PROGRAMMING LIFE CONTEXT
DELFT UNIVERSITY OF TECHNOLOGY

DNA is Not an Acronym

Product Vision

Joël Abrahams - joelabrahams - 4443268
Georgios Andreadis - gandreadis - 4462254
Casper Boone - cboone - 4482107
Niels de Bruin - ndebruin - 4375440
Felix Dekker - fwdekker - 4461002

May 2017

Contents

1 Introduction

Medical and biological research play a vital role in society. We expect it to come up with cures for the multitude of diseases we're confronted with, and we would be at a loss without them. Central to these fields is the analysis of DNA - the 'source code' of life. To be able to make informed, well-founded decisions, scientists in the field need to be able to read DNA like a book. A book, however, does not fit the proportions of data we are dealing with here. This is an opportunity for software to fulfil the task. Unfortunately, the software available at the moment is lacking in many regards, including in scalability and intuitiveness.

To this end, we present `Hygene`, a DNA sequence visualizer. It will enable scientists to analyse large DNA sequence alignment graphs, as well as easily identify mutations. Additionally, the tool will allow users to annotate their graphs and share these annotations with fellow users.

In this document, we illustrate our vision of this product. We start by listing the stakeholders of our product, in section 2. In the next two sections, we address the functional and non-functional requirements our product will adhere to. Next, in section 5, we compare the envisioned product against existing work in the field. Finally, we describe the timeframe and budget of this project in section 6.

2 Stakeholders

`Hygene` is primarily designed to aid biomedical researchers in visualizing, interactively exploring, and editing complex sequence alignment graphs. Our goal is to enable scientists to process very large datasets without a supercomputer, provide insights into genetic data of variable size and nature, and in doing so offer new insights in research benefiting society.

Next to the more general public of DNA researchers, our project supervision team is also a key stakeholder of the product:

- Dr. Thomas Abeel is the context coordinator of the "Programming Life" context and primary client. He is responsible for judging our product and providing feedback during the design and development phase of the product.
- The Teaching Assistants Jasper Linthorst, Tom Mokveld, and Sander van de Oever will assist us by providing information and direct feedback during the course of the project.

The last stakeholder and our first potential high-profile customer is the Broad Institute, which is an institution "committed to meeting the most critical challenges in biology and medicine, [allowing scientists to] pursue a wide variety of projects that cut across scientific disciplines and institutions".[1] Our primary stakeholder, Thomas Abeel, is closely connected to the Broad Institute and has indicated that there exists interest in sequence alignment graph visualization software capable of dealing with large datasets within this organization. At the end of the project, we hope to present our software to scientists at the Broad Institute and potentially obtain valuable feedback to further improve `Hygene`.

[1] About Us — Broad Institute. (n.d.). Retrieved from https://www.broadinstitute.org/about-us

3 Functional Requirements

Due to advances in DNA sequencing technologies, we are able to sequence more and more genomes. As more data is generated, it becomes difficult to analyse the data. The human genome for instance consists of 3 billion base pairs.[2] Such data may be easily processed by a powerful computer, but is infeasible for humans to analyse and draw meaningful conclusions from without taking special precautions.

3.1 Visualising and Interacting with Large Genome Data

Our product aims to utilize the "human visual system's highly tuned ability to see patterns, spot trends, and identify outliers" (Heer, Bostock, & Ogievetsky, 2010). By displaying the genome as a graph, and only displaying small sections at a time, we allow the user to identify what parts are important for their respective research.

Other means of visualisation, such as using different colours, can further help users of the application quickly identify parts of interest within the data. Furthermore, by allowing the user to interact with the data, we help them to gain further insight into the data.

3.2 Comparing Similar Genomes and Identifying Mutations

A key part of biomedical research is spotting trends (National Institutes of Health (US), 2007). A key part of spotting trends is the ability the compare large amounts of data and look for similarities and dissimilarities. That is another need that our product aims to address. With our product, researchers should be able to compare different DNA sequences and to spot differences, be it in a visual manner or by using differences calculated by the application itself.

Spotting differences and in turn identifying outliers allows researchers to identify different mutations, a key part of biomedical research (National Institutes of Health (US), 2007). Identifying mutations can help with identifying diseases, such as Cystic Fibrosis and Sickle-Cell Anemia (Encyclopaedia Britannica, 2014).

3.3 Annotating and Sharing Important Findings

A key part of research is the ability for researchers to quickly and efficiently share their findings. That is where annotations come in. Annotations allow researchers to annotate parts of the DNA which they believe to be of interest. These annotations make it easier to identify key parts of the data in future, be it by the annotator or by other researchers.

[2] Human Genome Project Completion. (n.d.). Retrieved from https://www.genome.gov/11006943/human-genome-project-completion-frequently-asked-questions/

3.4 Reliability

Lastly, the application should be reliable. To help researchers accurately analyse the data, the software should be easy to use and reliable. Therefore the application should not exhibit behaviour the user doesn't expect. Furthermore, data must be loaded and stored in a secure manner, as data corruption can result in inaccurate results and in turn compromise research.

4 Product Attributes

In this section we will discuss the aspects of our product that are most crucial to the client. We will describe why these aspects are important and how they influence the product development. First, we will discuss the need for the product to work with large sets of data, then we will explain the importance of usability of the product, and finally we will elaborate on the use of the product on multiple platforms.

4.1 Large datasets

The most important aspect of our software solution is the capability to work with very large datasets. The amounts of DNA data that we want the user to be able to inspect does not fit in main memory. Since we want customers to be able run our visualizer on commodity hardware, and cannot expect that they will have more than 16 GB of RAM available, we need an efficient way to load fragments of the data from the file that the user provided and visualize it on the screen. Simply loading all available data into memory is not a feasible option.

To facilitate efficient data loading, we will design a data structure tailored for this situation. This data structure will be able to keep general information about the whole provided data set, and will, on demand, allow the program to load more detailed information about certain segments or mutations.

Furthermore, we will make sure that we do not store any more (detailed) information than is needed at a certain point. This means we only keep detailed information for data that is currently being visualized and that we will not store unnecessary data about things that are, at that moment, not visible to the user.

4.2 Usability

Another aspect that is crucial for the customer is ease of use. Existing or similar software (as covered in section 5) often does not provide a good user experience, as we've found them to lay the focus more on functionality than on usability.

A suitable definition of usability was developed by Ferre, Juristo, Windl, and Constantine (2001). They propose the following usability attributes:

Learnability It is important that users of our product can get to work quickly, and do not need an extensive training. To ensure a quick learning process, we will show an introductory tutorial when a user uses the product for the first time.

Efficiency Performing actions should take as little time as possible. This means that all functionality should be easy to find and not require much cognitive work to use.

User retention over time If users only use our application sporadically, they should not have to go through a long learning process to get familiar with

the product functions every time they use it. This means the user interface should be comparable to common user interfaces found in frequently used software. It should also be comparable to other software products that target a scientific audience.

Error rate If an application is easy to use, the user will be able to immediately find the feature they are looking for, and not make many errors in terms of going in the wrong direction when trying to perform that action. This can again be solved by providing a familiar user interface, but also by making sure that options are not hidden behind complicated menus.

Satisfaction Although hard to measure objectively, the most important aspect of usability is the overall impression of the user. By optimizing the previous aspects, we hope that the user will be satisfied in terms of usability.

4.3 Cross platform support

Finally, it is important that the product can be used on multiple platforms. We want our users to have a broad choice of platform they want to work on.

There are two ways to develop an application that works cross-platform: develop specific versions for every operating system, or write code that works on multiple platforms but that might have little platform specific parts (Cusumano & Yoffie, 1999). We have chosen to do the latter. A consequence of this is the requirement that code needs to be tests on every supported platform, in order to guarantee that it will work properly in all possible configurations.

We want users of our product to be able to work on the following platforms: Windows 7 and higher, macOS 10.11 and higher, and a limited set of Linux distributions (Debian-based and Fedora). The users need to have the Java Runtime Environment (version 8) installed, as this is the platform that we will be mainly relying on for cross-platform compatibility.

5 Related Work

5.1 Competitors

Bandage[3] focusses on interactive visualization of *de novo* genome assemblies and is able to visualize assembly graphs with connections (Wick, Schultz, Zobel, & Holt, 2015). One particular application is to analyse circular bacterial chromosomes (Loder & Scott, 2015).
Our program is similar in that it shows genome assemblies and visualizes the mutations. However, our program cannot display assembly graphs with connections. While this may sound like a drawback, the focus on a particular type of graph allows us to focus on and add specific features for that type of graph.

Cytoscape[4] is a general platform for complex network analysis and visualization, though it was originally designed for biological research (Bastian, Heymann, & Jacomy, 2009). In particular, it allows its users to visualize the interactions between different genes in a genome.
While our program also visualizes graphs, it works at a different semantic level than Cytoscape. While our program shows the mutations that can occur in a genome, Cytoscape visualizes the interaction between different genes in the genome.

Gephi[5] is a network analysis and visualization software package. It can be used to display and explore large networks and their data in real-time.
Gephi is very similar to Cytoscape, and as such the differences with our program are similar as well.

Last year's context project has seen multiple other applications in this category. These projects implement the basic requirements as listed in section 3, and other features as well.[6]

5.2 General Comparison

The programs mentioned above can be characterized as being in one of two categories. One category consists of the programs that visualize the interactions between different genes. The other category consists of the programs that visualize the sequence alignment graphs of similar genomes. Our product belongs to the second category. The first category of programs misses the crucial focus on the alignment of sequences to visualize mutations such as insertions, deletions, and variants.
The second category of programs are similar to the program we will create.

[3] Bandage by rrwick. (n.d.). Retrieved from https://rrwick.github.io/Bandage/
[4] Cytoscape. (n.d.). Retrieved from http://www.cytoscape.org/
[5] Gephi - The Open Graph Viz Tool. (n.d.). Retrieved from https://gephi.org/
[6] ProgrammingLife2016. (n.d.). Retrieved from https://github.com/ProgrammingLife2016

While these programs usually meet the basic requirements aligned in section 3, none of these programs have some of the the more advanced features that we suggest such as a tutorial, cloud synchronization of bookmarks. Additionally, most of these programs suffer from severe performance issues. We can learn from these issues and will focus on making a high-performance application by designing for performance from the start.

6 Timeframe and Budget

On a macroscopic scale, this project has a realization timeframe of 8 weeks, as set by the client. Since we are following the Scrum framework, we are also delivering working versions at the end of every week. This iterative improvement cycle allows us to synchronize as often as possible with the client on how we are doing in the timeframe.

There is no budget available to realize this product.

References

Bastian, M., Heymann, S., & Jacomy, M. (2009). Gephi: an open source software for exploring and manipulating networks. Retrieved from https://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154/1009

Cusumano, M. A. & Yoffie, D. B. (1999, October). What netscape learned from cross-platform software development. *Commun. ACM*, *42*(10), 72–78. doi:10.1145/317665.317678

Encyclopaedia Britannica. (2014, February). Point mutation. Retrieved from https://www.britannica.com/science/point-mutation

Ferre, X., Juristo, N., Windl, H., & Constantine, L. (2001, January). Usability basics for software developers. *IEEE Software*, *18*(1), 22–29. doi:10.1109/52.903160

Heer, J., Bostock, M., & Ogievetsky, V. (2010, May). A tour through the visualization zoo. *Queue*, *8*(5), 20:20–20:30. doi:10.1145/1794514.1805128

Loder, F. & Scott, M. (2015). 'humpty dumpty' program 'bandage' helps piece dna sequences back together again and wins 2015 iawards. Retrieved from http://www.bio21.unimelb.edu.au/humpty-dumpty-program-bandage-helps-piece-dna-sequences-back-together-again-and-wins-2015-iawards

National Institutes of Health (US). (2007). Understanding human genetic variation. Retrieved from https://www.ncbi.nlm.nih.gov/books/NBK20363

Wick, R., Schultz, M., Zobel, J., & Holt, K. (2015, October). Bandage: interactive visualization of de novo genome assemblies. *Bioinformatics*, *31*(20), 3350–3352.