Biological data analysis

Single cell data analysis

Juan D. Montenegro

January 8th, 2024

Structure of the course

- Lecturers:
  - Alison Cole
  - Juan Montenegro
- Divided in 2 parts:
- First part
  - 1st week: Linux environment, basic commands and pipelines
  - 2nd week: Generation and assessment of mapping tool, read mapping
- Second part
  - 3rd week: Use of Seurat for basic single cell analysis
  - 4th week: Seurat for basic single cell analysis con’t.

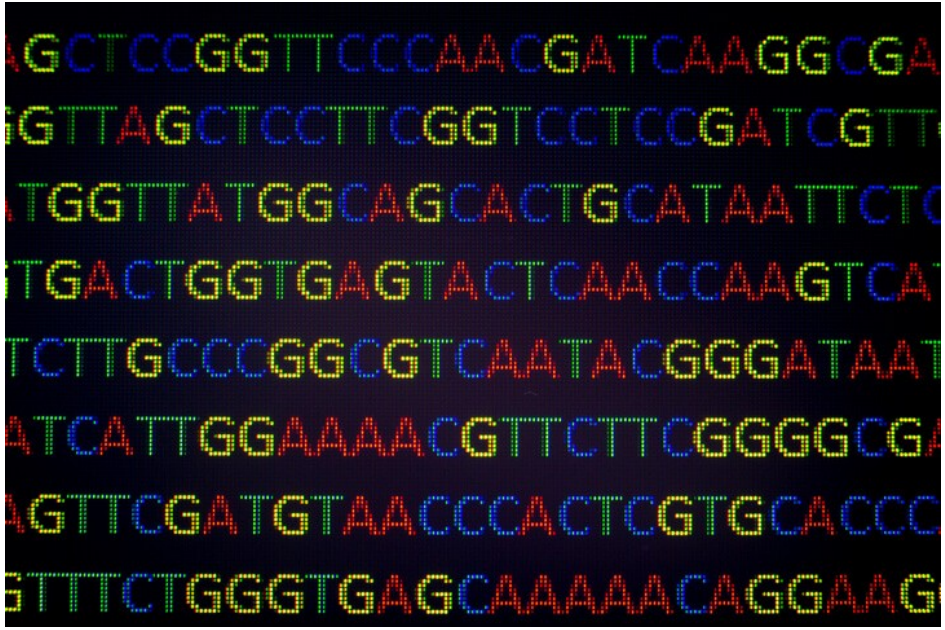
Week 1

Main objective:

- Be able to generate and assess a mapping tool that can be reliably used for downstream analysis.

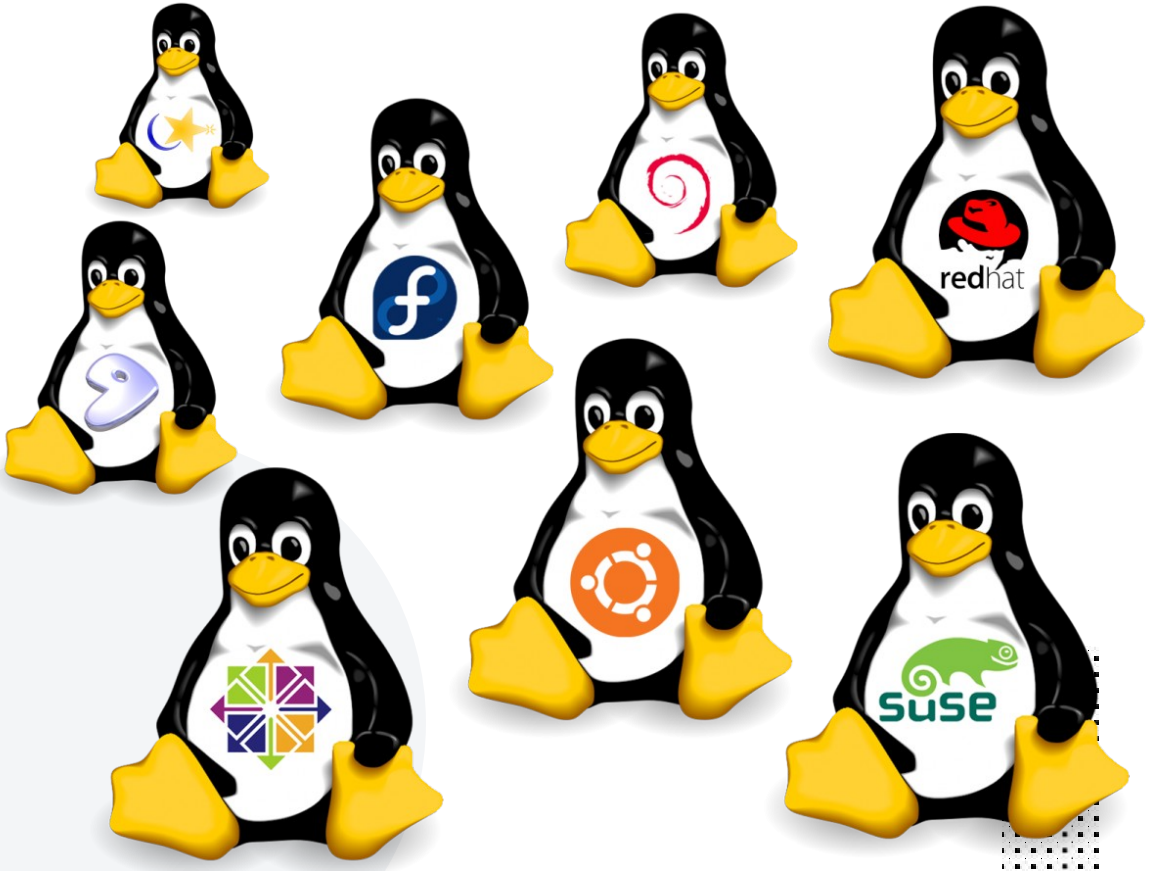
Secondary objectives:

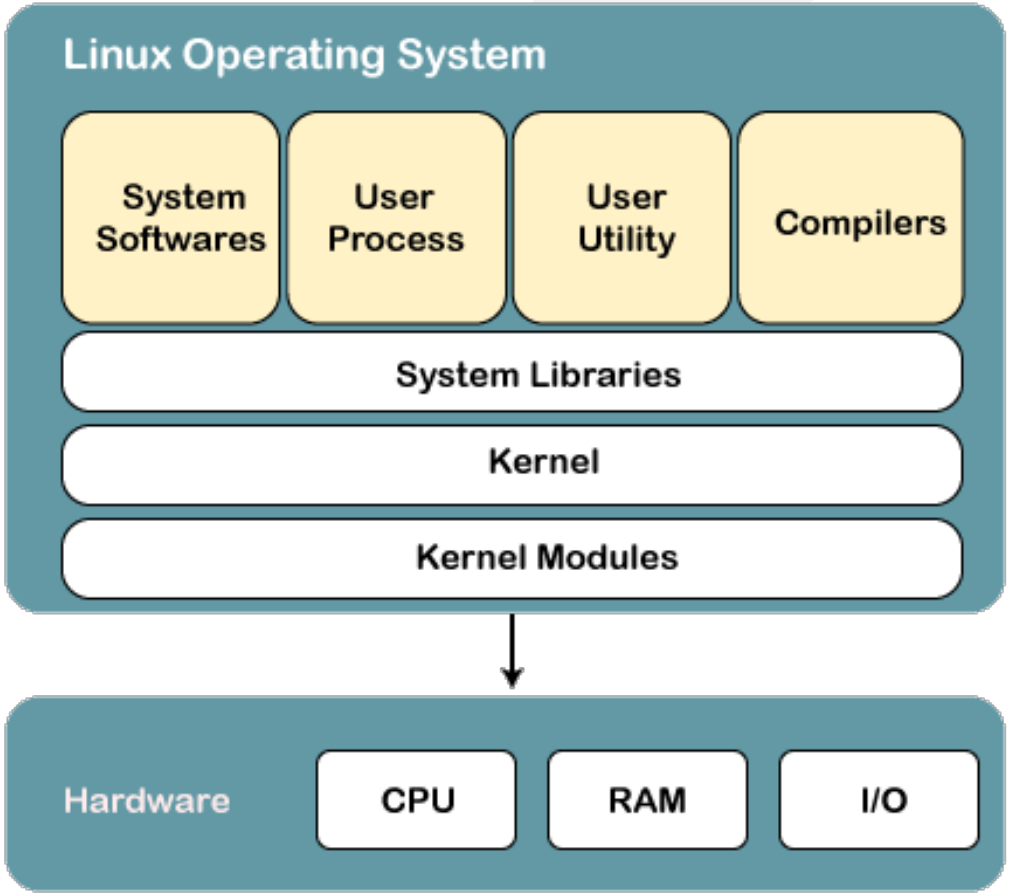
- Get familiar using the linux command line
- Basic bash scripting and slurm
- Align reads to your mapping tool.



Why Linux?

- Open source
- Stable
- Lightweight
- Flexible
- High performance




Linux Operating System
System Softwares
User Process
User Utility
Compilers
System Libraries
Kernel
Kernel Modules
Hardware
CPU
RAM
I/O

Supercomputing

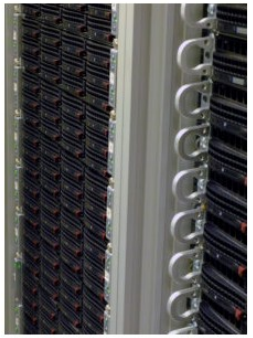
universität wien

Life Science Compute Cluster

Home | About | Get access | Troubleshooting | Statistics | Contact

Welcome to the Life Science Compute Cluster

The Life Science Compute Cluster (LiSC) is a specialised high-performance computing infrastructure for bioinformatics and computational life science. The main difference to larger, generic computing facilities, such as the Vienna Scientific Cluster (VSC) is the equipment with a rich, flexible and up-to-date bioinformatics software repository and the availability of major biological databases on-site. This installation allows typical users to analyse their data without any software installation, just by using the pre-installed tools and databases.



LiSC data storage units

Three organisational units of the University of Vienna

Recent Posts

First exercise

- How to connect to a server:
  - Username
  - Password
  - IP adress
  - Open a secure shell channel (MobaXTerm)

The LiSC folder structure

/	→	Root directory
/home	→	where all users and programs are
/home/user	→	where you login
/home/apps	→	where programs are installed
/archive	→	where to store old data and results
/scratch	→	where to work!!

Second exercise

- Find out where in the server you are “pwd”

```
pwd
```

- Change to another directory “cd”

```
cd /scratch/molevo
```

- List the contents of the current directory “ls”

```
ls
```

- Create your personal directory “mkdir”

```
mkdir <USERNAME>
```

LiSC applications

```
jmontenegro: /home/user/jmontenegro> module avail
------------------------------------------------------------------ /home/apps/modulefiles/visualisation ------------------------------------------------------------------
clinker/0.0.27-3.11.1  igv/2.7.2  intervene/0.6.5-3.7.16  intervene/0.6.5-3.8.16  intervene/0.6.5-3.9.16  intervene/0.6.5-3.10.9

------------------------------------------------------------------ /home/apps/modulefiles/system ------------------------------------------------------------------
conda/miniconda3  lisc/default  neurobiology/default  slurm/22.05.6  slurm/22.05.7

------------------------------------------------------------------ /home/apps/modulefiles/sequenceanalysis ------------------------------------------------------------------
admixture/1.3.0                 makeblastdb/2.9.0                                                                                                                tigmint/1.1.2
ancibd/0.2a2-3.10.9             mash/2.2                                                                                                                         tmhmm/2.0c
aragorn/1.2.41                  mash/2.3                                                                                                                         trimal/1.4.1
arb/19270                       meme/5.5.0                                                                                                                       trnascan/2.0.12
better-fasta-grep/1.0.3-3.11.1  mmseqs2/11-e1a1c                                                                                                                 ucsc/v391
bioawk/1.0                      mummer/4.0.0rc1                                                                                                                  usearch/5.2.236
bismark/0.24.0                  muscle/3.8.31                                                                                                                    usearch/11.0.667
blat/35.1                       ncbiblastplus/2.5.0                                                                                                              vsearch/2.14.1
cdhit/4.8.1                     ncbiblastplus/2.6.0
clustalomega/1.2.4              ncbiblastplus/2.7.1
clustalw/2.1                    ncbiblastplus/2.8.1
```

LiSC conda environments

```
jmontenegro@login01:~
jmontenegro: /home/user/jmontenegro> module load conda
jmontenegro: /home/user/jmontenegro> conda info --envs
# conda environments:
#
base                  *  /home/apps/conda/miniconda3
abyss-2.3.5              /home/apps/conda/miniconda3/envs/abyss-2.3.5
adapterremoval-2.3.3     /home/apps/conda/miniconda3/envs/adapterremoval-2.3.3
agat-1.0.0               /home/apps/conda/miniconda3/envs/agat-1.0.0
antismash-6.1.1          /home/apps/conda/miniconda3/envs/antismash-6.1.1
anvio-7.1                /home/apps/conda/miniconda3/envs/anvio-7.1
arcs-1.2.4               /home/apps/conda/miniconda3/envs/arcs-1.2.4
arcs-1.2.5               /home/apps/conda/miniconda3/envs/arcs-1.2.5
augustus-3.5.0           /home/apps/conda/miniconda3/envs/augustus-3.5.0
bakta-1.6.0              /home/apps/conda/miniconda3/envs/bakta-1.6.0
bakta-1.6.1              /home/apps/conda/miniconda3/envs/bakta-1.6.1
barrnap-0.9              /home/apps/conda/miniconda3/envs/barrnap-0.9
bbmap-39.01              /home/apps/conda/miniconda3/envs/bbmap-39.01
beast2-2.6.3             /home/apps/conda/miniconda3/envs/beast2-2.6.3
bracken-2.8              /home/apps/conda/miniconda3/envs/bracken-2.8
braker2-2.1.6            /home/apps/conda/miniconda3/envs/braker2-2.1.6
breseq-0.37.1            /home/apps/conda/miniconda3/envs/breseq-0.37.1
busco-5.4.3              /home/apps/conda/miniconda3/envs/busco-5.4.3
busco-5.4.4              /home/apps/conda/miniconda3/envs/busco-5.4.4
centrifuge-1.0.4         /home/apps/conda/miniconda3/envs/centrifuge-1.0.4
checkm-genome-1.2.2      /home/apps/conda/miniconda3/envs/checkm-genome-1.2.2
checkm2-0.1.3            /home/apps/conda/miniconda3/envs/checkm2-0.1.3
checkv-1.0.1             /home/apps/conda/miniconda3/envs/checkv-1.0.1
circos-0.69.8            /home/apps/conda/miniconda3/envs/circos-0.69.8
concoct-1.1.0            /home/apps/conda/miniconda3/envs/concoct-1.1.0
consent-2.2.2            /home/apps/conda/miniconda3/envs/consent-2.2.2
constax-2.0.18           /home/apps/conda/miniconda3/envs/constax-2.0.18
coverm-0.6.1             /home/apps/conda/miniconda3/envs/coverm-0.6.1
crest4-4.2.6             /home/apps/conda/miniconda3/envs/crest4-4.2.6
cutadapt-4.1             /home/apps/conda/miniconda3/envs/cutadapt-4.1
cutadapt-4.2             /home/apps/conda/miniconda3/envs/cutadapt-4.2
das_tool-1.1.5           /home/apps/conda/miniconda3/envs/das_tool-1.1.5
das_tool-1.1.6           /home/apps/conda/miniconda3/envs/das_tool-1.1.6
defensefinder-1.0.9      /home/apps/conda/miniconda3/envs/defensefinder-1.0.9
diamond-2.0.15           /home/apps/conda/miniconda3/envs/diamond-2.0.15
dram-1.4.3               /home/apps/conda/miniconda3/envs/dram-1.4.3
drep-3.4.0               /home/apps/conda/miniconda3/envs/drep-3.4.0
eggnog-mapper-2.1.9      /home/apps/conda/miniconda3/envs/eggnog-mapper-2.1.9
emirge-0.61.1            /home/apps/conda/miniconda3/envs/emirge-0.61.1
enrichm-0.6.5            /home/apps/conda/miniconda3/envs/enrichm-0.6.5
epa-ng-0.3.8             /home/apps/conda/miniconda3/envs/epa-ng-0.3.8
etetoolkit-3.1.2         /home/apps/conda/miniconda3/envs/etetoolkit-3.1.2
exonerate-2.4.0          /home/apps/conda/miniconda3/envs/exonerate-2.4.0
fasttree-2.1.11          /home/apps/conda/miniconda3/envs/fasttree-2.1.11
fgmp-1.0.3               /home/apps/conda/miniconda3/envs/fgmp-1.0.3
flye-2.9.1               /home/apps/conda/miniconda3/envs/flye-2.9.1
gtdbtk-2.1.1             /home/apps/conda/miniconda3/envs/gtdbtk-2.1.1
hgtector-2.0b3           /home/apps/conda/miniconda3/envs/hgtector-2.0b3
```

Third exercise

- Find the blast program and load it

```
module avail blast
module load XXXXXXX
```

- Load the module conda

```
module load conda
```

- List the environments currently available in conda

```
conda info --envs
```

- Identify a program you have used before and load its environment

```
conda activate XXXXXXX
```

Fourth exercise

- Enter NCBI and find the reference genome of Nematostella vectensis
- Download the reference genome to your directory in the server
- Enter NCBI and find raw reads of RNAseq experiments and Isoseq experiments
- Download the reads to your scratch directory in LiSC.

What have we downloaded

Fasta files
Fastq files
Annotation files

Biological data formats

- Fasta
- Fastq
- GFF/GTF
- SAM

```
>AT1G09780|1|training
GTGGAGTAGAAGAATTGAGAGCCTTATCAG
TTTTTGAAGAGAGGGCTGAAACTCTCTAGT
TATCTTTTGTTGCTTTTCTAATAATAAGAG
TTTACACACAG
>AT1G31812|0|testing
TCCTCATCTGCAGTAACTTTATCTTAAGCA
TCAAAATAACATTGCATAAGACTTGTTCTT
GCTCTTGTGTTTCTATCATATTTAAGCTAT
CTACTTTGTGA
```

Part 1

Part 2

Part 3

Identifier — @HWI-EAS209_0006_FC706VJ:5:58:5894:21141#ATCACG/1

Sequence — TTAATTGGTAAATAAATCTCCTAATAGCTTAGATNTTACCTTNNNNNNNNNNTAGTTTCTTGAGA

+ sign & identifier — +HWI-EAS209_0006_FC706VJ:5:58:5894:21141#ATCACG/1

Quality scores — efcfffffcfeefffcffffffddf`feed]`]_Ba_^__[YBBBBBBBBBRTT\]]][]dddd`

Base T

phred Quality] = 29

Col 1	Col 2	Col 3	Col 4	Col 5	Col 6	Col 7	Col 8	Col 9
chr21	HAVANA	transcript	10862622	10863067	.	+	.	gene_id "ENSG00000169..
chr21	HAVANA	exon	10862622	10862667	.	+	.	gene_id "ENSG00000169..
chr21	HAVANA	CDS	10862622	10862667	.	+	0	gene_id "ENSG00000169..
chr21	HAVANA	start_codon	10862622	10862624	.	+	0	gene_id "ENSG00000169..
chr21	HAVANA	exon	10862751	10863067	.	+	.	gene_id "ENSG00000169..
chr21	HAVANA	CDS	10862751	10863064	.	+	2	gene_id "ENSG00000169..
chr21	HAVANA	stop_codon	10863065	10863067	.	+	0	gene_id "ENSG00000169..
chr21	HAVANA	UTR	10863065	10863067	.	+	.	gene_id "ENSG00000169..

SLURM

Resource manager for computing clusters
Optimizes resource allocation

Example: example.sh

```
#!/bin/bash
#SBATCH --job-name=Test                # Job name
#SBATCH --nodes=1                      # Run on a single node
#SBATCH --ntasks-per-node=1            # Run a single task on each node
#SBATCH --partition=ai                 # Run in ai queue
#SBATCH --qos=ai                       # Run in qos (ai)
#SBATCH --account=ai                   # Run account (ai)
#SBATCH --time=1:0:0                   # Time limit days-hours:minutes:seconds
#SBATCH --output=test-%j.out           # Standard output and error log
#SBATCH --mail-type=ALL                # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=foo@bar.com        # Where to send mail

module load python/3.6.1
echo "Running plot script on a single CPU core"
python /kuacc/users/username/plot_template.py
```

Fifth exercise

- Write a slurm script.

```
vim myProgram.sh
```

- Submit the slurm script to the system

```
sbatch myProgram.sh
```

- Check the status of your program

```
squeue -j <JobId>
squeue -u $USER
```

- Check the log and error files

```
cd /scratch/students/<USER>/SCCourse/logs
less myProgram_<jobId>.log
less myProgram_<jobId>.err
```

- Check your results

Why do we align?

To measure the level of similarity between two sequences.
To rank and cluster sequences depending on their similarity value.

01 Index

Indexing a reference genome is essential to speed up the lookup and alignment process.

02 QC reads

Remove low quality reads and adapter sequences to improve mapping efficiency and accuracy.

03 Alignment

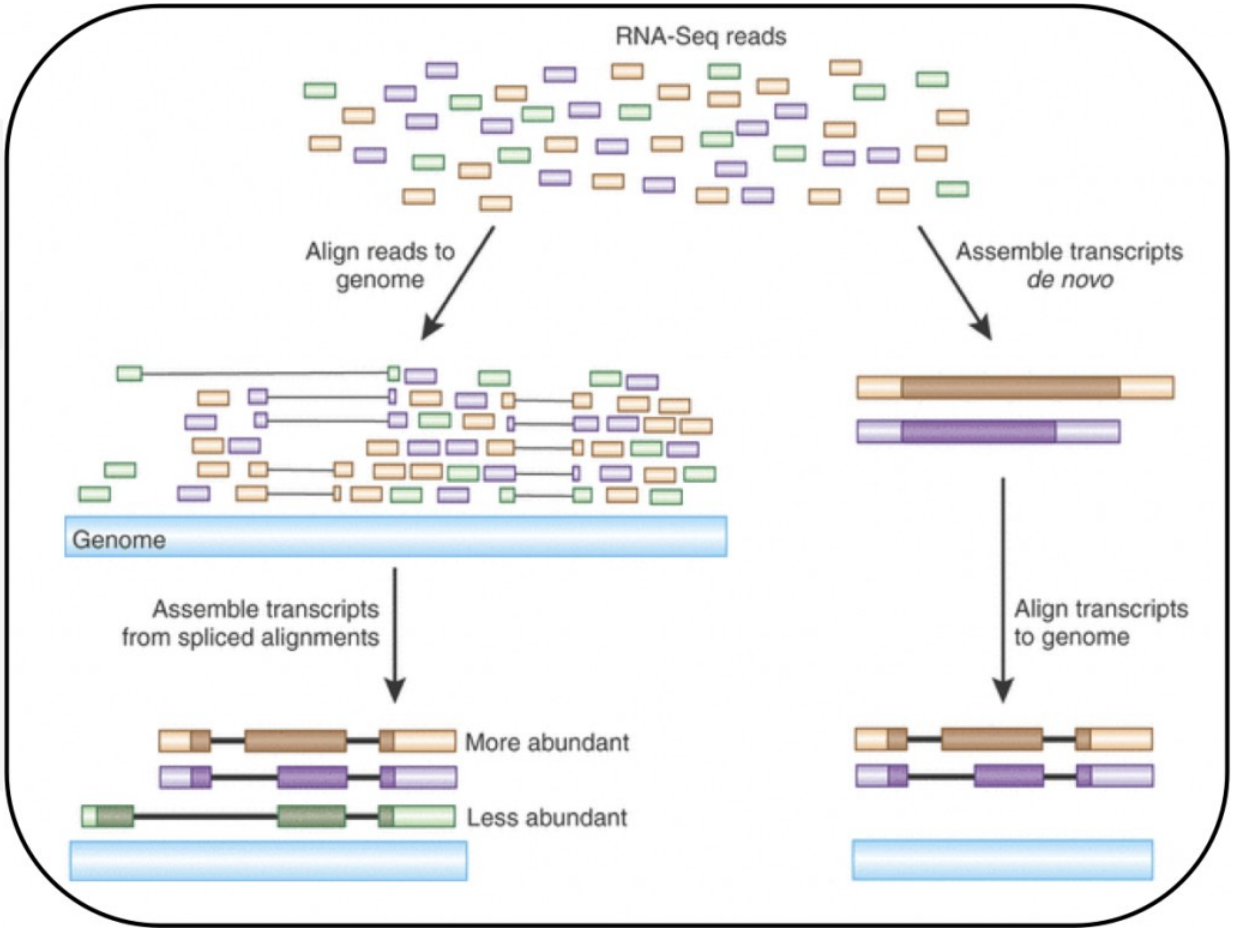
Different reads will require different alignment algorithms:
Short reads → BWT
Long reads → minimizers
Spliced reads → gap-aware alignment

04 QC alignment

Removes poorly mapping reads, mapping artifacts and only retains alignments that are informative.

Tools to align?

	Ungapped	Gapped
Short sequences	BWA Bowtie1 Bowtie2	Star TopHat2 HiSat2
Long sequences	Minimap2 LastZ Mummer	Minimap2 GMAP



Nature Biotechnology. Haas BJ and Zody MC. Advancing RNA-seq analysis. 28:421-423, copyright 2010 (10).

Sixth exercise

- Deduce command to index a reference genome

```
STAR --help
```

- Deduce command to align reads to an indexed genome

```
STAR --help
```

- Write a slurm script to index and another to align reads to a genome
- Submit the script

The SAM format

A

```
                    10         20         30         40
Coor     12345678901234  5678901234567890123456789012345
ref      AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT

+r001/1        TTAGATAAAGGATA*CTG
+r002         aaaAGATAA*GGATA
+r003       gcctaAGCTAA
+r004                     ATAGCT.............TCAGC
-r003                            ttagctTAGGC
-r001/2                                      CAGCGGCAT
```

B

```
@HD VN:1.5 SO:coordinate
@SQ SN:ref LN:45
r001   99 ref  7 30 8M2I4M1D3M = 37  39 TTAGATAAAGGATACTG *
r002    0 ref  9 30 3S6M1P1I4M * 0   0 AAAAGATAAGGATA    *
r003    0 ref  9 30 5S6M       * 0   0 GCCTAAGCTAA       * SA:Z:ref,29,-,6H5M,17,0;
r004    0 ref 16 30 6M14N5M    * 0   0 ATAGCTTCAGC       *
r003 2064 ref 29 17 6H5M       * 0   0 TAGGC             * SA:Z:ref,9,+,5S6M,30,1;
r001  147 ref 37 30 9M         = 7 -39 CAGCGGCAT         * NM:i:1
```

Header section

QUAL (read quality; * meaning such information is not available)

Alignment section

QNAME (query template name, aka. read ID)

FLAG (indicates alignment information about the read, e.g. paired, aligned, etc.)

RNAME (reference sequence name, e.g. chromosome /transcript id)

POS (1-based position)

MAPQ (mapping quality)

CIGAR (summary of alignment, e.g. insertion, deletion)

RNEXT (reference sequence name of the primary alignment of the NEXT read; for paired-end sequencing, NEXT read is the paired read; corresponding to the RNAME column)

PNEXT (Position of the primary alignment of the NEXT read in the template; corresponding to the POS column)

TLEN (the number of bases covered by the reads from the same fragment. In this particular case, it's 45 - 7 + 1 = 39 as highlighted in Panel A). Sign: plus for leftmost read, and minus for rightmost read

SEQ (read sequence)

Optional fields in the format of TAG:TYPE:VALUE

README.md

samtools

build passing | build passing | downloads 1.1M

This is the official development repository for samtools.

The original samtools package has been split into three separate but tightly coordinated projects:

- htslib: C-library for handling high-throughput sequencing data
- samtools: mpileup and other tools for handling SAM, BAM, CRAM
- bcftools: calling and other tools for handling VCF, BCF

See also http://github.com/samtools/

Building Samtools

See INSTALL for complete details. Release tarballs contain generated files that have not been committed to this repository, so building the code from a Git repository requires extra steps:

```
autoheader            # Build config.h.in (this may generate a warning about
                      # AC_CONFIG_SUBDIRS - please ignore it).
autoconf -Wno-syntax  # Generate the configure script
./configure           # Needed for choosing optional functionality
make
make install
```

By default, this will build against an HTSlib source tree in `../htslib`. You can alter this to a source tree elsewhere or to a previously-installed HTSlib by configuring with `--with-htslib=DIR`.

Citing

Please cite this paper when using SAMtools for your publications.

> Twelve years of SAMtools and BCFtools
> Petr Danecek, James K Bonfield, Jennifer Liddle, John Marshall, Valeriu Ohan, Martin O Pollard, Andrew

http://www.htslib.org/doc/samtools.html

Seventh exercise

- Find the tool samtools in the LiSC

```
module avail samtools
module load samtools
```

- Use samtools to compress and sort the alignments:

```
samtools view -b -o <outBAM> <inSAM>
samtools sort -o <sortedBAM> <outBAM>
```

- Filter the alignments

```
samtools view -b -f 2 -q 20 -o <filteredBAM> <sortedBAM>
```

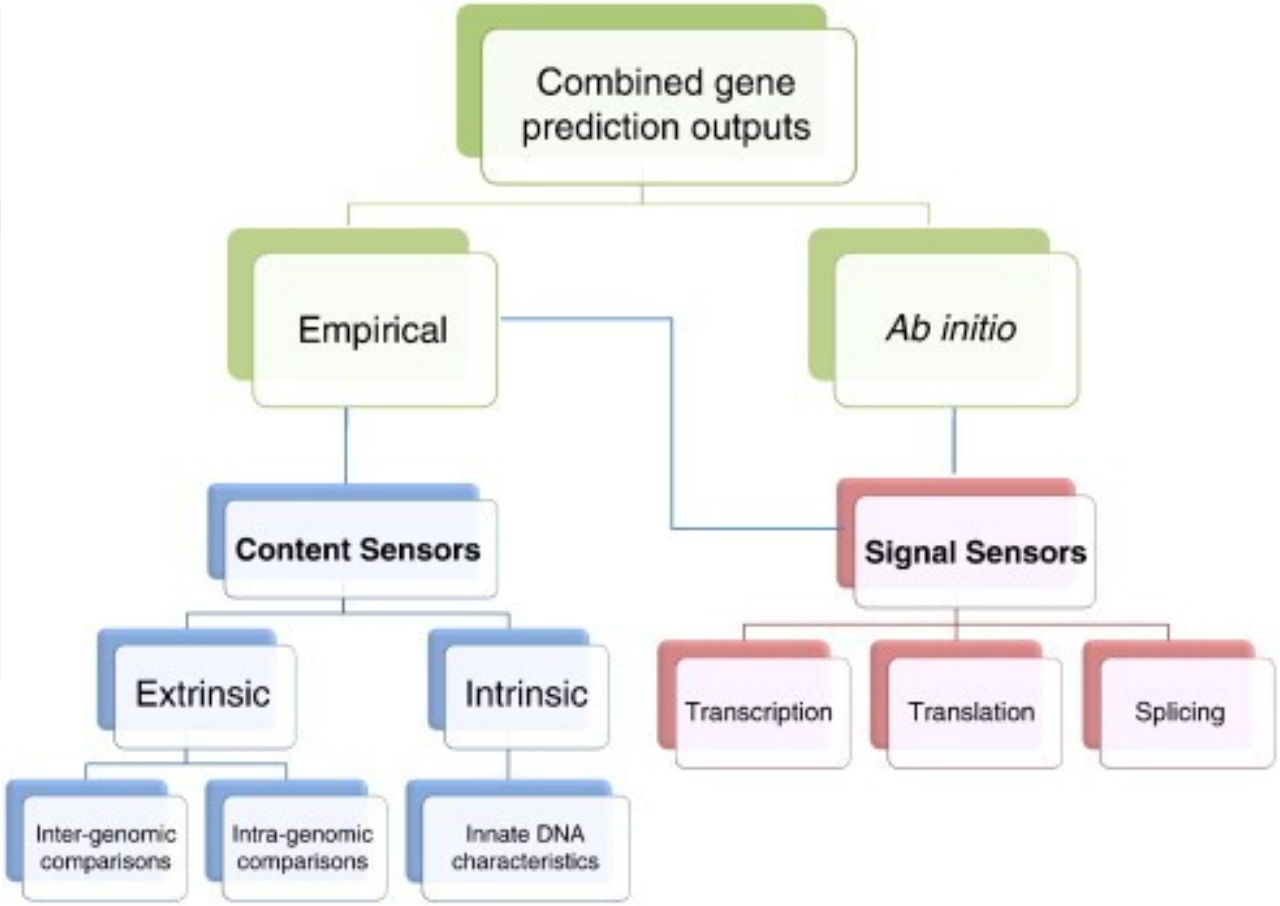
- Merge the alignments

```
samtools merge -o <mergedBAM> fBAM1 fBAM2 … fBAMN
```

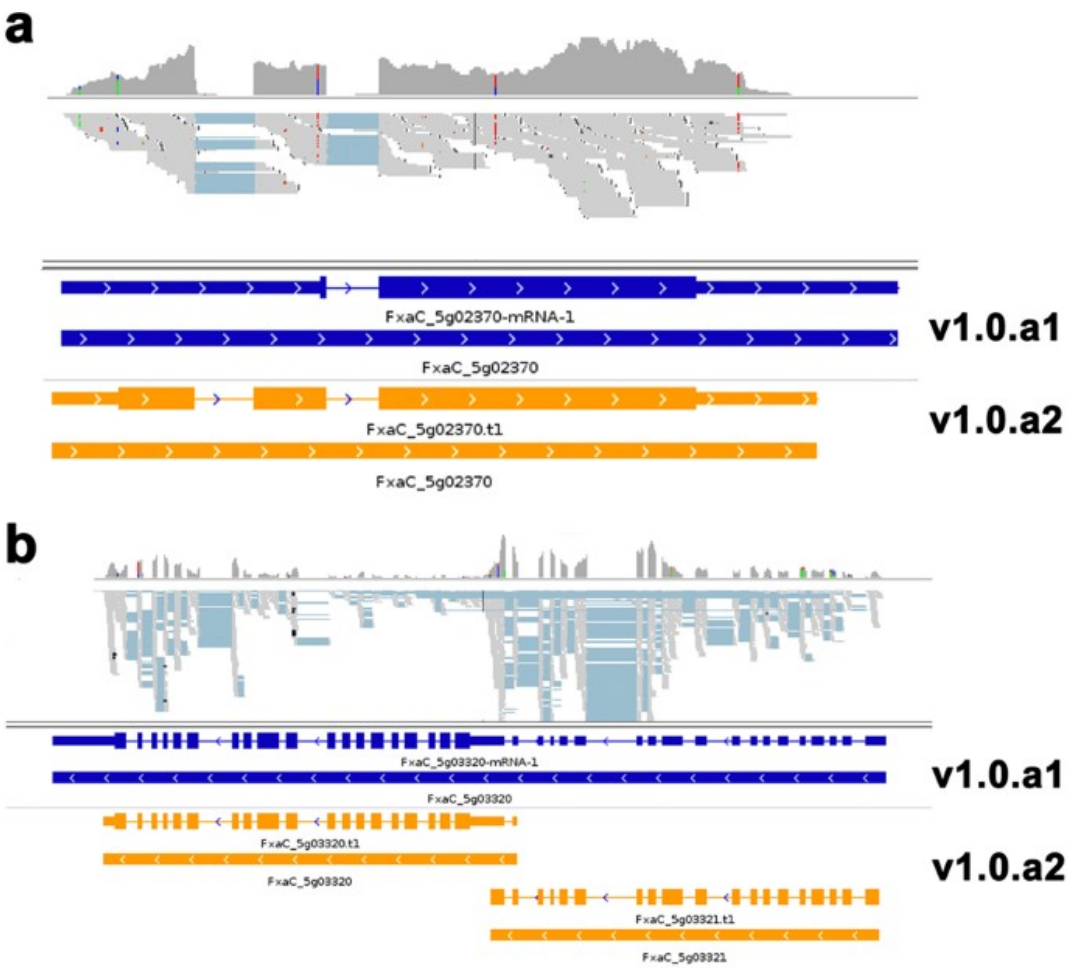
Gene prediction / Genome annotation

Process of identifying functional elements in the sequence of the genome

Strategies for gene prediction

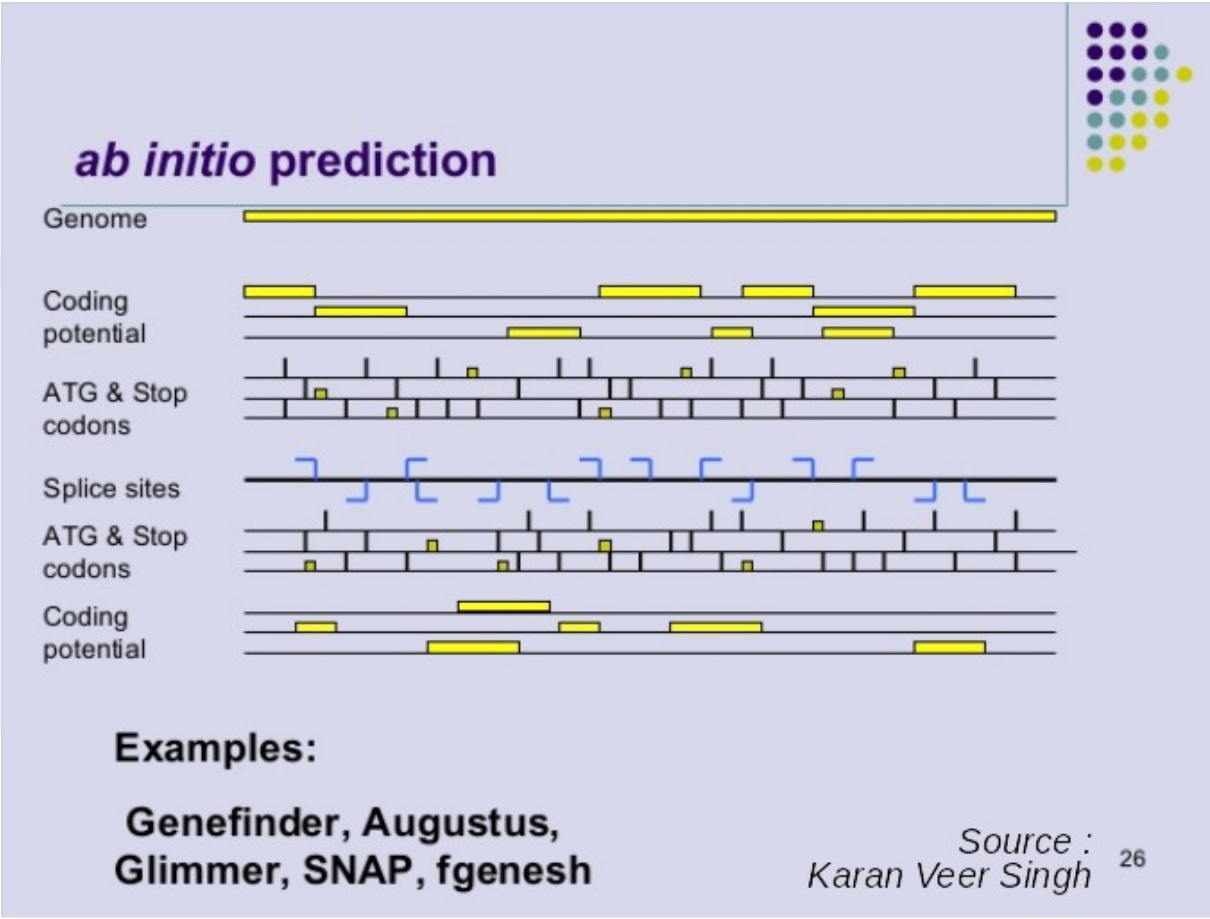


Evidence based gene prediction

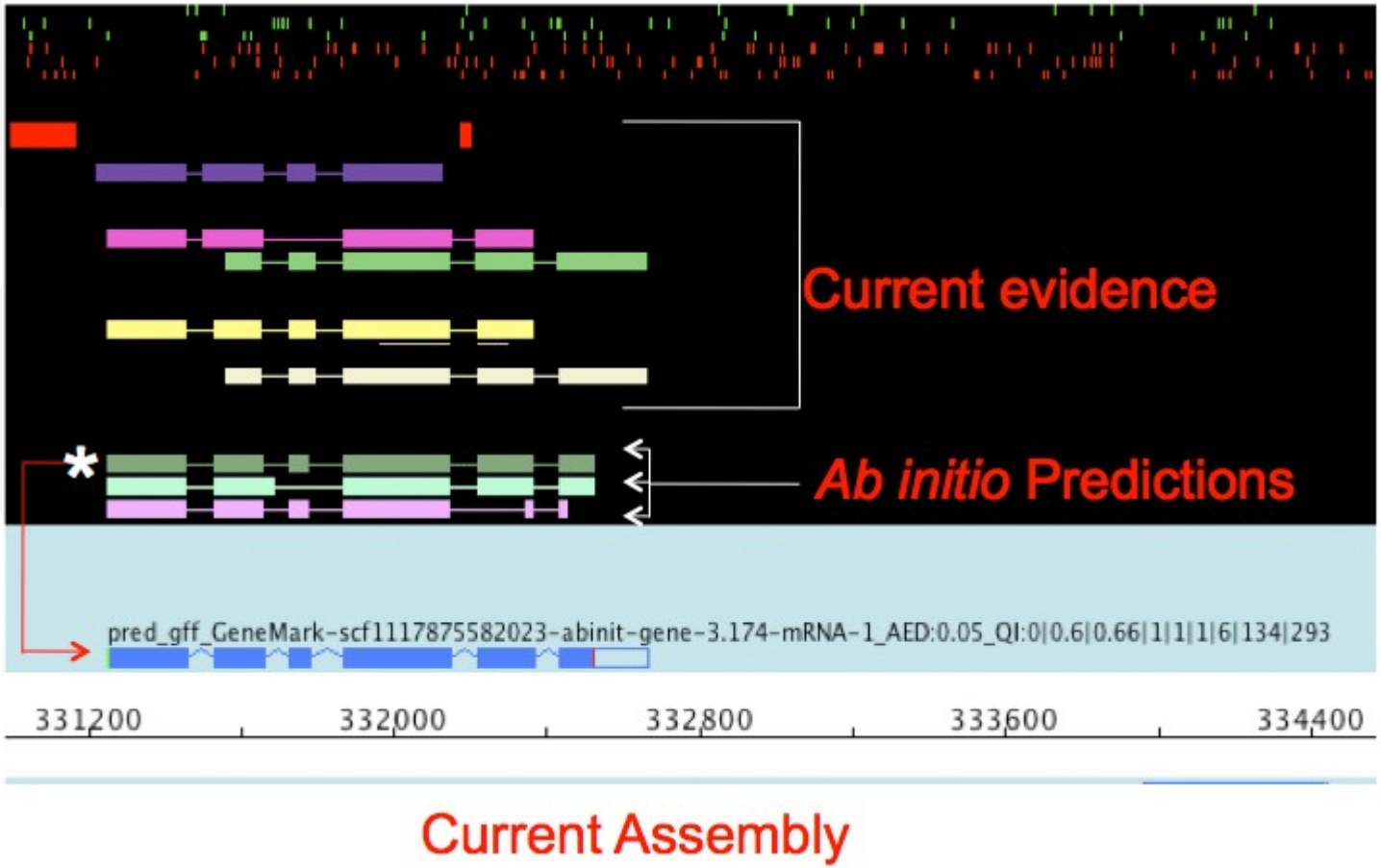


Adapted from Liu et al. (2021)

Ab initio gene prediction



Data reconciliation



Seventh exercise: Filter your alignments

- Load samtools in the LiSC

```
module load samtools
```

- Filter the alignments

```
samtools view -b -f 2 -q 20 -o <filteredBAM> <sortedBAM>
```

Eighth exercise: Evidence based Gene Annotation

- Find and load “stringtie in the LiSC”

  ```
  module avail ????
  conda activate ????
  ```

- Deduce a command line to execute stringtie

  ```
  stringtie --help
  ```

- Write a slurm script with the command line deduced by you
- Submit the slurm script and upload it to your github