# Numbered Heading 1

Every new chapter should start on an odd page.

## numbered Heading 2

### numbered Heading 3

#### numbered Heading 4

Body text in font Garamond 12pt

<https://www.duo.uio.no/bitstream/handle/10852/37680/Bragstad.pdf?sequence=5>

<https://www.duo.uio.no/bitstream/handle/10852/37431/Furenes_Master.pdf?sequence=1>

Tom side

Title for master thesis

Sabba Ifzal

May 1, 2014

Tom side

Acknowledgements

Takke folk som har hjulpet til

Veiledere – Torbjørn, Lex, Geir Kjetil og Ksenia

Andre folk i 10 – Sveinung, Kai, Jon

Ruth for sensur

Ghada for samtaler

etc

Abstract

Kort oppsummert hele oppgaven

Why you do this

How you do this

The results

Conclusion

Preface

Hvem er oppgaven ment for? Andre bioinformatikkstudenter

Contents

[Chapter 1 Numbered Heading 1 1](#_Toc377385015)

[1.1 numbered Heading 2 1](#_Toc377385016)

[1.1.1 numbered Heading 3 1](#_Toc377385017)

[Acknowledgements 5](#_Toc377385018)

[Abstract 6](#_Toc377385019)

[Preface 8](#_Toc377385020)

[Contents 10](#_Toc377385021)

[List of figures 12](#_Toc377385022)

[List of tables 13](#_Toc377385023)

[Chapter 2 Introduction 14](#_Toc377385024)

[2.1 Motivation 14](#_Toc377385025)

[2.2 Hypothesis? 14](#_Toc377385026)

[Chapter 3 Background 15](#_Toc377385027)

[3.1 DNA and RNA 15](#_Toc377385028)

[3.2 Sequencing 15](#_Toc377385029)

[3.2.1 De novo sequencing 15](#_Toc377385030)

[3.2.2 Next generation sequencing 15](#_Toc377385031)

[3.3 Genome Assembly 15](#_Toc377385032)

[3.3.1 Quality measures 15](#_Toc377385033)

[3.3.2 Quast 15](#_Toc377385034)

[3.4 Bioinformatics 15](#_Toc377385035)

[3.4.1 Benchmarking 16](#_Toc377385036)

[3.5 Problem areas 16](#_Toc377385037)

[3.5.1 Assembly 16](#_Toc377385038)

[3.5.2 GAGE-B 16](#_Toc377385039)

[3.5.3 Galaxy framework 16](#_Toc377385040)

[3.6 Technology and formats 16](#_Toc377385041)

[3.6.1 Python 16](#_Toc377385042)

[3.6.2 Bash 16](#_Toc377385043)

[3.6.3 Galaxy 16](#_Toc377385044)

[3.6.4 All assembler 16](#_Toc377385045)

[3.6.5 Fasta and fastq 16](#_Toc377385046)

[3.7 Evaluation methods 16](#_Toc377385047)

[Chapter 4 Material 17](#_Toc377385048)

[4.1 Dataset 17](#_Toc377385049)

[4.2 Reference genome 17](#_Toc377385050)

[4.3 Assemblers? 17](#_Toc377385051)

[Chapter 5 Methods 18](#_Toc377385052)

[5.1 Description 18](#_Toc377385053)

[5.2 Implementation 18](#_Toc377385054)

[Chapter 6 Results 19](#_Toc377385055)

[Chapter 7 Discussion 20](#_Toc377385056)

[7.1 Interpreting the results 20](#_Toc377385057)

[7.2 Analysis of galaxy tool 20](#_Toc377385058)

[7.2.1 Performance 20](#_Toc377385059)

[7.2.2 Potential use 20](#_Toc377385060)

[7.2.3 Weakness 20](#_Toc377385061)

[7.3 Further work 20](#_Toc377385062)

[7.4 Conclusion 20](#_Toc377385063)

[Appendix a (Plot/Figure) 21](#_Toc377385064)

[Appendix B (Tables) 22](#_Toc377385065)

List of figures

[figure 1 21](#_Toc377132394)

[Table 7-1 Bla bla bla 21](#_Toc377132395)

List of tables

[figure 1 21](#_Toc377132379)

[Table 7-1 Bla bla bla 21](#_Toc377132380)

# Introduction

## Background

Why decide to have this project…

Why should the project be done, why use time

## PROBLEM STATEMENT / AIMs

What you actually doing in the project.

Visualize quastresult and add information quast-like, that can be useful for determining which assembly is the best.

Same principle can be added.

Organize a automatized principle. Make something to compare.

## Literature review

Quast, gage assemblathon

Hvordan andre har funnet den beste løsningen

## Problem solution

Briefly about the solution. How you solved it.

Benefit of system. Quast need install etc, but not this.output structure is rather good so we reuse code for our output.

# Background

Since the thesis will involve discussion of problem areas in bioinformatics that require some biological knowledge, this chapter will provide the basics of genome assembly, benchmarking in bioinformatics and the Galaxy Project.

## Genome Assembly

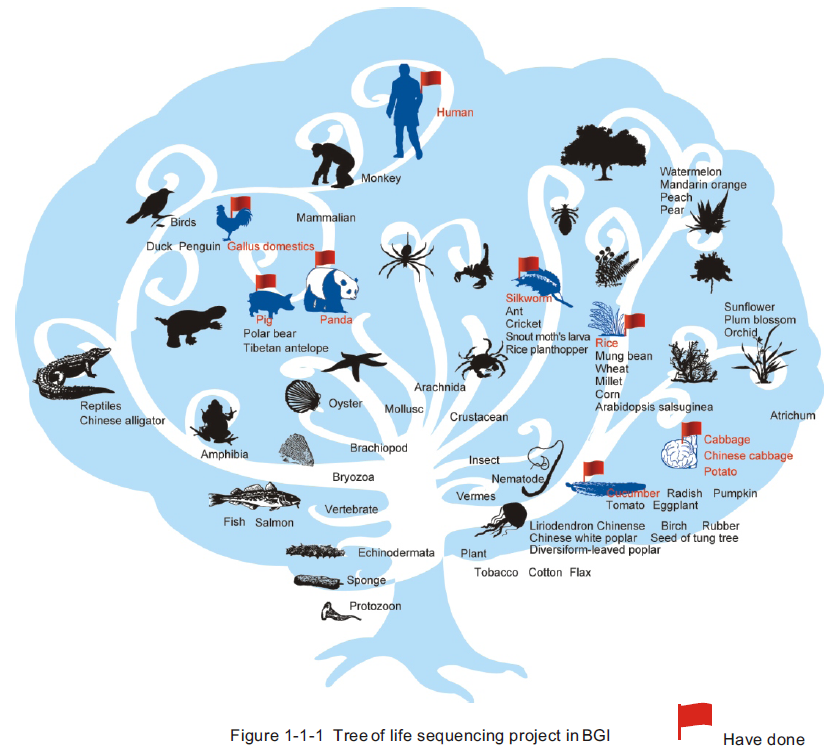
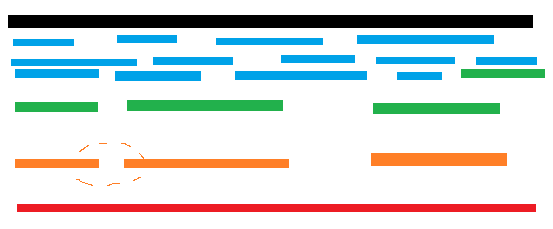


Figure ‑ – The tree of life sequencing project in beijing Genomics Institute  
Source - <http://www.hitseq.com/images/BGI-seq-denovo.png>

Genome assembly is the process of trying to reconstruct a sequence from smaller sequences called reads. This process could be trivial if each read had a unique placement, but it doesn’t, thus making assembly quite a difficult process.

### Sequencing

When looking at sequencing in a biological context, it is usually referred to as a process (a method or technology) that is used to determine the order of nucleotides in a DNA molecule. This process can be used to determine the order of genes, full chromosomes or entire genomes. This is important because the sequence in which the nucleotides appears in gives scientists valuable information about that part of the DNA which can, for instance, be used to look for disease-causing mutations in genes.



Sequencing can, I grove trek, be described as a process where some target DNA is shredded into reads that with a minimum of X coverage are used to make contigs, which are then used to make scaffolds. The goal is to end up with one continuous sequence equal to the target DNA. The new sequence can then be mapped back to a reference, if one exists, to check the correctness of the assembly to a certain degree.

Figure X gives a visual impression of sequencing as described I grove trek above.

The two major sequencing technologies, being de novo sequencing and next-generation sequencing, are briefly described in the next two sections.

#### De novo sequencing

De novo sequencing (from latin as “from the beginning”, “afresh” or “anew”) is a collective term used for methods that sequence a DNA sequence for the first time (meaning that it has no previously known sequence), known genomes where significant structural variation is expected, or for microbial sequencing that includes experimental strains and genomes with high plasticity. This kind of sequencing use starting materials such as bacterial, viral, phage, fungus, eukaryote genomic DNA, fragmented DNA or BACs and fosmids. It can use human genome, but substantial challenges exist for its application to this.

**Sanger and shotgun sequencing**

Shotgun sequencing was introduced by Sanger and other colleagues in 1977 and was the first mainstream sequencing technology that remained the leading genome sequencing method for nearly 25 years. The name came from the analogy of a rapidly expanding, random firing pattern of a shotgun. It obtains random sequence reads from a genome and assembles them into contigs on the basis of sequence overlap. It is straightforward for simple genomes with few repeat sequences and more challenging for complex genomes because of false overlap in the repeated regions. Even though it is widely used for smaller-scale projects and for obtaining especially long contiguous DNA sequence reads (>500 nucleotides) it has been supplanted by next generation sequencing methods for large-scale, automated genome analysis. [3]

#### Next generation sequencing

Next-Generation sequencing (NGS) technologies are usually applied as a general term for sequencing platforms that use post-Sanger technology to sequence DNA fragments in parallel. They were developed to reduce weaknesses in de novo sequencing. As a result, this technology had a major increase in efficiency as well as a decrease in cost per base and could produce a lot of data cheaply compared to de novo sequencing. The “down side” was that the read length was shorter, thus creating an assembly challenge [ ] as it made for example sequencing repeated elements difficult and could result in mis-assembled genomes as Figure X [ ] shows below.

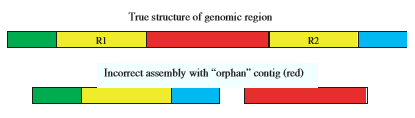


Figure ‑ – The challenge when sequencing repeated elements  
SOURCE: STEVEN L. SALZBERG AND JAMES A. YORKE   
 BEWARE OF MIS-ASSEMBLED GENOMES FIG.1

### Assembly

#### Assembly algorithms

#### Assembly challenges

As we have seen in the section above about assembly algorithms, quite a number of papers describe and discuss upon assembly methods, algorithms and how the assembly process works in general. But unfortunately, when it comes to the number of papers describing how (or discussing why) assembly is a difficult process, and challenges regarding it, the number decreases a lot. This contributes to making the understanding of genome assembly even more difficult for someone entering the field for the first time. Nonetheless, one paper that tries to discuss this is *Genome assembly reborn: recent computational challenges* by Mihai Pop,[1] where Pop use solving a jigsaw puzzle as a metaphor to an assembly process. Another complementary paper used for this section is *Genome assembly forensics: finding the elusive mis-assembly* by Phillippy, Scatz and Pop[2].

One of the problems regarding assembly process is about genome repeats which can be described as large stretches of sky in a jigsaw puzzle. Repeats tend to confuse the assembly process, because they seem identical to the assembler. They also make it difficult to distinguish between sequencing error and polymorphism among near-identical repeats. There are also the tandem repeats making it even more difficult. Many of the problems concerning repeats can be generalized in the following two categories:

**1. Repeat collapse and expansion**

The assembly can incorrectly gauge the number of repeats, thus including too few or too many copies, which can result in phenotypic differences, such as Huntigton’s disease.

The assembler can also mis-join reads originating from distinct repeat copies into one unit or include extra copies of repeat, both which can be noted as a higher or lower density of reads.

**2. Sequence rearrangement and inversion**

The assembler can shuffle the order of multiple repeat copies, which could be misinterpreted as a biological rearrangement event, meaning that one could draw wrong conclusions depending on the rearranged sequence.

During both repeat collapse and rearrangement, reads may get placed in a wrong copy of a repeat, therefore SNP could be a useful indicator of such a mis-assembly.

Another problem is looking at the complexity of an assembly, which depends on the number of reads being assembled. An assembly becomes more complex the more reads it assembles, making it quite complex, considering the development of shorter reads and sequencing tools that generates several million reads in a reasonable time and at a low cost.

One of the most time consuming task is probably the computation of overlaps. This task can have assembly errors which can occur due to limitations of the assembly algorithm, or by providing incorrect or incomplete assembly-parameters. It can be difficult to see where there are indels (an insertion or deletion of bases), mis-join, or find the exact placement of reads, and the detection of these errors are what scientists try to improve.

### Quality measures

There are many traps to avoid when it comes to assembly, and how well they are avoided can be measured and used to determine how well the results are. Some quality measures are easier to assess than others, especially with a reference genome. Of course, with a reference, the solution is already there, and the interesting part might be to spot the differences, compared to "normally" when the correctness of an assembly is undefined. It is therefore many criterias that can be used to assess the quality and correctness of an assembly such as the coverage and length of contigs or scaffolds, the length of the gaps between scaffolds, Nx (usuallyN50), how accurate or correct the sequence is compared to its reference, the error rate or how fast and cost-efficient it is, to mention some of the criterias. Other metrics such as the number of unaligned contigs, relocations, translocations and inversions can also be used by comparing to a reference genome. It is also possible to measure by metrics such as; the total number of contigs in the assembly, how long the assembly is (in number of bases), how long the misassembled contigs are or by looking at the (average) number of indels after x number of aligned bases.

Different measures can weight different depending on what the purpose of the assembly is. For instance, the size of scaffolds might be less important than the error rate in one case whereas the size of scaffolds might be crucial in another case. The problem is just that there’s not much that assess an assembly based on some kind of weighting. There is a statistical tool that measures some qualities, but it is rather mathematical giving only the countable/calculable scores without a proper weigth-adjustment. The next section will give a brief overview of this tool.

#### Quast

Quality Assessment Tool (QUAST) is a tool that evaluates and compares genome assemblies both with and without a reference genome. It is designed to improve existing assembly comparison software (such as GAGE) and produces results as reports, summary tables and plots that support SVG, PNG and PDF formats. An example of a metric that QUAST use is the NGx, which is like the Nx, but instead of comparing to the assembly length, the contigs are compared to the reference genome length. [15] As you can see in the figure below, quast gives a rather numerical report without giving the overall “best assembly” in the comparison.

Screenshot of quast-result html

## Bioinformatics

In N.M. Luscombe, D. Greenbaum and M. Gerstein’s review paper What is bioinformatics? An introduction and overview of the field (2001) [13] it is stated that bioinformatics is:

*“… conceptualising biology in terms of molecules (in the sense of physical chemistry) and applying* ***"informatics techniques"*** *(derived from disciplines such as applied maths, computer science and statistics) to* ***understand*** *and* ***organise*** *the* ***information*** *associated with these molecules, on a large scale. In short, bioinformatics is a management information system for molecular biology and has many* ***practical applications****.”* (page 83)

This definition, which was submitted to the Oxford English Dictionary, is a detailed way of saying that the science of bioinformatics is the use of (computer) technology in the field of biological macromolecules. But one thing this definition lacks is the fact that it does not clarify the difference between bioinformatics and computational biology that is commonly used interchangeably. Essential Bioinformatics by Jin Xiong (2006) [14] explains the difference as:

“Bioinformatics is limited to sequence, structural, and functional analysis of genes and genomes and their corresponding products and is often considered computational molecular biology. However, computational biology encompasses all biological areas that involve computation. For example, mathematical modeling of ecosystems, population dynamics, application of the game theory in behavioral studies, and phylogenetic construction using fossil records all employ computational tools, but do not necessarily involve biological macromolecules”. (p. 4)

In other words, bioinformatics is the practical approach concerning biological data, while computational biology is the theoretical approach of computer technology in biology.

#### Why is it important?

Bioinformatics is becoming a very important field because the amount of the data to work with is becoming bigger and bigger. And to store, work and analyze all this data to get the relevant information is a computer technical challenge. Some people might agree with the writer of Essential Bioinformatics, Xiong, when he writes that the goal for bioinformatics is to better understand a living cell at a molecular level to get a “global” perspective of the cell. Even though bioinformatic is having a major impact on many areas as in, for example, knowledge-based drug design, forensic DNA analysis and agricultural biotechnology, it is also worth mentioning that bioinformaticsc has its flaws as well. And to completely rely on something that can have limited accuracy or poor-quality can result in costly mistakes if not complete failures. But in the end, regardless of its limitation, bioinformatics is a field with a great potential for revolutionizing biological research in the coming decades. (Xiong, 2006)

### Benchmarking

Benchmarking can mean a specific standard that can be measured or judged by [12]. In computing, a benchmark is running a (set of) program or other operations in order to assess the relative performance. And in our case benchmarks provide a method of comparing two or more programs, assemblers, or tools to each other to determine which one is better fit for a given case/situation.

## Galaxy

Introduction to galaxy

# Material

For the purpose of reproducing the results in chapter 7, and the fact that I had two slightly separate paths towards the final product, I have chosen to explicitly divide the datasets, reference genomes and software used during my work-period. They will be separated based on whether they were used in reproducing the GAGE-B results or testing my Galaxy tool.

## Datasets

### GAGE-B

For the reproducing of the GAGE-B results, I initially started with all datasets available from <http://ccb.jhu.edu/gage_b/datasets/index.html> that I assembled using the recipe on their site[[1]](#footnote-1) with an assembler named Velvet[].

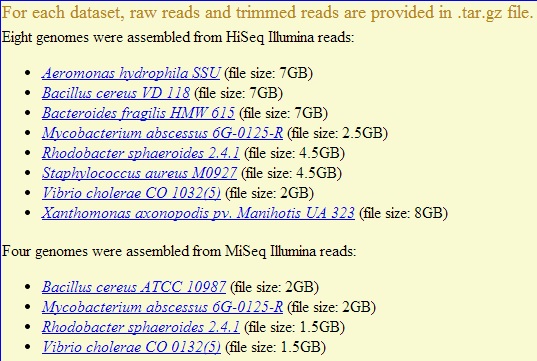


Figure ‑ – The list of all datasets used in the gage-b paper

After a while, realizing the amount of time and effort needed to reproduce the results fully with all datasets and all assemblers, I decided that I would focus on MiSeq and HiSeq data from the following specie:

Vibrio cholearea CO 0132(5) – MiSeq Illumina reads  
Download from: <http://ccb.jhu.edu/gage_b/datasets/V_cholerae_MiSeq.tar.gz>

Vibrio cholerae CO 1032(5) – HiSeq Illumina reads  
Download from: <http://ccb.jhu.edu/gage_b/datasets/V_cholerae_HiSeq.tar.gz>

V.ch was chosen because… it’s the smallest set with a total of 3.5 GB compared to a total of 4.5-8GB for the other sets containing both MiSeq and HiSeq data.

### Galaxy

The datasets used for testing/assessment of the Galaxy tool where the finished assemblies for Vibrio cholerae with various assemblers. The datasets were downloaded from link to download and consisted of

## Reference genomes

The reference genome used for assessing my reproduction of the GAGE-B paper and for testing of my Galaxy-tool where downloaded from GAGE-B’s webpage link to webpage. I’ve used both fasta files (i.e 1111.fna 1112.fna) and their corresponding gene files (i.e. 1111.gff and 1112.gff) from the list of many files available. The name of the reference genomes and link to their download site:

Reference 1 – link1

Reference 2 – link2

## Software

# Methods

This chapter covers a description of the tool and implementation.

## Reproducing the GAGE-B results

After a while, realizing the amount of time and effort needed to reproduce the results fully, I decided that I would focus on just one species.

Why vibrio cholera? Because of this and that.

## Galaxy

# Results

Samme som ch5

# Discussion

## Interpreting the GAGE-B results

## Analysis of galaxy tool

### Performance

### Potential use

### Weakness

## Further work

## Conclusion

Referanseliste!

Glossary =)

Glossary

|  |  |
| --- | --- |
|  |  |
|  |  |
| BAC | Abbreviation of *bacterial artificial chromosome* |
| bacterial artificial chromosome (BAC) | An artificially constructed segment of nucleic acid used for transforming and cloning in bacteria, usually E. coli. |
| bacteriophage | A virus that parasitizes a bacterial cell |
| chromosome | A threadlike, gene-carrying structure found in the nucleus. Each chromosome consists of one very long DNA molecule and associated proteins |
| contig | A continuous sequence of DNA that have been assembled from overlapping reads |
| coverage | The average number of reads representing a given nucleotide in a reconstructed sequence; also known as read depth or depth |
| de novo sequencing |  |
| deoxyribonucleic acid (DNA) | A double-stranded, helical nucleic acid molecule capable of replicating and determining the inherited structure of a cell's proteins |
| DNA | Abbreviation of *deoxyribonucleic acid* |
| eukaryote | An organism whose cells contain membraine-bound organelles and whose DNA is enclosed in a cell nucleus and is associated with proteins |
| fungus | About 80,000 known species of organisms of the kingdom Fungi, which includes the yeasts, rusts, smuts, mildews, molds, mushrooms, and toadstools |
| genome | The complete complement of an organism's genes; an organism's genetic material |
| inversion | A misjoin of a scaffold/contig where the two pieces map to the opposite strands on the same chromosome |
| nucleus | The chromosome-containing organelle of a eukaryotic cell |
| phage | A virus that infects bacteria; also called a bacteriophage |
| polymorphism |  |
| protein | A three-dimensional biological polymer constructed from a set of 20 different monomers called amino acids |
| read | Pieces of a sequence acquired under sequencing used for mapping/assembly that vary in length from less than 100 base pairs up to several thousand base pairs. Usually, with a double stranded chain, the reads contains the direction as well |
| read depth | See *coverage* |
| relocation | A misjoin of a scaffold/contig where the two pieces map to different locations on the reference genome |
| scaffold | A series of contigs that are in the right order but not necessarily connected in one continuous stretch of sequence. The remaining gaps between contigs in a scaffold can usually be sequenced because the placement of contigs are often known |
| translocation | A misjoin of a scaffold/contig where the two pieces map to different chromosomes or plasmids |
| viral | A biological virus |
| virus | A submicroscopic, noncellular particle composed of a nucleic acid core and a protein coat (capsid); parasitic; reproduces only within a host cell. |
|  |  |

Appendix a  
(Plot/Figure)



figure ‑

Appendix B  
(Tables)

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Table - Bla bla bla

1. Pop, M.,

*Genome assembly reborn: recent computational challenges.*

Briefings in bioinformatics, 2009. **10**(4): p. 354-66.

2. Phillippy, A.M., M.C. Schatz, and M. Pop,

*Genome assembly forensics: finding the elusive mis-assembly.*

Genome Biology, 2008. **9**(3).

1. <http://ccb.jhu.edu/gage_b/recipes/recipes.pdf> [↑](#footnote-ref-1)