
Oh My Genes Documentation

Release 0.0.1

Universe-Bob

Apr 27, 2018

Contents

1	Introduction	1
1.1	Purpose	1
1.2	Overview	1
1.3	User Characteristics	1
1.4	Terminologies & Abbreviations Explanation	1
2	Functional Requirements	3
2.1	Input	3
2.2	Output(Data Analyzing)	3
3	Non-functional Requirements	5
3.1	Response Time	5
3.2	Aesthetic Aspects	5
3.3	Confidentiality Policy	5
4	Constraints	7
4.1	Browser Compatibility	7
4.2	Space Complexity	7
4.3	Budget	7
4.4	System Downtime	7
5	Change Cases	9
6	Milestones	11
7	Appendices	13
8	References	15
9	Group Information	17
9.1	Group Name:	17
9.2	Group Member:	17
10	Indices and tables	19

This document is intended to specify a set of requirements for the project OMG, which means “Oh My Genes”, a web application for identifying genes’ expression differential.

1.1 Purpose

To identify differentially expressed genes given a gene expression file containing two cell samples. The main purpose is to help biologists analyze their experiment data.

1.2 Overview

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.

1.3 User Characteristics

- **Client User:** Biologists who devote themselves studying the expression of genes.
- **Site Maintainer:** Technicians who have the basic knowledge of flask and python to maintain a website.

1.4 Terminologies & Abbreviations Explanation

To introduce the project and specify the requirements in detail, this section is aimed to explain some terminologies and abbreviations, for the sake of better understanding of the content of the project.

1.4.1 Terminologies

- *Control sample* - A cell sample prepared in its normal condition.
- *Treatment sample* - A cell sample treated by special chemicals, or in which some genes are altered.
- *Differentially expressed genes* - The genes which have significantly different expression levels between two samples.
- *Up-regulation* - A gene is said to be up-regulated if it has higher expression in treatment than in control.

1.4.2 Abbreviations

- *OMG* - Oh My Genes, which is the name of this project.
- *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

Functional Requirements

2.1 Input

A valid submitted gene expression file has the following format:

- It is a TAB-delimited, plain text file with three columns;
- 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).

See the txt file pattern as following:

gene_id	ControlSample	KnockOutSample
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
AT1G01060	0.625334652	1.221697096
AT1G01070	1.719670292	0.950208853
AT1G01080	28.34850421	25.24840665
AT1G01090	58.26034505	42.96301455
AT1G01100	1066.508249	1308.030358
AT1G01110	2.709783491	1.425313279

2.2 Output(Data Analyzing)

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

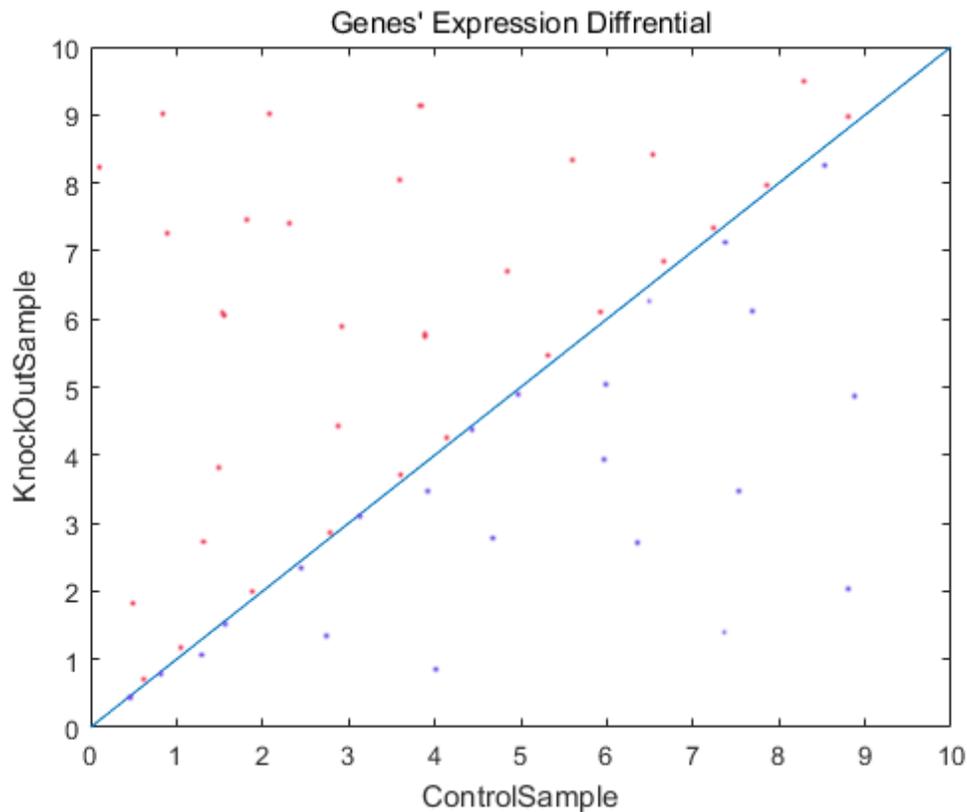
2.2.1 Table Analyzing

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

gene_id	control_sample	knockout_sample	log ₂ [FC]
AT1G01010	1.198558083	2.036161827	0.76

2.2.2 Plot Analyzing

The scatter plot displays differentially expressed genes. The X-axis is Control, and Y-axis is KnockOut. Replace 'Control' and 'KnockOut' with appropriated column names if provided in the uploaded file. The up-regulated genes are shown in red dots, and down-regulated genes are shown in blue.



Non-functional Requirements

For this section, we are going to clarify some non-functional requirements for a better using experience.

3.1 Response Time

Since this web application is used for biologists all around the world, the server maybe have to stand a heavy load of requests thus may cause some problems such as the system will down for several times or lose response to its clients. On the other hand, if users have a big amount of data to analyze, the time complexity must be taken into consideration. For the optimization aspects, our algorithm should be sufficiently effective to deal with this situation. After discussion, we have decided the limit of response time, it should **less than 5 seconds** for the most using cases.

3.2 Aesthetic Aspects

As a web application to solve biological analyzing. The interface should be designed as simply as possible and make the scatter plot and table more distinct.

3.3 Confidentiality Policy

As a public web application, the security of data must be a significant part to be considered. Especially for biologists, every experiment result is derived uneasily. Therefore, when users are using the application, their data must be used and stored properly. In other words, **the web application must maintain the experiment data's reliability, integration and secrecy and the experimental data will not be divulged without the agreement of the owner.**

For scientific and analyze uses, this application is orientated to various users(biologists) all over the world. Before developing the project, we have to take some main constraints into account so that we can come up any useful ideas to deal with them in the future.

4.1 Browser Compatibility

Since the application is designed for a wildly used application, it should have a cross platform compatibility to satisfy various users. Thus the application should be **accessible through Firefox, Chrome, and Safari**.

4.2 Space Complexity

Not only the time complexity should be considered, but also the space complexity. There are two aspects for space complexity. One is the size of the whole web application. **The application should not larger than 1GB**. For another aspect, when the application starts data analyzing, memory occupancy must be limited under a certain level to make sure the system work in a proper way.

4.3 Budget

Budget less than 10,000 USD(To Be Specified).

4.4 System Downtime

System **downtime less than 30 minutes per year** in order to satisfy users' large demand of data analyzing.

CHAPTER 5

Change Cases

1. In the future, maybe we can support Excel files, not only just a txt file.
 2. Analyzed result downloading can be supported in the long run.
 3. More functions will be provided from the application.
- (To Be Added...)

CHAPTER 6

Milestones

1. Submit SRS for review by **May 1st**.
2. Get SRS approved by **May 8th**.
3. Get design done by **May 10th**.
4. Get coding done by **May 24th**.
5. Acceptance tests by **June 1st**.
6. Release by **June 15th**.

CHAPTER 7

Appendices

Date	Change Log
April 26th	1. Specified basically functions and interface for the application.
	2. Get non-functional requirements in detail.
	3. Clarified some constrains about the application.
	4. Imagined some change cases and tried to find ways to slove them.
	5. Decided the project developing milestones.

CHAPTER 8

References

1. *DeepTools example for ReadTheDocs:* <http://deeptools.readthedocs.io>
2. *SRS example pattern:* <http://karaage.readthedocs.io/projects/karaage-srs/en/latest/>

Group Information

9.1 Group Name:

BuiGia

9.2 Group Member:

1. (Bob 201632120126): 827030988@qq.com
2. (Universe 201632120128): universe_black@qq.com

CHAPTER 10

Indices and tables

- `genindex`
- `search`