

User Manual  
Parameter Estimation for a Compartmental Tracer Kinetic  
Model Applied to PET data

Guadalupe Ayala<sup>1</sup>

Christina Negoita<sup>2</sup>

Rosemary A. Renaut<sup>3</sup>

May 12, 2005

<sup>1</sup>This work was supported in part by the Arizona Alzheimer's Disease Research Center which is funded by the Arizona Department of Health Services, NIH grant EB 2553301, NSF grant CMG-02223 and the Underrepresented Graduate Enrichment Match (UGEM) Fellowship award to Guadalupe Ayala

<sup>2</sup>Department of Mathematics, Oregon Institute of Technology, Klamath Falls, OR ([negoitac@oit.edu](mailto:negoitac@oit.edu)), Ph. 541 885 1474, <http://snoopy.oit.edu/~negoitac>.

<sup>3</sup>Department of Mathematics and Statistics, Arizona State University, Tempe, AZ 85287-1804 ([renaut@asu.edu](mailto:renaut@asu.edu)), Ph. 480 965 3795, Fax 480 965 8119, <http://math.asu.edu/~rosie>

## Abstract

This manual describes the functionality of an application developed with  $\text{\textcircled{R}}$ MATLAB's<sup>1</sup> graphical interface for the following analyses:

- FDG pixel-wise parameter estimation, such as  $k_1 - k_6$  where
  - $k_1$  is the transport rate from the blood to the extra-vascular space
  - $k_2$  is the transport rate back from the extra-vascular space to the blood
  - $k_3$  is the phosphorylation rate of the intra-cellular FDG by hexokinase enzymes to FDG-6-phosphate
  - $k_4$  is the dephosphorylation rate of the intra-cellular FDG-6-phosphate back to FDG.
  - $k_5$  is the spill-over from blood into tissue coefficient
  - $k_6$  is  $(k_1 * k_3) / (k_2 + k_3)$  which is proportional to the local cerebral metabolic rate of glucose and computed explicitly by pixel by pixel analysis
  - $\mathbf{K}$  is an analog to  $k_6$  computed using PATLAK analysis, which assumes  $k_4 = 0$ . It is the slope of the regression line in the PATLAK analysis that is valid only when  $k_4 = 0$ .
- Investigating different FDG parameter estimation methods.
- Investigating the effect of different filters, parameter constraints, and clustering techniques on the resulting FDG parameters.

Brief discussion on the use of Positron Emission Tomography as a diagnostic tool for Alzheimer's Disease studies is provided. Appropriate references to the standard literature give background on the topic and methods. Requirements for the tool and a complete description for its use are included.

---

<sup>1</sup> $\text{\textcircled{R}}$ MATLAB is a registered trade mark of The Mathworks, Inc.

# Contents

<b>1</b>	<b>Introduction</b>	<b>6</b>
<b>2</b>	<b>Kinetic Parameter Estimation</b>	<b>7</b>
2.1	Purpose of Kinetic Parameter Estimation . . . . .	7
2.2	Compartmental Model for FDG Dynamics . . . . .	7
2.3	Spill Over and Partial Volume . . . . .	8
2.4	Extensions . . . . .	8
<b>3</b>	<b>Requirements</b>	<b>9</b>
3.1	Hardware Requirements . . . . .	9
3.2	Software Requirements . . . . .	9
3.2.1	Operating System . . . . .	9
3.2.2	Necessary Components . . . . .	9
<b>4</b>	<b>Input Description, Format, File Type and Naming Convention Requirements</b>	<b>10</b>
4.1	The Plasma Time Activity Curve (PTAC) . . . . .	10
4.1.1	PTAC File Types . . . . .	10
4.2	The Tissue Time Activity Curve (TTAC) For Volume Data . . . . .	11
4.2.1	TTAC Volume File Type . . . . .	11
4.2.2	TTAC Volume File Format . . . . .	11
4.3	The Tissue Time Activity Curve (TTAC) for Cluster Data . . . . .	11
4.3.1	TTAC Cluster File Type . . . . .	11
4.3.2	TTAC Cluster File Format . . . . .	12
4.4	Purpose of Constraints (Only for Slice/Volume Data Runs) . . . . .	13
4.4.1	Constraint Sources . . . . .	13
4.4.2	Constraint Types . . . . .	14
4.4.3	Required Files to Set Constraints . . . . .	14
4.5	The Cluster Map . . . . .	15
4.5.1	Cluster Map File Type . . . . .	15
<b>5</b>	<b>Output Files and their Contents for Each Type of Run</b>	<b>17</b>
5.1	File Types and File Naming Conventions . . . . .	17
5.1.1	Header Files . . . . .	17
5.1.2	Results Files . . . . .	17
5.1.3	*.mat Files . . . . .	17
5.2	Types of Runs and the Format of their Output Files . . . . .	18
5.2.1	Single Cluster . . . . .	18
5.2.2	Multiple Clusters . . . . .	18

5.2.3	Single Slice . . . . .	19
5.2.4	Multiple Slices . . . . .	19
5.2.5	Entire Volume . . . . .	20
<b>6</b>	<b>Estimation Methods and Options</b>	<b>21</b>
6.1	Kinetic Parameter Estimation Methods . . . . .	21
6.1.1	The Generalized Linear Least Squares Algorithm - GLLS . . . . .	21
6.2	Options . . . . .	21
6.2.1	Estimation Model Option . . . . .	21
6.2.2	Process CSF Option . . . . .	22
6.2.3	Spatial Segmentation Option . . . . .	22
6.2.4	Filtering Option . . . . .	22
<b>7</b>	<b>General Procedures</b>	<b>23</b>
7.1	Reading Input . . . . .	23
7.2	Selecting a Data Processing Method . . . . .	23
7.3	Selecting a Constraints Option (Only for Volume Data Runs) . . . . .	23
7.4	Selecting a Constraints Source (Only for Volume Data Runs) . . . . .	23
7.5	Starting a Run . . . . .	24
7.6	Viewing Current Run Header . . . . .	24
7.7	View Data Fit Plot (Only For Cluster data Runs) . . . . .	24
7.8	Viewing Current Results for Kinetic Parameters . . . . .	24
7.8.1	Viewing Results as a *.txt file . . . . .	24
7.8.2	Viewing Results as.mat files (Only for Volume data runs) . . . . .	24
7.9	Saving Results . . . . .	25
7.10	Opening Past Run Headers . . . . .	26
7.11	Opening Past Run Results and *.mat Files . . . . .	26
7.12	Resetting Application . . . . .	26
<b>8</b>	<b>Sample of a Slice Run</b>	<b>27</b>
8.1	Step 1: Read PTAC Data from File . . . . .	27
8.2	Step 2: Read TTAC From File . . . . .	30
8.3	Step 3: Select Constraint Option . . . . .	32
8.4	Step 4: Select Constraint Source (if necessary, Only for Volume Data Runs) . . . . .	33
8.5	Step 5: Pick Model to Use . . . . .	35
8.6	Step 6: Verify Run Header Information Before Start of Run . . . . .	36
8.7	Step 7: Indicate the CSF processing Option and Start Run . . . . .	38
8.8	Step 8: View Results Automatically . . . . .	39
8.9	Step 9: Save Results . . . . .	41
8.10	Step 10: View Result Files and Header File . . . . .	43
8.11	Step 11: Reset Application . . . . .	44
<b>9</b>	<b>Future Implementation</b>	<b>51</b>
<b>10</b>	<b>Results Section</b>	<b>52</b>
<b>11</b>	<b>Installation, Execution, and Developers Guide</b>	<b>55</b>
11.1	Downloading and Installing Application . . . . .	55
11.2	Execution of Application . . . . .	55
11.3	Developer's Guide . . . . .	56

**12 Notation and Acronyms**

**57**

**A GLLS Note**

**60**

# List of Figures

2.1	Compartment model structure for FDG metabolism. . . . .	7
4.1	Real Study PTAC . . . . .	10
4.2	TTAC data: Brain Slice 16 over 22 Time Intervals . . . . .	12
4.3	TTAC data: An Entire Brain Volume at the Last Time Frame . . . . .	13
4.4	TTAC Cluster Curves . . . . .	14
7.1	Data Fit For Cluster Runs . . . . .	25
8.1	Step 1: Open File Menu . . . . .	28
8.2	Step 1: Browse for PTAC data File . . . . .	28
8.3	Step 1: Interface Displays PTAC Data . . . . .	29
8.4	Step 2: Open File Menu and Browse Menu to Read Slice . . . . .	30
8.5	Step 2: Dialog to Set Options for Reading TTAC File . . . . .	30
8.6	Step 2: Dialog to Enter Slice Number . . . . .	31
8.7	Step 2: Browse for TTAC File that Contains Slice . . . . .	31
8.8	Step 3: The Constraint Options Pop up Menu . . . . .	32
8.9	Step 4: Constraint Source Pop up Menu . . . . .	33
8.10	Step 4: Browser Dialog to Select Constraint Input File . . . . .	34
8.11	Step 5: Pick an Estimation Model . . . . .	35
8.12	Step 6: Open View Menu and View Run Header Data . . . . .	36
8.13	Step 6: Display Dialog for Run Header Information . . . . .	37
8.14	Step 7: Run Menu . . . . .	38
8.15	Step 7: Prompt for CSF Option . . . . .	38
8.16	Step 8: End of Run Dialog To Automatically View Results . . . . .	39
8.17	Step 8: Prompt to Scale Results for Display . . . . .	39
8.18	Step 8: Prompt to Mark Negative Pixels with Magenta . . . . .	39
8.19	Step 8: Pick Slices to View . . . . .	40
8.20	Step 9: Open File Menu to Save Results . . . . .	41
8.21	Step 9: Dialog to Enter Filename Prefix to Save Results . . . . .	42
8.22	Step 10: Open File Menu to View Past Runs Header and Result Files . . . . .	43
8.23	Step 10: Select Result File to View . . . . .	44
8.24	Step 10: slice_run_example_results.txt . . . . .	45
8.25	Step 10: Change the File Type to *.mat . . . . .	45
8.26	Step 10: Select Results File to View . . . . .	46
8.27	Step 10: Display window for $k_2$ *.mat File . . . . .	46
8.28	Step 10: Select Header File to View . . . . .	47
8.29	Step 10: Header File Displayed After Run . . . . .	47
8.30	Step 11: Reset To Start New Run . . . . .	48

8.31	The Volume Run Flow Chart . . . . .	49
8.32	The Cluster Run Flow Chart . . . . .	50
10.1	Parameter $k_1$ for Slice 16, assuming $k_4 = 0$ , spillover = 0, and No Constraints . . . .	52
10.2	Parameter $k_2$ for Slice 16, assuming $k_4 = 0$ , spillover = 0, and No Constraints . . . .	53
10.3	Parameter $k_3$ for Slice 16, assuming $k_4 = 0$ , spillover = 0, and No Constraints . . . .	53
10.4	Parameter $k_6$ for Slice 16, assuming $k_4 = 0$ , spillover = 0, and No Constraints . . . .	54
10.5	Parameter $\mathbf{K}$ for Slice 16, assuming $k_4 = 0$ , spillover = 0, and No Constraints . . . .	54

# Chapter 1

## Introduction

This user manual is a guide on how to use an application tool designed to estimate kinetic parameters which describe Fluoro-Deoxy-Glucose (FDG) metabolism in the brain of Alzheimer's disease patients. The tool uses dynamic PET data obtained from one-dimensional, two-dimensional or three-dimensional measurements. Chapter 2 provides the background information, the theory and purpose of kinetic parameter estimation, and a brief description of the compartmental kinetic model on which the estimation is based. It further describes other alternative applications for the tool in a computational research environment. Chapter 3 lists the hardware and software requirements. Chapter 4 contains the information on the inputs for the application. This section explains in detail the format, file type and naming convention for all of the inputs including Plasma Time Activity Curve (PTAC) , Tissue Time Activity Curve (TTAC) , Constraints, and Cluster Map. It explains in detail which of these inputs are necessary and which are optional. Chapter 5 describes all the different classes of possible output files, their file types and the naming conventions. Also, it outlines which output files are created for each type of run. It also lists all of the variables that are displayed in the header file for each type of run. Chapter 6 gives information on Generalized Linear Least Squares (GLLS), which is the numerical algorithm used in the estimation of kinetic parameters, and explains the different user options. Chapter 7 provides the step by step commands to read inputs, start runs, view results, and save results. Chapter 8 is intended to familiarize the user with the application screens and the run process by a walk through of a sample slice run, with detailed directions and shows the application screens along the way. Chapter 9 explains the future capabilities that are planned to be added to the application, and how this could help to evaluate estimation methods and filtering techniques. Chapter 10 shows results given by the estimation application. Finally, in Chapter 11 you will find an installation and execution guide, as well as information on the application modules. Notation used is provided in the final section.

## Chapter 2

# Kinetic Parameter Estimation

### 2.1 Purpose of Kinetic Parameter Estimation

We are interested in using PET to image brain activity in patients with Alzheimer's disease (AD). In AD studies, one way to measure disease progression is by measuring FDG, which is an analog of glucose uptake in the brain. Studies [14, 18, 19] which determine a local cerebral metabolic rate (LCMR) of FDG uptake in a region of interest have proved successful in understanding AD progression.

More specific information may be obtained by estimating the individual kinetic parameters which describe FDG metabolism. In particular, it is believed that the individual kinetic parameters may be used for early detection of AD. This application is used to estimate the kinetic parameters in order to be able to focus toward understanding the spatial distribution of kinetic parameters in AD, as well as toward developing a precise measure for utilization in the early detection of AD [15].

### 2.2 Compartmental Model for FDG Dynamics

The estimation of FDG dynamics kinetic parameters is based on the following compartmental tracer kinetic model for FDG:

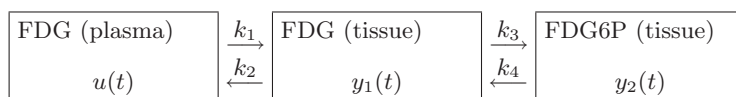


Figure 2.1: Compartment model structure for FDG metabolism.

The PET scanner provides a measure of the combined FDG  $y_1(t)$  and phosphorylated FDG concentration in tissue  $y_2(t)$ , ie a measure of  $y(t) = y_1(t) + y_2(t)$  [15], where

$$\begin{aligned} \frac{dy_1}{dt} &= k_1 u(t) - (k_2 + k_3) y_1(t) + k_4 y_2(t) \\ \frac{dy_2}{dt} &= k_3 y_1(t) - k_4 y_2(t) \\ y_1(0) &= 0, y_2(0) = 0. \end{aligned} \tag{2.1}$$

The system rate constants are interpreted as following:

- $k_1$  is the transport rate from the blood to the extra-vascular space
- $k_2$  is the transport rate back from the extra-vascular space to the blood
- $k_3$  is the phosphorylation rate of the intra-cellular FDG by hexokinase enzymes to FDG-6-phosphate
- $k_4$  is the dephosphorylation rate of the intra-cellular FDG-6-phosphate back to FDG.

and  $u(t)$  is the input data, namely the FDG tracer concentration in plasma.

Although clinical studies [14] have shown that the dephosphorylation-phosphorylation rate  $k_4$  is nonzero, many models such as the the PATLAK analysis [3] assume  $k_4 = 0$ .

## 2.3 Spill Over and Partial Volume

Real data suffers from noise introduced by spill-over and partial volume effects. Spill-over is usually bidirectional and accounts for a percentage of the tracer in plasma being counted as total tissue-tracer, as well as for some of the tracer in tissue being counted as plasma-tracer (due to limited spatial resolution). This leads to a blurring effect in the image. Spill-over coefficients can be introduced in the tracer model, and estimated as additional parameters. However, they can complicate the numerical identifiability of model parameters [9]. Consequently, one or more venous samples are made late in the study to improve their identification. Other methods for removing spill-over effects have been considered [11], but they suffer from being dependent on estimates of the imaged-object geometry as well as on other factors, like cardiac motion. Principal component analysis is also introduced to derive tissue-time activity curves free from spill-over effects [2], but we opt to consider the model dependent method which can be easily incorporated into the compartmental kinetic model.

For brain studies which use an image-derived input function  $u(t)$ , the tissue into blood spill-over coefficient is estimated [4] such that one obtains a spill-over corrected input function. In this case, clearly one needs to only account for the spill-over from blood into tissue coefficient, called here parameter  $k_5$ . We also introduce the parameter  $k_6 = (k_1 k_3)/(k_2 + k_3)$ , which is proportional to the local cerebral metabolic rate of glucose, and the parameter  $\mathbf{K}$ , which is the analogue to  $k_6$  but obtained by PATLAK analysis, and is valid only when  $k_4 = 0$ .

Partial volume effects (PVE) occur due to the intrinsic spatial resolution of the PET camera; in objects smaller than the resolution of the tomograph, true tracer concentration will be underestimated. The pixel target tissue mixes with surrounding tissue, and this can lead to an inaccurate parameter estimation overall [22, 13, 21]. This is a significant concern for region of interest studies of the heart. Several methods for correcting for partial volume effects have been considered [1, 11]. However, this is currently an active area of research, and we focus here only on model dependent methods which can account for PVE.

## 2.4 Extensions

This application also allows the user to compare results with respect to the computational and estimation methods, filters, constraints, and input sources chosen by the user. Comparing the results could help find out what are the best estimation methods, what are the constraints or what is the best filtering technique that provides optimal results. Results could be compared with expected results according to theoretical information, and an educated decision can be made on what are the best computational methods to use for every situation.

# Chapter 3

## Requirements

### 3.1 Hardware Requirements

This a highly computationally intensive application. The following is the system on which the application was tested and suggested to run:

- 2.6 GHz Intel Pentium 4 Processor or equivalent.
- Minimum 512 MB of RAM.

### 3.2 Software Requirements

#### 3.2.1 Operating System

This application will run on any linux based operating system.

#### 3.2.2 Necessary Components

This application requires ®MATLAB Version 6.0 or higher to be installed on your system. The following toolbox needs to be installed:

- Nonlinear Optimization Toolbox

## Chapter 4

# Input Description, Format, File Type and Naming Convention Requirements

This section describes the input requirements for the application. It outlines file types, format, and file naming conventions for the input data.

### 4.1 The Plasma Time Activity Curve (PTAC)

The PTAC,  $u(t)$ , displays the FDG concentration in the blood over time, see Figure 2.1. The FDG plasma time activity curve (PTAC) can be derived either invasively, by blood sampling or non-invasively, from a region of interest in the reconstructed PET images. [23]. Figure 4.1 shows a real time study PTAC.

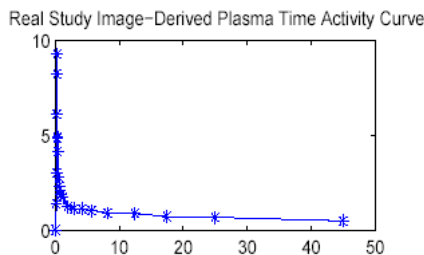


Figure 4.1: Real Study PTAC

#### 4.1.1 PTAC File Types

The source of the PTAC can be either blood sample data or a PTAC reconstruction from \*.cpt files.

- If collected through blood sampling, the data should be read from a file of type \*.txt. The \*.txt file format is required to be as follows:  
The text file should contain 2 columns. The length of the columns will be equal to the number

of blood samples. The first column should display the time vector for each of the time points at which a blood sample was taken in minutes. The second column should display the FDG concentration in blood for each of the time points.

- If collected through PTAC reconstruction, the data should be read from several files of type \*.cpt. The \*.cpt files are created using the CTI molecular imaging software. Note that this form means that any approximate input data, whether from simultaneous estimation or a population based model can also be used, see papers [12, 20, 8, 10].

## 4.2 The Tissue Time Activity Curve (TTAC) For Volume Data

The TTAC data can be read over an entire volume, as single or multiple slices of the entire brain volume. Each slice is a cross-sectional 2D image of the brain. TTAC volume data are PET images of the brain over time. PET images are collected by scanning the brains of patients over time. Furthermore, the brains are divided into a fixed number of slices. Each slice has its own time series. TTAC slice data is stored in 3D structure of size  $J * L * N$  where  $J * L$  is the image size of the slice in pixels, and  $N$  is the number of time points in the time series.

The entire volume data is stored in a 4D structure of size  $J * L * S * N$  where  $J * L$  is the image size of the slice in pixels,  $S$  is the number of slices in the entire volume, and  $N$  is the number of time frames in the time series. Figure 4.2 shows TTAC data of brain slice 16 over 22 time intervals.

Figure 4.3 shows TTAC data as the entire brain volume comprised of 32 slices at the last time frame.

### 4.2.1 TTAC Volume File Type

TTAC volume data should be read from a file of type \*.img. By convention, the TTAC filename should have numeric string embedded somewhere in the filename. This numeric string will be identified and interpreted as the \$PATIENT\_ID which will be used to create the directory where the output files are going to be saved. If more than one numeric string is embedded on the TTAC filename, then the first numeric string from left to right will be interpreted as the \$PATIENT\_ID.

### 4.2.2 TTAC Volume File Format

\*.img file - a bitmap graphic file.

## 4.3 The Tissue Time Activity Curve (TTAC) for Cluster Data

TTAC cluster curves are a representation of grouped data points on brain slices over time as one point over time. TTAC cluster curves are derived from entire brain volume. Cluster data is created by grouping together similar pixels according to their value within slice images. An example of 5 TTAC Cluster Curves is shown in Figure 4.4.

### 4.3.1 TTAC Cluster File Type

TTAC Cluster Curves should be read from a @MATLAB file of type \*.mat. By convention, the TTAC filename should have a numeric string embedded somewhere in the filename. This numeric

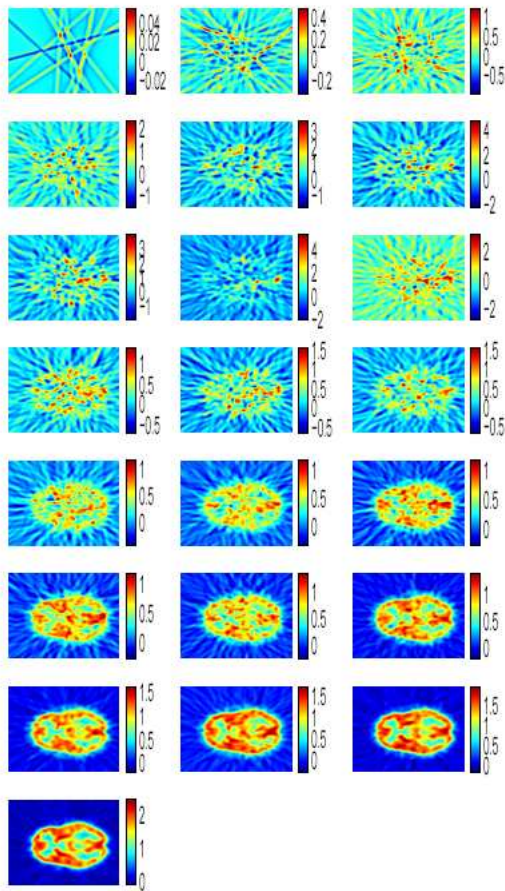


Figure 4.2: TTAC data: Brain Slice 16 over 22 Time Intervals

string will be identified and interpreted as the `$PATIENT_ID` which will be used to create the directory where the output files are going to be saved. If more than one numeric string is embedded on the TTAC filename, then the first numeric string from left to right will be interpreted as the `$PATIENT_ID`.

### 4.3.2 TTAC Cluster File Format

The `*.mat` file should contain the following variables names:

**grpcurves:** This variable holds the TTAC Cluster Curve Values. `grpcurves` variable should be a matrix of size  $C * N$  where  $C$  is the number of clusters and  $N$  is the number of time points.

**tm:** This variable holds the time vector information. Time vector should be of size  $1 * N$ .

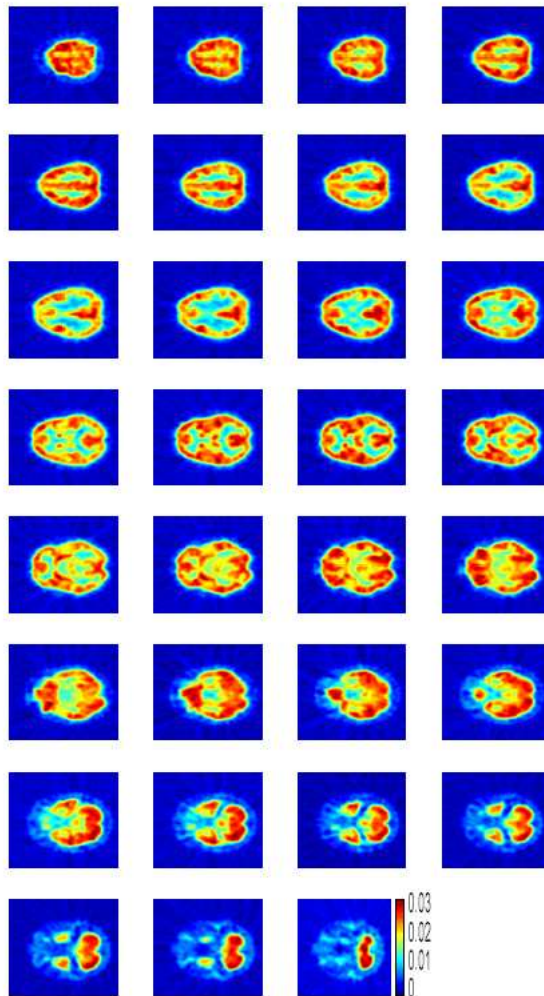


Figure 4.3: TTAC data: An Entire Brain Volume at the Last Time Frame

## 4.4 Purpose of Constraints (Only for Slice/Volume Data Runs)

Constraints are only used for volume runs and are not an available feature for cluster runs. Constraints are used as bounds during the calculation of kinetic parameters of the model. Constraints are optional and not necessary for a run, hence you can run without any constraints. They are intended to help improve estimates of the kinetic parameters.

### 4.4.1 Constraint Sources

Constraints are set from either the user input, a text file or an automatic calculation from cluster curves.

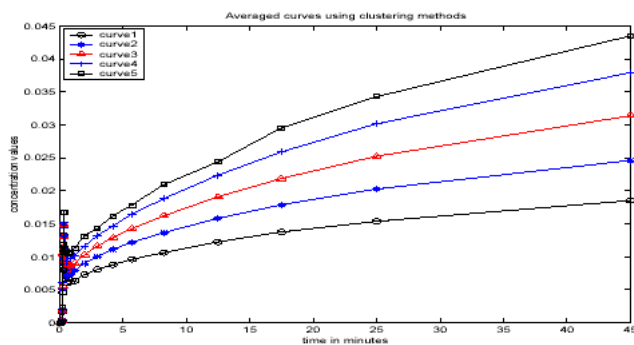


Figure 4.4: TTAC Cluster Curves

#### 4.4.2 Constraint Types

There are three types for constraints, positivity, global and by cluster. The user may also choose to run without constraints.

1. When the user selects to run method without constraints, then no constraints are imposed on the estimation of the kinetic parameters.
2. When the user selects Positivity Constraints, the lower bounds are set to 0 and upper bounds are set to 1. These type of constraints do not allow the kinetic parameter estimates to become negative.
3. When the user selects global constraints, the same set of constraints are used during the calculations of kinetic parameters on all of the volume data.
4. When the user selects constraints by cluster, different set of constraints are used during the calculations of the kinetic parameters on each cluster of the volume data. A cluster map is required when applying constraints by cluster, because we need to identify the label cluster of each pixel.

#### 4.4.3 Required Files to Set Constraints

Constraint files are required only when the user chooses to set the constraints from a text file or to automatically calculate the constraints from a cluster file.

1. If constraints are set from user input, no file is necessary.
2. If constraints are set by positivity constraints, no file is necessary. Consequently, the lower bounds will be set to 0 and upper bounds for all will be set 1 for all kinetic parameters.
3. If constraints are set from a text file with filename with extension \*.txt. This file should contain a matrix with the lower and upper bounds for all  $k_1 - k_5$  parameters. The following are examples of what the file might look like:

For the global constraints case:

```
0 1
0 1
0 1
0 1
0 1
```

The first column are the lower bounds and the second column are the upper bounds for  $k_1, k_2, k_3, k_4, k_5$  in that order.

For the by cluster constraints case:

```
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0

1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
```

The first matrix's rows are the lower bounds for  $k_1, k_2, k_3, k_4, k_5$  in that order. In this example each column lists the lower bounds for each of five clusters. The second matrix's rows are the upper bounds for the  $k_1, k_2, k_3, k_4, k_5$  in that order. Also, each column lists the upper bounds for each of the five clusters.

4. If automatically calculated from cluster curves, the file type should be \*.mat containing TTAC cluster data. From this data the application will calculate the lower and upper bounds for the  $k_1 - k_5$  kinetic parameters.

## 4.5 The Cluster Map

A Cluster Map maps every pixel on a slice to a cluster hence associating with each pixel a label pointing to its cluster membership. This information is used to apply different set of constraints when processing different pixels on a slice according to which cluster they belong. The Cluster Map file is only necessary when the user chooses to apply constraints by cluster and runs TTAC volume data.

### 4.5.1 Cluster Map File Type

The cluster map should be read from a \*.mat file.

#### Cluster File Map Format

The mat file should contain the following variable:

B: This is a variable of size  $J * L * 1$  where  $J$  is the slice width and  $L$  is the slice length.

## Chapter 5

# Output Files and their Contents for Each Type of Run

### 5.1 File Types and File Naming Conventions

Every Run will save its output into two or more of the following files:

#### 5.1.1 Header Files

The header file lists all the information required to replicate the run. Which variables are displayed on this file depends on the type of run. The naming convention for this file is  $F$ .header.txt where  $F$  is filename entered by the user in the **Save As** dialog while saving the results. Refer to Chapter 5.2 for a detailed description of this file for all the different types of runs. Chapter 5.2 classifies all runs into single cluster, multiple cluster, single slice, multiple slice, and entire volume, and describes their header file in detail.

#### 5.1.2 Results Files

The results file lists the resulting kinetic parameters for a cluster run or the filenames where the results were saved in case of volume data runs. The naming convention for this file is  $F$ .results.txt where  $F$  is a user defined filename while saving results. Refer to Chapter 5.2 for a detailed description of this file for all the different types of run. Chapter 5.2 classifies all runs into single cluster, multiple cluster, single slice, multiple slice, and entire volume, and describes their results file in detail.

#### 5.1.3 \*.mat Files

\*.mat file - MATLAB file

A single \*.mat file is used to store the information of a single kinetic parameter for a single slice of the brain. Each pixel on a slice image corresponds to a set of kinetic parameters. Consequently, for each slice run, 7 \*.mat files are created containing one variable for each of the kinetic parameters  $k_1, k_2, k_3, k_4, k_5, k_6$  and  $\mathbf{K}$  at different regions, where

- $k_1$  is the transport rate from the blood to the extra-vascular space.
- $k_2$  is the transport rate back from the extra-vascular space to the blood.

- $k_3$  is the phosphorylation rate of the intra-cellular FDG by hexokinase enzymes to FDG-6-phosphate.
- $k_4$  is the dephosphorylation rate of the intra-cellular FDG-6-phosphate back to FDG.
- $k_5$  is the spill-over from blood into tissue coefficient.
- $k_6$  is  $(k_1 * k_3)/(k_2 + k_3)$  computed explicitly by pixel by pixel analysis.
- $\mathbf{K}$  is an analog to  $k_6$  computed using PATLAK analysis, which assumes  $k_4 = 0$ .

The naming convention for this file is  $F\_SliceS\_kH.mat$  . The  $F$  is the user defined filename while saving the results,  $S\_NUMBER$  is the slice number, and  $H$  is the kinetic parameter number in the range of 1-6 and BigK.

## 5.2 Types of Runs and the Format of their Output Files

The types of runs can be classified by their type of Input. When the user chooses to save the results after a run, the data is written into header, results, and if applicable to \*.mat files. Which files are written and the information written depends on the type of run.

### 5.2.1 Single Cluster

The output for this type of run includes two files, a header file and results file.

#### Header File Format

The header file displays the following variables:

Data Type: Single Cluster

TTAC Filename: The TTAC Cluster Filename

PTAC Filenames: All the PTAC Filenames

Time Vector Size: The number of time frames

Time Vector: The vector with all the time point intervals

Model: The estimation model chosen by the user

Processing Method: The type of method used to calculate for  $k_1 - k_6$

Cluster Curve Number: The cluster curve number ran.

#### Results File Format

The results file contains the kinetic parameter results  $k_1 - k_6$ .

### 5.2.2 Multiple Clusters

The output for this type of run includes two files, a header file and results file.

#### Header File Format

The header file displays the following variables:

Data Type: Multiple Cluster Curves

TTAC Filename: TTAC Cluster Filename

TTAC Filename: The TTAC Cluster Filename

PTAC Filenames: All the PTAC Filenames  
Time Vector Size: The number of time frames  
Time Vector: The vector with all the time point intervals  
Model: The estimation model chosen by the user  
Processing Method: The type of method used to calculate for  $k_1 - k_6$   
Number of Cluster Curves: Number of cluster curves read from file

### Results File Format

The results file contains the kinetic parameter results  $k_1 - k_6$  for all clusters in the TTAC cluster file.

### 5.2.3 Single Slice

The output is a header file, results file, and 7 \*.mat files corresponding to each of the kinetic parameters  $k_1, k_2, k_3, k_4, k_5, k_6, \mathbf{K}$ .

#### Header File Format

The header file displays the following variables:

Data Type: Single Slice  
TTAC Filename: TTAC Volume Filename  
TTAC Filename: The TTAC Volume Filename  
PTAC Filenames: All the PTAC Filenames  
Time Vector Size: The number of time frames  
Time Vector: The vector with all the time point intervals  
Model: The estimation model chosen by the user  
Processing Method: The type of method used to calculate for  $k_1 - k_6$  and  $\mathbf{K}$   
Number of PTAC Files  
Number of Slices:1  
Slice Number  
Filter ON/OFF: Binary Option for Filtering Volume Data while Reading  
Spatial Segmentation Option: Binary Option for Spatially Segmenting Slices  
Process CSF: Binary Option to process CSF

### Results File Format

The Results file displays \*.mat filenames to which the kinetic parameter results were written.

### 5.2.4 Multiple Slices

The output is several header files, results files, and \*.mat files corresponding to each of the kinetic parameters for each slice.

#### Header File Format

The header file displays the following variables:  
Data Type: Multiple Slices  
TTAC Filename: TTAC Volume Filename

TTAC Filename: The TTAC Volume Filename  
PTAC Filenames: All the PTAC Filenames  
Time Vector Size: The number of time frames  
Time Vector: The vector with all the time point intervals  
Model: The estimation model chosen by the user  
Processing Method: The type of method used to calculate for  $k_1 - k_6$  and  $\mathbf{K}$   
Number of PTAC Files  
Number of Total Slices  
Slice Numbers  
Filter ON/OFF: Binary Option for Filtering Volume Data while Reading  
Spatial Segmentation Option: Binary Option for Spatially Segmenting Slices  
Process CSF: Binary Option to process CSF

### Results File Format

The Results file displays \*.mat filenames to which the kinetic parameter results were written.

### 5.2.5 Entire Volume

The output is several header files, results files, and 7 \*.mat files corresponding to each of the 7 kinetic parameters for each and all the slices in the entire volume. The kinetic parameters correspond to  $k_1 - k_6$  and  $\mathbf{K}$ .

### Header File Format

The header file displays the following variables:

Data Type: Multiple Slices  
TTAC Filename: TTAC Volume Filename  
TTAC Filename: The TTAC Volume Filename  
PTAC Filenames: All the PTAC Filenames  
Time Vector Size: The number of time frames  
Time Vector: The vector with all the time point intervals  
Model: The estimation model chosen by the user  
Processing Method: The type of method used to calculate for  $k_1 - k_6$  and  $\mathbf{K}$   
Number of PTAC Files  
Number of Total Slices in Volume: 31  
Slice Numbers: All Slices Were Read  
Filter ON/OFF: Binary Option for Filtering Volume Data while Reading  
Spatial Segmentation Option: Binary Option for Spatially Segmenting Slices  
Process CSF: Binary Option to process CSF

### Results File Format

The Results file displays \*.mat filenames to which the kinetic parameter results were written.

## Chapter 6

# Estimation Methods and Options

### 6.1 Kinetic Parameter Estimation Methods

The user has the ability to choose which method to use during the estimation of the kinetic parameters. The GLLS algorithm with one iteration will be the default method [16, 15].

#### 6.1.1 The Generalized Linear Least Squares Algorithm - GLLS

The GLLS algorithm is introduced in [8]. GLLS is an extension of the generalized least squares (GLS) algorithm to non-uniformly sampled data. Although GLLS has been used successfully in studies [8, 5, 3], there is no analytic proof of convergence of this algorithm. In addition, despite its derivation as an iterative algorithm, GLLS has mostly been used as a one step iteration - see for example [8, 5, 16, 15].

### 6.2 Options

Options allow the user to choose whether to process the Cerebral Spinal Fluid (CSF) region of the brain, spatially segment or filter the brain image slices before processing.

#### 6.2.1 Estimation Model Option

There is a model option available. This option allows the users to specify whether they want to account for spill-over, partial volume or if they want to assume  $k_4 = 0$ . Spill-over is bi-directional, as it accounts for a percentage of the tracer in plasma being counted as total tissue-tracer, as well as for some of the tracer in tissue being counted as plasma-tracer. But in our estimation we assume that the user PTAC input is already corrected for spill-over from tissue to plasma, so we only have to account for spill-over from plasma to tissue. Partial Volume effect is due to limited spatial resolution of PET scanners, and it does not allow an exact measurement of the FDG in brain tissue, so there is an underestimate of FDG concentration in small structures in the brain.  $k_4$  is the de-phosphorylation of FDG in the brain tissue and may be assumed to be zero since it is usually substantially small. The following are the 6 models available for the user to choose from:

- Model 1:  $k_4 = 0, SpillOver = 0$
- Model 2:  $k_4 = 0, SpillOver > 0$

- Model 3:  $k_4 = 0$ ,  $SpillOver > 0$  with PV
- Model 4:  $k_4 > 0$ ,  $SpillOver = 0$
- Model 5:  $k_4 > 0$ ,  $SpillOver > 0$
- Model 6:  $k_4 > 0$ ,  $SpillOver > 0$  with PV

The user can choose any of these models by clicking on the pop-up menu with the Model label. For more information on these models please refer to Appendix A.

### 6.2.2 Process CSF Option

The CSF option enables the user to choose whether the CSF region should be processed or not. This is useful, because a user might want to assume there is no activity in the CSF Region. Pixels on the slices are not processed if their intensity values are lower than a given threshold value. The intensity values of the CSF region are higher than 0.5. Furthermore, the background region intensity values are lower than 0.5. It is necessary to set the threshold value to 0.5 in order to segment out the background, but not the CSF region. Also, since CSF region pixel intensity values are lower than 1.0, we have to set the threshold value to 1.0 in order to segment the CSF region. Any pixel under 1.0 will not be processed, which includes most of the CSF region and background pixels.

### 6.2.3 Spatial Segmentation Option

The Spatial Segmentation Option allows the user to crop the slice image to segment the background of the slice out. This option is very useful if you want to process a smaller number of pixels, or provide a region of interest for analysis.

### 6.2.4 Filtering Option

The filter option allows the user apply an anisotropic diffusion filter to the slice images. Anisotropic diffusion [17] was developed to smooth an image, thus removing high frequency noise, while preserving the boundaries of structures of interest.

# Chapter 7

## General Procedures

### 7.1 Reading Input

1. Open **File menu**
2. Choose type of input to read either TTAC Cluster, TTAC Volume data, or PTAC data
  - a) **Read Single Cluster Curve** → Browse for cluster data file → **Enter Cluster Curve Number** → **Done**
  - b) **Read Multiple Cluster Curves** → Browse for cluster data file → **Done**
  - c) **Read Single Slice** → Select filter and spatial segmentation options → **Enter Slice Number** → Browse for volume data file → **Done**
  - d) **Read Multiple Slices** → Select filter and spatial segmentation options → Enter slices numbers comma separated [ex: 1,2,3] → Browse for volume data file → **Done**
  - e) **Read Entire Volume** → Select filter and spatial segmentation options → Browse for volume data file → **Done**
  - f) **Read PTAC data** → Browse of PTAC data files → **Done**

### 7.2 Selecting a Data Processing Method

Click on the **Data Processing Method** pop-up menu. Default is GLLS.

### 7.3 Selecting a Constraints Option (Only for Volume Data Runs)

Click on **Select Constraint Option** pop-up Menu. This pop-up Menu is disabled for cluster data runs.

### 7.4 Selecting a Constraints Source (Only for Volume Data Runs)

Click on the **Select Source for Constraints** pop-up menu. This pop-up menu is disabled when running with no constraints or positivity constraints.

## 7.5 Starting a Run

Open **Run** menu → Click on **Start Run**.

## 7.6 Viewing Current Run Header

After entering all the necessary input to start a run, it is possible to view the header file. This option aids the user in the verification of all the input parameters. By reviewing the header run data, the user can verify inputs like constraints, input filenames, and option settings are correct. After verification, the user can now proceed to start the run. To view run header you need to:

Open **View** menu → **View Run Header Data**

## 7.7 View Data Fit Plot (Only For Cluster data Runs)

This function is only available for cluster runs. The data fit plot allows the user to compare the original cluster TTAC in a run with the estimated cluster TTAC from the resulting  $k_1 - k_5$  kinetic parameters. You will be only to view the data fit plot after you run at least one of the cluster curves. To view the data fit plot:

Open **View** menu → **View Data Fit Plot**

The Figure 7.1 shows a example of data fit plot for a single cluster curve. In this case, the user ran cluster curve number 3 and then plotted the data fit. Note how closely this two curves lie, therefore by looking at these curves you should be able check how effective is the estimation method for calculating the kinetic parameters  $k_1 - k_5$ .

## 7.8 Viewing Current Results for Kinetic Parameters

It is possible to view current results at the end of each run. If the last previous run was volume data, the resulting kinetic parameter will be printed to different \*.mat files. Also, if the last previous run was cluster data, the resulting kinetic parameters will be printed to a \*.txt file.

### 7.8.1 Viewing Results as a \*.txt file

Open **View** menu → **View Kinetic Parameters**

### 7.8.2 Viewing Results as.mat files (Only for Volume data runs)

If completed run you want to view is single slice run, do the following:

Open **View** menu → **View Kinetic Parameters**

If completed run you want to view is a multiple slice run, do the following:

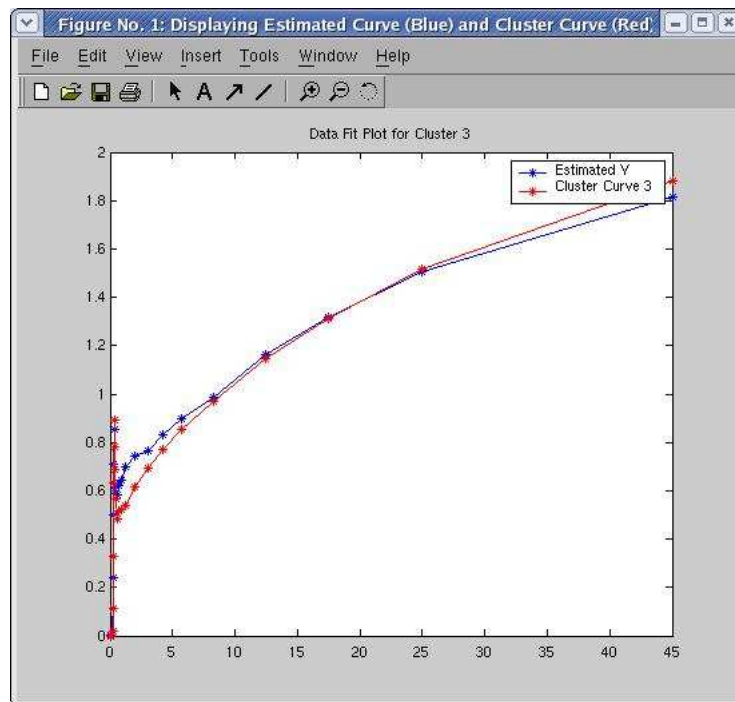


Figure 7.1: Data Fit For Cluster Runs

Open **View** menu → **View Kinetic Parameters** → Choose Slices to View → Click **Ok**

Note: that you can select several slices by holding the Ctrl key down while selecting slices from the list box.

## 7.9 Saving Results

Successful end of run is required to save results.

Open **File** menu → **Save Results As** → Type a user defined label in the dialog to use in the naming convention of all the results files. → Click **OK**

The resulting output is saved on the following type of files: header.txt files, results.txt files, or \*.mat files. Please refer to Chapter 5 for more information on the contents and formats of these files.

Unless the user browses to another directory, by default, the output files will be saved under \$HOME/Results/\$PATIENT\_ID, where \$HOME is the current user home directory and \$PATIENT\_ID is a numeric substring within the PTAC filename. The \$PATIENT\_ID is automatically derived from the TTAC filename by identifying the first numerical substring within the TTAC filename as the \$PATIENT\_ID.

## 7.10 Opening Past Run Headers

If the user wants to analyze past run results, they should be able to review the description of the run. Header files display all of the information required to replicate a run. Please refer to Chapter 5 for more information on this type of files.

Open **File** menu → **Open Header File** → Browse to directory where header file is located → **Pick Header File to View** → Click **Open**

## 7.11 Opening Past Run Results and \*.mat Files

If the user wants to analyze past runs results, they should be able to go back and look at all past run results data. The results file will be either \*.mat files for volume data runs or Results \*.txt files for cluster data runs.

Open **File** menu → **Open Results File** → Choose results file type in the **File Type** pop-up menu → Browse to directory where result file is located → **Pick Results File to View** → Click **Open**

Please refer to Chapter 5 regarding output for \*.mat and Results file description. You will be able to pick multiple results files by pressing Ctrl key while clicking to select the results files to view.

## 7.12 Resetting Application

It is possible to reset the application every time after the results are saved and a new run is to be started. This will re-initialize the user interface variables, so you can start additional runs. Resetting application ensures the run starts with new data and its not running with old data. Also, if the run is not reset, the user may run the data set again, but using different model. For details on different estimation models refer to section on Estimation Models in chapter 6. In order to re-initialize, follow directions below:

Open **Run** menu → Click **Reset**

## Chapter 8

# Sample of a Slice Run

This section will show in a step by step how to start a slice run, view and save the results. It is intended to serve as an example on how to run the application. The process requires completing the following steps:

1. Read PTAC data from file
2. Read TTAC data from file
3. Select constraint option (Only for Volume Data Runs)
4. Select constraint source (if necessary, Only for Volume Data Runs)
5. Choose estimation model
6. Verify run header information before start of run
7. Indicate the CSF processing option and start run
8. Choose to automatically view results
9. Save results for future reference
10. Open run result files for analysis

### 8.1 Step 1: Read PTAC Data from File

Open **File** menu and Click on **Read PTAC Data File** as shown in Figure 8.1

A dialog as in Figure 8.2 will appear and you will need to browse for the PTAC files. Select only one of the \*.cpt files and then click the **Open** button. The application will read all of the \*.cpt files in the directory that share the same prefix in their filenames.

Furthermore, the interface will display a plot of the PTAC data on the left axis. In addition, it will display the number of \*.cpt files read and the filename for the first \*.cpt file read as shown in Figure 8.3.

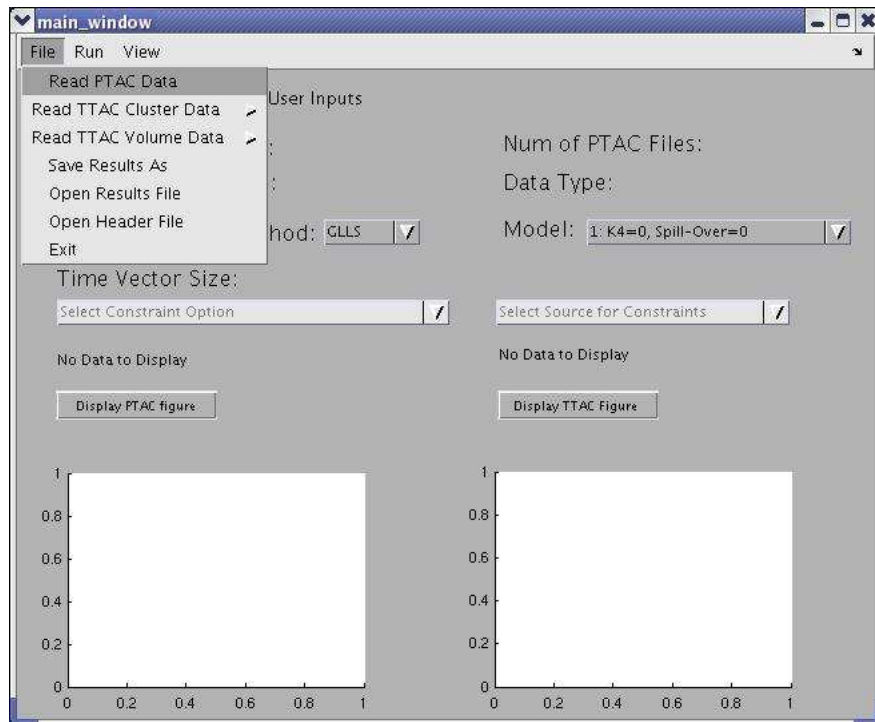


Figure 8.1: Step 1: Open File Menu

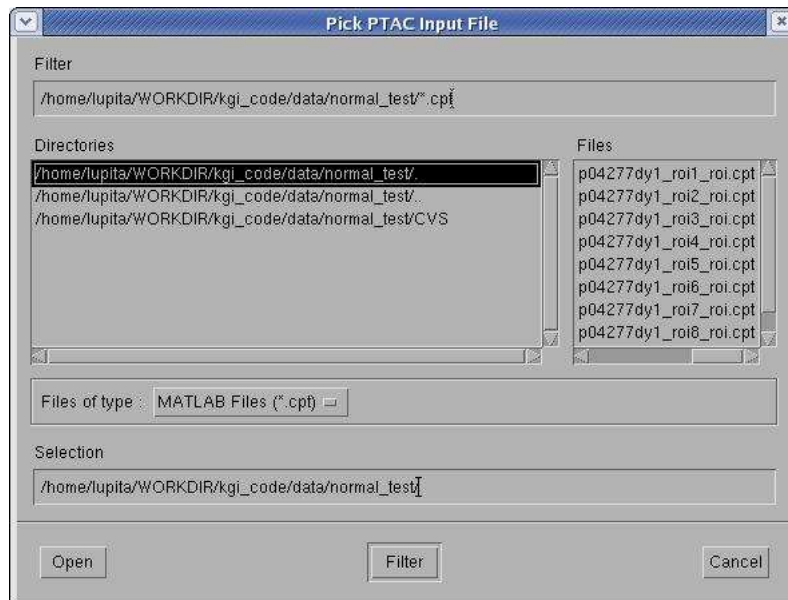


Figure 8.2: Step 1: Browse for PTAC data File

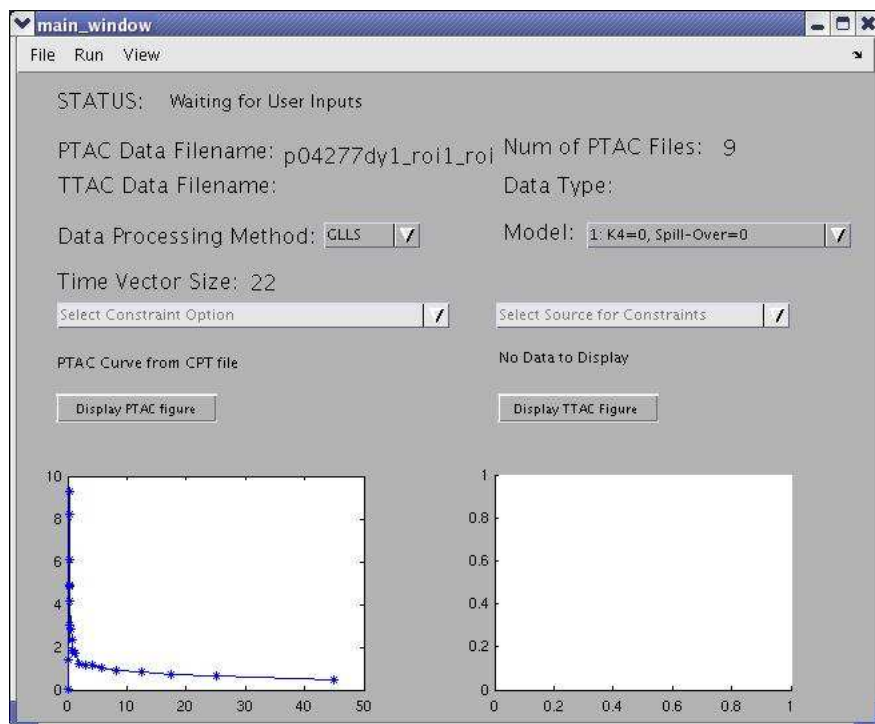


Figure 8.3: Step 1: Interface Displays PTAC Data

## 8.2 Step 2: Read TTAC From File

Open **File** menu, then browse the menu **Read TTAC Volume Data** to **Read Slice** as shown on Figure 8.4.

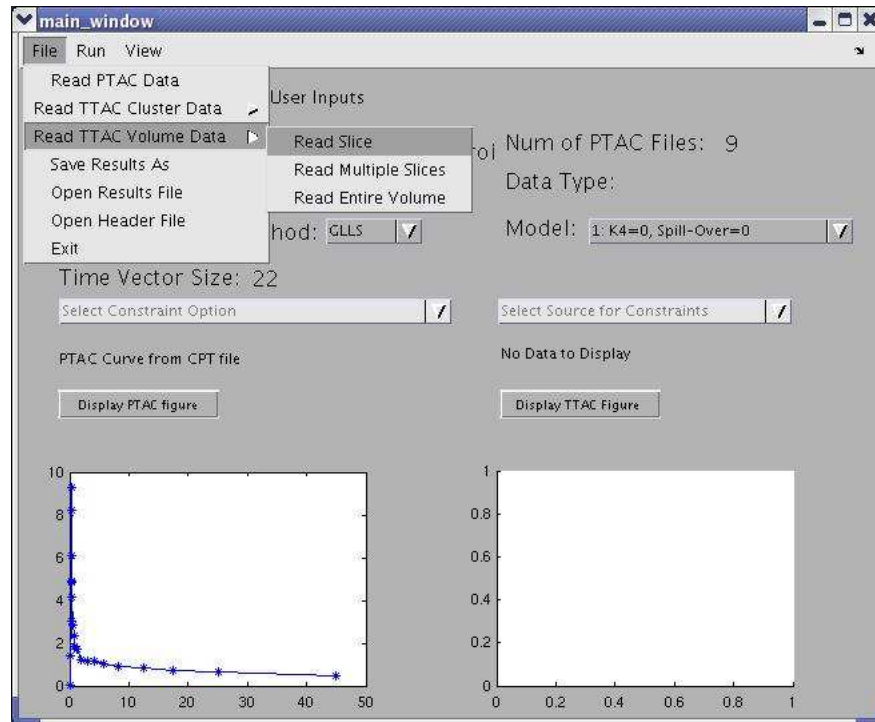


Figure 8.4: Step 2: Open File Menu and Browse Menu to Read Slice

When you click on **Read Slice**, a **Read Options** dialog as in Figure 8.5 will appear. Here you can select whether you want to spatially segment the slice by cropping the slice or if you want to apply the default anisotropic filter to the slice while reading. Check the options you desire and click **OK** button.

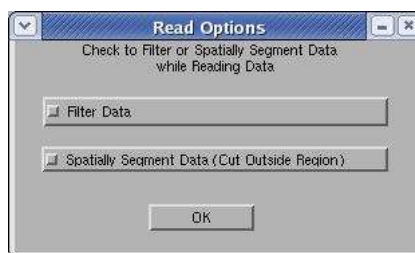


Figure 8.5: Step 2: Dialog to Set Options for Reading TTAC File

A dialog as in Figure 8.6 will appear to prompt for the slice number to read. For example, if we want to read slice 16, enter number 16 and click the **OK** button.

Another dialog will prompt you to browse and select the TTAC file. By default, the dialog filters for a \*.img file as in Figure 8.7. Browse to the directory where the TTAC data file is located, select

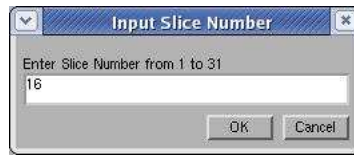


Figure 8.6: Step 2: Dialog to Enter Slice Number

the file and click the **Open** button.

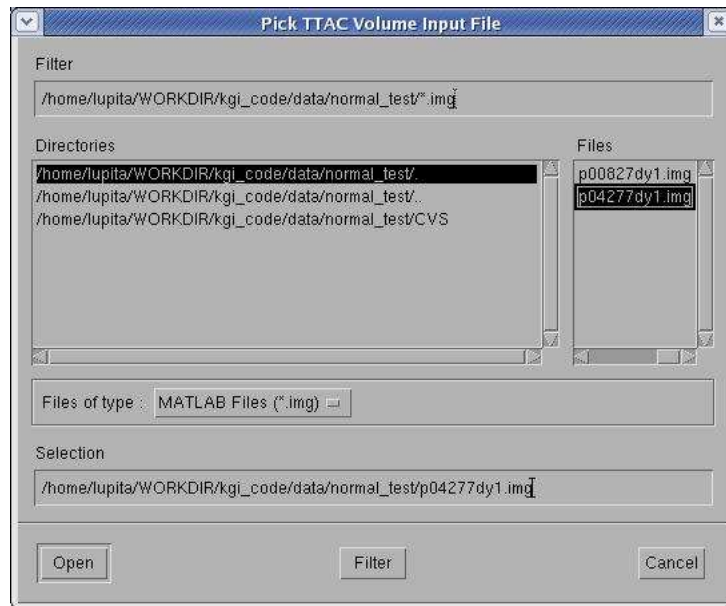


Figure 8.7: Step 2: Browse for TTAC File that Contains Slice

### 8.3 Step 3: Select Constraint Option

This step is done only for volume data runs. Since this is a sample slice run, you will have to select a constraint option for your run, before you can start the run. Select a constraint option through the pop up-menu shown in Figure 8.8. For demonstration purposes select **Run Method with Global Constraints**. Refer to Section 4.4 on constraint option types for a description of all the possible options on the pop-up menu.

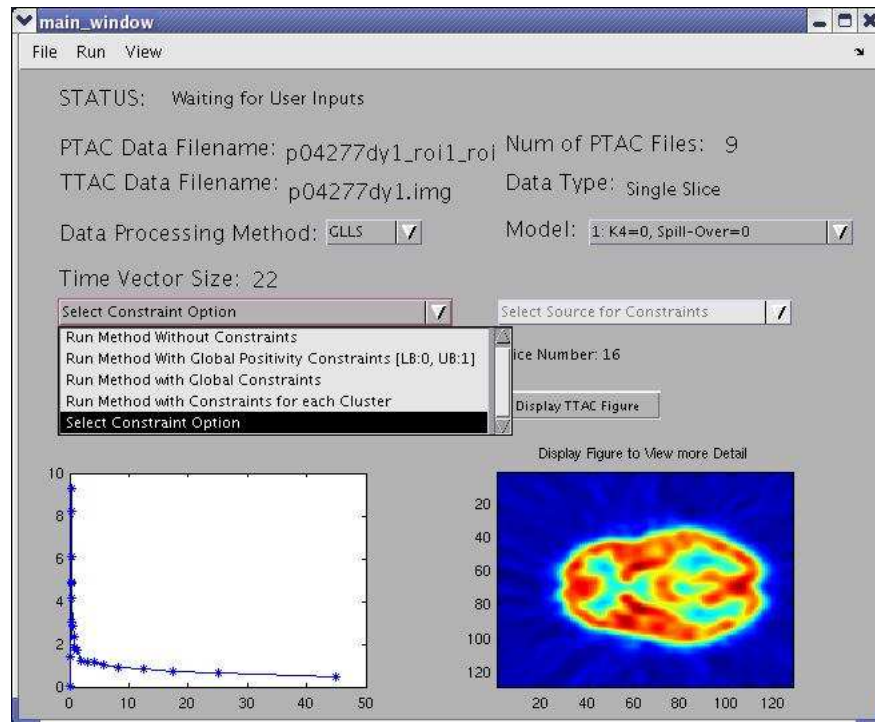


Figure 8.8: Step 3: The Constraint Options Pop up Menu

## 8.4 Step 4: Select Constraint Source (if necessary, Only for Volume Data Runs)

Since you chose to run with global constraints, you will have to select a source for the constraints. Select a constraint source by clicking on the pop up as shown in Figure 8.9. Choose **Calculate Constraints From \*.mat file**.

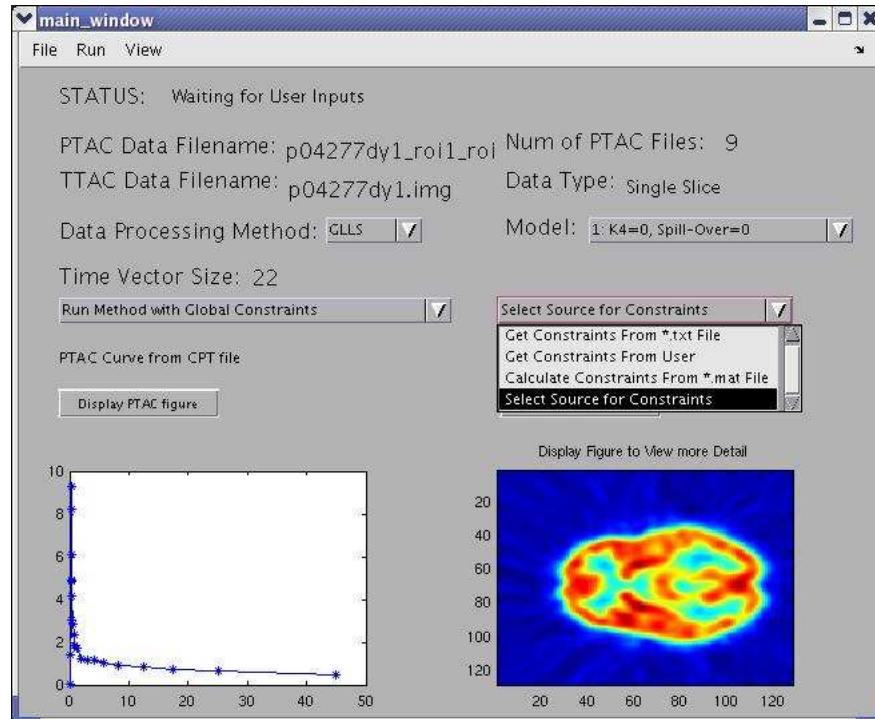


Figure 8.9: Step 4: Constraint Source Pop up Menu

Once you choose the constraint, a browser dialog will appear as shown in Figure 8.10, browse to the location where the \*.mat file is. This \*.mat file should contain the cluster data from which the constraints are going to be calculated. Then, select the \*.mat file and click on the **Open** button. Refer to Section 4.4 on constraints for more information on the other type of sources for constraints.

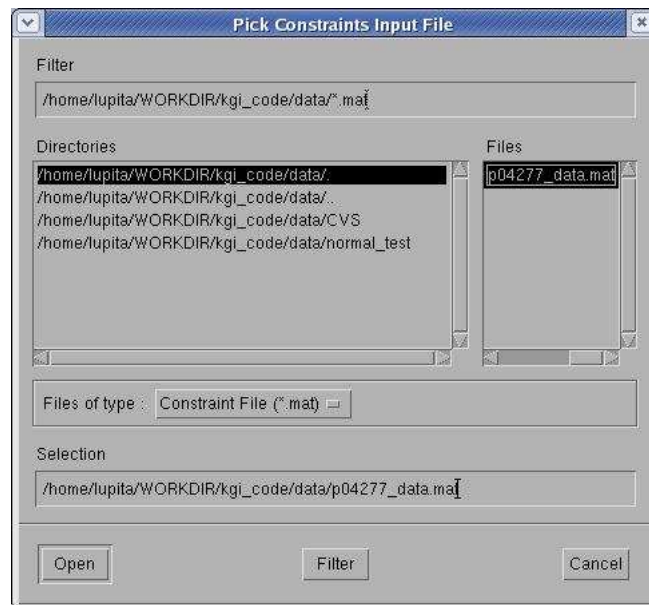


Figure 8.10: Step 4: Browser Dialog to Select Constraint Input File

## 8.5 Step 5: Pick Model to Use

Now, you can pick which model to use by clicking on the Model popup-menu as show in Figure 8.11. Refer to the options section for more information on each model. For demonstration purposes pick Model 1.

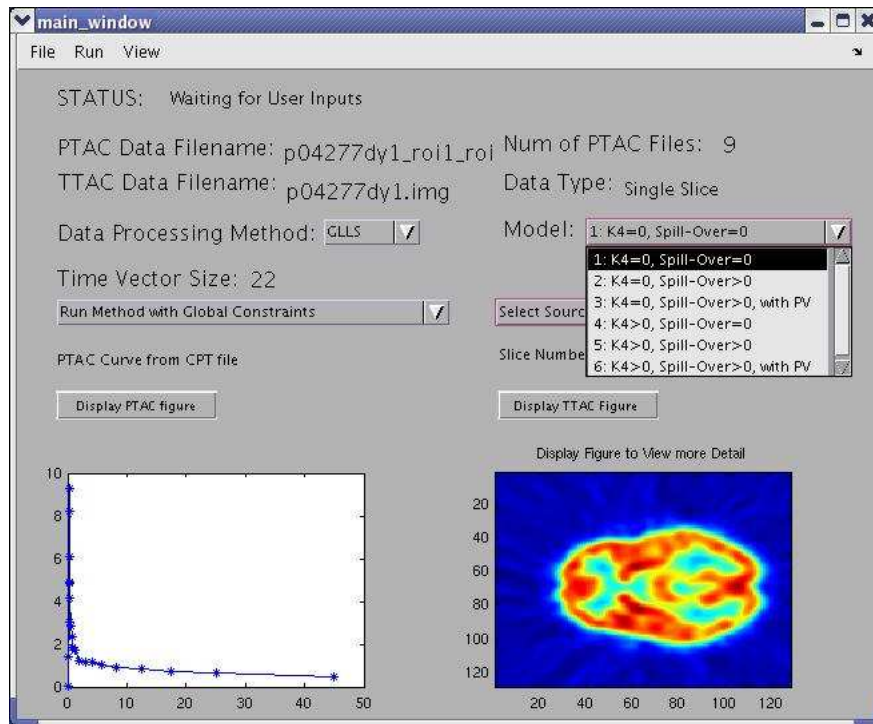


Figure 8.11: Step 5: Pick an Estimation Model

## 8.6 Step 6: Verify Run Header Information Before Start of Run

At this point the Status of the application changes from **Waiting for User Inputs** to **Ready to Run**. Since most users of this application might want to verify they are running the correct data with the correct parameters, it is recommended to view the run header data before running. For more information on what variables will be displayed in the run header data, please refer to Section 5.2. This Chapter gives a detailed description of all the header files for the different types of runs. To display and view the run header information, open the **View** menu and select **View Run Header Data**, as shown in Figure 8.12.

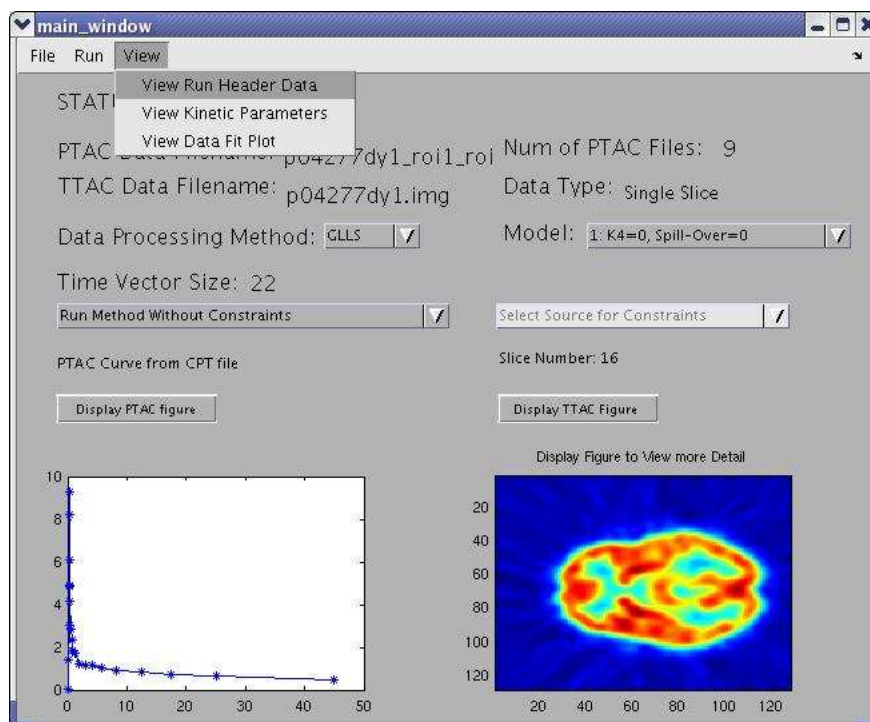


Figure 8.12: Step 6: Open View Menu and View Run Header Data

The display dialog shown on Figure 8.13 will appear. Here you can verify all the run information before starting the run.

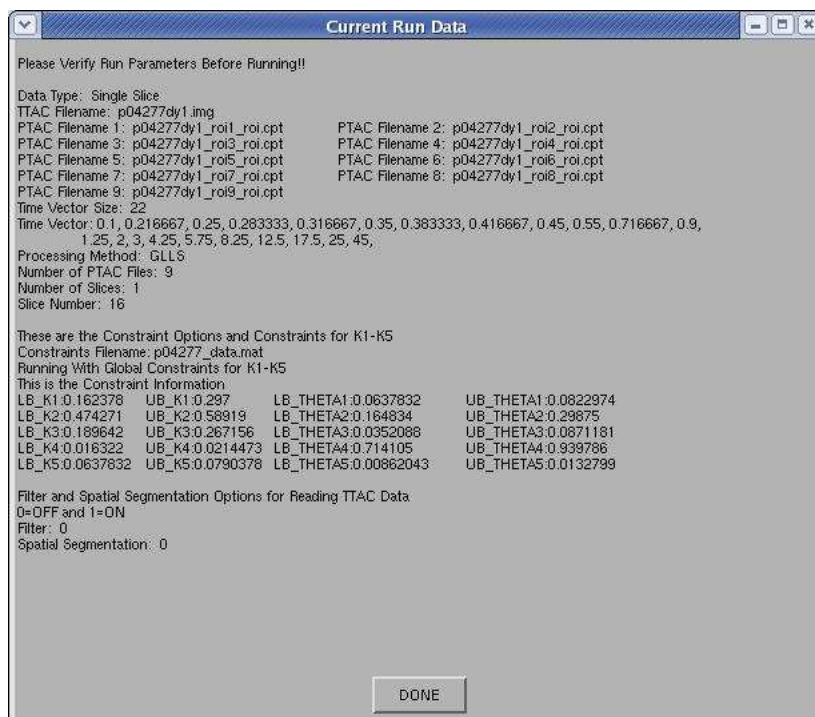


Figure 8.13: Step 6: Display Dialog for Run Header Information

## 8.7 Step 7: Indicate the CSF processing Option and Start Run

After verifying the run header information, start the run by opening the **Run** menu and clicking on **Start Run**, as shown in Figure 8.14.

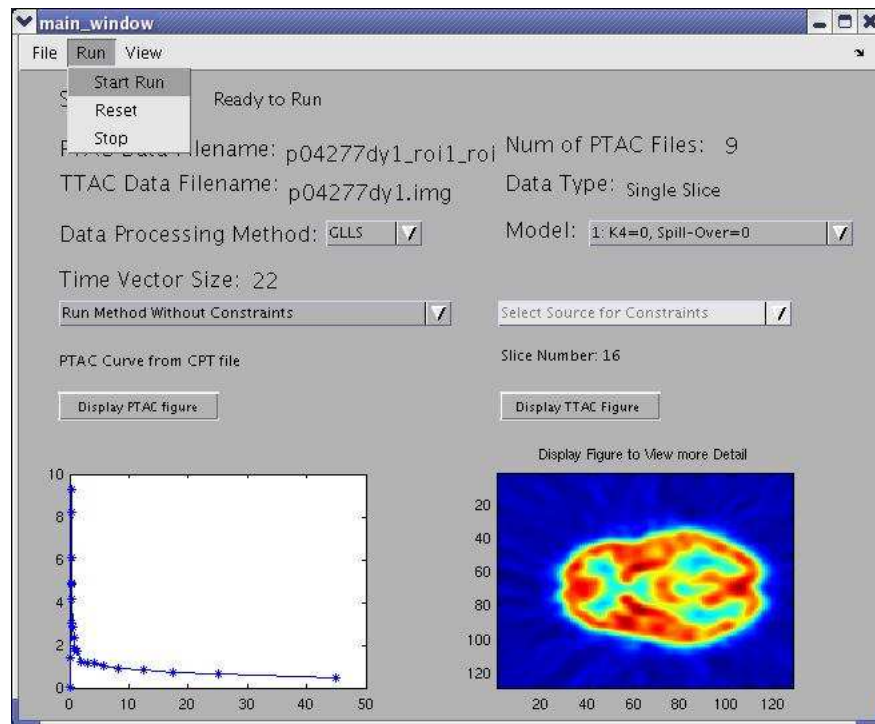


Figure 8.14: Step 7: Run Menu

When you click on **Start Run**, a question dialog as shown in Figure 8.15 will prompt you to whether you want to process CSF or not. For demonstration purposes press the **No** button, and the run will start.



Figure 8.15: Step 7: Prompt for CSF Option

Since ®MATLAB does not support multi-tasking, you will not be able to interact with the user interface until the run is finished.

## 8.8 Step 8: View Results Automatically

At the end of run, you will see a prompt to view results as shown in Figure 8.16. Click on **Yes** to view results at the end of run.



Figure 8.16: Step 8: End of Run Dialog To Automatically View Results

Then, you will be prompted to choose whether you want to scale the results to expected values as in Figure 8.17. This is useful, since large values tend do not display properly due to the limited resolution of the colormap. Click on **Yes**.



Figure 8.17: Step 8: Prompt to Scale Results for Display

Then, you will be prompted to choose whether you want to mark negative pixels with a magenta color as shown in Figure 8.18. Click on **Yes**.

Note: If you chose **No**, you might still get some magenta pixels since pixels that do not give accurate results are automatically marked.



Figure 8.18: Step 8: Prompt to Mark Negative Pixels with Magenta

At this point you will see results for Slice 16.

Notice that if you were running multiple slices, then you would have to select which results you want to view. Therefore, after you choose to view results, you will be prompted to pick which slice results to want to view. This dialog will look like the one in Figure 8.19. This dialog will contain all the slices available. You will be able to select multiple slices by pressing the Ctrl Key while clicking on the slice numbers in the listbox.

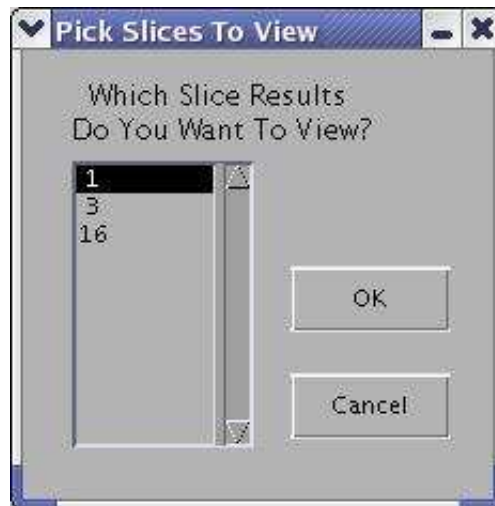


Figure 8.19: Step 8: Pick Slices to View

## 8.9 Step 9: Save Results

When the run is finished, the interface will prompt you to save your results and reset the application. Before you reset the application you have to view and save your results. Open **File** menu and click on **Save Results** as shown on Figure 8.20.

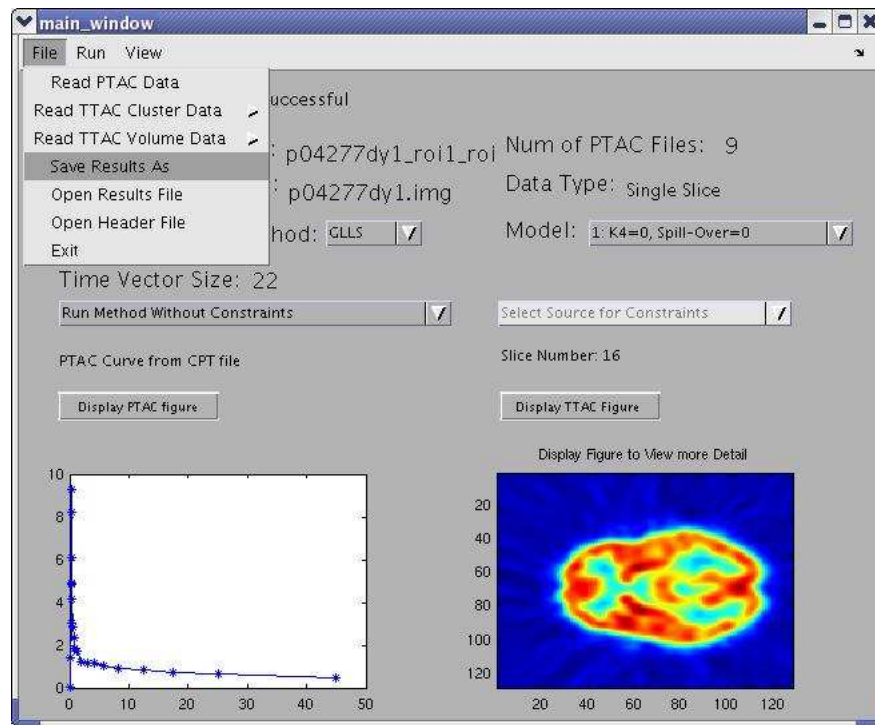


Figure 8.20: Step 9: Open File Menu to Save Results

Subsequently, a dialog will appear prompting for a filename prefix for the results file as shown in Figure 8.21. Refer to the Chapter 5 for more information on the naming convention for the result files that are going to be written. Type **slice\_run\_example** and click on **Save**. The result files had been saved. Now the next step is to open and view the result files created for by run.

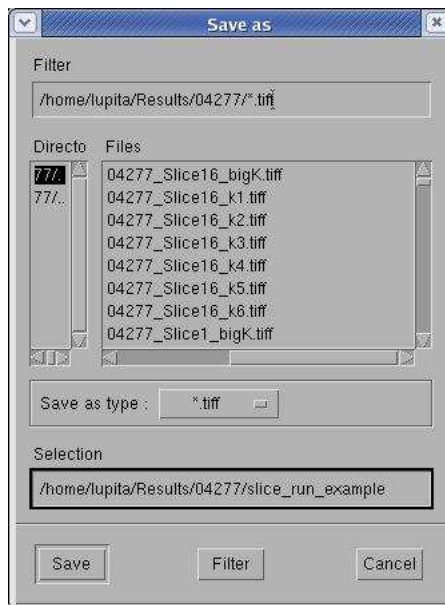


Figure 8.21: Step 9: Dialog to Enter Filename Prefix to Save Results

## 8.10 Step 10: View Result Files and Header File

Open the **File** menu and Click on **Open Results File** as shown in Figure 8.22.

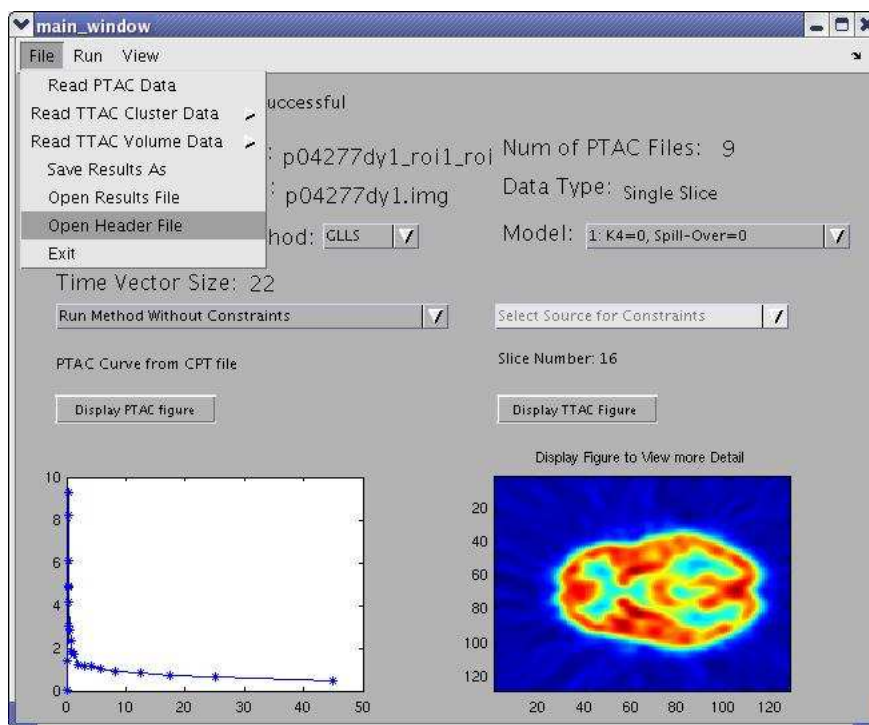


Figure 8.22: Step 10: Open File Menu to View Past Runs Header and Result Files

The PTAC filename used in this example is **p04277dy1\_roi1\_roi.cpt**, therefore the directory where the result files are saved by default is **\$HOME/Results/04277**. We browse to this directory and select file **slice\_run\_example\_results.txt** and push **Open** button as shown in Figure 8.23.

The file is opened and displayed on a window see Figure 8.24. For slice runs, this file contains the filenames of the files to which the kinetic parameter results were written. Kinetic parameter results were written to several \*.mat files. Note the filenames displayed and push the **Done** button to close this window.

Now, we would like to view these \*.mat files. Open the **File** menu and Click on **Open Results File** as shown in Figure 8.22. Browse to the directory where the result files are. Use the pop-up menu on the browser dialog to change the file type to view only desired result files as in Figure 8.25.

For demonstration purposes pick \*k2.mat, so you will only see the kinetic parameter  $k_2$  \*.mat files saved from the past runs as shown in Figure 8.26. Select file **slice\_run\_example.Slice16\_k2.mat** and push **Open** button.

A window opens up and displays the  $k_2$  kinetic parameter as an image see Figure 8.27. You can select and view any of the other kinetic parameters in the same manner.

In addition, you can also open the header file of this run. Opening header files will allow you to go back and review what PTAC, TTAC data and other run time parameters produced the result files. In order to open the past run header file, open the **File** menu and click on menu item **Open Header File**. A directory browser will appear and you need to browse to the directory where the results were saved see Figure 8.28. In this case the results were saved at **\$HOME/Results/04277**.

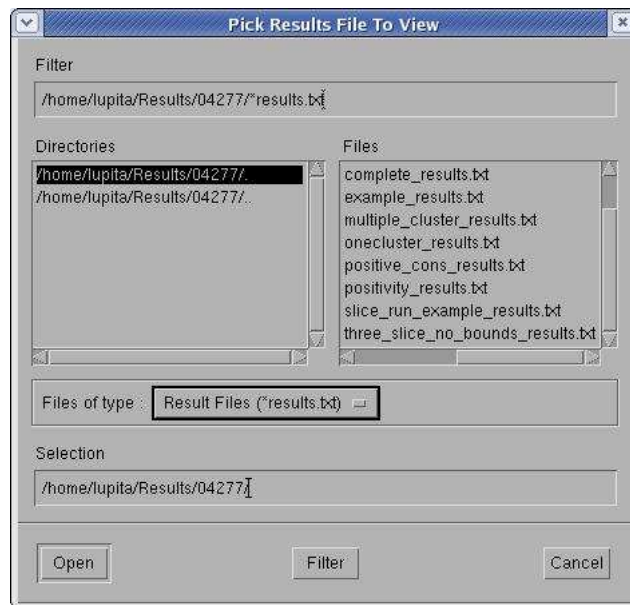


Figure 8.23: Step 10: Select Result File to View

Select header file **slice\_run\_example\_header.txt** and push **Open** button.

Then, a window displaying the header file will appear as shown in Figure 8.29.

## 8.11 Step 11: Reset Application

Now, Reset the application to setup and start a new run. Open the **Run** menu and click the **Reset** menu item, see Figure 8.30. The application will reinitialize and all variables will be cleared.

All of these steps for a volume run are illustrated in the flow chart next page, see Figure 8.31.

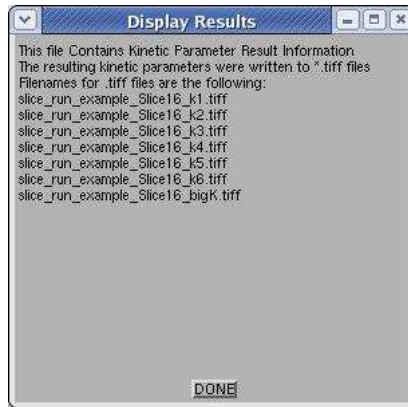


Figure 8.24: Step 10: slice\_run\_example\_results.txt

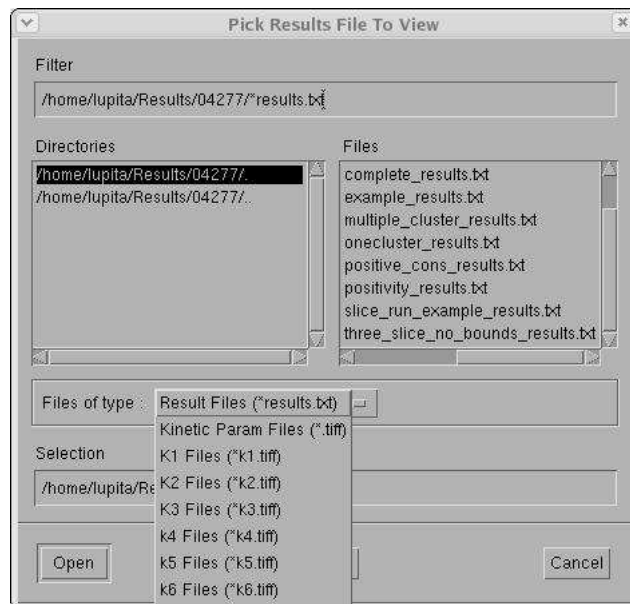


Figure 8.25: Step 10: Change the File Type to \*.mat

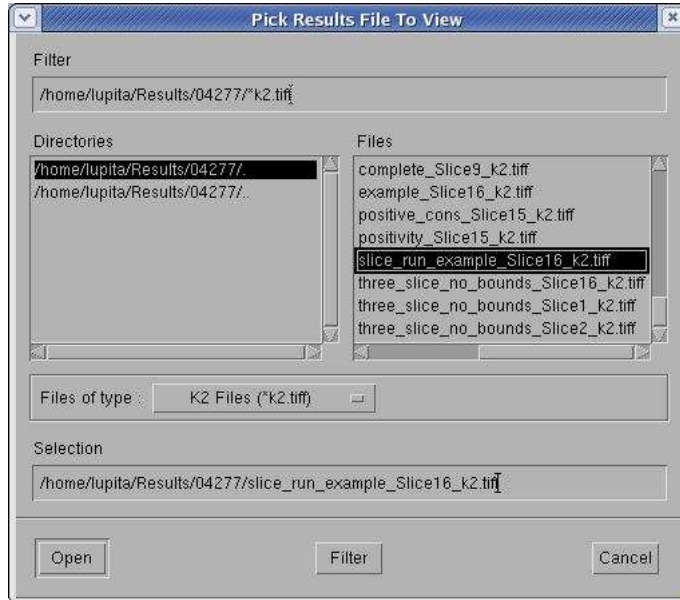


Figure 8.26: Step 10: Select Results File to View

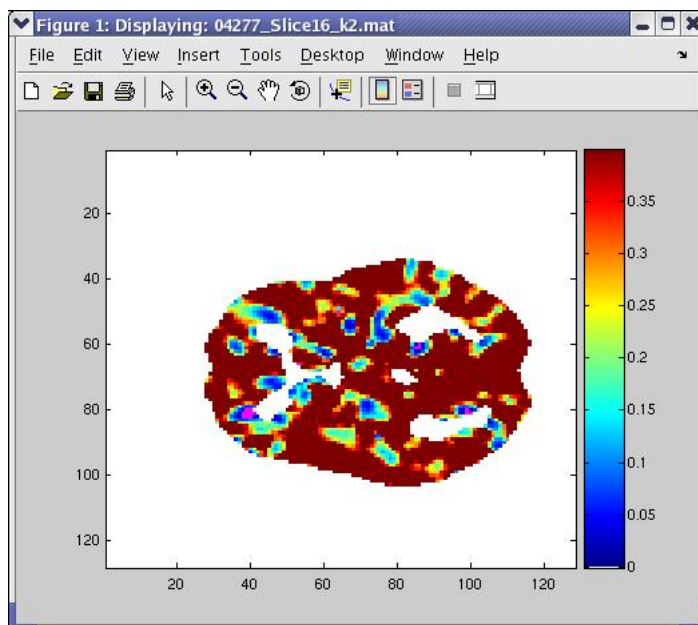


Figure 8.27: Step 10: Display window for  $k_2$  \*.mat File

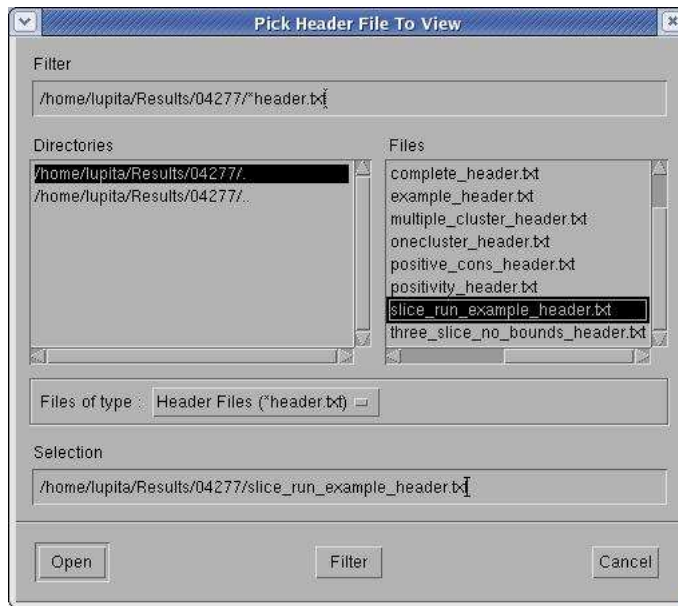


Figure 8.28: Step 10: Select Header File to View

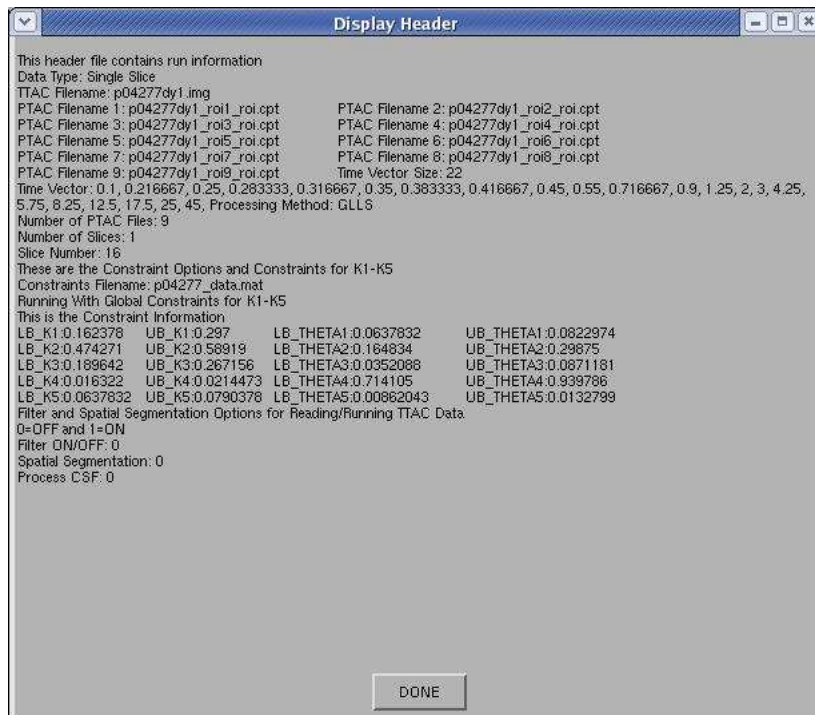


Figure 8.29: Step 10: Header File Displayed After Run

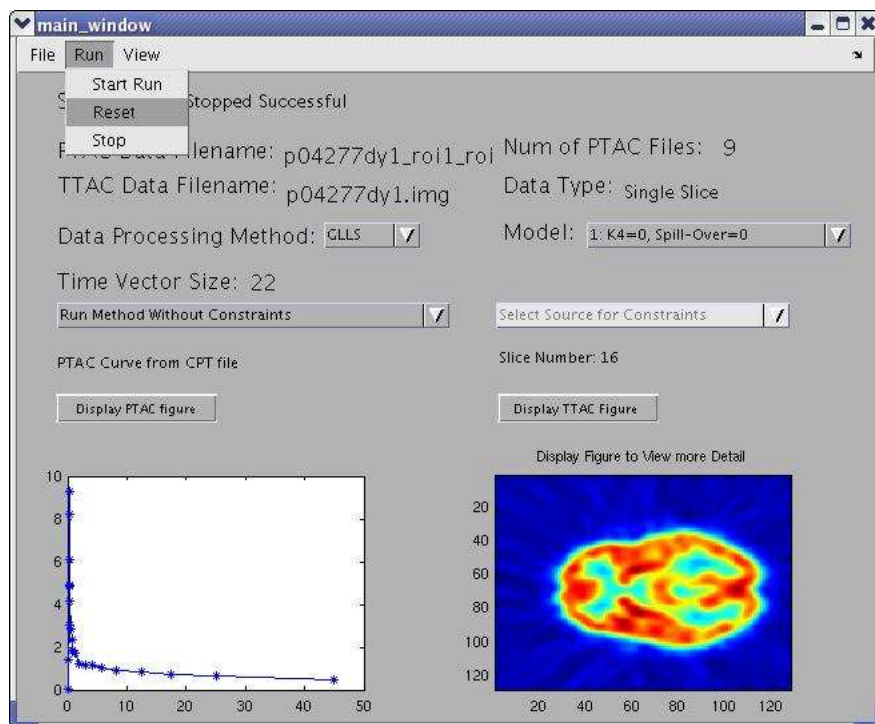


Figure 8.30: Step 11: Reset To Start New Run

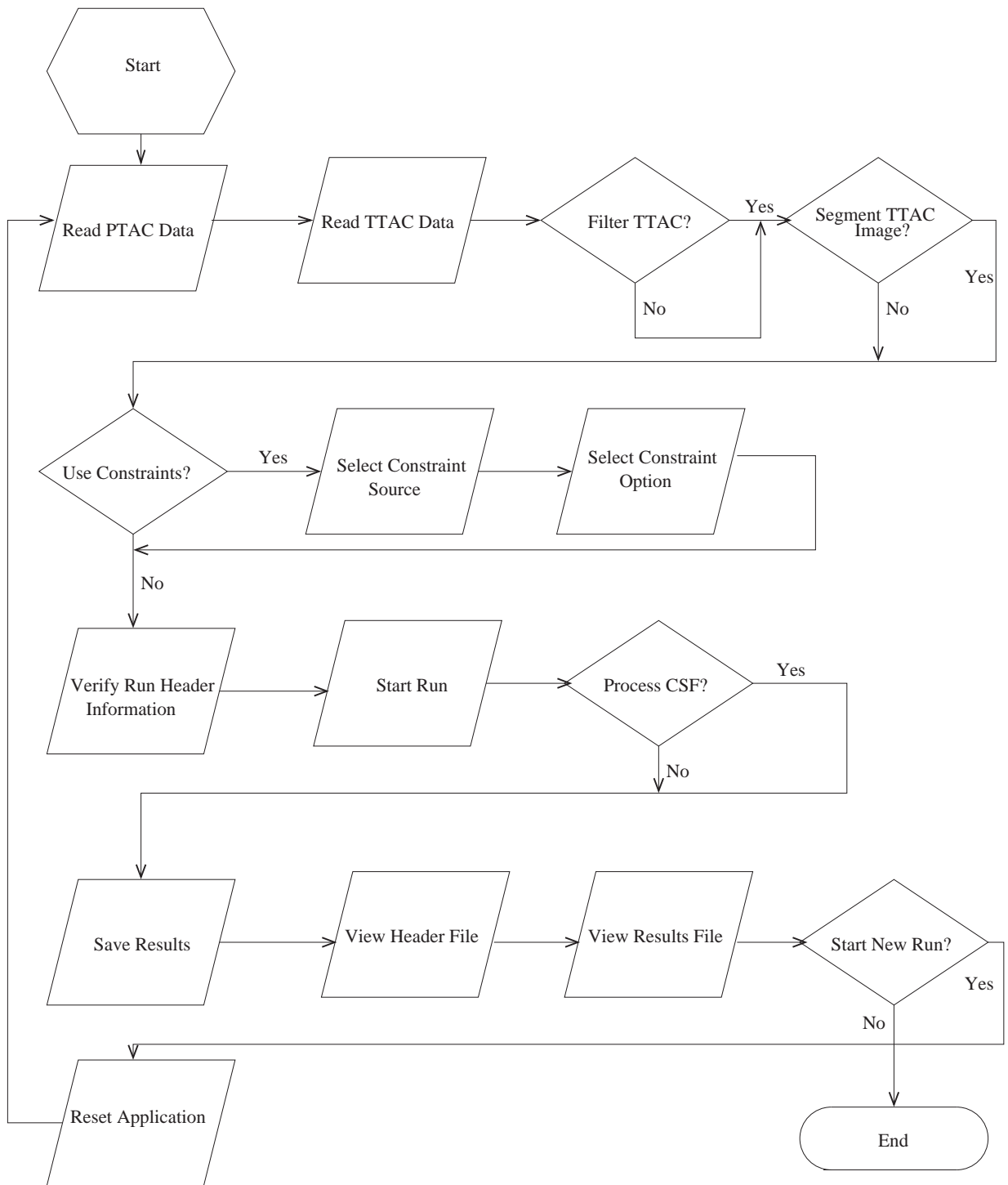


Figure 8.31: The Volume Run Flow Chart

Also, the steps for cluster run are illustrated in Figure 8.32.

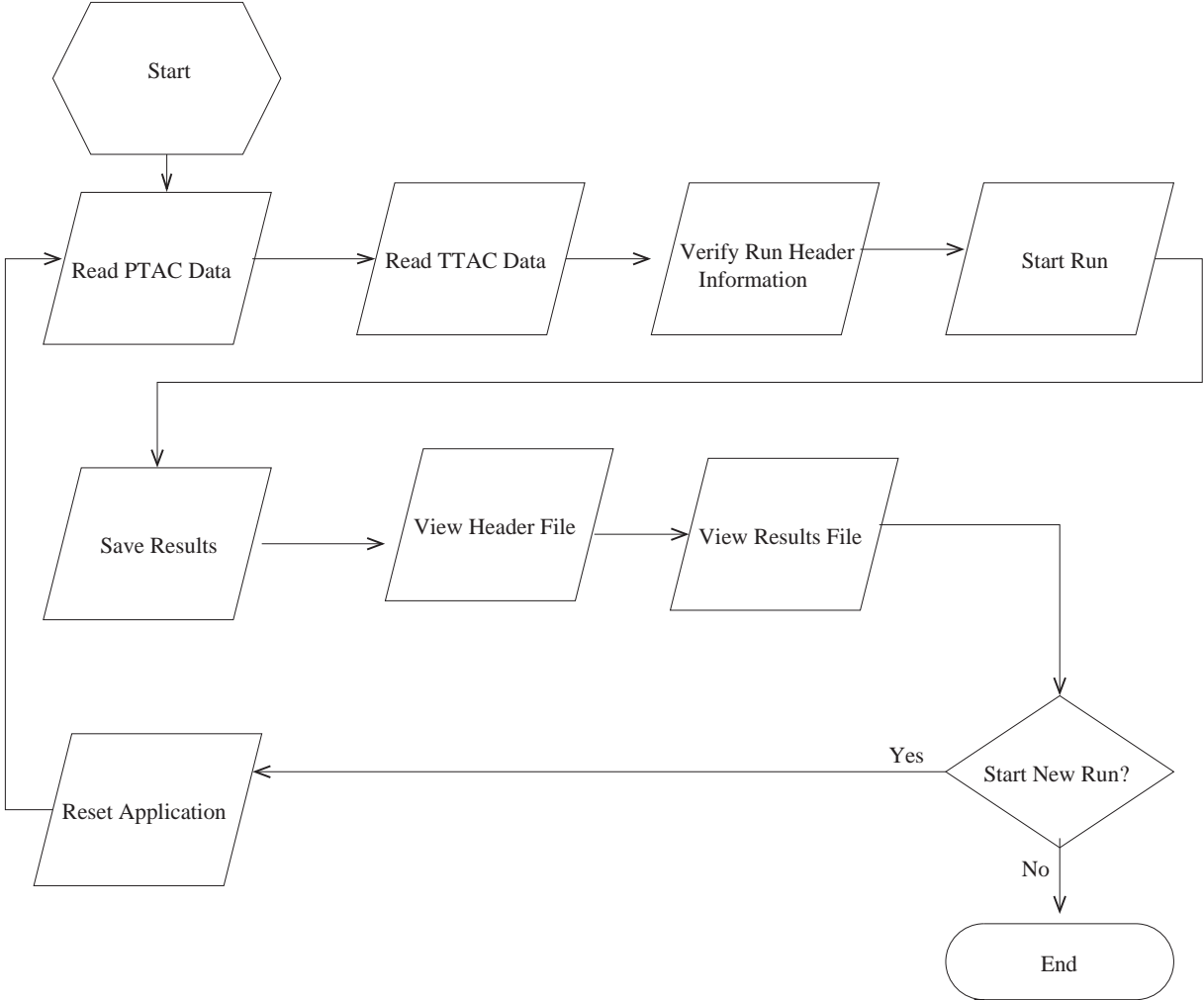


Figure 8.32: The Cluster Run Flow Chart

## Chapter 9

# Future Implementation

Currently, this application enables the user to apply several constraint options while estimating the kinetic parameters. On the other hand, this application is constrained to GLLS as its only estimation method for the kinetic parameters. Also, the application implements and applies only the anisotropic diffusion filter. It is intended that these tool could also be used as an evaluation tool for the different estimation methods and filters. Therefore, there is going to be future implementation of optional methods for kinetic parameter estimation.

## Chapter 10

# Results Section

The following are the results obtained from a slice run. The following results are for slice 16 assuming model 1 and constraints. For model 1 we do not get  $k_4$  and  $k_5$  results, since we assume that  $Spill - Over = 0$  and  $k_4 = 0$ . Note some pixels are magenta. Pixels that did not give accurate results due to the early noise in the input data were marked magenta. Results were also scaled in the range of expected values, otherwise colormap would be skewed.

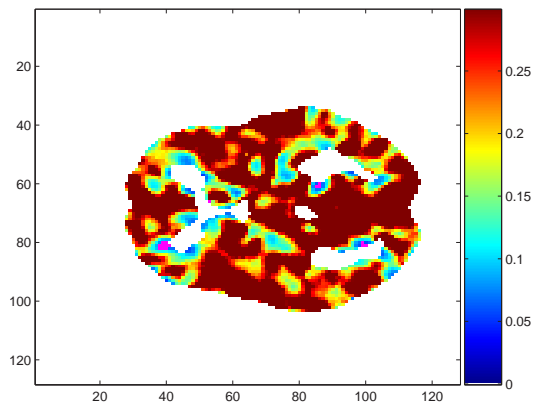


Figure 10.1: Parameter  $k_1$  for Slice 16, assuming  $k_4 = 0$ , spillover = 0, and No Constraints

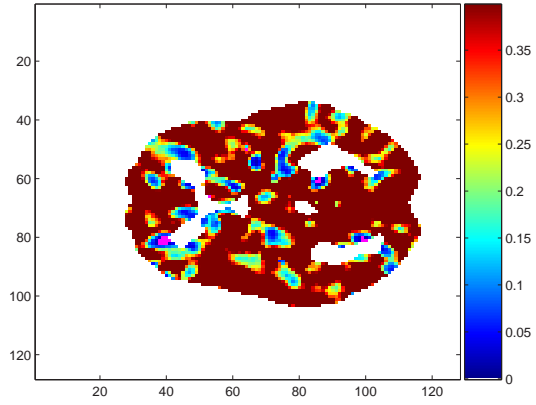


Figure 10.2: Parameter  $k_2$  for Slice 16, assuming  $k_4 = 0$ , spillover = 0, and No Constraints

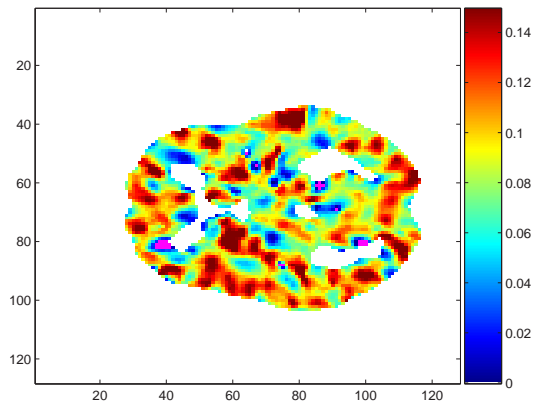


Figure 10.3: Parameter  $k_3$  for Slice 16, assuming  $k_4 = 0$ , spillover = 0, and No Constraints

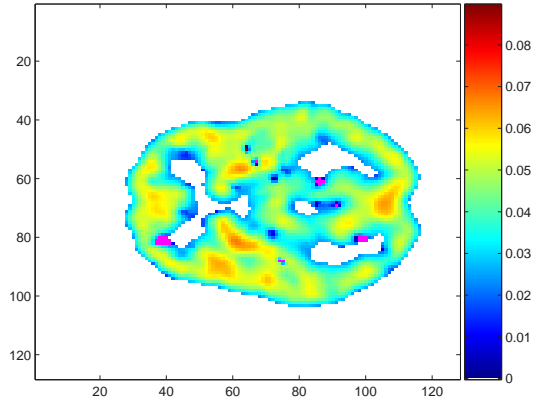


Figure 10.4: Parameter  $k_6$  for Slice 16, assuming  $k_4 = 0$ , spillover = 0, and No Constraints

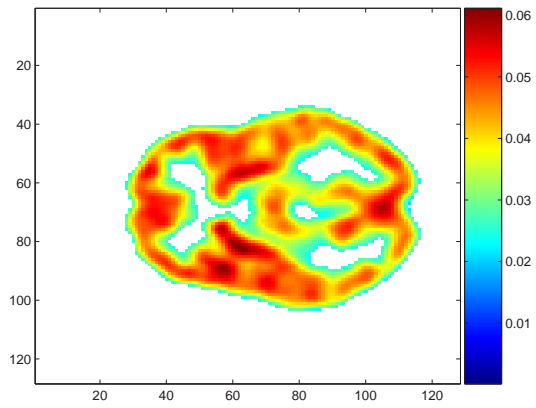


Figure 10.5: Parameter  $\mathbf{K}$  for Slice 16, assuming  $k_4 = 0$ , spillover = 0, and No Constraints

## Chapter 11

# Installation, Execution, and Developers Guide

This chapter is intended to describe the installation procedures, as well as how to start the application. The developers guide section aims to aid any developer in modifying the code for customization. Also, it helps the user understand where critical files for the application are located, as well as sample data.

### 11.1 Downloading and Installing Application

These are the installation procedures when downloading the code from the published web site.

NOTE: Remember you can only install and run this application on a Linux System with ®MATLAB 6.0 or higher installed.

1. Click on the CODE link.
2. Choose Save
3. Browse to the directory where you want to install the application, and save the file.
4. Browse to the directory where the file item
5. Uncompress file by typing `gunzip kgi_code.tar.gz`
6. Extract kgi\_code folder contents by typing `tar -xvf kgi_code.tar`
7. Now you are ready to run the application.

### 11.2 Execution of Application

The kgi\_code directory will contain several subdirectories. I will give a descriptions of their contents in the Developer's Guide. For execution it is required that you add the kgi\_code directory to the ®MATLAB path. In order to run the application you have to do the following:

1. Start  $\text{\textcircled{R}}$ MATLAB
2. Go to the File Menu and Click on *Set Path*
3. Browse to the `kgi_code` directory and click on *Add With SubFolders* button.
4. Save path if possible and Close Path Window. Note: You need to set path every time you restart  $\text{\textcircled{R}}$ MATLAB unless user saves path.
5. Now, Browse to the GUI directory under the `kgi_code` directory.
6. Open `main_window.m` file under the GUI directory
7. Click on *Run* to start the Application

### 11.3 Developer's Guide

This section has a brief description of the different directories within `kgi_code`. These directories represent several modules. This is very useful if you want to modify the code, and if you need to know where critical or sample data files.

1. Computation Files: This directory contains all the core code for the estimation algorithm GLLS.
2. data: This directory contains all the sample data files for PTAC, TTAC, Global Constraints.
3. Documentation: This directory contains documentation on the program flow.
4. GetInputDataFiles: This directory has files for the code to parse and process the Input TTAC and PTAC data files.
5. GUI: This directory contains all the files and figures for the application user interface, including all the Callbacks.
6. IOFiles: This directory contains all of the printing and formatting code in order to save results.
7. ObsoleteFiles: This directory is not in use, but has part of the initial code and scripts testing parts of the code.

## Chapter 12

# Notation and Acronyms

$k_1$	Transport rate from blood to extra-vascular space (1)
$k_2$	Transport rate back from extra-vascular space to blood (1)
$k_3$	Phosphorylation rate intra-cellular FDG to FDG-6-phosphate (1)
$k_4$	Dephosphorylation rate intra-cellular FDG-6-phosphate to FDG (1)
$k_5$	Spill-over from blood to tissue coefficient (1)
$k_6$	$(k_1 * k_3)/(k_2 + k_3)$ proportional to local cerebral metabolic rate of glucose (1)
<b>K</b>	Analog to $k_6$ computed using PATLAK analysis (1)
PTAC	Plasma Time Activity Curve, the FDG concentration in the blood over time (6)
TTAC	Tissue Time Activity Curve, the FDG concentration in the tissue over time. (6)
GLLS	Generalized Linear Least Squares, computational method for parameter estimation (6)
PET	Positron Emission Tomography (7)
AD	Alzheimer's disease (7)
FDG	Flouro-Deoxy-Glucose (7)
LCMR	Local Cerebral Metabolic Rate (7)
$y_1(t)$	The FDG concentration in tissue over time (7)
$y_2(t)$	The phosphorylated FDG concentration in tissue over time (7)
$y(t)$	The sum of phosphorylated and non-phosphorylated FDG concentration over time (7)
$u(t)$	Input data, namely the FDG tracer concentration in plasma over time (8)
PVE	Partial Volume Effects (8)
$J$	Length of slice in unit pixels (11)
$L$	Width of slice in unit pixels (11)
$N$	Number of time points (11)
$S$	Number of slices in the entire volume (11)
\$PATIENT_ID	Numerical string parsed from PTAC filename (11)
$C$	The number of clusters (12)
$F$	Filename for saving results entered in <b>Save As</b> dialog (17)
$S$	Slice number (18)
$H$	Kinetic Parameter Number in the range 1-6 (18)
CSF	Cerebral Spinal Fluid (21)
\$HOME	The current user home directory (25)

# Bibliography

- [1] J.A.D. Aston, V.J. Cunningham, M-C. Asselin, A. Hammers, A.C. Evans, and R.N. Gunn, *Positron emission tomography partial volume correction: Estimation and algorithms*, *J. Cerebral Flow Metab.* (2002), 1019–1034.
- [2] D. Barber, *The use of principal components in the quantitative analysis of gamma camera dynamic studies*, *Phys. Med. Biol.* (1980), 283–292.
- [3] R. Blasberg C. Patlak and J. Fenstermacher, *Graphical evaluation of blood to brain transfer constants from multiple uptake data*, *J. Cereb. Blood Flow and Metab.* (1983), 1–7.
- [4] K. Chen, D. Bandy, E. Reiman, S.C. Huang, D. Feng M. Lawson, L. Yun, and A. Palant, *Non-invasive quantification of the cerebral metabolic rate for glucose using positron emission tomography, F-Fluoro-2-Deoxyglucose the patlak method and an image-derived input function*, *J. Cerebral Flow Metab.* (1998), 716–723.
- [5] K. Chen, M. Lawson, E. Reiman, A. Cooper, S.C. Huang D. Feng, D. Bandy, D. Ho, L. Yen, and A. Palant, *Generalized linear least squares method for fast generation of myocardial blood flow parametric images with N-13 ammonia PET*, *IEEE Tran. Med. Imag.* **17** (1998), no. 2, 236–243.
- [6] S.C. Huang D. Feng, Z. Wang, and D. Ho, *An unbiased parametric imaging algorithm for nonuniformly sampled biomedical system parameter estimation*, *IEEE Trans. Medical Imaging* **15** (1996), no. 4, 512–518.
- [7] X. Li D. Feng and S.C.Huang, *A new double modeling approach for dynamic cardiac PET studies using noise and spillover contaminated lv measurements*, *IEEE Trans. Biomed. Eng.* (1996), 319–327.
- [8] S. Eberl, A. R. Anayat, R. R. Fulton, P. K. Hooper, and M. J. Fulham, *Evaluation of two population based input functions for quantitative neurological FDG PET studies*, *Eur. J. Nucl. Med.* **24** (1997), 299–304.
- [9] D. Feng and D. Ho, *Parametric imaging algorithms for multicompartmental models dynamic studies with positron emission tomography*, *Quantification of Brain Function* (1993), 1–10.
- [10] D. G. Feng, K.-P. Wong, C.-M. Wu, and W.-C. Siu, *A technique for extracting physiological parameters and the required input function simultaneously from PET image measurements: Theory and simulation study*, *IEEE Trans. Inform. Technol. Biomed.* **1** (1997), no. 4, 243–254.
- [11] S. Gambhir, *Quantification of the physical factors affecting the tracer kinetic modeling of cardiac positron emission tomography data*, Ph.D. thesis, UCLA, 1990.

- [12] Hongbin Guo, Rosemary Renaut, and Kewei Chen, *Improved imaged-derived input function for study of human brain FDG-PET*, submitted, 2004.
- [13] K. Herholz and C.S. Patlak, *The influence of tissue heterogeneity on results of fitting nonlinear model equations to regional tracer uptake curves with an application to compartmental models used in positron emission tomography*, *IEEE Trans. Med. Imag.* (2000), 1128–1131.
- [14] S.C. Huang, M. Phelps, E. Hoffman, K. Sideris, C. Selin, and D. Kuhl, *Non-invasive determination of local cerebral metabolic rate of glucose in man*, *Am. J. Physiology* (1980), E69–E83.
- [15] C. Negoita, *Global kinetic imaging using dynamic positron tomography data*, Ph.D. thesis, Arizona State University, 2003.
- [16] Cristina Negoita and Rosemary A Renaut, *On the convergence of the generalized linear least squares algorithm*, *BIT* (2004), accepted.
- [17] P. Perona and J. Malik, *Scale-space and edge detection using anisotropic diffusion*, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **12** (1990), 629–639.
- [18] M. Phelps, S.C. Huang, E. Hoffman, C. Selin, L. Sokoloff, and D. Kuhl, *Tomographic measurement of local cerebral glucose metabolic rate in humans with (F-18)2-Fluoro-2-Deoxy-D-Glucose: Validation of method*, *Ann. Neurol* **6** (1979), 371–388.
- [19] S.C. Huang R. Carson and M. Green, *Weighted integration method for local cerebral blood flow measurements with positron emission tomography*, *Journal Cerebral Flow Metab.* (1986), 245–258.
- [20] S. M. Sanabria-Bohorquez, A. Maes, P. Dupont, G. Bormans, T. de Groot, A. Coimbra, W. Eng, T. Laethem, I. De Lepeleire, J. Gambale, J. M. Vega, and H. D. Burns, *Image-derived input function for [<sup>11</sup>C] flumazenil kinetic analysis in human brain*, *Mol. Imag. Biol.* **5** (2003), no. 2, 72–78.
- [21] K. Schmidt, G. Mies, and L. Sokoloff, *Model of kinetic behavior of deoxyglucose in heterogeneous tissue in brain: a reinterpretation of the significance of parameters fitted to homogeneous tissue models*, *J. Cereb. Blood Flow Metab.* (1991), 10–24.
- [22] K. Uemura, H. Toyama, Y. Ikoma, K. Oda, M. Senda, and A. Uchiyama, *Correction of partial volume effect on rate constant estimation in compartment model analysis of dynamic PET study*, *J. Cereb. Blood Flow Metab.* (1991), 10–24.
- [23] I. Weinberg, S.C Huang, E. Hoffman, L. Araujo, C. Nienaber, M. GroverMcKay, M. Dahlbom, and H. Schelbert, *Validation of PET-acquired input functions for cardiac studies*, *Journal Nuclear Med.* **29** (1988), 241–247.

# Appendix A

## GLLS Note

### Notes on GLLS

#### Introduction

The generalized linear least squares (GLLS) algorithm was briefly introduced in [5] for the solution of a biomedical inverse problem. A more-detailed presentation of the algorithm followed in [8]. GLLS is an extension of the generalized least squares (GLS) algorithm to non-uniformly sampled data [8]. A careful discussion of the motivation for the introduction of GLLS is also provided in [8].

#### Basic GLLS

We provide here a very brief overview of the derivation of the GLLS algorithm for the general system of equations

$$\frac{d^n y}{dt^n} - \sum_{j=1}^n \alpha_j \frac{d^{n-j} y}{dt^{n-j}} = \sum_{j=0}^n \beta_j \frac{d^{n-j} u}{dt^{n-j}}, \quad (\text{A.1})$$

where we note that the right hand side term starts from  $j = 0$ , which is more general than the original version of the algorithm presented in [5], but is the form required when  $u(t)$  enters into the original system of differential equations, as is the case when spillover is included in the model, [3]. Here  $y = y(t)$  is the system output,  $u = u(t)$  is the system input, coefficients  $\alpha_1, \dots, \alpha_n, \beta_0, \dots, \beta_n$  are the system transfer parameters, and  $\frac{d^j}{dt^j}$  indicates the  $j$ th order time derivative operator. For the moment we do not consider the relationship of these parameters with the kinetic rate constants relevant to tracer dynamics, which is not relevant for the basic GLLS derivation. In the context of the derivation of the GLLS, it is standard to derive  $y(t)$  as a function of  $u(t)$  using the Laplace transform approach. Let  $Y(s)$  and  $U(s)$  be Laplace transforms of  $y(t)$  and  $u(t)$ , respectively, and introduce the polynomials

$$\begin{aligned} A(s) &= s^n - \sum_{j=1}^n \alpha_j s^{n-j}, \\ B(s) &= \sum_{j=0}^n \beta_j s^{n-j}. \end{aligned} \quad (\text{A.2})$$

Then assuming that all initial conditions on  $u(t)$ ,  $y(t)$  and their derivatives in time are identically zero, yields the Laplace domain equation

$$A(s)Y(s) = B(s)U(s). \quad (\text{A.3})$$

### The LS solution

For (A.3) the equivalent integral equation in the time domain is

$$y(t) = \sum_{j=1}^n \alpha_j I_j(y) + \beta_0 u(t) + \sum_{j=1}^n \beta_j I_j(u), \quad (\text{A.4})$$

using for any function  $g(t)$  the notation

$$I_1(g) = \int_0^t g(\tau) d\tau, \quad I_j(g) = \int_0^t \int_0^{\tau_1} \dots \int_0^{\tau_{j-1}} g(\tau) d\tau d\tau_{j-1} \dots d\tau_1, \quad j \geq 2.$$

Sampling (A.4) in time at  $m$  points,  $m > 2n$ , provides the overdetermined linear system

$$\mathbf{y} = X\theta \quad (\text{A.5})$$

where  $\mathbf{y} = [y_1, y_2, \dots, y_m]^T \in \mathcal{R}^m$ , with  $y_i = y(t_i)$ , and coefficient matrix  $X \in \mathcal{R}^{m \times (2n+1)}$  has entries which are integrals of the input and output functions, and the final column is the input itself, when  $\beta_0 \neq 0$ ,

$$X_{ij} = \begin{cases} I_{\frac{j+1}{2}}(u(t_i)), & 1 \leq j \leq 2n-1, \quad j \text{ odd} \\ -I_{\frac{j}{2}}(y(t_i)), & 2 \leq j \leq 2n, \quad j \text{ even} \\ u(t_i), & j = 2n+1 \end{cases}. \quad (\text{A.6})$$

The ordering of the columns here facilitates programming logic to handle cases in which either or both of  $\alpha_2$  and  $\beta_0$  do not occur in the equation, see below, through the lengthening of the vector  $\theta$  as appropriate. Assuming that  $X$  is of full column rank,

$$\theta_{\text{ls}} = (X^T X)^{-1} X^T \mathbf{y} \quad (\text{A.7})$$

is the unique least squares (LS) solution, where  $\theta$  contains the unknowns of the system

$$\theta = [\beta_1, -\alpha_1, \dots, \beta_n, -\alpha_n, \beta_0],$$

and notice that  $\theta$  uses negative  $\alpha$  so that the columns relating to integrals for  $y$  in  $X$  are also multiplied by a negative. For the tracer application considered here this switch in sign is useful because it forces the requirement that each component of  $\theta$  actually needs to be positive, see below.

### GLLS Algorithm

The original derivation of the GLLS algorithm made use of the lower degree in  $B(s)$  than in  $A(s)$  to derive an iterative scheme in which the noise is whitened by updating coefficients of  $A(s)$  such that the ratio  $A^{(k)}(s)/A^{(k-1)}(s)$ , which multiplies the noise term, tends to the identity with iteration level  $k$ , and requires only that one has an initial estimate for the parameters, which is typically obtained from the LS solution  $\theta_{\text{ls}} = \theta^{(0)}$ . Specifically, in (A.3) an error term  $A(s)E(s)$  is assumed on the right and the GLLS technique proceeds by division through by  $A^{(k-1)}(s)$

$$\frac{A^{(k)}(s)}{A^{(k-1)}(s)} Y(s) = \frac{B^{(k)}(s)}{A^{(k-1)}(s)} U(s) + \frac{A^{(k)}(s)}{A^{(k-1)}(s)} E(s), \quad (\text{A.8})$$

and uses the fact that the highest order term in  $A(s)$  has coefficient 1. Note that in general it is assumed that the notation  $F^{(k)}(s)$  for polynomial  $F(s)$  means that polynomial with coefficients dependent on level  $k$ . Now, using the form of  $A(s)$ , the left hand side of the equation may be rewritten as

$$\begin{aligned}
\frac{A^{(k)}(s)}{A^{(k-1)}(s)}Y(s) &= \frac{A^{(k-1)}(s) - [A^{(k-1)}(s) - A^{(k)}(s)]}{A^{(k-1)}(s)}Y(s) \\
&= Y(s) - \frac{s^n - \sum_{j=1}^n \alpha_j^{(k-1)} s^{n-j} - s^n + \sum_{j=1}^n \alpha_j^{(k)} s^{n-j}}{A^{(k-1)}(s)}Y(s) \\
&= Y(s) + \sum_{j=1}^n \alpha_j^{(k-1)} \frac{s^{n-j}}{A^{(k-1)}(s)}Y(s) - \sum_{j=1}^n \alpha_j^{(k)} \frac{s^{n-j}}{A^{(k-1)}(s)}Y(s). \tag{A.9}
\end{aligned}$$

Similarly

$$\begin{aligned}
\frac{B^{(k)}(s)}{A^{(k-1)}(s)}U(s) &= \frac{\beta_0^{(k)}A^{(k-1)}(s) + B^{(k)}(s) - \beta_0^{(k)}A^{(k-1)}(s)}{A^{(k-1)}(s)}U(s) \\
&= \beta_0^{(k)}U(s) + \frac{[B^{(k)}(s) - \beta_0^{(k)}A^{(k-1)}(s)]}{A^{(k-1)}(s)}U(s) \\
&= \beta_0^{(k)}U(s) + \sum_{j=1}^n (\beta_j^{(k)} + \beta_0^{(k)}\alpha_j^{(k-1)}) \frac{s^{n-j}}{A^{(k-1)}(s)}U(s) \tag{A.10}
\end{aligned}$$

Hence, substituting in (A.8) and ignoring the noise term, yields

$$\begin{aligned}
Y(s) + \sum_{j=1}^n \alpha_j^{(k-1)} \frac{s^{n-j}}{A^{(k-1)}(s)}Y(s) &= \beta_0^{(k)}U(s) + \\
\sum_{j=1}^n \alpha_j^{(k)} \frac{s^{n-j}}{A^{(k-1)}(s)}Y(s) + \sum_{j=1}^n \gamma_j^{(k)} \frac{s^{n-j}}{A^{(k-1)}(s)}U(s). \tag{A.11}
\end{aligned}$$

where  $\gamma_j^{(k)} = \beta_j^{(k)} + \beta_0^{(k)}\alpha_j^{(k-1)}$ ,  $j = 1, \dots, n$ . Let  $A'(s)$  denote the derivative of  $A$  with respect to  $s$ ,  $\lambda_j$  be  $n$  distinct roots of  $A(s)$ , and  $*$  denote the convolution operator. We introduce the functions

$$h_j(t) = e^{\lambda_j t}, \tag{A.12}$$

$$\psi_j(t) = \frac{(h_j * y)(t)}{A'(\lambda_j)}, \tag{A.13}$$

$$\phi_j(t) = \frac{(h_j * u)(t)}{A'(\lambda_j)}, \tag{A.14}$$

$$r(t) = y(t) + \sum_{l=1}^n \alpha_l^{(k-1)} \sum_{j=1}^n \lambda_j^{n-l} \psi_j(t), \quad i = 1, \dots, m. \tag{A.15}$$

Then taking the inverse of (A.11) yields the time domain expression

$$r(t) = \sum_{l=1}^n \left[ \alpha_l^{(k)} \sum_{j=1}^n \lambda_j^{n-l} \psi_j(t) + \gamma_l^{(k)} \sum_{j=1}^n \lambda_j^{n-l} \phi_j(t) \right] + \beta_0^{(k)}u(t), \tag{A.16}$$

which can be sampled in time to yield a LS system for the unknowns

$$\theta^{(k)} = [\gamma_1^{(k)}, -\alpha_1^{(k)}, \dots, \gamma_n^{(k)}, -\alpha_n^{(k)}, \beta_0^{(k)}]$$

:

$$Z^{(k-1)}\theta^{(k)} = \mathbf{r}^{(k-1)}, \quad (\text{A.17})$$

for  $\theta$  where  $Z^{(k-1)}$  and  $\mathbf{r}^{(k-1)}$  have row components, subscript  $i$ , in which all relevant variables are evaluated at iteration level  $k-1$

$$\mathbf{z}_i = \left[ \sum_{j=1}^n \lambda_j^{n-1} \phi_j(t_i), -\sum_{j=1}^n \lambda_j^{n-1} \psi_j(t_i), \dots, \sum_{j=1}^n \phi_j(t_i), -\sum_{j=1}^n \psi_j(t_i), u(t_i) \right] \quad (\text{A.18})$$

$$r_i = \left[ y_i + \sum_{l=1}^n \alpha_l^{(k-1)} \sum_{j=1}^n \lambda_j^{n-l} \psi_j(t_i) \right], \quad i = 1, \dots, m. \quad (\text{A.19})$$

For the specific problem it then remains to identify the coefficients in  $\theta$  correctly with the kinetic rate parameters, spillover and partial volume.

## The tracer model for use with GLLS

### Basic Model $k_4 = 0$

The simple system describing tracer dynamics, Sokoloff et al. 1977, [16], is given by the loosely coupled equations

$$\begin{aligned} \frac{dy_1}{dt} &= k_1 u(t) - (k_2 + k_3) y_1(t) \\ \frac{dy_2}{dt} &= k_3 y_1(t), \\ y_1(0) &= 0, \quad y_2(0) = 0. \end{aligned} \quad (\text{A.20})$$

Here

- $u(t)$  is the input function, the plasma FDG concentration;
- $y(t) = y_1(t) + y_2(t)$  is the output function, the tissue FDG concentration;
- $k_1 > 0$  - the transport rate of FDG from blood to the extra-vascular space; neuron loss results in lower  $k_1$  values;
- $k_2 > 0$  - the transport rate of FDG from the extra-vascular space into blood;
- $k_3 > 0$  - phosphorylation rate of the intra-cellular FDG;
- $K = \frac{k_1 k_3}{k_2 + k_3} > 0$  is used to determine the Local Cerebral Metabolic Rate (LCMR) of Glucose,  $LCMR_{glu} = \frac{K u_{glu}(t)}{LC}$ , where  $u_{glu}(t)$  is the plasma glucose
- $LC$  - lumped constant, the ratio of arterio-venous extraction fraction of FDG to that of glucose under steady state conditions, Phelps et al. 1979, [14].

This model is converted to second order in output  $y(t) = y_1(t) + y_2(t)$  to give the form (A.1):

$$\begin{aligned}\ddot{y}(t) &= \beta_1 \dot{u}(t) + \alpha_1 \dot{y}(t) + \beta_2 u(t), \\ \dot{y}(t) &= \theta_1 \dot{u}(t) + \theta_2 (-\dot{y}(t)) + \theta_3 u(t),\end{aligned}\tag{A.21}$$

in which

$$\begin{aligned}\beta_1 &= k_1 & \theta_1 &= k_1 & k_1 &= \theta_1 \\ \alpha_1 &= -(k_2 + k_3) & \theta_2 &= k_2 + k_3 & k_2 &= \theta_2 - \frac{\theta_3}{k_1} \\ \beta_2 &= k_1 k_3 & \theta_3 &= k_1 k_3 & k_3 &= \frac{\theta_3}{k_1}\end{aligned}\tag{A.22}$$

Notice here the solution for kinetic parameters in terms of coefficients of  $\theta$  and the relevance of ordering  $\theta$  as given above so that for this situation a vector of length three is appropriate, and because the kinetic parameters are by assumption always nonnegative each  $\theta$  is also nonnegative. Moreover, we should not expect that  $\theta_1 = 0$  because  $k_1$  is in general not near 0. In this case we have only three unknowns and the matrix  $X$  is obtained as the first three columns only given in (A.6):

$$X = [I_1(u), -I_1(y), I_2(u)], \quad \theta = [\beta_1, -\alpha_1, \beta_2].$$

The GLLS model in this case is simplified somewhat because  $A(s) = s^2 - \alpha_1 s$ , so that

$$\frac{A^{(k)}(s)}{A^{(k-1)}(s)} Y(s) = Y(s) + \frac{\alpha_1^{(k-1)} - \alpha_1^{(k)}}{s - \alpha_1^{(k-1)}} Y(s)$$

Similarly

$$\frac{B^{(k)}(s)}{A^{(k-1)}(s)} U(s) = \frac{\beta_1^{(k)}}{s - \alpha_1^{(k-1)}} U(s) + \frac{\beta_2^{(k)}}{s^2 - \alpha_1^{(k-1)} s} U(s),$$

so that

$$Y(s) + \alpha_1^{(k-1)} \frac{1}{s - \alpha_1^{(k-1)}} Y(s) = \alpha_1^{(k)} \frac{1}{s - \alpha_1^{(k-1)}} Y(s) + \frac{\beta_1^{(k)}}{s - \alpha_1^{(k-1)}} U(s) + \frac{\beta_2^{(k)}}{s^2 - \alpha_1^{(k-1)} s} U(s).$$

yielding the update equation

$$r(t) = \alpha_1^{(k)} (g_1(t) * y(t)) + \beta_1^{(k)} (g_1(t) * u(t)) + \frac{\beta_2^{(k)}}{\alpha_1^{(k-1)}} (g_1(t) * u(t) - u(t))$$

where

$$\begin{aligned}g_1(t) &= e^{\lambda t}, \quad \lambda = \alpha_1^{(k-1)} \\ g_2(t) &= 1 \\ r(t) &= y(t) + \alpha_1^{(k-1)} g_1(t) * y(t).\end{aligned}$$

Thus we have the system  $r = Z\theta$ , where

$$\begin{aligned}\theta^{(k)} &= [\beta_1^{(k)}, -\alpha_1^{(k)}, \beta_2^{(k)}], \\ Z &= [g_1(t) * u(t), -g_1(t) * y(t), \frac{1}{\alpha_1^{(k-1)}} (g_1(t) * u(t) - g_2(t) * u(t))].\end{aligned}$$

Notice that this is the exact expression one expects from (A.19, A.18) in which  $\alpha_1^{(k-1)}$  is one root of  $A^{(k-1)} = s^2 - \alpha_1^{(k-1)} s$ , the other being 0, and  $g_1(t) = h_1(t)$ ,  $g_2(t) = 1$ ,  $A'(\lambda_1) = \alpha_1^{(k-1)}$  and  $A'(\lambda_2) = -\alpha_1^{(k-1)}$ . Moreover, the kinetic parameters can still be updated using the expression in (A.22) because no spillover is included.

## Basic model with spillover included

In this case it is assumed that the measured tissue tracer concentration contains contamination from the plasma TAC, so that

$$y^m(t) = y(t) + m_{bt}u(t) = y_1(t) + y_2(t) + m_{bt}u(t). \quad (\text{A.23})$$

The second order system (A.21) is replaced in terms of  $y^m$  after differentiating by an equation which includes the second order derivative of  $u$ :

$$\ddot{y}^m(t) - \alpha_1 \dot{y}^m(t) = \beta_0 \ddot{u}(t) + \beta_1 \dot{u}(t) + \beta_2 u(t), \quad (\text{A.24})$$

yielding

$$X = [I_1(u), -I_1(y), I_2(u), u(t)], \quad \theta = [\beta_1, -\alpha_1, \beta_2, \beta_0].$$

Here, assuming that  $\theta_4 = 0$ , ( $k_4 = 0$ )

$$\begin{aligned} \beta_1 &= k_1 + m_{bt}(k_2 + k_3) & \theta_1 &= k_1 + \theta_5 \theta_2 & k_1 &= \theta_1 - \theta_5 \theta_2 \\ \alpha_1 &= -(k_2 + k_3) & \theta_2 &= k_2 + k_3 & k_2 &= \theta_2 - \frac{\theta_3}{k_1} \\ \beta_2 &= k_1 k_3 & \theta_3 &= k_1 k_3 & k_3 &= \frac{\theta_3}{k_1} \\ \beta_0 &= m_{bt} & \theta_5 &= m_{bt} & m_{bt} &= \theta_5 \end{aligned} \quad (\text{A.25})$$

Now when we move to GLLS we introduce the  $\gamma$  coefficients instead of  $\beta$  coefficients

$$\theta^{(k)} = [\gamma_1^{(k)}, -\alpha_1^{(k)}, \gamma_2^{(k)}, \beta_0^{(k)}],$$

and

$$\theta_1^{(k)} = \gamma_1^{(k)} = \beta_1^{(k)} + \theta_5^{(k)} \alpha_1^{(k-1)} = k_1 + \theta_5^{(k)} (\theta_2^{(k)} - \theta_2^{(k-1)})$$

yields

$$\begin{aligned} k_1 &= \theta_1^{(k)} - \theta_5^{(k)} (\theta_2^{(k)} - \theta_2^{(k-1)}) \\ k_2 &= \theta_2^{(k)} - \frac{\theta_3^{(k)}}{k_1} \\ k_3 &= \frac{\theta_3^{(k)}}{k_1} \\ m_{bt} &= \theta_5^{(k)}. \end{aligned}$$

We see that this is the same set of equations as for the LS solution except that occurrences of  $\theta_2$  are replaced by differences from level  $k$  and  $k - 1$  of this variable. In this case we note that  $\theta_4 = 0$ ,  $k_4 = 0$ .

## Basic Model with spillover and partial volume

Now assume not only (A.23) but that  $y$  is not fully recovered due to the impact of partial volume, so that

$$y^m(t) = r_y y(t) + m_{bt}u(t) = r_y(y_1(t) + y_2(t)) + m_{bt}u(t), \quad (\text{A.26})$$

where now  $r_y$  is the recovery coefficient,  $r_y + m_{bt} = r_y + \theta_5 = 1$ ,

$$\begin{aligned} \beta_1 &= r_y k_1 + m_{bt}(k_2 + k_3) & \theta_1 &= r_y k_1 + m_{bt}(k_2 + k_3) & k_1 &= \frac{\theta_1 - \theta_5 \theta_2}{r_y} \\ \alpha_1 &= -(k_2 + k_3) & \theta_2 &= k_2 + k_3 & k_2 &= \theta_2 - \frac{\theta_3 - \theta_5 \theta_4}{\theta_1 - \theta_5 \theta_2} \\ \beta_2 &= r_y k_1 k_3 & \theta_3 &= r_y k_1 k_3 & k_3 &= \frac{\theta_3}{\theta_1 - \theta_5 \theta_2} \\ \beta_0 &= m_{bt} & \theta_5 &= m_{bt} & m_{bt} &= \theta_5. \end{aligned} \quad (\text{A.27})$$

Here we can calculate as before but then divide out  $k_1$  by the recovery coefficient  $1 - \theta_5$ .

### Standard Model

Introducing the additional parameter  $k_4$ , the de-phosphorylation rate of the intra-cellular FDG-6-phosphate, yields

$$\begin{aligned}\frac{dy_1}{dt} &= k_1 u(t) - (k_2 + k_3)y_1(t) + k_4 y_2(t), \\ \frac{dy_2}{dt} &= k_3 y_1(t) - k_4 y_2(t), \\ y_1(0) &= 0, \quad y_2(0) = 0,\end{aligned}\tag{A.28}$$

and

$$\begin{aligned}\ddot{y}(t) &= \beta_1 \dot{u}(t) + \alpha_1 \dot{y}(t) + \beta_2 u(t) + \alpha_2 y(t), \\ \dot{y}(t) &= \theta_1 \dot{u}(t) + \theta_2 (-\dot{y}(t)) + \theta_3 u(t) + \theta_4 (-y(t)),\end{aligned}\tag{A.29}$$

so that

$$X = [I_1(u), -I_1(y), I_2(u), -I_2(y)], \quad \theta = [\beta_1, -\alpha_1, \beta_2, -\alpha_2].$$

Here

$$\begin{aligned}\beta_1 &= k_1 & \theta_1 &= k_1 & k_1 &= \theta_1 \\ \alpha_1 &= -(k_2 + k_3 + k_4) & \theta_2 &= k_2 + k_3 + k_4 & k_2 &= \theta_2 - \frac{\theta_3}{\theta_1} \\ \beta_2 &= k_1(k_3 + k_4) & \theta_3 &= k_1(k_3 + k_4) & k_3 &= \frac{\theta_3}{\theta_1} - k_4 \\ \alpha_2 &= -k_2 k_4 & \theta_4 &= k_2 k_4 & k_4 &= \theta_4 / k_2\end{aligned}\tag{A.30}$$

Notice here the similarity of the equations for the parameters  $k_i$  with (A.22) so that exactly the same equations can be used in either case, with  $k_4$  calculated before  $k_3$ , identically zero when  $\theta_4 = 0$  in the first case. For the GLLS there are no complications to do the update for the parameters since the GLLS equation uses the same definition for  $\theta$  as LS:

$$\theta^{(k)} = [\beta_1^{(k)}, -\alpha_1^{(k)}, \beta_2^{(k)}, -\alpha_2^{(k)}]$$

$$\begin{aligned}Z^{(k-1)}\theta^{(k)} &= \mathbf{r}^{(k-1)}, \\ \mathbf{z}_i &= \left[ \sum_{j=1}^2 \lambda_j \phi_j(t_i), -\sum_{j=1}^2 \lambda_j \psi_j(t_i), \sum_{j=1}^2 \phi_j(t_i), -\sum_{j=1}^2 \psi_j(t_i), \right] \\ r_i &= \left[ y_i + \sum_{l=1}^2 \alpha_l^{(k-1)} \sum_{j=1}^2 \lambda_j^{n-l} \psi_j(t_i) \right], \quad i = 1, \dots, m,\end{aligned}$$

again where the roots  $\lambda_j$  are roots of  $s^2 - \alpha_1^{(k-1)}s - \alpha_2^{(k-1)} = s^2 + \theta_2^{(k-1)}s + \theta_4^{(k-1)} = 0$ , and the equation is completely consistent with (A.19, A.18). (Use exactly the same equations for GLLS and LS - no spillover)

### Model with Spillover from blood to tissue $m_{bt}$

In this case including (A.23) the second order system (A.29) is replaced

$$y^{\ddot{m}}(t) - \alpha_1 y^{\dot{m}}(t) - \alpha_2 y^m(t) = \beta_0 \ddot{u}(t) + \beta_1 \dot{u}(t) + \beta_2 u(t),\tag{A.31}$$

yielding

$$X = [I_1(u), -I_1(y), I_2(u), -I_2(y), u(t)], \quad \theta = [\beta_1, -\alpha_1, \beta_2, -\alpha_2, \beta_0].$$

Here

$$\begin{aligned}
\beta_1 &= k_1 + m_{bt}(k_2 + k_3 + k_4) & \theta_1 &= k_1 + m_{bt}(k_2 + k_3 + k_4) & k_1 &= \theta_1 - \theta_5 \theta_2 \\
\alpha_1 &= -(k_2 + k_3 + k_4) & \theta_2 &= k_2 + k_3 + k_4 & k_2 &= \theta_2 - \frac{\theta_3 - \theta_5 \theta_4}{\theta_1 - \theta_5 \theta_2} \\
\beta_2 &= k_1(k_3 + k_4) + m_{bt}k_2k_4 & \theta_3 &= k_1(k_3 + k_4) + m_{bt}k_2k_4 & k_3 &= \frac{\theta_3 - \theta_5 \theta_4}{\theta_1 - \theta_5 \theta_2} - k_4 \\
\alpha_2 &= -k_2k_4 & \theta_4 &= k_2k_4 & k_4 &= \frac{\theta_4}{k_2} \\
\beta_0 &= m_{bt} & \theta_5 &= m_{bt} & m_{bt} &= \theta_5
\end{aligned} \tag{A.32}$$

Again, the same equations can be used for the calculations as above, with  $m_{bt}$  calculated last, and noting that  $\theta_5$  is only nonzero when the spillover is also nonzero. When we move to the GLLS system it is defined as given in the general derivation in terms of  $\gamma$  coefficients instead of  $\beta$  coefficients

$$\theta^{(k)} = [\gamma_1^{(k)}, -\alpha_1^{(k)}, \gamma_2^{(k)}, -\alpha_2^{(k)}, \beta_0^{(k)}],$$

and

$$\begin{aligned}
\mathbf{z}_i &= \left[ \sum_{j=1}^2 \lambda_j \phi_j(t_i), -\sum_{j=1}^2 \lambda_j \psi_j(t_i), \sum_{j=1}^2 \phi_j(t_i), -\sum_{j=1}^2 \psi_j(t_i), u(t_i) \right] \\
r_i &= \left[ y_i + \sum_{l=1}^2 \alpha_l^{(k-1)} \sum_{j=1}^2 \lambda_j^{2-l} \psi_j(t_i) \right], \quad i = 1, \dots, m.
\end{aligned}$$

Given

$$\begin{aligned}
\theta_1^{(k)} = \gamma_1^{(k)} &= \beta_1^{(k)} + \theta_5^{(k)} \alpha_1^{(k-1)} = k_1 + \theta_5^{(k)} (\theta_2^{(k)} - \theta_2^{(k-1)}) \\
\theta_3^{(k)} = \gamma_2^{(k)} &= \beta_2^{(k)} + \theta_5^{(k)} \alpha_2^{(k-1)} = k_1(k_3 + k_4) + \theta_5^{(k)} (\theta_4^{(k)} - \theta_4^{(k-1)}),
\end{aligned}$$

yields

$$\begin{aligned}
k_1 &= \theta_1^{(k)} - \theta_5^{(k)} (\theta_2^{(k)} - \theta_2^{(k-1)}) \\
k_2 &= \theta_2^{(k)} - (\theta_3^{(k)} - \theta_5^{(k)} (\theta_4^{(k)} - \theta_4^{(k-1)})) / k_1 \\
k_3 &= (\theta_3^{(k)} - \theta_5^{(k)} (\theta_4^{(k)} - \theta_4^{(k-1)})) / k_1 - k_4 \\
k_4 &= \theta_4^{(k)} / k_2 \\
m_{bt} &= \theta_5^{(k)}.
\end{aligned}$$

We see that this is the same set of equations as for the LS solution except that occurrences of  $\theta_2$  and  $\theta_4$  are replaced by differences from level  $k$  and  $k - 1$  of these variables.

### Standard Model with Spillover from blood to tissue and partial volume

Using again (A.26) we obtain the same system as (A.24) but now with

$$\begin{aligned}
\beta_1 &= r_y k_1 + m_{bt}(k_2 + k_3 + k_4) & \theta_1 &= r_y k_1 + m_{bt}(k_2 + k_3 + k_4) & k_1 &= \frac{\theta_1 - \theta_5 \theta_2}{r_y} \\
\alpha_1 &= -(k_2 + k_3 + k_4) & \theta_2 &= k_2 + k_3 + k_4 & k_2 &= \theta_2 - \frac{\theta_3 - \theta_5 \theta_4}{\theta_1 - \theta_5 \theta_2} \\
\beta_2 &= r_y k_1(k_3 + k_4) + m_{bt}k_2k_4 & \theta_3 &= r_y k_1(k_3 + k_4) + m_{bt}k_2k_4 & k_3 &= \frac{\theta_3 - \theta_5 \theta_4}{\theta_1 - \theta_5 \theta_2} - k_4 \\
\alpha_2 &= -k_2k_4 & \theta_4 &= k_2k_4 & k_4 &= \frac{\theta_4}{k_2} \\
\beta_0 &= m_{bt} & \theta_5 &= m_{bt} & m_{bt} &= \theta_5.
\end{aligned} \tag{A.33}$$

The same equations can again be used, except that where  $r_y \neq 0$  at the end  $k_1$  is divided by  $r_y = 1 - \theta_5$ . Moreover, the GLLS update follows again as for the non partial volume case, but with once more the division of  $k_1$  by the partial volume coefficient at the final step.

We conclude that we can use the same code for calculating the kinetic parameters for all cases. It is just important that for the  $k_4 = 0$  cases,  $\theta_4 = 0$ , and for the cases without spillover  $\theta_5 = 0$ .

```

function k=getkfromthetaglls(theta,thetaold)
if nargin==1, thetaold=zeros(size(theta)); end
    k(5)=theta(5);
    k(1)=theta(1)-theta(5)*(theta(2)-thetaold(2));
    k(2)=theta(2)-(theta(3)-theta(5)*(theta(4)-thetaold(4)))/k(1);
    k(4)=theta(4)/k(2);
    k(3)=theta(3)-k(2)-k(4);
% if partial volume
%k(6)=1-k(5); k(1)=k(1)/k(6);
% note also K = k(1)*k(3)/(k(2)+k(3));

```

# Bibliography

- [1] A. Bjorck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, PA, 1996.
- [2] K. Chen, D. Bandy, E. Reiman, S. C. Huang, M. Lawson, D. Feng, L. Yun, and A. Palant, *Noninvasive Quantification of the Cerebral Metabolic Rate for Glucose Using Positron Emission Tomography, F-Fluoro-2-Deoxyglucose, the Patlak Method, and an Image Derived Input Function*, J. Cereb. Blood Flow Metab., 18, 716–723, 1998.
- [3] K. Chen, M. Lawson, E. Reiman, A. Cooper, D. Feng, S. C. Huang, D. Bandy, D. Ho, L. Yen, and A. Palant, *Generalized Linear Least Squares Method for Fast Generation of Myocardial Blood Flow Parametric Images with N-13 Ammonia PET*, IEEE Trans. Med. Imag., 17, 2, 236–243, 1998.
- [4] A. R. Conn, N. I. M. Gould and P. L. Toint, *Trust-Region Methods*, MPS-SIAM, Philadelphia, PA, 2000.
- [5] D. Feng and D. Ho, *Parametric imaging algorithms for multicompartamental models dynamic studies with positron emission tomography*, in Quantification of Brain Function: Tracer Kinetics and Image Analysis in Brain PET, K. Uemura, N. A. Lassen, T. Jones, and I. Kanno, eds., Elsevier Science Publishers, Amsterdam, The Netherlands, 127–137, 1993.
- [6] D. Feng, D. Ho, K. Chen, L. Wu, J. Wang, R. Liu, and S. Yeh, *An Evaluation of the Algorithms for Determining Local Cerebral Metabolic Rates of Glucose Using Positron Emission Tomography Dynamic Data*, IEEE Trans. Med. Imag., 14, 4, 697–710, 1994.
- [7] D. Feng, S. C. Huang, and Z. Wang, *Models for Computer Simulation Studies of Input Functions for Tracer Kinetic Modeling with Positron Emission Tomography*, Intern. J. Bio-Med. Comp., 32, 95–110, 1993.
- [8] D. Feng, S. C. Huang, Z. Wang, and D. Ho, *An Unbiased Parametric Imaging Algorithm for Nonuniformly Sampled Biomedical System Parameter Estimation*, IEEE Trans. Med. Imag., 15, 4, 512–518, 1996.
- [9] D. Feng, Z. Wang and S. C. Huang, *A Study on Statistically Reliable and Computationally Efficient Algorithms for Generating Local Cerebral Blood Flow Parametric Images with Positron Emission Tomography*, IEEE Trans. Med. Imag., 12, 2, 182–188, 1993.
- [10] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, PA, 1996.
- [11] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.
- [12] C. T. Kelley, *Iterative Methods for Optimization*, SIAM, Philadelphia, PA, 1999.
- [13] C. Negoita, *Global Kinetic Imaging Using Dynamic Positron Emission Tomography Data*, PhD. thesis, Department of Mathematics and Statistics, Arizona State University, AZ., 2003.
- [14] M. Phelps, S.C. Huang, E. Hoffman, C. Selin, L. Sokoloff and D. Kuhl, *Tomographic Measurement of Local Cerebral Glucose Metabolic Rate in Humans with (F-18)2-Fluoro-2-Deoxy-D-Glucose: Validation of Method*, Ann. Neurol., 6, 371–388, 1979.
- [15] M. Piert, R. Koeppe, B. Giordani, S. Berent, and D. Kuhl, *Diminished Glucose Transport and Phosphorylation in Alzheimer's Disease Determined by Dynamic FDG-PET*, J. Nuclear Med., 37, pp. 201–208, 1996.

- [16] L. Sokoloff, M. Reivich and C. Kennedy, *The ( $^{14}\text{C}$ )deoxyglucose method for the measurement of local cerebral glucose utilization: theory, procedure and normal values in the conscious and anesthetized albino rat*, J. Neurochem., 28, 897–910, 1977.
- [17] K. Wong and D. Feng, *Generalized linear least squares algorithm for non-uniformly sampled biomedical system identification with possible repeated eigenvalues*, Comp. Methods Prog. Biomed., 57, 167–177, 1998.