Advanced Normalization Tools Quick Reference

Brian B. Avants

2016-08-08

Introduction

Advanced Normalization Tools[1] is software for biomedical image analysis with a focus on registration, segmentation, geometric quantification and statistics. Statistical methods are available in *ANTsR* [2] which tightly integrates *ANTs* with the *R* statistical computing language. This document briefly highlights *ANTs* features.

[1] `http://stnava.github.io/ANTs/`

[2] `http://stnava.github.io/ANTsR/`

Provenance and Testing

Much core functionality in *ANTs* lives in ITK, a project to which we contribute regularly. This core is tested on many platforms via `ctest` and the results are on the ITK dashboard[3]. *ANTs* is also tested with every commit via codeship [4] and Travis [5]. Many (not all) *ANTs* programs support `programName --version`. Our github commit hashes give the best way to track code versions by identifying core and dependency versions. See the *ANTs* website for current testing results and **installation** instructions.

[3] `https://open.cdash.org/index.php?project=Insight`

[4] `https://codeship.com`

[5] `https://travis-ci.org/stnava/ANTs`

In conjunction with other analysis systems

ANTs and ITK work synergistically within a well-defined I/O and world coordinate system. Using FSL, SPM or other pre-processing in conjunction with ITK and *ANTs* must be done with extreme care. The physical spaces (or interpretation of image headers) may not be the same in different systems. Such inconsistencies may lead to severe misinterpretation of results.

WARNING!

Data types

Images

Images are the core data type in ANTs. Valid extensions are determined by ITK [6] image input/output libraries [7]. ITK images can be of arbitrary dimensionality and pixel type, but in *ANTs* we instantiate `float` pixel type in 2, 3 or 4 dimensions.

[6] www.itk.org

[7] NiftiImageIO, NrrdImageIO, GiplImageIO, HDF5ImageIO, JPEGImageIO, GDCMImageIO, BMPImageIO, LSMImageIO, PNGImageIO, TIFFImageIO, VTKImageIO, StimulateImageIO, BIORadImageIO, MetaImageIO, MRCImageIO, GE4ImageIO, GE5ImageIO, MGHImageIO

CSV files and point sets

CSV (comma separated value) files are useful for storing tabular data. We may also use them to store point sets. It is critical when using point sets with *ANTs* to ensure that the physical space of the points matches that of the images. It is best to start with our standard example of registering images and applying the results to points [8]. Point sets may also be stored in the binary format provided by the Meta I/O library [9]. This format allows much faster I/O for large datasets.

[8] https://github.com/stnava/chicken

[9] http://www.itk.org/Wiki/ITK/MetaIO/Documentation

Registration

The most well-known components of *ANTs* relate to mapping between coordinate systems [10]. Briefly, these maps may relate to:

[10] Registration: targeting of points in one coordinate system onto another

- statistical image similarity: image difference, correlation or mutual information
- translation, rotation, affine, B-spline or various diffeomorphic models [11]
- spatial or spatiotemporal maps
- dense or sparse transformation models, the latter being similar to SIFT or HOGG
- applying composite transformations to images, labels or point sets.

[11] Diffeomorphism: differentiable map with differentiable inverse

These components all exist within an integrated framework which is nearly always capable of incorporating many features or applying to multiple different modalities.[12]

[12] http://journal.frontiersin.org/article/10.3389/fninf.2014.00044/abstract

Evaluation history

ANTs became well known because it performed well in a variety of open competitions related to image registration. Most of these successes occurred in the days before the community had many such competitions. [13] Highlights include:

[13] Klein 2009, Murphy 2011, SATA 201*, BRATS 2013, etc.

- finishing in the top rank in the Klein 2009 evaluation on brain MRI
- finishing in the top rank in the Murphy 2011 lung CT evaluation
- top SATA 2012 and 2013 finishers used ANTs
- performing well in a cardiac motion estimation competition
- well-known robust performance on large datasets

Although *ANTs* has often performed well without using domain knowledge, it is still valuable to use problem-specific solutions when feasible.

Quick start

ANTsR gives some quick registration options. One can achieve similar performance with `antsRegistrationSyNQuick.sh`.

```
fi <- antsImageRead(getANTsRData("r16"))
mi <- antsImageRead(getANTsRData("r64"))
mytxr <- antsRegistration(fixed = fi, moving = mi,
    typeofTransform = c("Rigid"))
mywarpedimager <- antsApplyTransforms(fixed = fi,
    moving = mi, transformlist = mytxr$fwdtransforms)
mytx <- antsRegistration(fixed = fi, moving = mi,
    typeofTransform = c("SyN"))
mywarpedimage <- antsApplyTransforms(fixed = fi,
    moving = mi, transformlist = mytx$fwdtransforms)

invisible(plot(fi, mi %>% iMath("Canny", 1, 5,
    12)))

invisible(plot(fi, mywarpedimager %>% iMath("Canny",
    1, 5, 12)))

invisible(plot(fi, mywarpedimage %>% iMath("Canny",
    1, 5, 12)))
```

Low-dimensional

translation, rigid, affine - optional multi-start exploration of the transformation space (`antsAI`)

High-dimensional

transformations with many parameters - primarily `SyN` and `B-spline SyN` as well as time-varying diffeomorphic models.

Parameter choices and testing

We wrote a paper that details procedures for evaluating analysis software [14]. We wrote this paper in order to help those who want to use or evaluate *ANTs* make more informed choices on how to proceed. Briefly: (1) ask developers questions; (2) leverage biologically motivated testing metrics that are independent of registration.

[14] `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3766821/`

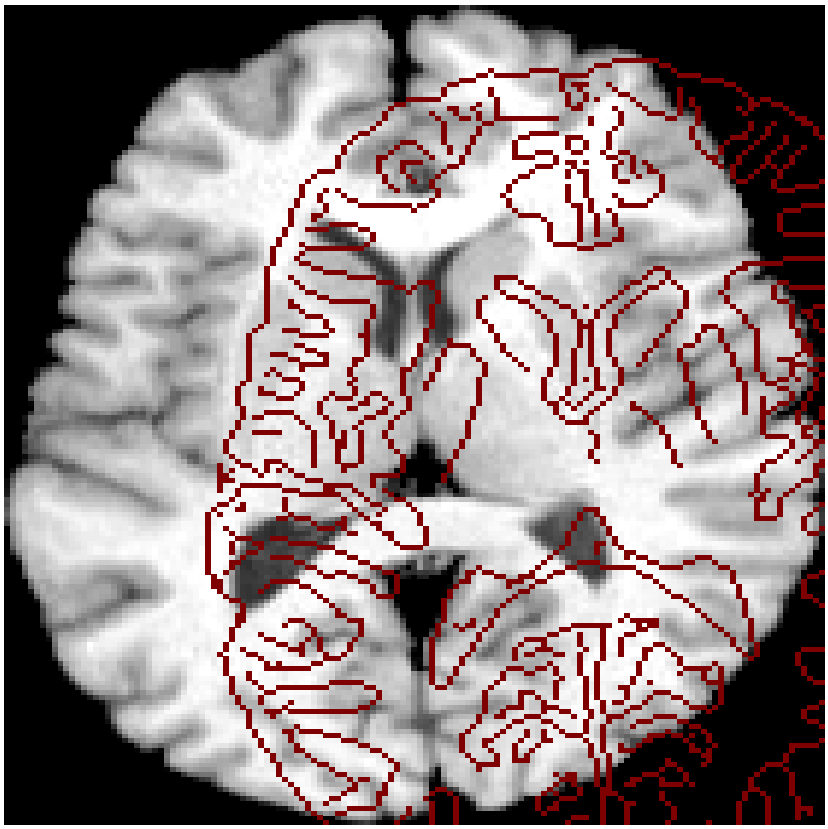

Figure 1: initial relative image positions

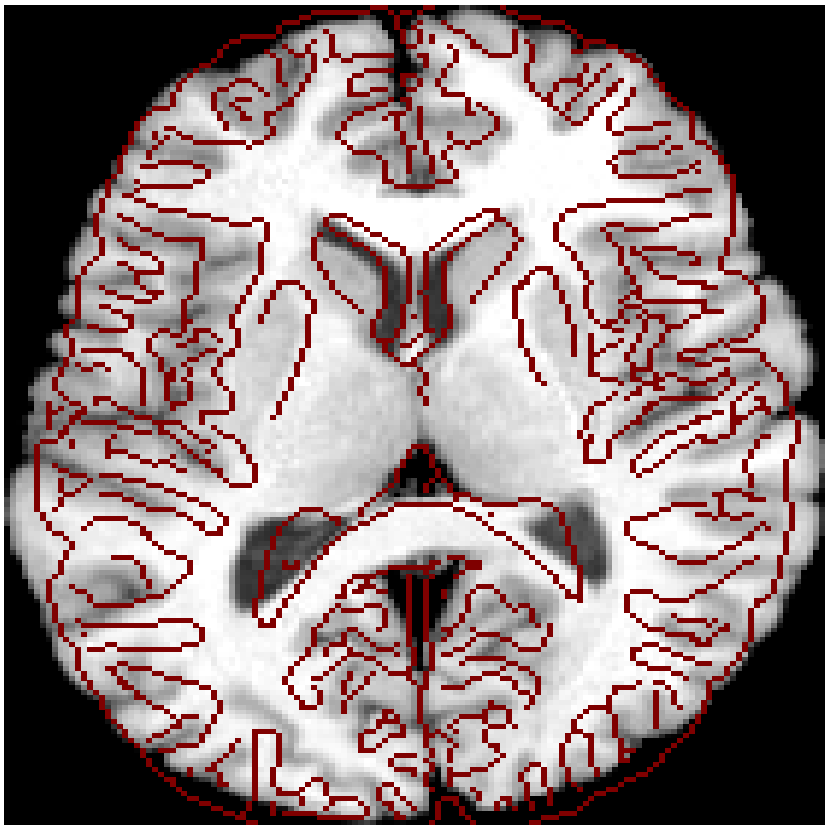

Figure 2: rigidly aligned image positions

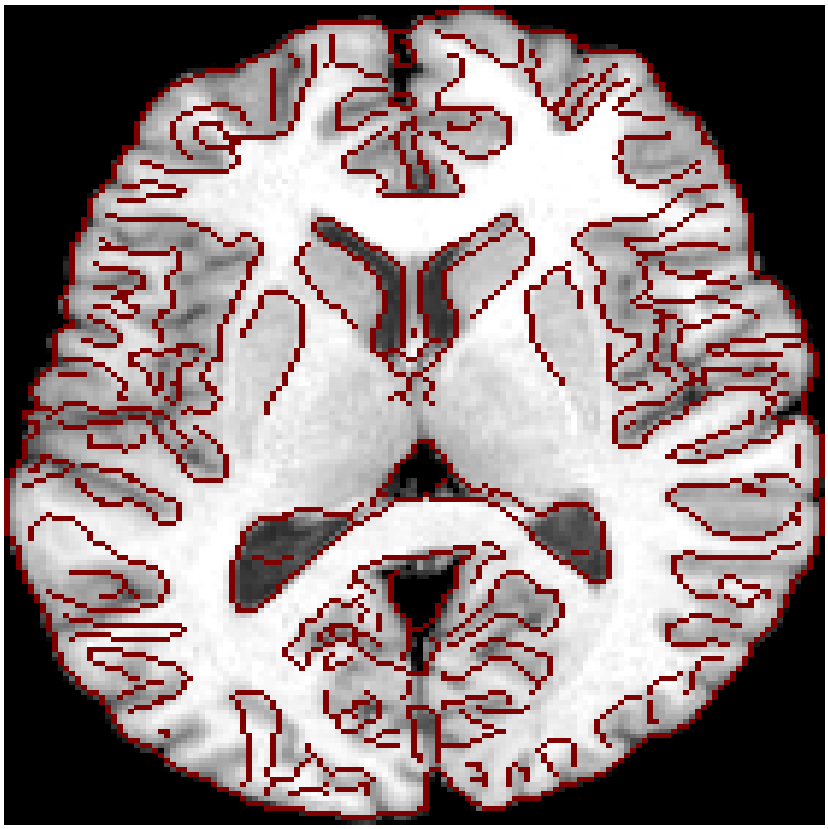

Figure 3: quick deformable registration

Segmentation, geometry and labeling

This category of methods relates to labeling and quantifying images. These methods may require data of reasonable quality to perform well.

WARNING!

N4

Nick's N3 [15] improves the original N3 inhomogeneity correction method.

[15] http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3071855/

```
ficorrupt = as.antsImage(as.array(fi) * 1:length(fi[fi >=
    0]))
fifixed = n4BiasFieldCorrection(ficorrupt)

invisible(plot(ficorrupt))
```

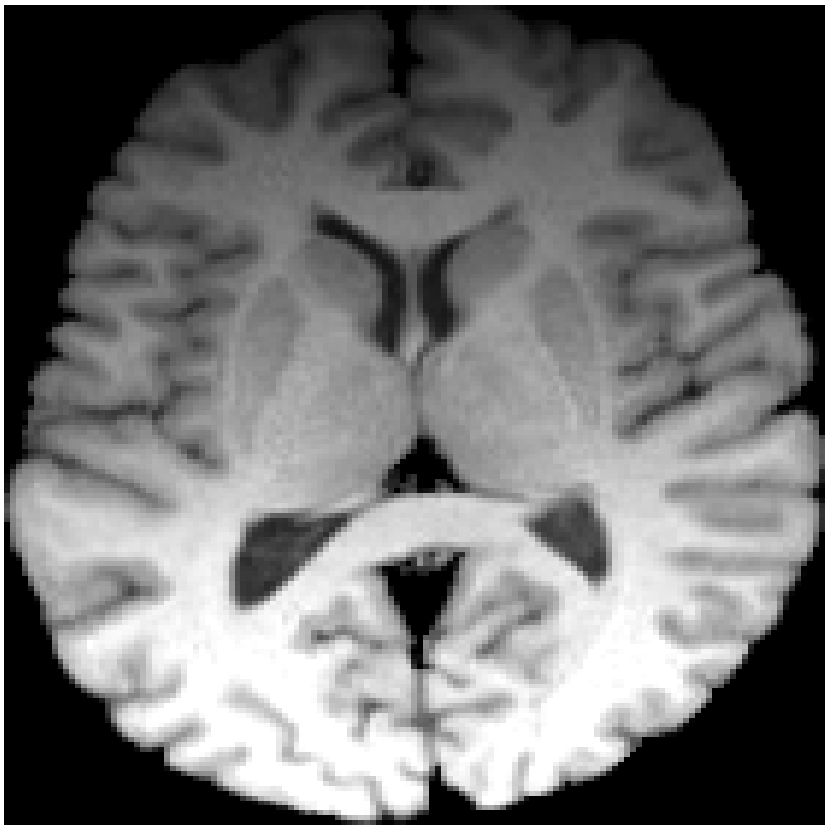

Figure 4: corrupted image

```
invisible(plot(fifixed))
```

Figure 5: n4 repair

Atropos

Expecation maximization segmentation with a variety of likelihood models and initialization strategies. Incorporates multiple modalities and allows control of prior strength. The finite mixture modeling (FMM) segmentation approach is the most popular. Prior constraints include the specification of a prior label image, prior probability images (one for each class), and/or an MRF prior to enforce spatial smoothing of the labels.

```
img <- antsImageRead(getANTsRData("r16"))
mask <- getMask(img)
segs1 <- kmeansSegmentation(img, 3, mask)

# Use probabilities from k-means seg as priors
feats <- list(img, iMath(img, "Laplacian"), iMath(img,
    "Grad"))
segs2 <- atropos(d = 2, a = feats, m = "[0.2,1x1]",
    c = "[2,0]", i = segs1$probabilityimages,
    x = mask)

invisible(plot(fi, segs1$segmentation))


invisible(plot(fi, segs2$segmentation))
```

Weingarten map-based surface curvature and area

The shape operator provides a beautiful way to compute the mean or Gaussian curvature (or related values) in any three-dimensional image [16]. The *ANTs* program that implements this is called `SurfaceCurvature`.

[16] http://www.ncbi.nlm.nih.gov/pubmed/15344450

```
fi <- antsImageRead(getANTsRData("mni"))
fiseg = kmeansSegmentation(fi, 3)
fik <- weingartenImageCurvature(fi)

invisible(plot(fik, axis = 3))


fisulc = antsImageClone(fik) * 0
selector = (fiseg$segmentation == 2 & fik < 0)
fisulc[selector] = fik[selector]
invisible(plot(fi, fisulc, axis = 3))
```

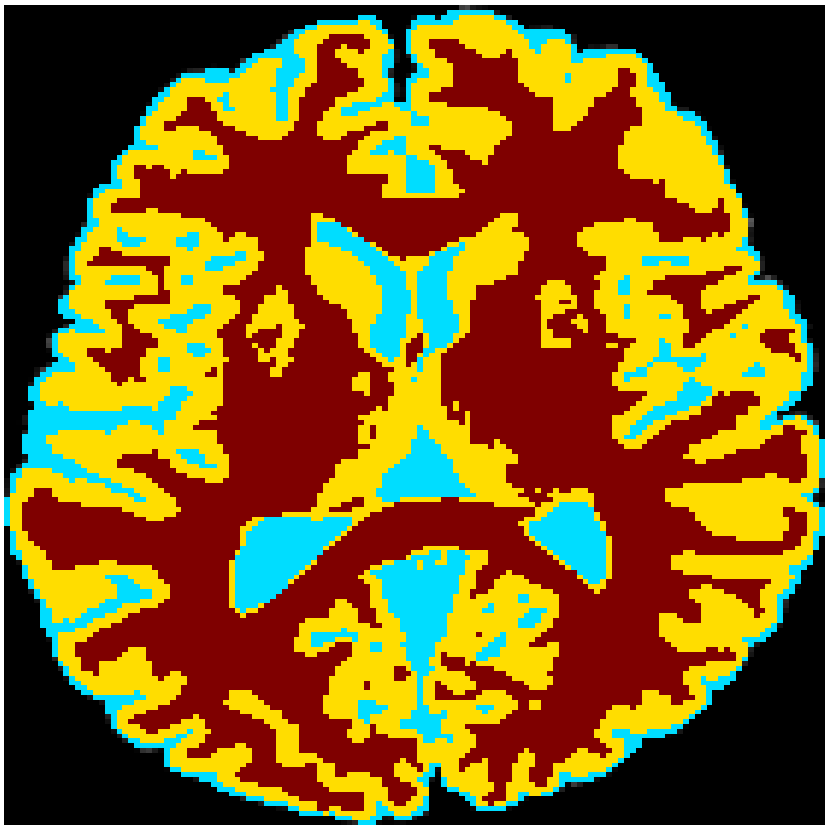

Figure 6: kmeans result

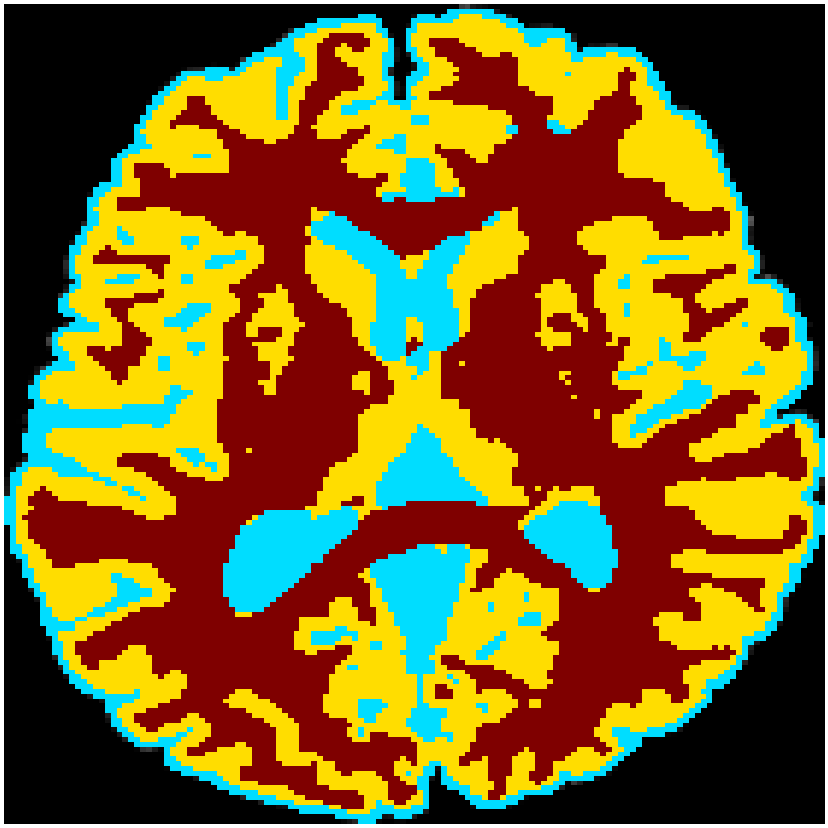

Figure 7: prior probability result with 2 feature channels

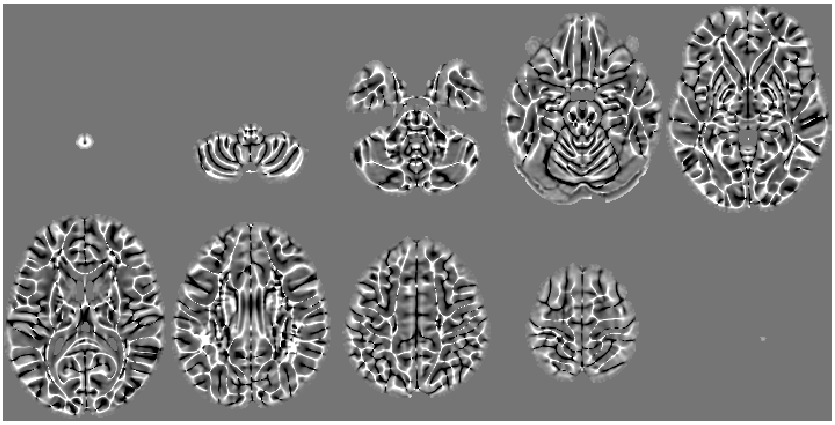

Figure 8: shape operator in 3D

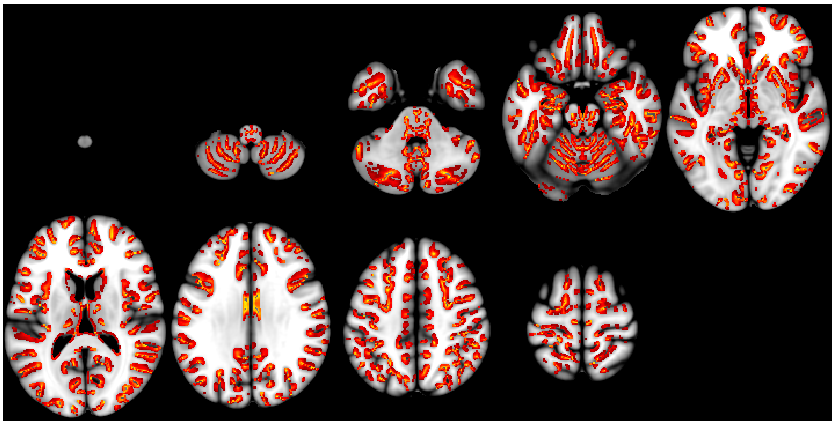

Figure 9: shape operator selects sulci

Cortical thickness pipeline

You can use `antsCorticalThickness.sh` to reproduce our lifespan thickness analysis from the Freesurfer evaluation paper [17].

[17] http://www.ncbi.nlm.nih.gov/pubmed/24879923

```
img <- antsImageRead(getANTsRData("r16"), 2)
mask <- getMask(img)
segs <- kmeansSegmentation(img, k = 3, kmask = mask)
thk <- kellyKapowski(s = segs$segmentation, g = segs$probabilityimages[[2]],
    w = segs$probabilityimages[[3]], its = 45,
    r = 0.5, m = 1)

invisible(plot(img, thk, window.overlay = c(0.1,
    max(thk))))
```

Multiple modality pipeline

You can use our *neurobattery* to reproduce our adolescent multiple modality analysis from the paper [18]. The neurobattery is online and administered by Jeffrey T. Duda [19].

[18] http://www.nature.com/articles/sdata20153

[19] https://github.com/jeffduda/NeuroBattery

Brain mapping in the presence of lesions or other obstructions

Our fully automated registration and segmentation approach for the lesioned or occluded brain uses machine learning to identify and down-weight missing data. One may also use pre-identified inclusive masks to focus registration on healthy tissue [20].

[20] `antsRegistration -d 3 ... etcetera ... -x healthyTissueMask.nii.gz`

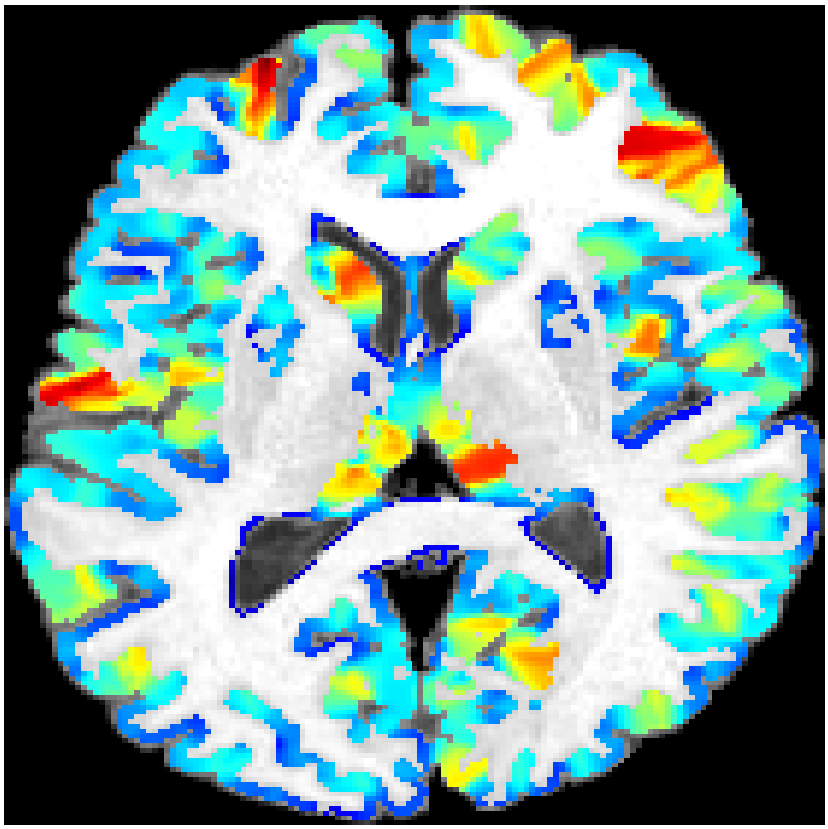

Figure 10: thickness calculation

Statistical pipelines

ANTs-based statistical pipelines are best accessed through *ANTsR*. We use these pipelines for:

- estimating cerebral blood flow with robust and data-driven methods
- performing structural and functional connectivity measurements
- employing biologically-regularized dimensionality reduction for hypothesis testing
- implementing exploratory multivariate analysis tools such as *sparse canonical correlation analysis for neuroimaging* [21]
- performing prediction studies [22]
- associating modalities [23]

and more basic studies such as mass univariate regression.

[21] http://www.sciencedirect.com/science/article/pii/S1053811913009841
[22] http://www.sciencedirect.com/science/article/pii/S1046202314003338
[23] http://www.sciencedirect.com/science/article/pii/S105381191400891X

Overview of ANTs *programs*

ANTs executables come in either binary (compiled from `C++`) or script form (usually `bash` or `R`). Here, we summarize the relevant programs.

Table 1: ANTs transformation programs

	program	description
4	ANTSIntegrateVectorField	ode integration of vector field
5	ANTSIntegrateVelocityField	ode integration of velocity field
8	ANTSUseLandmarkImagesToGetAffineTransform	fit affine tx to landmarks
12	AverageAffineTransform	average affine transformations
13	AverageAffineTransformNoRigid	average affine transformations without rigid part
19	CompositeTransformUtil	dis/assembles composite transform files
25	ConvertTransformFile	convert mat to txt, for example
30	CreateJacobianDeterminantImage	compute deformation gradient from displacement
38	FitBSplineToPoints	fit bspline to point set
64	ReorientTensorImage	reorient tensors by transformation
85	antsAI	multi-start low-dimensional registration
88	antsAffineInitializer	see antsAI
89	antsAlignOrigin	basic origin alignment
90	antsApplyTransforms	apply (multiple) transformations to image types
91	antsApplyTransformsToPoints	apply (multiple) transformations to points
96	antsIntermodalityIntrasubject.sh	example of how to map within subject

	program	description
103	antsMotionCorr	time series motion correction
104	antsMotionCorrDiffusionDirection	DWI specific motion correction
106	antsMotionCorrStats	summarize antsMotionCorr output
107	antsMultivariateTemplateConstruction.sh	multiple modality templates
108	antsMultivariateTemplateConstruction2.sh	multiple modality templates
111	antsRegistration	standard registration algorithms
112	antsRegistrationSpaceTime.sh	spatiotemporal registration methods
113	antsRegistrationSyN.sh	default decent quality registration
114	antsRegistrationSyNQuick.sh	default fast registration
115	antsSliceRegularizedRegistration	see spinal cord toolbox
132	geodesicinterpolation.sh	shape-based interpolate two images
133	guidedregistration.sh	example of landmark based registration
137	landmarkmatch.sh	example of landmark based registration

Table 2: ANTs segmentation programs

	program	description
11	Atropos	EM segmentation framework and tools
49	LabelClustersUniquely	label each isolated region
77	ThresholdImage	simple image thresholding
92	antsAtroposN4.sh	joint segmentation and bias correction
94	antsBrainExtraction.sh	brain extraction via registration and segmentation
98	antsJointFusion	better JLF implementation
99	antsJointLabelFusion.sh	better JLF script
150	skel.sh	topological skeleton of segmentation
163	mrvnrf	machine learning segmentation - see ANTsR

Table 3: ANTs processing programs

	program	description
14	AverageImages	average list of images
15	AverageTensorImages	average list of tensor images
21	ConvertImagePixelType	change pixel type
34	DenoiseImage	non-local denoising
39	GetConnectedComponentsFeatureImages	?
45	ImageMath	basic processing operations on images
47	KellyKapowski	image-based thickness estimator
50	LabelGeometryMeasures	measure geometry of labeled regions
58	N3BiasFieldCorrection	ancient bias corrector
59	N4BiasFieldCorrection	better bias corrector

	program	description
63	RebaseTensorImage	map tensor into new basis
71	SmoothImage	smooth in given units
73	SurfaceBasedSmoothing	smoothing restricted to segmentation
74	SurfaceCurvature	shape operator curvature
75	TextureCooccurrenceFeatures	texture based statistics
76	TextureRunLengthFeatures	texture based statistics
86	antsASLProcessing.R	BMKandel ASL processing
87	antsASLProcessing.sh	BMKandel ASL processing
95	antsCorticalThickness.sh	brain thickness pipeline
101	antsLongitudinalCorticalThickness.sh	longitudinal brain thickness pipeline
110	antsNeuroimagingBattery	align MR modalities to common space

Table 4: ANTs statistics programs

	program	description
27	CreateDTICohort	simulate DTI population
44	ImageIntensityStatistics	simple summary stats for image +/- ROIs
46	ImageSetStatistics	compute mean, median, etc of images
51	LabelOverlapMeasures	compute overlaps
54	MeasureImageSimilarity	similarity between image pairs
55	MeasureMinMaxMean	basic stats on an image
79	TimeSCCAN	cca for temporal images
146	sccan	utility for sparse decomposition - see ANTsR

Table 5: ANTs visualization programs

	program	description
23	ConvertScalarImageToRGB	use lookup table to make RBG from gray
31	CreateTiledMosaic	tile images for viewing
32	CreateWarpedGridImage	make a warped grid from displacement
72	StackSlices	stack up a population of image slices
78	TileImages	collect images in tile form for viewing
116	antsSurf	surface rendering and other operations

Table 6: ANTs utility programs

	program	description
10	ANTSpexec.sh	helper for parallel execution

	program	description
26	CopyImageHeaderInformation	copy header from one image to another
28	CreateDisplacementField	make vector field from component images
29	CreateImage	make an image
35	ExtractRegionFromImage	get subset of image using indices
36	ExtractRegionFromImageByMask	get subset of image using mask
37	ExtractSliceFromImage	get slice from image
41	ITK_bld_internal_H5detect	?
42	ITK_bld_internal_H5make_libsettings	?
43	ImageCompare	see if images are nearly the same
56	MemoryTest	memory profiler
57	MultiplyImages	multiply image1 by x
60	PasteImageIntoImage	put one image in another
61	PermuteFlipImageOrientationAxes	flip or permute image
62	PrintHeader	image header information
65	ResampleImage	change image resolution
66	ResampleImageBySpacing	change image resolution
67	ResetDirection	change image direction to identity
68	SetDirectionByMatrix	change image direction
69	SetOrigin	set origin in image
70	SetSpacing	set spacing in image
117	antsTransformInfo	investigate a transformation
118	antsUtilitiesTesting	see how metric changes with tx
130	compareTwoTransforms	as described
157	waitForPBSQJobs.pl	utility
158	waitForSGEQJobs.pl	utility
159	waitForXGridJobs.pl	utility