Advanced Normalization Tools (ANTS)

Release 1.00

Brian B. Avants, Nick Tustison and Gang Song

Abstract

We provide examples and highlights of Advanced Normalization Tools (ANTS) that address practical problems in real data.

Contents

1 Introduction

The ANTS framework provides open-source, encapsulated, software-engineered functionality for deformable normalization and segmentation. Independent evaluation of ANTS normalization tools, applied to "control" data, placed the toolkit as a top performer amongst 14 methods [12]. Developer evaluation showed stronger differences with other methodology in neurodegenerative neuroimaging data, where large deformation is required [5]. ANTS serves as both a base for further algorithm development and also as an application-oriented toolkit. ANTS enables diffeomorphic normalization with a variety of transformation models, optimal template construction, multiple types of diffeomorphisms, multivariate similarity metrics, diffusion tensor processing and warping, image segmentation with and without priors and measurement of cortical thickness from probabilistic segmentations. The normalization tools, alone, provide a near limitless range of functionality and allow the user to develop customized objective functions. Objective functions in ANTS are of the form:

$$\text{Deformation Cost} + \text{Data Terms},$$

and the command line reflects this balance of two terms. As mentioned above, the data term may combine multiple different measures of similarity that are optimized in parallel, for instance, image similarity and landmark terms. This document seeks to provide a practical overview of basic functionality and some of the common use cases that users seek. Additional information is available online – see ANTS Homepage : www.picsl.upenn.edu/ANTS. For compilation details, see: Compile and download: http://picsl.upenn.edu/ANTS/download.php. The most important ANTS programs are described below. [1]

1.1 The ANTS Executable

The ANTS program itself is the central program encapsulating normalization/registration functionality. Its main output is an affine transform file and a deformation field, potentially with inverse. Options to ANTS allows the user to navigate the similarity and transformation options that are available. ANTS takes multiple similarity and optimization criteria as options. The program is wrapped in `ants.sh` for normalization with "out of the box" parameters and in `buildtemplateparallel.sh` for computationally distributed optimal template construction.

---

[1]This document is a work in progress. Please check for updates with each release.

1.2   The WarpImageMultiTransform Executable

The WarpImageMultiTransform program applies ANTS mappings to images. One may apply an arbitrarily long series of transformations to an image through this program. Thus, it enables one to compose a series of affine and deformable mappings – and/or their inverses – such that one may avoid multiple interpolations of a single image. This program is used internally in ANTS optimization to avoid ever using more than one interpolation. The use-NN option applies nearest neighbor interpolation, otherwise linear interpolation is used.

1.3   The ImageMath Executable

This is a multi-purpose program that has the following syntax: "ImageMath ImageDimension outputfilename Operation InputFileName parameters ". Most basic scalar image operations – and some tensor operations – may be performed with this program. Some operations output text files, some output images and some output only to the terminal. ImageMath allows one to multiply images together (m), to negate images (Neg), to take an image to a power (pow), to test the invertibility of transformations (InvId), to compute the fractional anisotropy of an image (TensorFA) and to perform Markov Random Field segmentation with or without priors (Segment). Many other operations are available. Like all other ANTS programs, one may call ImageMath from the command line to see all of its options. ImageMath is used heavily in ANTS scripts.

1.4   ANTS Scripts

The ANTS/Scripts directory contains (hopefully) user-friendly wrappings of ANTS tools that enable higher-level error checking and combinations of basic ANTS functions. These scripts are called as `sh antsscriptname.sh` and provide usage when called from the command line. For instance, try `sh ants.sh`.

2   ImageRegistration with ANTS

There are two general applications of image registration. The first application is transforming labeled data from a template image space into an individual space. This strategy is important when appearance alone is not enough to locate a structure as, for example, in the case of hippocampus segmentation. The template provides a prediction of the hippocampus-amygdala boundary. The second application operates in the "inverse" direction of the first: instead of mapping template to individual, we map individual(s) to the template. Voxel-based population studies of either functional or structural variables depend on mapping to a template space. The common coordinate system enables a statistical evaluation of the likelihood of consistent activation across a group or, in other contexts, the differences in anatomy between two groups.

Quick Start: call ants.sh (from ANTS/Scripts/) to get usage and apply a normalization to some of your existing data. It is instructional to read the script, modify some of the parameters and re-run to witness the effect of your changes – image registration is an art as well as science. Many other "ready to go" scripts are available in ANTS/Scripts. The user must set his/her ANTSPATH environment variable – which points to the location of ANTS binaries – within these scripts or in their environment. Note: All ANTS programs provide usage when called from the command line. Most require the image dimension to be specified as the first parameter. E.g. ImageMath ImageDimension where ImageDimenion is 2 or 3.

Transformation Model   Appearance/Similarity Metrics

$$\int_0^1 < Lv(x,t), v(x,t) > dt + w_1 SSD(I,J) + w_2 MI(I,J) + w_3 \sum_i LM_i(I,J)$$

Diffeomorphic Regularization $+$ Intensity Difference $+$ Mutual information $+$ Landmark Guidance

ANTS -t Syn[0.5]   –m MSQ(I,J,w1,0) –m MI(I,J,w2,#bins) –m PSE(I,J,w3)

Note: the choice of *L* above relates to the -r (regularization) parameter in ANTS, which would be part of the blue above.
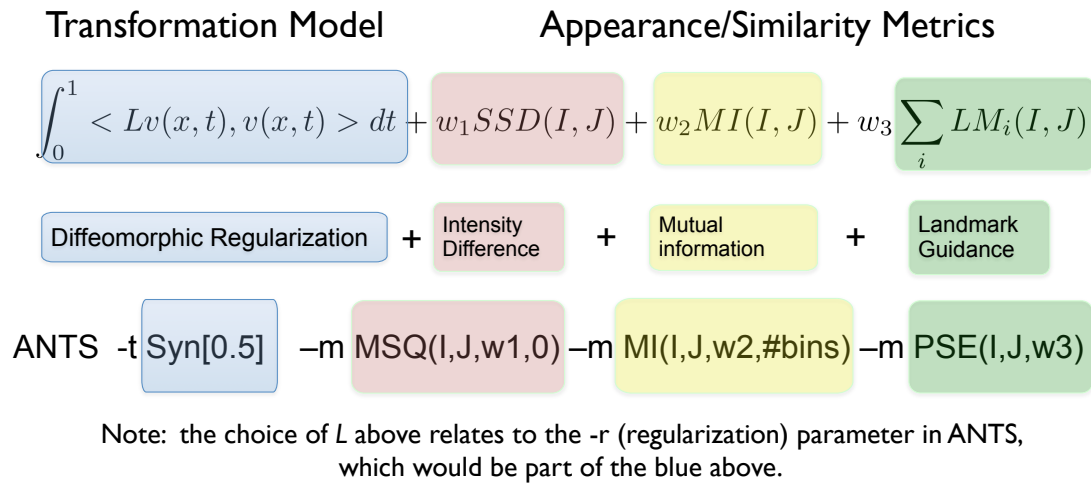
Figure 1: The relationship between the variational form that defines the optimization goals and the ANTS command line.

The ANTS toolkit enables both types of mapping.
The main challenge in image and brain mapping is defining the way in which images/anatomy are compared. There are two components to the comparison. The shape transformation space defines the range of shape variation that will be allowed in the optimization. The appearance similarity space defines the statistical assumptions that determine when one image is considered to appear similar to another.

These two components interact in a weighted optimization within a multiple resolution gradient descent framework. Each component may use either "strict" or "flexible" assumptions about the shape or appearance similarity. The selection of these models should be done in a principled way that fits the problem at hand. No single choice is appropriate for all scenarios (see the "no free lunch" theorem).

Thus, ANTS enables many operating points from both the transformation and appearance domains such that users may make choices appropriate for their problems. The ANTS command-line syntax, shown in figure 2, reflects these operating points and the various components that interact in the optimization. ANTS may be used to navigate the transformation and similarity metric space.

2.1 Use Your Header!

ANTS uses the direction/orientation, spacing and origin in its definitions of the mapping between images. That is, ANTS uses the nifti standard defintions of an image space. So, the image orientation (direction), origin and spacing are important! You may "check" for consistency between two image header definitions by using: ImageMath 3 Image2repaired.nii.gz CompareHeadersAndImages Image1.nii.gz Image2.nii.gz This will compare Image2 to Image1 and "fix" the header of Image2, writing out to Image2repaired.

2.2 ANTS Transformation Models

The ANTS toolkit provides a hierarchy of transformations with adjustable levels of complexity, regularization, degrees of freedom and behavior as optimizers. The simplest transformation model is the rigid and/or affine transform. The most complex – and most flexible – is a symmetric diffeomorphic transformation based on optimizing and integrating a time-varying velocity field. Computation time also increases with

Category	Transformation, ϕ	Similarity Measures	Brief Description
Linear	Rigid†	MI, MSQ	Rigid registration.
	Affine†	MI, MSQ	Affine registration.
Elastic	Deformable	CC, PR, MI, MSQ, PSE	Demons-like algorithm.
	DMFFD	CC, PR, MI, MSQ, PSE	FFD variant.
Diffeo.	Exponential†	CC,PR, MI, MSQ, PSE	min $v(\mathbf{x})$
	Greedy SyN†	CC, PR, MI, MSQ, PSE	locally in time min $v(\mathbf{x},t)$
	Geodesic SyN†	CC, PR, MI, MSQ, PSE	min $v(\mathbf{x},t)$ over all time

Table 1: Transformations and a subset of the similarity metrics available in ANTS. Similarity metric acronyms: CC = fast cross correlation, PR = pure cross correlation (the preferred metric), MSQ = mean squared difference, MI = mutual information, PSE = point-set expectation. ANTS also provides the inverse of those transformations denoted by the '$\dagger$' symbol. The brief descriptions of the diffeomorphic algorithms contrast the way in which the velocity field is optimized and used to parameterize ϕ, the mapping. All ANTS Diff algorithms generate $\phi(\mathbf{x},t)$ over $t \in [0,1]$ through gradient descent.

transformation model complexity, but most normalization needs may be met with under an hour of computation. We provide an overview of some of the ANTS models below and try to communicate intuition on what one gains/loses with each choice. An overview of available models and similarity terms is in Table 1.
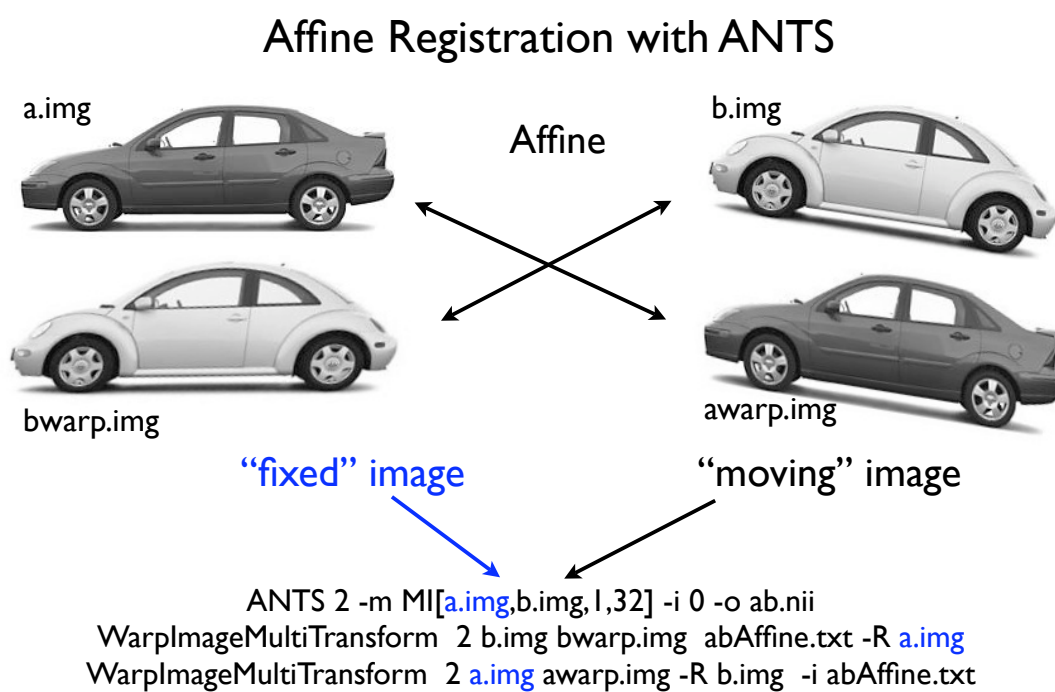
Linear Registration

The most basic mapping between images is an affine mapping. The ANTS affine mapping syntax is shown in the affine mapping figure 2.2. The parameter breakdown of the ANTS call in the affine figure is:

- ANTS – the normalization program

- 2 – the expected image dimension

- -m MI[...,...,1,32] – the similarity metric "mutual information" with a 32-bin square joint histogram and weight of 1. Metric details will be discussed in a later section.

- -i 0 – no deformable iterations (affine only).

- -o ab – the output prefix. A specific extension may also be specified, e.g. -o ab.mhd .

The parameter breakdown of the WarpImageMultiTransform call in the affine figure is:

- WarpImageMultiTransform – to apply the mapping output from ANTS to an image.

- 2 – the expected image dimension.

- the image to be deformed is input next.

- the deformed image output file name is next.

- the transform itself is next – if is affine and preceded by " -i ", we apply the inverse affine map.

Affine Registration with ANTS

a.img   Affine   b.img

bwarp.img   awarp.img

"fixed" image   "moving" image

ANTS 2 -m MI[a.img,b.img,1,32] -i 0 -o ab.nii
WarpImageMultiTransform 2 b.img bwarp.img abAffine.txt -R a.img
WarpImageMultiTransform 2 a.img awarp.img -R b.img -i abAffine.txt

Outputs: abAffine.txt
Note: the inverse affine map transforms "fixed" to "moving" space

Figure 2: The anatomy of an ANTS optimization and application of the resulting warping.

Figure 3: This example shows the degree to which the beetle (b.img) may be deformed to the ford (a.img) under different transformation models. Left to right increases the degrees of freedom in the mapping and thus the registration accuracy.

- the " -R " option dictates the domain you want to warp into – usually the "fixed" image, unless using the inverse mapping, in which case one switches the role of fixed and moving images.

Note that -i 0 (in the ANTS call) means that no deformable mapping is performed. The call "ANTS 2" means to treat the images as 2-dimensional. "ANTS 3" would be called for 3D images. The " -m " term defines the similarity metric. Note that no more than one image interpolation is used during optimization. That is, ANTS always references the original image before applying a transformation or series of transformations.

Affine mapping may become nontrivial when the region of interest occupies only a small portion of the image data. In such cases, ANTS enables the user to define a mask to focus the optimization effort on the region of interest. Example code for (affine) registration with a mask:

```
ANTS 3 -m MI[AtlasHead.nii,crAnatomical.nii,1,32]
-o TEST -i 10x10x0  -r Gauss[1.5,0] -t Exp[0.5] -x mask.nii.gz
```

The difference between this command and a regular ANTS call is the `-x` option, which specifies the mask, defined in the template space (here, AtlasHead.nii). The mask option also affects the deformable optimization. Other affine registration options include `--number-of-affine-iterations 5000x5000x5000` which specifies that the affine registration uses a 3 level image pyramid with each level 5000 iterations at most. `--MI-option 48x6000` means to use Mutual Information as similarity metric with 48 bins and 6000 samples. `--affine-gradient-descent-option` defines the options for the gradient descent procedure; the 4 numbers are maximum step length, relaxation factor, minimum step length and translation scale.

Deformable Registration

Affine mapping is adequate when the difference between images involves only rotation, scaling or shearing. Other data requires more deformable mappings to capture shape differences and find a good alignment of image features. Figure 2.2 shows how deformable mapping may improve the correspondence of the deformed beetle to the ford image. Most ANTS-based applications use symmetric diffeomorphic normalization. However, ANTS also enables a simpler parameterization of a deformable mapping via a regularized vector space. We term these types of transformations as "elastic". The original Demons algorithm provides a classic example of using a regularized vector space for nonlinear registration. Caveats are that a regularized vector space may not preserve the underlying topology and may also prove too inflexible to capture the shape changes one is after. Both of these shortcomings motivate the use of diffeomorphisms.

Elastic/Vector Space Transformations. If we assume no affine transformation, then an elastic transformation involves computing a mapping from image $I(\mathbf{x})$ to image $J(\mathbf{x})$ through a deformation field $\mathbf{u}(\mathbf{x})$. The deformation is defined in the physical space of the image and dictates the positional difference between

Image I   Image J

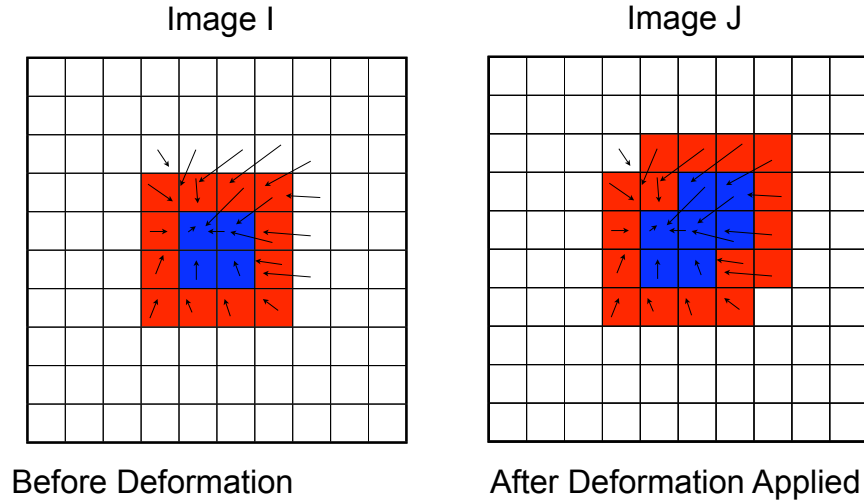Before Deformation   After Deformation Applied

Figure 4: A deformation in a digital domain.

corresponding features in the two images. Thus, if a feature defined at $I(\mathbf{x})$ matches a feature in J at position $\mathbf{y}$ then the deformation field at $\mathbf{x}$ should give $\mathbf{u}(\mathbf{x}) = \mathbf{y} - \mathbf{x}$. Such a deformation field may be applied to deform image J into image I by composing the mapping $J_{deformed}(\mathbf{x}) = J(\mathbf{x} + \mathbf{u}(\mathbf{x}))$. In a perfect world, then $I(\mathbf{x}) = J_{deformed}(\mathbf{x})$, though this is rarely the case. Figure 2.2 visualizes the deformation of an image under this standard model. Gradient descent optimization of an elastic mapping may be summarized (crudely) as:

$$\text{Compute the similarity gradient: } \nabla E = \partial_{\mathbf{u}} \Pi(I, J(\mathbf{x} + \mathbf{u}(\mathbf{x}))).$$
$$\text{Update the deformation field: } \mathbf{u}(\mathbf{x}) \leftarrow \mathbf{u}(\mathbf{x}) + \delta \nabla E.$$
$$\text{Regularize the deformation field: } \mathbf{u}(\mathbf{x}) \leftarrow G_\sigma \star \mathbf{u}(\mathbf{x}), \tag{1}$$

where Π is the similarity, δ is a gradient step length and G_σ is a gaussian smoother. This optimization is captured in the following ANTS command:

```
ANTS 3 -m PR[AtlasHead.nii,crAnatomical.nii,1,4] -r Gauss[0,3]
-o ElasticTest.nii.gz -i 30x20x10  -t Elast[1.5]
```

Here, we use the PR metric (a cross-correlation implementation) with window radius 4, weight 1 and gradient step length 1.5. The optimization will be performed over three resolutions with a maximum of 30 iterations at the coarsest level, 20 at the next coarsest and 10 at the full resolution. We use a Gaussian regularizer with a sigma of 3 that operates only on the deformation field and not on the similarity gradient, as 0 is passed as the first parameter to `Gauss`. One may see the correspondence, yet again, between the ANTS call and the optimization scheme. The optimization will stop when either the energy cannot be smoothly minimized or the maximum number of iterations is reached. BSpline regularization is also available in ANTS – see the DMFFD section below.

Warping and Invertibility. ANTS uses physical space to define mappings. The origin etc is in the coordinates of the world in which the picture (e.g. MRI) was taken. Then, warping between physical coordinates is relatively easy. Differences in bounding boxes etc present no problem – assuming you avoid inconsistent headers i.e. origins, directions, data orientation. One may use PrintHeader to check the data and run simple, fast tests (few iterations) to perform sanity checks before running through loads of data. An ANTS
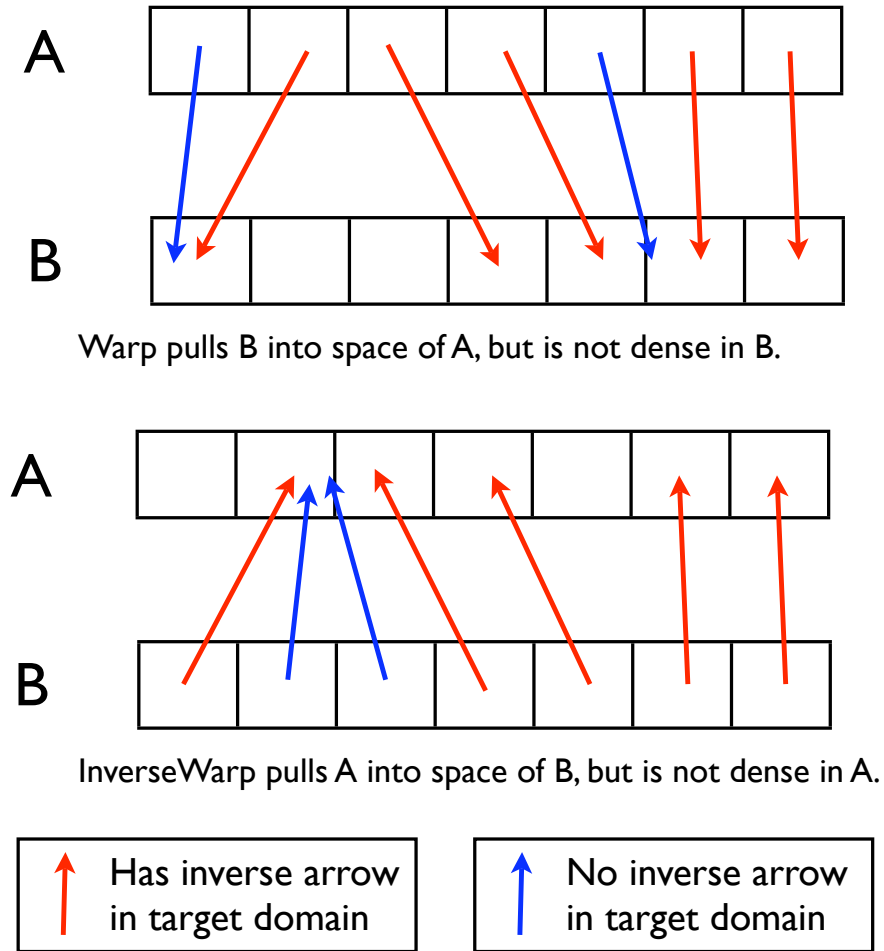
Figure 5: Digital invertibility presents some limitations. Here, we see that invertibility is not exact but is gained only by interpolation. Thus, in three-dimensional scenarios in particular, there are fundamental limits to the degree of invertibility that may be achieved. The second and third voxels – from left – in image A undergo an expansion by a factor of 2. That is, under the mapping, 2 voxels are mapped to 4. This gives the definition of the Jacobian – computed by ANTSJacobian – which is a unitless measure defined by the ratio of volumes. Thus, $\mathcal{J}(\mathbf{x}) = V(\phi(\mathbf{x}))/V(\mathbf{x})$ where V represents the volume operation and $\mathbf{x}$, here, may be a small object. Thus, if ϕ – the mapping – causes expansion, then $\mathcal{J}(\mathbf{x}) > 1$.

deformation consists of a standard naming prefix plus a standard naming extension. We usually assume nii but other file types may be used. The standard naming extension is Warpxvec.nii for the x component of the deformation with similar naming for y and z. Each component is stored in a separate scalar image. The value of a voxel of a deformation/warp component image is the physical space displacement from that voxel. Note that the inverse mapping is stored in InverseWarpxvec.nii and provides the mapping in the opposite direction. Note that an inverse – in the digital domain – is only approximate as shown in figure 2.2. A few important notes follow. (1) Deformation directionality: Warps/deformations applied to images occur in the opposite direction from warps/deformations applied to points. That is, if we map image B to A using

```
ANTS 2 -m PR[A,B,1,2] -o OUTPUT
```

then we get a deformation called `OUTPUTWarp+extensions` and an affine transform called `OUTPUTAffine.txt`. This composite mapping - when applied to B - will transform B into the space of A.
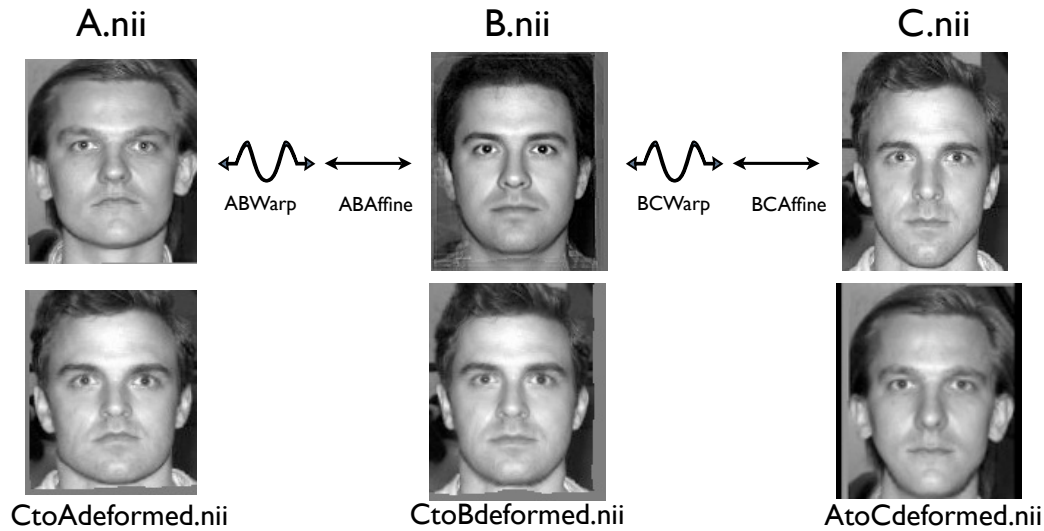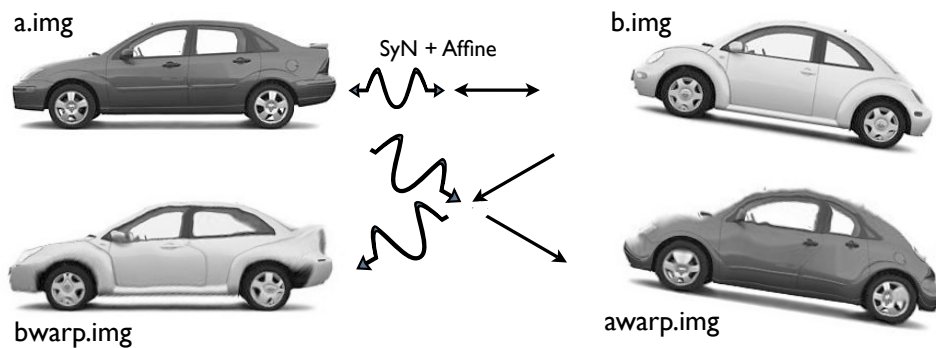
Figure 6: Here, we detail how one applies a deformation and associated inverses. We also see how WarpImageMulti-Transform can be used to concatenate a series of transformations.

However, if we have points defined in B that we want to map to A, we have to use `OUTPUTInverseWarp` and the inverse of `OUTPUTAffine.txt`. This is because the domain and range of the map's definition need to be dense in a way that is appropriate for the data type. Figure 2.2 illustrates the concept. (2) Older Image Formats: older image formats (e.g. Analyze) may not have proper origin/offset/direction support! In these cases, we recommend converting to nii and verifying that data overlays properly. (3) Transorm Composition: Composition of transforms may be achieved with ComposeMultiTransform. (4) Warping with inverses and concatenations – viable when using diffeomorphisms – are described in figure 2.2.

Diffeomorphic Transformations. The elastic mapping space – as indicated above – may prove inadequate for some large deformation mapping scenarios. Figures 2.2 illustrate the changing performance one may get in switching from affine to elastic to diffeomorphic normalization. Figure 2.2 shows how one may use ANTS to achieve a state-of-the-art diffeomorphic mapping. The ANTS diffeomorphic model chosen for this example – symmetric normalization – is invariant to the order of the input images (although the affine mapping is not). An additional advantage of the diffeomorphic model over the elastic model is that both forward and inverse transformations are computed thus allowing one to deform fixed to moving and also moving to fixed. The transformation model chosen here – `SyN[0.25]`– may be replaced with other diffeomorphic transformation models. The most general is global-in-time SyN via `SyN[0.25,2,0.01]` , where the time step for integration is 0.01 (lower is more accurate). A fast approximate solution may be gained through exponential mapping via `Exp[0.25,10]`, where 10 integrations points are used. Update schemes for diffeomorphic optimization are similar to the elastic case and are described elsewhere [5, **?**].

Figure 7: This example shows the benefit of the symmetric normalization model – invertibility, symmetry, highly deformable and accurate registration. The transformation model used here is highlighted in red.

2.3   ANTS Similarity Terms

Here is a script that will let you experiment with different similarity term and transformation model combinations. A few are listed, with parameters that are a reasonable starting point.

```
II=r16slice.nii # change to your own images
JJ=r64slice.nii
DIM=2 # two-dimensional images
ITS=" -i 100x100x30 " # 3 optimization levels
TSYNWITHTIME=" -t SyN[1,2,0.05]  -r Gauss[3,0.5] "
TGREEDYSYN=" -t SyN[0.25]  -r Gauss[3,0] "
TELAST=" -t Elast[1] -r Gauss[0,3] "
TEXP="  -t Exp[0.5,10] -r Gauss[0,3] "
LABELGUIDED=" -m PSE[${II},${JJ},${II},${JJ},0.75,0.1,25,0,10] "
INTMSQ=" -m MSQ[${II},${JJ},1,0.1] "
INTMI=" -m MI[${II},${JJ},1,32] "
INTPR=" -m PR[${II},${JJ},1,4] "
NAME=TEST
INT=$INTPR
TRAN=$TSYN
ANTS $DIM -o $NAME $ITS  $TRAN  $INT
INVW=" -i ${NAME}Affine.txt ${NAME}InverseWarp.nii "
FWDW=" ${NAME}Warp.nii ${NAME}Affine.txt "
WarpImageMultiTransform  2 ${II} a.nii -R ${JJ} $INVW
WarpImageMultiTransform  2 ${JJ} b.nii -R ${II} $FWDW
```

Convergence occurs under two conditions: (1) either the maximum number of iterations are reached at a given optimization level or (2) the slope of the change in the optimization objective is negative or very small. *Note: ANTS does not like spaces between brackets in the call to the metrics.*

2.4   Choosing a Metric

ANTS supports both volumetric registration and point set registration. The image / point set similarity metrics in ANTS are unified in the form of a function on the images or the point sets:

$$\textbf{Simlarity}[\textbf{fixedImage}, \textbf{movingImage}, \textbf{weight}, \textbf{parameters}].$$

The similarity type for the deformation transformation is specified by `-m` option, which contains two parts: simarity type and the parameters inside the brackets. (Note: no white spaces exist between parameters.) The possible similarity metrics for volumetric images are:

- Cross correlation: `-m CC[fixedImage,movingImage,weight,radius]`. This metric works well for intra-modality image registration. For exxample, `-m CC[fixed.nii,moving.nii,1,5]` specifies:
    - the fixed image: fixed.nii
    - the moving image: moving.nii
    - weight for this metric is 1 (i.e. only this metric drives the registration).

– the region radius for computing cross correlation is 5

- Mutual information: `-m MI[fixedImage,movingImage,weight,number-of-histogram-bins]`. This metric works both well for intra-modality and inter-modality image registration. For example, the first three parameters in `-m MI[fixed.nii,moving.nii,1,32]` is similar to the example above in cross correlation, except that the last parameter means that the number of bins in computing mutual information is 32.

- PR: `-m PR[fixedImage,movingImage,weight,radius]`. This metric works for intra-modality image registration. [TODO: what is different with CC???]. The meaning of parameters are similar to cross correlation.

- Mean square difference: `-m MSQ[fixedImage,movingImage,weight,0]`. This metric works for intra-modality image registration. The last parameter `0` is a padding value of no real meaning. For example, `-m MSQ[fixed.nii,moving.nii,1,0]`.

ANTS also support registration of point sets. The supported formats for point sets can be found in I/O section. The similarity metrics for point sets are:

- Point set expectation:

```
-m PSE [fixedImage,movingImage,fixedPoints,movingPoints,weight,
pointSetPercentage,pointSetSigma,boundaryPointsOnly,
kNeighborhood,PartialMatchingIterations=100000]
```

  – `fixedImage`: defines the space domain of the fixed point set.
  – `movingImage`: defines the space domain of the moving point set.
  – `fixedPoints`: defines the coordinates of the fixed point set. It can be an image with discrete positive labels, a VTK format point set file, or a text file. Details can be found in I/O section (TODO).
  – `movingPoints`: defines the coordinates of the moving point set.
  – `weight`: weight for this metric. `1` means that only this metric drives the registration.
  – `pointSetPercentage`: the percentage of points to be randomly sampled used in the registration.
  – `pointSetSigma`: the ???
  – `boundaryPointsOnly`: [TODO] 1 (or "true") means only the boundary points in the point sets are used to drive registration.
  – `kNeighborhood` is a positive discrete number. The first k neighbors are used to compute the deformation during the registration.
  – `PartialMatchingIterations` controls the symmtry in the matching. This option assumes the complete labeling is in the first set of label parameters ... more iterations leads to more symmetry in the matching - 0 iterations means full asymmetry

- Jensen-Tsallis BSpline

fixed image │ moving image │ MSE

Figure 8: registration using mean square intensity difference

```
-m JTB[fixedImage,movingImage,fixedPoints,movingPoints,weight,
pointSetPercentage,pointSetSigma,boundaryPointsOnly,kNeighborhood,alpha,
meshResolution,splineOrder,numberOfLevels,useAnisotropicCovariances]
```

- – `fixedImage`: defines the space domain of the fixed point set.
- – `movingImage`: defines the space domain of the moving point set.
- – `fixedPoints`: defines the coordinates of the fixed point set. It can be an image with discrete positive labels, a VTK format point set file, or a text file. Details can be found in I/O section (TODO).
- – `movingPoints`: defines the coordinates of the moving point set.
- – `weight`: weight for this metric. `1` means that only this metric drives the registration.
- – `pointSetPercentage`: the percentage of points to be randomly sampled used in the registration.
- – `pointSetSigma`: the ???
- – `boundaryPointsOnly`: [TODO] `1` (or "true") means only the boundary points in the point sets are used to drive registration.
- – `kNeighborhood` is a positive discrete number. The first *k* neighbors are used to compute the deformation during the registration.
- – `alpha`
- – `meshResolution`
- – `splineOrder`
- – `numberOfLevels`
- – `useAnisotropicCovariances`

In current implementation, the affine registration only supports two types of similarity metrics on volumetric images, which are specified using `--affine-metric-type`:

- Mutual information, specified as `--affine-metric-type MI`. This usually works for both inter and intra-modalities in 3D. Also, the options used in computing mututal information in affine registration can be specified as `--MI-option N1xN2`. The first parameter `N1` specifies the number of bins. The second parameter `N2` specifies the number of samples. For example: `--MI-option 32x8000`.

- Mean square difference, specified using `--affine-metric-type MSE`. There is no options for this metric.

Fig. 2.4 shows the registration result using intensity difference with the following command:

```
ANTS 2 -m MSQ[r16slice.nii,r64slice.nii,1,0] -r Gauss[3,0] -t SyN[0.5] -i 50x50x30
```

Here is an example script to register a pair of images using different image metrics:

```
#use cross-correlation with cc region radius 2
ANTS 2 -m CC[r16slice.nii,r64slice.nii,1,2] -r Gauss[3,0] -t SyN[1] -i 50x50x30
WarpImageMultiTransform 2 r64slice.nii resCC2.nii Warp.nii Affine.txt
MeasureImageSimilarity 2 0 r16slice.nii resCC2.nii metricexamplelog.txt
MeasureImageSimilarity 2 1 r16slice.nii resCC2.nii metricexamplelog.txt
MeasureImageSimilarity 2 2 r16slice.nii resCC2.nii metricexamplelog.txt
ConvertToJpg resCC2.nii resCC2.jpg

#use cross-correlation with cc region radius 4
ANTS 2 -m CC[r16slice.nii,r64slice.nii,1,4] -r Gauss[3,0] -t SyN[1] -i 50x50x30
WarpImageMultiTransform 2 r64slice.nii resCC4.nii Warp.nii Affine.txt
MeasureImageSimilarity 2 0 r16slice.nii resCC4.nii metricexamplelog.txt
MeasureImageSimilarity 2 1 r16slice.nii resCC4.nii metricexamplelog.txt
MeasureImageSimilarity 2 2 r16slice.nii resCC4.nii metricexamplelog.txt
ConvertToJpg resCC4.nii resCC4.jpg

#use cross-correlation with cc region radius 2
ANTS 2 -m PR[r16slice.nii,r64slice.nii,1,2] -r Gauss[3,0] -t SyN[1] -i 50x50x30
WarpImageMultiTransform 2 r64slice.nii resPR2.nii Warp.nii Affine.txt
MeasureImageSimilarity 2 0 r16slice.nii resPR2.nii metricexamplelog.txt
MeasureImageSimilarity 2 1 r16slice.nii resPR2.nii metricexamplelog.txt
MeasureImageSimilarity 2 2 r16slice.nii resPR2.nii metricexamplelog.txt
ConvertToJpg resPR2.nii resPR2.jpg

#use cross-correlation with cc region radius 4
ANTS 2 -m PR[r16slice.nii,r64slice.nii,1,4] -r Gauss[3,0] -t SyN[1] -i 50x50x30
WarpImageMultiTransform 2 r64slice.nii resPR4.nii Warp.nii Affine.txt
MeasureImageSimilarity 2 0 r16slice.nii resPR4.nii metricexamplelog.txt
MeasureImageSimilarity 2 1 r16slice.nii resPR4.nii metricexamplelog.txt
MeasureImageSimilarity 2 2 r16slice.nii resPR4.nii metricexamplelog.txt
ConvertToJpg resPR4.nii resPR4.jpg


#use intensity difference with radius 0 -- radius no effect on intensity difference
ANTS 2 -m MSQ[r16slice.nii,r64slice.nii,1,0] -r Gauss[3,0] -t SyN[0.5] -i 50x50x30
WarpImageMultiTransform 2 r64slice.nii resMSQ.nii Warp.nii Affine.txt
MeasureImageSimilarity 2 0 r16slice.nii resMSQ.nii metricexamplelog.txt
MeasureImageSimilarity 2 1 r16slice.nii resMSQ.nii metricexamplelog.txt
MeasureImageSimilarity 2 2 r16slice.nii resMSQ.nii metricexamplelog.txt
ConvertToJpg resMSQ.nii resMSQ.jpg


# use mutual information with radius 0 and step-size 1.0
ANTS 2 -m MI[r16slice.nii,r64slice.nii,1,32] -r Gauss[3,0] -t SyN[1] -i 50x50x30
WarpImageMultiTransform 2 r64slice.nii resMI2.nii Warp.nii Affine.txt
MeasureImageSimilarity 2 0 r16slice.nii resMI2.nii metricexamplelog.txt
MeasureImageSimilarity 2 1 r16slice.nii resMI2.nii metricexamplelog.txt
MeasureImageSimilarity 2 2 r16slice.nii resMI2.nii metricexamplelog.txt
```
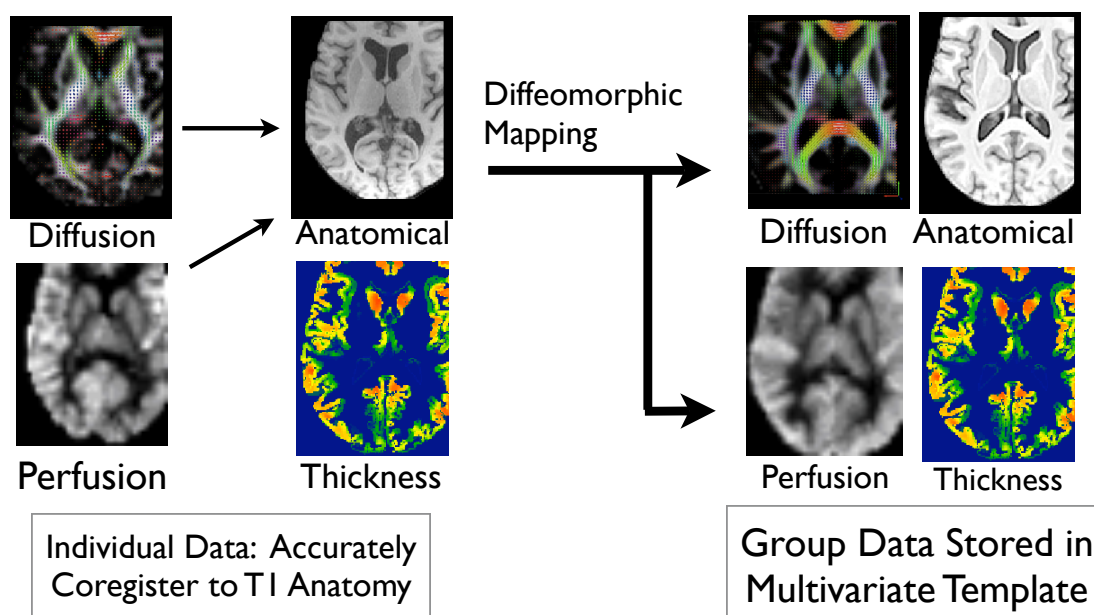
Figure 10: The current version of ANTS may perform normalization of multiple modalities by combining *intrasubject, intermodality* mappings with *intersubject, intramodality* maps to a group template. In brain imaging, the intersubject maps are usually guided by the T1 component.

```
ConvertToJpg resMI2.nii resMI2.jpg
```

2.5   Diffusion Tensor Normalization

The ImageMath program – via TensorFA and TensorMeanDiffusion – may derive scalar images from DTI. Such images may be used to distortion correct DTI to T1 or to map DTI together. See the ANTS/Scripts program called `antswithdt.sh` for a ready to go example. Figure 2.5 shows what might happen if your tensor entries are stored in the wrong order. ANTS expects the nifti standard ordering of the DT six vector.

2.6   Multivariate Normalization with ANTS

Multivariate normalization may be performed in two ways with ANTS. First, as shown in figure 2.6, intra-subject mappings may be used to conform all modalities into a common subject space defined by the "anchor" modality. In brain imaging, this is usually the T1 component.

A second type of multivariate normalization is able to use the T1 and DTI components directly in the optimization of the mapping. An example of this type of call is below, where we assume that one has resampled the DT data to the space of T1:
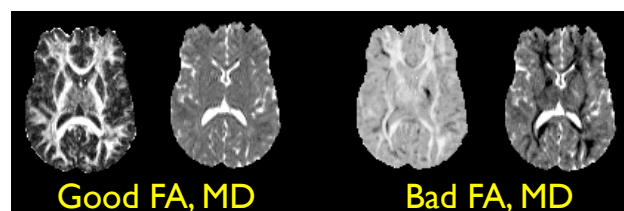

Figure 9: The FA and mean diffusion of the tensor may be corrupted, as at right, if the order of the DT entries is wrong.

```
ANTS 3 -m PR[T1template.nii,T1subject.nii,1.25,4]
```
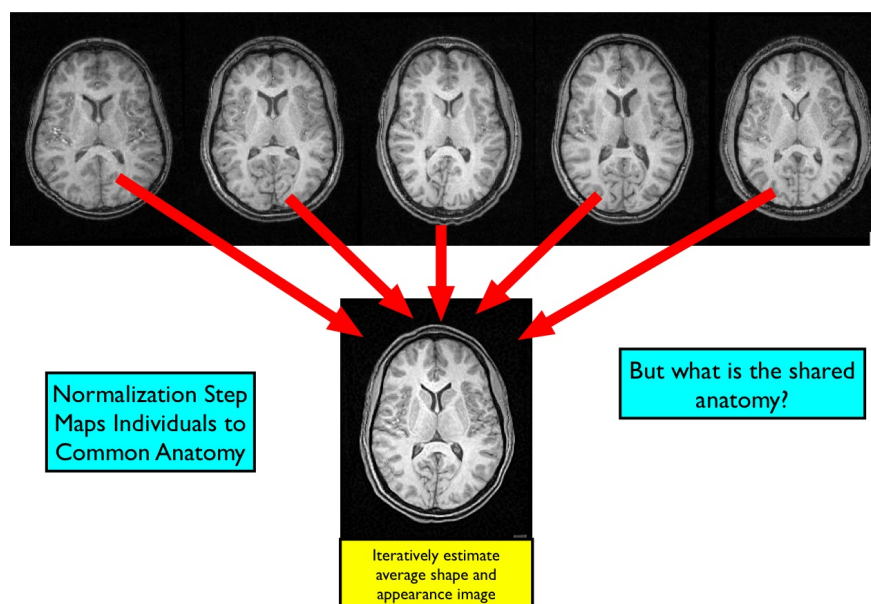
Figure 11: This example may be recreated by the user. The shared anatomy – across this dataset – is recovered in the derived optimal template.

```
        -m PR[FAtemplate.nii,FAsubject.nii,1,4]
-o MultiVar -i 30x30x20  -r Gauss[3,0] -t SyN[0.25]
```

In this case, dense features within white matter are gained by using the DT component in addition to the T1 component. The T1 PR metric, here, is weighted slightly more than the FA PR metric, 1.25 vs. 1.0. *The convergence criterion for the multivariate scenario is a slave to the* **last** *metric you pass on the ANTS command line*.

2.7   Optimal Template Construction with ANTS

A useful script for setting up optimal template construction is available in ANTS/Scripts. Two versions of available : serial and parallel. Data and a script for building an average face template is available the ANTS/Examples/Data directory. Two key points about optimal templates: 1. The outcome stabilizes at around ten images, for most populations. That is, if we randomly select images from individuals within the same demographic distribution, we will end up with very similar average images. 2. Optimality, for the ANTS-SyN approach, is defined by the minimum shape and appearance distance image. See the template page for some examples of this and users may contact ANTS developers for previously derived template images that may be useful.

A Concrete Template Construction Example, Step-by-Step

For this example, we will use the data in "ANTS/Examples/Data/B*.tiff." The files are B1.tiff, B2.tiff, B3.tiff, B4.tiff and B5.tiff. They are single slices from a dataset of real MRI brain images – individuals around 15 years of age. Our goal is to derive a "most representative" single image from this population. The steps are:

1. Make a directory for this example. Copy or link all the B*tiff images into it.

2. Copy the file ANTS/Scripts/buildtemplate.sh into that directory.

3. On the command line, within that directory, run the following command: sh `buildtemplate.sh` to get usage.

4. If you follow the usage, you will then call the script something like this: `sh buildtemplate.sh 2 B tiff OUT 4 .`

5. The script should exit and ask if the parameters are ok for your problem. Check the path to the programs, the fact that all the images you want to use are being listed and that the template output name is ok with you. Once everything is prepared, then edit the script to remove the exit call at the end of the parameter check.

6. Call the script as in step 4 and then wait for the output. The template will be called something like (in this example) `OUTtemplate.nii`.

The script is fairly easy to alter so that you can send the whole thing to distributed computing. Typically, one would use voxbo or the qsub program available in the Sun Grid Engine. This is what is done in `buildtemplateparallel.sh`. Figure 2.7 shows the data and a result that may be recreated by the user.

2.8   Notes on Large Deformation Mapping

Figure 2.8 defines what one might expect from a high-resolution, large deformation, successful normalization. Major and many minor features are well aligned between these images to an extent that is approaching the limits allowable by biological variation. **Turning "Failures" into Successes:** Below are some pointers to follow if you are unable to recreate such normalization quality. Usually, reasons for registration failure are one of a very few things:

1. The initialization is so off mark that affine registration fails – thus meaning all subsequent deformable registration will not be meaningful.

2. The information within headers is inconsistent e.g. origins are defined in different ways, the "directions" are not correct or some combination of these. The PrintHeader executable can help one in debugging this type of problem. Also, the ImageMath function CompareHeadersAndImages can fix some problems of this type.

3. The similarity or transformation model is inappropriate or has too small a capture region for the problem.

An example of this final point is given here. A recent study used the following call to ANTS across a dataset of elderly subjects:

```
ANTS 3 -m PR[template.nii.gz,subject.nii.gz,1,2] -i 10x50x50x20
 -o subjectmap.nii -t SyN[0.25] -r Gauss[3,0]
```

This succeeded for all subjects but one. This subject had severe neurodegeneration which caused "under-normalization" to result. A larger deformation mapping was required and so we increased the maximum
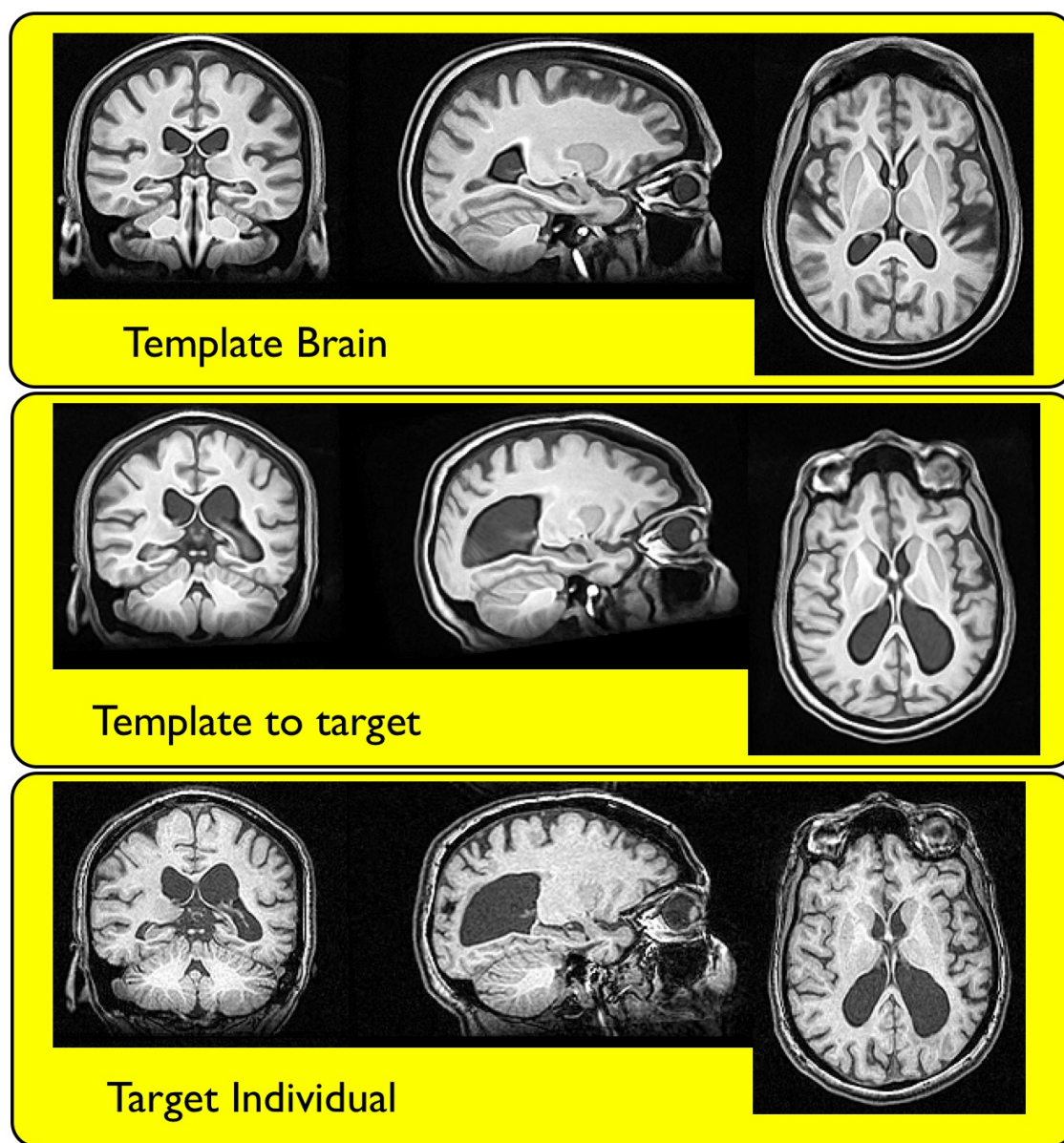
Figure 12: ANTS succeeds in a challenging normalization scenario. Comparing the template mapping to the individual shows that much of the cortex is well-aligned, as is the hippocampus, despite the relatively large difference between the initial template and the target. Here, one might note a limitation of whole brain mapping: occipital lobe sulcal variation is highly idiosyncratic and extremely difficult to capture, in particular when there is such severe atrophy nearby.
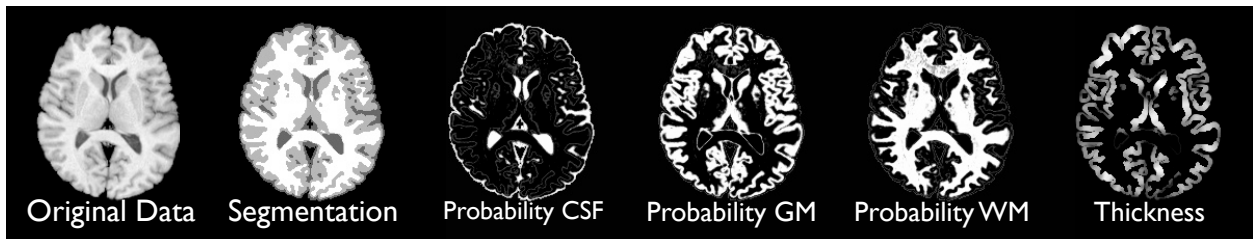
Figure 13: ANTS provides basic Markov Random Field regularized Gaussian model tissue segmentation. The ANTS program N3BiasFieldCorrection is very valuable as preprocessing for this naive approach to segmentation. The last panel – far right – shows the thickness derived from the white matter and gray matter probabilities where a prior on thickness was used to prevent thickness overestimation. Thickness was derived with DiReCT [9], a software tool available in binary form only – a future release may provide code.

number of iterations (the "-i" parameter). One might also increase the gradient-descent step-size (SyN[0.25] =¿ SyN[0.5]). Large deformation mapping is challenging not only because of the amount of deformation, but also because the "true" solution is difficult to find. Keeping this in mind, we also increased the "span" (radius) of the correlation window in the similarity metric (which also increases computation time). These modified parameters succeeded:

```
ANTS 3 -m PR[template.nii.gz,subject.nii.gz,1,4] -i 100x100x100x20
 -o subjectmap.nii -t SyN[0.25] -r Gauss[3,0]
```

Key changes were to increase the radius of the correlation and to allow more iterations at the coarsest resolution. This two coarsets levels of the computation take only about 10 percent of the time (about 5-10 minutes depending on the machine) but accounts for the large majority of the shape variation. Note that this is a 3D example – that is why the eyes appear only in the frame at left.

3 Image Segmentation and Labeling

ANTS has tools for both tissue based segmentation and prior-based segmentation that leverages spatial priors, usually based on a template mapping.

3.1 Basic Segmentation

A basic MRF segmentation algorithm is available in ANTS ImageMath. We apply the segmentation to the example data in: ANTS/Examples/Data/r16slice.nii.

```
ImageMath 2 r16slice.nii Segment r16slice.nii 4 0
% output = r16slice_seg.nii, r16slice_prob_0.nii, r16slice_prob_1.nii, r16slice_prob_2.nii .
```

The parameter 4 indicates 3 tissues plus background are sought. The 0 indicates that there are no prior images available to guide the segmentation. Output is shown in figure 3.1 and – with priors – in figure 3.2.

3.2 Prior and Template-based Image Segmentation

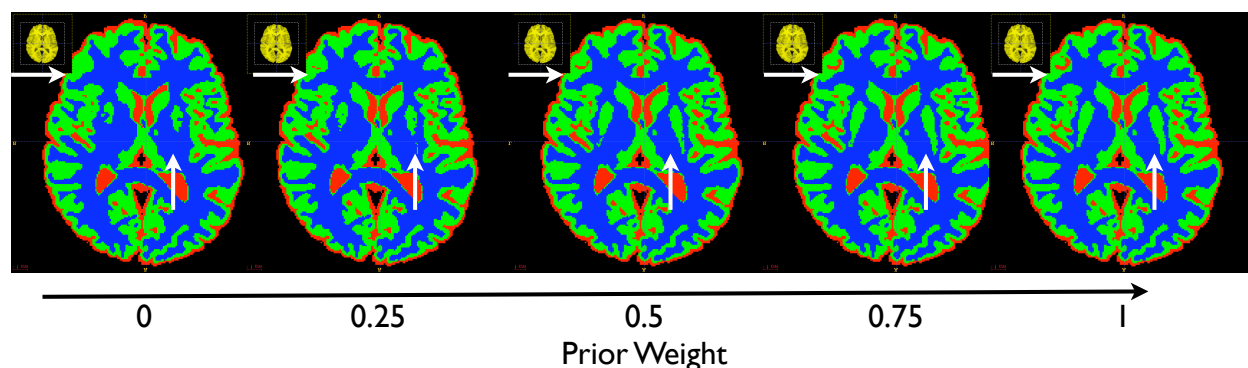The same algorithm may be augmented to perform Prior-based segmentation.

Figure 14: ANTS extends basic Markov Random Field regularized Gaussian model tissue segmentation to include priors, which allow spatially varying tissue models to determine segmentation. The ANTS program N3BiasFieldCorrection is less critical as preprocessing when using this approach to segmentation. Here, we see the improvement in segmentation as the locally varying prior models are weighted more heavily. The last panel – far right – shows the segmentation derived from data driven with an initialization founded purely on prior models based on template mapping. The prior model brings out the caudate and some CSF – highlighted by arrows – as their weight increases.

```
ImageMath 2 r16sliceprior.nii Segment r16slice.nii 4 0.5
   csfprior.nii wmprior.nii  gmprior.nii
% output = r16sliceprior_seg.nii  r16sliceprior_prob_0.nii
% r16sliceprior_prob_1.nii  r16sliceprior_prob_2.nii .
```

The parameter 4 indicates 3 tissues plus background are sought. The 0.5 indicates we weight the priors equally as the data term and use a spatially varying set of Gaussians to estimate the segmentation. In this way, data from the priors may be used to modify and guide the segmentation in a locally varying way, accomodating for both inhomogeneity and the different imaging signature that different tissues provide. Prior images should be of the same size and dimension as the input data and should have intensities in the range $[0, 1]$.

3.3   Cortical Thickness

Two forms of cortical thickness estimation from probability maps are available in ANTS: first, the traditional Laplacian cortical thickness estimation and, second, the more recently developed Diffeomorphic Registration-based Cortical Thickness (DiReCT) [9]. Both methods estimate thickness of an object from probability or binary images that define object boundaries. This tool is mainly of interest in brain mapping and cardiac imaging for morphometric studies. Cortical thickness, for instance, is known to correlate with language development and IQ in adolescents.

3.4   User/Label-Guided Normalization for Hippocampus Mapping

See the link for details on hippocampus labeling: http://picsl.upenn.edu/ANTS/hipptutorial.php.
Expectation-Based Point-Set Registration. Here, we apply the expectation-based point set registration method for mapping labeled points sets. ITK-SNAP may be used to label images and exported segmentation images may be input to the PSE metric below, as labeled data. The Frown and Smile data is used as example. This data is available in the ANTS/Examples/Data/ directory.
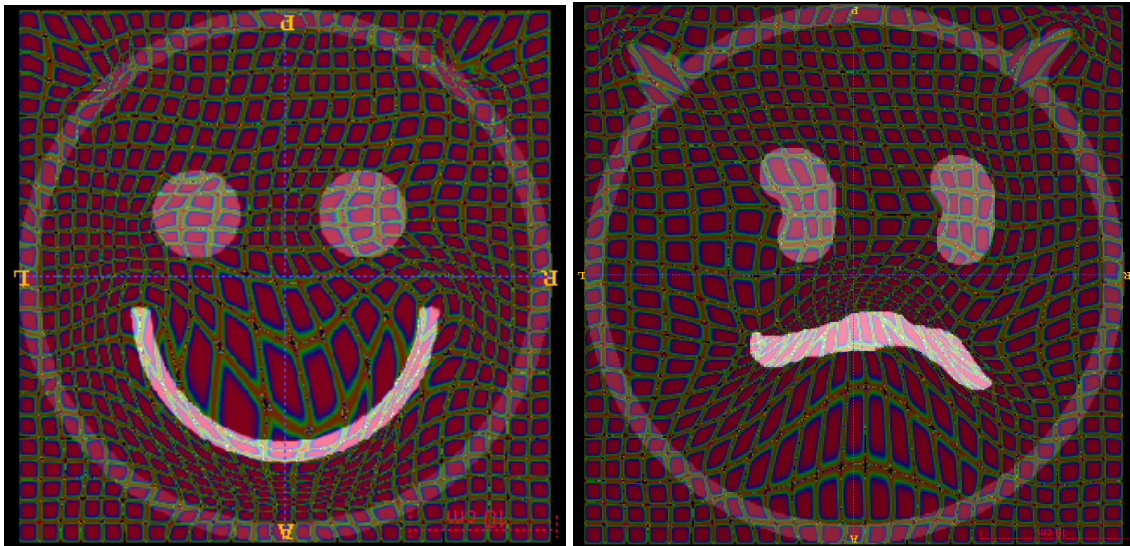
Figure 15: Expected output for the frown to smile shows a smooth, though large deformation. The grids are overlaid on the deformed images.

```
ANTS 2 -o PSEtest  -i 91x70x55x40x30  -r Gauss[3,0.] -t SyN[0.2]
   -m  PSE[Frown.nii,Smile.nii,Frown.nii,Smile.nii,0.75,0.1,11,0,10]
   -m  MSQ[Frown.nii,Smile.nii,1,0]     --number-of-affine-iterations 0

WarpImageMultiTransform  2 Frown.nii FrownToSmile.nii -R Smile.nii
   -i PSEtestAffine.txt PSEtestInverseWarp.nii

WarpImageMultiTransform  2 Smile.nii SmileToFrown.nii -R Frown.nii
    PSEtestWarp.nii PSEtestAffine.txt

CreateWarpedGridImage 2 PSEtestInverseWarp.nii grid1.nii
CreateWarpedGridImage 2 PSEtestWarp.nii grid2.nii
```

This example should run on the downloaded ANTS data so you may see the results.

4   Application to Studies

4.1   Brain Mapping in the Presence of Lesions

Many cases violate the basic assumption of a one-to-one mapping from a template image to a target image. Brain lesions caused by stroke or traumatic brain injury are a common instance of this.

The ANTS toolkit handles this situation by a constrained cost-function masking approach. In short, the mapping within an excluded region (as determined by an inclusive mask) is determined by the solution outside of the region. To achieve this with ANTS, one would use a call similar to this:

```
ANTS 3 -m PR[tp22_s1.nii,template.nii.gz,1,4] -i 50x20x0
   -o tp22map -t SyN[0.5] -x mask.nii.gz -r Gauss[3,0]
```
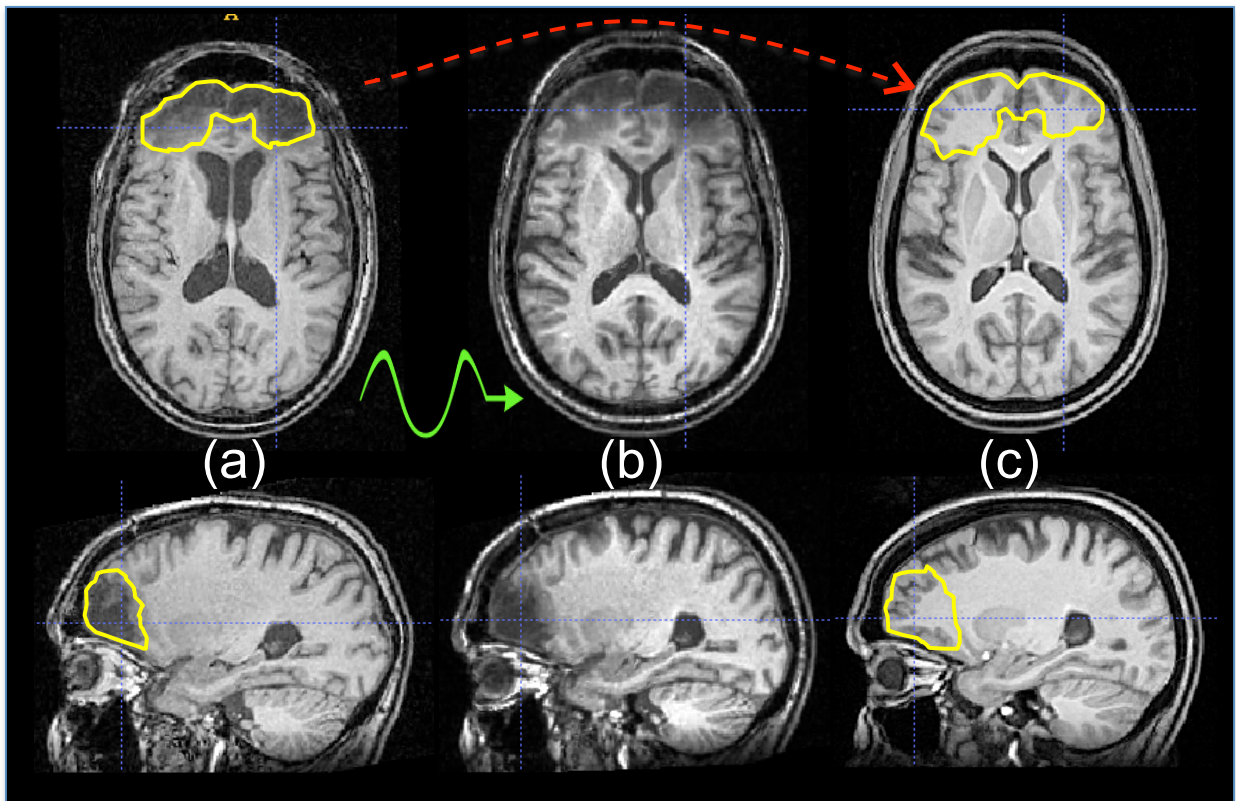
Figure 16: Here, we see a semi-automated approach for lesion analysis based on diffeomorphic normalization. The user outlines the lesion site in the subject space, as shown in (a). Diffeomorphic mapping is then used to deform the subject image in (a) into the space of the reference template. The deformed subject is shown in panel (b). Panel (c) shows the reference template space with the estimated position of the subject's lesion overlaid on the healthy tissue of the template. This approach enables one to compare the subject image against prior information stored in the template space such as anatomical labels or statistics on the appearance of lesioned and/or normal tissue.

The additional parameter needed for this approach is the `-x mask.nii.gz` which is the mask defined in the "fixed" (or individual) image space. Mask values above 0.5 are considered non-outlier or non-lesioned regions. The labeling may be performed with ITK-SNAP. Figure 4.1 illustrates the approach. Once the ANTS mapping is solved, one is able to estimate the location of the lesion in the template space by mapping the inverse lesion mask to the template. To take the negative image of the inclusive mask and warp the mask to the template :

```
ImageMath 3 lesionmask.nii.gz Neg inclusivemask.nii.gz
WarpImageMultiTransform 3 lesionmask.nii.gz lesiontotemplate.nii.gz
   -R template.nii.gz -i tp22mapAffine.txt tp22mapInverseWarp.nii
```

More   details   on   this   approach   are   being   prepared   for   publication.   See   also: http://www.ncrrn.org/papers/methodology_papers/sp_norm_kim.pdf.

4.2   Statistical Mapping with ANTS: Morphometry, Function, Jacobian, Thickness

ANTS has been applied in a wide array of imaging studies [4, 3, 7, 9, 12, 13, 14, 17, 18, 6, 5, 10, 11, 15, 16, 19, 1, 2, 8]. All of these studies benefit in some way from normalization whether the topic is cortical thickness, Jacobian-based morphometry, volumetric morphometry or functional studies. From a broad perspective, each of these applications requires the same steps:

1. Preprocess images – bias correction, segmentation, etc. Potentially construct an optimal template.

2. Normalize images – run `ants.sh` to map a population of images to a template and store the deformations.

3. Derive any data from the deformation that may be necessary, e.g. the log-Jacobian.

4. Apply the warp to any images one may want to analyze in template space.

5. Perform statistics in template space over the region of interest, e.g. all cortex.

We now illustrate this procedure using ANTS tools and example images from ANTS/Examples/Data. The expected output is in figure 4.2.

```
1.   ImageMath 2 r64slice.nii Segment r64slice.nii 4 0
1.   ImageMath 2 r16slice.nii Segment r16slice.nii 4 0
2.   sh ants.sh 2 r16slice.nii r64slice.nii SPM 100x100x100
3.   SmoothImage 2 r64slice_prob_1.nii  1. r64slice_prob_1.nii
3.   WarpImageMultiTransform 2 r64slice_prob_1.nii  r64slice_prob_1_norm.nii
      -R r16slice.nii SPMWarp.nii SPMAffine.txt
3.   ANTSJacobian 2 SPMWarp.nii SPM 1  # take the log
4.   ImageMath 2 SPMlogjacobianmask.nii m SPMlogjacobian.nii r64slice_prob_1_norm.nii
5.   Repeat for a population and run statistics on log jacobians.
```

This example uses a standard Jacobian-based approach to morphometry. Here, we use the log-Jacobian because it is symmetric about zero and mask with the gray matter segmentation to restrict the analysis to the cortex. The Jacobian is discussed in figure 2.2 and visualization of the Jacobian is shown in figure 4.2. This Jacobian comes from the mapping shown in figure 2.8. Similar analyses may be performed on thickness images, functional images or fractional anisotropy images derived from the diffusion tensor modality. In all these cases, steps similar to those above would be performed.

4.3   Statistics with ANTS and "R"

R is an open-source, cutting-edge statistical package used by statisticians world-wide. ANTS is designed to interface with **R** via an ImageMath tool that converts image sets into an **R** compatible format. Three steps are required. First, convert image data into vector and matrix formats through a mask of your statistical region of interest. Second, read that data into **R** and run your statistics. Third, write out the data from **R** and convert back to an image. Here are the steps, in code examples,

1. Create a vector for your output:

```
ImageMath 3 testresultvec.nii ConvertImageSetToMatrix 1 mask.nii image1.nii
```
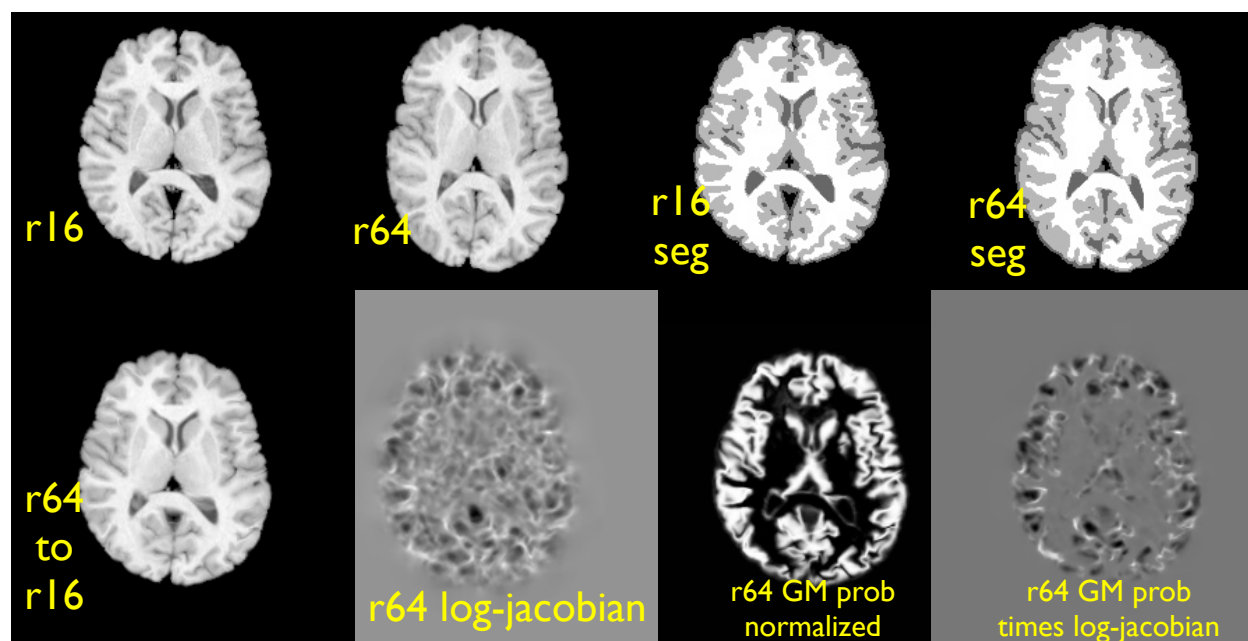
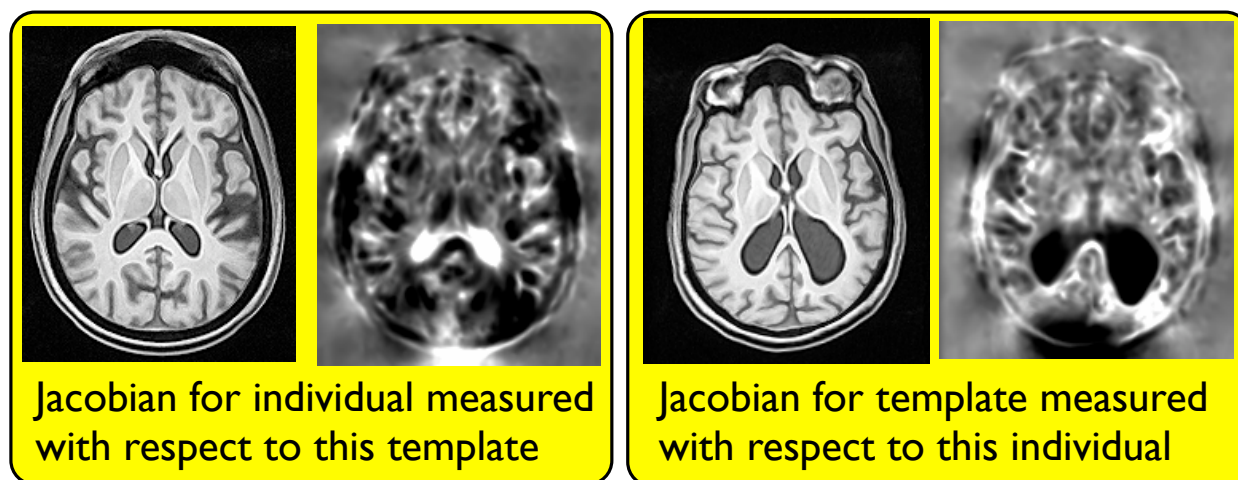Figure 17: The output from the ANTS morphometry example.

Figure 18: The Jacobian: Note that the bright values in the template image ventricles (left) indicate that the ventricles are relatively larger in the subject image. Similarly, the dark jacobian values in the individual image show that the template has smaller ventricles. This example is described on the large deformation page Large Def. Once one gains the Jacobian (or more appropriately, the log-Jacobian), then one may compute statistics across the population.

Here, we assume all images are called image1.nii, image2.nii, ... , imageN.nii so we can use a wildcard to pass to ImageMath and create a matrix for all your input data:

```
ImageMath 3 datamatrix.nii ConvertImageSetToMatrix 1 mask.nii image*.nii
```

2. Run **R** by reading your data above, performing whatever statistics you want and then writing out.

3. Convert the test result back to an image.

```
ImageMath 3 testresult.nii ConvertVectorToImage mask.nii testresultvec.nii
```

Most statistical requirements may be met with this setup.

5 Dependencies and Related Software

5.1 Dependencies and Compilation

ANTS depends on the most recent version of the Insight ToolKit (ITK) – www.itk.org. File types that may be read and written are all of those available in itk and a few more http://picsl.upenn.edu/ANTS/ioants.php.

ANTS and ITK are both compiled through pointing CMake – www.cmake.org – to a CMakeLists.txt file. The ANTS CMakeLists.txt is in the Examples directory.

See below for more details on compilation and binaries:
Compile and download: http://picsl.upenn.edu/ANTS/download.php.   We maintain a dashboard to give users an idea of what to expect from compilation:
http://picsl.upenn.edu/cdash/index.php?project=ANTS

5.2 Visualization and Quantification of ANTS Results

Use itk SNAP : www.itksnap.org or mricro / mricron for viewing images. The ANTS program StackSlices can be used to conveniently scan through a dataset in any of its axes. It is useful for checking both input data quality and output normalization quality. MeasureImageSimilarity will supply numerical measures of image similarity via three different metrics.

5.3 Statistics Beyond ANTS

Use SPM, **R** or npm (from Chris Rorden's mricron) for computing population statistics.

5.4 Pipelining with ANTS

Use PipeDream for automating large-scale processing:
PipeDream Homepage : https://sourceforge.net/projects/neuropipedream/
PipeDream is integrated with ANTS and automates more complex studies such as multivariate diffusion tensor and T1 cortical thickness studies, longitudinal mapping and reconstruction of nifti images from Dicom.

References

[1] Geoffrey K Aguirre, Andres M Komeromy, Artur V Cideciyan, David H Brainard, Tomas S Aleman, Alejandro J Roman, Brian B Avants, James C Gee, Marc Korczykowski, William W Hauswirth, Gregory M Acland, Gustavo D Aguirre, and Samuel G Jacobson. Canine and human visual cortex intact and responsive despite early retinal blindness from rpe65 mutation. *PLoS Med*, 4(6):e230, Jun 2007. 4.2

[2] B. Avants, C. Anderson, and M. Grossman. Tauopathic longitudinal gray matter atrophy predicts declining verbal fluency: A symmetric normalization study. In *Human Brain Mapping*, 2007. 4.2

[3] B. Avants, P. A. Cook, J. T. Pluta, J. nad Duda, H. Rao, J. Giannetta, H. Hurt, S. Das, and J. Gee. Follow-up on long term effects of prenatal cocaine exposure/poly-substance abuse on the young adult brain. In *Pediatric Academic Society, Annual Meeting*, 2009. 4.2

[4] B. Avants, P. A. Cook, J. T. Pluta, J. nad Duda, H. Rao, J. Giannetta, H. Hurt, S. Das, and J. Gee. Multivariate diffeomorphic analysis of longitudinal increase in white matter directionality and decrease in cortical thickness between ages 14 and 18. In *Human Brain Mapping, 15th Annual Meeting, oral presentation.*, 2009. 4.2

[5] B. B. Avants, C. L. Epstein, M. Grossman, and J. C. Gee. Symmetric diffeomorphic image registration with cross-correlation: Evaluating automated labeling of elderly and neurodegenerative brain. *Med Image Anal*, 12(1):26–41, Feb 2008. 1, 2.2, 4.2

[6] Brian Avants, Jeffrey T Duda, Junghoon Kim, Hui Zhang, John Pluta, James C Gee, and John Whyte. Multivariate analysis of structural and diffusion imaging in traumatic brain injury. *Acad Radiol*, 15(11):1360–1375, Nov 2008. 4.2

[7] Brian Avants, Alea Khan, Leo McCluskey, Lauren Elman, and Murray Grossman. Longitudinal cortical atrophy in amyotrophic lateral sclerosis with frontotemporal dementia. *Arch Neurol*, 66(1):138–139, Jan 2009. 4.2

[8] Brian B Avants, Hallam Hurt, Joan M Giannetta, Charles L Epstein, David M Shera, Hengyi Rao, Jiongjiong Wang, and James C Gee. Effects of heavy in utero cocaine exposure on adolescent caudate morphology. *Pediatr Neurol*, 37(4):275–279, Oct 2007. 4.2

[9] Sandhitsu R Das, Brian B Avants, Murray Grossman, and James C Gee. Registration based cortical thickness measurement. *Neuroimage*, 45(3):867–879, Apr 2009. 13, 3.3, 4.2

[10] M. Grossman, C. Anderson, A. Khan, B. Avants, L. Elman, and L. McCluskey. Impaired action knowledge in amyotrophic lateral sclerosis. *Neurology*, 71(18):1396–1401, 2008. 4.2

[11] Junghoon Kim, Brian Avants, Sunil Patel, John Whyte, Branch H Coslett, John Pluta, John A Detre, and James C Gee. Structural consequences of diffuse traumatic brain injury: A large deformation tensor-based morphometry study. *Neuroimage*, 39(3):1014–1026, Feb 2008. 4.2

[12] Arno Klein, Jesper Andersson, Babak A Ardekani, John Ashburner, Brian Avants, Ming-Chang Chiang, Gary E Christensen, D. Louis Collins, James Gee, Pierre Hellier, Joo Hyun Song, Mark Jenkinson, Claude Lepage, Daniel Rueckert, Paul Thompson, Tom Vercauteren, Roger P Woods, J. John Mann, and Ramin V Parsey. Evaluation of 14 nonlinear deformation algorithms applied to human brain mri registration. *Neuroimage*, 46(3):786–802, Jul 2009. 1, 4.2

[13] Lauren Massimo, Chivon Powers, Peachie Moore, Luisa Vesely, Brian Avants, James Gee, David J Libon, and Murray Grossman. Neuroanatomy of apathy and disinhibition in frontotemporal lobar degeneration. *Dement Geriatr Cogn Disord*, 27(1):96–104, 2009. 4.2

[14] J. Pluta, B. Avants, P. Yushkevich, S. Glynn, S. Awate, J. Detre, D. Mechanic, and J. C. Gee. Expectation matching for incomplete label driven semi-automated hippocampus segmentation in epilepsy. *Hippocampus*, page accepted, 2009. 4.2

[15] Tony J Simon, Zhongle Wu, Brian Avants, Hui Zhang, James C Gee, and Glenn T Stebbins. Atypical cortical connectivity and visuospatial cognitive impairments are related in children with chromosome 22q11.2 deletion syndrome. *Behav Brain Funct*, 4:25, 2008. 4.2

[16] Hui Sun, Brian B Avants, Alejandro F Frangi, Federico Sukno, James C Geel, and Paul A Yushkevich. Cardiac medial modeling and time-course heart wall thickness analysis. *Med Image Comput Comput Assist Interv Int Conf Med Image Comput Comput Assist Interv*, 11(Pt 2):766–773, 2008. 4.2

[17] Nicholas J Tustison, Brian B Avants, and James C Gee. Directly manipulated free-form deformation image registration. *IEEE Trans Image Process*, 18(3):624–635, Mar 2009. 4.2

[18] Paul A Yushkevich, Brian B Avants, John Pluta, Sandhitsu Das, David Minkoff, Dawn Mechanic-Hamilton, Simon Glynn, Stephen Pickup, Weixia Liu, James C Gee, Murray Grossman, and John A Detre. A high-resolution computational atlas of the human hippocampus from postmortem magnetic resonance imaging at 9.4 t. *Neuroimage*, 44(2):385–398, Jan 2009. 4.2

[19] Paul A Yushkevich, Brian B Avants, John Pluta, David Minkoff, John A Detre, Murray Grossman, and James C Gee. Shape-based alignment of hippocampal subfields: evaluation in postmortem mri. *Med Image Comput Comput Assist Interv Int Conf Med Image Comput Comput Assist Interv*, 11(Pt 1):510–517, 2008. 4.2