Final Thesis
A tool for statistical analyses of Genome Conformation Capture data

0796712
Mark Abrahams • Rotterdam University • March 27, 2012

Rotterdam University:
First supervisor: Dr. Bakker
Second supervisor: Ms. Van Brussel

Massey University:
First supervisor: Dr. O’Sullivan
Second supervisor: Dr. Gehlen
Student:
Name: Mark Abrahams
Address: Not. L.P. v/d Blinklaan 17 / 9 The Avenue 9d1
City: 3232 EK Brielle, The Netherlands / 0632 Albany, New Zealand
Phone number: +31652100266 / +64226821438 / Skype rockahh
E-mail: 0796712@student.hro.nl / mpabrahams@gmail.com

Organization
Name: Massey University
Address: Massey University Oteha Rohe Gate 4, Albany Highway
City: 0632 Albany, New Zealand

Organization first Supervisor
Name: J. O’Sullivan
Phone number: +649 4140800 extn:9811 / Skype jmosullivan
E-mail: J.M.OSullivan@massey.ac.nz

Organization second Supervisor
Name: L. Gehlen
Phone number: +649 4140800 extn:41110 / Skype taint_of_saidin
E-mail: massey@lutzgehlen.de

College
Name: Hogeschool Rotterdam / Rotterdam University
Institute: Communication, Media and Information technology
Address: G.J. de Jonghweg 4
City: 3015GG Rotterdam, The Netherlands

College 1st Supervisor
Name: R. Bakker
Phone number: +3110 794 4673 / Skype roel.bakker10
E-mail: r.bakker@hro.nl

College 2nd Supervisor
Name: A. Van Brussel
Phone number: +3110 794 5408
E-mail: brusa@hro.nl
# Table of Contents

**Preface**
- 4

**Summary**
- 5

**Introduction**
- 6

1. **Background**
   - 1.1 Genome
     - 8
   - 1.3 Genome Conformation Capture
     - 10
   - 1.4 High Throughput Sequencing
     - 11
   - 1.5 The analyses
     - 13

2. **Development methodology and application architecture**
   - 14
   - 2.1 Introduction
     - 14
   - 2.2 Project management
     - 14
     - 2.2.1 Waterfall model
       - 15
     - 2.2.2 Agile management
       - 15

3. **Storage**
   - 21
   - 3.1 Introduction
     - 21
   - 3.2 Problem analysis
     - 21
   - 3.3 Requirements
     - 22
   - 3.4 Solution
     - 22
     - 3.4.1 Database as storage solution
       - 22
     - 3.4.2 Relational vs non relational
       - 22
     - 3.4.3 Database models
       - 23
       - 3.4.3.1 BioSQL database
         - 24
     - 3.4.4 Connection to the storage
       - 28
       - 3.4.4.1 GCC API
         - 28
       - 3.4.4.2 BioPERL API
         - 29


3.5 Conclusion and Discussion 29

4. Web application 30

4.1 Introduction 30
4.2 Problem analysis 30
4.3 Requirements 31
4.3.1 Use cases 33
4.4 Solution 40
4.4.1 The overview page 40
4.4.4 The GCC analysis page 44
4.4.8 The GCC experiment page 51
4.4.10 The researcher page 52
4.5 Conclusion and Discussion 53

5. Conclusion 55

6. Discussion 57

6.1 Infrastructure 57
6.2 Methods 58
6.3 Graphical user interface 59
6.4 Other 59

References 60

Appendices 61

Glossary 62
Preface

This thesis is part of the internship of Mark Abrahams at Massey University, Albany, New Zealand.

The project involved developing a web application for statistical analyses on genome conformation capture (GCC) data stored in multiple MySQL databases.

This document is useful for those who are interested in the goals of the project, and the technical implementation of the IT-solution. Furthermore, this document forms the basis for assessing the internship.

I enjoyed working on this project, and the experience of working on the other side of the world. During this project I had help from my supervisors at Massey University and my supervisors from Rotterdam university. Therefore I would like to thank the following people gratefully:

Massey University
- Dr. Justin O’Sullivan of the Institute for Molecular BioSciences, who defined and supervised the project, gave his feedback on my reports and for giving me this awesome experience in New Zealand.
- Dr. Lutz Gehlen of the Institute for Molecular BioSciences, for his great support, his technical advice and guidance on the project, and his feedback on my reports.

Rotterdam University
- Dr Roel Bakker docent in BioInformatics at Rotterdam University, for being my first supervisor, his help with making me understand the biological aspects and giving me feedback on my reports. Even though there was a time difference between the Netherlands and New Zealand of 10 hours, he was quick in answering my questions by E-mail and was available every two weeks for a Skype meeting.
- Ms. van Brussel docent at Rotterdam University, for being my second supervisor and her feedback on my reports.
Summary

Cytological studies have demonstrated that the apparent sub-compartmentalization of the nucleus plays a role in coordinating the production of mRNA. These studies have identified distinct focal points within the nucleus where RNA is actively produced by RNA polymerases. These focal points, called transcription factories, are apparently associated with multiple genes. These studies have led to the proposal of 3D models for the regulation of gene expression in which similarly regulated genes loop out of repressive chromatin zones to enable the formation of transcription factories. The research group of Justin O’Sullivan is using Genome Conformation Capture (GCC) in combination with high throughput sequencing and analyses to increase their understanding of gene expression.

To store and analyze this huge amount of data, an IT solution is needed. The IT solution exists of two parts. A storage solution and a web application solution.

The storage solution was needed in order to obtain a complete set of data, that is easy to maintain, and exists of two MySQL databases. One for storing the output of GCC analyses, and one for storing the reference genome. With these databases can be communicated through API’s.

The web application solution was needed in order to perform statistical analyses and calculations on data that is stored in the new storage solution. This application exists of 10 dynamic HTML pages whom are created with the help of PERL scripts. Statistical calculations are done with a PERL file handler to R. 4 different analysis can be done. A correlation analysis to see how much related two GCC analyses are, a false detection rate analysis to control the expected proportion of incorrectly rejected null hypotheses, a regions analysis to find wether intergenic fragments rather interact with other intergenic fragments or with non-intergenic fragments, and a nearest gene analysis to find the distances between intergenic partners from intergenic fragments and their nearest genes.

This project is managed using the waterfall model and agile management. Also a development environment, a test environment and a production environment are set up to improve the quality of the web application.
Introduction

This project is part of the 3D Genome Lab project (http://3dgenome.co.nz/) that is lead by Dr. Justin O’Sullivan from Massey University - Institute for Molecular BioSciences. Their research centres on the use of high throughput techniques in the analysis of genome structures from both eukaryotes and prokaryotes. previous cytological studies have shown that the apparent sub-compartmentalization of the nucleus plays a role in coordinating the production of mRNA, the first step of gene expression. These studies have identified various focal points within the nucleus where RNA is actively produced by RNA polymerases. These focal points, called transcription factories, are apparently associated with multiple genes (Iborra, F., 1996). These discoveries have led to the proposal of 3D models for the regulation of gene expression in which similarly regulated genes loop out of chromosome territories to enable the formation of transcription factories (Figure 1). Such models begin to explain the non-random arrangement of genes within eukaryotic chromosomes and positional effects on transcription. (Becskei, A., 2005)

The research group of Justin O’Sullivan is using Genome Conformation Capture (GCC) in combination with high throughput sequencing and analyses to increase their understanding of gene expression, enabling them to begin to comprehend the mechanisms that coordinate this process over time. As such, it will have a significant impact on their understanding of the dynamics of biological systems, development and disease.

GCC and sequencing result in a large amount of data on which time consuming calculations are performed. Analyses that have to be performed are:

Mark Abrahams • email: 0796712@student.hro.nl • Hogeschool Rotterdam
• Calculate the statistically significant false positives (Type I error)
• Display graphically how strong two GCC analysis are related.
• Intergenic regions with their partners must be found
• Intergenic regions with their nearest gene must be found

These data are currently archived by hand, and analyzed with Microsoft Excel, which causes a number of problems:
• The growing amount of data causes data to easily become lost and unstructured in other data
• The data is not stored on a public place so if not working on your own desk, no analysis can be done
• Excel becomes unstable with the big amount of rows (±700000 rows)

The goal of the project is to design and implement a software solution which will solve these problems. Software requirements will be analyzed, literature will be consulted and solutions designed and implemented.

The structure of this document is as follows. Chapter 1 provides the biological background for this project. Chapter 2 describes the project management, architecture and environment decisions that were taken. Chapter 3 describes the storage IT solution, and chapter 4 the web application.

Each of these chapters contains the following sections: a specific introduction, requirements, followed by the basic approach of the solution, and the conclusion & discussion section.

The overall conclusion is drawn in chapter 5, and the overall discussion & outlook in chapter 6.
1. Background

1.1 Genome

The genome is the entirety of an organism’s hereditary information. This information is encoded in sequences of chemical compounds that lie along chromosomes in the cell of a living organism. These chemical compounds are called Deoxyribonucleic acid (DNA). Each chromosome contains a single molecule of DNA. Each strand of the DNA double helix is a linear arrangement of repeating similar units called nucleotides. A DNA nucleotide contains one of four different nitrogenous bases (Adenine, Thymine, Cytosine, and Guanine). The order of bases along a strand of DNA is what determines the genome sequence. The DNA of any individual living organism is often referred to as genotype. Every individual living organism has its own specific genotype.

Genes are chromosome pieces whose particular bases determine how, when and where the body makes each of the many thousands of different proteins required for life. These proteins describes the observable traits of an individual which the individual’s alleles (groups of genes) have expressed, and are often referred to as phenotype.

For example in a computer program, the genotype are the letters, numbers and symbols of the code for the program. With this code modules are created, which are the genes. From these modules, new instances are created. These instances are the proteins. These instances or proteins determine every observable trait of the program, which is the phenotype.

The process when a genotype is expressed is called transcription.
1.2 Transcription

Transcription begins after transcription factors attach to the promoter and assemble with the RNA polymerase enzyme at the start of a gene. These factors trigger the first phase of the process, reading off the information that is needed to make the protein. Once a particular part of DNA, called a transcription enhancer comes near the molecule that is created by the factors, it triggers the start of the transcription. The molecule then passes along the DNA reading the gene and unzipping the DNA to copy one side of the DNA string. This will give a new string called Messenger RNA (mRNA). mRNA is a copy of the genetic message. After the transcription process is complete it moves out of the nucleus and into the outer part of the cell. Then a ribosome is created around the mRNA string and translates the genetic information in the mRNA into a string of amino acids (phenotype) that will become a protein.

Depending on which genes are active and transcribed, different proteins are created, resulting in different cell types. (Figure 3)

The process of gene transcription affects the patterns of gene expression and, thereby, allows a cell to adapt to the changing environment, perform specialized roles within an organism and maintain basic metabolic processes necessary for survival.

Research has shown that similar genes loop out of repressive chromatin zones (chromosome territories) to enable the formation of transcription factories. (Mahy, N.L., 2002)
1.3 Genome Conformation Capture

The 3D genome lab is using a novel and generally applicable methodology called “Genome Conformation Capture” (GCC) to reveal the network of chromosome interactions for the yeast *Saccharomyces cerevisiae*. With this they hope to increase their understanding of gene expression.

Genome Conformation Capture or GCC is a high throughput molecular biology method for analyzing the organization of chromosomes in a cell’s natural state. With this method briefly, cells are cross-linked with formaldehyde. After that the chromatin is digested with the help of a restriction enzyme, ligated and purified to generate the interaction library.

![Diagram of Genome Conformation Capture process]

**Figure 3: Genome Conformation Capture.** The process of genome conformation capture.
Step 1 Cross linking:
By adding formaldehyde, DNA segments will cross link to proteins, and proteins will cross link with each other.

Step 2 Restriction Digest:
A restriction enzyme is added in excess to the cross-linked DNA, separating the non-cross-linked DNA from the cross-linked chromatin. The selection of the restriction enzyme in this step depends on the locus being analyzed and allows the separate analysis of different regulatory elements.

Step 3 Intramolecular Ligation:
Using very low concentrations of DNA favors the ligation of relevant DNA fragments with the corresponding junctions instead of the ligation of random fragments. There are two major types of ligation junctions that are over-represented. One is the junction that forms between neighboring DNA fragments due to incomplete digestion, which represents about 20-30% of all junctions. This number can be decreased by reducing the cross-linking stringency in the first step. The other type of junctions over-represented here is the junction that forms when one end of the fragment ligates with the other end of the same fragment, and contributes up to 30% of all junctions formed.

Step 4 Purification:
Now that the DNA is ligated, the cross link can be removed. This is what happens at the purification step. All non DNA and cross links are removed from the interaction library.

Step 5 Fragment:
Before the purified interaction library is sequenced, in step 5 the DNA is nebulised mechanically into smaller fragments that are ±200 base pairs long.

1.4 High Throughput Sequencing
High throughput sequencing includes several methods that are used for determining the order of the nucleotide bases in a molecule of DNA (adenine, guanine, cytosine, and thymine). The advent of high throughput sequencing has significantly accelerated biological research and discovery.

The fragments obtained from a GCC are not sequenced at IMBS, but sent to other research institutes or companies for sequencing. One very popular institute is BGI (http://
en.genomics.cn) which was founded in Beijing in 1999. Because the data is sequenced by different research institutes, a variety of different sequencing platforms are used.

- **The Illumina Genome Analyzer IIx** (*Figure 4*)
- **The Illumina HiSeq 2000** (*Figure 5*)
- **The Applied biosystems SOLiD 3** (*Figure 6*)

![Image of the Illumina Genome Analyzer IIx](http://www.illumina.com/, 2011)

![Image of the Illumina HiSeq 2000](http://www.illumina.com/, 2011)

*Figure 4: The Illumina Genome Analyzer IIx* (From http://www.illumina.com/, 2011)

*Figure 5: The Illumina HiSeq 2000* (From http://www.illumina.com/, 2011)

The Illumina Genome Analyzer IIx and the Illumina HiSeq 2000 both use the same technique since they are from the same company. Illumina developed a sequencing technology based on reversible dye-terminators. DNA molecules are first attached to primers on a slide and amplified so that local clonal colonies are formed (bridge amplification). Four types of ddNTPs are added, and non-incorporated nucleotides are washed away. The DNA can only be extended one nucleotide at a time. A camera takes images of the fluorescently labeled nucleotides, then the dye along with the terminal 3’ blocker is chemically removed from the DNA, allowing the next cycle.
The SOLiD 3 system uses a different technique developed by Applied Biosystems. This technique employs sequencing by ligation. Here, a pool of all possible oligonucleotides of a fixed length are labeled according to the sequenced position. Oligonucleotides are annealed and ligated; the preferential ligation by DNA ligase for matching sequences results in a signal informative of the nucleotide at that position. Before sequencing, the DNA is amplified by emulsion PCR. The resulting bead, each containing only copies of the same DNA molecule, are deposited on a glass slide. The result is sequences of quantities and lengths comparable to Illumina sequencing.

1.5 The analyses

The data retrieved from the research institute or organization who did the sequencing, is a fasta file containing the sequences. These sequences have to be mapped to the reference genome in order to identify the restriction fragments. This is done by a program that is programmed by a former intern (Fragment Mapper). The output is then a list of pairs of restriction fragments that were found to interact, stored in a tab delimited file. In the current situation these files are read in Microsoft Excel.
2. Development methodology and application architecture

2.1 Introduction

The previous chapter gave an introduction and some background information about the 3D genome lab project. This chapter will describe all decisions regarding development methodology and application architecture. In this chapter the following subjects will be described;

- 2.2 project management
- 2.3 staging
- 2.4 programming languages
- 2.5 architecture

2.2 Project management

Project management is the discipline of planning, organizing, securing and managing resources to achieve specific goals. The primary challenge of project management is to achieve all of the project goals and objectives while honoring constraints as scope, time and budget.

There are a number of different approaches to managing project activities. In this project I chose for a traditional approach in combination with an agile approach, because in my opinion these two approaches cover each others weaknesses.
2.2.1 Waterfall model

As traditional approach I used the waterfall model (http://en.wikipedia.org/wiki/Waterfall_model). This is a sequential design process, which is often used in software development processes. In this approach the progress is seen as flowing steadily downwards like a waterfall through the phases of conception, initiation, analysis, design, construction, testing, implementation and maintenance. With this model first all requirements are sorted out. After all requirements are collected, the design phase can start. Once the design phase is finished, the implementation can start. The verification phase is initiated after everything is implemented. This is the disadvantage of this approach in my opinion.

If an bug is found in the verification phase, it will consume much more time and effort to find and fix this bug. Because the bug can be in the whole program, in stead of just one part if the verification phase had been in an earlier stage.

Also there is no room for changes in the waterfall model, which I think is a big risk in a project for a research institute. Researchers are known for changing and/or adding ideas for the project daily, which will cause a lot of extra time and effort to fill these needs.

2.2.2 Agile management

In agile management, deliverables are submitted in stages. By submitting deliverables in stages, programmers are forced to test each part they deliver in stead of initiating the verification phase after the whole program is delivered. This creates a waterfall model where the verification phase is initiated together with the implementation phase.

However the agile management has some disadvantages too. Because parts can easily be added / adjusted, the risk of not finishing the total project in time increases. In order to manage this, I used partly the waterfall model and partly an agile management.

In figure 8 a flow diagram is shown of the current management that I used. First all requirements are collected. After this is discussed with the project leader (Justin O’Sullivan),
the solution is designed and separated into parts. After this is done the planning can be done. The implementation of the first part can be done after the planning is complete. Once the first part is finished, everything is verified and discussed with the project leader. If all is well, the next part can be developed. Etc.

2.3 staging

In most software projects, different environments are used to reduce the chance of incorrect results. Deploying the developed code directly to the production (final) environment, increases the risks of an incorrect implementation, and of incorrect code to damage valuable data. Risks of bringing the organization to a standstill are greatly reduced, when the developed code is first successfully deployed on one or more environment(s) prior to the production environment. In this project three different environments are used:

- A development environment (section 2.3.1)
- A test environment (section 2.3.2)
- A production environment (section 2.3.3)
2.3.1 Development environment

The development environment is the environment where the code is created. Here experimentation with new code and modules can be done. Also this environment can contain optional features, and unfinished code. The development environment is often referred to as "sandbox", because it is the playground of the developer.

<table>
<thead>
<tr>
<th>Linux Version</th>
<th>Ubuntu 10.04 LTS with LAMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perl Version</td>
<td>5.8</td>
</tr>
<tr>
<td>R Version</td>
<td>r-base version 2.10.1-2</td>
</tr>
<tr>
<td>External Modules</td>
<td>BioPerl 1.6.1, Bioperl-DB 1.6.0, Statistics::R 0.0.8, DBIx::Class</td>
</tr>
<tr>
<td>RAM</td>
<td>2 GB</td>
</tr>
<tr>
<td>CPU</td>
<td>1 Ghz</td>
</tr>
</tbody>
</table>

2.3.2 Test environment

If a software part is directly submitted from the development environment to the production environment, the inadequately tested software could have potentially disastrous results, bringing the organization to a standstill. Therefore a test environment is needed. After software is developed in the development environment, the software is first deployed in the test environment. This environment runs exactly the same software and versions as the production environment.

In this environment, the software is thoroughly tested with the help of test cases. If the software passes all tests, it is ready be deployed on the live environment. (Test results can be found in the software documentation.)

<table>
<thead>
<tr>
<th>Linux Version</th>
<th>Ubuntu 10.04 server LTS with LAMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perl Version</td>
<td>5.8</td>
</tr>
<tr>
<td>R Version</td>
<td>r-base version 2.10.1-2</td>
</tr>
<tr>
<td>External Modules</td>
<td>DBIx::Class</td>
</tr>
<tr>
<td>RAM</td>
<td>2 GB</td>
</tr>
<tr>
<td>CPU</td>
<td>1 Ghz</td>
</tr>
</tbody>
</table>
2.3.3 Production environment

Bugs are identified, fixed, redeployed on the test environment and the cycle continues. Once a stable build has been achieved, it is deployed on the production environment.

The production environment is the environment where the users will actually work on. Therefore it should have as little bugs as possible. A crash, or wrong code on this environment will cause possible disastrous results like corrupting data, or giving incorrect results.

<table>
<thead>
<tr>
<th>Linux Version</th>
<th>Ubuntu 10.04 server LTS with LAMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perl Version</td>
<td>5.8</td>
</tr>
<tr>
<td>R Version</td>
<td>r-base version 2.10.1-2</td>
</tr>
<tr>
<td>External Modules</td>
<td>DBIx::Class</td>
</tr>
<tr>
<td>RAM</td>
<td>8 GB</td>
</tr>
<tr>
<td>CPU</td>
<td>2.6 Ghz Quad Core</td>
</tr>
</tbody>
</table>

2.4 Programming languages

For the development of the IT solution programming languages are needed. These programming languages I chose with the following requirements in mind:

- The software solution must be easily accessible from different computers
- PERL is the preferred language of the 3D genome lab
2.4.1 PERL

PERL (Practical Extraction and Report Language) is a high-level, general purpose, interpreted, dynamic programming language which was originally developed as a general purpose unix scripting language to make report processing easier.

At the beginning of the project, Justin O’Sullivan and his research group asked if the software solution could be programmed in PERL. By using PERL the research group did not had to learn a new programming language and better help could be offered to me.

If the software solution would be programmed in PERL it would also fit more nicely with the infrastructure of the 3D genome lab.

As last I, as bioinformatician, would like to learn PERL since this is one of the most used languages in the bioinformatics field.

Now PERL was chosen as programming language for the back-end, I had to find a solution for the front-end (GUI).

2.4.2 HTML & CSS

Since the software solution must be easily accessible from different computers, a web application seemed like an obvious choice. With PERL it is possible to create dynamic HyperText Markup Language (HTML) pages, and in combination with cascading style sheets (CSS) this should be able to do the task.

HTML is the predominant markup language for web pages. CSS is a style sheet language used to describe the look and formatting of a document written in a markup language.
2.5 Architecture

The research server for this project is running Ubuntu 10.04 long time support (LTS). On the research server (called Savant) runs a “LAMP” Server. LAMP stands for Linux, Apache, MySQL, PERL.

- **Linux**
  Linux is a Unix-like open source computer operating system. This is in our case the Ubuntu 10.04 LTS.

- **Apache**
  Apache is a very popular open source web server system. This HTTP Server has a module architecture so that functionality can be extended easily. I used the module “mod_perl” in order to communicate with PERL through the “CGI” module of PERL.

- **MySQL**
  MySQL is an open source relational database management system (RDBMS), that runs as a server providing multi-user access to databases. I used the MySQL database to store the BioSQL database and the GCC database (Called “bioresearch”). To connect to the MySQL database from PERL I used the DBI module.

- **PERL**
  Perl is a high-level interpreted dynamic programming language. It originally was developed in 1987 as general purpose Unix scripting language to make report processing easier. Since then it has undergone many changes. Perl is a programming language that you see a lot in the Bioinformatics. PERL was used for programming the web application.

Figure 9: The architecture of the IT solution that is made during this project.
3. Storage

3.1 Introduction

This chapter will describe the storage part of the IT solution, starting with the analysis of the problems (section 3.2) prior to the development of the IT solution. These problems will result in purpose and requirements of the storage solution. One of the main requirements (section 3.3) states that the data generated by the Fragment Mapper is stored in a central place, which can be accessed by different tools whom need this information as input. To achieve these requirements a software solution is needed that consists of a storage space and an API to connect the tools to the storage space. The chapter will be finished with a conclusion.

Researchers of the 3D Genome Lab have to map every GCC analysis to the reference genome in order to identify the restriction fragments the sequences belong to. The output of these mappings are stored in tab delimited files on the local hard drive in order to use them as input for analysis. When two or more researchers are interested in the same GCC analysis, the data is often mapped and stored multiple times on different hard drives, by individual researchers. This leads to unnecessary used storage space and dual operations. Also when a researcher wants to share an analysis with a co-worker, this has to be done with flash drives or external hard drives which can get lost.

3.2 Problem analysis

One of the activities in the 3D genome lab (n.d.) is to perform analysis and calculations on data obtained from GCC analyses. This data are tab delimited files containing over 700 000 interactions, which are stored on the local hard drive. In practice a number of problems arise due to this storage method:

- When GCC data is retrieved from the research institute or organization, this needs to be stored on the computer running the Fragment Mapper. After the mapping is complete the text delimited file needs to be stored on a computer for analysis. When another researcher wants to use the same data, and is unaware of the work a co-worker already did, he commits unnecessary dual operations which cost unnecessary used space and time.

- When working on another computer the output file needs yet again be stored on another computer.
• Sharing a GCC analysis causes unnecessary used space.

• Due to the local storage, data is often stored inconsistent in the overall structure. This inconsistency causes data to easily get lost in the amount of other data.

• Flash drives and hard discs containing output data can easily be lost or stolen.

• Currently it is not possible to get a quick overview of the available data. This makes it harder to quickly summarize what kind of data is available, which costs time.

3.3 Requirements

The goal in this part of the project is to solve the problems listed in section 3.2. This will result in a storage solution which must be able to:

• Store data on a shared persistent destination

• Allow tools to retrieve data

• Allow tools to perform analysis on the data

Completion of the storage solution will primary result in saving time, a complete set of data and a new service, which can be used by tools that need GCC analysis data as input.

3.4 Solution

This section will provide a solution to tackle the problems described in the problem analysis (section 3.2).

3.4.1 Database as storage solution

Databases have been one of the most used methods for storing large amounts of data since the earliest days of electronic computing. They are suitable for storing organized data and are well performing in looking up data in huge amounts of data, which makes them very useful for this project.

3.4.2 Relational vs non relational

There are two different types of databases. Relational databases and non relation databases. Both of these databases have their advantages. Relational databases have more depth of functionality where non-relational databases are performing better.
Although non-relational databases often perform better, they have the disadvantages that they have less integrity, are unable to do joins or complex multi row transactions, and have no BioSQL database (biosql.org, 2011).

Also the structure of a relational database reflects the data it is to contain, rather than reflecting the structure of the application. This gives the advantage that the data, then, becomes somewhat application independent.

In my opinion, the relational database was best to use because:

- Data integrity is more important to this project than scalability
- With only one server, scalability is not an option
- A relational database is more application independent
- The BioSQL database (explained in section 3.4.3.1) is a relational database as well

### 3.4.3 Database models

In our opinion the best way to store the data was to create two databases: one BioSQL database and one GCC database. In the BioSQL database all sequence locations and features are stored. The GCC database holds all the output data of the Genome Conformation Capture experiments.

Mark Abrahams • email: 0796712@student.hro.nl • Hogeschool Rotterdam
3.4.3.1 BioSQL database

The BioSQL database is a MySQL (2011) database from which development has started in 2001 as a local relational store for GenBank. However over the years it has become a sufficiently generic schema for persistent storage of sequences, features and annotations. This database was useful for this project in several ways:

- By using this table we did not have to work on a design for storing sequences with their features and annotations.

- Because the BioSQL database is a mature and proven technology from which development has started in 2001, this would be a safer choice in comparison to creating a new design where we would have the risk of encountering problems that are already tackled in the BioSQL database.

- The BioSQL Database has a BioPERL API which contains useful features like methods for retrieving sequences, easily import and export fasta, genbank and swissprot files, translating nucleotide sequences and coding regions in mRNAs or cDNAs, etcetera.

![Figure 11: Database schema of the BioSQL Database. This schema shows the used tables from the BioSQL Database for this project.](image-url)
The Biodatabase table
In this table of the BioSQL database, every record represents a complete genome.

The Bioentry table
This is the core entity of the BioSQL schema. In the Bioentry table every row represents a chromosome which is associated with the Biodatabase table.

The Seqfeature table
Each record in the Seqfeature table is a section of a chromosome. On this section of the chromosome lies a feature (for example a gene).

The term table
The term table is a dictionary of ontology terms. In this table all the records hold “labels” that label a seqfeature’s name (for example: “gene” or “exon”).

The location table
This table holds all the locations of a particular “seqfeature” on a chromosome. Every record tells the start and stop of a seqfeature.

3.4.3.2 GCC database
The GCC database is a database from which the initial design was made by the research group of Justin O’Sullivan, for storing all the information obtained out of GCC analyses. After first use of the GCC database it became clear that it needed further modification. This database is fully normalized to isolate data so that additions, deletions and modifications of a field can be made in just one table and propagated through the rest of the database via the defined relationships. However, this might not be the best way since it significantly slows down the analyses. For example, if all the order indexes of the interacting fragments must be obtained from a GCC analysis in the normalized schema, this has to be looked up in three tables. This takes significantly more time as getting the same information from a table like figure 12.

<table>
<thead>
<tr>
<th>gcc_analysis_i d</th>
<th>fragment 1 order_index</th>
<th>fragment 2 order_index</th>
<th>Unique Strength</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>1</td>
<td>3</td>
<td>13</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>8</td>
<td>20</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Figure 12: Graphical example of the new created interactions table.
The fragment table

In the fragment table, every record is a section of a chromosome that is obtained by the GCC analysis. Every record has a chromosome name, a start and stop position and an order index. The order index is the unique id on the chromosome, the id is the unique id in the fragment table.

The gcc_experiment table

In this table, every record is a gcc experiment and holds the information that belongs to that particular gcc experiment. Information that can be found in this table is by which researcher the experiment is done, on which platform it is done the date of the experiment, which genome was used during this experiment. And optionally the start date, the number of cells,
wether the experiment had a paired end and the synchronization could be found here if the researcher has entered this information.

The gcc_analysis table
Once a GCC experiment is done, the data is sequenced. The gcc_analysis table, is the table that holds all the information about the parameters that are inserted at the GCC analyses. Every record holds a gcc_analysis id, a name and all the information about how the analysis has been done (which gcc experiments took place in the analysis, what the fragmentation_id is, with what analysis program it has been done, which alignment program is used, the minimal and maximal alignment length of the analysis and the allowed mismatches).
Optionally there could also be added a description about the analysis.

The interaction table
This table holds the information about the interaction between two fragments. For example if fragment 1 and 2 interact, this gives one row in this table. (Because this is one interaction.) An interaction belongs to a GCC analysis, and has two values: a total strength and a unique strength.
In a GCC analysis, two chromosomes get ligated to each other after they have been cross linked. After the purification step and the fragments have been cut into pieces, 36 basepairs are sequenced from each side of a fragment, and compared to the whole genome. When this particular sequence is found only once in the reference genome, it is certain that this particular fragment lays on this place in the reference genome, this is called the “unique strength”. However if the sequence is found on more places in the reference genome, it can not be guarantied on which place this fragment lays, therefore all the places are count and this number is the “total strength”. So the unique strength can always only be smaller or equal to the total strength.

The interaction_map table
In this table, every record is a fragment that is linked to a interaction. For example if fragment 1 and 2 interact, this will give two rows in the interaction_map table (Because one fragment is linked to one interaction, so fragment 1 is linked to interaction 1 and fragment 2 is linked to interaction 1). An interaction_map record has an id, an interaction_id and a fragment_id.

The platform table
In this table each row represents a platform. Each record has an id, a name and optionally a description.

The program table
In this table each record represents a program with their id, name and version number.
The *project table*
In this table all the projects are stored. A project has an id, a name and optionally a description.

The *project_experiment_map table*
The project experiment map table maps experiments to a project. This table has an id, a project id and either a transcription_id or a gcc_experiment_id.

The *project_researcher_map table*
This table maps a researcher to a project. Each record has an id, the id of the project and the id of the researcher.

The *researcher table*
This table represents all researchers. Here all the researchers their names and id’s can be found.

The *transcription_experiment table*
The last table is the transcription experiment table which represents all the transcription experiments. Here all the information with reference to the transcription experiment can be found (The id, name and the platform on which the experiment has been executed. And optionally the start date and the description of the project).

### 3.4.4 Connection to the storage

The connection to the storage is done by application programming interfaces (API’s). API’s are specification based source code that is intended to be used as an interface by software components to communicate with each other. In our case our API’s will be used for communication between the databases and the web application. In this section the API with their methods will be discussed.

#### 3.4.4.1 GCC API

The GCC application programming interface (API) is the API that handles the communication between the GCC database and the web application. The GCC API is partly automatic generated by the DBIx::Class::SchemaLoader module of PERL. This module automates the definition of a DBIx::Class::Schema (which creates database classes based on the schema) by scanning database table definitions and setting up the columns, primary keys, unique constraints and the relationships. Afterwards non existing relations can be added between tables.
3.4.4.2 BioPERL API

The BioPERL application programming interface (API) would have been the API that handles the communication between the BioSQL database and the web application, if not the installation documentation of the API was insufficiently documented.

Three researchers of Justin O’Sullivan’s research group (including me) have tried to get the BioPERL API to work, however with no success. BioPERL complained about missing modules though they were there, and gave bad error messages. After following different tutorials, and trying to find some clues in the BioPERL documentation, it was decided to abandon the BioPERL API at least until a working tutorial, or the documentation would be updated.

The BioPERL API does not only contain methods to connect to the BioSQL database. It can also be used for format conversion, report processing, data manipulation, sequence analyses, batch processing, etc.

This would give a lot of extra features, and might be something interesting to look for in the future.

3.5 Conclusion and Discussion

A data storage solution was needed in order to obtain a complete set of data, that is easy to maintain. This is done by creating two MySQL databases. One to store the output of GCC analyses (GCC database), and one to store the reference genome in (BioSQL database). With these two databases can be communicated through API’s, however the BioPERL API could not be installed correctly due to the lack of good documentation. Also the normalization in the GCC database caused analyses to become very time consuming.

With changes in the GCC database schema, the performance of the analyses increased tremendously.

This gives the discussion if the normalization is preferred over performance. In my opinion it is not. I think waiting hours for an analysis that can be finished in minutes because there is theoretically a slight chance that an interaction involves more than two fragments is not the best decision. The chance is to small to have such a big impact on the performance.
4. Web application

4.1 Introduction

To understand how the 3D spatial organization of genomes affects the transcription of genes, researchers of the 3D Genome Lab analyze the output of the GCC analyses for patterns. This is done by reading the tab delimited files holding the output data of the GCC analyses into Microsoft Excel. In Microsoft Excel different analyses can be executed on the data. When two or more researchers are interested in the same analyses, the analyses are often created and executed multiple times. Also Microsoft Excel was not able to handle such big datasets. According to “Microsoft Excel specification and limits” (http://office.microsoft.com/en-us/excel-help/excel-specifications-and-limits-HP005399291.aspx) the maximum amount of rows in Microsoft Excel 2003 is 65536 rows. However the results of the GCC analysis is ±700000 rows which causes Excel to respond very slow or causing force closes.

This chapter will describe how these problems are tackled, starting with the analysis of the problems (section 4.2) prior to the development of the IT solution. These problems will result in purpose and requirements of the web application solution. One of the main requirements states that the output data of the mappings can be displayed in a structured way with the option to run an analysis on different computers in the lab. To achieve these requirements a software solution is needed. This solution consists of a web application that connects to the storage space and is able to execute analyses. The chapter will be finished with a conclusion.

4.2 Problem analysis

One of the activities in the 3D genome lab is to perform analyses and calculations on data obtained from GCC analyses. This is done with Microsoft Excel. In practice a number of problems arise due to this method:

- Excel is not made for these big datasets which causes Excel to respond very slow or to terminate.

- Every time an analysis has to be executed, the researcher has to create it first.

Also after created a central storage for the sequenced GCC data, a web application for analyses on the data would be a more obvious choice rather than Microsoft Excel. An example is the webbioc project (http://bioconductor.org/packages/2.5/bioc/html/webbioc.html).
Webbioc is a web interface for doing microarray analysis and is intended to be deployed as a centralized bioinformatics resource for use by many users.

4.3 Requirements

The goal in this part of the project is to solve these listed problems. This will result in a web application solution. This web application solution must be able to:

Display:

- The web application should display the data in a clear way
- The web application should be able to display a list of researchers
- The web application should be able to display a list of projects
- The web application should be able to display the correlation in a graph
- The web application should be able to display all information about an GCC analysis
- The web application should be able to display all information about an GCC experiment
- The web application should be able to display all the projects a researcher works on
- The web application should be able to display the number of interactions that took place in a GCC analysis
- The web application should be able to display the number of interacting fragments from a particular GCC analysis
- The web application should be able to display the frequencies of distances between an intergenic fragment and their nearest gene graphically separated by intergenic and non-intergenic
- The web application should be able to display the frequencies of the amount of partners from intergenic partners graphically separated by intergenic and non-intergenic

Analysis:

- The web application should have an option to select two GCC analyses to correlate against each other by total strength or unique strength
- The web application should be able to calculate a false detection rate for a particular analysis
- The web application should be able to find intergenic fragments
• The web application should be able to find partners of intergenic fragments
• The web application should be able to find the nearest gene from a intergenic fragment
• The web application should be able to find partners of non-intergenic fragments
• The web application should be able to determine if there is a significant difference between intergenic fragments and non-intergenic fragments of a GCC analysis
• The web application should be able to determine the percentage of the genome that is intergenic
• The web application should be able to determine the percentage of the genome that is non-intergenic
• The web application should be able to determine what percentage of the partners of intergenic fragments is also intergenic
• The web application should be able to determine what percentage of the partners of intergenic fragments is non intergenic

Add:
• The web application should have an option to create a new project
• The web application should have an option to add researchers to a project
• The web application should have an option to add GCC experiments to a project
• The web application should have an option to add transcription experiments to a project

Remove:
• The web application should have an option to remove GCC experiments to a project
• The web application should have an option to remove transcription experiments to a project
• The web application should have an option to remove a project
• The web application should have an option to remove researchers from a project

Completion of the web application solution will primary result in a graphical user interface that gives a clear overview of the data, a set of analyses that can be executed on the data directly and a service that can be used on any computer that is able to run an internet browser.
4.3.1 Use cases

To map out the requirements, I used use case diagrams. These diagrams give a graphical representation of the functionalities of the web application.

Figure 14: UML Use case diagram of the overview page. On the overview page a scientist can create a new project, click on an existing project, view a list of all projects, view a list of all researchers or click on a researcher.
Figure 15: UML Use case diagram of the project page. On the project page a scientist can add or remove an experiment or researcher from the project, see a list with all the researchers that belong to the project, see a list of all the experiments that belong to the project or execute a strength correlation on two GCC analyses.
Figure 16: UML Use case diagram of the GCC analysis page. On the GCC analysis page a scientist can find all details about a GCC analysis, call the false detection rate analysis, regions analysis or the nearest gene analysis.
**Figure 17: UML use case diagram for the experiment page.** On the GCC experiment page a scientist can view all details about a GCC experiment, go to the page of the researcher that executed the experiment or go to the platform page of the platform that has been used for that particular GCC experiment.

**Figure 18: UML use case diagram for the platform page.** On the platform page a scientist can view all the details for the platform.
Figure 19: UML use case diagram for the researcher page. On this page a scientist can view all the projects a researcher is working on.

Figure 20: UML use case diagram for the transcription experiment page. On the transcription experiment page a scientist can view all details about a transcription experiment or go to the platform page of the platform that has been used for that particular
Figure 21: UML use case diagram for the correlation result page. On this page the results of the correlation analysis is shown containing a correlation graph, and the elapsed time of the analysis.

Figure 22: UML use case diagram for the false detection rate (FDR) result page. On this page the results of the FDR analysis are shown containing the amount of false positives at a significance rate of 0.05, the number of interactions and the number of fragments, and the elapsed time of the analysis.
Figure 23: UML use case diagram for the regions analysis result page. On this page the results of the regions analysis are shown containing two graphs. One for the frequency of intergenic partners of intergenic fragments and one for the amount of non intergenic partners. Underneath the graph a table is shown with the “Z-score” together with the critical values, the percentage of the genome that is intergenic, the percentage of the genome that is non intergenic, the percentage of the intergenic fragment partners that is intergenic and the percentage of the intergenic fragment partners that is non intergenic.
4.4 Solution

This section describes the solution that we came up with to tackle the problems described in the problem analysis (4.2 section).

4.4.1 The overview page

The overview page is the page that is displayed when the web application is opened. This page gives a clear overview of all the projects and all the researchers from the 3D genome lab. Also a new project can be created from this page.
4.4.2 The project page

After there is clicked on a particular project, the id of that relevant project is send to the apache web server who gives it through to PERL. PERL then creates a HTML page for that relevant project.

On the project page all related GCC analyses are shown with their GCC experiments. Also the researchers that are working on the project are represented in a list. On this page a researcher or GCC experiment can be added or removed from the relevant project. At the bottom of the page is an analysis section from where the correlation analysis (explained in section 4.4.3) can be called.
4.4.3 The correlation analysis

Researchers of the 3D genome lab analyze the output of the GCC analyses for patterns. One analysis that is used is the strength correlation analysis. This analysis shows how strong two GCC analyses are related regarding the interactions. In the first step of this process for each GCC analysis a list is created holding all the interacting fragments with their interaction strength. This strength can either be the total strength or the unique strength.

These lists are then compared to each other with help of a smart algorithm, and the strengths of the interactions are plotted. All the strengths of the interactions from GCC analysis 1 are the x-axis coordinates, and the strengths of interactions from GCC analysis 2 are the y-axis coordinates.

For example (The lists from figure 27 are used.) interaction 1 from GCC analysis 1 is fragment 1 that interacts with fragment 4 with a strength of 5. This interaction is not in GCC analysis 2, and therefore the strength is 0. This gives the first plotpoint (5,0). Interaction 2 is fragment 1 that interacts with fragment 8 with a strength of 3. This interaction is in both analysis with also both the same strength. This creates the plotpoint (3,3).

The more two analyses are related, the closer all plotpoints are near the 45° line (see figure 28). From the plotpoints on this 45° line researchers can conclude that these fragments interacted with the same strength in both analyses.

<table>
<thead>
<tr>
<th>Interaction</th>
<th>Order index fragment 1</th>
<th>Order index fragment 2</th>
<th>Strength</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1</td>
<td>6</td>
</tr>
</tbody>
</table>

**GCC analysis 1:**

<table>
<thead>
<tr>
<th>Interaction</th>
<th>Order index fragment 1</th>
<th>Order index fragment 2</th>
<th>Strength</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>

**GCC analysis 2:**

Figure 27: Graphical example of the created lists for the strength correlation analysis.
Figure 28: The outcome of a strength correlation analysis where two GCC analyses are exactly the same. All plotpoints are exactly on the 45° line which means that the two GCC analyses are exactly the same / very much related.
4.4.4 The GCC analysis page

The GCC analysis page exists of two parts. The details part that displays all details about the selected GCC analysis, and an analysis part where the false detection rate analysis (explained in section 4.4.5), the regions analysis (explained in section 4.4.6) or the nearest gene analysis (explained in section 4.4.7) can be executed on the selected GCC analysis.

4.4.5 The false detection rate analysis

The use of multiple hypothesis testing calls for a statistical method to correct the multiple comparisons. This is done with the false detection rate (false discovery rate) analysis. This analysis controls the expected proportion of incorrectly rejected null hypotheses, also known as type I errors.

In practical terms the false detection rate is the expected proportion of false positives among all significant hypotheses. Researchers at the 3D genome lab, want to know the amount of false positives at a cumulative probability of 0.05. This can be calculated with the binomial distribution functions in R.

This can be illustrated with the following example: Discover the chance of throwing a six in six trails. In figure 29 the cumulative probability distribution function is drawn.

![Cumulative distribution function](image)

**Figure 29: The cumulative probability distribution for a normal dice (six sides).**
With the cumulative probability distribution function `pbinom()` in R, the cumulative probability can be calculated for a particular probability. If for example a researcher would like to know the cumulative probability of throwing at least three or less times a six, this can be calculated with `pbinom(3,6,1/6)`. This gives a cumulative probability of 0.991298.

However for the false detection rate analysis of the 3D genome lab, the researchers would like to know how many number of successes there at a cumulative probability of 0.05. This can be calculated with the inverse cumulative probability distribution function.

For the example, if a researcher would like to know the number of successes there are at a cumulative probability of 0.9, this can be calculated with `qbinom(0.9,6,1/6)`. This gives a number of successes of 2. Which means that there is a chance of 90% that 2 or less times of the six throws will be a six.

As can be seen in the example, the `qbinom` function needs three input variables:

- **p**
  
  `p` is the vector of probabilities, which is a percentage that is setting the critical region. The researchers of the 3D genome lab are using `p=0.05` as percentage.

- **size**
  
  `size` indicates the number of trials, which is the number of fragments in the GCC analysis, the false detection rate analysis is called upon.

- **prob**
  
  `prob` is the probability of success on each trial. This is in practical terms the chance of two fragments ligate with each other.

For this analysis all fragments, and all interactions of a particular analysis are queried out of the database. This amount of fragments is used for the size variable of the `qbinom()` function. After that the probability of success on each trial is be calculated. This is done with the following formula: `(1/total amount of interacting fragments)^2`.

This gives the formula `qbinom(0.05, number of fragments, (1/number of interactions)^2)`.

The result of this analysis will give three values to the researcher. (See figure 30.)

- **k**
  
  This value is the expected proportion of false positives among all significant hypotheses for the GCC analysis, the false detection rate analysis was called upon.
• \( n \)

\( n \) is the number of interactions that occurred in the GCC analysis, the false detection rate analysis was called upon

• \( f \)

\( f \) is the number of interacting fragments that are in the GCC analysis, the false detection rate analysis was called upon

---

The way the probability of success for each trail is calculated is, in my opinion, doubtful. As I am no statistician nor biological expert, I would recommend to have a statistician who understands the biological aspect of the experiment, to have a look at this analysis.

---

4.4.6 The regions analysis

Another hypotheses the researchers from the 3D genome lab are interested in, is whether intergenic fragments rather interact with other intergenic fragments than non-intergenic fragments.

This section will explain the regions analysis. With help of this analysis, difference in behavior between intergenic and non intergenic fragments can be proven.

Intergenic fragments are fragments, located between clusters of genes that contain no genes. (See figure 31.)
The result of this analysis exists of several output:

- a graph that displays how often intergenic fragments has n amount of intergenic partners
- a graph that displays how often intergenic fragments has n amount of non-intergenic partners
- a Z-value
- a minimal critical value
- a maximal critical value
- the percentage of the genome that is intergenic
- the percentage of the genome that is non-intergenic
- the percentage of intergenic partners from intergenic fragments
- the percentage of non-intergenic partners from intergenic fragments

In order to obtain these results, input from a GCC analysis is needed, and calculations need to be done. First all interacting fragments from the GCC analysis are checked whether they are intergenic or not. This is done with a smart designed algorithm. This algorithm creates a digital representation of the chromosomes with all the fragments and genes located on their position. With the help of a variable that is keeping track if the currently checked variable does not lie within a gene, the start till end of the chromosome is checked for start and stop locations of fragments. If the start of a fragment (numbers 1 in figure 32) and the end of a fragment (numbers 2 in figure 32) is found not to be in a gene, and a gene is not to be found in a fragment, this fragment is found to be intergenic (fragment 2 in figure 32).

**Figure 31: Intergenic and non-intergenic fragments.** In this figure fragment 2 is an intergenic fragment, fragments 1, 3 and 4 are non-intergenic fragments since they contain (parts of) a gene.
After all interacting fragments are known to be intergenic or not, the partners of all intergenic fragments are found. From this information two graphs are created displaying how often a intergenic fragment has n amount of intergenic partners and how often a intergenic fragment has n amount of non intergenic partners. Together with this analysis a Z-test is done, who calculates the minimal and maximal critical value, together with the Z-value. This will tell the researchers from the 3D genome lab if there is a significant difference between intergenic fragments and non intergenic fragments on a significance level of 0.05.

The Z-value is calculated with the following formula in R:

\[
\frac{\bar{P} - P_0}{\sqrt{P_0 (1 - P_0) / n}}
\]

(sample proportion - hypothesized value) / \sqrt{hypothesized value * (1-hypothesized value) / size of sample)

- Sample proportion = amount of intergenic partners / total amount of partners
- hypothesized value = percentage of intergenic fragments in the genome
- size of sample = total amount of partners

The minimal and maximal critical value are calculated with the following formula in R:

\[
\text{significance.level} = 0.05 \\
\text{half.significance.level} = \text{qnorm(t-significance.level/2)} \\
\text{minimal critical value} = -\text{half.significance.level} \\
\text{maximal critical value} = \text{half.significance.level}
\]

Where qnorm() is the quantile function for the normal distribution.

Figure 32: visual representation of the algorithm for checking if a fragment is intergenic. The numbers 1 are the start of a fragment, the numbers 2 are the end of a fragment, the numbers 3 are the start of a gene, the numbers 4 are the end of a gene.
4.4.7 The nearest gene analysis

Another intergenic regions analysis is the nearest gene analysis. Earlier research shows that occasionally some intergenic DNA acts to control genes nearby. Therefore researchers from the 3D genome lab would like to know if there is a difference between the distances of intergenic partners from intergenic fragments and their nearest gene, and the distances of non-intergenic partners from intergenic fragments and their nearest gene. This is done with an analysis very similar to the regions analysis. The difference between these analyses are the graphs. In the nearest gene analysis distances between intergenic partners from intergenic fragments and their nearest genes are plotted against their frequencies, and distances between non-intergenic partners from intergenic fragments and their nearest genes are plotted against their frequencies. Also in this analysis a Z-test (section 4.4.6) is performed.

To find the nearest gene of the partners from intergenic fragments, two lists are created. A list with the positions of middle points of the intergenic fragments, and a list with the start positions of the genes. Every fragment is then compared with their chromosome name and position to the genes list. With the help of a variable that keeps track of what part of the list is...
already done, this is an efficient way to find the distances between the fragments and their nearest genes.

Figure 34: visual representation of the algorithm for finding the nearest gene. The arrows point to the places that are checked for the nearest gene. The arrow with the continuous line is the place that is found to be the nearest gene.

Figure 35: The result page of the nearest gene analysis.
4.4.8 The GCC experiment page

The GCC experiment page gives the researchers at the 3D genome lab the option to view all details about a particular GCC experiment.

![Experiment Details](image)

Figure 36: The experiment page.

4.4.9 The platform page

On the platform page, researchers from the 3D genome lab can find all details about the method that was used for sequencing the GCC experiment.

![Platform Details](image)

Figure 37: The platform page.
4.4.10 The researcher page

The researcher page is useful in several ways. First a researcher can click on his own name. This will give a personal page with all the projects he is working on. Secondly another researcher can click on the researcher ID to see who did the experiment. This is helpful if there are any questions about the experiment or a project.

Figure 38: the researcher page.
4.5 Conclusion and Discussion

A web application solution was needed in order to perform statistical analyses and calculations on data that is stored in the new storage solution. This is done by creating dynamic HTML pages with the help of PERL scripts. Statistical calculations are done with a PERL file handler to R. There are also some R modules available for PERL, however these were not found stable enough for all the tasks.

The web application exists of 10 dynamic web pages. The overview page is the home page of the web application. Here new projects can be created, or researchers can go to their researcher page, containing all their projects. The project page will display all analyses that belong to the project, all researchers that are working on the project, and an analysis section where a correlation analysis can be executed on two analyses from the project. Clicking on a GCC analysis or experiment will display all their details. Together with the GCC analysis details, an analysis section is shown, where the false detection rate analysis, regions analysis or nearest gene analysis can be executed.

This web application is able to do 4 different statistical analyses. A correlation analysis, a false detection rate analysis, a region analysis and a nearest gene analysis.

The correlation analysis is done by creating a graph with the strengths of the interactions from GCC analysis 1 on the x-axis and the strengths of the interactions from GCC analysis 2 on the y-axis. If an analysis only appeared in one GCC analysis, the other GCC analysis has a 0 as strength. The closer the plotted points are on the 45° line, the more related the two analyses are. Researchers can conclude that fragments interacted with the same strength in both analyses, if a plotpoint is on the 45° line.

The false detection rate analysis, gives the the expected proportion of false positives among all significant hypotheses. This is done by using the inverse of the binomial distribution function (quantile function) in R. The result for this analysis gives value k, the minimum false detection rate at which the test may be called significant. Value n, which is the number of interactions that occurred in the GCC analysis. And value f, which tells the researchers how many interacting fragments are in the GCC analysis. The way the probability of success for each trail is calculated is, in my opinion, doubtful. As I am no statistician nor biological expect, I would recommend to have a statistician who understands the biological aspect of the experiment, to have a look at this analysis.

The regions analysis is one of the more complex analyses in the web application. With this analysis researchers want to know if intergenic fragments preferably interact with other intergenic fragments. The result of this analysis are two graphs displaying how often a
intergenic fragment has n amount of intergenic fragments and how often a intergenic fragment has n amount of non intergenic fragments, a Z-test which tells the researchers from the 3D genome lab if there is a significant difference between intergenic fragments and non intergenic fragments on a significance level of 0.05.

The nearest gene analysis is very much similar to the regions analysis, only here researchers are interested in the distances between partners of intergenic fragments and their nearest gene. This is because earlier research shows that occasionally some intergenic DNA acts to control genes nearby. The result of the analysis is the same as the regions analysis only this graph shows the distances between intergenic partners from intergenic fragments with their nearest genes plotted against their frequencies, and distances between non-intergenic partners from intergenic fragments and their nearest genes, plotted against their frequencies.

Smart algorithms made the analyses more time efficient. However due to the normalization in the storage, the speed of the analyses is still not optimal.
5. Conclusion

The current situation where all the data obtained from GCC analysis are put in excel sheets to do statistics on the data, is definitely not fit for purpose. Excel is not able to handle the huge amounts of datasets, and data was all across different computers. In my opinion, the right way to go, was creating an IT solution that existed out of two parts. A data storage solution and a web application solution.

The data storage solution was needed in order to obtain a complete set of data, that is easy to maintain. Here-fore two MySQL databases were created. One for storing the output of GCC analyses (GCC database), and one for storing the reference genome (BioSQL database). With these two databases can be communicated through API’s, however the BioPERL API could not be installed correctly due to the lack of good documentation. Also the normalization in the GCC database caused analyses to become very time consuming. By changing the GCC database schema, performance of analyses would increase.

The web application solution was needed in order to perform statistical analyses and calculations on data that is stored in the new storage solution. This web application exists of dynamic HTML pages whom are created with the help of PERL scripts. Statistical calculations are done with a PERL file handler to R, since the PERL modules for R are not found stable enough. The web application exists out of 10 dynamic web pages. In this web application projects can be managed, and 4 different analyses can be done on the GCC data. The correlation analysis will tell a researcher how much related two GCC analyses are, the false detection rate analysis is there to control the expected proportion of incorrectly rejected null hypotheses. Two other analyses involve intergenic fragments. The regions analysis will tell a researcher if the intergenic fragments from a GCC analysis prefer to interact with other intergenic fragments from that particular GCC analysis, and the nearest gene analysis will look for distances between intergenic partners from intergenic fragments and the start of their nearest gene.

In order to achieve all of the project goals and objectives while honoring constraints as scope, time and budget, two different types of project management are applied. The waterfall model because getting requirements and design first improves quality, and the agile management to take into account that researchers are known for changing and/or adding ideas for the project daily. To obtain an even higher level of quality, the developed code is deployed on multiple environments. First code is created in the development environment, after the code looks...
ready, it is deployed on the test environment. If this gives no problems the code is ready to be deployed on the production environment.

Creating the storage - and the web application solution following the project management methods, brought this project to a success.
6. Discussion

6.1 Infrastructure

Ultimately when methods will be used by more web applications, creating an extra “higher level api” that would hold all the general functions, so these functions can be easily adopted by other web applications might be something to bear in mind. Also the BioPERL API might be something to take another look at once the documentation is updated, or a new tutorial becomes available.

Figure 39: Schema of the infrastructure containing the “higher level API”.

Mark Abrahams • email: 0796712@student.hro.nl • Hogeschool Rotterdam
6.2 Methods

In the future, the tool could be used on genomes other than that of the Saccharomyces cerevisiae (yeast). This will demand for changes in order to maintain speed and memory. An option would be to parallelize per chromosome and make use of the R parallel project. Other projects that could help maintain big data are R Hadoop or Revolution analytics.

In my opinion it is necessary to have a statistical engineer on the project, if new analyses are created. This will deal with problems I encountered with the False detection rate analysis.

Further new statistical analyses could be added the web application, or current analyses could be expanded for example interactions could be sorted into:

*adjacent interaction:*

![Figure 40: Visual representation of an adjacent interaction.](image)

*non-adjacent interaction:*

![Figure 41: Visual representation of a non-adjacent interaction.](image)
6.3 Graphical user interface

In the graphical user interface, the browsing could be improved for the user. For example

- a drop down lists could be added to the experiment page so that researchers could switch quicker between experiments.
- a progress bar could be shown when an analysis is being executed.
- a researchers name could “pop up” while hovering over a researcher id.

Also the appearance of the pages can be improved by a designer.

6.4 Other

More tests could be created for assuring the web application is working as it should. The more tests there are, the more certain can be that the web application is 100% functioning as it should.

Also at the moment the web application does not have multi user compatibility. When in the future the web application is more used, this might be an important feature to implement.
References


Appendices

- Install manual
- Software documentation
Glossary

Genome:
The genome is the entirety of an organism’s hereditary information.

Chromosome:
A chromosome is an organized structure of DNA and protein that is found in cells.

DNA:
DNA is a nucleic acid that contains the genetic instructions used in the development and functioning of all known living organisms.

mRNA:
mRNA is a molecule of RNA that encodes a chemical “blueprint” for a protein product.

Nucleotides:
Nucleotides are molecules that, when joined together, make up the structural units of RNA and DNA.

Eukaryotes:
The Eukaryotes are a group of organisms whose cells contain complex structures enclosed within membranes.

Prokaryotes:
The prokaryotes are a group of organisms that lack a cell nucleus, or any other membrane-bound organelles.

Genotypes:
The genotype is the genetic makeup of a cell, an organism, or an individual.

Phenotypes:
A phenotype is an organism’s observable characteristics or traits.