Product Vision

Programming Life Group 2: Pantzerfaust

April 28, 2016

Contents

0 Introduction

This document describes the analysis that has been made regarding the problem description for the Programming Life context. This work is part of the Contextproject course (course code: TI2806) at the EEMCS faculty of the Delft University of Technology.

In section 1 the problem and product definition as provided by the client will be explained. In section 2 we will provide a requirements specification, based on section 1. In section 3 we will analyze the problems that previous attempts at this product encountered and we will describe our proposed solutions to mitigate these problems. Finally, we will conclude with a summary of the key elements in our product in section 5

1 Problem and Product Definition

- Application to enable Exploratory Data Analysis
- Support for Semantic zooming
- this means that the application understands the data and automates as many steps as possible (finding genomes, finding mutations, diff'ing sequences, finding interesting areas)
- Scalable (both back-end and interface), meaning still functions when data input increases (dramatically)

2 Requirements

2.1 Must haves

- The application must provide a scalable interface
- The application must enable the user to zoom in on semantic elements
- The application must be able to parse an exponential amount of DNA sequences without hindering the user.
- The application must be able to identify structures, such as mutations and genomes, in the DNA sequences.
- The application graphical interface must present the calculations and identifications made to the user

3 Problems in Previous Work and Proposed Solutions

3.1 Data Storage

API writes to a file / database in cold storage, so that the data doesn't need to be on the live stack.

3.2 Data access

When the data is stored, it must be accessed when the GUI needs it. An efficient data structure must be able to find all the required data in reasonable time.

Possible Solutions

Possibilities include Fingertables (as seen in Computernetwerken for searches through a swarm), and pre-loading certain parts of the data that will likely be needed soon.

3.3 Scalability Visualization

Mergen van subtrees voor 2-way of 3-way diff, zodat de interface niet voller raakt door meer data (GUI scalability)

Selecteren van de diff via de phylogenetic tree.

3.4 Semantic Visualization

Creating different zoom layers, each with their own first-class elements (ranging from single bases, to single nodes, to single mutations, etc.)

4 Final Product Description

After reviewing all problems and finding solutions for them, we can define the specifications of final product.

The final product will be an interactive application. At start, the highest zoom level will be presented, providing an overview of a few (max 10) big trends in the data. These trends can be identified by the first 10 exclusive subtrees in the phylogenetic tree, or 'manually' by comparing the mutations.

The application will allow various levels of semantic zooming, which can be achieved by clicking on the elements at the current level. An example is clicking on a trend in the highest view, zooming into that trend.

The application will (possibly apart from the first overview screen) always show two sequences. These sequences will not always be individual organisms, but can be the merged sequences of all organisms in a certain subtree.

This way the interface becomes scalable. Apart from not showing individual bases at a high zoom level, the application does not show individual organisms at a high level. When the amount of sequences becomes larger, the interface is not affected.

The application will parse the data in a streaming fashion. This means that it will write the parsed information to a persistent datastructure on disk. This way the parsing will scale in terms of memory management.

5 Conclusion

TODO