Manual for CGITA

Last updated: July, 2012 by Yu-Hua Dean Fang, PhD. (fang.phd@gmail.com)

Table of content

1. Introduction

i. CGITA Background

Chang-Gung Image Texture Analysis toolbox, CGITA, is a software package for texture analysis of molecular images. Originated from Chang-Gung Memorial Hospital (CGMH), Taiwan, it is offered to the molecular imaging community as an open-source, free-for-academic toolbox.

We decided to make CGITA open-source and free for the following reasons:

1. We wish to promote the use of texture analysis for molecular images. At the present stage, there are not any free software for such purposes, to our knowledge. Therefore, we are hoping CGITA can help interested investigators to analyze their image data and evaluate the usefulness of texture analysis.
2. We hope to recruit interested developers to collaborate on the improvement of CGITA. By make the source code available, people around the world may examine the code of CGITA and help us identify the bugs. Also we would look forward to collaborators' contributing new image reading, segmentation and texture analysis functions.
3. We are hoping to use CGITA as a way to build strong collaboration between investigators. For example, different groups in different countries may be interested in the same disease. By using CGITA as a common software environment, data from different institutions can be compared and evaluated more objectively. Such efforts may stimulate the growth of molecular image texture analysis.

Implementation of CGITA began at 2011 in the Molecular Imaging Center, CGMH. In June 2012, the google project website of CGITA was set up to host it as an open-source project at **http://code.google.com/p/cgita/**.

CGITA was initially developed by Yu-Hua Dean Fang, PhD. (fang.phd@gmail.com). In the future, we hope to recruit more developers world-wide to collaborate on the CGITA project.

ii. What you shall know before you use CGITA

- First of all, please carefully read the terms of CGITA at: http://code.google.com/p/cgita/wiki/Terms_CGITA
- CGITA is free for academic use. If you want to use it for commercial purposes, you must obtain permission from the original developer Yu-Hua Dean Fang (fang.phd@gmail.com).
- CGITA is not intended for clinical use, either for diagnostic or therapeutic purposes.
- CGITA is based on MATLAB. But a MATLAB license is not required because we do offer a standalone executable of CGITA on the website.

- If you use CGITA for research purposes, please specifically cite the CGITA paper in your work. You may find the citation at http://code.google.com/p/cgita/.
- CGITA is designed for but not limited to PET images, PET. In principle, images conformed to DICOM shall be able to be processed by CGITA. However, as there are variations in the DICOM images, we do not guarantee that CGITA is able to process all DICOM images.
- We currently do not have any plans to commercialize CGITA. However, to be honest, we cannot guarantee how long CGITA will remain as an open-source project.
- We will try to offer technical support but cannot promise how promptly we can get a technical problem solved, due to the limited man power to maintain this project. All we can say is that we will try our best.
- **The list of current textural features can be found under the CGITA directory *CGITA\docs\texture_summary.xlsx***

iii. What you shall prepare for before you use CGITA

- If you wish use CGITA under MATLAB, you need MATLAB and Image Processing Toolbox. It is highly recommended that you use a 64-bit MATLAB for CGITA.
- If you want to use CGITA as a standalone application, you need to install MATLAB Component Runtime (MCR) first. You can download it at http://www.mathworks.com/products/compiler/mcr/index.html.
- You will then have to download and set up CGITA.
- You will need to prepare for your images, saved as DICOM format. All images must be under the same root folder. However, they do not necessarily have to be under the same subfolder.
- You have to decide how you would like to delineate the tumor contour. Currently, CGITA does not offer a ROI-drawing functionality. It does offer segmentation based on region-growing and fuzzy C-mean. If you want to draw the ROI, you need to do it on other software and then save it into DICOM-RT format. CGITA also supports PMOD-drawn ROIs.
- If you want to use a batch mode to analyze a list of subjects, there are special requirements as for how the data have to be organized. See the section of ' Using CGITA in the batch mode' for more details.

iv. Terms of CGITA usage

1. CGITA is an open-source project hosted on the world-wide web. Interested users shall download it through its official website. Users must not distribute the source code without the permission of the original developer, Yu-Hua Dean Fang, or share it with others by transferring the program on their own.
2. CGITA is not for clinical purposes, including diagnostic and therapeutic application. Users assume all risk.
3. CGITA is free only for academic research. Users who wish to apply CGITA for commercial or

business purposes must obtain the written permission from the original developer Yu-Hua Dean Fang.

4. CGITA users must properly and spontaneously cite the reference paper of CGITA in their publication or any kind of printed material.

5. CGITA users may not include any function distributed in the CGITA package in their own software package without written permission from the original developer Yu-Hua Dean Fang.

6. If a user wishes to use CGITA to analyze clinical data for research purposes, it is the user's responsibility to ensure such research comply with ethnics and regulations in the user's institution and country. The CGITA developers do not take any responsibility in the research use of CGITA.

v. Citing CGITA in your work

We are currently preparing for a manuscript to be submitted.

vi. Contact information of CGITA developers

Yu-Hua Dean Fang, PhD. (fang.phd@gmail.com)

2. Using CGITA

i. Setting up CGITA

There are two ways to use CGITA:

(1) Under MATLAB: It is highly recommended that you use CGITA under MATLAB, if you have a MATLAB license. You may run, modify and even debug CGITA much more easily under MATLAB, compared to the standalone application version.

(2) Under MATLAB Compiler Runtime (MCR): If you do not have a MATLAB license, you need to run CGITA as a standalone application. However, the standalone CGITA still requires a MCR installed on your computer. You can download the standalone CGITA online. To obtain MCR you can go to http://www.mathworks.com/products/compiler/mcr/index.html Note: The CGITA standalone version is currently only available on Windows.

- To set CGITA up under MATLAB:

1. Go to http://code.google.com/p/cgita/downloads/list and download the latest version of CGITA.
2. Uncompress the zip file.
3. Open MATLAB. Go to 'File' -> 'Set path' -> 'Add with subfolders'. Then select the folder that you uncompressed CGITA to on your computer. Press OK. Press save and exit this window.
4. Under the MATLAB command window, type 'CGITA_GUI' and enter. If the CGITA GUI pops up, the setup should have been properly done.

- To set up CGITA as standalone application:

1. First make sure you have installed MCR. Go to http://www.mathworks.com/products/compiler/mcr/index.html and download it for your platform. Install MCR on your computer.
2. Download the executable version of CGITA on http://code.google.com/p/cgita/downloads/list
3. Uncompress the zip file
4. Execute the .exe file of CGITA.

ii. Using CGITA in the GUI mode under MATLAB

(0) Open CGITA:

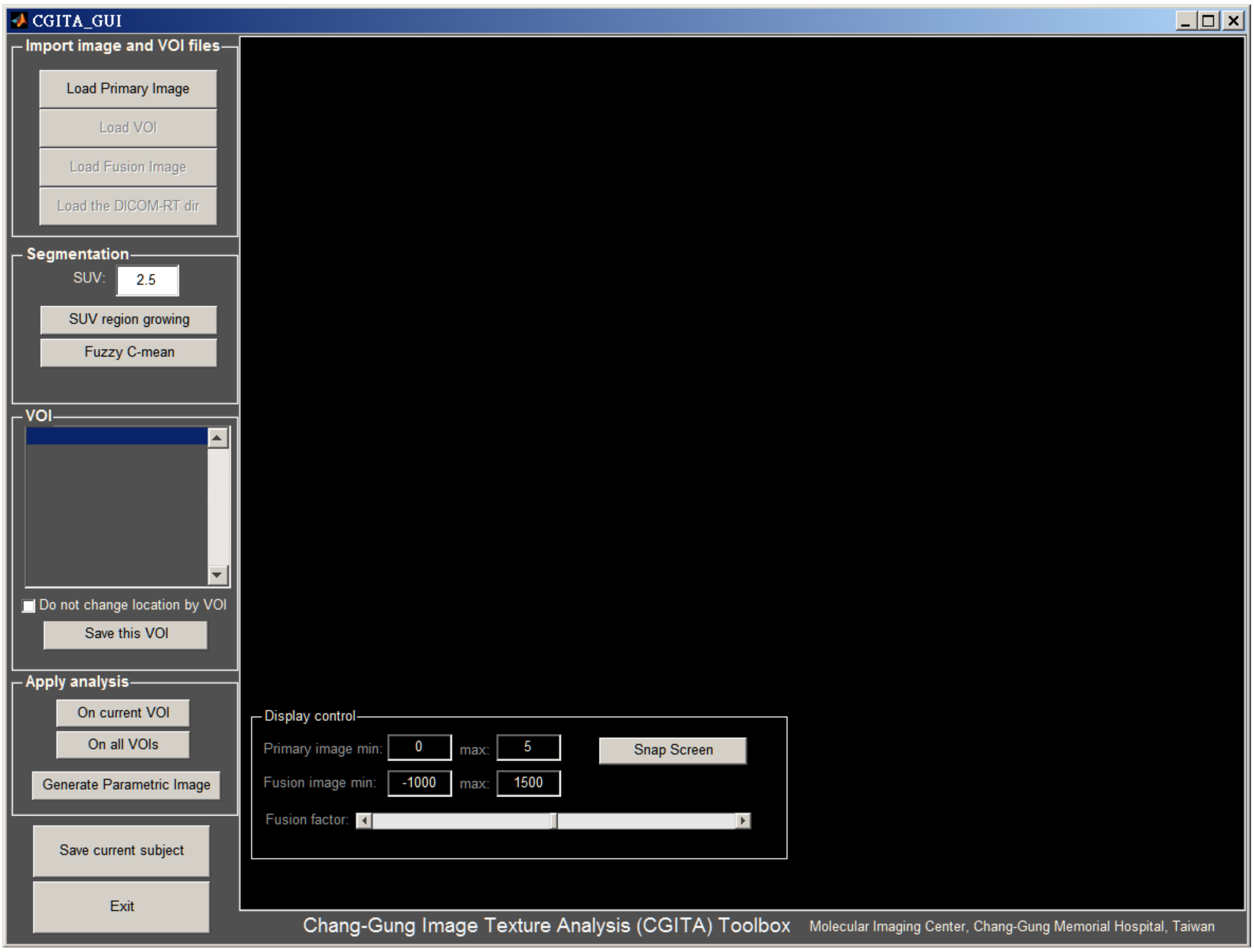
There are a few ways to open CGITA:

a. Under MATLAB command window, type CGITA_GUI and enter.

b. Use MATLAB editor to open CGITA_GUI.m. Then go to 'Debug' -> 'Run CGITA_GUI.m'.

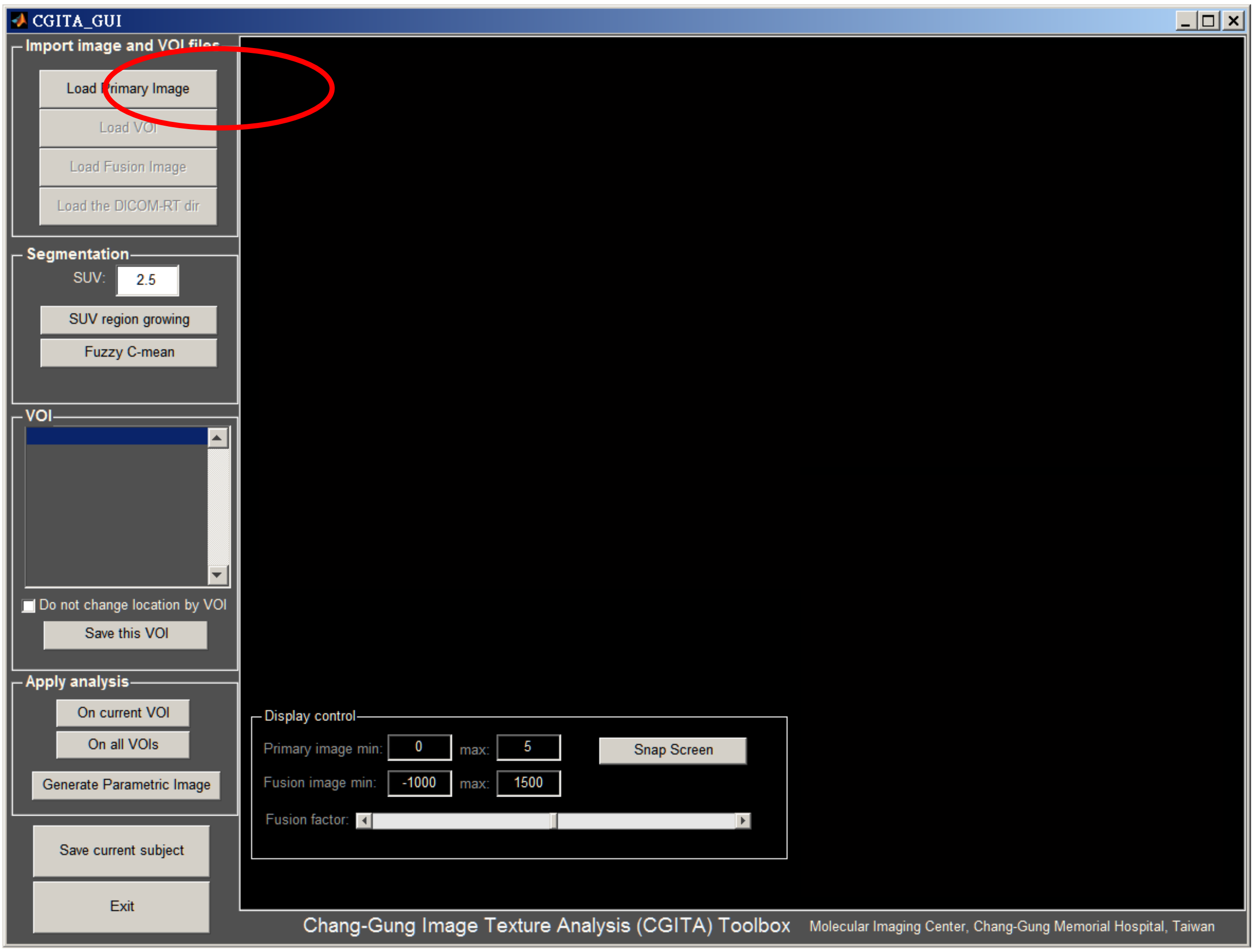
c. Under MATLAB command window, type 'guide' and enter. Open 'CGITA_GUI.fig'. Then go to 'Tools' -> 'Run'.

After any of the previous three actions, you shall see a window like this:



(1) Image reading

Click the 'Load Primary Image'.



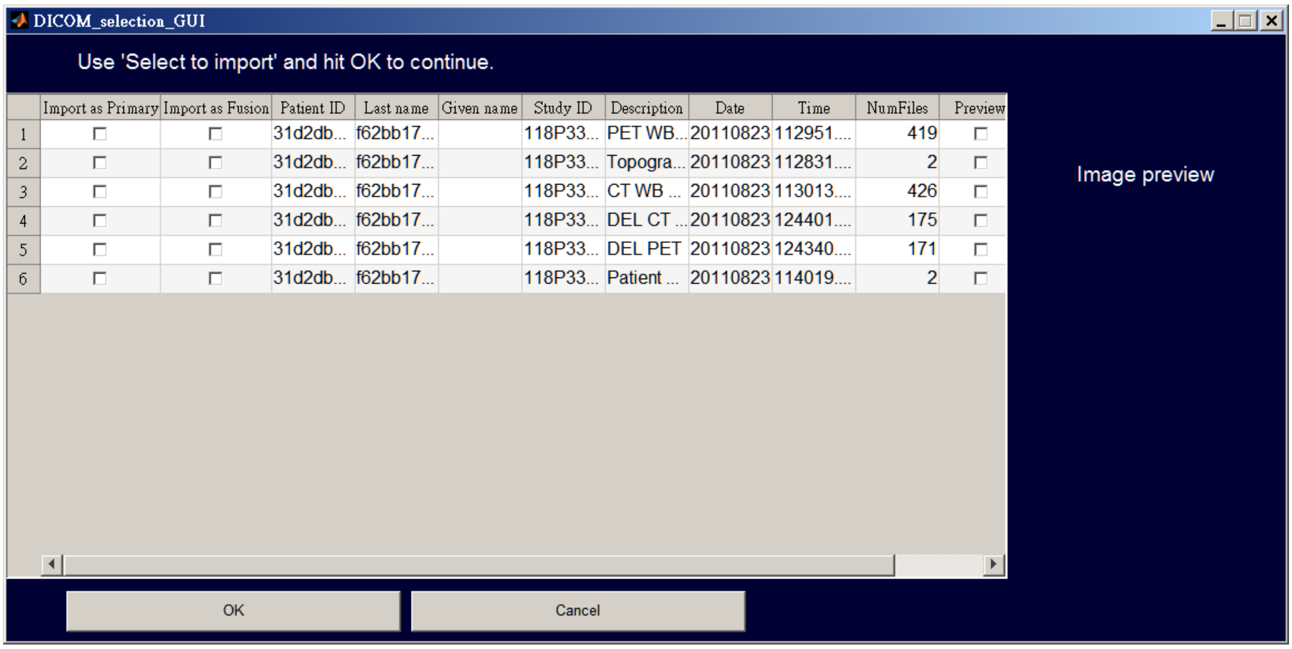
SUV: 2.5

Primary image min: 0 max: 5

Fusion image min: -1000 max: 1500

It will prompt you with a window to select folder. Select the folder that has the images.

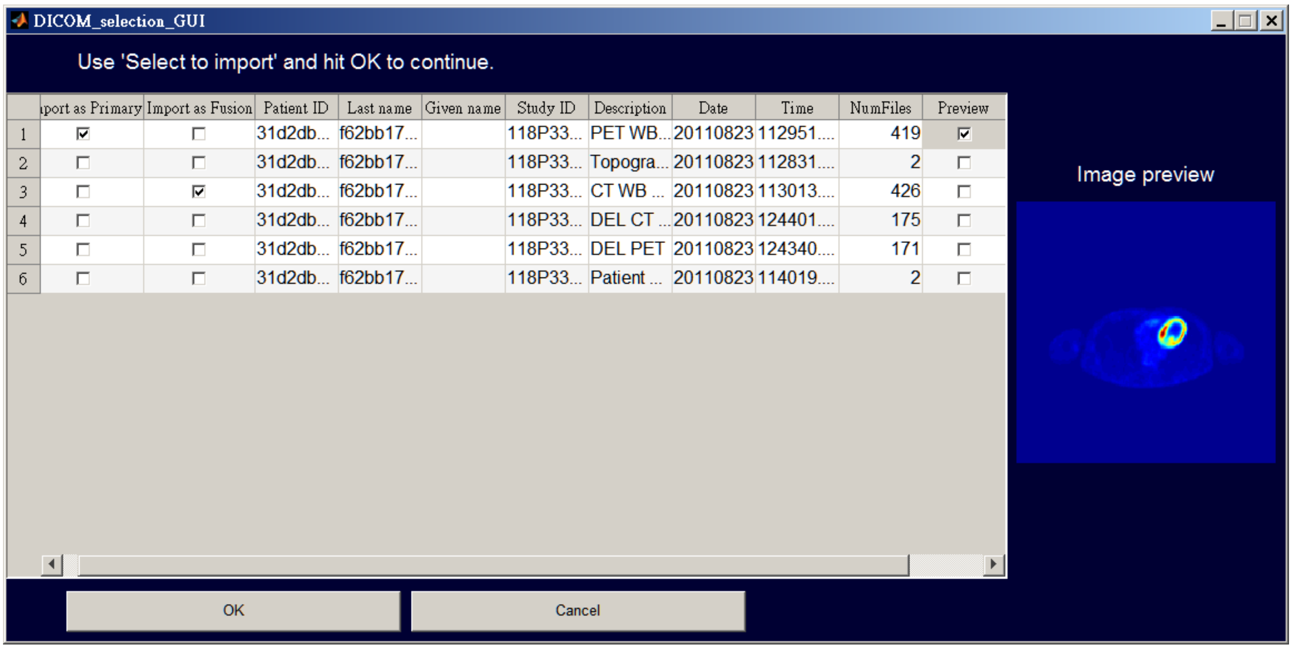
If there are more than one study contained in the selected folder or its subfolders, a window like this will pop up.



Use 'Select to import' and hit OK to continue.

	Import as Primary	Import as Fusion	Patient ID	Last name	Given name	Study ID	Description	Date	Time	NumFiles	Preview
1	□	□	31d2db...	f62bb17...		118P33...	PET WB...	20110823	112951....	419	□
2	□	□	31d2db...	f62bb17...		118P33...	Topogra...	20110823	112831....	2	□
3	□	□	31d2db...	f62bb17...		118P33...	CT WB ...	20110823	113013....	426	□
4	□	□	31d2db...	f62bb17...		118P33...	DEL CT ...	20110823	124401....	175	□
5	□	□	31d2db...	f62bb17...		118P33...	DEL PET	20110823	124340....	171	□
6	□	□	31d2db...	f62bb17...		118P33...	Patient ...	20110823	114019....	2	□

Click on the checkbox of 'Import as Primary' on the folder that contains the primary image volume and click 'OK'. If you wish to load the Fusion images as well, check the checkbox of 'Import as Fusion'.

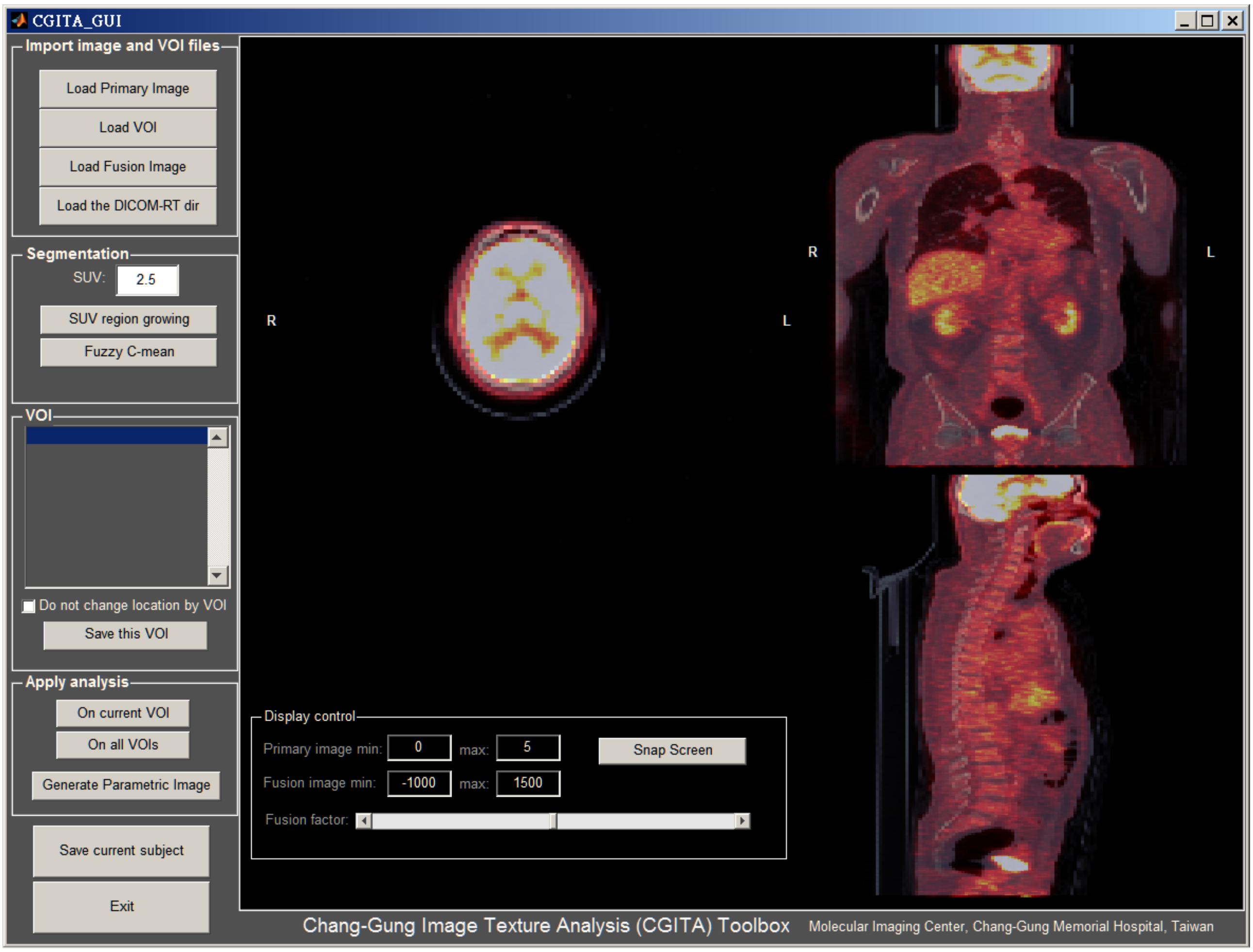


DICOM_selection_GUI

Use 'Select to import' and hit OK to continue.

	port as Primary	Import as Fusion	Patient ID	Last name	Given name	Study ID	Description	Date	Time	NumFiles	Preview
1	☑	□	31d2db...	f62bb17...		118P33...	PET WB...	20110823	112951....	419	☑
2	□	□	31d2db...	f62bb17...		118P33...	Topogra...	20110823	112831....	2	□
3	□	☑	31d2db...	f62bb17...		118P33...	CT WB ...	20110823	113013....	426	□
4	□	□	31d2db...	f62bb17...		118P33...	DEL CT ...	20110823	124401....	175	□
5	□	□	31d2db...	f62bb17...		118P33...	DEL PET	20110823	124340....	171	□
6	□	□	31d2db...	f62bb17...		118P33...	Patient ...	20110823	114019....	2	□

If the image reading is successful, you shall see the images displayed on the CGITA GUI:



SUV: 2.5

☐ Do not change location by VOI

Primary image min: 0 max: 5

Fusion image min: -1000 max: 1500

Chang-Gung Image Texture Analysis (CGITA) Toolbox Molecular Imaging Center, Chang-Gung Memorial Hospital, Taiwan

On the right panel, it shows the three views of the image. You can:

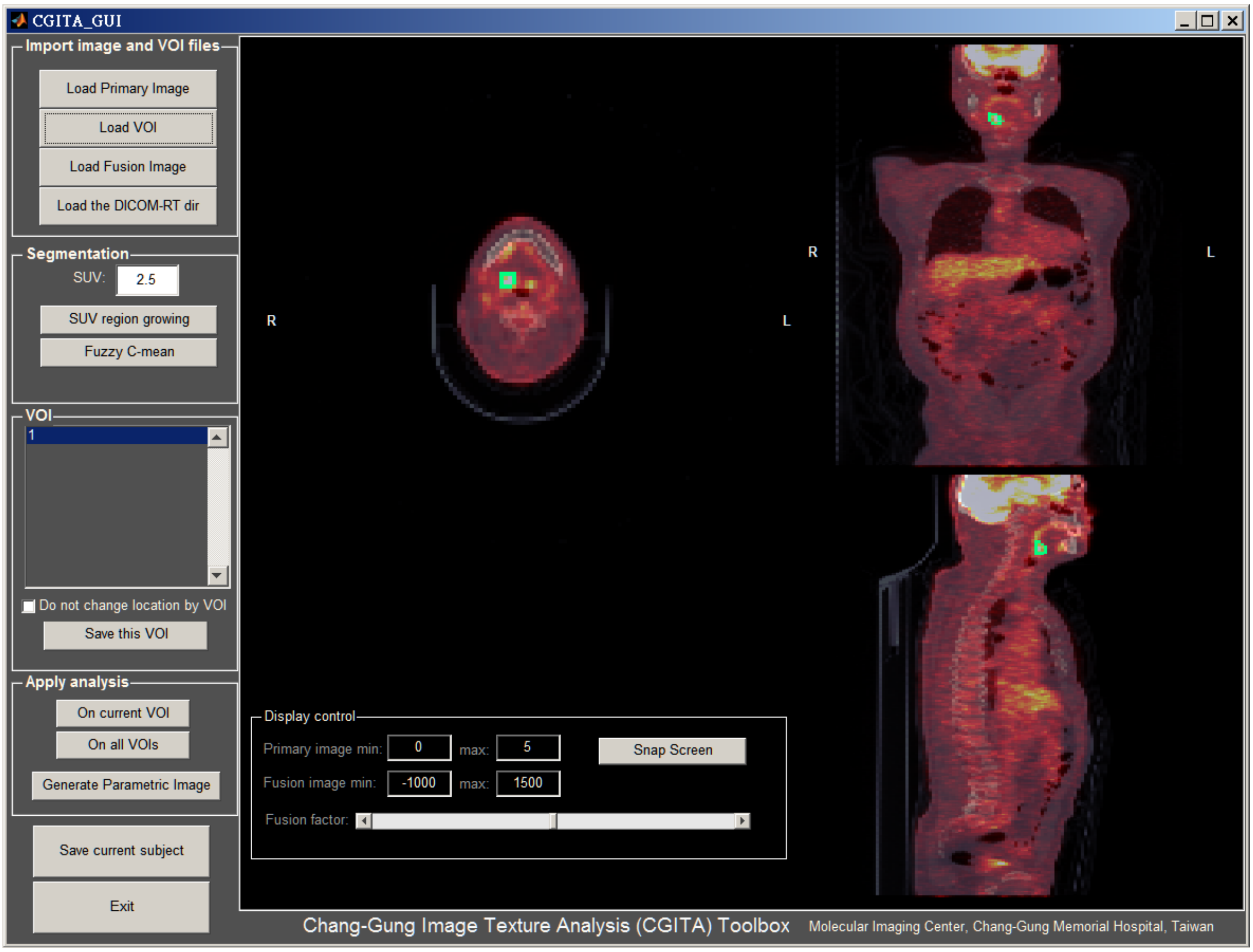
- Scroll on a view (middle button) = move to an adjacent slice
- Click on a view (left button) = re-slice over to the clicked position
- On the 'Display control' panel, you may adjust the contrast and fusion factor for the displayed image.

If you want to load Fusion images separately, click 'Load Fusion Image'.

(2) VOI reading

Currently, two formats are supported in CGITA. Click 'Load VOI' if your file is a PMOD .voi file. If your VOI file is saved as DICOM-RT, click 'Load the DICOM-RT dir'.

After the VOI is loaded, it will be shown on the display panel, overlapped with the images as shown in green.



SUV: 2.5

VOI: 1

□ Do not change location by VOI

Primary image min: 0 max: 5

Fusion image min: -1000 max: 1500

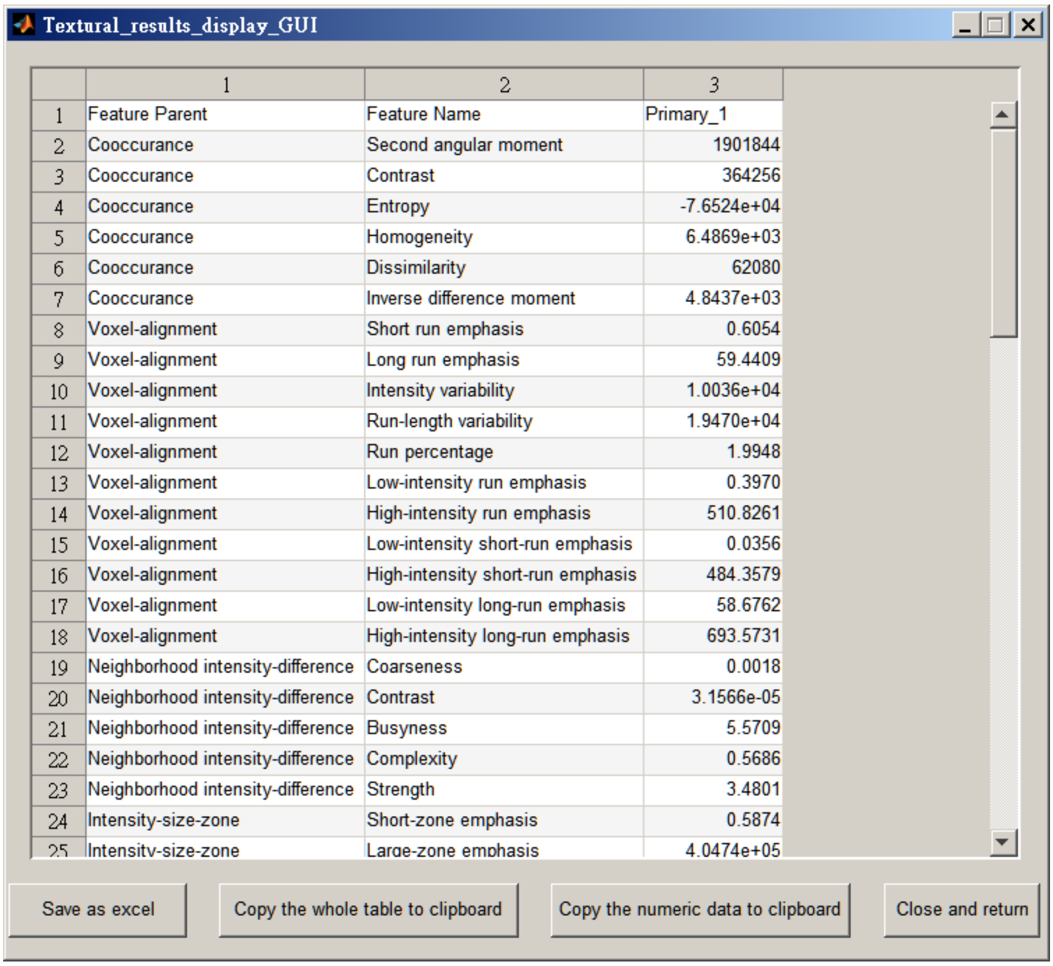
(3) Image segmentation

There are two ways to perform image segmentation in CGITA presently.

- Region growing: This approach uses a thresholding technique. The 'SUV' edit box above the 'SUV region growing' denotes the SUV that will be used for segmentation. First, move to a slice that shows the tumor. Enter the SUV value you would like to segment the tumor with. Click 'SUV region growing'. The cursor will become a crosshair. Then move the crosshair and click on the tumor. The tumor will be segmented.
- Fuzzy C-mean: This approach uses a Fuzzy C-mean technique. First, move to a slice that shows the tumor. Click 'SUV region growing'. The cursor will become a crosshair. Then move the crosshair and click on the tumor. The tumor will be segmented. Note: Such functionality requires 'Fuzzy Logic Toolbox'.

(4) Apply texture analysis

To apply texture analysis, click either 'On current VOI' or 'On all VOIs' in the 'Apply analysis' panel. The computation may take about twenty seconds to one minute. After completion, a window like the below will pop up:

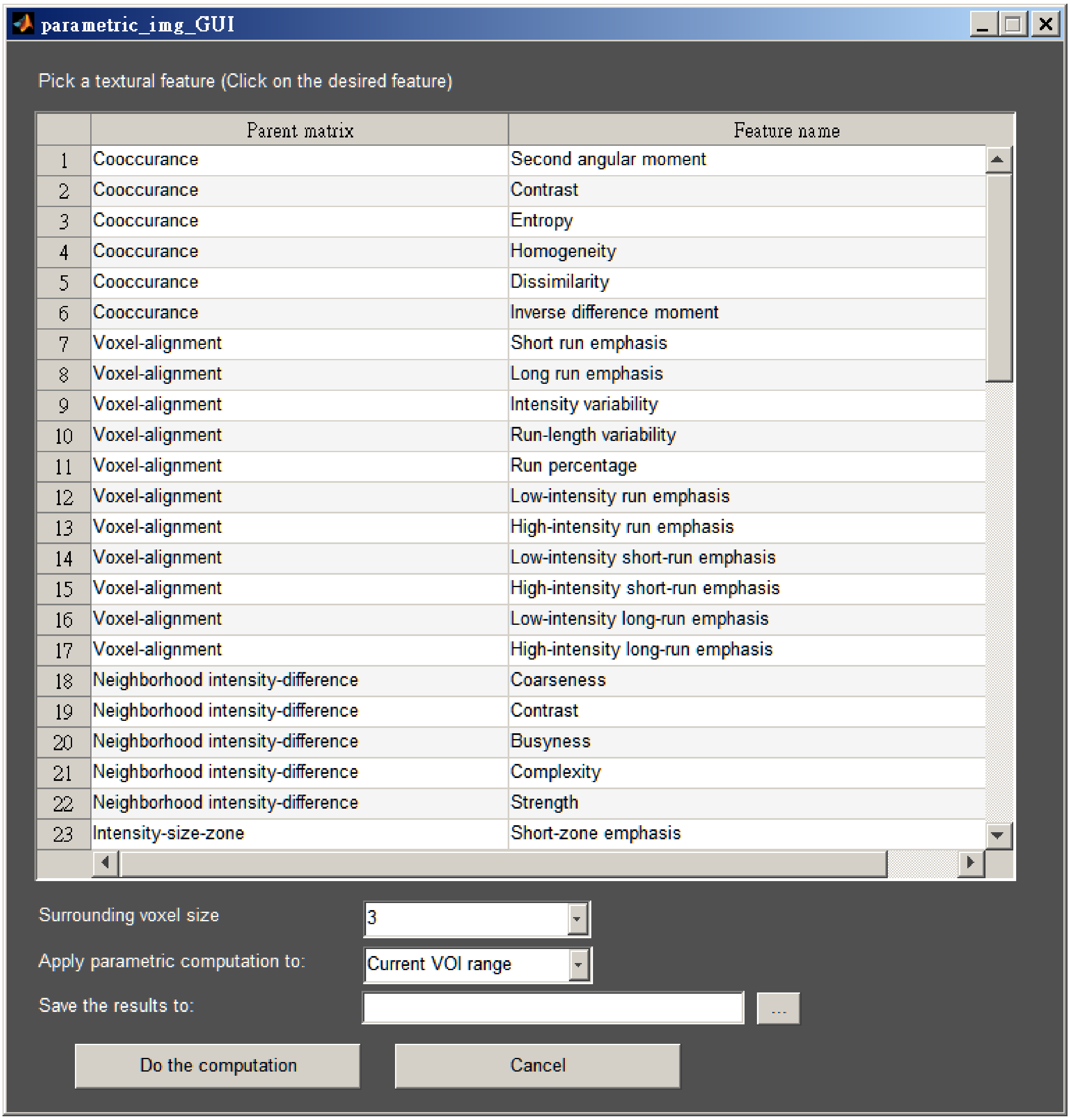

Textural_results_display_GUI
Save as excel
Copy the whole table to clipboard
Copy the numeric data to clipboard
Close and return

	1	2	3
1	Feature Parent	Feature Name	Primary_1
2	Cooccurance	Second angular moment	1901844
3	Cooccurance	Contrast	364256
4	Cooccurance	Entropy	-7.6524e+04
5	Cooccurance	Homogeneity	6.4869e+03
6	Cooccurance	Dissimilarity	62080
7	Cooccurance	Inverse difference moment	4.8437e+03
8	Voxel-alignment	Short run emphasis	0.6054
9	Voxel-alignment	Long run emphasis	59.4409
10	Voxel-alignment	Intensity variability	1.0036e+04
11	Voxel-alignment	Run-length variability	1.9470e+04
12	Voxel-alignment	Run percentage	1.9948
13	Voxel-alignment	Low-intensity run emphasis	0.3970
14	Voxel-alignment	High-intensity run emphasis	510.8261
15	Voxel-alignment	Low-intensity short-run emphasis	0.0356
16	Voxel-alignment	High-intensity short-run emphasis	484.3579
17	Voxel-alignment	Low-intensity long-run emphasis	58.6762
18	Voxel-alignment	High-intensity long-run emphasis	693.5731
19	Neighborhood intensity-difference	Coarseness	0.0018
20	Neighborhood intensity-difference	Contrast	3.1566e-05
21	Neighborhood intensity-difference	Busyness	5.5709
22	Neighborhood intensity-difference	Complexity	0.5686
23	Neighborhood intensity-difference	Strength	3.4801
24	Intensity-size-zone	Short-zone emphasis	0.5874
25	Intensity-size-zone	Large-zone emphasis	4.0474e+05

On this window, you may copy the data to the clipboard or just save it as an excel spreadsheet.

(5) Compute the textural parametric images

Click 'Generate Parametric Image' on the 'Apply analysis' panel. A window like this will pop up:



Pick a textural feature (Click on the desired feature)

	Parent matrix	Feature name
1	Cooccurance	Second angular moment
2	Cooccurance	Contrast
3	Cooccurance	Entropy
4	Cooccurance	Homogeneity
5	Cooccurance	Dissimilarity
6	Cooccurance	Inverse difference moment
7	Voxel-alignment	Short run emphasis
8	Voxel-alignment	Long run emphasis
9	Voxel-alignment	Intensity variability
10	Voxel-alignment	Run-length variability
11	Voxel-alignment	Run percentage
12	Voxel-alignment	Low-intensity run emphasis
13	Voxel-alignment	High-intensity run emphasis
14	Voxel-alignment	Low-intensity short-run emphasis
15	Voxel-alignment	High-intensity short-run emphasis
16	Voxel-alignment	Low-intensity long-run emphasis
17	Voxel-alignment	High-intensity long-run emphasis
18	Neighborhood intensity-difference	Coarseness
19	Neighborhood intensity-difference	Contrast
20	Neighborhood intensity-difference	Busyness
21	Neighborhood intensity-difference	Complexity
22	Neighborhood intensity-difference	Strength
23	Intensity-size-zone	Short-zone emphasis

Surrounding voxel size: 3

Apply parametric computation to: Current VOI range

Save the results to:

Click on the 'Feature name' column to select the textural feature you would like to generate the parametric image with. Modify the surrounding voxel size or range if desired. At the edit box of 'Save the results to', click on the button with '...' and select a file to write the computed results. (The computed parametric image will be written as a raw file of double precision values.) Click 'Do the computation'. Some textural features may take a while to compute.

iii. Using CGITA as a standalone executable

In general, using the CGITA under standalone executable is the same as under MATLAB.

iv. Using CGITA in the batch mode

(1) Prepare the data

The most important preparation is to save all the images (of the Primary study; Fusion study not supported yet) of the same patient under a single folder without subfolders. For each subject, get corresponding VOI files ready.

Also, you need to prepare for an excel that describes the patient code, image folder location and VOI folder location. An example excel file 'Run_batch_mode_input_excel_example.xls ' can be found under utils. There are three columns: Patient ID (or code), Primary image folder and VOI filename.

(2) Customizing the MATLAB script

Under 'utils' folder, make a copy of 'Run_batch_mode_example.m' and save it to another .m file. Then edit the first part of the script (Basic settings) according to your file location and preferred settings.

(3) Running the batched analysis

After you finish modifying the MATLAB script file, in the MATLAB editor go to 'Debug' -> 'Run XX.m'.

(4) Interpreting the results

The results of the texture analysis will be saved as your specified in the 'Basic settings' part.

4. Modifying CGITA

As CGITA is an open-source project, users may modify CGITA under the terms of use. We list a few potential functions that users may want to modify and describe briefly how to modify them.

i. Adding new textural features

(1) Data input and output

The main function for texture analysis is `Perform_TA_in_GUI`. In this function, it loops through the VOIs and then apply texture analysis on each of them sequentially. For each textural feature, a 'parent' function and the feature function are executed. For example, for the co-occurrence matrix-based features, the function ' compute_cooccurance_matrix.m' is the parent matrix function. After compute_cooccurance_matrix.m is executed, it creates a global variable ' glcm_global' for the feature functions to use. For example, ' cooccurrence_Correlation.m' computes the feature 'correlation' based on the glcm_global.

In Perform_TA_in_GUI, this part of the code is quite important and useful to understand the data input:

```
feval(parent_function_name, handles.image_vol_for_TA{idx_img_set}{idx_voi}, ...
                handles.resampled_image_vol_for_TA{idx_img_set}{idx_voi}, ...
                now_img_obj, ...
                handles, ...
                idx_img_set, ...
                idx_voi, ...
                handles.range{idx_img_set});
```

idx_img_set is the image index. 1 is for primary and 2 is for fusion.
idx_voi is the VOI index.
handles.image_vol_for_TA is the segmented image volume (in a cube).
handles.resampled_image_vol_for_TA is the segmented image volume that is resampled.

As different textural features may require different image data to compute, other variables are also passed. Users may refer to other parent functions for more information.

(2) Writing the parent matrix

Perhaps the best way to understand the parent matrix syntax is by referring to other built-in functions, such as ' compute_cooccurance_matrix.m', ' compute_NGLD_matrix.m' or ' compute_ISZ_matrix.m'. **The list of current textural features can be found under the CGITA**

directory *CGITA\docs\texture_summary.xlsx*

(3) Writing the computation of textural features

The textural features shall be computed simply from the global matrix/variable generated by the parent function. Again, the best way to implement new functions is to read the functions contained in the ' texture_fun' folder, such as 'cooccurrence_Homogeneity.m'.

(4) Modifying the texture configuration files

After you make your own textural functions, you must modify the configuration file so that CGITA GUI knows how to include your functions when computing the textural features. Go to *CGITA\settings* and find the 'create_default_feature_settings.m'. To add the new functions, follow the following steps:

Find, copy and paste the following part after it:

```
property_matrix = {'Second angular moment'      'cooccurrence_SAM';
                   'Contrast'                   'cooccurrence_Contrast_1';
                   'Entropy'                    'cooccurrence_Entropy';
                   'Homogeneity'                'cooccurrence_Homogeneity';
                   'Dissimilarity'              'cooccurrence_Dissimilarity';
                   'Inverse difference moment'  'cooccurrence_IDM'};

for idx = 1:size(property_matrix,1)
    feature_structure{1}{idx}.parent = 'Cooccurance';
    feature_structure{1}{idx}.name = property_matrix{idx, 1};
    feature_structure{1}{idx}.matlab_fun = property_matrix{idx, 2};
    feature_structure{1}{idx}.parentfcn = 'compute_cooccurance_matrix';
end
```

Then modify the pasted text with the following conditions:
The first column of the `property_matrix` defines the name of each textural feature.
The second column of the `property_matrix` defines the MATLAB function of each textural feature.
The line `feature_structure{1}{idx}.parent = 'Cooccurance';` in the for loop describes the name of the parent matrix. The line `feature_structure{1}{idx}.parentfcn = 'compute_cooccurance_matrix';` describes the MATLAB function for the texture parent matrix. Replace those variable names with the function you created.

After modifying the configuration script, run it to overwrite the existing configuration file (user_feature_settings.mat).

ii. Adding new segmentation methods

The best way to add new segmentation methods is to refer to the built in functions 'segment_suv_growing_Callback' and 'segment_fcm_Callback' within CGITA_GUI.m. We recommend that you use your own segmentation method to generate a masked matrix, containing 0s and 1s, with the same size as the primary image volume. Name this masked matrix 'segmented_mask'. Then, copy and paste the following code to convert the segmented mask into a VOI object.

```
voi_name = ['VOI_FCM'];
if length(handles.VOI_obj) < 1
   handles.VOI_obj = make_VOI_obj_from_mask(handles, segmented_mask, voi_name);
else
   handles.VOI_obj(end+1) = make_VOI_obj_from_mask(handles, segmented_mask, voi_name);
end
for idx = 1:length(handles.VOI_obj)
   a{idx} = handles.VOI_obj(idx).structureName;
end
set(handles.listbox1, 'String', a);
set(handles.listbox1, 'Value', length(handles.VOI_obj));

handles.VOI_loaded = 1;
handles = refresh_display(handles);

guidata(hObject, handles);
```

iii. Becoming a CGITA developer

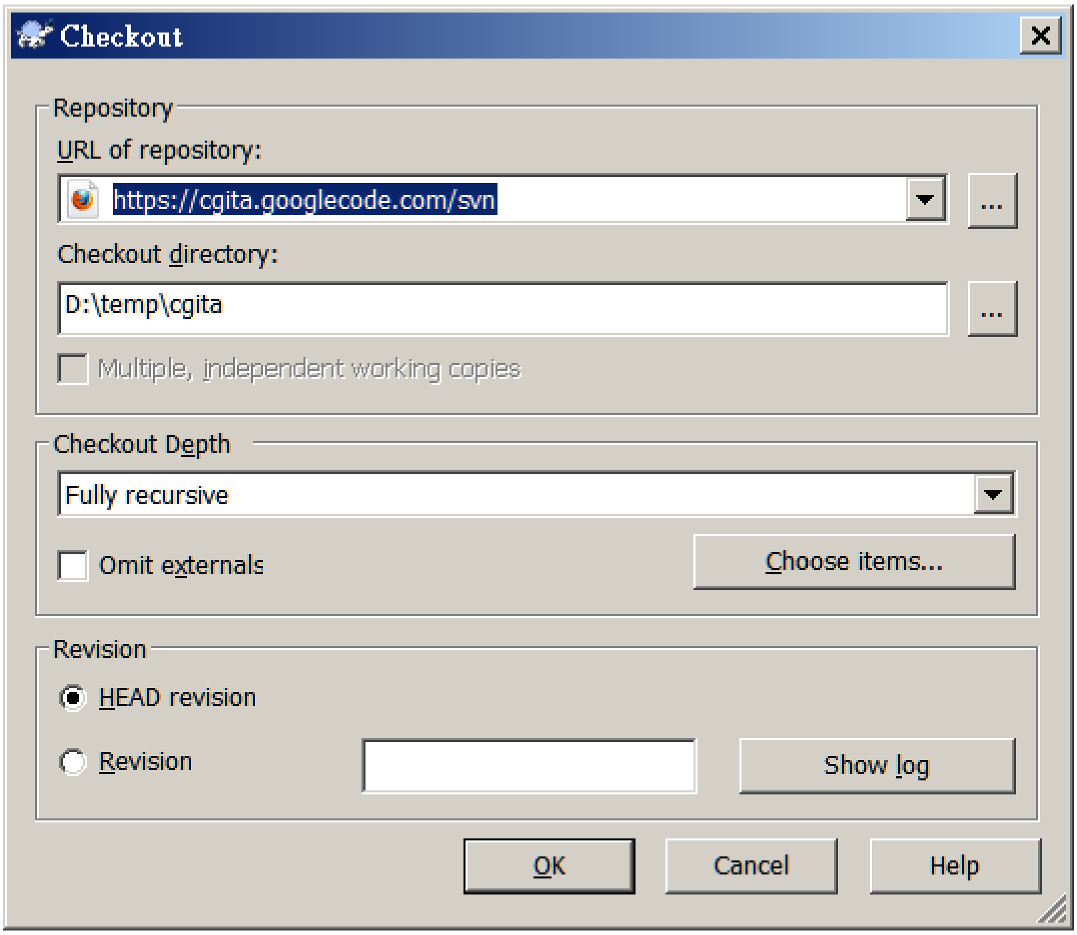
(1) How to become a CGITA developer

You may email the original author Yu-Hua Dean Fang to request developer access to the google project website. With such access, you can commit your changes to CGITA onto the SVN.

After you are added in the developer list of CGITA on the google project, you have to log in into google.com. Find 'Profile' -> 'Settings'. You should see a GoogleCode.com Password. That is the password you need for SVN.

(2) Using SVN for CGITA development

We recommend that you use TortoiseSVN (http://tortoisesvn.net/). Use 'SVN Checkout' and fill in the URL of repository as 'https://cgita.googlecode.com/svn'.



Checkout

Repository

URL of repository:

https://cgita.googlecode.com/svn

Checkout directory:

D:\temp\cgita

□ Multiple, independent working copies

Checkout Depth

Fully recursive

□ Omit externals

Revision

● HEAD revision

○ Revision