MSc BIOINFORMATICS
SOFTWARE DEVELOPMENT GROUP PROJECT 2022

Background and schedule

The overall aim of this module is to give you the experience of working together within a team to produce a functioning prototype of a web-based software tool for handling molecular biology data. Specifically, this project focusses on human transcription factors.

The project starts on Monday 24 January, with the release of this briefing document. An online session will be held at 13:00 GMT on Tuesday 25 January – before then you should connect with your teammates, 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.

The project finishes on Friday 4 March, 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 joining details can be found on QMPlus.

Team membership is as follows (team names are randomly assigned – they don't indicate leaders):

Team Zainab	Team Daniella	Team Carlo	Team Isabel	Team Liam	Team Celine	Team Christian	Team Harry
Ervin	Caterina	Carlo	Diego	Khairnar	Amanah	Christian	Alexandra
Gabriel	Daniella	Dhiraj	Isabel	Liam	Celine	Eri	Harry
Zainab	Kenneth	Jack	Janeesh	Uppy	Gracia	Rym	Laavanya
Zibo	Will	Shiloh	Venkata	Vi	Pavan	Yasemin	Manikanta
							Waleed
Thursday tutorial time slots (GMT):							
10:00	10:40	11:20	12:00	13:20	14:00	14:40	15:20

Project background

Population genomics is the science that studies the genetic diversity within and between populations. By analysing the similarities and differences of genetic variation between individuals from different populations, it is possible to infer historical (both neutral and adaptive) events that characterized the evolution of said species. Inferences from population genomic data are widely applied in conservation genetics (e.g., molecular monitoring of endangered species), evolutionary biology (e.g., demographic reconstruction), and precision medicine, among many other fields.

The discipline has received notable attention in recent years thanks to the technological advancements of sequencing genomes at large-scale. The advent of next-generation sequencing (NGS) technologies has allowed researchers to obtain vast amounts of genomic data for many samples belonging to several populations. After sequencing, mapping, filtering and variant calling, data is usually reported in variant call format (VCF) files which contain genotype (or haplotype) information for sequenced sampled at each called single nucleotide polymorphism (SNP). In the case of human population genomics, the most popular database is The International Genome Sample Resource which collects sequencing data from multiple human populations, mostly from the 1000 Genomes Project data set.

Software requirements

The aim of the group project is to build a web application that can retrieve SNP information for a genomic region of interest from a VCF file. It should also be able to calculate some summary statistics and return their value and plot their distribution.

Specifically, the web application should satisfy the following requirements:

1. The user should be able to retrieve SNP information given either a genomic coordinate (chromosome, start and end), SNP name (rs value), or gene name (or any aliases associated to it).

2. The application should return the following information for each SNP: name (rs value), genomic position, genotype frequencies, and allele frequency. Frequencies should be provided for each population separately.

3. If multiple SNPs are returned, the user should be able to select the population(s) and summary statistics of interest, and the application will calculate them and plot their distribution in sliding-windows along the region. The user will also be able to download a text file with the values of summary statistics. At least three summary statistics should be reported, for instance: one index of genetic diversity, one on haplotype diversity (e.g. homozygosity), one test against neutrality (e.g. Tajima's D). Additionally, if multiple populations are selected, then population genetic variation (FST value) for each pair of populations should be reported.

Some practical considerations: Initially, we suggest using sequencing data from only one of the human chromosomes (e.g. chromosome 22) and no more than five populations from the 1000 Genomes Project dataset, to minimise data storage requirements. As an extra (optional) point, the application should return the derived allele frequency which can be inferred as the frequency of the non-chimpanzee allele. Remember that results should be presented in a manner that will help to answer biological questions.

The software is expected to be a working prototype, which is to say that it will provide an interactive 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 with your team to do the following:

1. Ensure that you understand the software requirements and sketch out an architecture for the software (i.e. what components are needed and how they should interact).
2. Determine which data and technologies you need to produce the software.
3. Find out enough about the data and technologies so that you can approximate how long the different parts of the project will take, and who in your team 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 (e.g. Gant chart) 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 first tutorial, on Thursday 27 January. Provided that the development plan and architecture are acceptable, you will then embark on this plan to develop the software.

Getting the most out of the tutorials

To make the most of the weekly tutorials you need to be organised. Before the tutorial prepare to give a brief summary of what your group has been working on, perhaps using slides or a demo if you think that would help, and be sure to arrive in good time. Think in advance about the questions you want to ask and the advice you need.

Getting help and advice between tutorials

Please post any questions to the QMplus discussion forum for this module. This will be monitored every weekday until the end of the project. To avoid giving away your team's best ideas, be careful about what exactly you post.

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. Due to the nature of the project, no deadline extensions are possible for the group work.

Assessment	*Type*	*Description*	*Weighting*
Software and documentation Due 18:00 GMT Thursday 4 March	Group	Marks will be awarded according to the software functionality and its documentation.	60%
Presentation Presented Friday 5 March	Individual	A group presentation, in which each student is expected to contribute five minutes.	20%
Written reflective report Due 18:00 GMT Friday 12 March	Individual	A report of not more than 850 words about the role you played in the project.	20%

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

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 requirements. Maybe a few bugs.
1-4	Functionality matching some of the requirements.
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 you (as an individual) played in the project. It should reflect on the challenges that you 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 topics as the documentation 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 next 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?