A Toolkit For Reporting On Genome 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	0.2
Status	Draft

1 Project description

Making a toolkit for reporting on genome assembly quality, in reference to metagenomics. This toolkit should allow a user to provide input of a genome (most likely in the FASTA format) and receive output of a report detailing the quality of their genome. This project has a degree of research and understanding, and the programming of an application to carry out the task.

Before any technical work can be done, I must determine what quality is, with particular reference to metagenomics? Could be multiple things, mostly up for interpretation. Is it a chimera? Are the contiguous reads (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?

It would be useful to suggest to a user places where the genome 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.

Advancing the application, it may be worthwhile to suggest areas of the assembly to look at in detail using the reads. 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 of the contigs we have, or even portions of them where we think they are likely to exist in nature, have any close or exact matches to sections of known genomes.

This project is useful for checking the quality of an assembly, and can potentially be used for both single species and metagenomes, although my focus will be on the latter. Quality control in genomics is huge, as with mistakes things can go very wrong, and irregularities in sampling, aligning and assembly of genomes can very easily occur. The more quality checks we can do, the better.

In metagenomics, where we don't know what our sample might contain in the first place, and so the resulting assembly could be something not existing in nature but that an assembler (or many) believes to be correct based on the reads, we really must carry out quality checks otherwise any genome or its parts that we find interesting may be entirely useless.

By having an application that gives us this quality control, it could further advance the field of metagenomics in helping to assure those working with metagenomes that the assemblies they have are reliable, or where they are not, why exactly they might not be of good quality, for all the reasons we may cite something is good or bad quality.

2 Proposed tasks

Research metagenomics - Understand what they are, how are assemblies created, and how do they look? Use samples from Sam (FASTA files of limpit gut sample). Read papers on the subject, including... (Include links here - Saved in bookmarks toolbar) ...
Investigate existing/similar tools - Read about current quality assessment tools - Kmer counting, GC counting, n50, etc - See Jellyfish, BFC with Bloom Filter, REAPR, QC Chain. What tools exist for similar or the same functions? This will help me decide the direction of my application, and whether I should use outputs from other applications as input, or take the raw genome and work with that. I may also investigate what other applications do and don't do well, and find out why, to help improve my own application.
Set up local environment and version control - I'll be using Java, developing with IntelliJ IDE and using GitHub for my version control repository.
Development - Program some basics, both to learn and to start the application - Begin with perhaps GC count and Sam's sample, in Java, and build on the application from there. Whether I

use my own programming, use packages for methods of checking quality or even just ask the user to provide the output of other software for use, will be fully determined over time.
Implement more advanced techniques - Attempt to provide some sort of confidence report about the genome provided. In the report, give the user enough detailed information that they can use in order to find why this report has claimed the genome is or isn't of quality, and potentially offer solutions as to what they might find useful to look at either to improve the genome, or discover where the irregularities are located.
Working iteratively - Starting with core basics for quality and reporting, and adding more complex and detailed functionality as time goes on. The project has room for expansion with multiple existing quality control techniques to fit the needs of reporting on quality in a metagenome. It is possible I may leave the project open ended, but the main task of providing some sort of completed toolkit for the quality reporting should be done.
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.
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.
Software Application - Takes the input of a genome assembly, or of output results from other software results and returns a report on the quality of the genome in question.
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 - Documents my process, the work done, acknowledgements to any research I did, 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