

Product plan

Byzantine Generals

CTGGTGGTGCTCAGCTGCAAGTCAAGCTGCTCTCTGGGCTGTGATCTCCCTGAGACC
CACAGCCTGGATAACAGGAGGACCTTGATGCTCCTGGCACAAATGAGCAGAATCTCT
CCTTCCTCCTGTCTGATGGACAGACATGACTTTGGATTTCCCCAGGAGGAGTTTGAT
GGCAACCAGTTCCAGAAGGCTCCAGCCATCTCTGTCCTCCATGAGCTGATCCAGCAG
ATCTTCAACCTCTTTACCACAAAAGATTCATCTGCTGCTTGGGATGAGGACCTCCTA
GACAAATTCTGCACCGAACTCTACCAGCAGCTGAATGACTTGGAAGCCTGTGTGATG
CAGGAGGAGAGGGTGGGAGAACTCCCCTGATGAATGCGGACTCCATCTTGGCTGTG
AAGAAATACTTCCGAAGAATCACTCTCTATCTGACAGAGAAGAAATACAGCCCTTGT
GCCTGGGAGGTTGTCAGAGCAGAAATCATGAGATCCTCTCTTTATCAACAACTTGC
AAGAAAGATTAAGGAGGAAGGAATAA, TGTGATCTCCCTGAGACCCACAGCCTGGA
TAACAGGAGGACCTTGATGCTCCTGGCACAAATGAGCAGAATCTCTCCTTCCTCCTG
TCTGATGGACAGACATGACTTTGGATTTCCCCAGGAGGAGTTTGATGGCAACCAGTT
CCAGAAGGCTCCAGCCATCTCTGTCCTCCATGAGCTGATCCAGCAGATCTTCAACCT

Preface

Ali Smesseim, 4386248, asmesseim

Samuel Sital, 4225139, ssital

Kamran Tadzhibov, 4280784, ktadzhibov

Ravi Autar, 4361172, raviatar

Adam el Khalki, 4348435, aelkhalki

Byzantine Generals

Delft, April 2016

Contents

- 1 Product 1**
 - 1.1 High-level product backlog 1
 - 1.2 Roadmap 1
- 2 Product backlog 3**
 - 2.1 User stories of features 3
 - 2.2 User stories of know-how acquisition 3
 - 2.3 initial release plan. 3
- 3 Definition of Done 5**
 - 3.1 Backlog Items 5
 - 3.2 Sprint 5
 - 3.3 Releases. 5
- 4 Glossary 6**

Product

1.1. High-level product backlog

The product has a couple of crucial attributes that will guarantee the success of the product. The product should

- enable us to explore the genome architecture of multiple strains exploring using a sequence graph.
- put this graph in the context of the evolutionary relationship between bacteria
- put bubbles (mutations) in the graph in the context of well-known reference genomes with their gene annotations and integrate with other reference databases
- provide semantic zooming to enable useful visual interpretation at various zoom levels from whole-genome to individual mutations
- have indications for convergent evolution of variants
- identify mutations and determine the type of variant (insertion, deletion, SNP) uniformly across the samples
- have visual encodings for different classes of mutations and the ability to filter on mutation class
- integrate with other resources, such as literature databases, mutation databases, to identify graph features that are interesting for further investigation.
- provide visual representation and encoding of meta-data associated with samples, such as drug resistance, location of isolation, isolation date, etc.

1.2. Roadmap

Design phase

- Setup tooling
- Figure out what client-server model to use and how data is sent from client to server and vice-versa
- Figure out how cytoscape works

Sprint 1

- Create formal API
- Display mini map
- Basic styling for the visualizer
- Determine positions of nodes in server

- Implement naive zooming
- Collapse bubbles based on zoom level in server

Sprint 2

- Database for saving graph
- Efficient collapsing algorithm
- Phylogenetic tree visualization

Sprint 3

- Add metadata to the segments
- Advanced styling
- Highlight path of different genomes

Sprint 4

- Implement visual encodings for different classes of mutations and the ability to filter on mutation class

Sprint 5

- implement put bubbles (mutations) in the graph in the context of well-known references genomes with their gene annotations and integrate with other reference databases

Sprint 6

- put bubbles (mutations) in the graph in the context of well-known references genomes with their gene annotations and integrate with other reference databases

Sprint 7

- Implement visual representation and encoding of meta-data associated with samples, such as drug resistance, location of isolation, isolation date, etc.

Sprint 8

- Implement indications for convergent evolution of variants

Sprint 9

- integrate with other resources, such as literature databases, mutation databases, to identify graph features that are interesting for further investigation.

2

Product backlog

2.1. User stories of features

As a user,
Who uses the product,
I can see the DNA sequence

As a user,
Who drags the mouse,
I can move the DNA sequence

As a user,
Who scrolls with the mouse,
I can zoom in and out

As a user,
Who uses the product,
I can upload genomes to visualize

As a user,
Who uses the product,
I can reload a DNA sequence from history

As a user,
Who uses the product,
I can move the DNA sequence

2.2. User stories of know-how acquisition

As a user,
Who uses the product,
It should be clear how the product is used

2.3. initial release plan

For our project we have the following milestones:

- Basic client-server implementation
- High efficiency for creating bubble graph
- Visualizing DNA sequence with different types of mutation.

- Visualizing Phylogenetic tree
- Database implementation

The minimal release features are as follows:

- The release must visualize a DNA sequence.
- The release must be tested.

3

Definition of Done

The last section of this report will explain in detail when the final product can be considered as such. In particular, this section will handle the Definition of Done of a feature, sprint and release.

3.1. Backlog Items

Within the backlog, an item is considered done if all the checklist points below are checked and considered to be true:

- Code complete and approved by the lead programmer.
- Code satisfies coding standards.
- Unit tests written and pass.
- Integration system test pass.
- The code is documented.

3.2. Sprint

A sprint is considered done, when all the checklist points below are checked and considered to be true:

- All sprint items considered done.
- Application is tested globally, all unit tests pass.
- Tests pass in the continuous integration system.
- User tests pass.

3.3. Releases

A release is considered done, when all the checklist points below are checked and considered to be true:

- The product should pass all unit tests.
- Integration system test pass.
- Interface looks as the product owner demanded.
- (End-)user tests pass.
- Code documented and satisfies coding standards.

4

Glossary