .env = NULL,

```

```

    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

in_file	Character; file path. b0 fieldmap file Required.
args	Character. Additional parameters to the command
echo_time	Numeric. echo time in seconds
mask_file	Character; file path. brain mask file
out_file	Character; file path. output signal loss estimate file
slice_direction	Character; one of: "x", "y", "z". slicing direction
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_slice	<i>FSL Slice</i>
--------------	------------------

Description

Use fslslice to split a 3D file into lots of 2D files (along z-axis).

Usage

```

ni_fsl_slice(
  in_file,
  args = NULL,
  out_base_name = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)

```

Arguments

in_file	Character; file path. input filename Required.
args	Character. Additional parameters to the command
out_base_name	Character. outputs prefix
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_slice_timer	<i>FSL SliceTimer</i>
--------------------	-----------------------

Description

FSL slicetimer wrapper to perform slice timing correction

Usage

```
ni_fsl_slice_timer(
  in_file,
  args = NULL,
  custom_order = NULL,
  custom_timings = NULL,
  global_shift = NULL,
  index_dir = NULL,
  interleaved = NULL,
  out_file = NULL,
  slice_direction = NULL,
  time_repetition = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. filename of input timeseries Required.
args	Character. Additional parameters to the command
custom_order	Character; file path. filename of single-column custom interleave order file (first slice is referred to as 1 not 0)
custom_timings	Character; file path. slice timings, in fractions of TR, range 0:1 (default is 0.5 = no shift)
global_shift	Numeric. shift in fraction of TR, range 0:1 (default is 0.5 = no shift)
index_dir	Logical. slice indexing from top to bottom
interleaved	Logical. use interleaved acquisition
out_file	Character; file path. filename of output timeseries
slice_direction	Character; one of: "1", "2", "3". direction of slice acquisition (x=1, y=2, z=3) - default is z
time_repetition	Numeric. Specify TR of data - default is 3s
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_slicer

FSL Slicer

Description

Use FSL's slicer command to output a png image from a volume.

Usage

```
ni_fsl_slicer(
  in_file,
  all_axial = NULL,
  args = NULL,
  colour_map = NULL,
  dither_edges = NULL,
```

```

image_edges = NULL,
image_width = NULL,
intensity_range = NULL,
label_slices = TRUE,
middle_slices = NULL,
nearest_neighbour = NULL,
out_file = NULL,
sample_axial = NULL,
scaling = NULL,
show_orientation = TRUE,
single_slice = NULL,
slice_number = NULL,
threshold_edges = NULL,
.cwd = NULL,
.env = NULL,
.engine = NULL,
.profile = NULL,
dry_run = FALSE,
echo = interactive()
)

```

Arguments

<code>in_file</code>	Character; file path. input volume Required.
<code>all_axial</code>	Logical. output all axial slices into one picture
<code>args</code>	Character. Additional parameters to the command
<code>colour_map</code>	Character; file path. use different colour map from that stored in nifti header
<code>dither_edges</code>	Logical. produce semi-transparent (dithered) edges
<code>image_edges</code>	Character; file path. volume to display edge overlay for (useful for checking registration)
<code>image_width</code>	Integer. max picture width
<code>intensity_range</code>	Character or numeric vector. min and max intensities to display
<code>label_slices</code>	Logical. display slice number
<code>middle_slices</code>	Logical. output picture of mid-sagittal, axial, and coronal slices
<code>nearest_neighbour</code>	Logical. use nearest neighbor interpolation for output
<code>out_file</code>	Character; file path. picture to write
<code>sample_axial</code>	Integer. output every n axial slices into one picture
<code>scaling</code>	Numeric. image scale
<code>show_orientation</code>	Logical. label left-right orientation
<code>single_slice</code>	Character; one of: "x", "y", "z". output picture of single slice in the x, y, or z plane

slice_number	Integer. slice number to save in picture
threshold_edges	Numeric. use threshold for edges
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_smm	<i>FSL SMM</i>
------------	----------------

Description

Spatial Mixture Modelling. For more detail on the spatial mixture modelling

Usage

```
ni_fsl_smm(
  mask,
  spatial_data_file,
  args = NULL,
  no_deactivation_class = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

mask	Character; file path. mask file Required.
spatial_data_file	Character; file path. statistics spatial map Required.
args	Character. Additional parameters to the command
no_deactivation_class	Logical. enforces no deactivation class

.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_smooth	<i>FSL Smooth</i>
---------------	-------------------

Description

Use fslmaths to smooth the image

Usage

```
ni_fsl_smooth(
  fwhm,
  in_file,
  sigma,
  args = NULL,
  smoothed_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

fwhm	Numeric. gaussian kernel fwhm, will be converted to sigma in mm (not voxels) Required.
in_file	Character; file path Required.
sigma	Numeric. gaussian kernel sigma in mm (not voxels) Required.
args	Character. Additional parameters to the command
smoothed_file	Character; file path
.cwd	Working directory override.
.env	Named character vector of environment variables.

.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_smooth_estimate
FSL SmoothEstimate

Description

Estimates the smoothness of an image

Usage

```
ni_fsl_smooth_estimate(
  dof,
  mask_file,
  args = NULL,
  residual_fit_file = NULL,
  zstat_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

dof	Integer. number of degrees of freedom Required.
mask_file	Character; file path. brain mask volume Required.
args	Character. Additional parameters to the command
residual_fit_file	Character; file path. residual-fit image file
zstat_file	Character; file path. zstat image file
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_spatial_filter *FSL SpatialFilter*

Description

Use fslmaths to spatially filter an image.

Usage

```
ni_fsl_spatial_filter(
  in_file,
  operation,
  args = NULL,
  internal_datatype = NULL,
  kernel_file = NULL,
  kernel_shape = NULL,
  kernel_size = NULL,
  nan2zeros = NULL,
  out_file = NULL,
  output_datatype = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. image to operate on Required.
operation	Character; one of: "mean", "median", "meanu". operation to filter with Required.
args	Character. Additional parameters to the command
internal_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for calculations (default is float)
kernel_file	Character; file path. use external file for kernel
kernel_shape	Character; one of: "3D", "2D", "box", "boxv", "gauss", "sphere", "file". kernel shape to use
kernel_size	Numeric. kernel size - voxels for box/boxv, mm for sphere, mm sigma for gauss
nan2zeros	Logical. change NaNs to zeros before doing anything

out_file	Character; file path. image to write
output_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for output (default uses input type)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_split	<i>FSL Split</i>
--------------	------------------

Description

Uses FSL Fslsplit command to separate a volume into images in

Usage

```
ni_fsl_split(
  dimension,
  in_file,
  args = NULL,
  out_base_name = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

dimension	Character; one of: "t", "x", "y", "z". dimension along which the file will be split Required.
in_file	Character; file path. input filename Required.
args	Character. Additional parameters to the command
out_base_name	Character. outputs prefix

.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_std_image	<i>FSL StdImage</i>
------------------	---------------------

Description

Use fslmaths to generate a standard deviation in an image across a given

Usage

```
ni_fsl_std_image(
  in_file,
  args = NULL,
  dimension = "T",
  internal_datatype = NULL,
  nan2zeros = NULL,
  out_file = NULL,
  output_datatype = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. image to operate on Required.
args	Character. Additional parameters to the command
dimension	Character; one of: "T", "X", "Y", "Z". dimension to standard deviate across
internal_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for calculations (default is float)
nan2zeros	Logical. change NaNs to zeros before doing anything

out_file	Character; file path. image to write
output_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for output (default uses input type)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_susan	<i>FSL SUSAN</i>
--------------	------------------

Description

FSL SUSAN wrapper to perform smoothing

Usage

```
ni_fsl_susan(
  brightness_threshold,
  fwhm,
  in_file,
  args = NULL,
  dimension = 3,
  out_file = NULL,
  use_median = 1,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

brightness_threshold	Numeric. brightness threshold and should be greater than noise level and less than contrast of edges to be preserved. Required.
fwhm	Numeric. fwhm of smoothing, in mm, gets converted using $\sqrt{8 \cdot \log(2)}$ Required.
in_file	Character; file path. filename of input timeseries Required.
args	Character. Additional parameters to the command
dimension	Character; one of: "3", "2". within-plane (2) or fully 3D (3)
out_file	Character; file path. output file name
use_median	Character; one of: "1", "0". whether to use a local median filter in the cases where single-point noise is detected
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_swap_dimensions
FSL SwapDimensions

Description

Use fslswapdim to alter the orientation of an image.

Usage

```
ni_fsl_swap_dimensions(
  in_file,
  new_dims,
  args = NULL,
  out_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. input image Required.
new_dims	Character or numeric vector. 3-tuple of new dimension order Required.
args	Character. Additional parameters to the command
out_file	Character; file path. image to write
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_temporal_filter

FSL TemporalFilter

Description

Use fslmaths to apply a low, high, or bandpass temporal filter to a

Usage

```
ni_fsl_temporal_filter(
  in_file,
  args = NULL,
  highpass_sigma = -1,
  internal_datatype = NULL,
  lowpass_sigma = -1,
  nan2zeros = NULL,
  out_file = NULL,
  output_datatype = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. image to operate on Required.
args	Character. Additional parameters to the command
highpass_sigma	Numeric. highpass filter sigma (in volumes)
internal_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for calculations (default is float)
lowpass_sigma	Numeric. lowpass filter sigma (in volumes)
nan2zeros	Logical. change NaNs to zeros before doing anything
out_file	Character; file path. image to write
output_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for output (default uses input type)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_text2_vest	<i>FSL Text2Vest</i>
-------------------	----------------------

Description

Use FSL Text2Vest [https://web.mit.edu/fsl_v5.0.10/fsl/doc/wiki/GLM\(2f\)CreatingDesignMatricesByHand.htm](https://web.mit.edu/fsl_v5.0.10/fsl/doc/wiki/GLM(2f)CreatingDesignMatricesByHand.htm)

Usage

```
ni_fsl_text2_vest(
  in_file,
  out_file,
  args = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. plain text file representing your design, contrast, or f-test matrix Required.
out_file	Character; file path. file name to store matrix data in the format used by FSL tools (e.g., design.mat, design.con design.fts) Required.
args	Character. Additional parameters to the command
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_threshold	<i>FSL Threshold</i>
------------------	----------------------

Description

Use fslmaths to apply a threshold to an image in a variety of ways.

Usage

```
ni_fsl_threshold(
  in_file,
  thresh,
  args = NULL,
  internal_datatype = NULL,
  nan2zeros = NULL,
  out_file = NULL,
  output_datatype = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. image to operate on Required.
thresh	Numeric. threshold value Required.
args	Character. Additional parameters to the command
internal_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for calculations (default is float)
nan2zeros	Logical. change NaNs to zeros before doing anything
out_file	Character; file path. image to write
output_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for output (default uses input type)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_topup	<i>FSL TOPUP</i>
--------------	------------------

Description

Interface for FSL topup, a tool for estimating and correcting

Usage

```
ni_fsl_topup(
  encoding_direction,
  encoding_file,
  in_file,
  readout_times,
  args = NULL,
  config = "b02b0.cnf",
  estmov = NULL,
  fwhm = NULL,
  interp = NULL,
  max_iter = NULL,
```

```

minmet = NULL,
numprec = NULL,
out_base = NULL,
out_corrected = NULL,
out_field = NULL,
out_jac_prefix = "jac",
out_logfile = NULL,
out_mat_prefix = "xfm",
out_warp_prefix = "warpfield",
reg_lambda = NULL,
regmod = NULL,
regrid = NULL,
scale = NULL,
splineorder = NULL,
ssqlambda = NULL,
subsamp = NULL,
warp_res = NULL,
.cwd = NULL,
.env = NULL,
.engine = NULL,
.profile = NULL,
dry_run = FALSE,
echo = interactive()
)

```

Arguments

encoding_direction	Character or numeric vector. encoding direction for automatic generation of encoding_file Required.
encoding_file	Character; file path. name of text file with PE directions/times Required.
in_file	Character; file path. name of 4D file with images Required.
readout_times	Character or numeric vector. readout times (dwell times by # phase-encode steps minus 1) Required.
args	Character. Additional parameters to the command
config	Character. Name of config file specifying command line arguments
estmov	Character; one of: "1", "0". estimate movements if set
fwhm	Numeric. FWHM (in mm) of gaussian smoothing kernel
interp	Character; one of: "spline", "linear". Image interpolation model, linear or spline.
max_iter	Integer. max # of non-linear iterations
minmet	Character; one of: "0", "1". Minimisation method 0=Levenberg-Marquardt, 1=Scaled Conjugate Gradient
numprec	Character; one of: "double", "float". Precision for representing Hessian, double or float.
out_base	Character; file path. base-name of output files (spline coefficients (Hz) and movement parameters)

out_corrected	Character; file path. name of 4D image file with unwarped images
out_field	Character; file path. name of image file with field (Hz)
out_jac_prefix	Character. prefix for the warpfield images
out_logfile	Character; file path. name of log-file
out_mat_prefix	Character. prefix for the realignment matrices
out_warp_prefix	Character. prefix for the warpfield images (in mm)
reg_lambda	Numeric. Weight of regularisation, default depending on <code>--ssqlambda</code> and <code>--regmod</code> switches.
regmod	Character; one of: "bending_energy", "membrane_energy". Regularisation term implementation. Defaults to bending_energy. Note that the two functions have vastly different scales. The membrane energy is based on the first derivatives and the bending energy on the second derivatives. The second derivatives will typically be much smaller than the first derivatives, so input lambda will have to be larger for bending_energy to yield approximately the same level of regularisation.
regrid	Character; one of: "1", "0". If set (=1), the calculations are done in a different grid
scale	Character; one of: "0", "1". If set (=1), the images are individually scaled to a common mean
splineorder	Integer. order of spline, 2->Quadratic spline, 3->Cubic spline
ssqlambda	Character; one of: "1", "0". Weight lambda by the current value of the ssd. If used (=1), the effective weight of regularisation term becomes higher for the initial iterations, therefore initial steps are a little smoother than they would without weighting. This reduces the risk of finding a local minimum.
subsamp	Integer. sub-sampling scheme
warp_res	Numeric. (approximate) resolution (in mm) of warp basis for the different sub-sampling levels
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

 ni_fsl_tract_skeleton *FSL TractSkeleton*

Description

Use FSL's `tbss_skeleton` to skeletonise an FA image or project arbitrary

Usage

```
ni_fsl_tract_skeleton(
  in_file,
  alt_data_file = NULL,
  alt_skeleton = NULL,
  args = NULL,
  project_data = NULL,
  skeleton_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

<code>in_file</code>	Character; file path. input image (typically mean FA volume) Required.
<code>alt_data_file</code>	Character; file path. 4D non-FA data to project onto skeleton
<code>alt_skeleton</code>	Character; file path. alternate skeleton to use
<code>args</code>	Character. Additional parameters to the command
<code>project_data</code>	Logical. project data onto skeleton
<code>skeleton_file</code>	Character or numeric vector. write out skeleton image
<code>.cwd</code>	Working directory override.
<code>.env</code>	Named character vector of environment variables.
<code>.engine</code>	Execution engine override.
<code>.profile</code>	Runtime profile override.
<code>dry_run</code>	Logical; preview command without executing.
<code>echo</code>	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

 ni_fsl_unary_maths *FSL UnaryMaths*

Description

Use fslmaths to perform a variety of mathematical operations on an image.

Usage

```
ni_fsl_unary_maths(
  in_file,
  operation,
  args = NULL,
  internal_datatype = NULL,
  nan2zeros = NULL,
  out_file = NULL,
  output_datatype = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. image to operate on Required .
operation	Character; one of: "exp", "log", "sin", "cos", "tan", "asin", "acos", "atan", "sqr", "sqrt", "recip", "abs", "bin", "binv", "fillh", "fillh26", "index", "edge", "nan", "nanm", "rand", "randn", "range". operation to perform Required .
args	Character. Additional parameters to the command
internal_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for calculations (default is float)
nan2zeros	Logical. change NaNs to zeros before doing anything
out_file	Character; file path. image to write
output_datatype	Character; one of: "float", "char", "int", "short", "double", "input". datatype to use for output (default uses input type)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_vec_reg	<i>FSL VecReg</i>
----------------	-------------------

Description

Use FSL vecreg for registering vector data

Usage

```
ni_fsl_vec_reg(
  in_file,
  ref_vol,
  affine_mat = NULL,
  args = NULL,
  interpolation = NULL,
  mask = NULL,
  out_file = NULL,
  ref_mask = NULL,
  rotation_mat = NULL,
  rotation_warp = NULL,
  warp_field = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. filename for input vector or tensor field Required.
ref_vol	Character; file path. filename for reference (target) volume Required.
affine_mat	Character; file path. filename for affine transformation matrix
args	Character. Additional parameters to the command
interpolation	Character; one of: "nearestneighbour", "trilinear", "sinc", "spline". interpolation method : nearestneighbour, trilinear (default), sinc or spline
mask	Character; file path. brain mask in input space
out_file	Character; file path. filename for output registered vector or tensor field
ref_mask	Character; file path. brain mask in output space (useful for speed up of nonlinear reg)

rotation_mat	Character; file path. filename for secondary affine matrix if set, this will be used for the rotation of the vector/tensor field
rotation_warp	Character; file path. filename for secondary warp field if set, this will be used for the rotation of the vector/tensor field
warp_field	Character; file path. filename for 4D warp field for nonlinear registration
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_vest2_text	<i>FSL Vest2Text</i>
-------------------	----------------------

Description

Use FSL Vest2Text [https://web.mit.edu/fsl_v5.0.10/fsl/doc/wiki/GLM\(2f\)CreatingDesignMatricesByHand.htm](https://web.mit.edu/fsl_v5.0.10/fsl/doc/wiki/GLM(2f)CreatingDesignMatricesByHand.htm)

Usage

```
ni_fsl_vest2_text(
  in_file,
  args = NULL,
  out_file = "design.txt",
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

in_file	Character; file path. matrix data stored in the format used by FSL tools Required.
args	Character. Additional parameters to the command
out_file	Character; file path. file name to store text output from matrix
.cwd	Working directory override.

.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

<code>ni_fsl_warp_points</code>	<i>FSL WarpPoints</i>
---------------------------------	-----------------------

Description

Use FSL `img2imgcoord` <<http://fsl.fmrib.ox.ac.uk/fsl/fsl-4.1.9/flirt/overview.html>>_

Usage

```
ni_fsl_warp_points(
  dest_file,
  in_coords,
  src_file,
  args = NULL,
  coord_mm = NULL,
  coord_vox = NULL,
  warp_file = NULL,
  xfm_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

<code>dest_file</code>	Character; file path. filename of destination image Required.
<code>in_coords</code>	Character; file path. filename of file containing coordinates Required.
<code>src_file</code>	Character; file path. filename of source image Required.
<code>args</code>	Character. Additional parameters to the command
<code>coord_mm</code>	Logical. all coordinates in mm
<code>coord_vox</code>	Logical. all coordinates in voxels - default

warp_file	Character; file path. filename of warpfield (e.g. intermediate2dest_warp.nii.gz)
xfm_file	Character; file path. filename of affine transform (e.g. source2dest.mat)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_warp_points_from_std
FSL WarpPointsFromStd

Description

Use FSL std2imgcoord <<http://fsl.fmrib.ox.ac.uk/fsl/fsl-4.1.9/flirt/overview.html>>_

Usage

```
ni_fsl_warp_points_from_std(
  img_file,
  in_coords,
  std_file,
  args = NULL,
  coord_mm = NULL,
  coord_vox = NULL,
  warp_file = NULL,
  xfm_file = NULL,
  .cwd = NULL,
  .env = NULL,
  .engine = NULL,
  .profile = NULL,
  dry_run = FALSE,
  echo = interactive()
)
```

Arguments

img_file	Character; file path. filename of a destination image Required.
in_coords	Character; file path. filename of file containing coordinates Required.
std_file	Character; file path. filename of the image in standard space Required.
args	Character. Additional parameters to the command
coord_mm	Logical. all coordinates in mm
coord_vox	Logical. all coordinates in voxels - default
warp_file	Character; file path. filename of warpfield (e.g. intermediate2dest_warp.nii.gz)
xfm_file	Character; file path. filename of affine transform (e.g. source2dest.mat)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_warp_points_to_std
FSL WarpPointsToStd

Description

Use FSL img2stdcoord <<http://fsl.fmrib.ox.ac.uk/fsl/fsl-4.1.9/flirt/overview.html>>_

Usage

```
ni_fsl_warp_points_to_std(
  img_file,
  in_coords,
  std_file,
  args = NULL,
  coord_mm = NULL,
  coord_vox = NULL,
  premat_file = NULL,
  warp_file = NULL,
  xfm_file = NULL,
  .cwd = NULL,
  .env = NULL,
```

```

    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

img_file	Character; file path. filename of input image Required.
in_coords	Character; file path. filename of file containing coordinates Required.
std_file	Character; file path. filename of destination image Required.
args	Character. Additional parameters to the command
coord_mm	Logical. all coordinates in mm
coord_vox	Logical. all coordinates in voxels - default
premat_file	Character; file path. filename of pre-warp affine transform (e.g. example_func2highres.mat)
warp_file	Character; file path. filename of warpfield (e.g. intermediate2dest_warp.nii.gz)
xfm_file	Character; file path. filename of affine transform (e.g. source2dest.mat)
.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_warp_utils	<i>FSL WarpUtils</i>
-------------------	----------------------

Description

Use FSL fnirtfileutils <http://fsl.fmrib.ox.ac.uk/fsl/fsl-4.1.9/fnirt/warp_utils.html>_

Usage

```

ni_fsl_warp_utils(
  in_file,
  reference,
  write_jacobian,
  args = NULL,
  knot_space = NULL,

```

```

    out_file = NULL,
    out_format = NULL,
    out_jacobian = NULL,
    warp_resolution = NULL,
    with_affine = NULL,
    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

in_file	Character; file path. Name of file containing warp-coefficients/fields. This would typically be the output from the <code>-cout</code> switch of <code>fnirt</code> (but can also use fields, like the output from <code>-fout</code>). Required.
reference	Character; file path. Name of a file in target space. Note that the target space is now different from the target space that was used to create the <code>-warp</code> file. It would typically be the file that was specified with the <code>-in</code> argument when running <code>fnirt</code> . Required.
write_jacobian	Logical. Switch on <code>-jac</code> flag with automatically generated filename Required.
args	Character. Additional parameters to the command
knot_space	Character or numeric vector. Alternative (to <code>-warpres</code>) specification of the resolution of the output spline-field.
out_file	Character; file path. Name of output file. The format of the output depends on what other parameters are set. The default format is a (4D) field-file. If the <code>-outformat</code> is set to <code>spline</code> the format will be a (4D) file of spline coefficients.
out_format	Character; one of: "spline", "field". Specifies the output format. If set to <code>field</code> (default) the output will be a (4D) field-file. If set to <code>spline</code> the format will be a (4D) file of spline coefficients.
out_jacobian	Character; file path. Specifies that a (3D) file of Jacobian determinants corresponding to <code>-in</code> should be produced and written to filename.
warp_resolution	Character or numeric vector. Specifies the resolution/knot-spacing of the splines pertaining to the coefficients in the <code>-out</code> file. This parameter is only relevant if <code>-outformat</code> is set to <code>spline</code> . It should be noted that if the <code>-in</code> file has a higher resolution, the resulting coefficients will pertain to the closest (in a least-squares sense) file in the space of fields with the <code>-warpres</code> resolution. It should also be noted that the resolution will always be an integer multiple of the voxel size.
with_affine	Logical. Specifies that the affine transform (i.e. that which was specified for the <code>-aff</code> parameter in <code>fnirt</code>) should be included as displacements in the <code>-out</code> file. That can be useful for interfacing with software that cannot decode FSL/ <code>fnirt</code> coefficient-files (where the affine transform is stored separately from the displacements).

.cwd	Working directory override.
.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An ni_result object.

ni_fsl_x_fibres5	<i>FSL XFibres5</i>
------------------	---------------------

Description

Perform model parameters estimation for local (voxelwise) diffusion

Usage

```
ni_fsl_x_fibres5(
  bvals,
  bvecs,
  dwi,
  mask,
  n_fibres,
  all_ard = NULL,
  args = NULL,
  burn_in = 0,
  burn_in_no_ard = 0,
  clinear = NULL,
  f0_ard = NULL,
  f0_noard = NULL,
  force_dir = TRUE,
  fudge = NULL,
  gradnonlin = NULL,
  logdir = ".",
  model = NULL,
  n_jumps = 5000,
  no_ard = NULL,
  no_spat = NULL,
  non_linear = NULL,
  rician = NULL,
  sample_every = 1,
  seed = NULL,
  update_proposal_every = 40,
```

```

    .cwd = NULL,
    .env = NULL,
    .engine = NULL,
    .profile = NULL,
    dry_run = FALSE,
    echo = interactive()
)

```

Arguments

bvals	Character; file path. b values file Required.
bvecs	Character; file path. b vectors file Required.
dwi	Character; file path. diffusion weighted image data file Required.
mask	Character; file path. brain binary mask file (i.e. from BET) Required.
n_fibres	Character. Maximum number of fibres to fit in each voxel Required.
all_ard	Logical. Turn ARD on on all fibres
args	Character. Additional parameters to the command
burn_in	Character. Total num of jumps at start of MCMC to be discarded
burn_in_no_ard	Character. num of burnin jumps before the ard is imposed
cnlinear	Logical. Initialise with constrained nonlinear fitting
f0_ard	Logical. Noise floor model: add to the model an unattenuated signal compartment f0
f0_noard	Logical. Noise floor model: add to the model an unattenuated signal compartment f0
force_dir	Logical. use the actual directory name given (do not add + to make a new directory)
fudge	Integer. ARD fudge factor
gradnonlin	Character; file path. gradient file corresponding to slice
logdir	Character; directory path
model	Character; one of: "1", "2", "3". use monoexponential (1, default, required for single-shell) or multiexponential (2, multi-shell) model
n_jumps	Integer. Num of jumps to be made by MCMC
no_ard	Logical. Turn ARD off on all fibres
no_spat	Logical. Initialise with tensor, not spatially
non_linear	Logical. Initialise with nonlinear fitting
rician	Logical. use Rician noise modeling
sample_every	Character. Num of jumps for each sample (MCMC)
seed	Integer. seed for pseudo random number generator
update_proposal_every	Character. Num of jumps for each update to the proposal density std (MCMC)
.cwd	Working directory override.

.env	Named character vector of environment variables.
.engine	Execution engine override.
.profile	Runtime profile override.
dry_run	Logical; preview command without executing.
echo	Logical; echo stdout/stderr in real time.

Value

An `ni_result` object.

`ni_golden_cmdline_generate`

Generate golden command fixtures for all specs

Description

Builds deterministic command/argument snapshots for each spec and writes a single JSON fixture suitable for regression testing.

Usage

```
ni_golden_cmdline_generate(  
  output = "tests/golden/cmdline_golden.json",  
  spec_ids = NULL,  
  spec_dir = "inst/specs"  
)
```

Arguments

<code>output</code>	Path to the golden JSON fixture file.
<code>spec_ids</code>	Optional spec IDs. Defaults to all bundled specs.
<code>spec_dir</code>	Directory containing specs when running from source.

Value

Path written (invisibly).

ni_inputs	<i>List spec inputs as a table</i>
-----------	------------------------------------

Description

List spec inputs as a table

Usage

```
ni_inputs(spec_id)
```

Arguments

spec_id Spec ID, spec path, or ni_spec object.

Value

A data frame (or tibble, if available) describing input parameters.

ni_lint_specs	<i>Lint niflowr specs with optional autofix</i>
---------------	---

Description

Performs mechanical checks on spec JSON files (or parsed specs), including: shell metacharacters in CLI arg strings, positional collisions, invalid flag formats, and broken constraint references.

Usage

```
ni_lint_specs(
  spec_dir = "inst/specs",
  spec_paths = NULL,
  strict = FALSE,
  fix = FALSE,
  write = fix
)
```

Arguments

spec_dir Directory containing spec JSON files. Ignored if spec_paths is provided.

spec_paths Optional character vector of spec file paths to lint.

strict Logical; if TRUE, abort when lint errors remain.

fix Logical; if TRUE, apply safe autofixes in-memory.

write Logical; if TRUE and fix = TRUE, write modified specs back to disk.

Value

A data frame (or tibble, if available) with lint findings.

ni_lock_read	<i>Read lockfile</i>
--------------	----------------------

Description

Read lockfile

Usage

```
ni_lock_read(path = NULL)
```

Arguments

path Lockfile path. Defaults to config runtime.lockfile.

Value

Parsed lockfile list.

ni_lock_validate	<i>Validate current runtime config against lockfile</i>
------------------	---

Description

Validate current runtime config against lockfile

Usage

```
ni_lock_validate(
  path = NULL,
  cfg = NULL,
  profiles = NULL,
  strict = FALSE,
  check_sif = TRUE
)
```

Arguments

path Lockfile path. Defaults to config runtime.lockfile.
 cfg Optional resolved config list. Defaults to current effective config.
 profiles Optional subset of profile names to validate.
 strict Logical; if TRUE, abort on any validation failures.
 check_sif Logical; verify SIF file existence/checksum when present.

Value

Data frame with validation checks.

ni_outputs	<i>Get output file paths from a result</i>
------------	--

Description

Get output file paths from a result

Usage

```
ni_outputs(result)
```

Arguments

result	An ni_result object.
--------	----------------------

Value

A named list of output file paths.

ni_pin	<i>Create/update a runtime lockfile</i>
--------	---

Description

Resolves configured runtime profiles and writes a lockfile that captures the pinned references used for reproducible execution.

Usage

```
ni_pin(
  path = NULL,
  cfg = NULL,
  profiles = NULL,
  pull = TRUE,
  include_specs = TRUE
)
```

Arguments

path	Lockfile path. Defaults to runtime.lockfile in config (or "niflowr.lock.yml").
cfg	Optional resolved config list. Defaults to current effective config.
profiles	Optional subset of profile names to lock. Defaults to all configured profiles.
pull	Logical; allow pulling/resolving container artifacts when needed.
include_specs	Logical; include spec id/version/profile index.

Value

Parsed lock object (invisibly).

ni_provenance	<i>Get provenance metadata from a result</i>
---------------	--

Description

Get provenance metadata from a result

Usage

```
ni_provenance(result)
```

Arguments

result	An ni_result object.
--------	----------------------

Value

A list of provenance information.

ni_provenance_read	<i>Read a provenance JSON sidecar</i>
--------------------	---------------------------------------

Description

Read a provenance JSON sidecar

Usage

```
ni_provenance_read(path)
```

Arguments

path	Path to a provenance JSON file.
------	---------------------------------

Value

A list of provenance metadata.

ni_provenance_write *Write a provenance JSON sidecar*

Description

Write a provenance JSON sidecar

Usage

```
ni_provenance_write(result, path = NULL)
```

Arguments

result	An ni_result object.
path	Output path for the JSON sidecar. If NULL, derived from the primary output path.

Value

The path written (invisibly).

ni_read_output *Read an output from an ni_result as a neuroim2 object*

Description

Loads an output file using neuroim2::read_vol() or neuroim2::read_vec() based on file dimensionality.

Usage

```
ni_read_output(result, output_name = NULL, ...)
```

Arguments

result	An ni_result object.
output_name	Name of the output to read. If NULL, reads the first output.
...	Additional arguments passed to the neuroim2 reader.

Value

A neuroim2 object (NeuroVol or NeuroVec).

ni_read_transform	<i>Read a transform output using neurotransform</i>
-------------------	---

Description

Loads a spatial transform file from an `ni_result` using `neurotransform::read_transform()`.

Usage

```
ni_read_transform(result, output_name = NULL, type = NULL, ...)
```

Arguments

<code>result</code>	An <code>ni_result</code> object.
<code>output_name</code>	Name of the transform output. If <code>NULL</code> , guesses from output names (looks for "warp", "affine", "matrix", "transform").
<code>type</code>	Transform type (e.g. "ants", "fsl", "freesurfer"). If <code>NULL</code> , inferred from the spec id.
<code>...</code>	Additional arguments passed to <code>neurotransform::read_transform()</code> .

Value

A neurotransform morphism object.

ni_result	<i>S3 class for execution results</i>
-----------	---------------------------------------

Description

An `ni_result` captures the outputs, runtime info, and provenance from running a neuroimaging tool.

ni_run	<i>Execute an ni_call</i>
--------	---------------------------

Description

Validates inputs, builds an argument vector, resolves runtime engine (native, docker, apptainer), executes via `processx::run()`, checks outputs, and returns a structured result.

Usage

```
ni_run(
  call,
  ...,
  dry_run = FALSE,
  echo = interactive(),
  provenance = TRUE,
  error_on_status = TRUE,
  return = c("result", "files")
)
```

Arguments

call	An <code>ni_call</code> object, or a spec ID (in which case remaining args are passed to <code>ni_call()</code>).
...	If call is a spec ID, passed to <code>ni_call()</code> .
dry_run	Logical; if TRUE, print the resolved command and return without executing.
echo	Logical; if TRUE, print stdout/stderr in real time. Defaults to <code>interactive()</code> .
provenance	Logical; write a provenance JSON sidecar. Default TRUE.
error_on_status	Logical; if TRUE (default), error when the command exits with a non-zero status. If FALSE, issue a warning instead.
return	One of "result" (default) or "files".

Value

An `ni_result` object, or (when `return = "files"`) a character vector of output files with the full result attached as `ni_result` attribute.

ni_run_files	<i>Run an niflowr call and return output file paths</i>
--------------	---

Description

Convenience function for use inside `tar_target()` expressions. Runs the call and returns a character vector of output paths, suitable for `format = "file"` targets.

Usage

```
ni_run_files(spec_id, ...)
```

Arguments

spec_id	Spec ID or ni_spec object.
...	Named arguments passed to ni_call() .

Value

Character vector of output file paths.

ni_spec_list	<i>List available bundled specs</i>
--------------	-------------------------------------

Description

List available bundled specs

Usage

```
ni_spec_list()
```

Value

A character vector of spec IDs.

ni_spec_read	<i>Read a niflowr interface spec</i>
--------------	--------------------------------------

Description

Load and validate a JSON spec by ID (looks in bundled specs) or by file path.

Usage

```
ni_spec_read(id_or_path, cache = TRUE)
```

Arguments

id_or_path	A spec ID (e.g. "fsl.bet") or path to a JSON file.
cache	Logical; cache the parsed spec for reuse. Default TRUE.

Value

An ni_spec object (S3 list).

tar_ni_step	<i>Create a targets-compatible neuroimaging processing step</i>
-------------	---

Description

A target factory that wraps an niflowr call as a targets::tar_target() with format = "file", so outputs are tracked as file dependencies.

Usage

```
tar_ni_step(
  name,
  spec_id,
  ...,
  packages = "niflowr",
  priority = 0,
  pattern = NULL
)
```

Arguments

name	Symbol; target name.
spec_id	Spec ID string (e.g. "fsl.bet").
...	Named arguments passed to ni_call() .
packages	Character vector of packages to load. Default includes "niflowr".
priority	Numeric priority for target scheduling (0 to 1).
pattern	A dynamic branching pattern (e.g. map(subjects)).

Value

A `tar_target` object.

Index

ni_afni_a_boverlap, 9
ni_afni_afn_ito_nifti, 10
ni_afni_align_epi_anat_py, 11
ni_afni_allineate, 12
ni_afni_auto_tcorrelate, 16
ni_afni_auto_tlrc, 17
ni_afni_autobox, 19
ni_afni_automask, 20
ni_afni_axialize, 21
ni_afni_bandpass, 22
ni_afni_blur_in_mask, 24
ni_afni_blur_to_fwhm, 25
ni_afni_brick_stat, 26
ni_afni_bucket, 27
ni_afni_calc, 29
ni_afni_cat, 30
ni_afni_cat_matvec, 31
ni_afni_center_mass, 32
ni_afni_clip_level, 33
ni_afni_convert_dset, 34
ni_afni_copy, 35
ni_afni_deconvolve, 36
ni_afni_degree centrality, 40
ni_afni_despike, 41
ni_afni_detrend, 42
ni_afni_dot, 43
ni_afni_ecm, 44
ni_afni_edge3, 46
ni_afni_eval, 47
ni_afni_fim, 48
ni_afni_fourier, 49
ni_afni_fwh_mx, 50
ni_afni_gcor, 52
ni_afni_hist, 53
ni_afni_lfcd, 54
ni_afni_local_bistat, 55
ni_afni_localstat, 56
ni_afni_mask_tool, 59
ni_afni_maskave, 60
ni_afni_means, 61
ni_afni_merge, 62
ni_afni_net_corr, 63
ni_afni_notes, 66
ni_afni_nwarp_adjust, 67
ni_afni_nwarp_apply, 68
ni_afni_nwarp_cat, 69
ni_afni_one_d_tool_py, 70
ni_afni_outlier_count, 72
ni_afni_quality_index, 73
ni_afni_qwarp, 75
ni_afni_qwarp_plus_minus, 82
ni_afni_re_ho, 90
ni_afni_refit, 91
ni_afni_remlfit, 93
ni_afni_resample, 96
ni_afni_retroicor, 98
ni_afni_roi_stats, 99
ni_afni_seg, 101
ni_afni_skull_strip, 102
ni_afni_svm_test, 103
ni_afni_svm_train, 104
ni_afni_synthesize, 106
ni_afni_t_cat, 107
ni_afni_t_cat_sub_brick, 108
ni_afni_t_corr1_d, 111
ni_afni_t_corr_map, 109
ni_afni_t_correlate, 112
ni_afni_t_norm, 113
ni_afni_t_project, 114
ni_afni_t_shift, 117
ni_afni_t_smooth, 118
ni_afni_t_stat, 119
ni_afni_to3_d, 120
ni_afni_undump, 122
ni_afni_unifize, 123
ni_afni_volreg, 125
ni_afni_warp, 126
ni_afni_z_cut_up, 128

- ni_afni_zcat, 129
- ni_afni_zeropad, 130
- ni_ants_affine_initializer, 131
- ni_ants_ai, 133
- ni_ants_ants, 134
- ni_ants_ants_introduction, 136
- ni_ants_apply_transforms, 137
- ni_ants_apply_transforms_to_points, 139
- ni_ants_atropos, 140
- ni_ants_average_affine_transform, 142
- ni_ants_average_images, 143
- ni_ants_brain_extraction, 144
- ni_ants_buildtemplateparallel, 145
- ni_ants_compose_multi_transform, 147
- ni_ants_composite_transform_util, 148
- ni_ants_convert_scalar_image_to_rgb, 149
- ni_ants_cortical_thickness, 150
- ni_ants_create_jacobian_determinant_image, 153
- ni_ants_create_tiled_mosaic, 154
- ni_ants_denoise_image, 156
- ni_ants_gen_warp_fields, 157
- ni_ants_image_math, 158
- ni_ants_joint_fusion, 160
- ni_ants_kelly_kapowski, 162
- ni_ants_label_geometry, 164
- ni_ants_laplacian_thickness, 165
- ni_ants_measure_image_similarity, 166
- ni_ants_multiply_images, 167
- ni_ants_n4_bias_field_correction, 168
- ni_ants_registration, 170
- ni_ants_registration_syn_quick, 172
- ni_ants_resample_image_by_spacing, 174
- ni_ants_threshold_image, 175
- ni_ants_warp_image_multi_transform, 176
- ni_ants_warp_time_series_image_multi_transform, 177
- ni_bids_app, 179
- ni_bids_inputs, 180
- ni_call, 180
- ni_call(), 475–477
- ni_cmd, 181
- ni_config, 182
- ni_constraints, 183
- ni_dcm2niix_convert, 183
- ni_deriv_path, 185
- ni_doctor, 186
- ni_dry_run, 187
- ni_fastsurfer_run, 187
- ni_fastsurfer_segment, 189
- ni_fmriprep_confounds, 190
- ni_fmriprep_derivatives, 191
- ni_fmriprep_preproc, 192
- ni_freesurfer_add_x_form_to_header, 193
- ni_freesurfer_aparc2_aseg, 194
- ni_freesurfer_apas2_aseg, 196
- ni_freesurfer_apply_mask, 197
- ni_freesurfer_apply_vol_transform, 198
- ni_freesurfer_bb_register, 200
- ni_freesurfer_binarize, 202
- ni_freesurfer_ca_label, 204
- ni_freesurfer_ca_normalize, 205
- ni_freesurfer_ca_register, 206
- ni_freesurfer_check_talairach_alignment, 208
- ni_freesurfer_concatenate, 209
- ni_freesurfer_concatenate_lta, 210
- ni_freesurfer_contrast, 212
- ni_freesurfer_curvature, 213
- ni_freesurfer_curvature_stats, 214
- ni_freesurfer_dicom_convert, 215
- ni_freesurfer_edit_w_mwith_aseg, 216
- ni_freesurfer_em_register, 217
- ni_freesurfer_euler_number, 218
- ni_freesurfer_extract_main_component, 219
- ni_freesurfer_fit_ms_params, 220
- ni_freesurfer_fix_topology, 221
- ni_freesurfer_fuse_segmentations, 222
- ni_freesurfer_glm_fit, 223
- ni_freesurfer_gtm_seg, 227
- ni_freesurfer_gtmpvc, 228
- ni_freesurfer_image_info, 232
- ni_freesurfer_jacobian, 233
- ni_freesurfer_label2_annot, 234
- ni_freesurfer_label2_label, 235
- ni_freesurfer_label2_vol, 236
- ni_freesurfer_logan, 238
- ni_freesurfer_lta_convert, 242
- ni_freesurfer_make_average_subject, 243
- ni_freesurfer_make_surfaces, 244

- [ni_freesurfer_mni_bias_correction](#), 246
- [ni_freesurfer_mp_rto_mni305](#), 247
- [ni_freesurfer_mr_is_ca_label](#), 248
- [ni_freesurfer_mr_is_calc](#), 249
- [ni_freesurfer_mr_is_combine](#), 251
- [ni_freesurfer_mr_is_convert](#), 252
- [ni_freesurfer_mr_is_expand](#), 254
- [ni_freesurfer_mr_is_inflate](#), 255
- [ni_freesurfer_mri_convert](#), 256
- [ni_freesurfer_mri_coreg](#), 261
- [ni_freesurfer_mri_fill](#), 264
- [ni_freesurfer_mri_marching_cubes](#), 265
- [ni_freesurfer_mri_preteess](#), 266
- [ni_freesurfer_mri_tessellate](#), 267
- [ni_freesurfer_mris_preproc](#), 268
- [ni_freesurfer_mris_preproc_recon_all](#), 270
- [ni_freesurfer_mrtn1](#), 272
- [ni_freesurfer_mrtn2](#), 275
- [ni_freesurfer_ms_lda](#), 279
- [ni_freesurfer_normalize](#), 280
- [ni_freesurfer_one_sample_t_test](#), 281
- [ni_freesurfer_paint](#), 285
- [ni_freesurfer_parcellation_stats](#), 286
- [ni_freesurfer_parse_dicom_dir](#), 288
- [ni_freesurfer_recon_all](#), 289
- [ni_freesurfer_register](#), 291
- [ni_freesurfer_register_av_ito_talairach](#), 292
- [ni_freesurfer_relabel_hypointensities](#), 293
- [ni_freesurfer_remove_intersection](#), 294
- [ni_freesurfer_remove_neck](#), 295
- [ni_freesurfer_resample](#), 296
- [ni_freesurfer_robust_register](#), 297
- [ni_freesurfer_robust_template](#), 299
- [ni_freesurfer_sample_to_surface](#), 301
- [ni_freesurfer_seg_stats](#), 303
- [ni_freesurfer_seg_stats_recon_all](#), 306
- [ni_freesurfer_segment_cc](#), 309
- [ni_freesurfer_segment_wm](#), 310
- [ni_freesurfer_smooth](#), 311
- [ni_freesurfer_smooth_tessellation](#), 312
- [ni_freesurfer_sphere](#), 314
- [ni_freesurfer_spherical_average](#), 315
- [ni_freesurfer_surface2_vol_transform](#), 321
- [ni_freesurfer_surface_smooth](#), 316
- [ni_freesurfer_surface_snapshots](#), 317
- [ni_freesurfer_surface_transform](#), 319
- [ni_freesurfer_synthesize_flash](#), 322
- [ni_freesurfer_talairach_avi](#), 323
- [ni_freesurfer_talairach_qc](#), 324
- [ni_freesurfer_tkregister2](#), 325
- [ni_freesurfer_unpack_sdicom_dir](#), 327
- [ni_freesurfer_volume_mask](#), 328
- [ni_freesurfer_watershed_skull_strip](#), 329
- [ni_from_openneuro](#), 330
- [ni_fsl_apply_mask](#), 332
- [ni_fsl_apply_topup](#), 333
- [ni_fsl_apply_warp](#), 334
- [ni_fsl_apply_xfm](#), 336
- [ni_fsl_ar1_image](#), 339
- [ni_fsl_av_scale](#), 340
- [ni_fsl_b0_calc](#), 341
- [ni_fsl_bedpostx5](#), 342
- [ni_fsl_bet](#), 344
- [ni_fsl_binary_maths](#), 346
- [ni_fsl_change_data_type](#), 347
- [ni_fsl_cluster](#), 348
- [ni_fsl_complex](#), 350
- [ni_fsl_contrast_mgr](#), 352
- [ni_fsl_convert_warp](#), 353
- [ni_fsl_convert_xfm](#), 355
- [ni_fsl_copy_geom](#), 356
- [ni_fsl_dilate_image](#), 357
- [ni_fsl_distance_map](#), 358
- [ni_fsl_dti_fit](#), 359
- [ni_fsl_dual_regression](#), 361
- [ni_fsl_eddy](#), 362
- [ni_fsl_eddy_correct](#), 366
- [ni_fsl_eddy_quad](#), 367
- [ni_fsl_epi_de_warp](#), 368
- [ni_fsl_epi_reg](#), 369
- [ni_fsl_erode_image](#), 371
- [ni_fsl_extract_roi](#), 372
- [ni_fsl_fast](#), 373
- [ni_fsl_feat](#), 375
- [ni_fsl_feat_model](#), 376
- [ni_fsl_filmgls](#), 377
- [ni_fsl_filter_regressor](#), 379
- [ni_fsl_find_the_biggest](#), 380
- [ni_fsl_first](#), 381
- [ni_fsl_flameo](#), 382
- [ni_fsl_flirt](#), 384

ni_fsl_fnirt, 387
ni_fsl_fugue, 390
ni_fsl_glm, 392
ni_fsl_ica_aroma, 394
ni_fsl_image_maths, 396
ni_fsl_image_meants, 397
ni_fsl_image_stats, 398
ni_fsl_inv_warp, 399
ni_fsl_isotropic_smooth, 401
ni_fsl_make_dyadic_vectors, 402
ni_fsl_maths_command, 403
ni_fsl_max_image, 404
ni_fsl_maxn_image, 405
ni_fsl_mcflirt, 406
ni_fsl_mean_image, 408
ni_fsl_median_image, 409
ni_fsl_melodic, 410
ni_fsl_merge, 413
ni_fsl_min_image, 414
ni_fsl_motion_outliers, 416
ni_fsl_multi_image_maths, 417
ni_fsl_overlay, 418
ni_fsl_percentile_image, 420
ni_fsl_plot_motion_params, 421
ni_fsl_plot_time_series, 422
ni_fsl_power_spectrum, 423
ni_fsl_prelude, 424
ni_fsl_prepare_fieldmap, 426
ni_fsl_prob_track_x, 427
ni_fsl_prob_track_x2, 430
ni_fsl_proj_thresh, 433
ni_fsl_randomise, 434
ni_fsl_reorient2_std, 436
ni_fsl_robust_fov, 437
ni_fsl_sig_loss, 438
ni_fsl_slice, 439
ni_fsl_slice_timer, 440
ni_fsl_slicer, 441
ni_fsl_smm, 443
ni_fsl_smooth, 444
ni_fsl_smooth_estimate, 445
ni_fsl_spatial_filter, 446
ni_fsl_split, 447
ni_fsl_std_image, 448
ni_fsl_susan, 449
ni_fsl_swap_dimensions, 450
ni_fsl_temporal_filter, 451
ni_fsl_text2_vest, 452
ni_fsl_threshold, 453
ni_fsl_topup, 454
ni_fsl_tract_skeleton, 457
ni_fsl_unary_maths, 458
ni_fsl_vec_reg, 459
ni_fsl_vest2_text, 460
ni_fsl_warp_points, 461
ni_fsl_warp_points_from_std, 462
ni_fsl_warp_points_to_std, 463
ni_fsl_warp_utils, 464
ni_fsl_x_fibres5, 466
ni_golden_cmdline_generate, 468
ni_inputs, 469
ni_lint_specs, 469
ni_lock_read, 470
ni_lock_validate, 470
ni_outputs, 471
ni_pin, 471
ni_provenance, 472
ni_provenance_read, 472
ni_provenance_write, 473
ni_read_output, 473
ni_read_transform, 474
ni_result, 474
ni_run, 475
ni_run_files, 476
ni_spec_list, 476
ni_spec_read, 477

processx::run(), 475

tar_ni_step, 477