Biomembrane Acceptance Test Document

Nick Klose, Richard Leung, Cameron Mann, Glen Nelson, Omar Qadri, James Wang

Introduction

This document describes the Acceptance tests run on the Biomembrane application ("ICS") to verify functionality. There are two interfaces that were tested – the Local GUI, and the Web GUI. Each interface has the same functionality, and minimal changes should be necessary to use the same tests on both. This document is meant for both upper management (in this case Eleni) and the team clients (Nils Petersen, etc.). It ensures the final product meets the requirements of the client.

Installation

Once the web service is deployed, an installer can be downloaded from there. Until then, the installer is available on my Dropbox. The link will be sent to relevant clients. The source code is available on github, and a zipped form of the source code will be supplied to the clients.

To install the Local GUI from an installer (Windows):

1. Download the installer
2. Run the installer, follow instructions
3. If the Microsoft C++ 2008 redistributable setup pops up (only if you have not installed it) then install it as well.
4. ImageMagick setup will always pop up. Install it if it is not installed on your computer.
5. Run ICS from the Start menu entry.

To install the Local GUI from source:

1. Checkout the source code from github.
2. Install prerequisite packages:
   a. On Windows:
      i. Install python 2.7.3 http://www.python.org/getit/releases/2.7.3/
      ii. Install scipy, numpy, pythonmagick, matplotlib, pip, and pil (use pillow) for 32-bit. Use premade packages available at: http://www.lfd.uci.edu/~gohlke/pythonlibs/
      iii. Install py2exe from http://sourceforge.net/projects/py2exe/files/py2exe/0.6.9/
      iv. Install pyqt from http://www.riverbankcomputing.com/software/pyqt/download
      v. Follow http://docs.wand-py.org/en/0.2-maintenance/guide/install.html#install-imagemagick-on-windows

    vi. open a command prompt, navigate to pip (C:\Python27\Scripts) and call `pip.exe install Wand`

    vii. Install mingw (32-bit). See http://www.mingw.org/wiki/Getting_Started.

    viii. Install Visual Studio C++ Redistributable version here: http://www.microsoft.com/en-us/download/details.aspx?id=29 (This is also in the repository vendor/vc-redist.exe)

    ix. Open vendor/windows-matplotlib.zip in the repository. Extract these files over matplotlib on windows (extract to C:\Python27\Lib\site-packages\)

    x. Launch a mingw 32-bit shell. Navigate to the repository `cd /c/Users/<username here>/Documents/Biomembrane`

    xi. Type `make` then enter. This should create a “bin” directory that contains ICS.exe that can be run to start the program.

    xii. If the icon does not work, pywin32 might need to be installed from http://sourceforge.net/projects/pywin32/files/pywin32/Build%20218/

    xiii. To create a setup file, install Inno Setup from http://www.jrsoftware.org/isinfo.php. Then open installer-windows.iss and compile the installer. The final installer will be in the directory “Output”.

b. On Ubuntu:

    i. `sudo apt-get install python-pythonmagick imagemagick pip python-qt4 gcc cx-freeze git`

    ii. `sudo pip install numpy`

    iii. `sudo pip install scipy`

    iv. `sudo pip install matplotlib`

    v. `sudo pip install pillow` or `pip install --no-index -f http://dist.plone.org/thirdparty/ -U PIL`

    vi. `sudo pip install Wand`

    vii. Clone the repository with git.

    viii. Navigate to the repository

    ix. Call “`make`” to create the backend library and an executable file.

    x. Call `python local_start.py` or `bin/ICS`.

c. On OSX:

    i. Install Xcode

    ii. Install the command line tools from the Downloads tab in Xcode preferences

    iii. Install Homebrew

    iv. brew install fftw pip boost gfortran pkg-config pyqt

        1. For PyQt pay attention to the caveat about setting PYTHONPATH

    v. sudo pip install numpy scipy wand pil matplotlib

    vi. Install ImageMagick from source using: `./configure --with-magick-plus-plus`

    vii. Install python from source using:

```
./configure CPPFLAGS=-
I/System/Library/Frameworks/Python.framework/Versions/2.7/include/pytho
n2.7 LDFLAGS="-ldl -framework CoreFoundation -lpython2.7"
```

To install the Web GUI:

We have not tried to install the Web GUI on a windows machine. As such, some of the following steps will need to be adjusted.

1. Complete installation of the local GUI from source
2. `sudo apt-get install python-django python-sqlite rabbitmq-server`
3. `Sudo pip install django-celery django-registration`
4. Configure RabbitMQ:
   ```
   sudo rabbitmqctl add_user biomembrane
   sudo rabbitmqctl add_vhost biomembrane
   sudo rabbitmqctl set_permissions -p biomembrane biomembrane ".*" ".*"
   ".*"
   ```
5. Create the database: `python manage.py syncdb`
6. Run the development server:
   ```
   python manage.py celery worker --loglevel=info &
   python manage.py runserver
   ```
7. Apache can be configured using a wsgi script for production environments. See https://docs.djangoproject.com/en/dev/topics/install/ for more information.
8. Configure Mail Client Settings in settings.py:
   ```
   ACCOUNT_ACTIVATION_DAYS= <days>
    EMAIL_HOST = <host address>
    EMAIL_PORT = <portnumber>
    EMAIL_HOST_USER = <host>
    EMAIL_HOST_PASSWORD = <password>
    EMAIL_USE_TLS = <True,False>
    DEFAULT_FROM_EMAIL = <email address>
   ```
9. Start a SMTP server:
   ```
   python -m smtpd -n -c DebuggingServer <server address>:<port>
   ```

System Tests

Following each acceptance test is a four item result table. This table checks if the test was successful when running from Linux source, Linux executable, Windows Executable, and web interface.

We assume, for the web interface that during the “syncdb” call a superuser was created: “admin” with password “abc”.

Image Inputs

Requirement #: 1

Description: The program will handle either single or separate RGB images in a variety of lossless formats.

Acceptance Tests:

Acceptance Test #1: Load Single Bitmap

Test Level: Major

Pre-conditions: Installation complete

Procedure:

1. Open the Local GUI
2. Switch to the Single RGB Image tab and Select the single RGB image `accTests/inputs/RGBtemp/rgb_001.bmp`
3. Switch to All tab and set range = 20, g0 = 1, w = 10, and ginf = 0.
4. Press Start.
5. On the triple output tab, set sample resolution to 32x32 and press continue.
6. Set parameters to range = 15, g0 = 1, w = 10, and ginf = 0 and press continue.
7. Save output to a new folder. ("Save all Results")
8. Compare output to the output in `accTests/outputs/RGBtemp/*.txt.`

Expected Behavior: Output should match the sample outputs within error of 0.05. There is a "compare_folders.py" script to do this automatically (`python compare_folders.py accTests/outputs/RGBtemp <place where output was saved>`).

Minor Deviations: In the web page you will have to navigate to the upload webpage after logging in. You will have to extract the results from a zip on the web interface. compare_folders does not work with the web format by default. This can be fixed by using `web2local_file_names.py <where zip extracted>`

Linux Source	Linux Executable	Windows Executable	Web Gui
Pass	Pass	Pass	Pass

Acceptance Test #2: Load three channel separated Bitmaps

Test Level: Major

Pre-conditions: Installation complete

Procedure:

1. Open the Local GUI
2. Select the three single channel images `accTests/inputs/RGBtemp/r_001.bmp, accTests/inputs/RGBtemp/g_001.bmp, accTests/inputs/RGBtemp/b_001.bmp`
3. Switch to all tab and set range = 20, g0 = 1, w = 10, and ginf = 0.
4. Press Start.
5. On the triple output tab, set sample resolution to 32x32 and press continue.
6. Set parameters to range = 15, g0 = 1, w = 10, and ginf = 0 and press continue.

7. Save output to a new folder. ("Save all Results")
8. Compare output to the output in `accTests/outputs/RGBtemp/*.txt.`

Expected Behavior: Output should match the sample outputs within error of 0.05. There is a "compare_folders.py" script to do this automatically (`python compare_folders.py accTests/outputs/RGBtemp <place where output was saved>`).

Minor Deviations: In the web page you will have to navigate to the upload webpage after logging in. You will have to extract the results from a zip on the web interface. compare_folders does not work with the web format by default. This can be fixed by using `web2local_file_names.py <where zip extracted>`

Linux Source	Linux Executable	Windows Executable	Web Gui
Pass	Pass	Pass	Pass

Acceptance Test #3: Load Single Assorted Files
Test Level: Major
Pre-conditions: Installation complete
Procedure:

1. Open the Local GUI
2. Select the single RGB image `accTests/inputs/RGBtemp/rgb_001.png`
3. Run all correlations using range = 20, g0 = 1, w = 10, and ginf = 0.
4. Select `accTests/inputs/RGBtemp/rgb_001.tif`, and repeat the correlations.
5. Select `accTests/inputs/RGBtemp_gif/rgb_001.gif` and repeat the correlations. Data should be slightly different.
6. Select `accTests/inputs/RGBtemp/rgb_001.jpeg`.

Expected Behavior: Correlations should be able to run with each of png, tif, and gif. The jpeg file should fail to load.

Minor Deviations: In the web page you will have to navigate to the upload webpage after logging in.

Linux Source	Linux Executable	Windows Executable	Web Gui
Pass	Pass	Pass	Pass

Acceptance Test #4: Load three channel separated assorted files
Test Level: Major
Pre-conditions: Installation complete
Procedure:

1. Open the Local GUI

2. Select the three single channel images `accTests/inputs/RGBtemp/r_001.png, accTests/inputs/RGBtemp/g_001.png, accTests/inputs/RGBtemp/b_001.png`
3. Run all correlations using range = 20, g0 = 1, w = 10, and ginf = 0.
4. Repeat correlations with `accTests/inputs/RGBtemp/r_001.tif, accTests/inputs/RGBtemp/g_001.tif, accTests/inputs/RGBtemp/b_001.tif`
5. Repeat correlations with `accTests/inputs/RGBtemp_gif/r_001.gif, accTests/inputs/RGBtemp_gif/g_001.gif, accTests/inputs/RGBtemp_gif/b_001.gif`
6. Attempt to load `accTests/inputs/RGBtemp/r_001.jpeg, accTests/inputs/RGBtemp/g_001.jpeg, accTests/inputs/RGBtemp/b_001.jpeg.`

Expected Behavior: Correlations should be able to run with each of png, tif, and gif. The jpeg file should fail to load.

Minor Deviations: In the web page you will have to navigate to the upload webpage after logging in.

Linux Source	Linux Executable	Windows Executable	Web Gui
Pass	Pass	Pass	Pass

Image Processing

Requirement #: 2

Description: The program will process the images using known algorithms.

Acceptance Tests:

The results of Acceptance Tests 1 to 4 also tests this requirement.

Batch Processing

Requirement #: 3

Description: The program will accept multiple images/sets of images as input.

Acceptance Tests:

Acceptance Test #5: bad_data batch calculations

Test Level: Major

Pre-conditions: Installation complete

Procedure:

1. Open the Local GUI
2. Click on “Switch to Batch Mode”
3. Click “Load Images Folder” and select `accTests/inputs/badData/`.

4. In Auto/cross correlation parameters, set Range=20, g0 = 1, w = 10, and ginf = 0.
5. In Triple parameters set Sample Resolution to 64x64, Range=15, g0=1, w=10, ginf=0.
6. Press Start
7. All 40 images should be looped through, and the final output should appear in `accTests/inputs/badData_output/`

Expected Behavior: Data generated should match `accTests/outputs/badData/` within error value of 0.05. This can be done automatically: `batch_compare_folders.py accTests/outputs/badData accTests/inputs/badData_output/ 1 40`.

Minor Deviations: In the web page you will have to navigate to the batch webpage after logging in.

Linux Source	Linux Executable	Windows Executable	Web Gui
Pass	Pass	Pass	Pass

Performance

Requirement #: 4

Description: The program will perform the computations in a reasonable timeframe.

Acceptance Tests:

Due to the nature of this requirement, it is difficult to set a baseline, as hardware matters. As such the following tests note the use of a Baseline Computer. It has the following relevant hardware:

- Intel i5 460 M at 2.53 GHz
- 4 GB of DDR3 1066 MHZ SDRam
- A 250 GB SSD.

Acceptance Test #6: Performance single calculation
Test Level: Major
Pre-conditions: Installation complete, on Baseline Computer
Procedure:

1. Open the Local GUI
2. Open a stopwatch program of some sort.
3. Follow Acceptance Test #1 up to the end of Step 3.
4. Start the timer.
5. Proceed to the end of Acceptance Test #1.
6. Stop the timer

Expected Behavior: Entire calculation took less than 10 seconds.

Minor Deviations: In the web page you will have to navigate to the upload webpage after logging in.

Linux Source	Linux Executable	Windows Executable	Web Gui
Pass	Pass	Pass	Pass

Acceptance Test #7: Performance batch calculation
Test Level: Major
Pre-conditions: Installation complete, on Baseline Computer
Procedure:

1. Open the Local GUI
2. Open a stopwatch program of some sort.
3. Follow Acceptance Test #5 up to the end of Step 5.
4. Start the timer.
5. Proceed to the end of Acceptance Test #5.
6. Stop the timer.

Expected Behavior: Entire batch calculation (40 sets of 3 channel images) took less than 2 minutes.

Minor Deviations: In the web page you will have to navigate to the batch webpage after logging in.

Linux Source	Linux Executable	Windows Executable	Web Gui
Pass	Pass	Pass	Pass

Portability

Requirement #: 5

Description: The program will run on a variety of platforms, including Windows, OSX, and Linux.

Acceptance Tests:

Acceptance Test #8: Program runs on variety of platforms
Test Level: Major
Pre-conditions: Installation complete
Procedure:

1. On each platform (Windows, OSX, and Linux) Open the Local GUI
2. Run acceptance Tests 1 to 5.

Expected Behavior: No unexpected failures should occur.

Linux Source	Linux Executable	Windows Executable	Web Gui
Pass	Pass	Pass	N/A

OSX Source: Pass

Authentication

Requirement #: 6

Description: The web frontend should have an authentication system to control who has access.

Acceptance Tests

Acceptance Test #9: Check login page
Test Level: Major
Pre-conditions: Installation complete
Procedure:

1. Click on each tab at the top of the webpage
2. Should be required to login for the following pages: “Single Mode”, “Batch Mode”, “Batch Output”
3. Login using the default user created when creating the database. (“admin”, password of “abc”)
4. Should be able to access the following pages now: “Single Mode”, “Batch Mode”, “Batch Output”

Expected Behavior: No unexpected failures should occur.

Linux Source	Linux Executable	Windows Executable	Web Gui
N/A	N/A	N/A	Pass

Automatic Parameter Detection

Requirement #: 7

Description: Parameters for the image processing algorithms can be determined from the input data.

This Requirement was deferred. It was low priority and appears to be more of a math derivation then a programming concern.

Microscope Metadata

Requirement #: 8

Description: The program will be able to determine parameters for the image processing algorithms using metadata from the microscope.

This requirement was deferred, due to lack of files and knowledge on the metadata formats.

Acceptance Test Result Summary

Number of Passes: 33/33

Number of Fails: 0/33

Support Contact

If a show-stopping problem arises, please contact Glen Nelson at ganelson@ualberta.ca. Any requests will be forwarded onto appropriate team members as required.