LONI Brain Parser Documentation

1. Overview
The LONI Brain Parser software is a learning-based algorithm that efficiently performs whole brain image segmentation to parse an input MRI image into 56 anatomical structures of interest. In training, it automatically fuses a rich set of image features and context information from a large candidate pool to perform classification and segmentation. You can also use it to train to segment a particular set of structures of interest if a set of training images with the corresponding ground truth labels are provided.

Brain Parser specifically includes skull and scalp removal, image non-uniformity compensation, voxel-based tissue classification, topological correction, rendering, and editing functions. The collection of tools is designed to require minimal user interaction to produce cortical representations.

a. Comparison chart (This would be an image with a description and/or caption) I don’t have it now though so you can leave it off.

2. Usage
a. Assumptions
A set (minimally 10) of T1 MRI data along with delineated ROIs is required to train the algorithm. Correspondence between the delineations and MR is a requirement. The files must be in Analyze Image format.

b. Limitations
If you are trying to use the trained model to perform automatic segmentation for the brain, then a basic requirement is that the input MRI should globally have the same orientation as the template image, 0000.brain.img. The analyze image is in 16 bit short format and you are expected not to have too extreme intensity range of the image. The brain parser algorithm will perform automatic intensity normalization. However, the similar the intensity patterns to the template image, the better result you will have.

c. Requirements
   i. System Requirements
Operating System: Windows 2000(SP4)/XP(SP2)/Vista(SP1) or Linux kernel 2.6.x.
CPU: INTEL® Core™ 2 compatible processor above 2.8GHz.
Memory: At least 2 GB of RAM.
Hard Disk: 150MB for program and the model trained.
  • In training, a temporary free space which is five times the size of training data is required.
  • In testing, the processed result requires the same size of space as testing data.
Internet Connection: Optional. To access LONI Pipeline service remotely, a connection above 50Mbps is suggested.

ii. Preprocessing Requirements

Several types of preprocessing steps, depending on the difference between the original 3D input image and the template, are necessary before the LONI Brain Parser binary can be run. In the following paragraph, ‘Input’ stands for the preprocessed 3D input image for Brain Parser instead of the original 3D input image. These possible preprocessing steps include:

1. Data validation: Input should only have one brain object inside. Non-brain objects or unrecognizable brain data should be avoided. The 3D image file format is Analyze (pair of *.img and *.hdr) with 16-bit short data type.

2. Reorientation: Input should have the same orientation as the template. If the original 3D input image has a different orientation from the template, reorient it first.

3. Outlier strip: Input should exclude any parts other than the brain, especially the skull. For example, Fig 1.a is not ready for Brain Parser processing, which need to be outlier stripped as Fig 1.b.

![Fig. 1.a Before stripping](image1)

![Fig. 1.b After stripping](image2)
4. **Resize**: Input should have the same image size in all the three dimensions as the template.

5. **Alignment**: An alignment between the brain regions in the input and the template images, at least an affine transform, is required for correct. Fig 2.b and 2.c shows two examples of both resizing and alignment of the input image in Fig. 1.

![Fig. 2.a Template image](image1)

Fig. 2.a Template image

![Fig. 2.b Aligned by affine transform](image2)

Fig. 2.b Aligned by affine transform

![Fig. 2.c Aligned by deformable model](image3)

Fig. 2.c Aligned by deformable model

### iii. Postprocessing Requirements

1. **Inverse operations**: If any operation of type 2, 4, or 5 in the previous section is applied to the input, its corresponding inverse operation is required for a correct mapping between the parsed result and the original 3D input image.
Conversely, you can download the *LONI Pipeline/LONI Brain Parser workflow* which has been setup with all the additional processing steps needed to produce results right away. This workflow was produced to reduce the demands on researchers to haggle with various external steps to run this program.

**d. Installation**
Simply download the precompiled binaries, currently it is available for Linux.

**e. Tutorial/Quick Start Guide/Link to howto website**
http://www.loni.ucla.edu/twiki/bin/view/CCB/CCBPrincipalsMeetings_IndentividResearcherProj_ZhuowenTu

**f. Applications**
To train, you need to organize your data (in analyze image format) as the following:

i. Name all the MRI analyze images as “yourname.brain.hdr” and “yourname.brain.img” with the corresponding manual annotations as “yourname.label.hdr” and “yourname.label.img”.

ii. Choose the best image in your training set and name it as “0000.brain.hdr” and “0000.brain.img” with the corresponding manual annotations as “0000.label.hdr” and “0000.label.img”. This will be your template image.

iii. Copy all the images and their annotations into a directory.

iv. Create a directory for saving the learned models.

v. run:

```
brain_parser_v2.0 -r 1 "your data directory" "your output directory"
-n "number of structures"
```

• “-r 1” refers to training
• “-n” gives the number of anatomical structures you want the algorithm learn to segment

vi. After the training is finished (ranging from several hours to a few days), Create a model directory with a subdirectory named “appearance”. The resulting naming convention is “your model directory/appearance”

vii. Copy files: 0000.brain.hdr, 0000.brain.img, 0000.label.hdr, 0000.label.img, setting.txt, atlas.3d, known.3d, label_mapping.txt, mri_intensity_common.3d, and mri_intensity_template.3dm to “your model directory/appearance”

viii. Copy clf files from training result directories /1, /2, and /3.

   Rename the clf files start from /1 as clf, clf1, clf2. Copy them to “your model directory/appearance”.

Taking Windows for example, the resulting structure of the model directory should look like Fig. 3:
To Test, simply assign the model directory to “your model directory”, not “your model directory/appearance” though.

i. `brain_parser_v2.0 -r 0 "your_input.img" "your_label.img" -m "your model directory" -s "smoothing number"
   - “your_input.img” is the input image which satisfies the preprocessing requirements.
   - “your_label.img” is the output labeling image having the same image size as “your_input.img”.
   - “-r 0” refers to testing.
   - “-m” refers to your model directory.
   - “-s” gives a positive real number which is the smoothing factor. The larger the number, the smoother the resulting parsed boundaries. The typical range is 0.0~2.0 and default = 1.0.

ii. your_label.img can be checked by overlapping it with your_input.img as Fig. 4.a to Fig. 4.c. Refer to Preprocessing/Postprocessing Requirements if you want to have a label image corresponding to the coordinate of the original input image as Fig. 4.d.
Fig. 4.a Input for Brain Parser

Fig. 4.b Output label image

Fig. 4.c Overlap of 4.a and 4.b

Fig. 4.d Overlap of the original input and the postprocessed label image.