Genome Analysis

General notes taken from lectures, seminars and labs.

Content

[Grading 2](#_Toc35951199)

[Recap 4](#_Toc35951200)

[Lecture 1 and lab introduction 6](#_Toc35951201)

# Grading

We will receive an individual grade for the lab project.

To get a 4 as grade I need to do at least two of the following:

* Lab manual questions: you’ve answered the questions in the lab manual, and you’ve done so correctly.
* GitHub wiki and presentation: your wiki and final presentation are well-structured and display a clear understanding of the methods used and the meanings of the results.
* Extra analysis: you’ve preformed at least one extra analysis and extensively discussed the result.

For a 5 you need to fulfill all above, and do multiple extra analysis.

**Article 4**

Basic analyses:

* Metagenome assembly.
* Binning.
* Quality check of assembly and bins.
* Functional annotation
* Basic phylogenetic placement of bins (Taxonomic ID).
* Reads preprocessing: trimming + quality check (before and after)
* Analyses of activity (expression level) of different bins.

Extra analyses:

* Abundance of different organisms/ bins.
* Refine taxonomic ID of assembled genomes.
* Metabolic pathway reconstructions for chosen bins
* Analysis of expression data of chosen gene groups (i.e: respiratory genes, genes involved in carbohydrate metabolism, etc).
* Comparisons across bins (pathways, expression certain genes groups, etc).
* Comparative genomics of bins
* Orthologue gene clustering of bins

**To keep in mind**

Don’t use raw data directly in the analysis because quality might not be optimal. First: assess the quality of your data first. Perhaps remove reads with low quality or that adapters is still present in the reads. FastQC: check quality of short reads.

You want to preprocess reads by remove the low quality base-calls and present adapters from the reads. Trimmomatic: can be used to trim and crop illumina (FASTQ) data as well as remove adapters.

Multiple softwares for genome assembly.

Assembly evaluation: QUAST and MUMmerplot.

Binning: after assembling reads coming from metagenomes we end up with a mix of contigs that come from different organisms. Binning is a process of classifying those contigs into different organisms to reconstruct their genome. Metabat software can be used for this. After binning it’s important to evaluate the recovered bins. We want statistics for each bin such as size, GC content, verifying if each bin represents a complete genome or only a part of it. CheckM provides a set of tools for assessing this.

Annotation: after assembling genome we want to see what genetic element it encodes.

Homology search: wanting to see how similar organisms are. Blastn.

Phylogenetic placements: metagenomes might contain a large microbial diversity. If we want to study organism under evolutionary context, phylogenies are useful. Phylophlan.

Mapping: aligning reads against a reference to determine reads coverage or to improve/correct assembly, account for differential expression of certain genes. BWA and Tophat.

**Deadlines**

* 30/3: Project plan
* 17/4: Assembly
* 23/4: Binning
* 29/4: Bin quality assessment + annotation
* 5/5: RNA mapping
* 8/5: Phylogeny

# Recap

**Taxonomy** = to sort organisms into groups. You classify by diving biological organisms into groups with consideration to a set of predefined criteria such as morphology and genetics. Organisms are grouped together into taxa and these groups are given a taxonomic rank. The ranks used today are domain, kingdom, phylum (plural phyla), class, order, family, genus and species. A taxon is a group of one or more populations of an organism or organisms seen to form a unit. A taxon is usually known by a particular name and given a ranking.

**Clade** = monophyletic group; a group consisting of organisms with a common ancestor and all its lineal descendants. The group consists of all the descendants!

**Metabolism** = the set of life-sustaining chemical reactions in organisms. Metabolism can be divided into catabolism and anabolism. Catabolism = degrading processes which generates building blocks, energy and reduction power. Anabolism is construction of building blocks, mainly reductive, electron absorbing reactions. The chemical reactions of metabolism are organized into metabolic pathways, in which one chemical is transformed through a series of steps into another chemical, each step being facilitated by an enzyme. Enzymes are crucial to metabolisms.

Microorganisms are facultative if they can alter between different metabolic pathways (f.ex. how they receive energy).

**Metagenomics**

Metagenomics it the science that studies genes and genomes of non-cultative organisms. It’s a cultative independent set of methods used for large scale analysis of organisms. It describes what organisms are present in different environments and thereby gives us understanding of the diversity and evolution of organisms in their natural habitat. Metagenomics will answer the following questions:

* What organisms are present in the sample? What kind of bacteria?  
  You often use ribosomal genes to get some understanding of what organisms are present. You want to make PCR and can’t therefore use species-specific primers – need primers that can attach to a gene most organisms have. It’s common to use 16S/18S and amplify with PCR for phylogenetic analysis.
* What is the purpose of the organisms? What happens as a whole in the system?   
  You want to investigate what genes are present in the environment because it will tell a great deal of what functions exist.
* Which organism does what?

**Metatranscriptomic**  
Science that studies gene expression of microbes within their natural environments. It allows to obtain whole gene expression profiling of microbial communities. Metagenomics focuses on studying the genomic content and to identify which microbes are present. Metatranscriptomics is used to study the diversity of the active genes and quantify their expression levels and monitor how levels change under different conditions.

De novo assembly:

Single cell sequencing

# Lecture 1 and lab introduction

Prepared material for sequencing methods and data management plan on stadium.

Block on de novo-assembling methods starts on Friday. Two lectures focus on DNA single genome algorithms and design. Different blocks: de novo, annotation, reference-based analyses and comparative genomics.

Will try to invite us to a Zoom-meeting in the end of each scheduled lecture.

3 seminars. The questions on seminars will be on the written exam (the type of questions that we should be able to answer after the course). 2 seminar groups.

Computer lab: lab introduction 24/3 directly after the introduction.

**Lab introduction**

* Sequencing data
* Reads pre-processing
* Assembly
* Quality checks
* Annotation
* Reads mapping
* Extract biological information

1. Read paper  
2. Extract main ideas and define project plan: main tools and steps they use to get the results. Write plan that resembles what they did in the paper.  
3. Perform your analyses and interpret the results

Use informative file names. Documents EVERYTHING you do. Biostars and stackexchange.

Things that will be graded  
- Github wiki: report everything you did. Everything needed to reproduce your product. Project plan, methods and interpretation of result.  
- Presentation: Monday 25/5 – upload slides. 10 + 5 min presentation. Should include aim, methods, results and biological interpretation.

**Project plan**Should be placed in the wiki and should include:

* Overview of the project: goals, the questions you want to answer, type of sample and type of data.
* Mention analyses and softwares you will use
* Time frame: need to plan long-running times. Attention to checkpoints and plan for long running times.
* Data management plan

Can draw a figure showing how the data is processed in each step. What is the output. Nice flow chart.

Project plan before the first scheduled lab session. Doesn’t have to be perfect. Lab manual, instructions. Second section in the manual.

**Version control – Git, GitHub**We let the computer handle this for us. Git is a specific protocol for version control. Open-source: free to use. Distributed version control system = version history of files is stored on a server. Users can then make a copy of the respository locally and add files to it. Send changes back to the server. Multiple servers at the same time. If you loose files: the server will still have a copy of the file and version history of it. Github is a cloud-based system to store the respositories. They have also lots of other features linked to the respositories.

You have a working directory where you edit your file. Then you add them to the index; which is a place between the directory you’re working in and the local repository. This is done by the add-command. Once you’ve added all the files you want to add to the index, you commit them to the local repository. With this command you also add a message explaining the changes you’ve done. Then the changes are stored at the local computer = local respository. Then you will push the files and changes to the remote repository found on GitHub. This is done by the push-command.

Store all code for project in GitHub. You should store python-scripts as well as one-linears that is run in the terminal. HOW TO STORE THIS?

Included in the wiki

* Which paper you are working on
* The project Plan
* Your goals with the project
* A section for each analysis you run, including methods results and a short discussion.
* Any general throughts and discussion about the project.
* Suggestion: daily log of what you’ve done.

**UPPMAX**It’s the UU center for high-performance computing. They have multiple computer clusters specialized for different tasks. We’re going to use one, Rackham, that are using Linux. Suggest looking up some basic bash scripting, it will help a lot. It will make scripts easier to debug. Tab-completion in the terminal.

Rackham is a computer cluster: a collection of small computers that are connected and have access to the same storage. You connect to the server using something called SSH. When you connect you will connect to a log-in node. A node that I shared by everyone. You can’t run any big jobs on that node. We then use worker-node when running something. We can specify if we want to use a whole node or single cores within the node. We will only request cores for this project. If we use 2 cores for one hour that is equal to 2 h processing.

Two ways to request time

* Salloc: get into interactive mode. You get sent from log in-node to worker-node. Then you keep working in terminal. If something goes wrong you see that immideialty and can fix it.
* Sbatch: get into batch mode. You sent scripts to the queue and that will be run once they reach top of the queue. It automatically save error messages: error log. See where it went wrong. As soon sas the script finishes it stops using processing hours, more time efficient than salloc.

We should use interactive session to test commands. When you know you have correct input parameters – put in script and run everything in batch mode. You have to put cod ein github – easier to manage it if it’s in one script.

You can check a scripts progress using squeue. We can see when they started, how long they have been running for and estimated waitining time until they start. Can you scancel to stop a script from running.

Rackham have a lot of softwares installed – so much they have to split up into modules. This way you have to load softwares manually using module load-command. If you load a module and it doesn’t work – probably named something slightly different. Use module spider in that case, search function, you then usually find the right module.

Example script  
First all the input parameters. Once the time runs out the script will be terminated. If the work is not finished by then it is lost – add extra time. Can find estimated of time in lab manual + add 20% of time for a buffer.

Then you add the modules you need. F.ex. bioinfo-tools is one, containing other modules.

Each user has 32GB of storage on Rackham. Need to be careful how much space you use. Don’t use multiple copies of one file – don’t want redundant information. Use instead symbolic links (linux introduction).

Before sending a script to the queue: check that everything looks correct. One thing is how you load modules and make sure that all softwares you’re running are also using all cores you want. Must specify that in some cases. Find in manual of softwares how many cores it uses.

Use interactive mode first to check that your commands are working first.

Look into UPPMAX user guides and FAQs

**Seminar 1**

**Data management plan**

Know from start of project how to organize your files. Know how much space the files that we produce will take up. Naming scheme we use for the files. It should be consistent throughout the project. Decide that before we start the project. Uppmax requires you to have something like that.

Is it only the softwares in appendix 2 in the manual that we should perform or are these just some of them? (If we only do the ones that are for grade 3).

The softwares in the manual are recommended softwares. It should be everything that you need for the labs. But there are other alternatives.

How can you tell that you have reach limit of 32gb storage? Will you just now start the job or will it through an error?

Jobs would start normally but would throw an error if we reach limit. A command uquota will let you know how much storage you have.

Time limit on how long we can access UPPMAX? Maximum time is 10 days to run. Time limit for the whole course.

How many cores can we get from UPPMAX? Each node has max 20 cores. But most things we run have like 2-4 cores. During the scheduled lab sessions they have booked cores. It’s enough for 2 cores per person. At least one analyses requires 3 cores.

Going back to the first question regarding the data management plan, you said that we should know how much space the files that we produce will take up? Is there a way to know this before starting the project? If we should include this already in the project plan.

It’s really hard the files will take up. It depends on which project and softwares you’re using for the different analyses. The most important is that you know how you should structure the files in the project. Most important thing. That we can include in the project plan.

question regarding soft links: you recommended that instead of copying files, we should make soft links. But how would soft links prevent us from messing the original data?

Didn’t catch the answer but look soft links up.

The appendix 2 in the manual says for example that paper 5 should run Trinity but I did not read about Trinity in the article (paper 5). Is it just that we will use other softwares than they did or did I miss a part of the article? Or have I misinterpreted something?

Yes, you can use different softwares from the paper. You can use different softwares for the same purposes.