
Software Requirements Specifications Documentation

Release latest

Jul 21, 2019

Contents: 1. Introduction: 2. Overview: 3. Input: 4. Output: 5. Application
functionality overview: 6. Web application maintenance:

1	1.1 Introduction	3
2	Project Scope	5
3	1.2 Purpose	7
4	2. Overview	9
5	2.1 Model-View-Controller	11
6	2.2 Interactions Between Components	13
7	2.3 High-level description of the software and functional requirements	15
8	3. Input	17
9	4. Output	19
10	5. Application functionality overview	21
11	6. Web application maintenance	23
12	7. Functional requirements	25
13	7.1 Hardware requirements for interface	27
14	7.2 Software requirements for the software	29
15	7.3 Communication protocols and interfaces	31
16	8. Intended User	33
17	8.1 Software Attributes intended for user	35
18	9. Functional requirements use	37
19	9.1 Flask	39

20	9.2 Function 2	41
21	10. Constraints	43
22	11. Other Non-functional Requirements	45
23	12. Change cases	47
24	13. Milestones	49
25	14. Appendices	51
26	15. References	53
27	Indices and tables	55

Group Information

Group Simplynovation

Group_member

Enock Khondowe (201636110139)enockkhondowe94@yahoo.com

Samantha Rusike (201632120140)samrusike@gmail.com

date 01 May 2018, version 1.0

CHAPTER 1

1.1 Introduction

This document describes the Little Hill Lab's initial requirements for an online application (Oh My Genes) which allows our scientists to upload gene expression files and quickly get differentially expressed genes.

This SRS for Oh My Genes is for developers, project managers, salesmen, users and testers. The article mainly introduces the overall description, external interface requirements, system features and other non-functional requirements. I suppose the developers and project managers to read the whole article carefully and salesmen pay attention to overall description especially. Users and testers read the system features carefully.

CHAPTER 2

Project Scope

Oh My Genes is used to identify differentially expressed genes given a gene expression file containing two cell samples. This document basically shows the visual idea of the OMG web app and its functionality requirements.

CHAPTER 3

1.2 Purpose

why do we develop this software? This web application targets Biologists in our lab (potentially worldwide) with the purpose to identify differentially expressed genes given a gene expression file containing two cell samples. Functionally the web application has a simple interface with a single button [Upload and GO]. Our scientists upload a plain text file containing gene expression levels from two samples, representing two experimental conditions. Accepting the file, the software will return a table of differentially expressed genes and a scatter plot of these genes whose X-axis is control and Y-axis is treatment. If an invalid gene expression is given, the web application returns a page informing the user to provide the correct format. So, the purpose mainly scopes from;

- i. Control sample - a cell sample prepared in its normal condition.
- ii. Treatment sample - a cell sample treated by special chemicals, or in which some genes are altered.
- iii. Differentially expressed genes - the genes which have significantly different expression levels between two samples.
- iv. Up-regulation - a gene is said to be up-regulated if it has higher expression in treatment than in control.
- v. logFC - log fold change of gene expression. $\log_2 [T/C]$, where T is the gene expression level from a treatment sample, while C is the gene expression level from a control sample.

CHAPTER 4

2. Overview

2.1 Model-View-Controller

Model–View–Controller (MVC) is an architectural pattern for implementing user interfaces. It divides an application into three interconnected parts: The Model, the View, and the Controller. Separating the “internal representations of information from the ways that information is presented to and accepted from the user” allows us to increase modularity for simultaneous development and code reuse

Each of the components is defined as follows:

Component Description

Model -handles application data and data management -central component of MVC

View -can be any output representation of information to user -renders data from model to user interface

Controller -accepts input and converts to commands for model/view

2.2 Interactions Between Components

The MVC design pattern also defines interactions between components.

- The model stores data files for the DNA samples that is retrieved according to commands from the controller
- The view generates output that is the scatter diagram and table of the DNA samples for the user based on changes in the model
- The controller acts on both model and view; it sends/uploads commands/files to the model to update its state and to the view to change information presented to users

2.3 High-level description of the software and functional requirements

For a basic description now, OMG web app perspective is to design an application preferably using Flask framework(<http://flask.pocoo.org/>). The web application has a simple interface with a single button [Upload and GO]. Our scientists upload a plain text file containing gene expression levels from two samples, representing two experimental conditions. Accepting the file, the software will return a table of differentially expressed genes and a scatter plot of these genes whose X-axis is control and Y-axis is treatment.

If an invalid gene expression is given, the web application returns a page informing the user to provide the correct format.

The Pictures below shows the Code in Python using flask for the web Application “Oh My Genes”.

3. Input

A valid submitted gene expression file has the following format. It is a TAB-delimited, plain text file with three columns (see the attached file for a full example). The file contains an optional head line, followed by each gene's expression in a control sample (e.g., ControlSample) and in a treatment sample (e.g., KnockOutSample).

gene_id Control Sample Knockout Sample

AT1G01010 1.198558083 2.036161827

AT1G01020 13.75736234 13.370796

AT1G01030 0.833779536 0.203616183

AT1G01040 9.58846466 7.126566394

AT1G01046 0 0

AT1G01050 23.81482799 21.10821094

AT1G01080 28.34850421 25.24840665

AT1G01090 58.26034505 42.96301455

AT1G01100 1066.508249 1308.030358

AT1G01110 2.709783491 1.425313279

AT1G01060 0.625334652 1.221697096

AT1G01070 1.719670292 0.950208853

4. Output

The web application displays a table and a scatter plot given a gene expression file.

The table contains a list of differentially expressed genes with the following format:

```
gene_id control_sample treat_sample log_2[FC] AT1G01010 1.198558083 2.036161827 0.76
```

```
gene_id ControlSample KnockOutSample log_2[FC]
```

```
AT1G01010 1.198558083 2.036161827 0.76
```

```
AT1G01020 13.75736234 13.370796 0.77
```

```
AT1G01030 0.833779536 0.203616183 0.78
```

```
AT1G01040 9.58846466 7.126566394 0.79
```

```
AT1G01046 0 0 0
```

```
AT1G01050 23.81482799 21.10821094 0.80
```

```
AT1G01080 28.34850421 25.24840665 0.81
```

```
AT1G01090 58.26034505 42.96301455 0.82
```

```
AT1G01100 1066.508249 1308.030358 0.83
```

```
AT1G01110 2.709783491 1.425313279 0.85
```

```
AT1G01060 0.625334652 1.221697096 0.86
```

```
AT1G01070 1.719670292 0.950208853 0.87
```

5. Application functionality overview

Action: There are two buttons on the web application (upload and go). Users upload a plain text file containing gene expression levels from two samples, representing two experimental conditions. Result: Accepting the file, the software will return a table of differentially expressed genes and a scatter plot of these genes whose X-axis is control and Y-axis is treatment. If an invalid gene expression is given, the web application returns a page informing the user to provide the correct format. The software needs to draw the uploaded data to graph. The points on the graph will show the users some information, such as gene id and name. This web application can update the data intelligently and save the new genes into the data base. Every time the users upload a new file, the application should save and update the gene list. Regular log records and updates are designed to better improve the software system, which is lower in comparison, so the priority is low in the software system. The last main user group to manage and maintain the data directly in the background is the database administrator. Although the group is not very large, its importance cannot be ignored. After all, the highlight of the software system is the database, so its priority is the highest. I believe this is also the center of the application database system software.

6. Web application maintenance

As the web application will be constantly growing and evolving it is not simple as there it has to be maintained keeping the application secure, stable and up-to-date takes time so OMG app will undergo

1. BUG fixes
2. Third party API updates
3. Security patches and updates
4. Implementing new functionality
5. Hardware upgrades/ app scaling
6. Monitoring

All this regular web application maintenance retains customers and balances costs

CHAPTER 12

7. Functional requirements

CHAPTER 13

7.1 Hardware requirements for interface

-64-bit PC with TCP protocol - TCP/IP stands for Transmission Control Protocol/Internet Protocol, which is a set of networking protocols that allows two or more computers to communicate. The Defense Data Network, part of the Department of Defense, developed TCP/IP, and it has been widely adopted as a networking standard.

7.2 Software requirements for the software

1. Windows operation system
2. Web browser
3. SQL server

7.3 Communication protocols and interfaces

1. TCP protocol
2. Secure transmission and techniques
3. Microsoft- edge browser
4. File transfer rate

CHAPTER 16

8. Intended User

The OMG web app is targeted for Biologists in our lab (potentially worldwide).

8.1 Software Attributes intended for user

This includes user documentation that is the,

1. User manual
2. Tutorial books
3. online platform

CHAPTER 18

9. Functional requirements use

Flask is the best Python web framework for our use case. As the name implies, the Flask microframework is a lightweight web framework that we can extend to get the functionality we require. From the documentation: Flask aims to keep the core simple but extensible. Flask won't make many decisions for you, such as what database to use. Those decisions that it does make, such as what templating engine to use, are easy to change. Everything else is up to you, so that Flask can be everything you need and nothing you don't. Taking a build-your-own framework approach to web development allows us to get projects off the ground quickly; we only use the extensions we require for our specific use case. With just a few lines of code, we can create a website with:

1. URL routing
2. template rendering
3. two-way client communication
4. redirects and error handling
5. logging

The Flask documentation has great advice on how to grow and become big with Flask. We can also leverage the list of common design patterns to ensure we are following best practices as outline by the project contributors.

CHAPTER 20

9.2 Function 2

1. Performance requirements - It can support large amount of calculations and the response time should in 3 seconds whether the file is large or not. Users can add or delete the data and the web application can save them timely. And the application can't occupy large space. And the picture of the result should be clear either the points or the lines and what the coordinates present for and the correct units.
2. Safety requirements - The data both from users input or from database should be safe. And the whole file transfer to others should be secured as we can encrypt it. And it is important to create backups in case of data loss this also helps when the application has errors, backups can decrease the loss. And the application should proceed the user authentication.
3. Security requirements - For security the web can be set to measures of being accessed by authorized users though use of passwords and verifications.
4. Software attributes - This part will contain two requests of the user documentation: 1. User's manual: electronic document. 2. Tutorial: electronic document.

CHAPTER 21

10. Constraints

- i. How to set a threshold for logFC
- ii. Accessible only through Firefox, Chrome, and Safari
- iii. Web space less than 1GB
- iv. Budget less than 10,000 USD
- v. System downtime less than 30 minutes per year
- vi. OMG Web App Accessible through:
- vii. Firefox, Chrome, and Safari

11. Other Non-functional Requirements

This part will show you other non-functional requirements. It contains response time, aesthetic aspects and confidentiality policy

1. Confidentiality policy - This document remains a copyright of Symplynovation - Zhejiang Normal University - Spring 2018
2. Response time - <5 seconds
3. Aesthetic aspect
 - The OMG web app should look and feel critical to its overall success that is it should be easy to use and intuitive to use so basically execution is everything
 - OMG web app Aesthetic aspects functionality here is to cover on the user-friendly aspects and Human Computer Interaction where our main objective is to create effective website where the user quickly and efficiently can obtain the desired pieces of information without being delayed by long downloading times or blind alleys when navigating on the site.

CHAPTER 23

12. Change cases

Change cases will be done the lines of our web app appearance, also means to improve processing time possible by our coding style

CHAPTER 24

13. Milestones

- i. Submit SRS for review by 1 May 2018
- ii. Get SRS approved by May 15th 2018
- iii. Get design done by May 15th 2018
- iv. Get coding done by May 31st 2018
- v. Acceptance tests by 1 June 2018
- vi. Release by ... 15th June 2018

14. Appendices

- i. Control sample - a cell sample prepared in its normal condition.
- ii. Treatment sample - a cell sample treated by special chemicals, or in which some genes are altered.
- iii. Differentially expressed genes - the genes which have significantly different expression levels between two samples.
- iv. Up-regulation - a gene is said to be up-regulated if it has higher expression in treatment than in control.
- v. logFC - log fold change of gene expression. $\log_2 [T/C]$, where T is the gene expression level from a treatment sample, while C is the gene expression level from a control sample.
- vi. Change cases- are changes that, although not explicitly mentioned in the user requirements document, might happen in the future.
- vii. Flask- is also known as “micro-framework” because it uses simple core and uses extension to add other functions. Flask has no default database and form validation tools
- viii. User guide or user’s guide - also commonly known as a manual, is a technical communication document intended to give assistance to people using a particular system.
- ix. TCP/IP stands for Transmission Control Protocol/Internet Protocol, which is a set of networking protocols that allows two or more computers to communicate. The Defense Data Network, part
- x. Model–View–Controller (MVC) is an architectural pattern for implementing user interfaces. It divides an application into three interconnected parts: The Model, the View, and the Controller. of the Department of Defense, developed TCP/IP, and it has been widely adopted as a networking standard.
- xi. model stores data that is retrieved according to commands from the controller
- xii. view generates output for the user based on changes in the model
- xiii. controller acts on both model and view; it sends commands to the model to update its state and to the view to change information presented to users • The term aesthetics in the context of this article covers visual, sound and interactive means of effect. However, the article focuses primarily on the visual means of effect in terms of colors, typography, design, pictures, video clips, flash animations, etc • Human Computer Interaction - HCI

CHAPTER 26

15. References

1. <https://alysivji.github.io/flask-part2-building-a-flask-web-application.html><https://alysivji.github.io/flask-part2-building-a-flask-web-application.html>
2. <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>
3. https://www.researchgate.net/publication/238106570_The_Role_of_Aesthetics_in_Web_Design

CHAPTER 27

Indices and tables

- `genindex`
- `modindex`
- `search`