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	0.1
Status	Draft

---

1 Project description

Making a toolkit for reporting on genome assembly quality, in reference to metagenomics

Determine what quality is? Could be multiple things, mostly up for interpretation. Is it a chimera? Are the contigs long enough to be useful? Or are they so long that it just seems unplausible to even be part of a metagenomic sample? Suggest places where the genome may be split if there are irregularities or instances where it seems like the assembly doesn't seem right, or 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. Suggest areas of the assembly to look at in detail with the reads. Use the reads themselves to try to report on the quality, 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 and suggest possibilities for what we have in our assembly.

Useful for checking the quality of an assembly, can potentially be used for both single species and metagenomics, 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 part of a genome, 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).

Research existing tools - Read about current quality assessment tools - Kmer counting, GC counting, n50, etc - See Jellyfish, BFC with Bloom Filter, REAPR, QC Chain

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 and attempt to provide some sort of confience 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.

Compare outputs/the report 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. Second to this, create a pool of known 'good' and 'bad' quality assemblies artificially, and see if the application gives the expected results.

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 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

Test Scripts - Most likely included in the project and 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.

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 - Within the final report appendix will be a document detailing the stories I undertook during the project process

Fiinal Report - Documents my process, the work done, acknowledgements to any research I did, third party software and tools used during the project

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] H. M. Dee and D. C. Hogg, "Navigational strategies in behaviour modelling," *Artificial Intelligence*, vol. 173(2), pp. 329–342, 2009.

This is my annotation. I should add in a description here.

[2] S. Duckworth, "A picture of a kitten at Hellifield Peel," http://www.geograph.org.uk/photo/640959, 2007, copyright Sylvia Duckworth and licensed for reuse under a Creative Commons Attribution-Share Alike 2.0 Generic Licence. Accessed August 2011.

This is my annotation. I should add in a description here.

[3] M. Neal, J. Feyereisl, R. Rascunà, and X. Wang, "Don't touch me, I'm fine: Robot autonomy using an artificial innate immune system," in *Proceedings of the 5th International Conference on Artificial Immune Systems*. Springer, 2006, pp. 349–361.

This paper...

[4] W. Press *et al.*, *Numerical recipes in C*. Cambridge University Press Cambridge, 1992, pp. 349–361, 0123456789.

This is my annotation. I can add in comments that are in **bold** and *italics and then other content.*

[5] Various, "Fail blog," http://www.failblog.org/, Aug. 2011, accessed August 2011.

This is my annotation. I should add in a description here.