

# A User's Guide to Graphical-Model-based Multivariate Analysis

Rong Chen  
December 2006

## 1. Introduction

Graphical-Model-based Multivariate Analysis (GAMMA) is a Bayesian data mining software for structural and functional magnetic-resonance data. GAMMA can be applied to the cross-sectional or longitudinal study of morphometric difference in structure MR data, or the between subject analysis in fMR study. From the machine learning point of view, GAMMA can be used for either unsupervised or supervised learning.

The input of GAMMA is a dataset  $D$  which has  $d$  observed variables  $V_i$  and a function variable  $C$ .  $V_i$  is a voxel in a MR image.  $C$  is a *function variable* which can either be a demographic variable such as age, or a clinical variable reflecting performance on a neuropsychological battery of tests. The output of GAMMA is a model  $M$  containing a label field and a Bayesian network. GAMMA uses a Bayesian network to represent the associations among voxels and the function variable.

GAMMA uses a contextual-clustering method based on a Markov random field (MRF) to find regions in which all voxels have similar associations with the function variable. Loopy belief propagation is used to infer the unobserved label field and belief map. GAMMA uses an ensemble learning method to generate a stabilized model for small sample size data.

The features of GAMMA are:

- It is a fully automatic non-parametric algorithm with high sensitivity and specificity.
- Handle the tasks of both clustering and classification.
- Find probabilistic associations among the brain regions and the function variable.
- Provide ensemble learning methods to generate stabilized model.

Generally, GAMMA includes two stages: data preprocessing and GAMMA analysis.

- ***A longitudinal study of brain morphometry.*** The task is to find the brain regions that can characterize group differences. In this case,  $C$  is a subject's group membership. For each subject, we obtained structural MR images for two different times,  $t_1$  and  $t_2$ , along with measurements of  $C$ . For a pair of subject's MR images, after preprocessing steps such as segmentation, mass-preserving registration, and smoothing, we obtain a map in which voxels' intensities represent the volume of a region. After discretization, this map is referred to as the *difference map*. The input of GAMMA is a set of difference maps and the associated function variables. The whole process is depicted in Figure 1.
- ***Detecting group difference of brain activation recorded by fMRI.*** The task is to identify brain regions characterizing group differences. Data was preprocessed using SPM2 (<http://www.fil.ion.ucl.ac.uk/spm/spm2.html>) or Voxbo

(<http://www.voxbo.org>). After creating a statistical parametric map ( $T$ -map or  $F$ -map) for each effect. We chose a significant level  $\alpha$  to threshold the resulting statistical parametric map, to generate a binary map representing voxel activation; each voxel assumed a value in  $\{0, 1\}$ , corresponding to off (no activation) and on (activation), respectively. We provided binary *difference maps* and  $C$  as input for GAMMA.

GAMMA can use the generated model  $M$  for classification.

GAMMA is a freeware under GPL. Source code is available. Please send us your comments and bug report to [rong.chen@uphs.upenn.edu](mailto:rong.chen@uphs.upenn.edu) or [r\\_chen\\_mail@yahoo.com](mailto:r_chen_mail@yahoo.com).

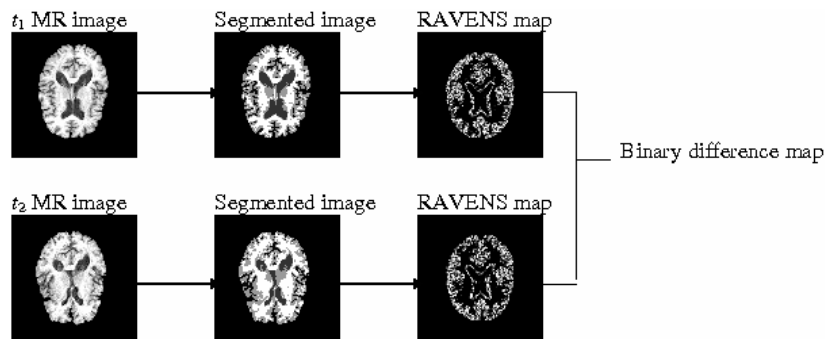


Figure 1. Longitudinal study for brain morphometry.

## 2. Installation

The core of GAMMA is written in C++. A GUI in Matlab is provided. It needs Matlab's image processing and statistics toolbox. It also uses the MRI toolbox version 2.0 which can be downloaded from <http://sourceforge.net/projects/eeg/>. Matlab function `conf2ind.m` is from Kevin Murphy's BN toolbox. `Jackrsp.m` is from Abdelhak M. Zoubir and D. Robert Iskander's bootstrap toolbox.

Since the source C++ code is available, users can re-compile it in their platform which could be Unix, Linux, or Windows. However, GAMMA GUI is specifically designed for Unix/Linux. If users use Windows, they may have to install the cygwin.

Here is the installation step under Unix/Linux.

- 1) Unzip the `gamma.tar.gz` to a directory. Assume the directory is `'~/soft/'`.
- 2) Go to directory `'~/soft/gamma1.0'`; then go to directory `'ccode'`.
- 3) Run `install_gamma`. It will compile the C++ code and generate executable programs. These generated programs are in directory `'~/soft/gamma1.0/bin'`.
- 4) Add the directory containing the executable programs to Unix/Linux search path. For example, if these programs are in `'~/soft/gamma1.0/bin'`, then the commands are

- *bash users*: `export GAMMADIR=~/soft/gamma1.0/bin; export PATH=$PATH:$GAMMADIR/bin;`
- *tcsh users*: `setenv GAMMADIR ~/soft/gamma1.0/bin; setenv PATH ${PATH}:$GAMMADIR/bin; rehash;`

- 5) Edit the GAMMA\_HOME in file start\_gamma.m (in folder ~/soft/gamma1.0/) and change it to be the directory where GAMMA is installed.
- 6) Run start\_gamma.m

### 3. GAMMA GUI

- 1) Create a directory and copy all difference maps of a study to this directory.
- 2) Go to that directory, run gamma\_gui.m
- 3) If this is the first time you work on this dataset, choose 'create a new project'. Otherwise, go to step 9.
- 4) If you choose creating a new project, a dialog window will appear (Figure 2). Set the project home directory to be the directory that contains the dataset. Then set project name.



Figure 2. Create a project – step 1

- 5) Press button 'apply', then press 'close'.
- 6) In dialog 'create a project – step 2' (Figure 3), create or load an image file list. An example file is here

```

6                               -- total number of subjects
1 sub1                          -- subject id, file name (sub1.img)
2 sub10
3 sub100
4 sub11
5 sub12
6 sub13

```

GAMMA can handle analyze format files. Each subject should have .hdr and .img files.

- 7) In dialog 'create a project – step 2' (Figure 3), input or load a function variable. If you input it, just input 1 0 1 0 .... If you load it from a file, the file should be a text file in which function variable values are separated by space.
- 8) Set the image size and parameters. In general, you can use the default values of parameters. Then press 'OK'.

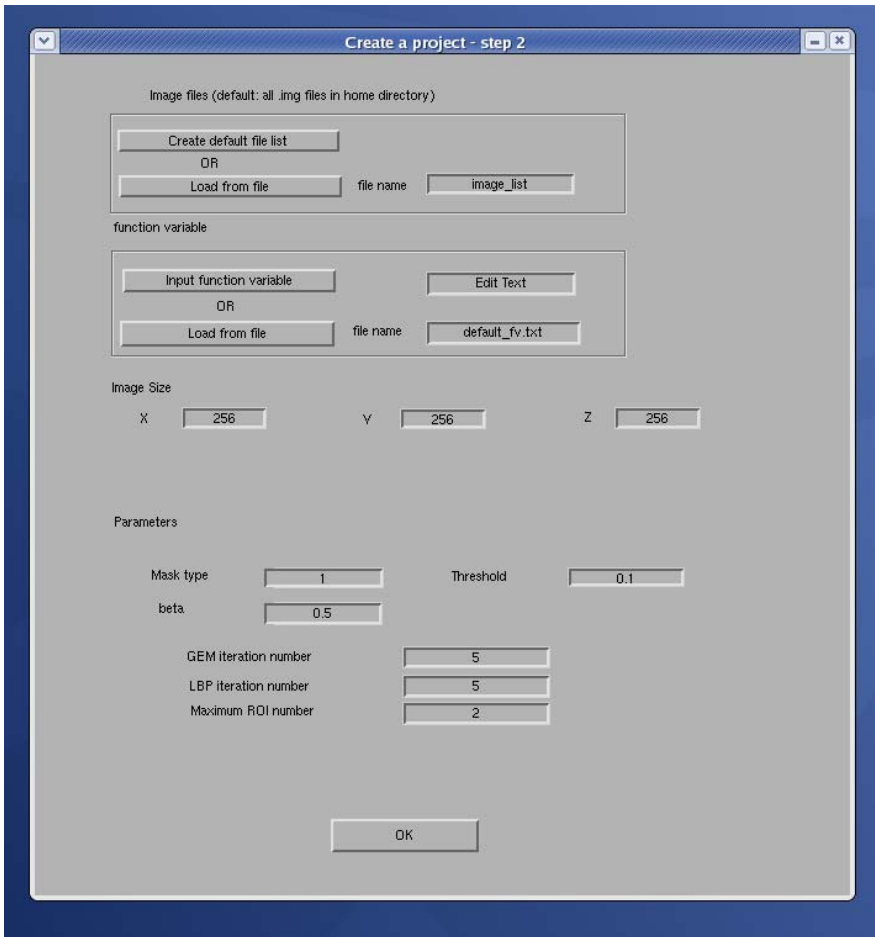


Figure 3. Create a project – step 2.

9) Press 'run a gamma project'. A GUI as Figure 4 will appear.

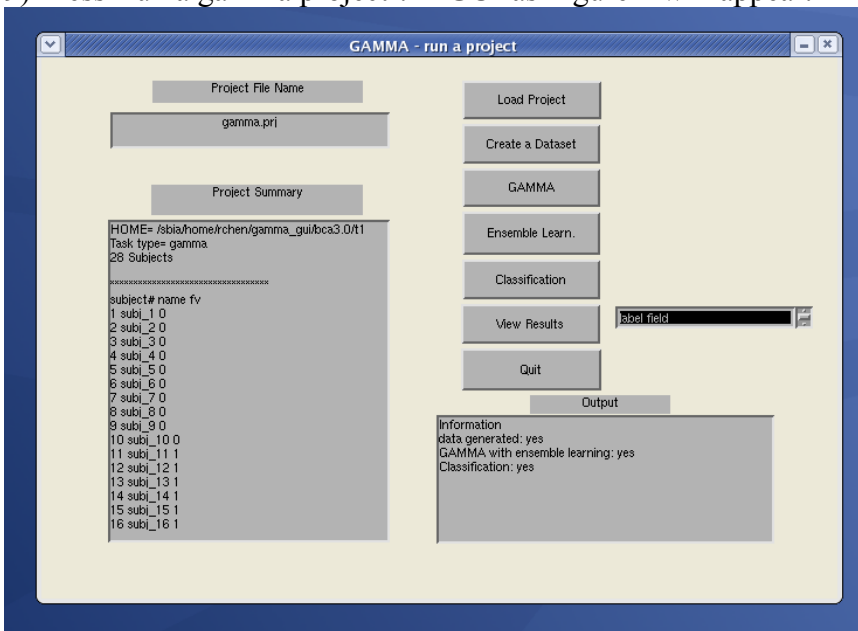


Figure 4. Run a GAMMA project

- 10) Press 'load'. The project's details are in 'project summary'.
- 11) Press 'create a dataset'; then press 'GAMMA' to analyze the data.
- 12) Press 'Ensemble Learn.' to do the ensemble learning.
- 13) Press 'Classification' if you want to use the generated model as a classifier.
- 14) In 'view results', you can view the results. There are three options you can choose: 'label field', 'representative variable', and 'Bayesian network'.
- 15) Press 'Quit' to exit.

#### 4. Example study – automatic atrophy analysis

A sample dataset for is provided in folder test\_data. In this dataset, a voxel in the difference map represents whether or not a region is atrophic. We indicate by  $[A = 1; B = 1; F = 1]$  that the region A was atrophic, the region B was atrophic, and there was a functional deficit. These data consist of ten  $[A = 0; B = 0; F = 0]$ , ten  $[A = 1; B = 0; F = 1]$ , and eight  $[A = 0; B = 1; F = 1]$  samples. Data are noisy.

GAMMA correctly detects two ROIs; and models the associations among regions A and B voxels and F using a Bayesian network. The ROIs is in Figure 5; the representative voxels is in Figure 6; and the learned Bayesian network is in Figure 7. The ROC curves are in Figure 8.

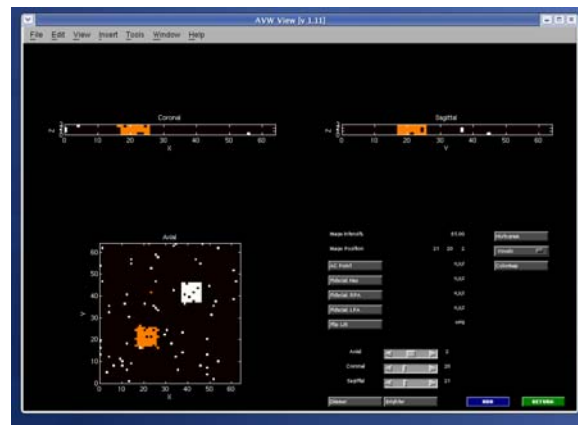


Figure 5, ROI for simulated data.

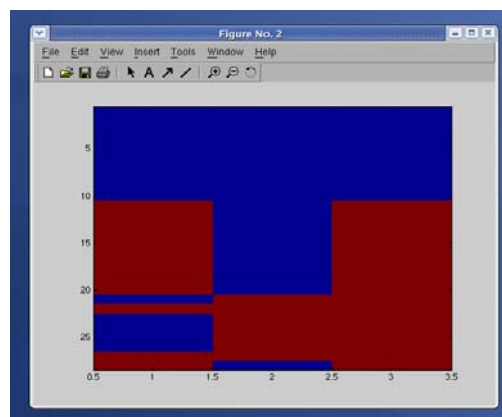


Figure 6, representative voxel for simulated data.

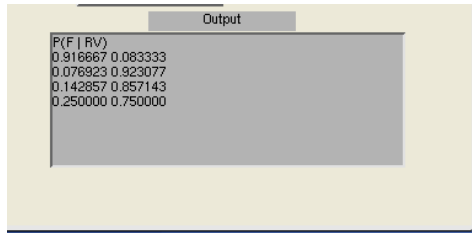


Figure 7, BN for simulated data.

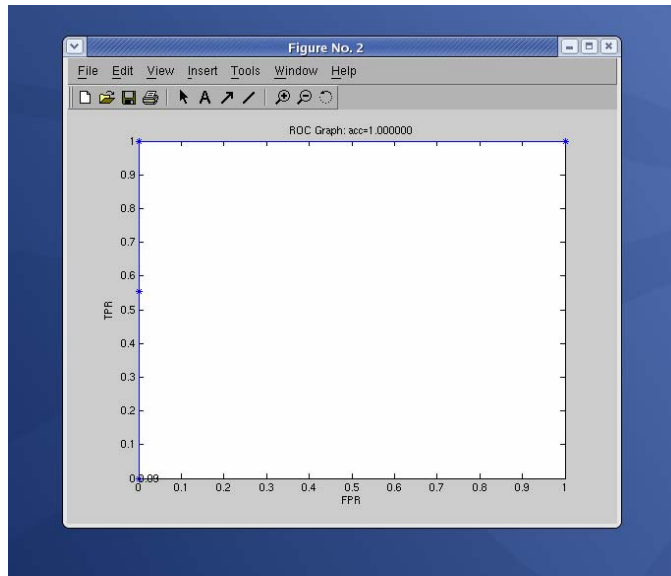


Figure 8, ROC curves for simulated data.