Published in final edited form as:

Neuroinformatics. 2012 April ; 10(2): 119–127. doi:10.1007/s12021-011-9129-7.

Graphical model based multivariate analysis (GAMMA): an open-source, cross-platform neuroimaging data analysis software package

Rong Chen¹ and Edward H Herskovits¹

¹Department of Radiology, University of Pennsylvania, Philadelphia, PA 19104, USA

Abstract

The GAMMA suite is an open-source, cross-platform data-mining software package designed to analyze neuroimaging data. Analyzing brain image volumes is a very challenging problem, due to undersampling and the potential for multivariate nonlinear interactions among variables. The GAMMA suite provides a set of tools to facilitate the analysis of neuroimaging data.

1. Introduction

Neuroimaging-based studies often focus on elucidating associations among brain structure (e.g., volumetric) and function (e.g., functional magnetic resonance) measurements, and clinical measurements, such as a group-membership variable that represents whether or not a subject has Alzheimer's disease. We refer to such associations as structure-function associations. We denote by C the clinical measurement of interest, which we also call the function variable. One of the central tasks in brain research is to detect brain regions that characterize a cognitive or disease process, and use these brain regions as biomarkers to build diagnostic or prognostic models. For example, many studies aim to detect brain regions demonstrating morphometric (size, volume, and shape) abnormalities in patients with Alzheimer's disease, and use these morphometric variables to distinguish normal elderly from patients with Alzheimer's disease.

Clinical and research neuroscientists have developed a broad array of neuroimaging techniques to measure brain structure and function. Typically, image data acquired using these neuroimaging techniques include millions of voxel variables, in addition to clinical data. Since these experiments involve, at most, hundreds of subjects, the data are severely undersampled. The analysis of these data has presented a challenging problem to the neuroscience and neuroinformatics research communities.

Due to the large number of voxel variables in brain images, the dominant approach to detecting structure-function association is mass-univariate. The major limitations of this approach are its focus on functional specialization, and its inability to model multivariate interactions among brain regions. Functional specialization and integration are two fundamental principles of brain functional organization (Friston, 2005). Functional specialization posits that a brain structure is specialized for some aspect of cognitive process, whereas functional integration emphasizes interactions among brain regions. In many neuroscience applications, functional integration may be more important. For

example, it has been suggested that the core feature of schizophrenia is disruption of prefrontotemporal integration (Friston and Frith, 1995). In general, functional integration is more difficult to assess than functional specialization, in that the mass-univariate approach cannot detect functional-integration patterns (Friston, 2005).

To address these problems, we developed a set of algorithms that model interactions among brain regions and a clinical variable. These algorithms include:

- Graphical model based multivariate analysis (GAMMA) (Chen and Herskovits, 2005b). GAMMA is a nonparametric, multivariate, voxel-based, Bayesian approach to modeling structure-function associations. We designed GAMMA to remedy two of the principal limitations of general linear model based mass-univariate methods. First, GAMMA is multivariate, rather than mass-univariate, and therefore has the potential to detect multivariate associations among brain regions and the function variable C . Second, GAMMA is nonparametric: it does not rely on statistical assumptions about the data, such as normality. These two features allow GAMMA to detect complex nonlinear multivariate associations among image features and a clinical variable. GAMMA has been applied to the analysis of structural MR images, functional MR (fMR) data, and lesion-deficit data (Chen and Herskovits, 2005b, Chen and Herskovits, 2007b, Chen et al., 2008, Chen and Herskovits, 2007a).

The goal of GAMMA is to detect a set of brain regions that are jointly predictive of the clinical variable C . Toward this end, GAMMA iteratively 1) identifies voxels characterizing group differences, and 2) groups voxels into regions. We refer to these two steps as Markov-blanket identification and voxel-space partitioning, respectively (Chen and Herskovits, 2005b). In Markov-blanket identification, GAMMA detects a set of voxels (which we call representative voxels) that are jointly predictive of C . The associations among representative voxels and C are described by a Bayesian-network (BN) (Pearl, 1988) model with inverse-tree structure. In voxel-space partitioning, for each representative voxel, GAMMA identifies voxels that are probabilistically equivalent to a particular representative voxel. In this manner, each representative voxel is associated with a region of interest (ROI), and voxels in the same ROI are probabilistically equivalent to that ROI's representative voxel. A label field stores the ROI definition. The output of GAMMA is a model consisting of a Bayesian network and a label field, $M = \{B, \Lambda\}$.

- GAMMA with ensemble learning (GAMMA-EL) (Chen and Herskovits, 2007b). GAMMA-EL augments GAMMA with functions to generate stable models. As we noted, many neuroimaging experiments are undersampled; in addition, these data may be noisy. In this case, it is important that the model be stable under small perturbations of the data. We use ensemble learning (Duda et al., 2001) to stabilize the model-generation process. The rationale behind ensemble learning is to reduce variability in model generation, via averaging. In GAMMA-EL, we first resample the data set and obtain a set of perturbed data sets. For each perturbed data set, we generate a model using GAMMA. This set of models

constitutes a model ensemble. We selectively combine these models to generate a stable model.

- **Regional state inference (RSI)** (Chen and Herskovits, 2005a). This algorithm is designed to use biomarkers generated by GAMMA for prediction. Either GAMMA or GAMMA-EL can be used to generate a set of markers (the ROIs in the label field). Let R denote the region state (effect/no effect) for a particular ROI. RSI will infer R using a latent-variable model, with an online Gibbs sampling algorithm (Chen and Herskovits, 2005a). After obtaining $\mathbf{R}=\{R_i\}$, we can use different classification methods, such as support vector machines, decision trees, or BN classifiers, to train a classifier that classifies C based on $\mathbf{R}$.

Although Bayesian methods hold great promise in the analysis of data acquired from a broad array of neuroimaging experiments, this approach has been under-utilized, primarily due to accessibility: these algorithms are complicated. For a typical neuroscience investigator, who may not have expertise in probabilistic graphical models, implementing these algorithms could be difficult. To address this problem, we have developed an open-source, cross-platform, and thoroughly documented software package that includes these Bayesian multivariate analysis tools, with enhanced usability, interoperability, and accessibility. We call this software package the GAMMA suite.

2. Design and Implementation

2.1 Overall design

The GAMMA suite is an open-source, cross-platform software package. The architecture of the GAMMA suite is depicted in Figure 1.

We developed the GAMMA suite based on the following principles:

- *Freely available source code.* We have released the source code of these Bayesian data-mining algorithms under the GNU General Public License (<http://www.gnu.org/copyleft/gpl.html>). The implementations of these algorithms do not depend on any proprietary third-party library.
- *Cross-platform.* The GAMMA suite runs under Windows (XP, Vista, and Windows 7), Mac OS X (versions 10.5 and above), and Linux. The GUI is uniform across platforms, allowing collaboration among researchers using a variety of computing environments.
- *Balance of generalizability and specificity.* The GAMMA suite is not a general-purpose data-mining software package; its main purpose is to provide a set of advanced Bayesian data-mining algorithms for the analysis of brain-image data. However, we do expect that some modules will be reused by other investigators to solve problems outside this domain. For example, the GAMMA suite includes BN representation and data-mining functions that have wide applicability. To facilitate reuse of these modules, we have provided an application programming interface (API) for some statistical models and data-mining algorithms.

The GAMMA suite includes two end-user programs: GAMMA core and the graphical user interface (GUI). We used C/C++ to implement both GAMMA core and the GUI. GAMMA core includes command-line programs, and GAMMA GUI is a Qt-based (<http://trolltech.com/>) graphical user interface that calls core functions. The primary tool for clinical investigators is GAMMA GUI. GAMMA core requires that a user have experience in using command-line programs. We provide GAMMA core for users who want to call algorithms in the GAMMA suite from a script or via a system call.

The input data $\mathbf{D}$ to the GAMMA suite consists of a collection of image volumes and the associated function variable C : $\mathbf{D} = \{(\mathbf{V}^1, C^1), \dots, (\mathbf{V}^k, C^k)\}$. Let V_i^k represent the i^{th} voxel for sample k . V_i^k is a binary variable representing effect/no effect (or abnormal/normal). For example, in lesion-deficit studies, we label $V_i^k = 1$ to represent a lesioned voxel, and $V_i^k = 0$ to represent a normal voxel. Let $\mathbf{V}^k$ denote the collection of V_i^k for subject k . $\mathbf{V}^k$ is provided as an image, so the collection $\{\mathbf{V}^k\}$ must be defined in the same stereotaxic space across subjects. We refer to $\mathbf{V}^k$ as an effect map. Appendix 1 describes how to generate binary effect maps for a morphometric study, functional MR-based study, and lesion-deficit study.

To facilitate interoperability with other data analysis tools, the GAMMA suite adopts widely used data-exchange formats. For example, it includes support for Analyze 7.5 and NIfTI image formats, which are widely used in the neuroimaging community. The GAMMA suite imports and exports non-image variables as comma-separated-values (CSV) files. In addition, It writes the BNs that it generates in xdsl format, which can be used by Genie (<http://genie.sis.pitt.edu/>). GAMMA GUI is based on Qt4, which is a freely available, widely used toolkit for developing cross-platform GUI applications. GAMMA GUI includes three modules: reporting, project management, and visualization. The project-management module is the main user interface. It allows the user to create a new neuroimaging study, specify the data sources, load an existing project, run or re-run a specific step in the analysis, display an analysis report, and export the report and the generated model. GAMMA GUI's reporting module presents to the user the models generated during analysis. The visualization module displays the label field in a simple slice-by-slice fashion.

2.2 Data-mining algorithms in the GAMMA suite

Figure 2 shows the implementations of GAMMA and GAMMA-EL. The major difference between GAMMA and GAMMA-EL is that GAMMA-EL uses resampling methods to stabilize the model-generation process. However, the resampling process typically consumes significant computational resources, as GAMMA-EL calls GAMMA many times.

The input to GAMMA is a collection of image volumes and the associated function variable C . In Figure 2, modules for BN generation and equivalence-map generation correspond to Markov-blanket identification and voxel-space partitioning, respectively.

During BN generation, GAMMA identifies a set of representative variables $\mathbf{RV}^* = \{RV_1, RV_2, \dots, RV_k\}$ such that

$$RV^* = \arg \max_{RV: pa(C) = RV} BDe(S, D), \quad (1)$$

where S is the structure of a BN, and the parent set of C in this BN (i.e., $pa(C)$) is $\mathbf{RV}$, BDe is the Bayesian Dirichlet score (BDe) score (Cooper and Herskovits, 1992).

The equivalence-set generation module seeks variables that are probabilistically equivalent to the representative variable RV . We define probabilistic equivalence between voxels V and RV as similarity, with respect to a given metric, of the conditional probability distributions $\Pr(V|C)$ and $\Pr(RV|C)$. Let $\mathbf{A}$ denote the set of candidate variables that could potentially be probabilistically equivalent to RV . We use a clustering algorithm to group $\mathbf{A}$ into clusters. We label those voxels in the cluster that has the highest mean similarity metric, as being probabilistically equivalent to RV . That is, each RV is associated with a ROI.

Equivalence-set generation is achieved by belief-map generation. In belief-map generation, each variable V in $\mathbf{A}$ is associated with an unobserved variable L , which represents cluster membership for this variable. Let $\mathbf{L} = \{L_i\}$. We use a Markov random field, a statistical image model, to model the interactions among $\mathbf{L}$. A Markov random field is used to group together voxels that have similar probabilistic associations and spatial proximity to each other. Regarding delineating brain regions characterizing group difference, we demonstrated that GAMMA superior to Bayesian morphometry analysis, which does not use any spatial model, in both sensitivity and specificity (Chen and Herskovits, 2005b). The spatial model used in GAMMA can partially address the problem of image misalignment and noisy data. Belief-map generation employs loopy belief propagation (Murphy et al., 1999) to infer the mode of $\mathbf{L}$.

The output of GAMMA is a model consisting of a BN and a label field: $M = \{B, \Lambda\}$. B is a BN that represents interactions among the representative voxels and C ; the label field Λ stores voxelwise ROI memberships.

For GAMMA-EL, in the data-resampling step, we use the jackknife or bootstrap method to generate a subset of samples $\mathbf{D}^m$ which is the perturbed version of $\mathbf{D}$. For each $\mathbf{D}^m$, we generate a model $M^m = (B^m, \Lambda^m)$; thus, M^m constitutes a model ensemble. The model-fusion module aggregates these models to construct an ensemble by taking the mode of $\Pr(B^m)$, and generating aggregate label fields corresponding to this mode.

In regional state inference, our goal is to infer the regional state, R , which represents the state of a ROI consisting of voxel variables $\mathbf{V} = \{V_1, \dots, V_n\}$, where $\mathbf{V}$ are in the same ROI. We use a BN to model associations among R and $\mathbf{V}$. In the BN, R is the parent node of $\mathbf{V}$. We have

$$\Pr(R, \mathbf{V}) = \Pr(R)\Pr(\mathbf{V} | R) \quad (2)$$

The maximum likelihood estimation of R is obtained by maximizing $\Pr(R, \mathbf{V})$. Note that if R is observed, $\Pr(R, \mathbf{V})$ is the BDe score. We use Gibbs sampling to estimate R ; however, as conventionally implemented, Gibbs sampling for this problem is computationally expensive. We have developed a specialized version of Gibbs sampling that incrementally updated the BDe score (Chen and Herskovits, 2005a), greatly reducing computation time for estimating R .

2.3 GAMMA GUI processing pipeline

Figure 3 shows a schematic representation of the processing pipeline in GAMMA GUI. For a neuroimaging study, investigators first prepare the effect maps and the clinical data file. To create an effect map, an investigator creates an image file list (a plain text file), which includes subject ID and the corresponding image file name. The clinical data file is in CSV format, where the first column is the subject ID and the last column is the function variable C .

On startup, the user imports the image file list and the clinical data file into GAMMA GUI. Then the user sets algorithm parameters. GAMMA GUI then creates a project file from this information, allowing researchers to re-analyze data with minimal or no additional effort.

When the GAMMA project file is ready, the user can perform GAMMA or GAMMA-EL analysis. Either application will yield a model consisting of a BN and a label field. The BN represents the interactions among representative voxels and C ; this BN can be used to understand the functional integration relevant to the disorder or cognitive process being examined. GAMMA GUI saves this BN in a plain text format, and in the standard XDSL format. The user can work with this BN model inside GAMMA GUI, or load the XDSL file into GeNie for BN visualization and evaluation. The label field stores voxelwise ROI definitions; the user can view the label field using GAMMA GUI, MRICro (<http://cnl.web.arizona.edu/mricro.htm>), FSL (Smith et al., 2004), AFNI (<http://afni.nimh.nih.gov/afni>), or SPM (SPM, 2010).

Having completed a GAMMA analysis and obtained a model, the user can perform RSI analysis to obtain regional states for prediction. The input to RSI is the label field or an external ROI file, such as a manually labeled ROI. If the user uses an external ROI file, it should be in plain text format. The output is a CSV file in which each column represents a ROI and the last column is C . The user can load this file into Weka or other data mining software to build a predictive model.

2.4 Re-useable data-mining modules

Three key algorithms in the GAMMA suite: GAMMA, GAMMA-EL, and RSI, are built based on a set of data-mining and data-management modules. These modules can be re-used by researchers to build their own software.

- BN creation and data mining. This module includes: 1) BN representation; 2) computing BN parameters using maximum-likelihood or maximum a posteriori methods; 3) generating BN structures from data, based on the BDe score; 4) a fast updating algorithm for computing the BDe score.
- Image- and clinical-data management. We developed two classes for data management: one for image data and one for clinical data. For image data, we provide functions to read a set of images (a data set), read or set a single subject, read or set a single voxel, and extract a subset, in Analyze or NIfTI format. We provide a similar class to manage clinical data in CSV format.

- Loopy belief propagation (Murphy et al., 1999). This module supports inference on probabilistic graphical models using loopy belief propagation. The GAMMA suite includes classes for message creation and propagation on an arbitrary graph structure.
- Markov random field (Geman and Geman, 1999) representation and inference. This module defines a Markov random field (specifying neighborhood type and potential function), and infers the mode of the defined Markov random field using loopy belief propagation.
- Data resampling and model aggregation. Researchers can use this module to resample a data set based on bootstrap or Jackknife resampling. This module generates a histogram for a model ensemble and identifies the mode of that histogram.

3. Experimental results

We present an example of using the GAMMA suite to compare subjects with Alzheimer's disease to normal controls. This study of Alzheimer's disease included 29 subjects individuals recruited from the registry of the Washington University Alzheimer Disease Research Center. For the Alzheimer's disease group ($N = 19$), the mean age was 76 years ($SD = 5$), and female:male ratio was 13:6. For the normal controls ($N = 12$), the mean age was 77 years ($SD = 6$), and female:male ratio was 10:2.

A high-resolution T1-weighted structural MR examination of each subject was acquired, using a MP-RAGE sequence ($TR = 9.7$ msec, $TE = 4$ msec, flip angle = 10, $TI = 20$ msec, $TD = 200$ msec) using a Siemens 1.5-T Vision system (Erlangen, Germany). Pixel size was 1 by 1 mm, slice thickness was 1.25 mm, and image size was $256 \times 256 \times 150$ voxels. The raw data are available from the National fMRI Data Center (<http://www.fimridc.org>), with access number 2-2004-1168x. We used the image-processing pipeline described in section A1.3 to obtain an effect map; in this effect map, $V_I^k = 1$ represented volume reduction and $V_I^k = 0$ represented no volume reduction. The function variable C represented whether or not a subject had Alzheimer's disease ($C = 0$, normal elderly; $C = 1$, Alzheimer's disease).

Figure 4 provides an overview of the process of creating a new project. We set the home directory (Figure 4 (a)), loaded the image list (Figure 4 (b)), imported the clinical variable (Figure 4 (c)), and set up algorithm parameters (Figure 4 (d)). We then used GAMMA-EL with Jackknife resampling to detect voxels that best distinguish the control and the Alzheimer's groups. GAMMA-EL required approximately 8 hours (using a Linux desktop with Intel Xeon 3.4GHz CPU and 4Gb RAM) to find a single ROI whose volume was predictive of C . Based on the resulting model, for a new subject, if voxels in this ROI demonstrate volume reduction, we would label this subject as belonging to the Alzheimer's disease group with probability 0.95. Figure 5 (a) is a screenshot of using GAMMA GUI to view the generated label field. This region was primarily located in the left medial temporal lobe (Figure 5 (b)), although voxels on the right were also useful for discrimination.

We then used regional state inference to infer the regional state for this ROI; this task required approximately 3 hours. We subsequently used the generated regional state variable to predict C using naïve Bayes and support vector machines (SVM). For both naïve Bayes and SVM, the classification accuracy using ten-fold cross-validation was 0.97.

4. Conclusion and Discussions

The GAMMA suite can be used in four different ways. 1) Researchers can use the GAMMA suite for analyzing high-dimensional, undersampled data. 2) Researchers who are interested in high-dimensional data analysis can build their own applications based on the GAMMA suite. For example, a researcher who has developed a classification algorithm could merge the marker detection module with his/her classification algorithm. This process would result in a new predictive-model generation system. 3) Researchers can use algorithms provided in the GAMMA suite for other purposes. For example, the GAMMA suite includes classes for BN construction, BN structure and parameter computation, loopy belief propagation for graphical-model inference, and Markov random field computation. 4) Researchers can incorporate algorithms from the GAMMA suite into their software applications. Since the GAMMA suite is an open-source, cross-platform, and well-documented library, it can be incorporated readily into other software applications as a library.

The GAMMA suite emphasizes Bayesian data mining; that is, it is not a software package for image processing, image visualization, or general-purpose classification. For image processing, researchers could use SPM or FSL, among other choices. For image visualization, researchers could use MRICro, SPM, or FSL. For general-purpose classifiers, researchers could use Weka. The GAMMA suite uses widely accepted file formats to support interoperability.

Algorithms in the GAMMA suite are designed to handle binary effect maps. The problem domain of interest is brain-image data analysis, primarily lesion-based studies, functional MR-based studies, and morphometric studies. The majority of neuroimaging experiments belong to one of these three study types; algorithms in the GAMMA suite are widely applicable to this domain.

An important question is validity of using binary effect maps. For lesion-based studies, most studies focus on how the lesioned voxels (or brain regions) are associated with a deficit. Usually investigators manually delineate lesioned voxels. Therefore, labeling a voxel as lesioned/non-lesioned is natural. For functional MR-based studies, we focus on whether or not voxels are activated during a task, and how this activation pattern is associated with other variables. Therefore, using binary activation map is also reasonable (Smith et al., 1998, Larsen et al., 2007, Cosman et al., 2004, Lashkari et al., 2010, Chen and Herskovits, 2007b). For morphometric studies, we can consider tissue loss in a voxel location to be associated with a latent variable that indicates whether or not this voxel is affected by the disease; from this perspective, the practice of using a binary effect map to model morphometric changes is justified. However, using binary effect maps is a limitation of our algorithms, because thresholding a continuous variable may cause loss of information.

One of the limitations of our current implementation of GAMMA-EL and RSI is its computational requirements. For an experiment including 29 high-resolution MR images of size $256 \times 256 \times 150$, GAMMA-EL requires approximately 8 hours using a Linux desktop with Intel Xeon 3.4GHz CPU and 4Gb RAM. RSI is also computationally intensive; the computation time for RSI is determined primarily by the number of voxels in the ROI. For a ROI consisting of approximately 3000 voxels, RSI requires approximately 3 hours using the same Linux desktop described above. One solution to this problem is to employ parallel computing: for GAMMA-EL, both the search for representative voxels and the data-resampling step can be parallelized, and for RSI, the online Gibbs sampling step can be parallelized. We plan to implement parallel versions of GAMMA-EL and RSI, based on the CUDA computing engine (<http://developer.nvidia.com/category/zone/cuda-zone>).

One direction of our future works is a comprehensive simulation-based evaluation of GAMMA's robustness to noise and misregistration. We will generate a set of simulated data sets with different noise and misregistration levels. Because GAMMA uses Markov random fields to improve robustness and GAMMA's Bayesian foundation has the benefit of graceful degradation as noise (or misregistration) increases (Gelman et al., 2003), we hypothesize that GAMMA will graceful degradation as noise or misregistration increases.

Information Sharing Statement

Both the entire source code and the documentation are available at http://www.nitrc.org/projects/gamma_suite, under the GPL license. The GAMMA suite compiles under Mac OS X, Linux, and Microsoft Windows. The GAMMA suite does not depend on other software packages. The accompanying documentation includes a tutorial, installation instructions, a description of how to use the GAMMA suite, and an extensive API reference.

Acknowledgements

This work was supported by National Institutes of Health grant R03 EB-009310. This work was supported by National Institutes of Health grant R01 AG13743, which is funded by the National Institute of Aging, and the National Institute of Mental Health; this work was also supported by the American Recovery and Reinvestment Act.

Appendix

A1. The pipeline for generating an effect map

Although the GAMMA suite analyzes effect maps to generate structure-function models, this software does not generate effect maps. In this appendix we provide examples of how to generate effect maps for several commonly encountered scenarios.

A1.1 Lesion-deficit analysis

For lesion-deficit studies, we can delineate abnormal brain voxels in MR or CT images using either manual labeling or automatic segmentation tools (Admiraal-Behloul et al., 2005). After generating binary lesion maps, we must register each subject's lesion map to a brain template. The registration process is as follows:

1. We co-register the image from which a subject's lesion map was derived (for lesion studies, this could be T2, diffusion-weighted imaging (DWI) or perfusion-weighted imaging, among other sequences) to that subject's high-resolution T1-weighted MR image. Then we apply the co-registration parameters to the lesion map.
2. We register the subject's T1-weighted MR image to the template (e.g., using SPM). We then apply the resulting registration parameters to the T1-registered lesion map resulted in step 1. The lesion maps will now be registered to the template.

A1.2 Functional MR analysis

We preprocess the data using SPM (<http://www.fil.ion.ucl.ac.uk/spm/>) or Voxbo (<http://www.voxbo.org>), resulting in a statistical parametric map (T -map or F -map) for a contrast. We then choose a significance level α to threshold these statistical parametric maps, and generate binary maps representing voxel activation: each voxel assumes a value in $\{0, 1\}$, corresponding to off (no activation) and on (activation), respectively. The registration process is as follows:

1. We co-register a subject's image from which the activation map is derived, to that subject's high-resolution T1-weighted MR image. Then we apply the resulting co-registration parameters to the activation map.
2. We register the subject's T1-weighted MR image to the template (e.g., using SPM). We then apply the resulting registration parameters to the T1-registered activation map obtained in step 1. The activation map will now be registered to the template.

Notice that Step 1 and 2 are included in the standard fMR image processing procedures in SPM. Therefore, if users use SPM or other software which has these procedures, these two steps are not necessary.

A1.3 MR-based morphometric studies

For brain-morphometry experiments, we can use the image-processing pipeline described in (Chen and Herskovits, 2006) to generate regional volume maps. In these volume maps, voxel intensity represents the voxel-wise regional volume of gray matter, white matter, or cerebrospinal fluid. These volume maps are defined in a common stereotaxic space; that is, the process of generating a voxel-wise morphometry map yields an image volume that is registered to a template selected by the user. After thresholding (the threshold can be determined by sample statistics such as sample median, or determined by clinical experts), we obtain a binary effect map, in which voxels with value 1 represent volume reduction, and voxels with value 0 represent no reduction.

A.2 Robustness analysis

We reported our assessment of how the significance level used to threshold SPM maps affects GAMMA's performance in Table 3 of (Chen and Herskovits, 2007a). In that report,

we applied GAMMA with ensemble learning and RSI to an fMR study of young, nondemented older, and demented older adults. We performed three experiments: (a) young versus nondemented older, (b) young versus demented older, and (c) nondemented older versus demented older. We compared classification accuracy and area under the ROC curve (AUC) for each of five different thresholds (significance level = 0.05, 0.025, 0.01, 0.0075, 0.005), and found that, for the experiments involving young versus demented older subjects and young versus nondemented older subjects, classification performance was stable with respect to this threshold. For the experiment involving nondemented versus demented older subjects, classification performance changed with the threshold; this result is consistent with the spatial diffuseness of regions characterizing group differences, and most likely resulted from the noisy activation maps for the demented older group.

Reference

- Admiraal-Behloul F, van den Heuvel DM, Olofsen H, van Osch MJ, van der Grond J, van Buchem MA and Reiber JH (2005) Fully automatic segmentation of white matter hyperintensities in MR images of the elderly, *Neuroimage*, 28, 607–17. [PubMed: 16129626]
- Chen R and Herskovits EH (2005a) A Bayesian Network Classifier with Inverse Tree Structure for Voxel-wise MR Image Analysis, In Proceedings of the eleventh conference of SIGKDD.
- Chen R and Herskovits EH (2005b) Graphical-model based morphometric analysis, *IEEE Transaction on Medical Imaging*, 24, 1237–1248.
- Chen R and Herskovits EH (2006) Network analysis of mild cognitive impairment, *NeuroImage*, 29, 1252–1259. [PubMed: 16213161]
- Chen R and Herskovits EH (2007a) Clinical diagnosis based on bayesian classification of functional magnetic-resonance data, *Neuroinformatics*, 5, 178–88. [PubMed: 17917129]
- Chen R and Herskovits EH (2007b) Graphical-model-based multivariate analysis of functional magnetic resonance data, *NeuroImage*, 35, 635–647. [PubMed: 17258473]
- Chen R, Hillis AE, Pawlak M and Herskovits EH (2008) Voxelwise Bayesian Lesion Deficit Analysis, *NeuroImage*, 40, 1633–1642. [PubMed: 18328733]
- Cooper GF and Herskovits EH (1992) A Bayesian method for the induction of probabilistic networks from data, *Machine Learning*, 9, 309–347.
- Cosman ER, Fisher JW and Wells WM (2004) Exact MAP Activity Detection in fMRI Using a GLM with an Ising Spatial Prior, In *MICCAI*, pp. 703–710.
- Duda R, Hart R and Stork DG (2001) *Pattern Classification (second edition)*, Wiley, New York.
- Friston KJ (2005) Models of brain function in neuroimaging, *Annu Rev Psychol*, 56, 57–87. [PubMed: 15709929]
- Friston KJ and Frith CD (1995) Schizophrenia: a disconnection syndrome?, *Clin Neurosci*, 3, 89–97. [PubMed: 7583624]
- Gelman A, Carlin JB, Stern HS and Rubin DB (2003) *Bayesian Data Analysis*, 2nd edition, Chapman and Hall/CRC.
- Geman S and Geman D (1999) Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, *IEEE. Trans. PAMI*, 6, 721–741.
- Larsen S, Kikinis R, Talos IF, Weinstein D, Wells W and Golby A (2007) Quantitative comparison of functional MRI and direct electrocortical stimulation for functional mapping, *Int J Med Robot*, 3, 262–70. [PubMed: 17763497]
- Lashkari D, Sridharan R, Vul E, Hsieh P-J, Kanwisher N and Golland P (2010) Nonparametric Hierarchical Bayesian Model for Functional Brain Parcellation, In *MMBIA: IEEE Computer Society Workshop on Mathematical Methods in Biomedical Image Analysis*.
- Murphy K, Weiss Y and Jordan M (1999) Loopy belief propagation for approximate inference: an empirical study, In *UAI* 1999.

- Pearl J (1988) Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann Publishers, San Mateo, CA.
- Smith SM, Jenkinson M, Woolrich MW, Beckmann CF, Behrens TE, Johansen-Berg H, Bannister PR, De Luca M, Drobnjak I, Flitney DE, Niazy RK, Saunders J, Vickers J, Zhang Y, De Stefano N, Brady JM and Matthews PM (2004) Advances in functional and structural MR image analysis and implementation as FSL, Neuroimage, 23 Suppl 1, S208–19.
- Smith SM, Matthews PM, Gurd JM and Slater A (1998) Multi-subject statistic map combination, In HBM98.
- SPM (2010) <http://www.fil.ion.ucl.ac.uk/spm/>, In SPM.

Figure 1.
The architecture of the GAMMA suite.

Figure 2.
GAMMA and GAMMA-EL.

Figure 3.

A schematic representation of the processing pipeline in GAMMA GUI.

Figure 4.
Creating a new project.

Figure 5. Using GAMMA GUI to analyze the Alzheimer's disease data described in Section 3. Left: Viewing the label field using GAMMA GUI. Right: Using MRIcro to display GAMMA results. The label field is red and is overlaid on the atlas. Images are in radiologic convention.