Version 1.0

Graham Searle, Christopher Coello and Roger Gunn
CONTENTS

1. Overview ........................................................................................................................................................ 3
   1.1. Introduction .......................................................................................................................................... 3
   1.2. Citing MIAKAT™ ..................................................................................................................................... 3
   1.3. Copyright ............................................................................................................................................... 3
   1.4. Authors .................................................................................................................................................. 4

2. Installation ...................................................................................................................................................... 5
   2.1. System Requirements ........................................................................................................................... 5
   2.2. Installing MIAKAT™ ............................................................................................................................... 5

3. Using MIAKAT™ .............................................................................................................................................. 7
   3.1. Software and Data Philosophy .............................................................................................................. 7
   3.2. Data Formats ......................................................................................................................................... 7
   3.3. Setting Up A New Study / Subject ......................................................................................................... 8
   3.4. The Standard Analysis Pipeline ........................................................................................................... 13
   3.4. Visualisation And Quality Control Of The Results ................................................................................ 33

4. Advanced Information .................................................................................................................................. 40
   4.1. Description Of The Main Window ....................................................................................................... 40
   4.2. Analysis Manager Data Structure ........................................................................................................ 43
   4.3. Analysis Data Structure ....................................................................................................................... 43
   4.4. Key Concepts ....................................................................................................................................... 43
   4.5. Using Manual ROIs .............................................................................................................................. 44

5. Bibliography ................................................................................................................................................. 46

6. Annexes ........................................................................................................................................................ 47
   6.1. ANC File Description ............................................................................................................................ 47
   6.2. Parent Fraction Models ........................................................................................................................ 51
   6.3. Plasma-Over-Blood Models ................................................................................................................ 52
   6.4. Licence ................................................................................................................................................. 53
   6.5. Test Dataset ........................................................................................................................................ 56

Document version history:

<table>
<thead>
<tr>
<th>Document version</th>
<th>Date:</th>
<th>Notes:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>17th December 2015</td>
<td>First release to accompany MIAKAT_release_4.2.0</td>
</tr>
</tbody>
</table>
1. **OVERVIEW**

This is the installation and user guide for MIAKAT™ (Molecular Imaging And Kinetic Analysis Toolbox). For more information about analysis of dynamic positron emission tomography (PET) imaging, the authors recommend the review paper entitled Quantitative imaging of protein targets in the human brain with PET [Gunn, 2015].

1.1. **INTRODUCTION**

MIAKAT™ is a software package for the quantitative analysis of PET neuroimaging data. It enables the efficient application of state of the art image processing and kinetic modelling techniques within a single software tool that ensures the quality and reproducibility of the analysis performed. The workflow framework central to the MIAKAT™ implementation allows the user to configure an analysis pipeline for a given neuroimaging research study, minimizing the user interaction required and enabling quality control (QC) of the complete analysis process.

MIAKAT™ is implemented in MATLAB™, and has a central graphical user interface (GUI) that facilitates “point and click” operation, without needing to know how to use MATLAB™ which enables users with a range of backgrounds to understand and interact with the software. However, the software is distributed as MATLAB™ m-code, free for non-commercial use, so that users can either run preconfigured standard analysis workflows, or bespoke pipelines and processes to suit their needs.

The standard brain analysis pipeline implemented in MIAKAT™ takes the primary experimental data (including dynamic PET and structural MR images, along with ancillary data such as arterial blood measurements) and performs a sequence of processes, including:

- Brain Extraction
- Brain Tissue Segmentation
- Motion Correction
- Region of interest (ROI) Definition via a Neuroanatomical Atlas
- Blood/Plasma Function Modelling
- ROI Tracer Kinetic Modelling
- Parametric Imaging

which ultimately produces results in the form of regional (or voxel-wise) parameters extracted via a comprehensive set of modelling techniques. MIAKAT™ combines in-house code with wrappers for FMRIB Software Library (FSL, http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/) and Statistical Parametric Mapping (SPM, http://www.fil.ion.ucl.ac.uk/spm/) commands in order to provide state-of-the-art functionality within a coherent framework. The standard neuroanatomical atlas included with MIAKAT is the CIC atlas version 2.0, defined on the nonlinear ICBM152 template and developed using the region of interest definitions described in Tziorti et al. 2011.

1.2. **CITING MIAKAT™**

We hope to have an appropriate publication in the literature soon, and request that you cite it in work that you publish where you have used MIAKAT™. This document (and the www.miakat.org webpage) will be updated once a suitable reference is available.

1.3. **COPYRIGHT**

MIAKAT™ is freely available ONLY for academic purposes. It is not available for industrial or commercial applications of any kind without explicit and prior arrangement with the authors. The software (in full or part)
must not be posted on any WWW or ftp sites or distributed in any other way without prior permission of the authors. Please refer to the Annexe 6.4 for a copy of the standard MIAKAT licence agreement.

1.4. AUTHORS

Developed by Graham Searle & Roger Gunn

Graham Searle is Principal PET Analyst at Imanova. He holds a Master’s degree in mathematics and a PhD in mathematical modelling. He has been working in the field of PET image analysis since 2006, and in that time has focused on human neuroimaging, particularly in the context of drug development. His previous work centred on the development of scientific software for modelling applications, and this background enabled him to play the lead role in the development of the MIAKAT™ software framework.

Roger Gunn is Chief Scientific Officer and Head of Analysis at Imanova, Professor of Molecular Neuroimaging in the Division of Brain Sciences at Imperial College London and Visiting Professor in the Department of Engineering at the University of Oxford. He is an applied mathematician who has held positions in academia and industry in the UK and North America. MIAKAT™ contains analytical techniques developed during his 20 year career in the field of PET Neuroimaging.

With contributions from Christopher Coello, Cristian Salinas, Andri Tziortzi & Qi Guo

Christopher Coello is a PET Analyst at Imanova since 2013. He is an Electrical Engineer from Bordeaux INP with an additional Master’s Degree in Medical Imaging from University Paris XI where he worked on SPECT/PET reconstruction. His doctoral degree was spent studying dual-energy x-ray imaging, in collaboration with INSA Lyon and CEA Grenoble. A post-doctoral work on preclinical PET image analysis and kinetic modelling of opioid receptor tracers at the University of Oslo definitely convinced him that PET imaging was fun.

Cristian Salinas is an Associate Principal Research Scientist at Merck Research Laboratories. He holds a PhD in Biomedical Engineering from Case Western reserve University USA. He has been actively working in the field of PET kinetic modeling since 2001. He is currently involved in the application of optimal experimental design techniques applied to PET in order to characterize non-trivial PK-PD relationships to support the development of novel pharmaceutical therapies.

Andri Tziortzi is a Physics Teacher and a Honorary Research Fellow at Imanova. She holds a Physics degree from the National and Kapodistrian University of Athens and a Master’s degree in Medical Physics. She obtained her DPhil in Clinical Neurosciences at the University of Oxford. During her studies and work at GSK she has been involved in the analysis of PET, CT, MRI and DTI images in the context of drug development. Her previous work centred primarily around the construction of structural and functional brain atlases.

Qi Guo is a senior scientist at AbbVie. She received a DPhil in Biomedical Engineering from University of Oxford, UK and Bachelor in Electronic Engineering from Peking University China. She has been working in PET imaging since 2006 with Oxford, Imperial College, King’s College London, GSK and Imanova. Her previous work centered around the development of mathematics for novel PET probes discovery and innovation in tracer quantification methods with a particular focus on imaging neuroinflammation in various diseases.
2. INSTALLATION

The source files can be downloaded at www.miakat.org/downloads

After completing registration information, the user will receive an email with links to:

- The source code (MIAKAT_release_4.2.0.zip) which is required for the installation of MIAKAT™
- The test dataset (optional)

The test data is described in detail in Annexe 6.5.

2.1. SYSTEM REQUIREMENTS

MIAKAT™ is available for Linux (tested on RedHat and Debian distributions) and Mac OS X.

MATLAB™ (release 2015a or newer) has to be already installed along with the MATLAB™ Optimisation Toolbox. Please ensure that this toolbox is present; to do so, copy the following MATLAB command in your MATLAB command window and ensure ‘Optimization Toolbox’ is one of the displayed results:

```
>> v = ver; setdiff((v.Name),'MATLAB')`
```

In order to execute some of the MIAKAT™ functions, the following programs are required and should be downloaded, installed and tested before MIAKAT™ installation:

- SPM 12: http://www.fil.ion.ucl.ac.uk/spm/software/spm12/
- FSL 5.0.4 or more recent: http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/

These two programs are free to download and install for academics. Please refer to the websites of FSL and SPM for details of their installation.

2.2. INSTALLING MIAKAT™

Once the source code archive has been downloaded using the link received in the email, the steps to install MIAKAT™ are as follows:

1. uncompress the archive in a temporary location. The location where the archive is downloaded will NOT be the final MIAKAT™ installation location. For example

```
user@mydatamachine:~/Downloads$ unzip MIAKAT_XXXX.zip
```

2. (remove any pre-existing MIAKAT™ configuration files from your home directory, if an older version of MIAKAT™ has previously been installed)

3. launch MATLAB and make sure that the right SPM version path is added to MATLAB paths (see SPM installation documentation)

4. navigate to the temporary location where the archive has been uncompresssed

5. type in the MATLAB command window:

```
>> install_MIAKAT
```

6. MIAKAT™ will proceed to the installation. During the installation, an interface will ask the user for information about the location in which the user wants to install MIAKAT™ and where SPM / FSL are located. By default, MIAKAT™ will try to harvest some of this information (for example, FSL location) and populate the relevant fields with a possible path / version.
7. MIAKAT™ will then propose two alternatives to ensure that MIAKAT™ is available every time that you start MATLAB:
   a. **Create / modify your startup.m file.** The startup.m file is read automatically when MATLAB is launched. If this option is chosen, MIAKAT™ will add two lines to the end of the startup.m file. These two lines will load the MIAKAT™ and SPM repositories each time one starts MATLAB.
   b. **Launch pathtool and save manually.** This option will launch the MATLAB pathtool and the user will have the opportunity to save the modified path at his / her convenience.

8. If MIAKAT™ has been successfully installed, type:

   ```
   >> MIAKAT
   ```

   to launch the graphical interface of MIAKAT™. Remember that SPM must already be in the MATLAB path for MIAKAT™ to work properly.

During the installation, a configuration file called `miakatConfig.txt` will be saved in the user’s home directory. For Linux and MAC users, the user’s home directory can be found typing in a terminal:

```
user@mydatamachine:~/$ echo $HOME
```

This configuration file contains the information provided during the installation of MIAKAT™ and can be edited to reflect changes or updates in the working environment of the user. (This configuration can also be edited from within the main MIAKAT GUI.) It is important to note that the information in this file is the one MIAKAT™ uses during the pipeline.

A typical configuration file can be seen in Figure 1.

---

**FIGURE 1. TYPICAL CONFIGURATION FILE (MIAKATConfig.TXT) THAT IS AUTOMATICALLY CREATED DURING INSTALLATION OF MIAKAT™ AND STORED IN THE USER’S HOME DIRECTORY.**
3. Using MIAKAT™

3.1. Software and Data Philosophy

The MIAKAT workflow and data model is constructed on a subject-centric basis. Each workflow or pipeline is constructed and run for a subject’s data, and simply replicated for further subjects in a given study. The analysis pipeline for a SUBJECT is configured and recorded in a single MATLAB structure called the AnalysisManager. A subject’s data set is itself considered to be composed of a number of EXAMINATIONS. Strictly, the definition of an examination is somewhat flexible, but it is usually considered to be equivalent to “a session on a scanner”. Thus, a subject’s data set would commonly consist of one MRI exam and one or more PET exams. For each examination, there might be more than one image. These could correspond to multiple MRI sequences, multiple PET/CT acquisitions (e.g. CT, PET) or simply multiple PET reconstructions (e.g. with and without attenuation correction). MIAKAT results and records specific to the analysis pipeline of a given EXAMINATION are stored in a MATLAB structure called the Analysis data structure.

The AnalysisManager data structure is stored as a variable named AnalysisManager, alone in a binary mat-file and contains information regarding:

- Examinations and processes to be done
- Configuration options for each process
- Progress of the analysis throughout the pipeline
- Paths to input/output data (i.e. images, ancillary files, numeric results …)
- References to all relevant Analysis data structure mat-files for a given subject.

The Analysis data structure is stored as a variable named Analysis, alone in a binary mat-file and contains:

- Information (low-level functions used to achieve a process, options specified, software version, etc…) regarding processes of a given examination. Therefore the number of Analysis data structures will be equal to the number of examinations.
- Inputs / outputs of processes of an examination. If inputs / outputs are images, the Analysis data structure stores the only the image path to these images. If inputs / outputs are numeric data of time activity curves or kinetic modelling, it stores the numbers within its fields for a given examination.

The AnalysisManager data structure can be modified using either the MIAKAT GUI or the MATLAB™ command line. This allows MIAKAT™ to be flexible such that dynamic PET data analysis becomes available to users that don’t have experience with MATLAB, as well as experienced MATLAB users that, for example, might like to perform batch runs with multiple options on multiple subjects.

3.2. Data Formats

Image files (dynamic PET image and structural MR image) are required to be in NIfTI (.nii), NIfTI Pair (.img/.hdr) or Analyze (.img/.hdr) format or the zipped version of these format (.nii.gz and .img.gz/.hdr.gz). The working format used in MIAKAT is NIfTI Pair. As a consequence, a pre-processing step will convert the images not in NIfTI Pair format into NIfTI Pair format.

Images are required to be in radiological orientation.
Time frames, blood data, plasma data, metabolite data, subject information and others are gathered in an ancillary file (text file with predefined format, suffix .anc). Description of the content of the ancillary file can be found in the Annexe 6.1. An example of ancillary file can be found with the test dataset provided with this software.

Important: the ancillary file should be named the same as the Analysis filename (see Paragraph 3.2.4.2 for more details).

3.3. Setting Up A New Study / Subject

In order to set up a new image analysis pipeline using MIAKAT™, one should:

- Ensure that input image file(s) and ancillary file(s) are in a MIAKAT™ compatible format (required),
- Set up a subject-oriented folder structure (this step is not mandatory but highly recommended to be able to navigate easily through the analysis results),
- Construct or select an AnalysisManager data structure template for the study/subject (required).

3.3.1. Folder Structure

A subject-oriented folder structure (Figure 2) is strongly recommended to be able to navigate easily through the output results. This folder structure can simply be created by assigning one folder to each subject in the study and use this folder as a root containing:

- one Analysis Manager file specific to the subject,
- a separate folder for each examination in the analysis pipeline. The content of this folder is at the discretion of the user. All the output results (images, Analysis data structure file, etc...) of each examination will be stored in that examination folder where the input images (e.g. dynamic PET, MRI) are located.

![FIGURE 2. EXAMPLE OF A RECOMMENDED SUBJECT-ORIENTED FOLDER STRUCTURE WHEN USING MIAKAT™. A MAIN SUBJECT FOLDER (/SCRATCH/USER/DATA/SUBJ001/) CONTAINS THE ANALYSIS MANAGER FILE SPECIFIC TO THE SUBJECT (AM_SUBJ001.MAT) AND FOLDERS FOR EACH EXAMINATION THE SUBJECT UNDERWENT (PET_00001, PET_00002 AND MRI_00001).]
3.3.2. CREATE AN ANALYSIS MANAGER TEMPLATE

An AnalysisManager Template (AMT) file contains an AnalysisManager data structure configured with a predefined analysis pipeline with no image information. During the installation of MIAKAT™ installation, several of these template files are unpacked in a folder defined in the MIAKAT configuration file miakatConfig.txt (see paragraph 2.2). The AnalysisManager Template data structure can be customized to the specificity of your study and used as the starting AnalysisManager file for every subject, allowing an easy reproduction of the same analysis pipeline subject after subject.

Two possibilities are offered to you in order to start analysing a new subject:

- Use an existing AnalysisManager Template (File > New From Template)
- Customize a default AnalysisManager Template using a graphical interface (File > New From Wizard) or the MATLAB command line.

AnalysisManager and AnalysisManager Template files can be loaded in MATLAB™ using the MATLAB command load function or the MIAKAT processAnalysisManagerInput function:

```matlab
>> load('myAnalysisManagerTemplate.mat')
```

will load the structure stored in the file ‘myAnalysisManagerTemplate.mat’ to a variable called AnalysisManager.

```matlab
>> AMT = processAnalysisManagerInput('myAnalysisManagerTemplate.mat')
```

will assign to the variable AMT the structure saved in the file ‘myAnalysisManagerTemplate.mat’.

3.3.2.1. USING AN EXISTING TEMPLATE

An existing AMT can be loaded in MIAKAT using the GUI (File > New From Template), a popup window will then ask the user to choose the AMT file of interest which can be either one of the default ones provided with the MIAKAT™ software, or a previously customised template. When the AMT file containing the Analysis Manager data structure is loaded, the data structure will be displayed in the AnalysisManager Structure Explorer Panel on the left hand side of the main GUI.

3.3.2.2. CUSTOMIZE A DEFAULT TEMPLATE

A default AMT can be customised using the GUI (File > New From Wizard) by selection of workflow elements from predefined set of options (Figure 3):
In case of doubt, default option selection is recommended. A help button at the bottom right corner of the window is available to assist users with their selection. Alternatively, by hovering the mouse for more than 2 seconds on an option, a tooltip containing extra information on that option will appear.

Once the workflow and its options are customised (see an example in Annex 6.5), it can be saved as a new AMT file using the “Save As Template” button at the bottom part of the window. If one prefers to continue without saving the selected workflow and associated options, it can be directly loaded in MIAKAT using the “OK” button.

### 3.3.3. Run Pipeline

At this stage, the user should normally have an AnalysisManager Template loaded in the MIAKAT™ main GUI (as in Figure 4). To run the pipeline, the user can either click on the icon ![Run icon](image) or launch from the menu bar of the main interface (Pipeline > Run).
Further GUIs will prompt you to fill in general information about the subject to be analysed and all the examinations related to this subject. Every single GUI has an embedded help button on the bottom right of the window, together with extra information in the tooltip of every item.

3.3.3.1. **Subject Info GUI**

The first window that will show up will allow the user to enter the location where MIAKAT is going to store the Analysis Manager for the subject to be analysed and other subject related information (Figure 5).

![Subject Info GUI](image)

**FIGURE 5.** THE SUBJECT INFO GUI ALLOWS THE USER TO DEFINE THE LOCATION AND NAME OF THE ANALYSIS MANAGER FOR THIS Subject, TOGETHER WITH SOME STUDY AND SUBJECT SPECIFIC INFORMATION.

3.3.3.2. **Exam Info GUI**

For each examination (MRI, PET1, PET2, etc...), an exam specific information window will pop up to allow the users to provide information about the location of the images, the units of the image voxels (PET only), the name of the Analysis filename you would like to choose, etc.... MIAKAT will fill automatically some fields with
some predefined values to help the users. In addition, the Help icon (?) will provide a detailed description of each of the fields.

**Important:** The Analysis file for the PET examination should have the same name as the ANC file.

![Image of MRI and PET exam specific windows](image)

**FIGURE 6. MRI (TOP) AND PET (BOTTOM) EXAM SPECIFIC WINDOWS WHERE THE USERS PROVIDE THE LOCATION OF THE IMAGES. THE RED FIELDS WILL TURN TO GREEN IF THE FORMAT OF THE IMAGE IS RECOGNIZED**

Once the general information is correctly passed to the software, a progress window (Figure 7) will appear to guide you through every step of the analysis pipeline and provide information on the current process being performed.
3.3.4. VISUALISE RESULTS
When the pipeline is finished, MIAKAT will automatically load the completed AnalysisManager data structure in the MIAKAT main interface. The user is able to navigate through the structure using the left hand side Explorer panel. In paragraph 3.4, the outputs of each process for a test dataset are described together with the proposed methodology to QC the results and report the QC.

3.4. THE STANDARD ANALYSIS PIPELINE
Two default AnalysisManager Template are provided with the MIAKAT™ distribution:

- AnalysisManagerTemplate_1MRI_2PET_withBlood.mat : paragraph 3.3.1
- AnalysisManagerTemplate_1MRI_2PET_noBlood.mat : paragraph 3.3.2
3.4.1. **Overview of the Analysis Pipeline with Blood Data**

**Figure 8.** Detailed schematic of the analysis pipeline included in the analysis manager template called `ANALYSISMANAGERTEMPLATE_1MRI_2PET_WITHBLOOD.MAT`. The black arrows show the dependency of the processes on each other; i.e. a process can only start when all processes pointing to it are completed.
3.4.2. Overview of the Analysis Pipeline Without Blood Data

In this section, detailed information on each process of the analysis pipeline will be provided: inputs, outputs and main options are described, together with the main MATLAB command line functions that can be used independently of MIAKAT GUI.
3.3.3.1. PREPROCESS IMAGES

This process prepares the images to the pipeline. This process creates a backup copy of the images, converts the format of the image to a convenient format (NIfTI Pair), strips the information contained in the header and finally ensures that the units of the dynamic PET AC image are in kBq/ml.

Input:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
<td>Image (static or dynamic)</td>
</tr>
</tbody>
</table>

Output:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
<td>Image (static or dynamic)</td>
</tr>
</tbody>
</table>

Main Option:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical</td>
<td>makeBackUps</td>
<td>true</td>
</tr>
<tr>
<td>Logical</td>
<td>unitConvert</td>
<td>true</td>
</tr>
</tbody>
</table>
3.3.3.2. **MAKE ISOTROPIC**

This process transforms a 3D volume with any voxel size to a 3D volume with isotropic voxels with a chosen size. If the output voxel size is not given, the minimum of the x-, y- and z-dimensions of the input volume voxel will be chosen.

**Input:**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
<td>Image (static or dynamic)</td>
</tr>
</tbody>
</table>

**Output:**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
<td>Image (static or dynamic)</td>
</tr>
</tbody>
</table>

**Main Option:**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numerical</td>
<td>voxelSize</td>
<td>2 (mm)</td>
</tr>
<tr>
<td>String</td>
<td>interpMethod</td>
<td>‘linear’</td>
</tr>
</tbody>
</table>

**Command-line function:**

MIAKAT_makeIsotropic
3.3.3.3. Brain Extraction

This process uses the FSL toolbox (e.g. [FSL Brain Extraction Tool](https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/BrainExtraction)) to extract the brain from an input 3D image (e.g T1 MRI), creating several output images as follows:

- **brain** - extracted brain volume
- **skull** - estimated skull image.
- **brainMask** - binary brain mask corresponding to estimated brain volume.
- **brainOverlay** - original image with brain segmentation overlaid (meant for visual validation of segmentation)
- **brainPreMask** - intermediate image produced if using the FSLbetPremask option.

**Input:**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
<td>Structural MRI (e.g. T1)</td>
</tr>
</tbody>
</table>

**Main Outputs:**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
<td>Brain MRI</td>
</tr>
<tr>
<td>Image</td>
<td>Brain mask</td>
</tr>
</tbody>
</table>

**Main Options:**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numerical</td>
<td>fracIntThreshold</td>
<td>0.32</td>
</tr>
<tr>
<td>Numerical</td>
<td>vertGradThreshold</td>
<td>0</td>
</tr>
</tbody>
</table>

Command-line function:

`MIAKAT_extractBrain`
3.3.3.4. **Brain Segmentation**

General Description: This process uses a specified tool (default: SPM) to segment the brain MRI image creating a selection of output images:

- grey - brain grey matter image
- white - brain white matter image
- csf - brain csf image.

The above mentioned output images are in the form of probability maps, with a continuous range of values between 0 and 1 where each voxel value shows the partial volume contribution of that tissue to the voxel.

**Input:**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
<td>Skull-stripped MRI</td>
</tr>
</tbody>
</table>

**Outputs:**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
<td>Grey matter probability map</td>
</tr>
<tr>
<td>Image</td>
<td>White matter probability map</td>
</tr>
<tr>
<td>Image</td>
<td>CSF probability map</td>
</tr>
</tbody>
</table>

**Main Options:**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
</table>

**Command-line function:**

MIAKAT_segmentBrain
3.3.3.5. RIGID REGISTRATION TO TEMPLATE

This process rigidly registers (six degrees of freedom) the brain MRI image to a template MRI using (per default) normalised mutual information as cost function. The transformation matrix, obtained from the registration, will be saved in the Analysis data structure and applied to the brain tissue probability maps acquired from the previous process. The default options are set to use SPM co-registration and realignment functions.

Inputs:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
<td>Brain MRI</td>
</tr>
<tr>
<td>Image</td>
<td>Grey matter probability map</td>
</tr>
<tr>
<td>Image</td>
<td>White matter probability map</td>
</tr>
<tr>
<td>Image</td>
<td>CSF probability map</td>
</tr>
</tbody>
</table>

Outputs:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
<td>Brain MRI in MNI space</td>
</tr>
<tr>
<td>Image</td>
<td>Grey matter probability map in MNI space</td>
</tr>
<tr>
<td>Image</td>
<td>White matter probability map in MNI space</td>
</tr>
<tr>
<td>Image</td>
<td>CSF probability map in MNI space</td>
</tr>
</tbody>
</table>

Main Options:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numerical</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Command-line function:

MIAKAT_registerImage
3.3.3.6. **NONLINEAR REGISTRATION OF TEMPLATE**

This process nonlinearly registers the template MRI to the subject’s rigidly register brain MRI and saves the resulting deformation field. The default options are set to use SPM normalisation tool.

**Inputs:**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
<td>Brain MRI in MNI space</td>
</tr>
</tbody>
</table>

**Output:**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS*</td>
<td>Warp field</td>
</tr>
</tbody>
</table>

**Main Options:**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numerical</td>
<td>smoothSrc</td>
<td>8 (mm)</td>
</tr>
<tr>
<td>Numerical</td>
<td>smoothRef</td>
<td>0 (mm)</td>
</tr>
</tbody>
</table>

**Command-line function:**

MIAKAT_registerImage

*AS: Analysis Structure*
3.3.3.7. Define ROIs

This process defines ROIs using *this* examination’s images. ROIs being associated with this exam simply means that they were defined using this exam, not that they will be applied *to* this exam. ROIs to be defined on PET exams should be detailed in the relevant PET exam’s Pipeline. TACs are generated in a separate process, so a given Exam will not in general have a matching set of input/outputs for the ‘Define ROIs’ and ‘Generate TACs’ processes. e.g. some ROIs could be defined on an MRI, some on a baseline PET, and both applied to both baseline and post-dose PET images. For manual ROI delineation, this process effectively just prompts for input, and then stores the details of images specified by the user. The ROI images might well be in a space other than the ‘final space’ that they’ll need to be in in order to be applied to the dynamic image later. This is ordinarily handled by a separate process (‘Make images in final space’), however, if the ROIs specified are actually drawn / supplied in the final space already then it is straightforward. For atlas based definition of ROIs, this process will seek the nonlinear warping parameters determined in an earlier execution of ‘Nonlinear registration of template’, and then apply them to the atlas to transform the atlas into the subject space. Again though, this will not necessarily generate the warped atlas in the ‘final space’ - often (e.g. when doing template/atlas warps to high res MR images) this will generate an atlas ROI image in a subject-based space that can then be fed into a ‘Make images in final space’ process.

Inputs:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS</td>
<td>Warp field</td>
</tr>
<tr>
<td>Image</td>
<td>Brain atlas in the MNI space</td>
</tr>
</tbody>
</table>

Output:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS*</td>
<td>Brain atlas warped to the brain MRI in MNI space</td>
</tr>
</tbody>
</table>

Main Options:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numerical</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Command-line function:

If atlas, MIAKAT_spatialTransformImage
3.3.3.8. **MAKE INTO FINAL SPACE**

This process reslices any specified volume to the final space. "Final space" means that they'll be registered with PET dynamics from which TACs will be extracted. It generally includes the output of the Brain Segmentation process ("masks") and the output of the Define ROIs process ("rois"). This normally involves downsampling high resolution images to match the PET resolution. The final space is by default defined as the space of the Brain MRI image in the MNI space with 2mm voxels. By default, it will use the SPM reslice tool.

**Inputs:**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
<td>Static image in any space</td>
</tr>
</tbody>
</table>

**Output:**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
<td>Static image in the final space</td>
</tr>
</tbody>
</table>

**Main Options:**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>finalSpaceDimMatches</td>
<td>‘MNI 2mm’</td>
</tr>
</tbody>
</table>

**Command-line function:**

none
3.3.3.9. REVIEW ANC FILE

This process checks the existence and content of the ancillary file according to a predefined criteria that can be customized to each study (e.g. if the study does not include blood sampling, checking for existence of blood data can be skipped).

Inputs:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text file</td>
<td>Ancillary file (.anc)</td>
</tr>
</tbody>
</table>

Output:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Result</td>
<td>Pass / Fail</td>
</tr>
</tbody>
</table>

Main Options:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structure</td>
<td>Criterias</td>
<td></td>
</tr>
</tbody>
</table>

Command-line function:

none
This process realigns the PET dynamic data using a frame-to-frame rigid registration algorithm (with mutual information as the cost function). One frame from the PET dynamic data is selected as a reference, and all the other frames are registered to the reference frame. This method has the advantage of being relatively simple and not requiring any additional hardware, but cannot correct for within-frame motion, and its robustness depends on the amount of information available within each PET frame to support robust registration.

In addition, this process will rigidly register the dynamic PET to the MRI of the same subject. By default, the skull-stripped MRI already registered to the MNI space is used (Brain MRI in MNI space) as reference image. The frame-by-frame registration is often more accurate when done with the non-attenuated corrected (NAC) dynamic PET. If some misalignment occurred between the PET and the CT, the attenuation-correction step could have induce edge artefacts that might induce incorrect movement parameters.

### Inputs:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
<td>Dynamic PET image</td>
</tr>
<tr>
<td>Image</td>
<td>Individual MRI (e.g Brain MRI in MNI space)</td>
</tr>
</tbody>
</table>

### Output:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
<td>Motion-corrected registered dynamic PET</td>
</tr>
</tbody>
</table>

### Main Options:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numerical</td>
<td>refFrameNo</td>
<td>16</td>
</tr>
</tbody>
</table>

Command-line function:

MIAKAT_realignDynamic
3.3.3.11. MAKE INTEGRAL IMAGES

This process generates one or more static images where each voxel value is the average value of that voxel over a selected interval of time. The time interval over which the averaging is performed can be specified by the user.

Input:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
<td>Dynamic PET image</td>
</tr>
</tbody>
</table>

Outputs:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
<td>Add Image 1 (static image)</td>
</tr>
<tr>
<td>Image</td>
<td>Add Image 2 (static image)</td>
</tr>
</tbody>
</table>

Main Options:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numerical array</td>
<td>integralLimits</td>
<td>[10  EndOfScan; 0  10; 0  EndOfScan; 30  EndOfScan; 60  EndOfScan]</td>
</tr>
</tbody>
</table>

Command-line function:

MIAKAT_makeIntegralImages
3.3.3.12. **CHECK LR FLIP**

This process evaluates the lateral (left-right) orientation match between PET and MRI images. The cost function value from the rigid registration of the Add Image 1 (output from the ‘Make integral images’ process) to the MRI is compared to the cost function value of the rigid registration of the *flipped* Add Image 1 to the MRI and a decision is taken:

- if the cost function of the non-flipped is better than the flipped, the output is labelled ‘Probably OK’
- if the cost function of the flipped is better than the non-flipped, the output is labelled ‘Probably failed’
- if the cost function are similar (thr: 0.0008), the output is labelled ‘Not determined’

No corrective action is applied if the output is ‘Probably failed’.

**Inputs:**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
<td>Add Image 1</td>
</tr>
<tr>
<td>Image</td>
<td>Individual MRI (e.g Brain MRI in MNI space)</td>
</tr>
</tbody>
</table>

**Output:**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Result</td>
<td>Not checked / Probably failed / Probably OK / Not determined</td>
</tr>
</tbody>
</table>

**Main Options:**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numerical</td>
<td>Threshold</td>
<td>0.0008</td>
</tr>
</tbody>
</table>

**Command-line function:**

MIAKAT_checkLRConsistency
3.3.3.13. **GENERATE TIME ACTIVITY CURVES (TACs)**

This process generates TACs for all regions of interest (ROIs) sets defined in the Options of the process and stores them in the Analysis data structure together with their region volumes.

**Input:**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
<td>Motion-corrected registered dynamic PET</td>
</tr>
</tbody>
</table>

**Output:**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AnalysisData</td>
<td>Time Activity Curves</td>
</tr>
</tbody>
</table>

**Main Options:**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Struct</td>
<td>ROIset</td>
<td></td>
</tr>
</tbody>
</table>

**Command-line function:**

MIAKAT_generateTACs
3.3.3.14. **MAKE INPUT FUNCTION**

The primary output of this process is the parent in plasma input function to be used in subsequent tracer kinetic modelling. For a typical acquisition protocol with both continuous and discrete blood sampling, the continuous and discrete whole blood data are merged to form a whole blood TAC spanning the full duration of the dynamic PET scan. A plasma-over-blood model (POB model) is then fitted to the ratios of discrete plasma and whole blood samples enabling the production of a plasma TAC with the same temporal resolution as the whole blood curve. A parent fraction model (PF model) is also fitted to the sparse parent fraction data to interpolate and extrapolate it before applying it to the plasma TAC to produce a parent in plasma TAC that will be the input function for the kinetic modelling to follow.

In addition, a temporal delay between tissue TAC and parent in plasma TAC is estimated and the parent in plasma TAC is corrected by translating the plasma TAC by the estimated delay.

**Input:**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File</td>
<td>Ancillary file</td>
</tr>
</tbody>
</table>

**Output:**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AnalysisData</td>
<td>Parent in plasma input function</td>
</tr>
</tbody>
</table>

**Main Options:**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>Plasma-over-blood model</td>
<td>'constant'</td>
</tr>
<tr>
<td>String</td>
<td>Parent fraction model</td>
<td>'exp_plus_constant'</td>
</tr>
</tbody>
</table>

**Command-line function:**

MIAKAT_makeInputFunction

**Parent Fraction models available (see Annexe 2 for more details about the models):**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Variable Name</th>
<th>Equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>No model</td>
<td>‘no_metab’</td>
<td>Annexe 2</td>
</tr>
<tr>
<td>One exponential</td>
<td>‘one_exp’</td>
<td>Annexe 2</td>
</tr>
<tr>
<td>One exponential with alpha=1</td>
<td>‘one_exp_fix_alpha_equal_1’</td>
<td>Annexe 2</td>
</tr>
<tr>
<td>One exponential approach to constant</td>
<td>‘exp_plus_constant’</td>
<td>Annexe 2</td>
</tr>
<tr>
<td>One exponential approach to reference region constant</td>
<td>‘exp_plus_ref_con’</td>
<td>Annexe 2</td>
</tr>
<tr>
<td>Two exponentials</td>
<td>‘two_exp’</td>
<td>Annexe 2</td>
</tr>
<tr>
<td>Two exponentials with alpha=1</td>
<td>‘two_exp_fix_alpha_equal_1’</td>
<td>Annexe 2</td>
</tr>
<tr>
<td>Two exponentials approach to reference region constant</td>
<td>‘two_exp_ref_con’</td>
<td>Annexe 2</td>
</tr>
<tr>
<td>Sigmoid 1</td>
<td>‘sigmoid1’</td>
<td>Annexe 2</td>
</tr>
<tr>
<td>Sigmoid 2</td>
<td>‘sigmoid2’</td>
<td>Annexe 2</td>
</tr>
</tbody>
</table>
Plasma-over-blood models available:

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Variable Name</th>
<th>Equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>'linear,'</td>
<td>Annexe 3</td>
</tr>
<tr>
<td>Exponential approach to constant</td>
<td>'exp_approach_to_constant'</td>
<td>Annexe 3</td>
</tr>
<tr>
<td>Constant</td>
<td>'constant'</td>
<td>Annexe 3</td>
</tr>
</tbody>
</table>
3.3.3.15. Kinetic Modelling

Once the blood and image data have been processed to produce a parent in plasma input function, and time activity curves, these can be considered as inputs and outputs for a SISO system (Single Input Single Output) to be modelled. The goal of such a model is to describe the underlying biological system in terms of the parameters of interest such as volume of distribution (V<sub>T</sub>) or non-displaceable binding potential (BP<sub>ND</sub>). The model parameters are estimated from the measured input / output data enabling the calculation of the outcome measure of interest.

Some kinetic modelling techniques replace the need for blood data with data from a reference region (a region of the brain that behaves similarly to the target region, except that it is devoid of the radiotracer’s target and therefore the signal from these region is solely a non-specific component).

Input:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AnalysisData</td>
<td>Time Activity Curves</td>
</tr>
</tbody>
</table>

Output:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AnalysisData</td>
<td>Model parameters</td>
<td></td>
</tr>
</tbody>
</table>

Main Options:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>Model Specification</td>
<td>&quot; (all models)</td>
</tr>
<tr>
<td>String</td>
<td>Reference Region</td>
<td>'cerebellum'</td>
</tr>
</tbody>
</table>

Command-line function:

MIAKAT_kineticModelling

Models available:

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Variant Name</th>
<th>Blood Required</th>
<th>Ref Region Required</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blood Volume</td>
<td></td>
<td>yes</td>
<td>no</td>
<td>Gunn et al. 2001</td>
</tr>
<tr>
<td>One TC Irreversible</td>
<td>zeroBV, fixedBV, fitBV</td>
<td>yes</td>
<td>no</td>
<td>Gunn et al. 2001</td>
</tr>
<tr>
<td>One TC Reversible</td>
<td>zeroBV, fixedBV, fitBV</td>
<td>yes</td>
<td>no</td>
<td>Gunn et al. 2001</td>
</tr>
<tr>
<td>Two TC Irreversible</td>
<td>zeroBV, fixedBV, fitBV</td>
<td>yes</td>
<td>no</td>
<td>Gunn et al. 2001</td>
</tr>
<tr>
<td>Two TC Reversible</td>
<td>zeroBV, fixedBV, fitBV</td>
<td>yes</td>
<td>no</td>
<td>Gunn et al. 2001</td>
</tr>
<tr>
<td>MA1</td>
<td>zeroBV, fixedBV, fitBV</td>
<td>yes</td>
<td>no</td>
<td>Ichise et al. 2002</td>
</tr>
<tr>
<td>Logan</td>
<td>zeroBV, fixedBV</td>
<td>yes</td>
<td>no</td>
<td>Logan et al. 1990</td>
</tr>
<tr>
<td>Patlak</td>
<td>zeroBV, fixedBV</td>
<td>yes</td>
<td>no</td>
<td>Patlak et al. 1983</td>
</tr>
<tr>
<td>SRTM</td>
<td>zeroBV, fixedBV</td>
<td>no for zeroBV, yes for fixed BV</td>
<td>yes</td>
<td>Lammertsma et al. 1996</td>
</tr>
<tr>
<td>SRTM2</td>
<td>zeroBV, fixedBV</td>
<td>no for zeroBV, yes for fixed BV</td>
<td>yes</td>
<td>Wu et al. 2002</td>
</tr>
<tr>
<td>MRTM</td>
<td>zeroBV, fixedBV</td>
<td>no for zeroBV, yes for fixed BV</td>
<td>yes</td>
<td>Ichise et al. 2003</td>
</tr>
<tr>
<td>MRTM2</td>
<td>zeroBV, fixedBV</td>
<td>no for zeroBV, yes for fixed BV</td>
<td>yes</td>
<td>Ichise et al. 2003</td>
</tr>
<tr>
<td>RefLogan</td>
<td>zeroBV</td>
<td>no</td>
<td>yes</td>
<td>Logan et al. 1996</td>
</tr>
<tr>
<td>RefPatlak</td>
<td>zeroBV</td>
<td>no</td>
<td>yes</td>
<td>Patlak et al. 1985</td>
</tr>
</tbody>
</table>
### 3.3.3.16. Parametric Images

Generation of parametric images involves fitting a model to a TAC for each voxel in the image. The process is computationally intensive and vulnerable to noise, but allows visualization of the parameters of interest without assumptions of homogeneity in anatomically defined regions. Parametric images are 3D images of the model parameters themselves and are not spatially normalized through nonlinear registration to a standard space so cross-subject statistical analysis of parameters at a voxel level is not directly achievable with the output(s) of this process.

**Input:**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
<td>Motion-corrected registered dynamic PET</td>
</tr>
</tbody>
</table>

**Output:**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
<td>Parametric images</td>
</tr>
</tbody>
</table>

**Main Options:**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>Model Specification</td>
<td>‘SRTM’</td>
</tr>
<tr>
<td>String</td>
<td>Reference Region</td>
<td>‘cerebellum’</td>
</tr>
</tbody>
</table>

**Command-line function:**

**Models available:**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Variant Name</th>
<th>Blood Required</th>
<th>Reference Region Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>One TC 1K</td>
<td>fitBV</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Logan</td>
<td>zeroBV</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Patlak</td>
<td>zeroBV</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>SRTM</td>
<td>zeroBV</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>SRTM2</td>
<td>zeroBV</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>RefLogan</td>
<td>zeroBV</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>RefPatlak</td>
<td>zeroBV</td>
<td>no</td>
<td>yes</td>
</tr>
</tbody>
</table>
3.4. **VISUALISATION AND QUALITY CONTROL OF THE RESULTS**

By default, MIAKAT is configured so that certain processes expect human validation of the results whereas other processes don’t. In the following paragraph, the results of the test dataset (see Annexe 6.5) are shown process by process. Careful QC of these processes must be undertaken by the user, bearing in mind many factors which are study-specific, such as with brain regions are of particular interest. Precise criteria cannot therefore not be given here for what counts as success, however, some of the following sections do contain some useful tips for selected processes.

### 3.4.1. BRAIN EXTRACTION

**QC:** check if the brain was successfully extracted from the brain. For this, the user can use the slices displayed within the MIAKAT main GUI or (preferably) display the whole volumes in an external viewer (e.g., fslview).

What to do if QC fails: as brain extraction is very subject dependent, failure of this process might happen often. A possibility to fix the brain extraction is provided on the right of the GUI. This fix will run the brain extraction with different values for the fracIntThreshold parameter and display the set of brain extracted images using fslview. The user will report the value of the fracIntThreshold parameter that generates the best brain extraction. The pipeline is reset to the Brain Extraction process and restarted. As most of the processes following the Brain Extraction process are dependent on the Brain Extraction process, the user should be aware that fixing and re-running the pipeline takes time.

![FIGURE 10. BRAIN EXTRACTION OUTPUT](image-url)
3.4.2. **BRAIN SEGMENTATION**

![Figure 11. Brain Segmentation Output](image)

3.4.3. **NON-LINEAR REGISTRATION OF TEMPLATE**

![Figure 12. Non-linear Registration Output](image)
3.4.4. **Define ROIs / Make Images in Final Space**

![Figure 13. Make Images in Final Space Output](image13)

3.4.5. **Review ANC**

![Figure 14. Review ANC Output](image14)
3.4.6. **MOTION CORRECTION**

QC: The output information has three plots and two animated images.

![Motion Correction Output](image)

**FIGURE 15. MOTION CORRECTION OUTPUT**

The top plot informs about the translation parameters along the x-, y- and z-axis estimated for each frame during the Motion Correction procedure. Two dotted lines represents -10mm and +10mm.

The middle plot informs about the rotation parameters along the x-, y- and z-axis estimated for each frame during the Motion Correction procedure. Two dotted lines represents -5 degrees and +5 degrees.

The bottom plot informs about the informational content of each frame, the frames that have been discarded due to too low count (red dots) and the reference frame (green dot) used during the frame-by-frame registration.

The animated gifs on the right hand side display on the top the non-motion corrected dynamic image in the final space and on the bottom the motion corrected dynamic image in the final space.

**Steps to QC this process:**

- Check visually if there is obvious movement in the original dynamic PET image (input) using the top animated gif and/or loading the input image in fslview.
- Check if this movement is attenuated in the motion-corrected dynamic PET image (output)
- Check if there is any outlier in the two plots displaying the translation and rotation parameters, especially in the early frames

**What to do if QC fails:**

- If an early frame doesn’t register correctly and as a consequence produce erratic value for the translation and rotation parameters, try to increase the Option called “frameToRealignThreshold” (def:0.25).
- If all the translation and/or rotation parameters are shifted from the baseline, try to change the reference frame by changing the Option called “refFrameNo” (def: 16).
3.4.7. **Check LR Flip**

3.4.8. **Generate Time Activity Curves**

![Image of a screenshot showing a software interface for generating TACS output.](image)

**FIGURE 16. GENERATE TACS OUTPUT**
3.4.9. **MAKE INPUT FUNCTION**

QCs:

Plasma over blood (top right plot): verify that the model estimated (red line) fits well the plasma over blood data (blue circles)

Parent fraction (bottom right plot): verify that the model estimated (red line) fits well the parent fraction data (blue circles)

Input function (left plot): verify that the data and units make sense and that the shape of the input function corresponds to the type of injection (e.g. bolus)

**FIGURE 17. GENERATE PLASMA INPUT FUNCTION OUTPUT**
3.4.10. KINETIC MODELLING

![Figure 18. Kinetic Modelling Output](image1)

3.4.11. PARAMETRIC IMAGES

![Figure 19. Parametric Images Output](image2)
4. Advanced Information

MIAKAT™ is an open source software consisting of a set of MATLAB functions that are freely available for users to visualise or modify. It also offers the possibility of adding your own MATLAB function to the existing set of functions. We encourage users who produce developments to share these with us and we will attempt to include them in future releases.

There are four different types of functions available in MIAKAT:

- Core functions which perform a certain task (e.g. brain segmentation, generation of TACs, etc.) and can be called from MATLAB command line independently. The name of these functions usually starts with MIAKAT_doThis where doThis shows the purpose of the function (e.g. MIAKAT_segmentBrain, MIAKAT_generateTACs, etc.).

- Framework functions which are one level higher than the core functions, and determine the inputs, options, and outputs of each process depending on the user-specifications that have been set in the AnalysisManager data structure.

- GUI functions which create the graphical interface between the AnalysisManager data structure and the user. They allow a more comprehensive understanding of the status of the analysis pipeline and facilitate the visualisation of the results.

- Utility function which includes the rest of the functions.

4.1. Description of the Main Window

After having analysed your subject, the interface will look as follows:

![FIGURE 20. MIAKAT™ MAIN GUI AFTER LOADING AN ANALYSIS MANAGER DATA STRUCTURE OF A COMPLETED ANALYSIS. 1: ANALYSIS MANAGER STRUCTURE EXPLORER PANEL, 2: PROGRESS PANEL, 3. PROCESS INFORMATION PANEL, 4. QC PANEL]

The following paragraphs describe more in detail the different panels of the interface (Figure 20) and how to interact with the interface in order to export output results.
4.1.1. Analysis Manager Structure Explorer Panel

This panel (Figure 20, Panel 1) allows users to navigate through the AnalysisManager data structure in a hierarchical (tree) format with list of entries representing the fields of the MATLAB™ structure.

![Diagram of Analysis Manager Structure](image)


Each entry, from left to right, has: an icon (+) allowing to expand the field to visualise the sub-fields, an icon specific to the class of the entry (e.g. string, numerical, cell, etc.), the name of the entry and a description of the entry.

- Left-click on the (+) of an entry displays the sub-fields of the entry.
- Right-click on the name or on the description of an entry gives the possibility of editing or copying the entry and it is only available for certain classes (such as strings and digits).
- Right-click on a Process (sub fields of AnalysisManager -> Exams -> Pipeline), gives the possibility to reset the pipeline from the selected process.

4.1.2. Progress Panel

This panel (Figure 20, Panel 2) displays the status of the analysis progress at subject level (left), as well as the progress for each individual examination (right). The left table entitled Subject will have the same number of columns as the number of examinations for a given subject. The right table entitled Exam: XXX (YYY) (e.g. Exam: PET1 (Dynamic PET)) has the same number of columns as the number of processes of a given examination.

The rows are similar for both tables: progress indicates the status of the analysis progress and QC indicates the status of the QC progress.
The cells of both tables are colour coded such that each colour represents a certain progress or QC status:

<table>
<thead>
<tr>
<th></th>
<th>White</th>
<th>Grey</th>
<th>Red</th>
<th>Orange</th>
<th>Green</th>
</tr>
</thead>
<tbody>
<tr>
<td>Progress</td>
<td>Process not done</td>
<td>Process failed</td>
<td>Completed</td>
<td>(Awaiting QC)</td>
<td>Completed</td>
</tr>
<tr>
<td>QC</td>
<td>QC not required</td>
<td>QC not done</td>
<td>QC failed</td>
<td>QC Passed with Caution</td>
<td>QC Passed</td>
</tr>
</tbody>
</table>

4.1.3. PROCESS INFORMATION PANEL

This panel (Figure 20, Panel 3) is informing the user about the specific content of the processes of the workflow. This panel is initially blank and displays content only when the user clicks on a specific process in the Explorer Panel (sub fields of AnalysisManager -> Exams -> Pipeline). The content of the Process Information panel is:

- the inputs (resp. outputs) of the selected process embedded in a Table situated at the top (resp. bottom) of the panel
- some key results situated between the input and output tables.

An example is presented in Figure 22. The panel always displays the input table on the top part of the panel listing the input of the process and the output table on the bottom of the panel listing the output of the process.

![FIGURE 22. EXAMPLE OF PROCESS INFORMATION PANEL CONTENT FOR PROCESS “MAKE INTEGRAL IMAGES”](image-url)
4.1.4. QC PANEL
This panel (Figure 20, Panel 4) is meant to record the comments of the user on the quality of the outputs of a specific process. In addition to the three offered verdicts options proposed by MIAKAT™ (Failed, Passed with Caution and Passed), users can also add notes to explain their choice.

4.2. ANALYSIS MANAGER DATA STRUCTURE
The AnalysisManager data structure contains information regarding the specification and progress of the analysis of data for a given subject. The AnalysisManager data structure is stored as a variable named AnalysisManager, alone in a binary mat-file. This file is typically named appropriately to avoid ambiguity within a study (different name for every subject, e.g. AM_subj001.mat for subject one of the study). The AnalysisManager data structure of a subject has all the information regarding all the analysis processes with their detailed configuration options, location of source data (i.e. images and associated ancillary files), and progress of the analysis pipeline and locations of all outputs. But the actual results (e.g. outcome parameters such as VT or BPND values) themselves are not stored within the AnalysisManager.

The command to load an Analysis Manager file in the MATLAB command window is either load or processAnalysisManagerFile.

4.3. ANALYSIS DATA STRUCTURE
The Analysis data structure contains information regarding processes which have been applied to a given examination. The Analysis data structure is stored as a variable named Analysis, alone in a binary mat-file. This file is typically named appropriately to avoid ambiguity within a study and within a subject (different name for every subject, e.g. subj001_PET001.mat for PET examination 1 of subject one of the study) as several Analysis data structure might exist for a given subject. The AnalysisManager data structure contains references to all of those Analysis mat-files. The Analysis mat-file does not contain any information regarding processes yet to be completed, or any pipeline structure. It does contain the outputs of completed processes for a certain examination. For example, for output images, the Analysis data structure contains information only about the low-level functions used to generate the outputs, options specified, and locations of the images. Numeric results (e.g. generate TACs, kinetic modelling) are stored within the Analysis structure. The top level of the Analysis structure has various fields, which clump together classes of processes. These fields are added as processes are performed.

The command to load an Analysis file in the MATLAB command window is processAnalysisFileInput.

4.4. KEY CONCEPTS

IMAGE SPACE
Images (PET or MRI) are often characterise by the image space they “live” in. This term can be define as how the imaged object (e.g. brain) is orientated in the imaged space (e.g. FOV of the scanner). During the brain pipeline described in paragraph 3.4, one of the objective is to register the atlas image and the dynamic PET image so that they are in fine in the same image space, which is the pre-requisite to use the atlas to define ROIs in the dynamic PET image.
Several image spaces have been predefined and are embedded in the Analysis Manager data structure. Users can find the list of image spaces below.

- 'MRI T1 orig'
- 'Isotropic T1'
- 'MNI T1 full res'
- 'PET orig'
- 'Isotropic PET'
- 'Final space'

During the process, these Image spaces will be automatically filled with the information specific for this subject. For example, when the first process of the MRI examination runs (Preprocess images), the voxel size and the volume dimensions are extracted from the original image and recorded respectively in the AnalysisManager.Imagespaces(1).voxDim and AnalysisManager.Imagespaces(1).dim fields.

4.5. USING MANUAL ROIS

MIAKAT™ allows the users to add region of interests that have been defined outside the software to the workflow. For example, users can add manually delineated regions or labels that have been defined with another software (FreeSurfer, etc...).

An example of manually delineated ROI image file and label text file can be found with the dataset in the MRI folder.

To include a set of region in the MIAKAT workflow, two files have to be prepared:

- An image file containing the spatial extent of the regions. This image file should follow certain rules:
  o Rule 1: this file should only contain integers
  o Rule 2: each integer correspond to one and only one region
  o Rule 3: the image format should be one of the accepted format (see paragraph 3.2.1)
  o Rule 4: the region of interest image should exist in one of the following spaces:
    ▪ MRI original space
    ▪ MRI – MNI 1mm
    ▪ MRI – Final space
    ▪ PET - Final space

- A text file that does the correspondence between the integers found in the image file and the name of the region of interest. This file should contain one integer and one string (name of the region surrounded by quotes) per row (see example in dataset) separated by a space or tab. The number of rows in the text file should be exactly similar to the number of integers in the ROI image. MIAKAT will issue an error if a discrepancy is found between text file and image file. The name of the text file should be the same as the image filename and should be stored in the same folder as the image file.
Case study 1: in addition to the atlas definition, the user want to draw the substantia nigra on the MRI image of one subject.

- Using a third party software (Analyze, ITKsnap, etc...), the original MRI is loaded. The substantia nigra is defined manually and the label image is saved in an appropriate format (e.g. SN.nii). A text file (SN.txt) is created in the same folder as the image with as only content:
  1 ‘SubstantiaNigra’

- When customizing the Analysis Manager using the Wizard, the users should select:
  o ‘Atlas + Manual’ in the field ROI
  o ‘Original MRI’ in the field Manual ROI

- During the pipeline run, at process ‘Define ROIs’ of the examination MRI, a dialog box will pop up asking the user to select the ROI image.

Case study 2: the new tracer has been analysed in several subjects using MIAKAT. It shows abnormal heterogeneity in the cerebellum that we are interested to study more in details. Manual ROI should be drawn using the PET image on this regions.

- Using a third party software (Analyze, ITKsnap, etc...), the PET add generated by MIAKAT during the first analysis (for example, MNI_raiSubject000005_PET1_AC_add_0_90) image is loaded. The three regions of the cerebellum with different uptake (low, medium, high) are defined manually and the label image is saved in an appropriate format (e.g. HeteroCereb.nii). A text file (HeteroCereb.txt) is created in the same folder as the image with as only content:
  1 ‘LowUptake’
  2 ‘ModerateUptake’
  3 ‘HighUptake’

- When customizing the Analysis Manager using the Wizard, the users should select:
  o ‘Atlas + Manual’ in the field ROI
  o ‘PET – Final space’ in the field Manual ROI

- During the pipeline run, at process ‘Define ROIs’ of the examination PET1, a dialog box will pop up asking the user to select the ROI image.
5. BIBLIOGRAPHY


6. ANNEXES

6.1. ANC FILE DESCRIPTION

Associated with any PET image there are many ancillary data. These data include contextual information about how, where and when the experiment was performed, along with data that are required for analysis such as frame times and measured blood radioactivity levels. Often these data are recorded in disparate sets of files of various formats, but in order to succinctly gather, store and supply them to MIAKAT, we define a convenient text-based file format: the PET ancillary data (.anc) file format.

The ancillary data to be recorded in this file fall into the following categories:

- Summary info
- Radiochemistry
- Time data
- Metabolite data
- Plasma data from discrete samples
- (Whole) blood data from discrete samples
- (Whole) blood data from continuous sampling
- Blood glucose measurements

The following is an example of an .anc file (also included in the test data set that can be downloaded from the MIAKAT website).

```
########################################
# File revision history                #
########################################
Created: 11:21 10 Nov 2008 template
Modified: 15:17 05 Jul 2010 GES53673
Modified: 20:19 14 Jul 2010 ges53673
Modified: 14:45 29 Oct 2015 ccoello
########################################
# Summary info                         #
########################################
Info.programme = ABC
Info.study = ABC123456
Info.subject = ABC123456_000005
Info.visit = 1_PHNO_00083
Info.examination = 1_PHNO_00083
Info.contactName = Graham Searle
Info.contactEmail = graham.searle@imanova.co.uk
Info.centre = CIC, London, UK
Info.tomograph = PET/CT2 (Biograph TruePoint 6 CT45544)
Info.anesthesia = ""
Info.bodyPart = ""
Info.Tracer.name = PHNO
Info.Tracer.isotope = C11
Info.Tracer.MW = 247
Info.Tracer.injectionType = bolus
Info.Pharmaceutical.name = ""
Info.Pharmaceutical.doseAmount = NaN
Info.Pharmaceutical.doseUnits = mg/kg
Info.Pharmaceutical.doseRegimen = n/a
Info.Pharmaceutical.doseTime = NaN
Info.Pharmaceutical.doseTimeUnits = min
Info.SubjectData.species = human
Info.SubjectData.strain = n/a
Info.SubjectData.population = HV
Info.SubjectData.bodyWeight = 105.7
Info.SubjectData.bodyWeightUnits = kg
Info.SubjectData.age = 49
Info.SubjectData.ageUnits = years
Info.SubjectData.gender = male
```
Radiochem.injectedRadioactivity = 123.66
Radiochem.injectedRadioactivityUnits = MBq
Radiochem.injectedMass = 0.5
Radiochem.injectedMassUnits = ug
Radiochem.specificRadioactivity = 61.0967
Radiochem.specificRadioactivityUnits = GBq/umol
Radiochem.minPurity = 100
Radiochem.minPurityUnits = fraction

Time.scanDate = 19 May 2010
Time.scanDateUnits = dd Mmm yyyy
Time.scanStart = 11:32:16
Time.scanStartUnits = hh:mm:ss
Time.injectionStart = 11:32:15
Time.injectionStartUnits = hh:mm:ss
Time.injectionEnd = 11:32:15
Time.injectionEndUnits = hh:mm:ss
Time.FrameTimes.labels = {frameStart, frameEnd}
Time.FrameTimes.units = {s, s}
Time.FrameTimes.values =
[0 15
15 30
30 45
45 60
60 75
75 90
90 105
105 120
120 180
180 240
240 300
300 420
420 540
540 660
660 780
780 900
900 1200
1200 1500
1500 1800
1800 2100
2100 2400
2400 3000
3000 3600
3600 4200
4200 4800
4800 5400]

Plasma.freefraction = 0.32
Plasma.Data.decayCorrected = true
Plasma.Data.decayCorrectionTime = 11:32:16
Plasma.Data.type = radioactivity
Plasma.Data.labels = {sampleStartTime, sampleDuration, activity}
Plasma.Data.units = {s, s, kBq/ml}
Plasma.Data.values =
[344 14 1.797709
613 14 1.930659
905 12 1.797653
1220 17 1.5779266
1518 10 1.422872
1800 14 1.449933
2404 9 1.2503189
3090 24 1.1643844
3673 9 1.119016
4202 57 0.0128106
4842 10 0.995991
5410 40 0.9519462]
# Metabolite

Metabolite.Data.type = plasmaParentFraction
Metabolite.Data.labels = {sampleStartTime, sampleDuration, parentFraction}
Metabolite.Data.units = {s, s, fraction}
Metabolite.Data.values =

\[
\begin{array}{ccc}
344 & 14 & 0.4727496 \\
613 & 14 & 0.2965415 \\
905 & 12 & 0.2244061 \\
1220 & 17 & 0.20639742 \\
1800 & 14 & 0.15137567 \\
3090 & 24 & 0.13739121 \\
4202 & 57 & 0.13872553 \\
\end{array}
\]

# Blood - discrete

Blood.Discrete.haematocrit = NaN
Blood.Discrete.bloodDensity = NaN
Blood.Discrete.bloodDensityUnits = g/ml
Blood.Discrete.Data.decayCorrected = true
Blood.Discrete.Data.decayCorrectionTime = 11:32:16
Blood.Discrete.Data.type = radioactivity
Blood.Discrete.Data.labels = {sampleStartTime, sampleDuration, activity}
Blood.Discrete.Data.units = {s, s, kBq/ml}
Blood.Discrete.Data.values =

\[
\begin{array}{ccc}
344 & 14 & 1.527497 \\
613 & 14 & 1.357687 \\
905 & 12 & 1.214242 \\
1220 & 17 & 1.0765655 \\
1518 & 10 & 1.0003059 \\
1800 & 14 & 0.9304565 \\
2404 & 9 & 0.86777416 \\
3090 & 24 & 0.77890135 \\
3673 & 9 & 0.74512136 \\
4202 & 57 & 0.69666605 \\
4842 & 10 & 0.66249005 \\
5410 & 40 & 0.64035065 \\
\end{array}
\]

# Blood - continuous

Blood.Continuous.withdrawalRate = NaN
Blood.Continuous.withdrawalRateUnits = ml/min
Blood.Continuous.tubingType = ""
Blood.Continuous.tubingLength = NaN
Blood.Continuous.tubingLengthUnits = mm
Blood.Continuous.Data.decayCorrected = true
Blood.Continuous.Data.decayCorrectionTime = scan start
Blood.Continuous.Data.type = radioactivity
Blood.Continuous.Data.labels = {time, activity}
Blood.Continuous.Data.units = {s, kBq/ml}
Blood.Continuous.Data.values =

\[
\begin{array}{cc}
0 & 0.15121 \\
1 & 0.10086 \\
2 & 0.10092 \\
3 & 0.10098 \\
4 & 0.15155 \\
5 & 0.20218 \\
6 & 0.10115 \\
7 & 0.30362 \\
8 & 0.10126 \\
9 & 0.152327 \\
10 & 0.05066 \\
11 & 0.304139 \\
12 & 0.304312 \\
13 & 0.152327 \\
14 & 0.0508043 \\
15 & 0.101667 \\
16 & 0.101725 \\
17 & 0.050891 \\
18 & 0 \\
19 & 0.101898 \\
20 & 0.305872 \\
21 & 0.204029 \\
\end{array}
\]
6.1.1. FILE DESCRIPTION
The ancillary data file is a plain text (ASCII) file with the extension ‘.anc’. The file format is designed to facilitate both human- and machine-readability. The contents of the file are separated into distinct sections, corresponding to the data categories listed above. Comments and section delimiters within the file are denoted by a leading hash (’#’) symbol. Data are detailed within the file as ’data name = data value’ pairs, with the name specifications being themselves structured much like a C or Matlab structure, so that a data-defining line in the file could read, for example:

```
DataType.DataSubType1.DataSubType2.dataName = dataValue
```

The data name may contain only letters, digits, periods and underscores. The first character and each character after a period must be a letter.

There are three data value delimiter sets:

- [ ] for numeric arrays,
- "" for text strings, and
- {} for lists of text strings

which optionally enclose data value specification text.

A data item specification begins with the data item’s name, followed by the equals sign. Following the equals sign, the data item specification is deemed complete only when

- some non-whitespace text has been found after the equals sign, and
- a text line is terminated with no outstanding unclosed data value delimiters.

Thus,

- No more than one data item definition can be contained on each line of the file.
- If a data-specifying line contains, for example, only the text ’Info.notes ’ then the data specification continues to following lines until the data value text is found.

No data value field can be left empty – this will lead to parsing errors. Instead, appropriate values indicating that a value is unknown or not applicable must be used.

For a more detailed .anc file format specification document please contact the MIAKAT team.
6.2. PARENT FRACTION MODELS

One parameter: $\beta$
- MIAKAT name: `one_exp.fix.alpha.equal.1`
  \[ y = \exp^{-\beta x} \]
- MIAKAT name: `constant`
  \[ y = \beta \]

Two parameters: $\alpha, \beta$
- MIAKAT name: `one_exp`
  \[ y = \alpha \exp^{-\beta x} \]
- MIAKAT name: `exp.plus.constant`
  \[ y = \alpha \exp^{-\beta x} + 1 - \alpha \]
- MIAKAT name: `two_exp.ref.con`
  \[ y = \alpha \exp^{-\beta x} + (1 - \alpha) \exp^{-\text{wash } x} \]
  wash: reference tissue washout - plasma washout
- MIAKAT name: `linear`
  \[ y = \alpha x + \beta \]

Three parameters: $\alpha, \beta, \gamma$
- MIAKAT name: `exp.approach.to.constant`
  \[ y = \alpha (1 - \exp^{-\beta x}) + \gamma \]
- MIAKAT name: `two_exp.fix.alpha.equal.1`
  \[ y = \alpha \exp^{-\beta x} + (1 - \alpha) \exp^{-\gamma x} \]
- MIAKAT name: `sigmoid1`
  \[ y = (1 - \frac{x^3}{x^3 + 10^x})\beta + \gamma^2 \]
- MIAKAT name: `sigmoid2`
  \[ y = \frac{(1 - \frac{x^3}{x^3 + 10^x})\beta + \gamma}{1 + \gamma} \]

Four parameters: $\alpha, \beta, \gamma, \delta$
- MIAKAT name: `two.exp`
  \[ y = \alpha \exp^{-\beta x} + \gamma \exp^{-\delta x} \]
6.3. **Plasma-Over-Blood Models**

**One parameter:** $\beta$
- MIAKAT name: *constant*

$$y = \beta$$

**Two parameters:** $\alpha, \beta$
- MIAKAT name: *linear*

$$y = \alpha x + \beta$$

**Three parameters:** $\alpha, \beta, \gamma$
- MIAKAT name: *exp_approach_to_constant*

$$y = \alpha(1 - \exp^{-\beta x}) + \gamma$$
6.4. LICENCE

ACADEMIC
Non-Exclusive Software Licence Agreement

The MIAKAT™ software as may from time to time be updated by Imanova Limited (“Imanova”), ("the Software") has been developed by Imanova and certain contributing researchers and other institutions ("the Developers") and is made available by Imanova for your internal, non-profit use. Imanova and the Developers allow the named individual that has been provided with the MIAKAT™ software (“you”, “your”) to run, display and modify the Software on the following terms:

1. Conditions of Use

1.1. The Software remains with you and shall not, without the express permission of Imanova, be published, distributed, reproduced, modified, transmitted or otherwise transferred in any form or by any means, electronic or mechanical, or made available to anyone other than you.

1.2. If, through the use of the Software, you: make any results available to other academic researchers for non-profit research purposes; or publish any results, you agree to acknowledge the contribution Imanova, the Software and any modifications made to your research and to cite appropriate references in your publications.

1.3. You agree to keep in the Software and any modifications to the Software, the copyright, trademark, or other notices pertaining to the Software as provided by Imanova.

1.4. You are not permitted under this licence to use the Software for commercial purposes, which shall be defined as use for which any financial return is received, directly or indirectly, including: integration of all or part of the source code or the Software into a product for sale or license by or on your behalf to third parties; use of the Software or any derivative of it for research with the final aim of developing software or non-software products for sale or license to a third party; or use of the Software to provide any service to an external organisation or person for which payment is received. If you wish to use the Software for commercial purposes, including fee-based service projects or studies or projects funded or part funded by a commercial sponsor, you will need to execute a separate licensing agreement and pay a fee. To obtain details for a commercial licence please contact Imanova: +44 (0) 208 008 6000.

1.5. Imanova reserves the right to inspect and have access to any premises (and to the computer equipment located there) at or on which the Software is being kept or used, and have access to any records kept in connection with this licence, for the purposes of ensuring that you are complying with the terms of this licence, provided that Imanova provides you with reasonable notice of such inspections, which shall take place at reasonable times.

1.6. You acknowledge that the Developers, Imanova and its licensees (the “Imanova Parties”) may develop modifications to the Software that may be substantially similar or functionally equivalent to your modifications and improvements, and that the Imanova Parties shall not be constrained in any way by you in their use or management of such modifications; if you obtain patent protection for any modification or improvement to the Software you agree
not to allege or enjoin infringement of your patent by the Imanova Parties obtaining modifications or improvements to the Software from Imanova.

1.7. Where you choose to provide Imanova with feedback on the use or development of the Software, including by provision of any modifications to, improvements upon and/or derivative works of, the Software (“Feedback”), the Imanova Parties shall have a perpetual, irrevocable, royalty-free, worldwide right to make, have made, sell, offer for sale, import, use, reproduce, distribute, prepare derivative works of, publicly perform and publicly display such Feedback in any media now known or hereafter known, whether for commercial or non-commercial purposes and including in any release by Imanova of the Software.

2. Intellectual Property Rights

2.1. As between you and Imanova, the Software remains the property of Imanova. You shall have no rights in or to the Software other than the right to use it in accordance with the terms of this licence.

3. As-Is Warranty, Limitation of Liability and Indemnification

3.1. Any risk associated with your use of the Software is with you and your institution. The Software is experimental in nature and is made available under this licence solely for non-commercial use as a research courtesy "AS IS," without obligation by Imanova to provide accompanying services or support.

3.2. Imanova expressly disclaims any and all warranties, representations, terms and conditions regarding the Software, whether express or implied, including but not limited to those pertaining to non-infringement, accuracy, or fitness for a particular purpose or use under specific conditions.

3.3. Neither Party excludes or limits its liability for: (a) death or personal injury caused by its negligence, or that of its employees, agents or sub-contractors; (b) fraud or fraudulent misrepresentation by it or its employees; (c) breach of any obligations as to title implied by section 2 of the Supply of Goods & Services Act 1982; or (d) any other liability to the extent the same may not be excluded or limited as a matter of law.

3.4. Imanova shall not be liable under any circumstances for any losses or damages directly or indirectly incurred under or in connection with this licence, whether consequential or otherwise, and whether arising in contract, tort (including negligence) or otherwise, including but not limited to loss of profits, loss of business opportunity, loss of goodwill or loss or corruption of data.

3.5. You agree to indemnify, defend and hold harmless Imanova, the Developers and its respective officers, employees and agents from and against any and all third party claims, damages and liabilities (including claims for negligence) arising directly or indirectly from the use of the Software or the sale of any products based on the Software.
4. Other

4.1. Imanova may terminate this licence at any time if you fail to comply with any of its terms. On such termination, you must destroy all copies of the Software you received or made under this licence. Clauses 1.5, 1.6, 2 and 3 above shall survive termination of this agreement.

4.2. This agreement and any dispute or claim arising out of or in connection with it or its subject matter or formation (including non-contractual disputes or claims) shall be governed by and construed in accordance with the laws of England and Wales. Each party irrevocably agrees to submit to the exclusive jurisdiction of the courts of England and Wales, save that nothing shall prevent either party from seeking injunctive or other equitable relief in any court of competent jurisdiction.
6.5. **Test Dataset**

This dataset is available to download at [www.miakat.org](http://www.miakat.org).

6.5.1. Dataset Content

The archive downloaded from the MIAKAT website contains the following files:

- **Subject000005_MRI.img/.hdr**: MRI T1 MPRAGE image in NIfTI Pair format
- **Subject000005_PET1.img/.hdr**: [11C]PHNO dynamic PET image (corrected for attenuation)
- **Subject000005_PET1_NAC.img/.hdr**: [11C]PHNO dynamic PET image (not corrected for attenuation)
- **Subject000005_PET1.anc**: an ancillary file (text file) containing the blood and plasma data, time frames, metabolite fraction and other information related to the PET examination
  - One PET scan
  - Blood models:
    - Plasma-over-Blood model: constant
    - Parent fraction model: exponential plus constant (see Annexe 2.)
  - Kinetic models:
    - SRTM with fixed BV correction (5%), reference region Cerebellum without mask
    - One TC reversible with fit BV correction,
    - Two TC reversible with fit BV correction
  - Parametric images:
    - SRTM with no BV correction, reference region Cerebellum without mask

6.5.2. Running the Dataset

The procedure to run this dataset is as follows:

a) Uncompress the file testData_PHNO.zip in a folder (where you have write permission)
   - `cd /home/username/Documents/`
   - `unzip testData_PHNO.zip`

b) Launch MATLAB and type MIAKAT in the command window of MATLAB

c) Click on File > New From Wizard

d) When the Wizard GUI appears (Figure 3), you would have to customize the default template with the following analysis specifications:
   - One PET scan, one MRI scan
   - Blood models:
     - Plasma-over-Blood model: constant
     - Parent fraction model: exponential plus constant (see Annexe 2 for more details)
   - Kinetic models:
     - SRTM with fixed BV correction, reference region Cerebellum
     - One TC reversible with fit BV correction,
     - Two TC reversible with fit BV correction
     - MA1 with fixedBV and fitBV
   - Parametric images:
     - SRTM, reference region Cerebellum

For the kinetic models, hold the Ctrl key as you click to select several models. For the blood volume correction, as both fixed and fitted are asked, use the All option to run all variants of the same model. Your Wizard should look like [FIGURE 23](#) before you move to the next step.
e) Click on OK. The default Analysis Manager Template will be modified with the specification you entered and loaded in the MIAKAT.

f) Click on Pipeline > Run or on the icon

g) One the first window,
   - Location: select the location where the test dataset has been unzipped as the location of the Analysis Manager (e.g. /home/username/Documents/PHNO/)
   - Filename: give a name to your Analysis Manager file (PHNO_testDataset)
   - Subject: give a subject number (001)
   - The other fields are optional

h) On the next window,
   - select the location of the MRI T1 image (/home/username/Documents/PHNO/1_MRI_00499/PD/Subject000005_MRI.img) in the field at the bottom of the screen called

---

FIGURE 23. WIZARD CONFIGURED FOR THE PHNO TEST DATASET

FIGURE 24. FIRST WINDOW THAT REQUIRES THE USER'S INPUT WHEN STARTING THE PIPELINE
Input images information (FIGURE 25). Once selected, the colour of the Input images information field turns green.

- In addition, the other fields (Exam ID, Exam folder and Analysis filename) in the panel Examination information are automatically filled. Click OK when you are finished.

![FIGURE 25. SPECIFICATION OF THE EXAMINATION DETAILS: START BY SELECTING THE IMAGE IN THE BOTTOM FIELD. THE FIELDS ABOVE WILL THE POPULATED AUTOMATICALLY AS MUCH AS POSSIBLE. THEN MANUALLY FILL THE FIELDS IN THE PANEL EXAMINATION INFORMATION](image)

i) Repeat the procedure with the third window selecting the attenuation-corrected dynamic PET image (1_PHNO_00083/PD/Subject000005_PET1.img) and the non-attenuation-corrected dynamic PET image (1_PHNO_00083/PD/Subject000005_PET1_NAC.img). Click OK when you are finished

j) The workflow starts and the user can follow the different stages of the analysis on the progress GUI (Figure 7). Depending on your machine, the entire analysis duration would be between 30min and one hour.

k) Once done successfully, the main GUI will reload with the AnalysisManager data structure and the users will be able to proceed to the visualisation and quality control of the results.