A Toolkit For Reporting On Metagenome Assembly Quality

Report Name	Outline Project Specification
Author (User Id)	James Edward Euesden (jee22)
Supervisor (User Id)	Amanda Clare (afc)
Module	CS39440
Degree Scheme	G401 (BSC Computer Science (inc Integrated Industrial and Professional Training))
Date	February 4, 2016
Revision	0.3
Status	Draft

1 Project description

The project aims to produce a software application to serve as a toolkit for reporting on metagenome assembly quality. This toolkit should allow a user to provide input of a metagenome [4] (most likely in the FASTA format [5]) and receive output of a report detailing the quality of their metagenome. This project has a degree of research and understanding, and the designing and developing of an application to carry out the task.

Before any technical work can be done, I must determine what quality is in metagenomics. Quality could be multiple things, and could be up for interpretation depending what result the user wishes to see. The user may ask questions such as: Is it a chimera assembly, or is there something actually interesting and real? Is the assembly long enough to be useful? Or are they so long that it just seems irregular when it should be assembled from a varying amount of species and sub-species? Understanding the users requirement on quality will help me develop the application with the focus on the user.

It could also be useful to a user if the report suggests places where the metagenome may be split if there are irregularities or instances where it seems like the assembly doesn't seem right e.g. where a chimera may have been made by combining multiple reads in the wrong way. Can we give a confidence factor on how likely the assembly is to be of good quality? We can never truly say "This is good quality" as we don't know what we have to begin with, but the aim is to present a report on what areas of the assembly could be looked at in closer detail. We may be able to present comparison results between the same sample processed through different assemblies.

Advancing the application, it may be worthwhile to suggest areas of the assembly to look at in detail using the raw reads, before the assembly is created. By considering the reads themselves to try to report on the quality, a user can further delve into both interesting and suggested problem areas of the provided assembly. There is also the potential for using a large database of known genomes to see if any known genome matches all or part of the provided assembly.

Quality control in genomics is huge, as mistakes can be costly in time and financially. Without good quality control, a user may attempt to synthesize a gene based on an assembly that has no way of existing in nature, and not achieve the result they were originally aiming for. While the application may not be able to give exact results on how 'good' an assembly is, by highlighting significant areas where there may be issues, it will potentially aid the field of metagenomics in understanding the samples taken and reaching assemblies in better ways. It may also help users select better assembly processes by giving them tools to compare different assemblies from the same samples, and see if there are any known existing genomes that match their provided assembly.

2 Proposed tasks

Research metagenomics - Understand what they are, how are assemblies created, and how do they look in text-form? I will need to have basic understanding of the domain in order to develop the application to be useful to the users. Reading papers on metagenomics and determining quality in the field will help me do this, and a number of reading sources have been provided in the annotated bibliography at the end of the document.

Investigate existing/similar tools - There are a number of tools that exist to do some quality control checks, many for single species genome assemblies, and some for metagenomes. Some of the techniques I intend to investigate are K-mer [7] counting, GC counting [6], n50 and anything else I discover. Some applications already carry out these techniques, and I will look at these, such as Jellyfish [2], BFC [3] with Bloom Filter, REAPR [1], QC Chain [9]. Looking at these tools will help me decide the direction of my application, and whether I should use outputs from other applications as input, or take the raw metagenome and work with that. I

will also investigate what other applications do and don't do well, and find out why, to help improve my own application.
Investigate visualisation tools - For outputting the report of the quality of a metagenome, it would be good to display the assembly and highlight the areas of significance using some visualisation tools. I will be doing research into what tools exist that would fit the needs of my application.
Set up local environment and version control - Initially, I plan to use the Java language to develop my application, using the IntelliJ IDE. For version control, I will use git and keep my repository on my personal GitHub account during the projects lifetime. During this period, I will also decide which way to develop the application, whether I take a plan driven approach or an agile approach, using all or elements from Scrum, XP and Feature Driven Development.
Development - To begin with, I will just program some basics, such as GC content counting, both to learn and to start the application, and approach each technique as an addition to the application, building on it iteratively to quickly have a minium viable product that can easily be expanded upon and maintained over time.
Implement more advanced techniques - Once a minimum product has been made, that may only test using one or two techniques and provide a basic report, I will intent to further develop the application to provide a visual report about the metagenome provided. In the report, it will give the user enough detailed information that they can use in order to find why this report has claimed the metagenome is or isn't of quality, and potentially offer solutions as to what they might find useful to look at either to improve the metagenome, or discover where the irregularities are located. This will involve more advanced quality control checks, and the potential for checking databases for similar or matches to existing known genomes, such as using BLAST [8].
Construct a set of test scripts and files - In order to determine whether my application is successful in its tasks, I will write a number of test scripts for its functionality, and use real and artificially created data where I have expected and assumed outputs and see if my application gives me the anticipated results.
Compare outputs/the report - Compare the results from my program with outputs from similar applications that do similar tasks, or communicate with those in the field to determine whether the application has some degree of success with their expectations.
Project meetings and online blog - Attend project meetings with my supervisor (minimum) once a week, and discuss my progress and plans. These will also be documented on my blog, and will reflect the stories I am taking into each week to work on.
Preparation for demonstrations - There are two demonstrations. The first is a mid-project demonstration in the week before Easter, while the second is a final demonstration after the submission of my technical work and final report. Both of these demonstrations will be planned for and practised before being given, and through them I hope to show my markers the function of my application, any research I have conducted and any technical challenges or interesting sections of my application.

3 Project deliverables

Mid project demonstration notes - A compiled set of notes used in planning and giving a mid-project demonstration. This will be included in the appendix of the final report.
Test Files - Taken from sources I am able to use, or artificially generated for checking that my application does indeed return a report of expected quality when used in practice. These will be as part of the technical submission.
Test Scripts - Most likely using JUnit as the base, these will be included in the technical submission of the software application, and where relevent run with the test files provided. A series of scripts to test the expected functionality of my application, included in the technical

submission.
Software Application - Takes the input of a metagenome assembly, or of output results from other software results, and returns a report on the quality of the metagenome in question. This will be the bulk of the technical work and the focus of this project.
Story cards and planning documents - Within the final report appendix will be a document detailing the stories I undertook during the project process and any planning documents I created to aid the project development.
Fiinal Report - The final report documents my process over the projects life time, the work done and acknowledgements to any papers and journals read during my research, third party software and tools used during the project. An appendix will be attached, including any deliverables that are required to be included but not part of the report itself.
Final Demonstration - While there will be no documents, this is an event that will take place and I will need to consider how I structure the demonstration and how to present it during the time of my project

Annotated Bibliography

[1] M. Hunt, T. Kikuchi, M. Sanders, C. Newbold, M. Berriman, and T. Otto, "Reapr: a universal tool for genome assembly evaluation," http://www.sanger.ac.uk/science/tools/reapr/, May 2013, accessed February 2016.

> REAPR is a tool that attempts to evaluate genome assemblies, much as this project intends to do. It will be interesting to try the software and their approach to evaluation.

[2] G. Marais and C. Kingsford, "Jellyfish - fast, parallel k-mer counting for dna," http://www.cbcb.umd.edu/software/jellyfish/, July 2011, accessed February 2016.

> Rather than writing my own k-mer counting tool, it may be apt to either use, or request the results from, the JELLYFISH application, that could potentially provide better and faster results for k-mer counting than I would be able to do with the time available to me during the lifetime of this project.

[3] P. Melsted and J. K. Pritchard, "Efficient counting of k-mers in dna sequences using a bloom filter — bmc bioinformatics," http://pritchardlab.stanford.edu/bfcounter.html/, Aug. 2011, accessed February 2016.

> BFC is a tool for k-mer counting that uses a bloom filter. This would be a good alternative to investigate, similar to JELLYFISH, when deciding how to determine quality with the application written as part of this project.

[4] T. Thomas, J. Gilbert, and F. Meyer, "Metagenomics - a guide from sampling to data analysis," http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3351745/, Feb. 2012, accessed February 2016.

> A detailed explanation of metagenomics in the field of biology, its uses and suggested construction for scientific usage. Reading and understanding this paper will help me understand the domain of the project.

[5] Various, "Fasta format - wikipedia, the free encyclopedia," https://en.wikipedia.org/wiki/FASTA_format/, Nov. 2015, accessed February 2016.

> FASTA is a common format for representing assemblies in text-based form. I expect to make large use of FASTA files for my application to process a metagenome assembly and for my test files.

[6] ——, "Gc-content - wikipedia, the free encyclopedia," https://en.wikipedia.org/wiki/GC-content/, Oct. 2015, accessed February 2016.

> From this web link, the GC content is "the percentage of nitrogenous bases on a DNA molecule that are either guanine or cytosine". By measuring this in small equal sections of the provided assembly, the application may be able to naively judge where there is a split in the data that may indicate an irregularity.

[7] ——, "k-mer - wikipedia, the free encyclopedia," https://en.wikipedia.org/wiki/K-mer/, Oct. 2015, accessed February 2016.

> k-mers are a way of representing subsequences of strings, and are often used in sequence assembly, sequence alignment and genome quality assessment. I intend to use the technique to help product the report of quality.

[8] ——, "Blast: Basic local alignment search tool," http://blast.ncbi.nlm.nih.gov/Blast.cgi/, unknown, accessed February 2016.

> BLAST is a tool used for searching the NCBI database of known sequenced genome assemblies. I could use this to see if any assembly provided by the user, or portions of them, match known assemblies in the NCBI database.

[9] Q. Zhou, X. Su, G. Jing, and K. Ning, "Meta-qc-chain: comprehensive and fast quality control method for metagenomic data - pubmed - ncbi," http://www.ncbi.nlm.nih.gov/pubmed/24508279/, Feb. 2014, accessed February 2016.

> Meta-QC-Chain is a quality control tool for metagenomic data that attempts to highlight areas of errors, irregularities and noise in the data through known quality control techniques, similar to the project posed here. This will be useful in seeing what alternatives and techniques exist that I may also use, or to avoid where there may be other available techniques these tools do not cover and use.