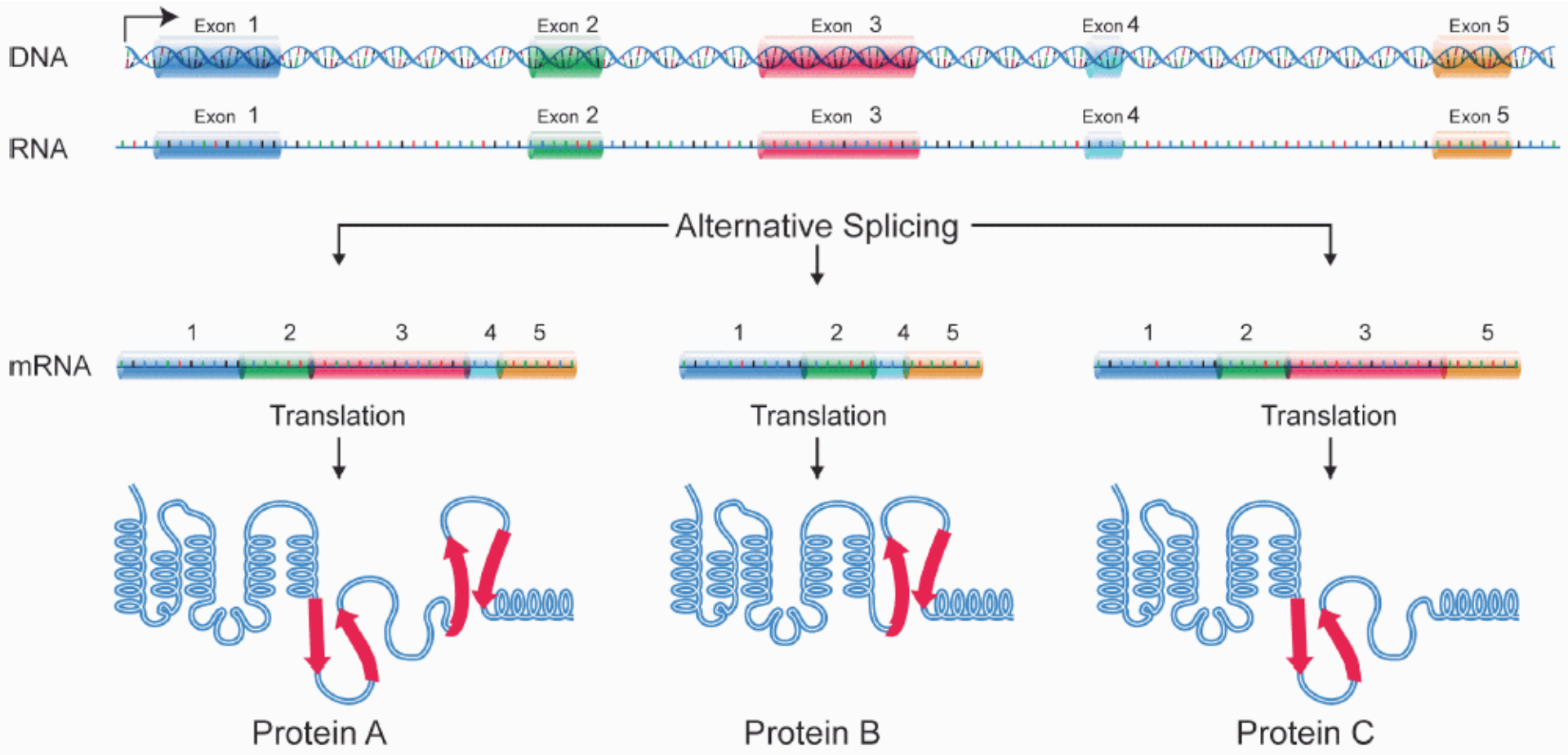
RNA-seq data analysis

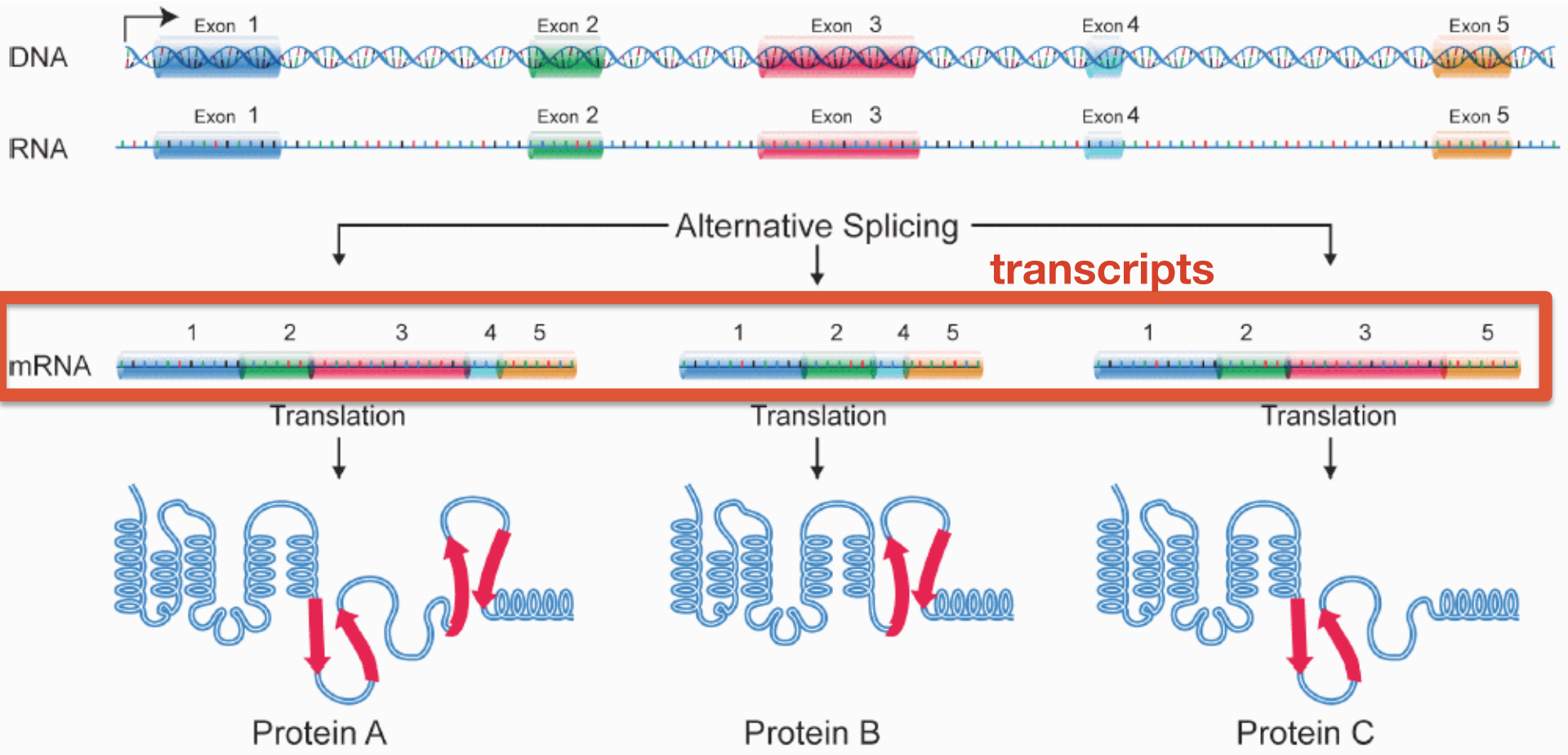
Katharina Hembach

STA426 - 11.11.2019

INTRODUCTION: TRANSCRIPTION & FILE FORMATS


DNA
Exon 1
Exon 2
Exon 3
Exon 4
Exon 5
RNA
Exon 1
Exon 2
Exon 3
Exon 4
Exon 5
Alternative Splicing
mRNA
1 2 3 4 5
1 2 4 5
1 2 3 5
Translation
Translation
Translation
Protein A
Protein B
Protein C

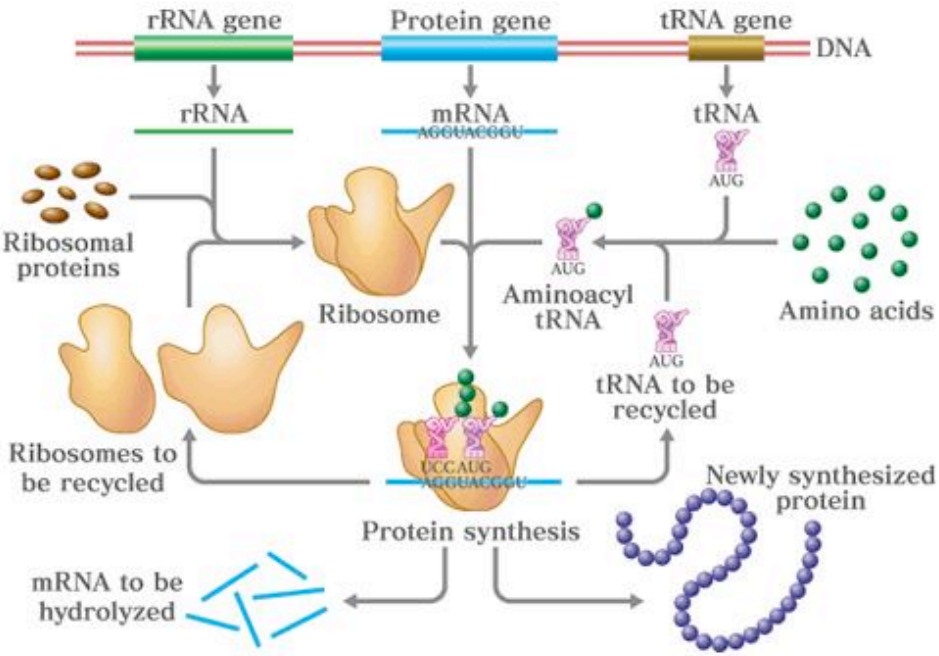
wikipedia


DNA
Exon 1
Exon 2
Exon 3
Exon 4
Exon 5
RNA
Exon 1
Exon 2
Exon 3
Exon 4
Exon 5
Alternative Splicing
transcripts
mRNA
1 2 3 4 5
1 2 4 5
1 2 3 5
Translation
Translation
Translation
Protein A
Protein B
Protein C

wikipedia

Different types of RNA

- messenger RNA
- ribosomal RNA
- transfer RNA



https://step1.medbullets.com/biochemistry/102014/rna-types

Raw data: FASTQ file

- sequence & base quality (phred score)

```
@SRR2002765.2 HWI-D00522:40:HBA10ADXX:1:1101:1788:2212 length=100
CCCATATTTACCAATCCCATGAAGCTCAATTGGATACTTCCACTGCTTTGTCAGGTATTCATCTGAGAACTTGACAATGGTTTTTGCCCGAAGATCGTAG
+SRR2002765.2 HWI-D00522:40:HBA10ADXX:1:1101:1788:2212 length=100
CCCFFFFFHHHHHJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJBGHIJJJJJJJJJJJJJJJJJJJJHIHHHFFFFDDDDDDDDDDB
@SRR2002765.3 HWI-D00522:40:HBA10ADXX:1:1101:1955:2246 length=100
CAAAAATCTAAAATAGATTTTCATAATAAAGTCTATTTGCTCATAATGGTTGGAAATGGGTTACTAAATTTGAAGAAGTTAAATATATGCAACATTGAAA
+SRR2002765.3 HWI-D00522:40:HBA10ADXX:1:1101:1955:2246 length=100
CCCFFFFFHHHHHJJJJJJJJJJJJJJJJJHIJJJJJJJJJJJJJJJJGIJJIJJJJJJIIJJJJJJJJJJJJJJJJJJHHHHHHHHFFFFFFEEEEEE
@SRR2002765.4 HWI-D00522:40:HBA10ADXX:1:1101:2246:2230 length=100
CACCGCTGCACTCCAGCCTGGGCGACAGAGCAAGACTCCGTCTCAAAAAAAAAAAAAAAAAGTCCTTTTAGCACCTTTTTGGGGAAAAAAAAAAAAAAGA
+SRR2002765.4 HWI-D00522:40:HBA10ADXX:1:1101:2246:2230 length=100
CCCFFFADHHHFHIJIJJIIIJJJIJJJJJJJJJJIJIIIHHIJJJJJJHFDDDDDDDDD;&+3:>C(:((((+29@C(4&)59(58<()09@DB3&&&(
```

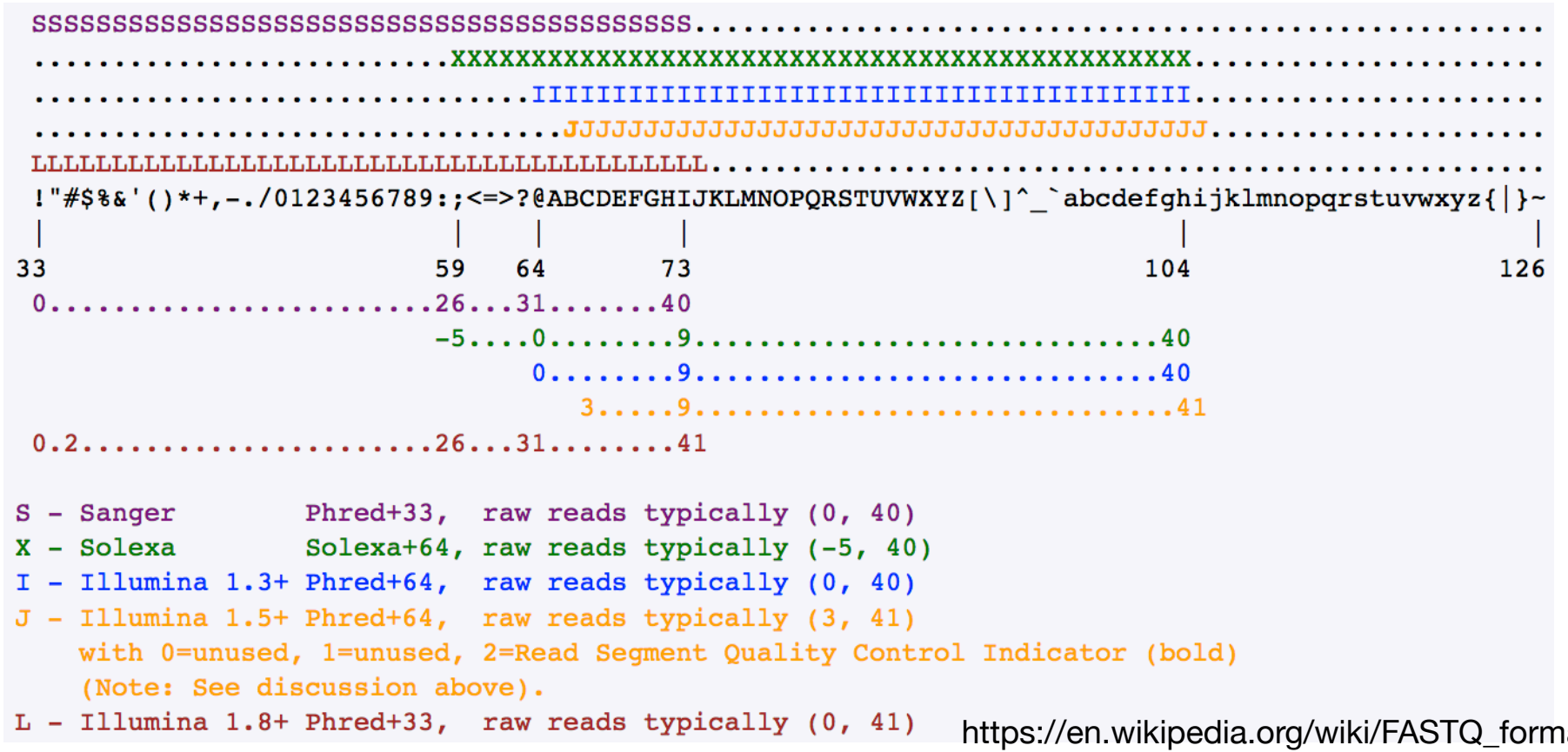
Raw data: FASTQ file

- sequence & base quality (phred score)
- 4 lines per read:

  1: @ Identifier
  2: raw sequence
  3: + (Identifier)
  4: encoded quality

```
@SRR2002765.2 HWI-D00522:40:HBA10ADXX:1:1101:1788:2212 length=100
CCCATATTTACCAATCCCATGAAGCTCAATTGGATACTTCCACTGCTTTGTCAGGTATTCATCTGAGAACTT
+SRR2002765.2 HWI-D00522:40:HBA10ADXX:1:1101:1788:2212 length=100
CCCFFFFFHHHHHJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJBGHIJJJJJJJJJJJJ
```

Quality encoding



https://en.wikipedia.org/wiki/FASTQ_format

Reference files

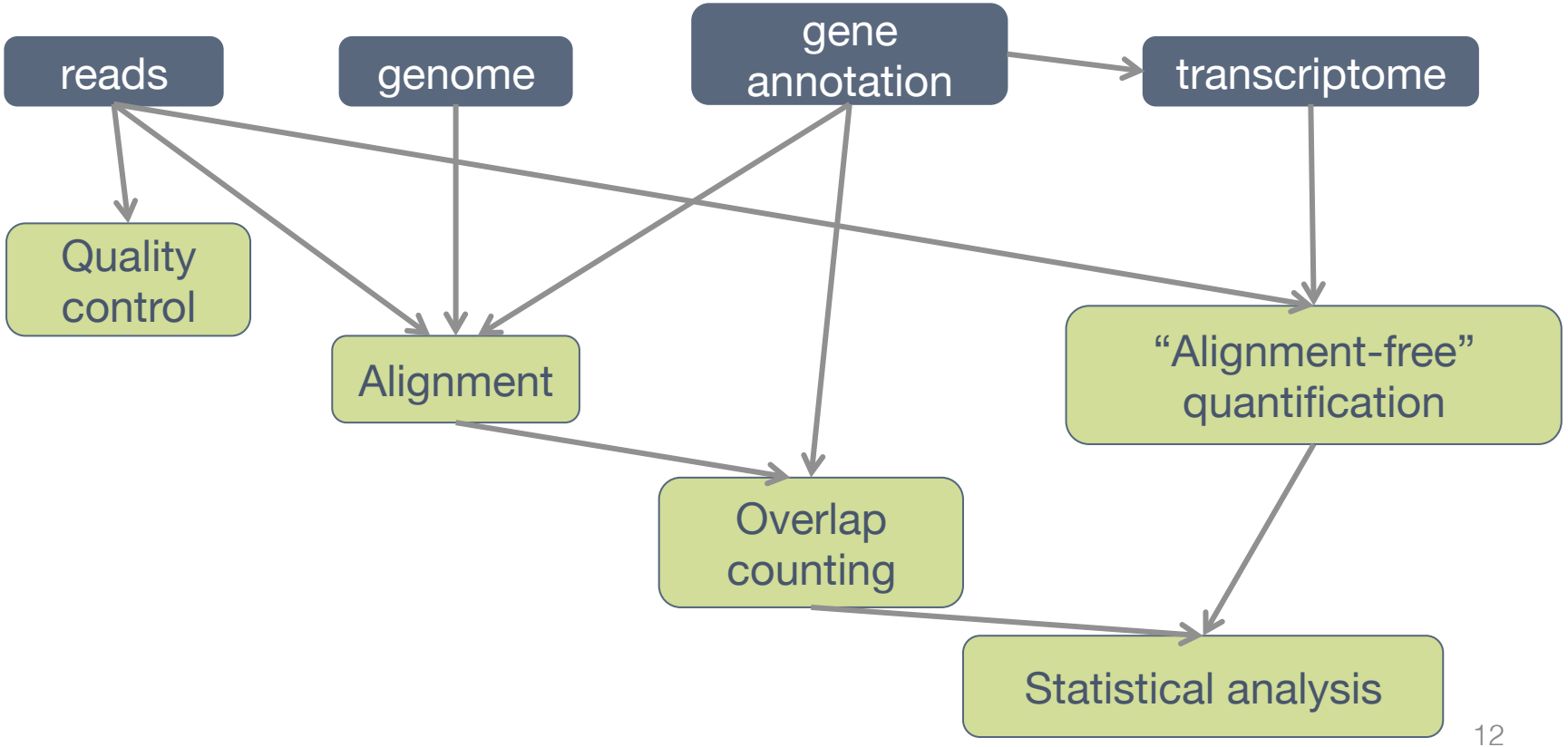
- genome and transcriptome sequences
- Different resources:
  - Ensembl (https://www.ensembl.org/info/data/ftp/index.html)
  - Gencode (https://www.gencodegenes.org/)
  - UCSC (https://www.gencodegenes.org/)
  - RefSeq (https://www.ncbi.nlm.nih.gov/refseq/about/human/)
  - …
- Different versions and builds available! Keep track and do not mix them up!

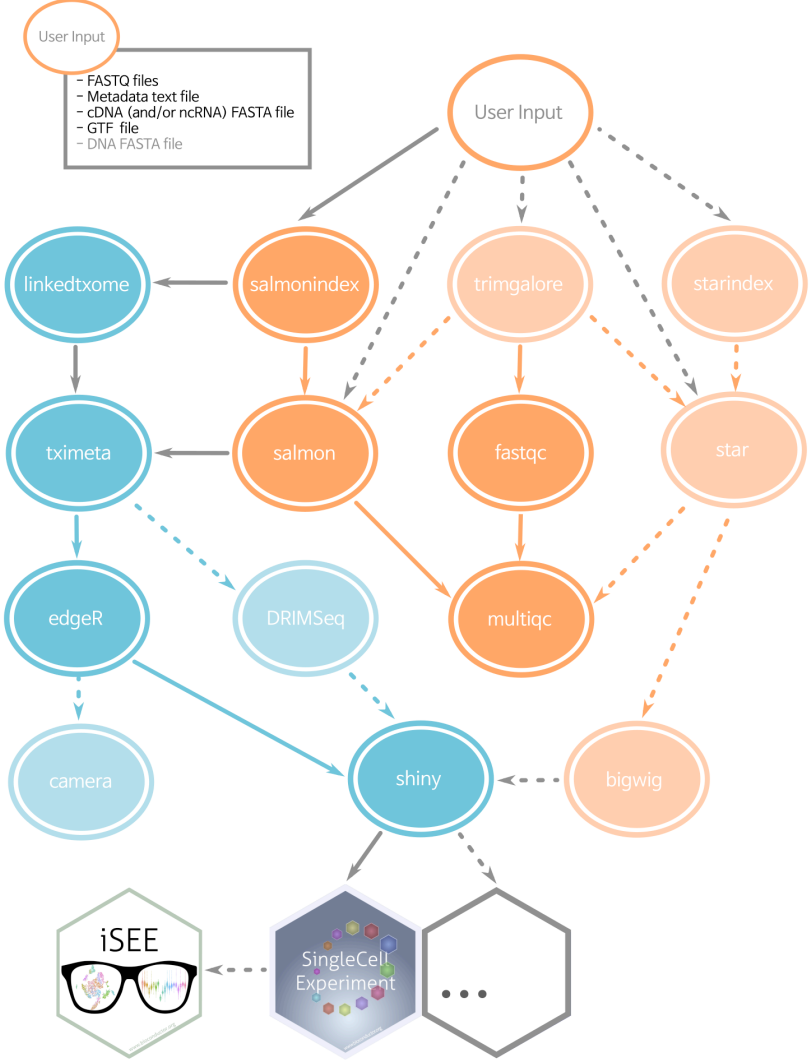
Gene annotation

- GTF/GFF format:
  - http://mblab.wustl.edu/GTF22.html
  - https://github.com/The-Sequence-Ontology/Specifications/blob/master/gff3.md
- chromosome, location, strand, type, attributes
- one line per element:
  - gene, transcript, CDS, exon, …

RNA-SEQ ANALYSIS PIPELINE

RNA-seq analysis pipeline




User Input
- FASTQ files
- Metadata text file
- cDNA (and/or ncRNA) FASTA file
- GTF file
- DNA FASTA file
User Input
linkedtxome
salmonindex
trimgalore
starindex
tximeta
salmon
fastqc
star
edgeR
DRIMSeq
multiqc
camera
shiny
bigwig
iSEE
SingleCell Experiment
...

Preprocessing of RNA-seq reads

- Quality filtering (remove reads with bad quality)
- Adapter trimming (remove adapter sequences)
- Alignment to reference genome
- Quantification of feature of interest (gene or transcript)

Quality control

- **FastQC** https://www.bioinformatics.babraham.ac.uk/projects/fastqc/
- **MultiQC** https://multiqc.info/
  - aggregates FastQC results from multiple samples, as well as Salmon and STAR output
- Statistics: # reads, read length
- You can check read quality, GC content, % duplicated reads, adapter contamination, …

Quality control

- Depending on FastQC result, you need to fix QC problems.
- Tools for quality filtering/adapter trimming:

  cutadapt, TrimGalore!, Trimmomatic, FASTX-toolkit, …

Alignment

Two possible approaches:

1. genome alignment: Find the most likely origin of each read on the genome.
   - create reference genome index for fast access
   - align reads to reference → BAM file
   - count number of reads overlapping with each feature (e.g. gene)
   - tools: STAR https://github.com/alexdobin/STAR or HISAT2 http://ccb.jhu.edu/software/hisat2/index.shtml

2. transcriptome mapping: “alignment-free” quantification
   - index reference transcriptome
   - quantify transcript abundance
   - summarise transcript counts to gene level (based on transcript to gene mapping)
   - tools: Salmon https://combine-lab.github.io/salmon/about/ or kallisto https://pachterlab.github.io/kallisto/about

Statistical analysis

- **Differential gene expression** analysis with edgeR or DEseq2 (see material from Mark): Which genes change in expression in different genotypes, treatments, time points, …?
- **Differential transcript usage**: Does the transcript composition of a given gene change? (DRIMseq https://bioconductor.org/packages/release/bioc/html/DRIMSeq.html)
- **Gene set analysis**: are the DE genes enriched for a specific gene annotation category? (*camera()* function from limma R package https://academic.oup.com/nar/article/40/17/e133/2411151)

HOW TO ORGANIZE YOUR SOFTWARE?

- Open source package and environment management system for any programming language.
- quickly install, run and update packages and their dependencies
- https://docs.conda.io/projects/conda/en/latest/user-guide/getting-started.html
- packages are stored on different “channels” (locations)
- you need to specify the channel(s) when installing things
- bioconda is the channel for bioinformatics software (https://bioconda.github.io/)

Conda environments

- you can manage packages/programs and their dependencies in environments
- no interaction with other environments
- easy to control package/language versions and avoid conflicts
- you can exported an environment to a YAML file (https://yaml.org/spec/1.2/spec.html) and easily share it → reproducibility!

Snakemake  +  = 

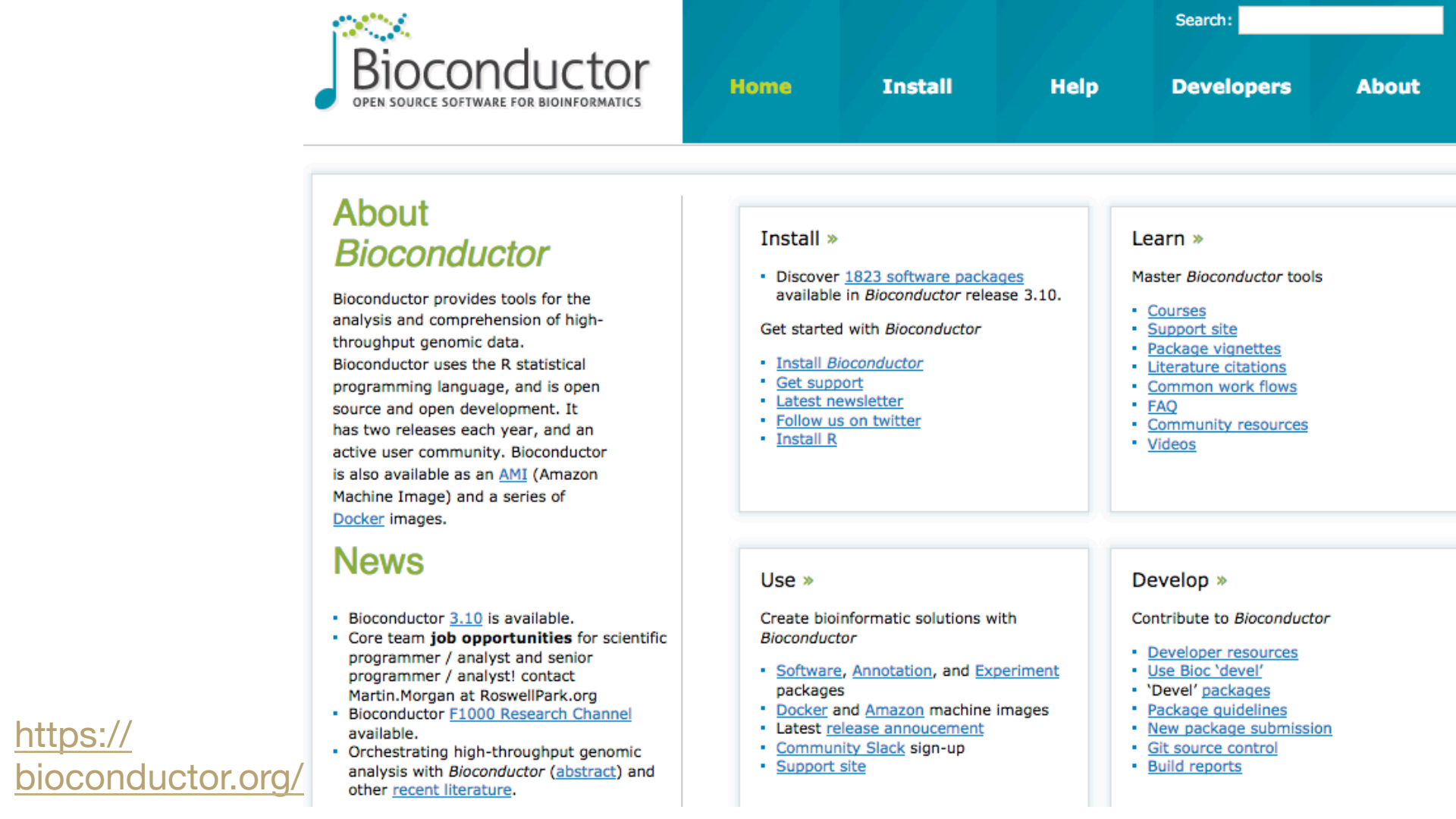
- python + GNU make = snakemake
- https://snakemake.readthedocs.io/en/stable/
- workflow management system

→ reproducible and scalable data analyses

- specify rules that describe how to create output files from input files
- file/rule dependencies are automatically determined
- rules can use shell commands, python code or external python/R scripts
- can be run on laptops, clusters, the cloud without modifications
- you can automatically deploy required software using conda or Singularity

Snakemake: useful commands

- `--help` to get detailed help message
- `--use-conda` to run rules in conda environments
- `-n` dry run → only display what would be done but do not execute anything
- `-p` print shell commands that will be executed
- `-r` print reason for each executed rule
- can be combined in `-npr`
- `-l` list all available rules
- --cores to use at most this number of cores in parallel
- --configfile path to config.yaml file (in case it is not in the same directory as the Snakefile)

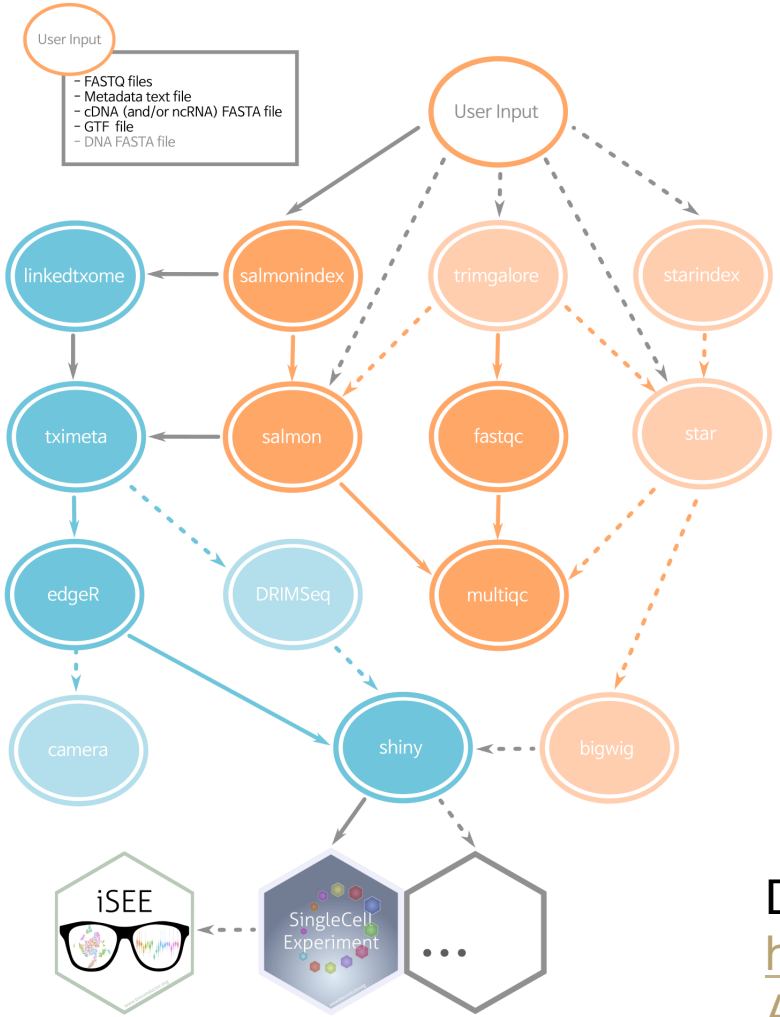


https://bioconductor.org/

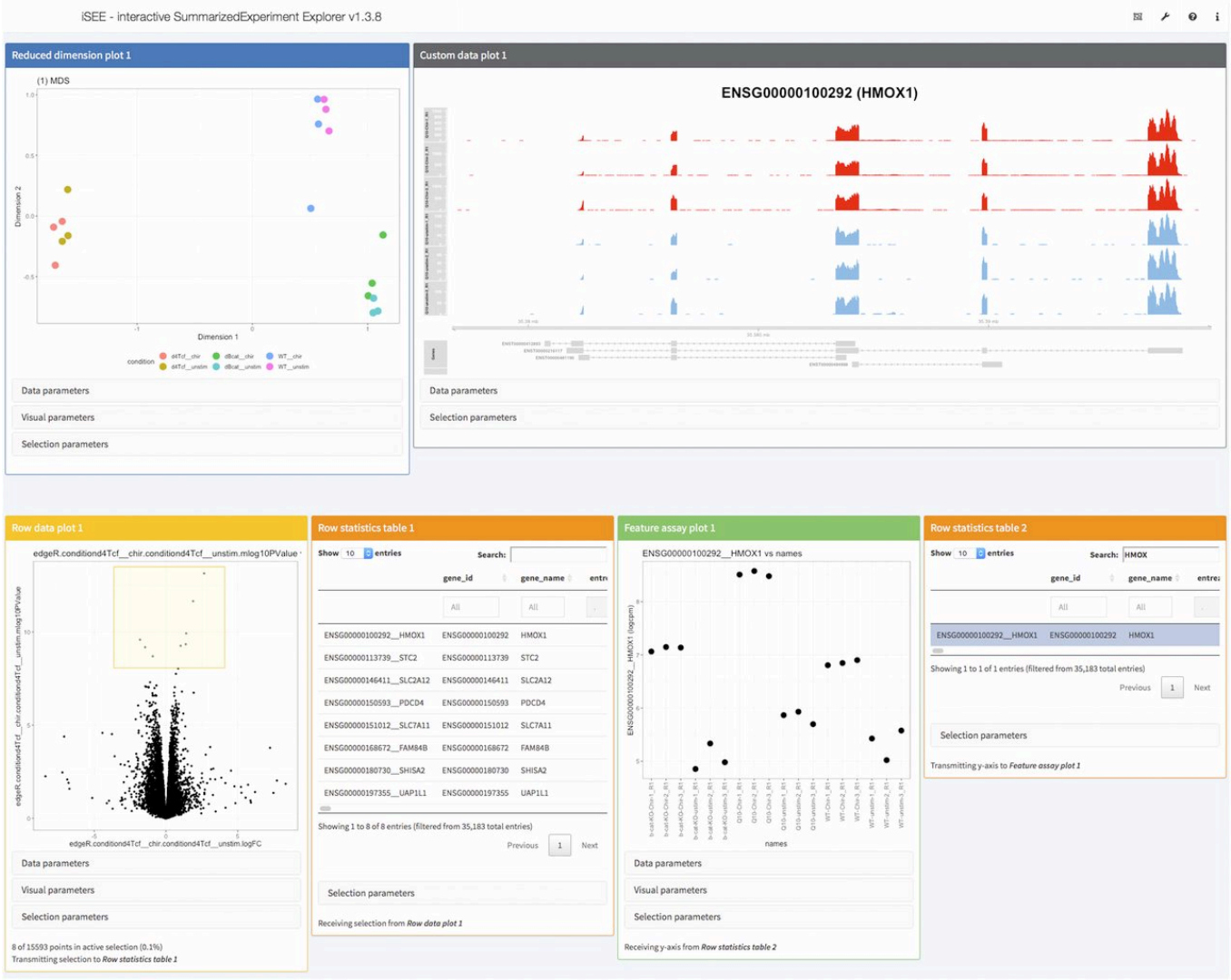
ARMOR WORKFLOW

ARMOR

- **A**utomated **R**eproducible **MO**dular **R**NA-seq
- https://github.com/csoneson/ARMOR
- https://www.g3journal.org/content/9/7/2089
- Snakemake workflow
- reproducible, automated, partially contained
- mix of command line tools and R
- Snakefile, configuration file and R scripts
- all software can be installed in conda environments
- visualization with iSEE R package → shiny app
- can be extended by adding rules to the Snakefile

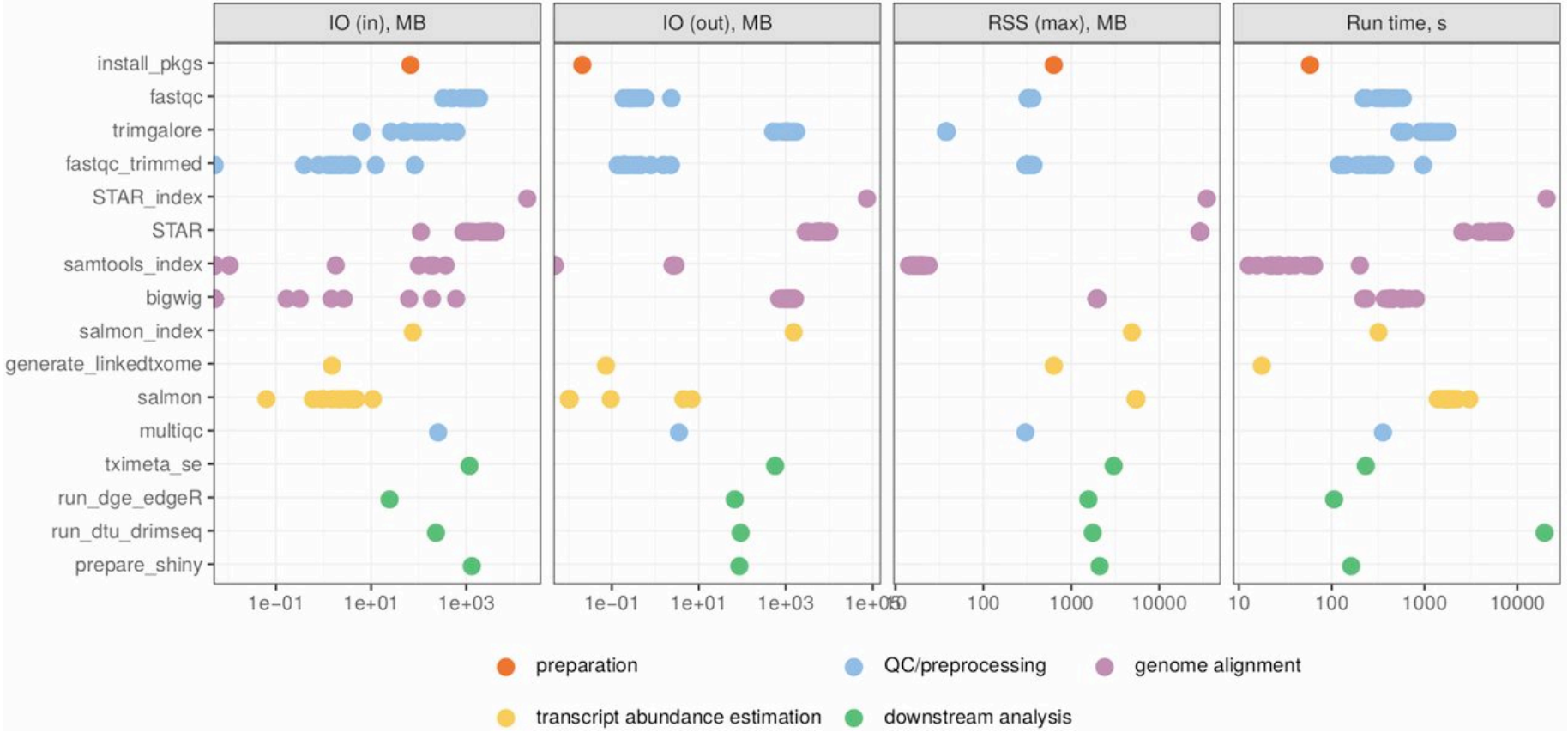


Detailed wiki:
https://github.com/csoneson/ARMOR/wiki



iSEE visualization of the ARMOR output

https://bioconductor.org/packages/release/bioc/html/iSEE.html



ARMOR benchmarks all rules.
We can plot the required resources for the generation of the output files.

ARMOR: useful commands

- snakemake –npr to see what snakemake will be executing
- snakemake setup to see if all required software is available
- snakemake checkinputs to see if your specified design and contrast matrix is valid

Exercise

Run the ARMOR workflow on an example RNA-seq dataset and create a visualization of the results in iSEE or an R markdown document. The example dataset contains a subset of the samples from Doumpas et al. 2019 (https://www.embopress.org/doi/full/10.15252/embj.201798873) and only the reads that map to chromosome 22. You will have three WT and three CHIRON treated samples and we want to know which genes are differentially expressed upon CHIRON treatment. Have a look at the paper if you want to know more about the goal of the study.

You need to:

1. Clone the ARMOR github repo from https://github.com/csoneson/ARMOR
2. Get a tarball of the example dataset from http://imlspenticton.uzh.ch/dump/sta426hs2019/example_dataset.tar and extract its content. metadata.txt contains information about the different files.
3. Create a new config.yaml file and define the location of the FASTQ and reference file. Do not forget to specify the design matrix and the contrast you want to test!
4. Run ARMOR and specify your config file with "`–configfile`"
5. Look at the FastQC/MultiQC output and describe what you observe.
6. Find a differentially expressed gene and plot its logCPM values in all samples. You can either load the resulting "outputR/shiny_sce.rds" object in iSEE and create the visualization there, or you can load the "outputR/edgeR_dge.rds" object in an R session and create an RMD/HTML file with the plot.