

# **Requirements Document**

*Biomembrane Team - Nick Klose, Richard Leung, Cameron Mann, Glen Nelson, Omar Qadri, James Wang*

# Project Overview

The Biomembrane Project encompasses the redevelopment of the current MATLAB program, which computes auto/cross/triple-correlations on images of biological membranes retrieved from microscopes. Currently, this program can only analyze singular sets of RGB images, and the processing time is substandard. Analysis of the MATLAB algorithm has led to the discovery of various bottlenecks. It has been decided that the development team will port the code to the Python programming language and refactor it to improve run-time speed. Batch processing is a major feature that will be added to handle more than one set of images at once. The main goal of this project is to improve the efficiency of computation, and these improvements will satisfy that goal.

In addition to an increase in processing speed, there is a demand for a separate web based interface. Sets of images will be uploaded to a central server to be processed through various web browsers. This server can be accessed by individual researchers or students globally via the internet, when given a login name and password. A separate batch page will be made to manage the upload and analysis of multiple sets of data, which will implement a queueing system to determine the order in which to process the jobs.

The Biomembrane program will be used by university level students and professors, as well as researchers. Therefore the user interface design will not be ultra simplistic, but it will maintain efficient input of data and parameters. The redesigned local interface can be downloaded as a package from the server, and it will contain everything necessary to perform both singular and batch tests. The local version of this program will not require connection to the server and works as a standalone means of biomembrane analysis.

## Requirements

### Image Inputs (High)

#### Requirement #: 1

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

**Rationale:** Ease of use for the end user.

**Originator:** Nils Petersen

**Fit Criterion:** Users will be able to upload either singles or triples of images in various formats.

## **Image Processing (High)**

**Requirement #:** 2

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

**Rationale:** The resulting data from the image processing is the primary function of the program.

**Originator:** Nils Petersen

**Fit Criterion:** The program will run the following algorithms:

- FFT
- Auto and cross correlation
- Reverse FFT
- 2D gaussian fitting

## **Batch Processing (High)**

**Requirement #:** 3

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

**Rationale:** Allowing users to process multiple sets of data in one step is more efficient.

**Originator:** Nils Petersen

**Fit Criterion:** Users will be able to upload multiple sets of independent data in a single step.

## **Performance (High)**

**Requirement #:** 4

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

**Rationale:** The computations currently take a significant amount of time; improvements would greatly improve usability for users.

**Originator:** Nils Petersen

**Fit Criterion:** The program will perform the computations faster than the existing MATLAB solution.

## **Portability (High)**

**Requirement #:** 5

**Description:** The program will run on a variety of platforms.

**Rationale:** Make it easier for users to access the program from their own machines.

**Originator:** Nils Petersen

**Fit Criterion:** The web frontend does not need to be portable, but allows users to access the program from anywhere with an internet connection. The local GUI will run on Windows, OSX and Linux.

## **Authentication (High)**

**Requirement #:** 6

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

**Rationale:** Prevent unauthorized and/or wasteful use of resources.

**Originator:** Nils Petersen

**Fit Criterion:** An administrative system will be available to allow the addition and removal of user accounts.

## **Automatic Parameter Detection (Medium)**

**Requirement #:** 7

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

**Rationale:** Remove the need for users to manually enter this information.

**Originator:** Nils Petersen

**Fit Criterion:** Parameters that must currently be manually entered will be automatically determined instead.

# Microscope Metadata (Low)

Requirement #: 9

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

**Rationale:** Remove the need for users to manually enter this information.

**Originator:** Nils Petersen

**Fit Criterion:** Users will be able to upload microscope metadata files which can be used to automatically configure parameters.

## Storyboard

This section provides a visual overview of the tentative layout of the local GUI. Note that in the following images, white squares indicate images and graphs where applicable.

### Local GUI

1. Input tab, three channel images, auto-correlation.

Image Correlation Spectroscopy - [Preview]

Input Parameters Output

Images

All Channels Red Green Blue

Image Settings

Three Channel Images Single RGB Image

Load Red Channel Image

Load Green Channel Image

Load Blue Channel Image

Image Attributes

Image Size: 128 x 128

Average Intensity per Pixel:

Red -	100
Green -	100
Blue -	100

Sample Resolution

64 x 64

48 x 48

32 x 32

Correlation Settings

Auto-Correlation Cross-Correlation Triple-Correlation All

Channels

☒ Red

☒ Green

☒ Blue

Auto-Correlation Parameters

Range:

$g(0)$ :

w:

ginf:

START RESTART SWITCH TO BATCH MODE

## 2. Input tab, single RGB image, cross-correlation.

Image Correlation Spectroscopy - [Preview]

Input Parameters Output

Images

All Channels Red Green Blue

Image Settings

Three Channel Images Single RGB Image

Load RGB Image

Image Attributes

Image Size: 128 x 128

Average Intensity per Pixel:

Red -	100
Green -	100
Blue -	100

Sample Resolution

64 x 64

48 x 48

32 x 32

Correlation Settings

Auto-Correlation Cross-Correlation Triple-Correlation All

Channels

☒ Red-Green

☒ Red-Blue

☒ Green-Blue

Cross-Correlation Parameters

Range:

g(0):

w:

ginf:

START RESTART SWITCH TO BATCH MODE

## 3. Input tab, single RGB image, triple-correlation.

Image Correlation Spectroscopy - [Preview]

Input Parameters Output

Images

All Channels Red Green Blue

Image Settings

Three Channel Images Single RGB Image

Load RGB Image

Image Attributes

Image Size: 128 x 128

Average Intensity per Pixel:

Red -	100
Green -	100
Blue -	100

Sample Resolution

64 x 64

48 x 48

32 x 32

Correlation Settings

Auto-Correlation Cross-Correlation Triple-Correlation All

Triple-Correlation Parameters

Range:

g(0):

w:

ginf:

START RESTART SWITCH TO BATCH MODE

4. Input tab, single RGB image, all correlations.

The screenshot shows the 'Input Parameters' tab of the 'Image Correlation Spectroscopy' software. The interface is divided into several sections:

- Images:** Four empty image boxes labeled 'All Channels', 'Red', 'Green', and 'Blue'.
- Image Settings:**
  - Three Channel Images / Single RGB Image:** A 'Load RGB Image' button.
  - Image Attributes:** A table showing 'Image Size' as '128 x 128' and 'Average Intensity per Pixel' for Red, Green, and Blue channels, all set to '100'.
  - Sample Resolution:** A vertical slider with three positions: '64 x 64', '48 x 48', and '32 x 32'.
- Correlation Settings:**
  - Auto-Correlation / Cross-Correlation / Triple-Correlation / All:** A row of tabs with 'All' selected.
  - Auto/Cross Parameters:** Fields for 'Range:', 'g(0):', 'w:', and 'ginf:'.
  - Triple-Correlation Parameters:** Fields for 'Range:', 'g(0):', 'w:', and 'ginf:'.

At the bottom, there are 'START' and 'RESTART' buttons, a progress bar, and a 'SWITCH TO BATCH MODE' button.

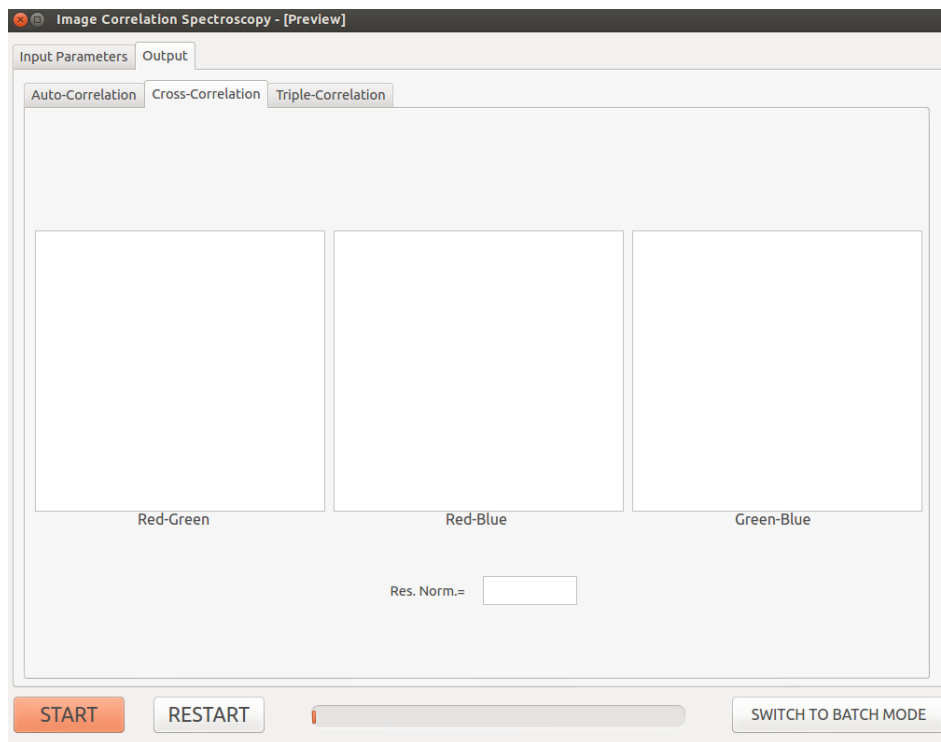
5. Output tab, auto-correlation results.

The screenshot shows the 'Output' tab of the 'Image Correlation Spectroscopy' software. The interface is divided into several sections:

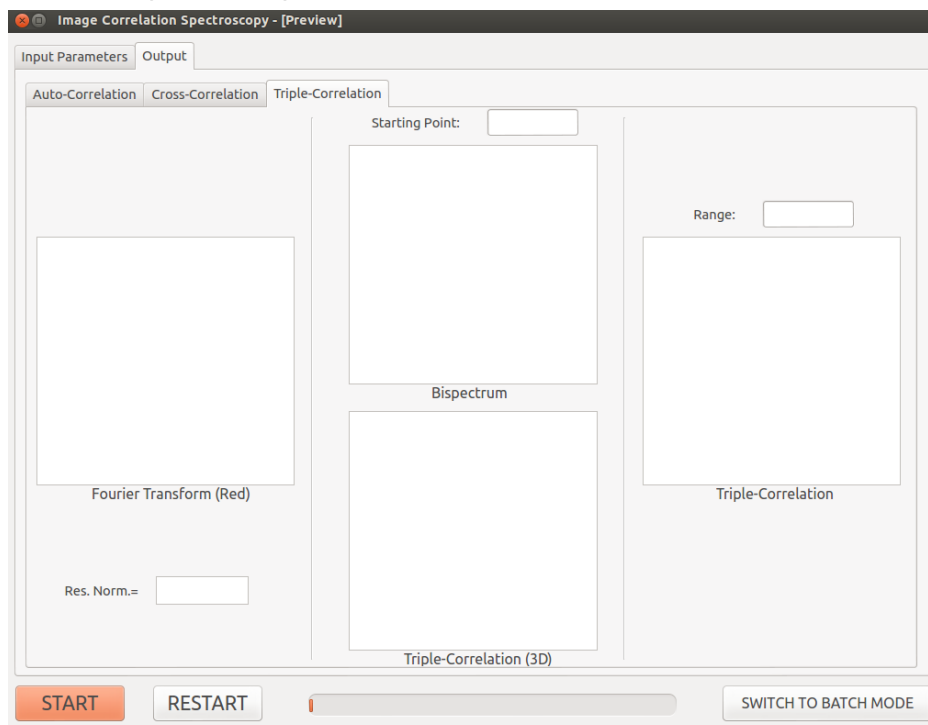
- Auto-Correlation / Cross-Correlation / Triple-Correlation:** A row of tabs with 'Auto-Correlation' selected.
- Output Area:** Three large empty image boxes labeled 'Red', 'Green', and 'Blue'.
- Res. Norm.:** A text label followed by an empty input field.

At the bottom, there are 'START' and 'RESTART' buttons, a progress bar, and a 'SWITCH TO BATCH MODE' button.

6. Output tab, cross-correlation results.

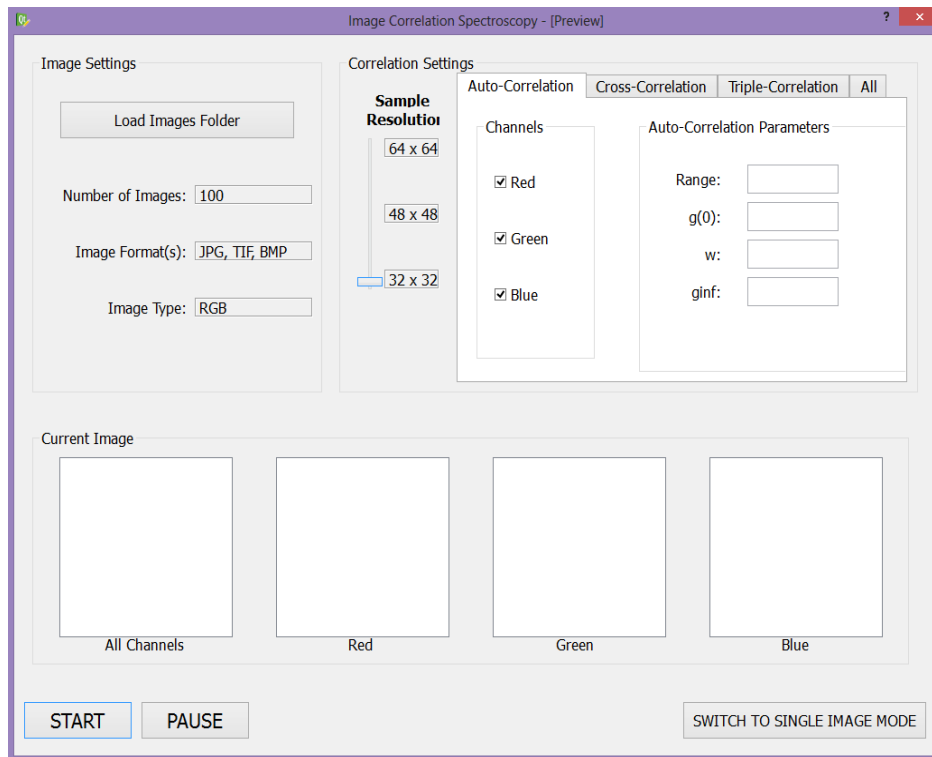


7. Output tab, triple-correlation results.





## 8. Batch mode.

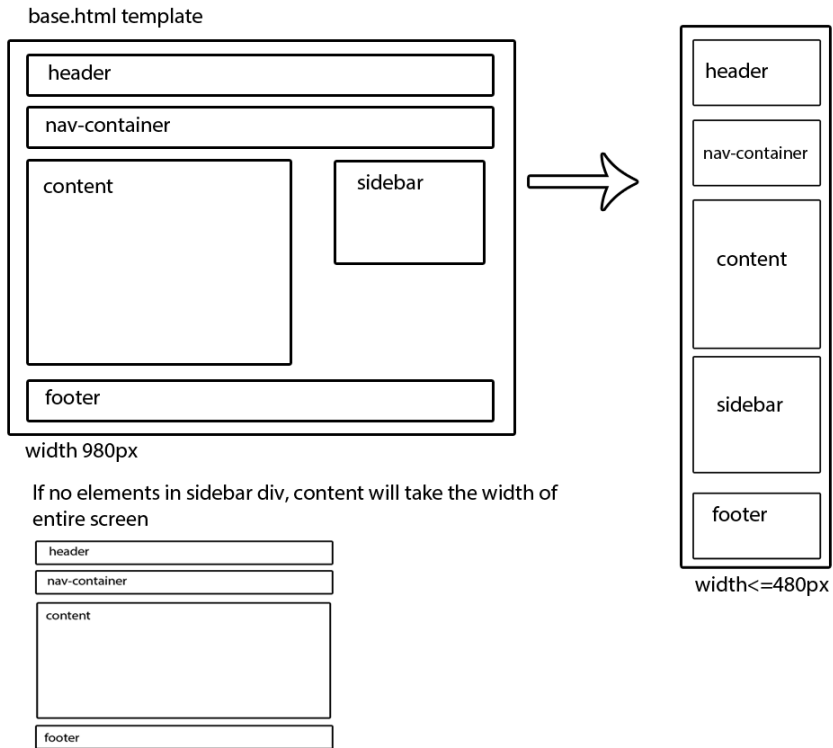


### Storyboard details:

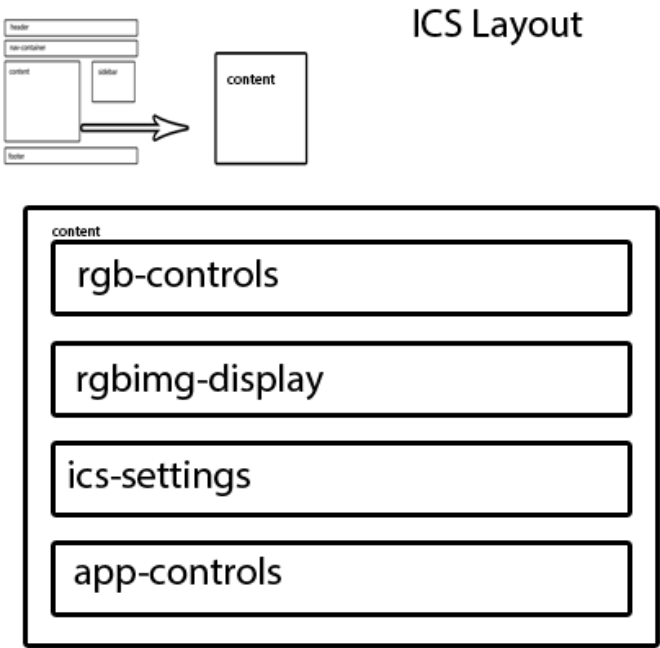
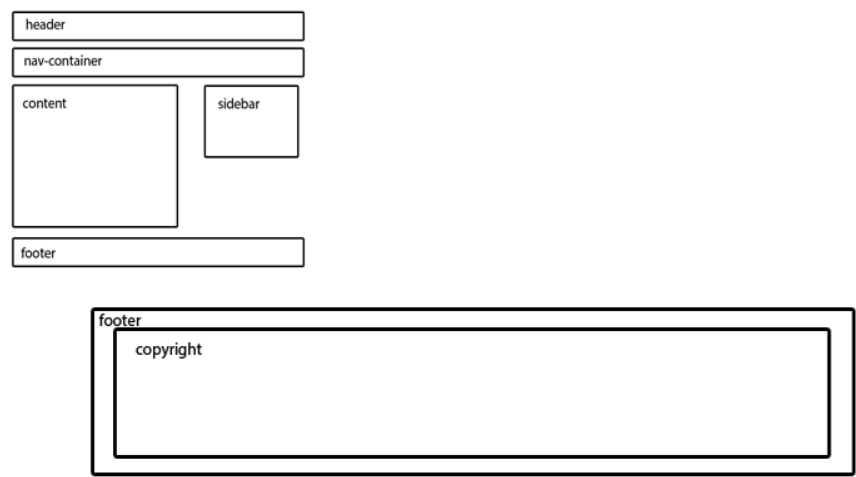
- Processing of the selected correlation based on the input parameters begins once the user clicks Start. If the input parameters are filled in incorrectly, one of the following will happen (depending on ease of programming):
  - The incorrectly filled-in parameters will be highlighted in red
  - A dialog box will show up explaining which parameters are incorrect and why
- The progress bar fills up until all correlations are finished. It has not yet been decided whether the progress bar is for the full set of selected correlations, or if it will restart for each correlation.
- The restart button resets the most recently processed correlation and begins processing it again. The restart button will only be available while processing is taking place.
  - We will need to double-check with the client that this is needed and that there isn't a better method.
- When the user uses one of the "load image" buttons, only files with types compatible with the interface will be selectable. This will prevent the user from loading files with invalid extensions.
- Only a single image of the batch mode is shown as its functionality is very similar to the single-image mode.

## Web GUI

The interface for the web client will be largely the same as the local GUI. As such, only the basic layout of the web interface is shown below.



generic.html extends base.html



# Runtime Exceptions

This section documents possible runtime exceptions and how they can be handled.

1. User clicks START button when parameters are incomplete or partially complete.
  - One of the following:
    - Any parameters which are blank or incorrectly filled in will be highlighted.
    - A dialog box will pop up informing the user of the incorrect parameters.
2. User attempts to load a non-image or corrupted image with a compatible file extension.
  - If there is an exception while the program is attempting to load the image, a dialog box will be shown to the user informing them of the error, and the error will be outputted into an error log. The file will then be unloaded and the user will be required to load a different file.
3. An image file in use is deleted or becomes unreadable.
  - The correlation process halts. A dialog appears detailing the error, and the user is given the opportunity to attempt to continue the correlation or stop the process.
4. The default output directory is read-only.
  - A dialog appears explaining that the disk cannot be written, and the process halts.

# Intellectual Property

The Biomembrane project will be licensed under the “Intellectual Property Agreement” created for the 401 class.

This Agreement, effective the 1st day of April 2013, is entered into by and between the Client (hereinafter “client”) and the students of the Biomembrane team (hereinafter “the development team”), in order to establish terms and conditions concerning the completion of the Biomembrane Application (hereinafter “The Application”) which is limited to the application domain of application-domain (hereinafter “the domain of use for the application”)

It is agreed by the client and the development team that all domain specific knowledge and compiled research is the intellectual property of the client, regarded as a copyrighted collection. The framework and code base created by the development team is their own intellectual property, and may only be used for the purposes outlined in the documentation of the application, which has been provided to the client. The development team agrees not to use their framework for, or take part in the development of, anything that falls within the domain of use for the application, for a period of 6 (six) months after the signing of this agreement.

## Project Glossary

- **Application-Domain** - The domain of use for the “Biomembrane” application.
- **Application, the** - The “Biomembrane” application, based off of the ICS software.
- **Auto-Correlation** - A cross-correlation of a waveform with itself.
- **Client** - The client of the project (Nils Peterson)
- **Cross-Correlation** - A measure of similarity between two waveforms.
- **FFT (Fast Fourier Transform)** - An algorithm used to compute the discrete Fourier transform and its inverse.
- **ICS Software** - Image Correlation Spectroscopy Software.
- **MATLAB** - A high-level language and interactive environment for numerical computation, visualization, and programming.
- **Metadata** - A collection of raw data.
- **Triple-Correlation** - A statistical analysis used to search for nonlinear interactions between waveforms.

## Repairs

### Repair #1: March 6, 2013

- Revamped storyboard and added multiple images to show functionality. Images are screenshots of current version of PyQt interface as this is likely a better representation of the final interface. Added explanation to clarify storyboard where needed.
- Updated project glossary to contain more definitions.
- Added copyright information for Python modules used within this project.
- Added section detailing possible runtime exceptions.
- Added web interface information and storyboard.
- Added batch mode storyboard.

## Resources

Image Correlation Spectroscopy Software. “ICS Soft” and ICS with Poisson N  
<https://github.com/UniversityOfAlberta/BioMembrane/tree/master/ICS>

## Research Material

Petersen, N. O., Hoddellius, P. L., Wiseman, P. W., Seger, O., & Magnusson, K. (1993).  
Quantitation of Membrane Receptor Distributions by Image Correlation Spectroscopy: Concept

and Application. Biophysical Journal, Volume 65, 1135-1146.

Keating, E., Brown, C. M., & Petersen, N. O. (2002). Mapping Molecular Interactions and Transport in Cell Membranes by Image Correlation Spectroscopy. *Mapping Molecular Interactions by Image Correlation Spectroscopy*.

Heinze, K. G., Jahnz, M., & Schwille, P. (2004). Triple-Color Coincidence Analysis: One Step Further in Following Higher Order Molecular Complex Formation. Biophysical Journal, 86, 506–516.

Kolin, D. L., & Wiseman, P. W. (2007). Advances in Image Correlation Spectroscopy: Measuring Number Densities, Aggregation States, and Dynamics of Fluorescently labeled Macromolecules in Cells. Cell Biochem Biophys, 49, 141-164.

Nohe, A., & Petersen, N. O. (2003). Analyzing protein–protein interactions in cell membranes.

Ridgeway, W. K., Millar, D., & Williamson, J. (0). Accompanying: The spectroscopic basis of Fluorescence Triple Correlation Spectroscopy, S1-S7.

Ridgeway, W. K., Millar, D. P., & Williamson, J. R. (2012). The Spectroscopic Basis of Fluorescence Triple Correlation Spectroscopy. Journal of Physical Chemistry B, 116, 1908–1919.

All material can be found at

<https://github.com/UniversityOfAlberta/BioMembrane/tree/master/docs>

## External Python Libraries Used:

NumPy (2012) Numpy developers (Version 16.2) [Open Source Python Library]

<http://www.numpy.org/>

SciPy (2012) Community Project sponsored and supported by Enthought (Version 0.11.0) [Open Source Python Library] <http://www.scipy.org/>

Matplotlib (2012) John Hunter (Version 1.20) [Python Plotting Library] <http://matplotlib.org/>

Django (2012) Django Software Foundation (Version 1.43) [Web Framework]

<https://www.djangoproject.com/>

## NumPy

Is an extension for python which provides support for scientific computations of large n-dimensional arrays and matrices. Also has libraries to calculate correlation and Fourier transformations functions we need. Numpy also has tools to integrate Fortran and C/C++ code.

## **SciPy**

Is an open source library of mathematical tools and algorithms designed for Python. It is used perform the curve fitting in our functions.

## **Matplotlib**

Used to plot graphs and shares the same look and feel as graphs generated in Matlab.

## **Django**

An open source python web framework that has powerful backend and frontend features.

## **PyQT**

A GPL-licensed python framework for development of GUIs for multiple platforms.