MSc Bioinformatics
Software Development Group Project 2017

Background and schedule

The overall aim of this module is to give you the experience of working together within a team to produce a working prototype of a web-based software tool for interpreting the results of transcriptomics experiments. Specifically, results obtained from RNA-seq followed by *de novo* assembly.

The project starts on Monday 9 January, with the release of this document. You should study this document, think about how you might go about tackling the project, and assemble a list of questions that would help you clarify what you need to do and how you are going to do it. On Tuesday 10 January at 11:00am there will be a tutorial in Fogg 3.15 to discuss the project specification in more detail, field any questions that you may have, and ensure that you have everything you need to make a start on the project.

The project finishes on Friday 17 February, with a demonstration of your final prototypes and a presentation during which each student will explain their contribution to the project. During the project, tutorials will be held weekly and help on specific topics provided as necessary – dates and locations will be posted on QMPlus.

Team membership is as follows:

Mango	Melon	Mandarin
STEVEN	NADIM	JAMES
MAHIN	MODUPEH	KRISTINA
KANTEH	MADELEINE	DANIA
SETH	ANDREW	YUSEF

Software specification

The software should allow the user to upload a set of FASTA files containing the sequences of *de novo* assembled transcripts and their FPKM values from multiple samples (details of the format are provided below in the sample data section). It will then allow the user to:

1. Infer which genes each transcript relates to (if any) by sequence homology.
2. Identify transcripts that have significantly different expression between different samples.
3. Explore the expression data graphically. Visualisations might include principal components analysis, volcano plots, and hierarchical cluster analysis.

The software is expected to be a working prototype, which is to say that it will provide a working demonstration of the functionality described above but would need to be passed to professional developers/web designers to turn into a fully polished web application. Documentation should therefore be provided, both within the program code and in a dedicated document, to explain how the software is structured and how it works.

How to begin the project

Successful completion of this project requires a combination of technical skill, good organisation, logical thinking, web-based research and possibly a visit to the library. Your first task is to work together to do the following:

1. Ensure that you understand the software specification and sketch out some kind of architecture for the software (i.e. what components are needed and how they should interact).
2. Determine which technologies you need to produce the software.
3. Find out enough about the technologies so that you are able to approximate how long the different parts of project will take, and who is best suited to complete them.
4. Agree on the optimal way of working together to complete the project.
5. Identify any specific new skills that need to be learned by team members.
6. Produce a development plan for the duration of the project, detailing the various tasks and who will be responsible for them.

The software architecture and development plan should be presented at the second tutorial, on the morning of Thursday 12 January (see QMPlus for time and location). Provided that the development plan is acceptable, you will then embark on this plan to develop the software.

Assessment

The assessment composes three elements, listed in the table below. Detailed marking schemes for each part of the assessment are provided on the following pages. The written components and presentations should be submitted electronically via QMPlus by 6pm on Thursday 18 February. Due to the nature of the project, no deadline extensions are possible.

Assessment	*Type*	*Description*	*Weighting*
Software and documentation	Group	Marks will be awarded according to the software functionality and its documentation.	60%
Written reflective report	Individual	A report of not more than 850 words about the role you played in the project.	20%
Presentation	Individual	A group presentation, in which each student is expected to contribute five minutes.	20%

To pass the module, a mark of at least 50% is needed in <u>each</u> of these assessments.

Sample data

For development purposes, we provide example data from *Pteropus Alecto*. This data, which is available via QMplus, follows this experimental design:

File name	Sample status	Replicate
10s0r1.fasta	Uninfected	1
11s0r2.fasta	Uninfected	2
12s0r3.fasta	Uninfected	3
13s8r1.fasta	8 hours after infection with Hendra virus	1
14s8r2.fasta	8 hours after infection with Hendra virus	2
15s8r3.fasta	8 hours after infection with Hendra virus	3
16s24r1.fasta	24 hours after infection with Hendra virus	1
17s24r2.fasta	24 hours after infection with Hendra virus	2
18s24r3.fasta	24 hours after infection with Hendra virus	3

Each FASTA file contains all the transcripts assembled from one sample, in no particular order, each with a header containing an arbitrary name like `asmbl_10005` and the FPKM value for the transcript (note that the transcript names are not common across different samples). A portion of the FASTA file from one of the samples looks like this:

```
>asmbl_10005;2.70
AAGTATTTTCTCTACCACGGTACCAAAAATATCTAGAGACGTGATAACATTTATGTTCAGAAAAAAACACCGAGAATAATTT
TCACAGACGTAACAAATGTGATGCCCGCACACAACTAAGGCTGTTGAGGGAAAATCAAATTAGGATCATTTTCTGCTTTGTC
TAGAATGTTCTAAGTAGAAACTATTCAGAAATGCATTGTGGACTACCAGTCCAAATAGAATAAAACAACAAATATTTT
>asmbl_10006;3.63
AAATTTCCTCACTCCTACCCTTCCCCTTCCTTGACATGTCTCATATTCCTTCATTCTAACTCAGAATACCAGGCTATATTTG
AAGGTAGACGTCGAAGAGGCTGGCATTGGTTTTTAGAACCAGCAGAATAAAGTGTATACTTGTTGACTCAAACAGCAGATAT
AGATAGGAAAGGATATTACATTACATTAATAGCACCTTAGAGTTACAAATTCTTAAGAA
>asmbl_10007;2.06
CAGAGAGTTAATCAAATACCAAGTACCAAGAGGCAAAGATCTCAAGTATAAAGGGAGAAATGAGTGTGACGGTCTATGCTTT
TTCCCCCGAAGGCATTTCCCAAGATGAGAGTAATAAACTAAGAGTTGAATAAGCTGAGCGTGGCTTTGGAGTCTTACAGGAC
TAAGAGGCAAAAATACACTTCCGTATACTCCAGAGAAGAGGGGTACTAGTAGACACTTCGCTTTCA
>asmbl_10008;3.66
GCTCATGGCTCTCCTGCTGTCTGGGGAAACGCTGTTAGGTACTTGTCATCTTTTCTTGGCCTCGGGCTCCCTCACTGGTACA
GTGCCCGGCATGCAACAGAGACACCATCAGTGTTGGTGAACAAATGACTGCTGAATCAGGAAAACGAAGGAGTTGAACTAGA
CCTGTAAGGTCCCTTCTGGTGTTGCCAGCCTATGAATTACAATTGCTTAACACAGTAACTGCGAG
>asmbl_10009;4.46
TTTTTTTCCTAGAATGAATAATTTTTTTTTTCCTAATTAAAAAAAAAAAAATAGGAATCTTAACAGGTAAAGCAGATTTGGGA
GGTTTTGGTCCTTACAAGAATATCTCATTTCTTATTCTTATGTCAGGAAGAAGTTGGTGAATCAAACCACACCGGTGGGTAT
GCTCTCCCATTGAAGGCCAGGGACATCTGCCTCATATATGGTGCTG
```

Although we provide this specific example dataset:

- It is important that the software works with any dataset fitting this general format.
- Because these files are quite large it would be sensible to create a smaller dataset for developing, testing and demonstrating the prototype.

MSc Bioinformatics: Group Project Marking Scheme

Assessment of the group project module comprises three pieces of work. To pass the module, a mark of at least 50% is required in each piece.

Software and written documentation (60% of module mark)

Marks for this section are awarded per group, rather than individually. The total mark (out of 30) is determined by summing the individual components below.

Software: The software should be provided in a convenient way such that the organisation of the files is clear. An explanation of how to execute the software should be provided. The source code should include comments to help the reader understand how it works. The assessors may evaluate the software using a dataset of their choice. Marks are awarded with reference to the tables below.

Mark	Component 1: Software functionality
10	Functionality substantially beyond what was requested, e.g. professional look, smooth and useful interactivity, parameter options for the various analyses.
7-9	All the required functionality, in a usable form. Very minor bugs.
5-6	Basic functionality that just about meets the specification. Maybe a few bugs.
1-4	Functionality matching some elements of the specification.
0	No software provided.

Mark	Component 2: Software readability and coding style
10	Code organised and commented to a professional standard.
7-9	Readable well-structured code with some pertinent comments.
5-6	Code shows some structure and is understandable with some effort.
2-4	Code is messy and hard to follow.
0-1	No comments; confusing code

Documentation: The report should explain how the software is structured and how it works. It should outline the overall design philosophy, then go on to explain the technologies that were used to develop the software – why these were chosen and how they were used. Limitations of the software and opportunities for future development should also be explored. There is no specific word or page limit, but a document of around 10-20 pages is likely to be sufficient.

Mark	Component 3: Documentation
10	Professional standard report in terms of clarity, content and presentation.
7-9	A clear explanation of how the software works, showing critical insight in terms of the technical solutions adopted and/or the limitations of the software produced.
5-6	A basic explanation of the software implementation, which is mostly readable and technically correct.
1-4	Report gives some explanation of the software, but it is not particularly clear, accurate, or comprehensive.
0	No report provided.

Reflective report (20%)

This should explain the role that the student played in the project. It should reflect on the challenges that the student faced during the project and what was learnt to overcome them – both in terms of technical skills and transferable skills.

When assessing this report, markers should consider questions such as (i) is the information in the report technically accurate; (ii) are the new skills reported actually relevant to the project; (iii) was

the contribution claimed by the student critical to the project; (iv) was the contribution claimed by anyone else.

Mark	Reflective piece
10	Evidence of a leading contribution by the student, critical insight and significant personal development.
7-9	A strong contribution from the student, with evidence of insightful thinking and personal development.
5-6	Evidence that the student has contributed to the project, and learnt new skills in the process.
1-4	Basic information about the student's contribution but no evidence of personal development or critical insight.
0	No report provided.

Presentation (20%)

This will cover essentially the same material as (1) and will include a demonstration of the software. Each student is expected to contribute five minutes, and is marked individually according to the SBCS presentation marking scheme (see new page).

SBCS Presentation marking scheme

Point	Corresponding grade *(for guidance only)*	Criteria
0	F--	No show
1	F-	Poor in every element (see below for elements to include)
2	F	Poor in most elements
3	E	Poor in several elements
4	D	Fair; below average
5	C	Good; average
6	B	Good; above average
7	A-	Excellent; substantially above average
8	A	Excellent; evidence of creativity and originality
9	A+	Excellent; significant creativity and originality
10	A++	Superb in every element; masses of creativity and originality

Use your judgement to moderate the balance between different elements or to include items that are relevant but have not been specifically mentioned. On the mark sheet, enter a mark out of ten for each aspect of the seminar. Sum these to obtain an overall mark out of 40 for the seminar presentation. Please record the literal grade use the scale indicated on the control panel

Presentation skills – *how well was the interest/attention of the audience stimulated/maintained?*

- quality of the delivery (e.g. audibility, speaking clearly and at an appropriate pace, voice modulation when appropriate/necessary, not reading from extensive notes, maintaining good eye contact, avoidance of irritating/distracting mannerisms)
- quality of the visual or other aids (e.g. creativity, adequacy, legibility)
- keeping to the allotted time

Ability to convey information – *how well was the 'message' communicated?*

- logicality of the structure/sequence of concepts
- clarity/comprehensibility of the descriptions/arguments
- balance/timing between the various components

Academic content and rigour – *what was the quality of the information being conveyed?*

- was the level commensurate with that of a MSc degree?
- were there any significant omissions/weaknesses?
- the accuracy/veracity of the statements/observations
- the validity of the conclusions which were drawn

Questions – *what was the extent of his/her current awareness and background knowledge?*

- was the wider significance/relevance of the work made apparent enough to inspire the audience to seek further information?
- were questions answered clearly, competently, concisely?