

SVPASEG - Version 2.1 documentation

Jussi Tohka

October 18, 2010

Contents

| | | |
|----------|-------------------------------------|----------|
| 1 | INTRODUCTION | 2 |
| 1.1 | LICENSE | 3 |
| 1.2 | PROGRAMS IN THE BUNDLE | 3 |
| 2 | INSTALLING THE SOFTWARE | 4 |
| 2.1 | SYSTEM REQUIREMENTS | 4 |
| 2.2 | INSTALLATION | 4 |
| 3 | USAGE OF THE MAIN PROGRAMS | 5 |
| 3.1 | QUICK START GUIDE | 5 |
| 3.2 | GENERAL GUIDELINES | 5 |
| 3.3 | GAMIXTURE | 6 |
| 3.4 | SVPASEG | 8 |
| 4 | FILE FORMAT SPECIFICATIONS | 9 |
| 4.1 | ATLAS SPECIFICATION FILES | 9 |

| | | |
|-----|--------------------------------------|----|
| 4.2 | MIXTURE FILES | 17 |
| 5 | USEFUL SCRIPTS | 17 |
| 6 | READY MADE ATLAS SPECIFICATION FILES | 18 |
| 7 | ATLASES | 19 |

1 INTRODUCTION

SVPASEG is a collection of open-source software tools implementing a flexible voxel classification framework. It replaces the earlier MRFSEG+GAMIXTURE software bundle. The framework is based on a novel genetic algorithm based finite mixture model (GAMIXTURE) and a local 3-D Markov random field (MRF) segmentation algorithm (SVPASEG) based on the iterative conditional modes (ICM) algorithm. The key novelty in the SVPASEG algorithm is its ability to use different MRF models on different parts of the brain. Also, a configuration file is applied to control MRF parameters, atlases, allowed labels, and other constraints so the software is quite flexible for different purposes. The software package is to be extended with a novel inhomogeneous MRF algorithm allowing even more flexible spatial and intensity information modeling.

At the moment the software supports Analyze 7.5 and NIFTI1 images. The key references are:

- 1 J. Tohka , E. Krestyannikov, I.D. Dinov , A. MacKenzie-Graham, D.W. Shattuck , U. Ruotsalainen, and A.W. Toga. Genetic algorithms for finite mixture model based voxel classification in neuroimaging. IEEE Transactions on Medical Imaging , 26(5):696 - 711, 2007.
- 2 J. Tohka , I.D. Dinov, D.W. Shattuck, and A.W. Toga. Brain MRI Tissue Classification Based on Local Markov Random Fields, Magnetic Resonance Imaging 28(4): 557 - 573,2010.

If you use this software in your research, please cite both of the above papers.

1.1 LICENSE

See the source code files. Please note that no representations about the suitability of this software for any purpose are made. It is provided "as is" without express or implied warranty.

1.2 PROGRAMS IN THE BUNDLE

The main programs are:

- **SVPASEG** for brain image segmentation using local Markov random fields.
- **GAMIXTURE** for fitting FMMs to the image data using genetic algorithms.

The purpose is to 1) estimate the image parameters by gamixture, and 2) use these parameters to classify voxels in a 3-D image. There is a ready made shell script implementing this pipeline available (see Section 5 of this document). More detailed descriptions of the two main programs are provided in the sections 4.2 and 4.3.

The software package contains also smaller programs. The most important of these are:

- **erodemask**: Performs morphological erosion to a binary volume.
- **knnfilter**: Filters 3-D image with a KNN-filter
- **medfilt3D**: Performs 3-D median filtering of an input image
- **relabel_analyze**: Switches labels of an Analyze Image
- **showatlasspec**: Pretty prints the content of an atlas specification file, mainly for validation purposes. See the section about atlas specification files for further information.

2 INSTALLING THE SOFTWARE

2.1 SYSTEM REQUIREMENTS

The software has been developed using the Linux operating system. It has been tested on Linux kernels 2.4 and 2.6. However, the software should work on Unix systems as well.

The only requirements are the C++ compiler and makefile utility. The software has been tested using g++ (versions 3.3 and 4.1).

2.2 INSTALLATION

1. Untar the tar file containing the package to the directory to which you wish to base the installation using `tar xvf svpaseg.tar`.
2. Go to the `src` sub-directory by typing `cd src` and type `make`. In the case the standard C libraries are located in an extraordinary directory, you may have to edit relevant parts of the `Makefile`.
3. Next, you will need to obtain the atlas-files for SVPASEG. These provided in separate zip-packages, downloadable from the same place as the code. The easiest way to proceed is to place the zip-file into the sub-directory `atlases` under the `svpaseg` directory structure. This sub-directory has been created during the step one. Unzip the package and run the script `configure_atlases` under the main `svpaseg` directory. This script sets the directories in the provided atlas-specification files to match your settings.
- 3b Alternatively, if you wish to place atlases in some alternate directory, you may do this. However, then you should run `configure_atlases_custom` with the atlas directory as the only argument instead of the `configure_atlases`. For example, if the atlases are in `/share/atlases/` type

```
./configure_atlases_custom /share/atlases/
```

3 USAGE OF THE MAIN PROGRAMS

3.1 QUICK START GUIDE

For a quick start: use the scripts provided in svpaseg/scripts subdirectory. For SVPASEG-PVE (see the publication [2] above for different flavours of the SVPASEG) type:

```
/usr/svpaseg/script/svpaseg_classifier /data/myimage.nii \  
/data/my_mask.nii /usr/svpaseg/atlasspecs/svpaseg-pve.txt \  
/results/out_my_fmm_params.txt \  
/results/out_segmented_myimage.nii
```

You can also type

```
/usr/svpaseg/script/svpaseg_classifier /data/myimage.nii \  
default /usr/svpaseg/atlasspecs/svpaseg-pve.txt \  
/results/out_my_fmm_params.txt \  
/results/out_segmented_myimage.nii
```

if myimage.nii is skull-stripped and has zero values outside the brain.

Note that the file /results/out_my_fmm_params.txt is just for writing FMM-parameters file, you just need to provide a filename of the file where to write these, i.e. don't worry about crafting this file..

Important: All the algorithms using the atlases assume that the image to be segmented is in MNI-305 space (1mm x 1mm x 1mm, image dimensions 181 x 217 x 181).

3.2 GENERAL GUIDELINES

- The input images are expected to be in the Analyze 7.5 or NIFTI1 format. The type of the file in question is determined based on the extension of the file name, i.e., *.nii for NIFTI1 and *.img or *.hdr for Analyze. At the moment, the software supports writing the NIFTI1-files ONLY if the input

file is in NIFTI1 too. This is because there is much more information in the NIFTI1 header than in the Analyze header.

- The filenames should be given with FULL PATHNAMES. They can be given with `hdr` or `img` or `nii` extensions or in some cases without extensions. If they are given without extensions, the files are assumed to be Analyze 7.5 files. The latter choice is not reliable in all cases (particularly when path name contains the character `'.'`), but it is often the most convenient one.
- A command line help summary can be obtained by calling programs without arguments.
- Saying "default" instead of the brainmask filename causes programs to assume that voxels with zero intensities belong to the background.
- The Analyze 7.5 specifications mandate that images represented with more than 8 bits per voxel are stored in signed units. Giving `-unsigned` as the second argument will make the programs assume that the data in image volumes is unsigned.
- **Important:** All the atlas files are provided in the MNI-305 space and have the dimensions of 181 x 217 x 181 voxels and voxel size of 1 mm x 1 mm x 1mm. All the algorithms using the atlases assume that the image to be segmented is in register with these atlas files.
- There are probably other implicit limitations that I don't just now remember/know of.

3.3 GAMIXTURE

Genetic algorithm for mixture model optimization

The algorithm is described in:

J. Tohka , E. Krestyannikov, I.D. Dinov , A. MacKenzie-Graham, D.W. Shattuck , U. Ruotsalainen, and A.W. Toga. Genetic algorithms for finite mixture model based voxel classification in neuroimaging. IEEE Transactions on Medical Imaging , 26(5):696 - 711, 2007.

Usage: `gamixture [-unsigned] imagefile maskfile atlasfile
fmmparamfile [parameters]`

Specifying `-unsigned` right after `gamixture` makes the program assume that data is unsigned. The reason behind this switch is that unsigned integers are not supported by default by the Analyze 7.5 file format.

Parameters for the program:

- `imagefile` is the input image
- `maskfile` is the file specifying the brain mask. All voxels with the value greater than 0.5 (in the mask) are assumed to belong to the brain.
- `atlasfile` is the name of the atlas specification file (see the next section).
- `fmmparamfile` is the name of the file where the mixture parameters are to be written (see the next section).
- Optional parameters are :
 - alpha: parameter for blended crossover, defaults to 0.5
 - size: population size, defaults to 100
 - terminationthr: threshold for terminating the algorithm, defaults to 0.0005
 - xoverrate: crossover rate, defaults to 1
 - maxgenerations: maximum number of generations, defaults to 500
 - sortpop whether to use the permutation operator, defaults to 1
 - parzenn number of points for Parzen estimate (for approximate ML), defaults to 101
 - parzensigma window width parameter for the Parzen estimate, defaults to 1
 - equalvar whether component densities should have equal variances, defaults to 0
 - restarts the number of runs of the GA. The one with the highest likelihood score (or the lowest KL divergence) is selected as the result. Defaults to 10.

3.4 SVPASEG

Tissue classification based on local Markov Random Fields and the ICM algorithm.

Reference: J. Tohka, I.D. Dinov, D.W. Shattuck, and A.W. Toga. Brain MRI Tissue Classification Based on Local Markov Random Fields, Magnetic Resonance Imaging, 28(4): 557 - 573,2010.

Usage: `svpaseg img brainmask atlas_def mixture_params labeling`
[optional parameters]

- `img` is the name of the input image
- `brainmask` is the file containing the brain mask
- `atlas_def` is the name of the atlas specification file
- `mixture_params` is the file containing the FMM parameters (e.g. by GAMIXTURE). Note that the mixture parameter files must match the atlas specification file.
- `labeling` is the filename for the tissue classified image (only pure tissue types)
- Optional parameters are:
 - beta1: regularization parameter first first-order interactions, defaults depend on particular model setting. It is NOT recommended to change the defaults.
 - beta2: regularization parameter for second order interactions i.e. controls the strength of the spatial regularization component. defaults to 0.05. Sometimes it may be adequate to increase this value.
 - pveLabels: controls whether intermediate PVE classification labels are written, not relevant except for p-type atlases. Should be followed by the filename where the pve label image should be written.

4 FILE FORMAT SPECIFICATIONS

A reader who is not planning to edit any files and/or code, may wish to skip this section.

File format definitions: The programs in the bundle read and write several types of files in addition to the image files. More precisely, files can contain information about the Parzen estimates, FMM parameters and atlas specifications. All these files are text files. No comments are allowed in the files.

4.1 ATLAS SPECIFICATION FILES

We begin with the atlas specification files:

These files are used for inputting the information about SVPA, MRF and FMM constraints to the programs. There are two types of such files, those that impose constraints to certain mixture components and those that don't. The latter type exists for historical reasons and these are not supported any more.

The files of the former type - those that impose constraints to certain mixture components - should start with an identifier 'p', 'r', or 't'. These refer to various uses of svpaseg:

| id | meaning |
|----|---|
| p | svpaseg considers all labels (pure and mixed) as SVPASEG-PVE in the paper. Then, it reclassifies those voxels originally labeled as mixed voxels. |
| r | svpaseg considers only pure labels during the tissue classification. There is no equivalent in the paper. |
| t | svpaseg considers only pure labels during the tissue classification and uses tissue probability maps to aid the tissue classification. As SVPASEG-TPM in the paper. |

Thereafter, the number of atlas regions should be given (this should be 0 if there is no atlas), and thereafter the number of labels should be given. Then, the atlas regions should be defined (name, filename, constraints related to that region). The constraints indicate to GAMIXURE what kind of proportions are expected for each label. (If you don't want any constraints just give 0.0 for the lower limit and

1.0 for the upper limit for each label). Let α_i denote the proportion of the label i in a region. For example the following constraints,

```
0 0.1
0.2 1
0.2 1
0 0.05
0 0.3
0 0.3
```

mean that $\alpha_1 \in [0, 0.1]$, $\alpha_2 \in [0.2, 1]$, $\alpha_3 \in [0.2, 1]$, $\alpha_4 \in [0, 0.05]$, $\alpha_5 \in [0, 0.3]$, $\alpha_6 \in [0, 0.3]$.

Then, the definitions of labels should follow (name, whether partial volume label or pure tissue label, components of the label (give 0 0 if pure label). Note that these should be ordered so that pure labels come first and thereafter partial volume labels. Pure labels should be ordered according to their expected intensity values in the images. For example, the order should be CSF, GM, WM in the case of t1-weighted MR images.

Thereafter, the matrix specifying the 2nd order MRF should be given. It should be symmetric.

Note that the background is counted as a label, but the images are constrained to contain no background voxels in the brain region. Therefore, the background label should not be defined in the specification file. The background has always the label 0.

An example follows: (This is the file for MRFSEG-PVE in the paper; see explanation below)

```
jtohka@[10]% more mrfseg-pve.txt
p 0 7
0.0 0.1
0.0 1.0
0.0 1.0
0.0 0.05
0.0 0.3
```

```

0.0 0.3
csf 1 0 0
gm 1 0 0
wm 1 0 0
csfbg 0 1 0
csfgm 0 1 2
gmwm 0 2 3
0 0 1 1 0 1 1
0 -1 1 1 0 0 1
1 1 -1 1 1 0 0
1 1 1 -1 1 1 0
0 0 1 1 -1 1 1
1 0 0 1 1 -1 1
1 1 0 0 1 1 -1

```

End of the example

The first line

```
p 0 7
```

means that the FMMs are constrained, SVPASEG accepts PVE labels, and there are 7 labels (the background is automatically included and has always label zero.) The 0 after p means that this is the MRFSEG algorithm, so there are no distinct sub-domains of the image.

The lines

```

0.0 0.1
0.2 1.0
0.2 1.0
0.0 0.05
0.0 0.3
0.0 0.3

```

mean that there must be more than 0% and less than 10% of the label 1, more than 20% and less than 100% of the label 2 etc. in the volume The lower limit (i.e. the first number) should always be smaller than the upper limit.

The lines

```
csf 1 0 0
gm 1 0 0
wm 1 0 0
csfbg 0 1 0
csfgm 0 1 2
gmwm 0 2 3
```

mean that the label number 1 is called csf and it is a pure tissue label;
the label number 2 is called gm and it is a pure tissue label;
the label number 3 is called wm and it is a pure tissue label;
the label number 4 is called csfbg and it is a mixed tissue label consisting of csf (the label number 1) and the background (the label number zero);
the label number 5 is called csfgm and it is a mixed tissue label consisting of csf (the label number 1) and the gm (the label number 2);
the label number 6 is called gmwm and it is a mixed tissue label consisting of gm (the label number 2) and wm (the label number 3);

Important note: it is a good idea to keep the pure labels as first labels (i.e. 1, ... k) and thereafter number the pve labels. Also the order of the pure labels should be according to the order of the (expected) means within these labels in the image. For example, in T1-weighted MRI CSF should be label 1, GM should be label number 2, and WM should be label number 3.

Finally, the lines

```
0 0 1 1 0 1 1
0 -1 1 1 0 0 1
1 1 -1 1 1 0 0
1 1 1 -1 1 1 0
0 0 1 1 -1 1 1
1 0 0 1 1 -1 1
1 1 0 0 1 1 -1
```

specify the second order MRF based on pairwise interactions.

A second example:

```

jupeto@hopeatilhi atlasspecs $ more svpaseg-tpm.txt
t 12 7
centralr /share/sig_pic/jupeto/tmp/svpaseg/atlasses/central_right
0 0.1
0.2 1
0.2 1
0 0.05
0 0.3
0 0.3
centrall /share/sig_pic/jupeto/tmp/svpaseg/atlasses/central_left
0 0.1
0.2 1
0.2 1
0 0.05
0 0.2
0 0.2
cerebellumr /share/sig_pic/jupeto/tmp/svpaseg/atlasses/cerebellum_right
0 0.1
0 1
0 1
0 0.05
0 0.3
0 0.3
cerebelluml /share/sig_pic/jupeto/tmp/svpaseg/atlasses/cerebellum_left
0 0.1
0 1
0 1
0 0.05
0 0.3
0 0.3
frontalr /share/sig_pic/jupeto/tmp/svpaseg/atlasses/frontal_right
0 0.1
0 1
0 1
0 0.05
0 0.3
0 0.3
frontall /share/sig_pic/jupeto/tmp/svpaseg/atlasses/frontal_left

```

```

0 0.1
0 1
0 1
0 0.05
0 0.3
0 0.3
occipitalr /share/sig_pic/jupeto/tmp/svpaseg/atlasses/occipital_right
0 0.1
0 1
0 1
0 0.05
0 0.3
0 0.3
occipitall /share/sig_pic/jupeto/tmp/svpaseg/atlasses/occipital_left
0 0.1
0 1
0 1
0 0.05
0 0.3
0 0.3
parietalr /share/sig_pic/jupeto/tmp/svpaseg/atlasses/parietal_right
0 0.1
0 1
0 1
0 0.05
0 0.3
0 0.3
parietall /share/sig_pic/jupeto/tmp/svpaseg/atlasses/parietal_left
0 0.1
0 1
0 1
0 0.05
0 0.3
0 0.3
tempor /share/sig_pic/jupeto/tmp/svpaseg/atlasses/temporal_right
0 0.1
0 1
0 1

```

```

0 0.05
0 0.3
0 0.3
tempol /share/sig_pic/jupeto/tmp/svpaseg/atlasses/temporal_left
0 0.1
0 1
0 1
0 0.05
0 0.3
0 0.3
csf 1 0 0
/share/sig_pic/jupeto/tmp/svpaseg/atlasses/prob_csf_53
gm 1 0 0
/share/sig_pic/jupeto/tmp/svpaseg/atlasses/prob_gray_53
wm 1 0 0
/share/sig_pic/jupeto/tmp/svpaseg/atlasses/prob_white_53
csfbg 0 1 0
csfgm 0 1 2
gmwm 0 2 3
0 0 1 1 0 1 1
0 -1 1 1 0 0 1
1 1 -1 1 1 0 0
1 1 1 -1 1 1 0
0 0 1 1 -1 1 1
1 0 0 1 1 -1 1
1 1 0 0 1 1 -1

```

End of example 2.

The first line

```
t 12 7
```

means that FMMS are constrained, SVPASEG does not allow PV labels, SVPASEG uses TPMs, the brain-domain is divided into 12 sub-domains, and there are 7 labels altogether (background,CSF,GM,WM,background/CSF,CSF/GM, GM/WM).

The section starting with

```

centralr /share/sig_pic/jupeto/tmp/svpaseg/atlasses/central_right
0 0.1
0.2 1
0.2 1
0 0.05
0 0.3
0 0.3

```

and ending with

```

tempol /share/sig_pic/jupeto/tmp/svpaseg/atlasses/temporal_left
0 0.1
0 1
0 1
0 0.05
0 0.3
0 0.3

```

defines the sub-domains.

The part

```

csf 1 0 0
/share/sig_pic/jupeto/tmp/svpaseg/atlasses/prob_csf_53
gm 1 0 0
/share/sig_pic/jupeto/tmp/svpaseg/atlasses/prob_gray_53
wm 1 0 0
/share/sig_pic/jupeto/tmp/svpaseg/atlasses/prob_white_53
csfbg 0 1 0
csfgm 0 1 2
gmwm 0 2 3

```

defines the labels. The filename after each pure label defines the tissue probability map (TPM) for that label.

And - as before -

```

0 0 1 1 0 1 1

```



```

0 -1 1 1 0 0 1
1 1 -1 1 1 0 0
1 1 1 -1 1 1 0
0 0 1 1 -1 1 1
1 0 0 1 1 -1 1
1 1 0 0 1 1 -1

```

defines the MRF interaction matrix. Presently, 2nd order component (H -functions) of the MRF cannot vary from one sub-domain to another.

4.2 MIXTURE FILES

A reader who is not going to edit the code may wish to skip this sub-section.

Mixture files are plain text files listing the mixture parameter values. The mixture file always requires an atlas file to accompany it and the mixture file must match the atlas specification. Mixture file contains a single line consisting of FMM parameter values for each subdomain, the number of which is specified in the atlas specification file. When there are k types of pure tissue, and $kpve$ types of mixed tissues, the line looks like:

$$\mu_1 \quad \sigma_1^2 \quad p_1 \quad \mu_2 \quad \sigma_2^2 \quad p_2 \cdots \mu_k \quad \sigma_k^2 \quad p_k \quad p_{k+1} \cdots p_{k+kpve}$$

5 USEFUL SCRIPTS

At the time of writing there are few helpful scripts. These are located in the scripts subdirectory.

- `svpaseg_classifier` combines the GAMIXTURE with SVPASEG to tissue classify an MR image
- `svpaseg_classifier_s2` does the same but smooths intensity histogram (Parzen estimates) more. Good for lower resolution images.

- `svpaseg_classifier_equalvar` combines the GAMIXTURE with SVPASEG to tissue classify an MR image while assuming that the tissue class intensities have equal variances.
- `eae_classifier` implements the combination of programs for mouse brain image tissue classification. This requires also the program `relabel_analyze`.

6 READY MADE ATLAS SPECIFICATION FILES

There are a few ready made atlas specification files available in the `atlaspecs` sub-directory. The algorithm names refer to the paper to appear in *Magnetic Resonance Imaging* journal. The complete reference to this paper was provided in Section 1.

- `mousemri_segmentation.txt` : An atlas specification file for the tissue classification in mouse images (T2 weighted).
- `mrfseg-tpm.txt` : An atlas specification file implementing the MRFSEG-TPM algorithm for high resolution (i.e. approximately 1 mm^3 voxels) human T1 weighted brain MRI.
- `mrfseg-tpm_lowres.txt` : The same as above, but for poor quality, low resolution images (i.e. 3 mm^3 or lower). Note: it might be useful to resample the TPM files to reflect the low image resolution.
- `mrfseg-pve.txt` : An atlas specification file implementing the MRFSEG-PVE algorithm for high resolution (i.e. approximately 1 mm^3 voxels) human T1 weighted brain MRI.
- `svpaseg-tpm.txt` : An atlas specification file implementing the SVPASEG-TPM algorithm for high resolution (i.e. approximately 1 mm^3 voxels) human T1 weighted brain MRI.
- `svpaseg-tpm_lowres.txt` : The same as above, but for poor quality, low resolution images (i.e. 3 mm^3 or lower). Note: it might be useful to resample the TPM files to reflect the low image resolution.

- `svpaseg-pve.txt` : An atlas specification file implementing the SVPASEG-PVE algorithm for high resolution (i.e. approximately 1 mm^3 voxels) human T1 weighted brain MRI.
- `mrfseg-nopve.txt` : An atlas specification file implementing the MRFSEG algorithm considering only pure labels during tissue classification but not using an TPM. The algorithm has no equivalent in the paper, but it is the same as MRFSEG-TPM when TPMs are not used. For high resolution (i.e. approximately 1 mm^3 voxels) human T1 weighted brain MRI.
- `mrfseg-nopve_lowres.txt` : The same as above, but for poor quality, low resolution images (i.e. 3 mm^3 or lower). Note: it might be useful to resample the TPM files to reflect the low image resolution.
- `svpaseg-nopve.txt` : An atlas specification file implementing the SVPASEG algorithm considering only pure labels during tissue classification but not using an TPM. The algorithm has no equivalent in the paper, but it is the same as SVPASEG-TPM when TPMs are not used. For high resolution (i.e. approximately 1 mm^3 voxels) human T1 weighted brain MRI.
- `svpaseg-nopve_lowres.txt` : The same as above, but for poor quality, low resolution images (i.e. 3 mm^3 or lower). Note: it might be useful to resample the TPM files to reflect the low image resolution.

7 ATLASES

Two atlases are provided with the software:

- The sub volume probabilistic atlas derived atlas for breaking the image domain into sub-domains;
- The tissue probability maps (TPM), which some program versions use as prior information on voxel labeling in a similar way as tissue classification algorithms in SPM do.

The sub-volume probabilistic atlas derived atlas is based on ICBM Lobular Probabilistic Atlas and ICBM Deep Nuclei Probabilistic Atlas. See <http://www>.

loni.ucla.edu/Atlases/Atlas_Methods.jsp?atlas_id=7 for a complete description of these atlases. To create the atlas distributed with software, these atlases were combined and some regions were merged. Finally, a heat equation based technique was utilized to extend these gray matter atlases to provide sub-divisions of the white matter. For a detailed account of the approach, see reference [2] above.

The TPM atlas is derived based on the ICBM Probabilistic Tissue Atlas, see http://www.loni.ucla.edu/Atlases/Atlas_Methods.jsp?atlas_id=7. The TPM atlas provided with the package contains tissue classifications of the 53 subjects applied to construct the sub-volume probabilistic atlas. These TPMs can (and probably should) be replaced by the widely used ICBM152 tissue probability maps; the 53 subjects in the TPM distributed with this software is probably a subset of the 152 subjects used for ICBM152.