

**SOFTWEB - QUICK INSTALATION MANUAL**

**Web page parameter configuration**

|  |  |  |
| --- | --- | --- |
| Name | Description | Default value |
| cluster | True if we want to run our program on the cluster | false |
|  |  |  |
| computer | Computer name where we want to execute our program  [Not needed if cluster=true] | comic1.cs.ucl.ac.uk |
| ip\_computer | Computer IP or IP's where we execute our program  [Not needed if cluster=true] | null |
| user\_computer | User name on the computer where we will run the program  [Not needed if cluster=true] | tigweb |
| password\_computer | User password on the computer  [Not needed if cluster=true] | \*\*\*\*\*\* |
| path\_computer | Path where TIG Website has to copy the data  [Not needed if cluster=true] | /home/tigweb/data |
| allowed\_connections | Names or IP of the computer that can see and use our webpage. It cand be an array or a string | null |
| contact\_email | Your email address, it will appear just in case of big problems | f.carrasco@ucl.ac.uk |
|  |  |  |
| program | Program name for saving at the database. It has to be the same that the filename. It can not include spaces.  Ex: BSI, BRAIN-STEPS, GIF,... | PROGRAM NAME FOR DB |
| link\_short | This will appear at the menu bar  Ex: BSI, BSTEPS, GIF... | LINK NAME |
| link\_long | This will appear as title above the image  Ex: BOUNDARY SHIFT INTEGRAL, BRAIN STEPS,... | LINK TITLE NAME |
| image | Description image for the main page | image.jpg |
| title | Main heading title | MAIN HEADING TITLE |
| initial\_text | Initial introductory text, it could include HTML tags | Initial introductory text |
| references | Text with the references to your work, it could include HTML tags. It will be send inside the email with the results, and it will appear in the webpage | We would appreciate it if the following references would be included in any published work that uses this resource: |
|  |  |  |
| num\_uploaders | Number of different files that need the program. Ex: if you need baseline and followup and a mask for each one, you should use 4 | 1 |
| queue\_file\_distribution | How the files of each uploader need to be sorted. If we select row, we will take each uploader row as one execution. Possible values are: all, row, column | row |
| all\_queues\_same\_number\_files | Check that all uploaders has the same number of files | true |
| text\_button\_files | String with the text for the button that starts the uploading process | Upload button text |
| max\_number\_files | Max number of files per uploader, if we define it in array, we can define different max numbers for each uploader. Ex: num\_uploaders=3 then if we desire: array(10,5,2) | 10 |
| file\_max\_size | Max file size in Mb per file, if we define it in array, we can define different max file size for each uploader. Ex: num\_uploaders=3 then if we desire: array(100,5,10) | 100 |
| extensions\_allowed\_text | It is the text that will see the users in each uploader advising which are the extensions allowed. If we want, we can define different text for each uploader. Ex: num\_uploaders=3 then: array('.nii or .nii.gz or .hdr,'.doc and .xls','.txt') | .nii or .nii.gz |
| files\_description | A different description for each uploader can be defined. Ex: num\_uploaders=2 then: array(' Baseline images','Follow-up images ') | Select T1 files |
| file\_type | A different file type for each uploader can be defined. Ex: num\_uploaders=2 then:  $file\_types\_1=array(  array(  'description' => 'NIFTI GZ files',  'extension' => 'nii.gz'  ));  $file\_types\_1=array(  array(  'description' => 'Spreadsheet',  'extension' => 'xls'  ));  $file\_type=array($file\_types\_1,$file\_types\_2); | array(  array(  'description' => 'NIFTI GZ files',  'extension' => 'nii.gz'  ),  array(  'description' => 'NIFTI files',  'extension' => 'nii'  )  ) |
|  |  |  |
| parameter\_list | It is a set of user configurable parameters. Each one of them could be a list or a text. *Name* is the name of the parameter, and it will be used as a flag name. *Description* is the text that the user will see on the screen. *Value* is the possible values/values (string/array). *Validation* is a regular expression for validating the value (useful for texts). *error\_ms*g is the message that will see the user in case that the validation fails. Ex:  $text\_example=array(  'name' => 'matching',  'description' => 'Number of brains to fuse [0-99]',  'value' => array(30),  'validation' => '/\d?\d/',  'error\_msg' => 'Something is wrong'  );  $list=array(  'name' => 'area',  'description' => 'Area to segment',  'value' => array(  'BRAIN' => 'Full Brain',  'HIPPO' => 'Both Hippocampus',  'WHAT\_WE\_RECEIVES'=>'What the user sees'  );  $parameter\_list=array($text\_example,$list); | empty |
|  |  |  |
| results\_extension | Optional variable to provide the extension of the final result file. The preferable extensions are .zip and .nii.gz | .nii.gz |

**Desktop computer configuration**

1. Make a user with the login and password provided on ***user\_computer*** and ***password\_computer*** variables on the PHP configuration file.

adduser tigweb tigweb

1. Make the directory specified at ***path\_directory*** variable.

mkdir /home/tigweb/data

1. Copy the *daemon\_SOFTWEB.py* in the new user home directory, and test if it works running it without parameters

cd /home/tigweb

wget http://cmictig.cs.ucl.ac.uk/softweb/daemon\_SOFTWEB.py

daemon\_SOFTWEB.py

You have to see just this message: Error, you should define a program to be executed. It means that the daemon and Python are properly installed. Otherwise you have to include the needed packages using:

easy\_install package\_name

1. We have to do the script that will be executed by the *daemon\_SOFTWEB.py*, for example for filling lesions program the script is named FILLING as program variable and the content is:

#!/bin/bash

program=`basename ${0}`

dir\_data=`dirname ${2}`

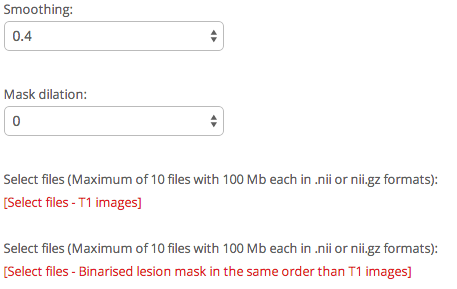
output=${dir\_data}/${1}\_${program}.nii.gz

smo=${5}

dil=${7}

seg\_FillLesions ${2} ${3} ${output} -smo ${smo} -dil ${dil}

The TIG-website for filling lesions program looks like this:



And the parameter list is:

$parameter\_list=array(

array(

'name' => 'smo',

'description' => 'Smoothing',

'value' => array(

'0.4' => '0.4',

'0.0' => '0.0',

'0.1' => '0.1',

[...]

'0.6' => '0.6'

),

'validation' => '',

'error\_msg' => '',

),

array(

'name' => 'dil',

'description' => 'Mask dilation',

'value' => array(

'0' => '0',

[...]

'4' => '4'

),

'validation' => '',

'error\_msg' => '',

)

);

The *daemon\_SOFTWEB.py* will call our program as:

FILLING 15 absolute\_path\_to\_t1\_image.nii.gz absolute\_path\_to\_mask\_image.nii.gz -smo 0.4 -dil 1

Then, the general calling structure is:

PROGRAM ID FILE\_1 FILE\_2 ... FILE\_N -NAME\_PARAM\_LIST\_1 VALUE\_PARAM\_LIST\_1 -NAME\_PARAM\_LIST\_2 VALUE\_PARAM\_LIST\_2 ... -NAME\_PARAM\_LIST\_N VALUE\_PARAM\_LIST\_N

As final result our program has to generate a file on the directory defined in the ***path\_to\_computer*** variable with this name:

ID\_PROGRAM.[nii.gz/zip]

For example for filling lesions program, if ID number is 15 we will generate this output file:

15\_FILLING.nii.gz

1. Finally, we have to set up the crontab for running the daemon each N minutes, we need to do it from the command line:

crontab -e

It will open an editor (usually vi). We have to add the follow line, where PROGRAM need to have the same value than program variable on PHP configuration file. It will execute the daemon each minute looking for new data to be processed:

0-59 \* \* \* \* /home/tigweb/daemon\_SOFTWEB.py PROGRAM

For example, if we want to run the boundary shift integral, and we have to set up the ***program*** variable as program='BSI'; then we should do:

0-59 \* \* \* \* /home/tigweb/daemon\_SOFTWEB.py BSI

**Cluster configuration**

If we setup the ***cluster*** variable on the PHP configuration file to *true*, we will be able to run our programs in the CS cluster. Moreover, we will have to write two scripts more. We need to write a script that does the submission to the cluster queue. This script has to be named as the ***program*** variable value on the PHP configuration file. The init\_SOFTWEB.py will call our script as:

PROGRAM ID FILE\_1 FILE\_2 ... FILE\_N -NAME\_PARAM\_LIST\_1 VALUE\_PARAM\_LIST\_1 -NAME\_PARAM\_LIST\_2 VALUE\_PARAM\_LIST\_2 ... -NAME\_PARAM\_LIST\_N VALUE\_PARAM\_LIST\_N

At the end of the execution we have to generate and *ID\_PROGRAM.nii.gz* or *ID\_PROGRAM.zip* file. Generic example:

**Script name: PROGRAM**

**Absolute Path: /home/tigweb\_user/tigweb/PROGRAM**

#!/bin/bash

export SGE\_ROOT=/opt/gridengine

export SGE\_QMASTER\_PORT=536

export SGE\_EXECD\_PORT=537

MEMORY=8192

EXECUTION\_TIME=48

absolute\_path\_to\_your\_script=/home/your\_user/.../example-script-QSUB.sh

program=`basename ${0}`

dir\_data=`dirname ${2}`

/opt/gridengine/bin/lx26-amd64/qsub \

-l vf=${MEMORY}M \

-l tmem=${MEMORY}M,h\_vmem=${MEMORY}M \

-l h\_rt=${EXECUTION\_TIME}:0:0 \

-l s\_stack=10240 \

-j y -S /bin/bash \

-b n -cwd -V \

${absolute\_path\_to\_your\_script} \

${program} ${1} ${2} ${dir\_data}/${1}\_${program}.nii.gz

Our PROGRAM script will use example-script-QSUB.sh, that it can have the structure that we decide, however it has to include 2 lines for getting the starting time, and a final line to notify that we have finished. See example below:

**Script name: example-script-QSUB.sh**

**Absolute Path: ${absolute\_path\_to\_your\_final\_script}/example-script-QSUB.sh**

#!/bin/bash

DIR=/home/tigweb\_user/tigweb

init=`date +"%H:%M:%S %d-%m-%Y"`

initsql=`date +"%Y-%m-%d %H:%M:%S"`

absolute\_path\_to\_your\_final\_script=/home/your\_user/.../

${absolute\_path\_to\_your\_final\_script}/our\_final\_program.sh -input ${3} -output ${4}

${DIR}/python/bin/python \

${DIR}/end\_SOFTWEB.py ${1} ${2} "${init}" "${initsql}"

**PHP configuration file - Desktop computer full example**

<?php

$allowed\_connections=null;

$computer='blakey.cs.ucl.ac.uk';

$ip\_computer='128.16.9.118';

$user\_computer='tigweb';

$password\_computer='ZmVycmFucGM=';

$path\_computer='/home/tigweb/tmp/';

$contact\_email='f.carrasco@ucl.ac.uk';

$program='BSI';

$link\_short='BSI';

$link\_long='BOUNDARY SHIFT INTEGRAL';

$image='bsi.png';

$title='Online atrophy calculation tool based on BSI (FIFO Queue)';

$initial\_text='The "Online atrophy tool" is a fully automated software-as-service tool, which uses the boundary shift integral algorithm to compute the atrophy between two timepoints longitudinal studies. ';

$initial\_text.='BSI is a robust method for measuring brain atrophy rates from longitudinal magnetic resonane imaging scans using linear registration to align the baseline and repeat images and then track the shift of the brain boundary location.';

$references='<b>Publications on GBSI</b><br><br>';

$references.='We would appreciate it if the following references would be included in any published work that uses this resource:</div><br>';

$references.='<ul>';

$references.='<li><b>http://cmictig.cs.ucl.ac.uk/softweb/</b></li>';

$references.='<li>Ferran Prados, M. Jorge Cardoso, Kelvin K. Leung, David M. Cash, Marc Modat, Nick C. Fox, Claudia A.M. Wheeler-Kingshott, Sebastien Ourselin for the Alzheimer\'s Disease Neuroimaging Initiative. ';

$references.='<i><a href="http://..." target="gbsi">Measuring brain atrophy with a generalized formulation of the boundary shift integral</a></i>. Neurobiology of aging. (2014).</li>';

$references.='<li>"The GPU card used for this research was donated by the NVIDIA Corporation." or "We gratefully acknowledge the support of NVIDIA Corporation with the donation of the GPU card used for this research".</li>';

$references.='</ul>';

$text\_button\_files='Upload files and run the atrophy calculation';

$num\_uploaders=2;

$files\_description=array(

"Baseline T1 images",

"Follow-up T1 images"

);

$file\_types\_1=array(

array(

'description' => 'NIFTI GZ files',

'extension' => 'nii.gz'

),

array(

'description' => 'NIFTI files',

'extension' => 'nii'

)

);

$file\_type=array($file\_types\_1,$file\_types\_1);

$parameter\_list=array(

array(

'name' => 'method',

'description' => 'Boundary Shift Integral method',

'value' => array(

'GBSI' => 'GBSI',

'KNBSI' => 'KN-BSI'

)

),

array(

'name' => 'area',

'description' => 'Select the region of interest',

'value' => array(

'BRAIN' => 'Full brain',

'HIPPO' => 'Hippocampus'

)

)

);

$results\_extension=".zip";

**PHP configuration file - Cluster full example**

$contact\_email='f.carrasco@ucl.ac.uk';

$cluster=true;

$program='BRAIN-STEPS';

$link\_short='BSTEPS';

$link\_long='BRAIN STEPS';

$image='steps.png';

$title='Online ROI extraction tool based on STEPS (FIFO Queue)';

$initial\_text='The "Online ROI extraction tool" is a fully automated software-as-service tool, which uses the STEPS algorithm to segment the brain or the hippocampi in T1-weighted images. ';

$initial\_text.='STEPS is a multi-atlas label propagation and fusion strategy implemented as part of <a href="http://sourceforge.net/projects/niftyseg/">NiftySeg</a>.';

$references='<b>Publications on STEPS</b><br><br>';

$references.='We would appreciate it if the following references would be included in any published work that uses this resource:</div><br>';

$references.='<ul>';

$references.='<li><b>http://cmictig.cs.ucl.ac.uk/softweb/</b></li>';

$references.='<li>M. Jorge Cardoso, Kelvin Leung, Marc Modat, Shiva Keihaninejad, David Cash, Josephine Barnes, Nick C. Fox, Sebastien Ourselin, for the Alzheimer\'s Disease Neuroimaging Initiative. ';

$references.='<i><a href="http://www.sciencedirect.com/science/article/pii/S1361841513000200" target="steps">STEPS: Similarity and Truth Estimation for Propagated Segmentations and its application to hippocampal segmentation and brain parcelation</a></i>. Medical Image Analysis 17(6), 671-684 (2013).</li>';

$references.='<li>"The GPU card used for this research was donated by the NVIDIA Corporation." or "We gratefully acknowledge the support of NVIDIA Corporation with the donation of the GPU card used for this research".</li>';

$references.='</ul>';

$text\_button\_files='Upload files and run the ROI extraction';

$files\_description=array(

'Select T1 files'

);

$parameter\_list=array(

array(

'name' => 'area',

'description' => 'Area to segment',

'value' => array(

'BRAIN' => 'Full brain',

'HIPPO' => 'Hippocampus'

)

)

);

$results\_extension=".nii.gz";

**Scripts on the Cluster - Full Example**

**Script name: BRAIN-STEPS**

**Absolute Path: /home/fprados/tigweb**

#!/bin/bash

export SGE\_ROOT=/opt/gridengine

export SGE\_QMASTER\_PORT=536

export SGE\_EXECD\_PORT=537

MEMORY=8192

EXECUTION\_TIME=48

program=`basename ${0}`

dir\_data=`dirname ${2}`

if [ "${4}" == "BRAIN" ] ; then

area='brain'

else

area='hippo'

fi

/opt/gridengine/bin/lx26-amd64/qsub \

-l vf=${MEMORY}M \

-l tmem=${MEMORY}M,h\_vmem=${MEMORY}M \

-l h\_rt=${EXECUTION\_TIME}:0:0 \

-l s\_stack=10240 \

-j y -S /bin/bash \

-b n -cwd -V \

/home/fprados/tigweb/brain\_steps-QSUB.sh \

${program} ${1} ${area} ${2} ${dir\_data}/${1}\_${program}.nii.gz

**Script name: brain-steps-QSUB.sh**

**Absolute Path: /home/fprados/tigweb**

#!/bin/bash

DIR=/home/fprados/tigweb

init=`date +"%H:%M:%S %d-%m-%Y"`

initsql=`date +"%Y-%m-%d %H:%M:%S"`

/home/fprados/work/brain\_steps/${3}\_steps\_comicTIGWEB.sh -input ${4} -output ${5} -n3

${DIR}/python/bin/python \

${DIR}/end\_SOFTWEB.py ${1} ${2} "${init}" "${initsql}"