RNA-seq data processing and analysis with ARMOR

STA426 – 09.11.2020

Katharina Hembach

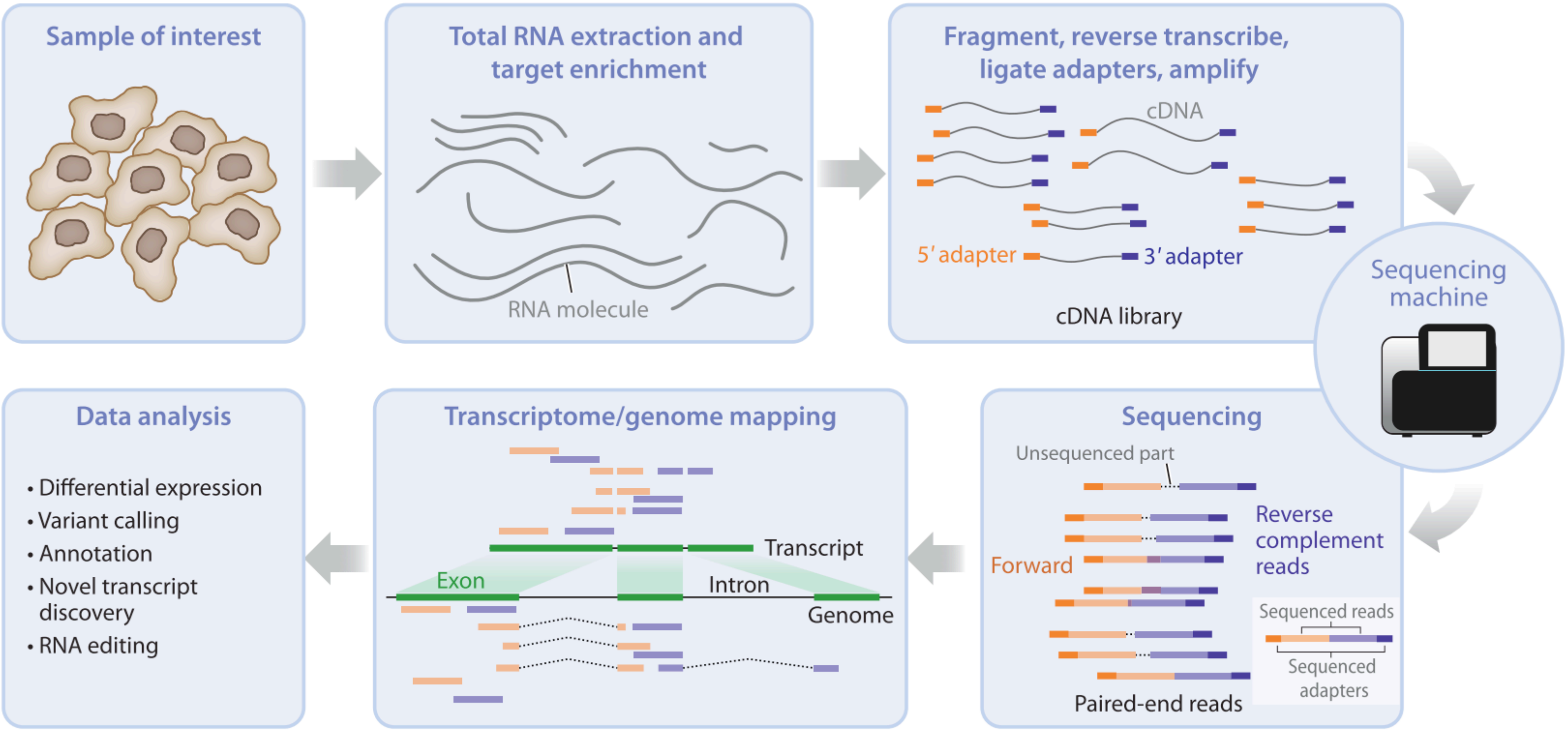


Figure 1

Overview of the experimental steps in an RNA sequencing (RNA-seq) protocol. The complementary DNA (cDNA) library is generated from isolated RNA targets and then sequenced, and the reads are mapped against a reference genome or transcriptome. Downstream data analysis depends on the goal of the experiment and can include, among other things, assessing differential expression, variant calling, or genome annotation.

Van den Berge, K., et al. (2019). RNA Sequencing Data: Hitchhiker's Guide to Expression Analysis. *Annual Review of Biomedical Data Science*

ARMOR workflow

- **A**utomated = snakemake

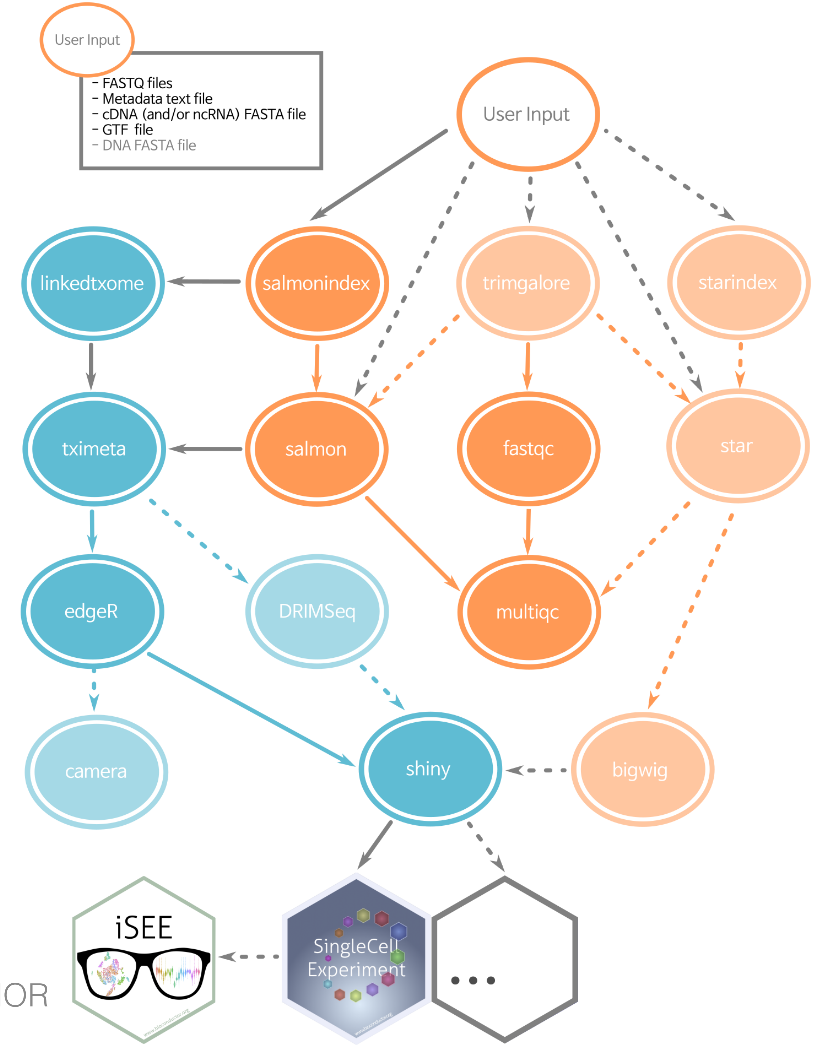
snakemake
A framework for reproducible data analysis

- **R**eproducible = conda and GitHub

CONDA GitHub

- **MO**dular = snakemake rules + configuration file

- **R**NA-seq



https://github.com/csoneson/ARMOR
Orjuela et al., G3 2019

Preprocessing of RNA-seq reads

1. Quality filtering & adapter trimming

   (remove reads with bad quality & adapter sequences)

2. Alignment to reference genome
3. Quantification of feature of interest

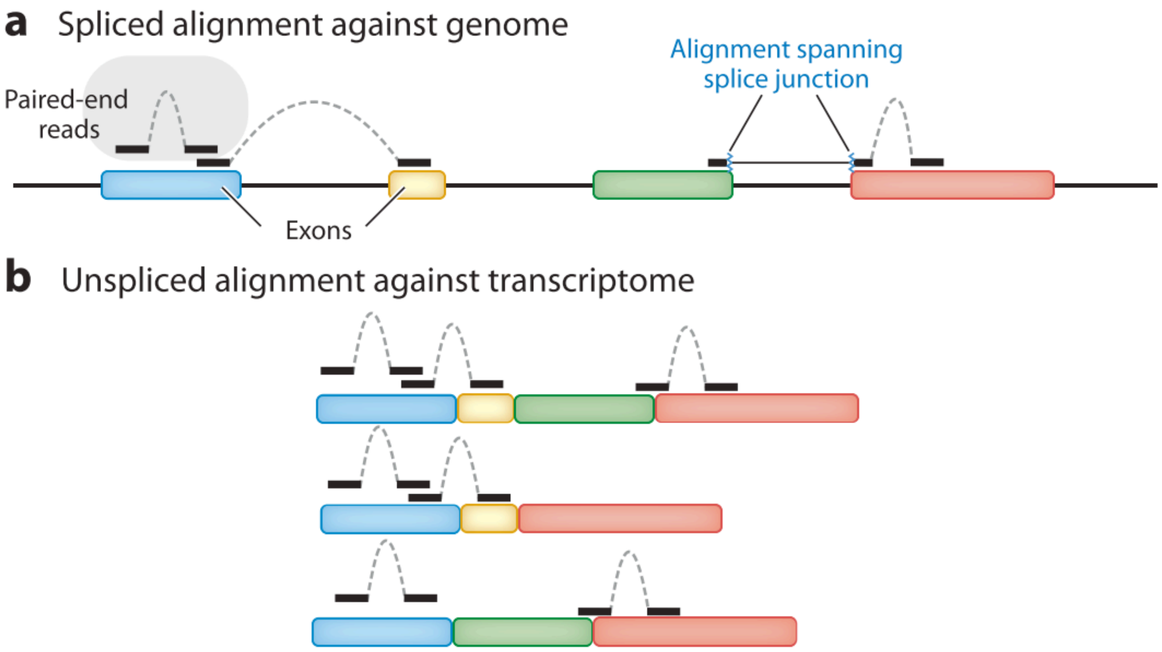
   (gene or transcript)

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
- # reads, read length, read quality, GC content, % duplicated reads, adapter contamination, …
- Tools for quality filtering/adapter trimming:

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

Alignment


a Spliced alignment against genome
Paired-end reads
Alignment spanning splice junction
Exons
b Unspliced alignment against transcriptome

STAR

https://github.com/alexdobin/STAR

HISAT2

http://ccb.jhu.edu/software/hisat2/index.shtml

Salmon

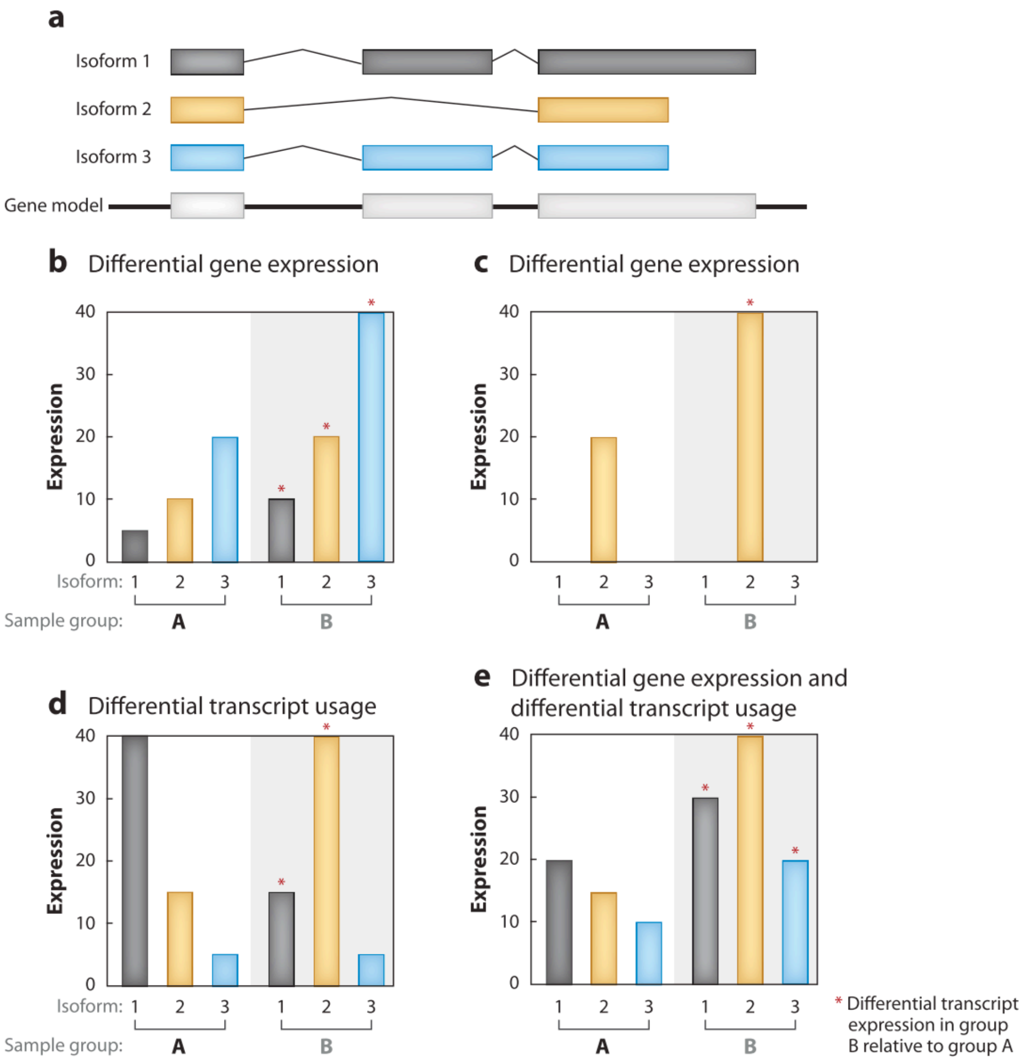
https://combine-lab.github.io/salmon/about/

kallisto

https://pachterlab.github.io/kallisto/about

Van den Berge, K., et al. (2019). RNA Sequencing Data: Hitchhiker's Guide to Expression Analysis. *Annual Review of Biomedical Data Science*

Variants of differential expression



Van den Berge, K., et al. (2019). *Annual Review of Biomedical Data Science*

Statistical analysis

- **Differential gene expression:** Which genes change in expression in different genotypes, treatments, time points, …?

(edgeR http://bioconductor.org/packages/release/bioc/html/edgeR.html or DEseq2 http://bioconductor.org/packages/release/bioc/html/DESeq2.html)

- **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?

CONDA

https://docs.conda.io/projects/conda/en/latest/user-guide/getting-started.html

- Open source package and environment management system for any programming language.
- quickly install, run and update packages and their dependencies
- 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/

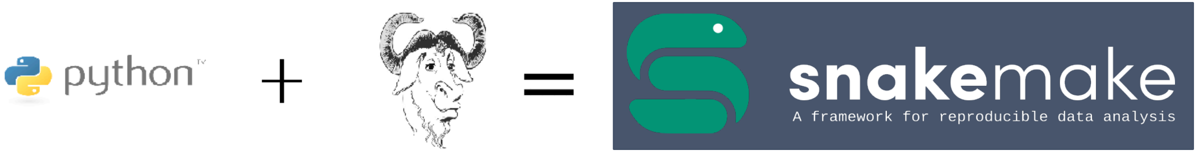
BIOCONDA®

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 export an environment to a YAML file (https://yaml.org/spec/1.2/spec.html) and easily share it

→ reproducibility!

Snakemake



https://snakemake.readthedocs.io/en/stable/

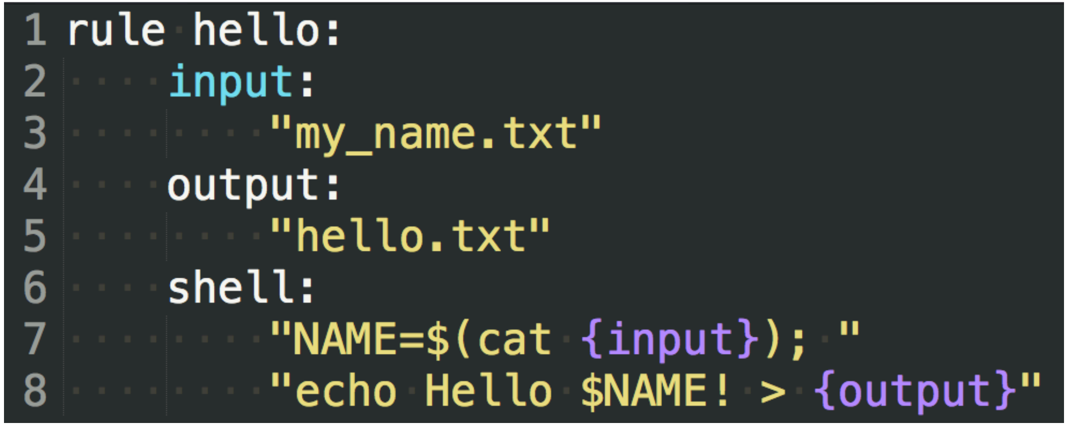
- workflow management system

→ 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
- runs on laptops, clusters, the cloud without modifications
- you can automatically deploy required software with conda

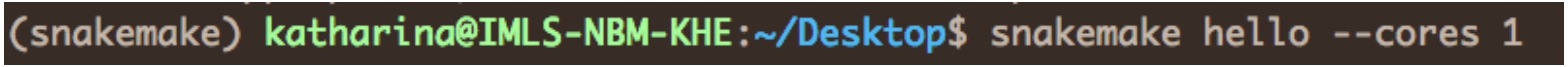
What does it look like?

- Define Snakefile with rules



```
rule hello:
    input:
        "my_name.txt"
    output:
        "hello.txt"
    shell:
        "NAME=$(cat {input}); "
        "echo Hello $NAME! > {output}"
```

- Execute with



```
(snakemake) katharina@IMLS-NBM-KHE:~/Desktop$ snakemake hello --cores 1
```

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`

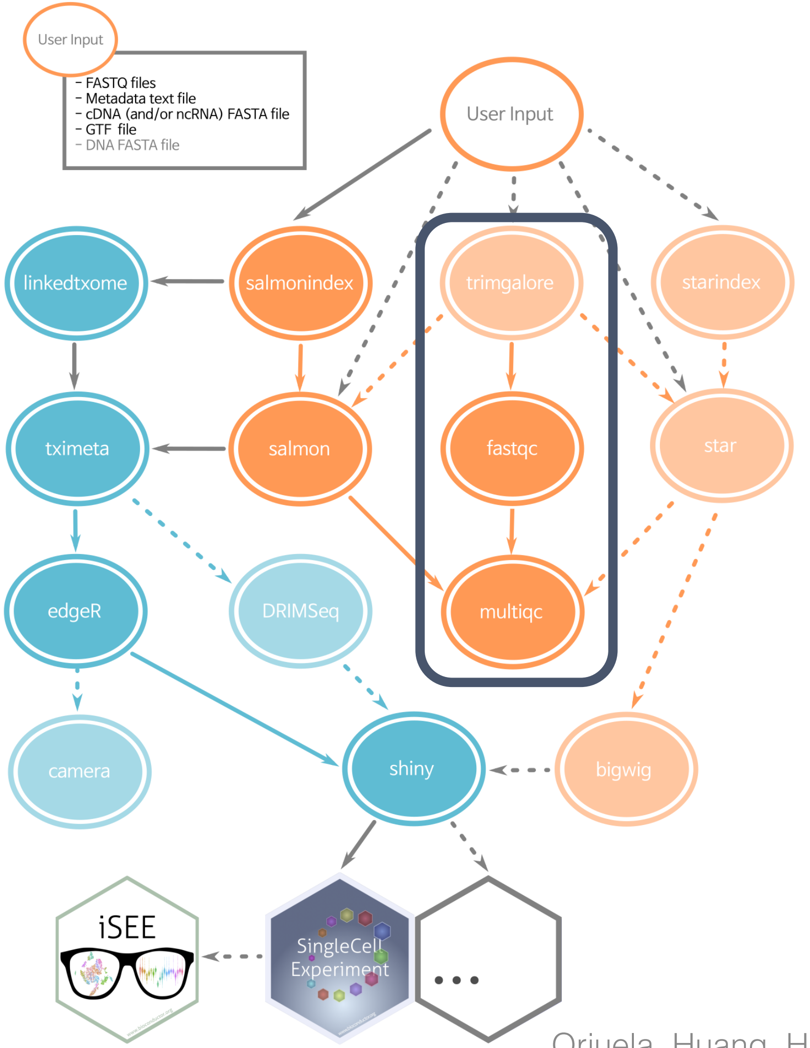
`-l` list all available rules

--cores to use at most this number of cores in parallel

--configfile path to configuration file (e.g. config.yaml)

ARMOR WORKFLOW

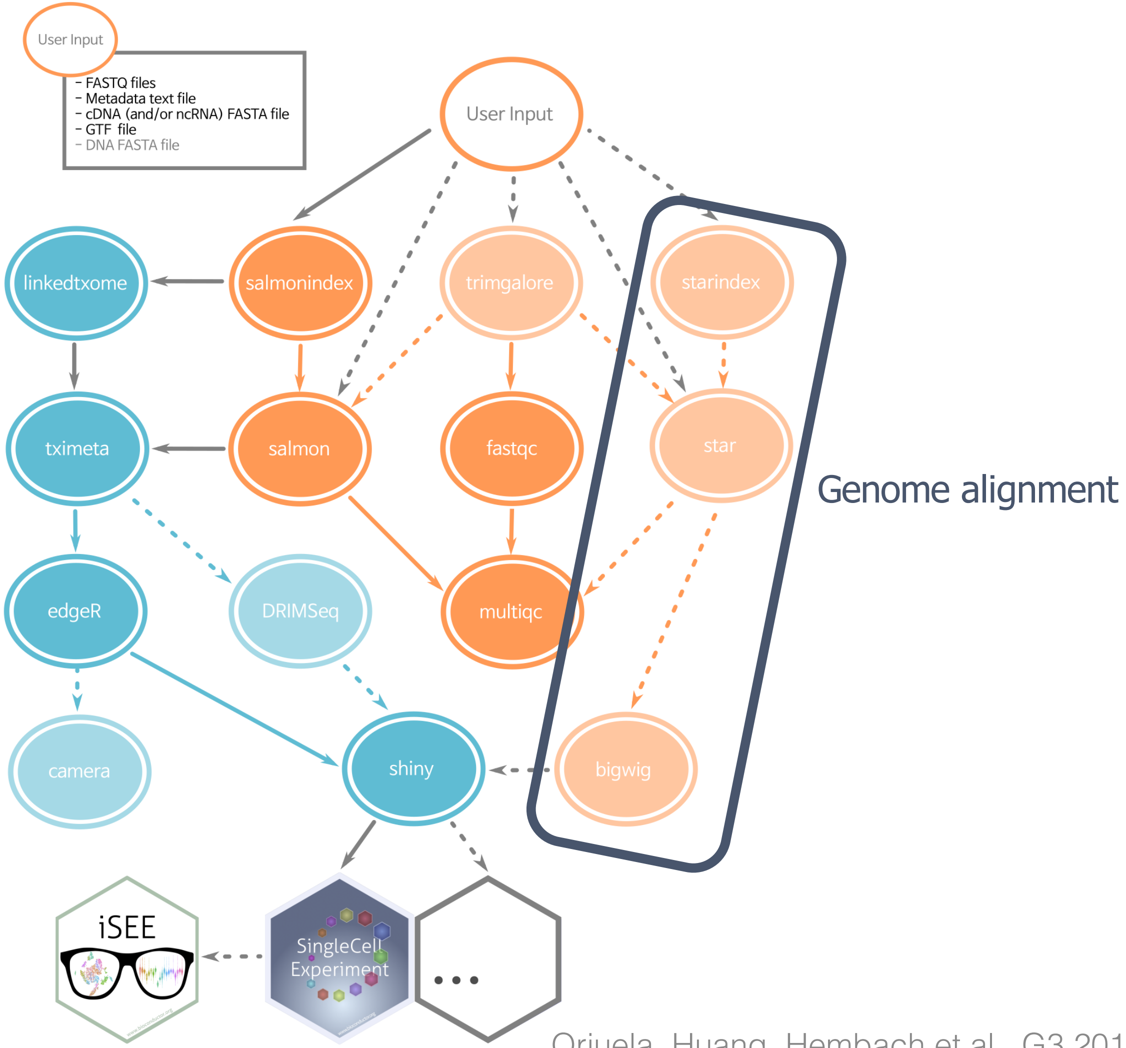
ARMOR workflow



Quality control

Orjuela, Huang, Hembach et al., G3 2019

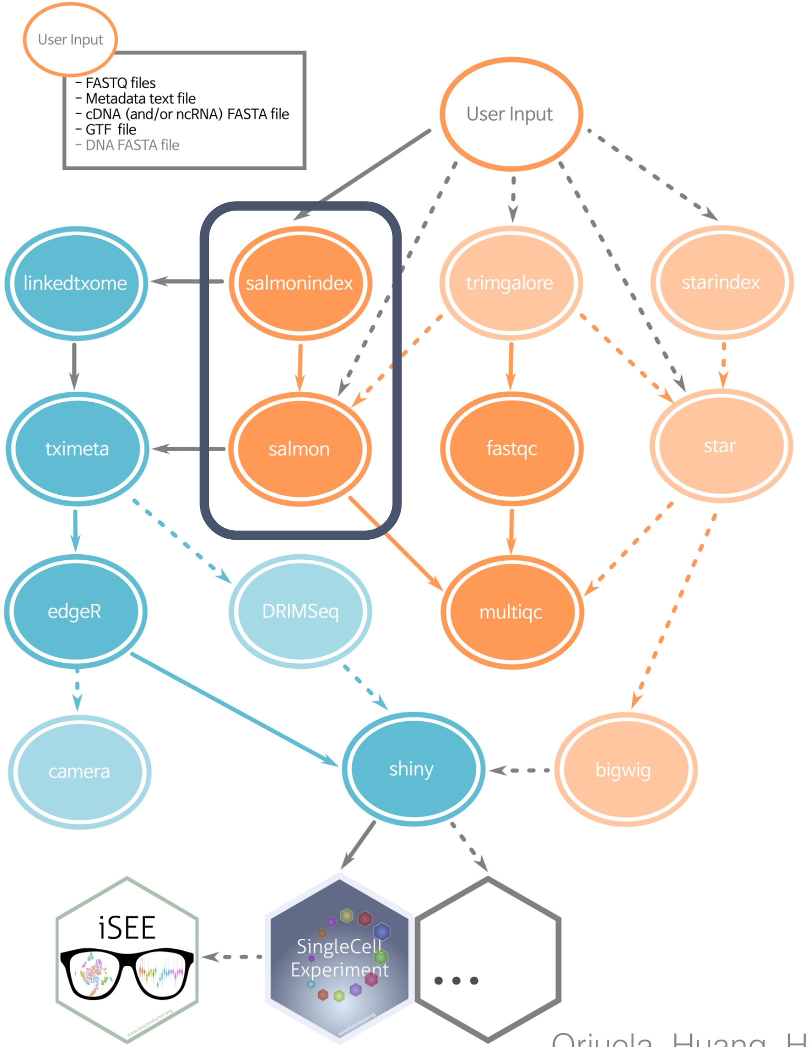
ARMOR workflow



Orjuela, Huang, Hembach et al., G3 2019

ARMOR workflow

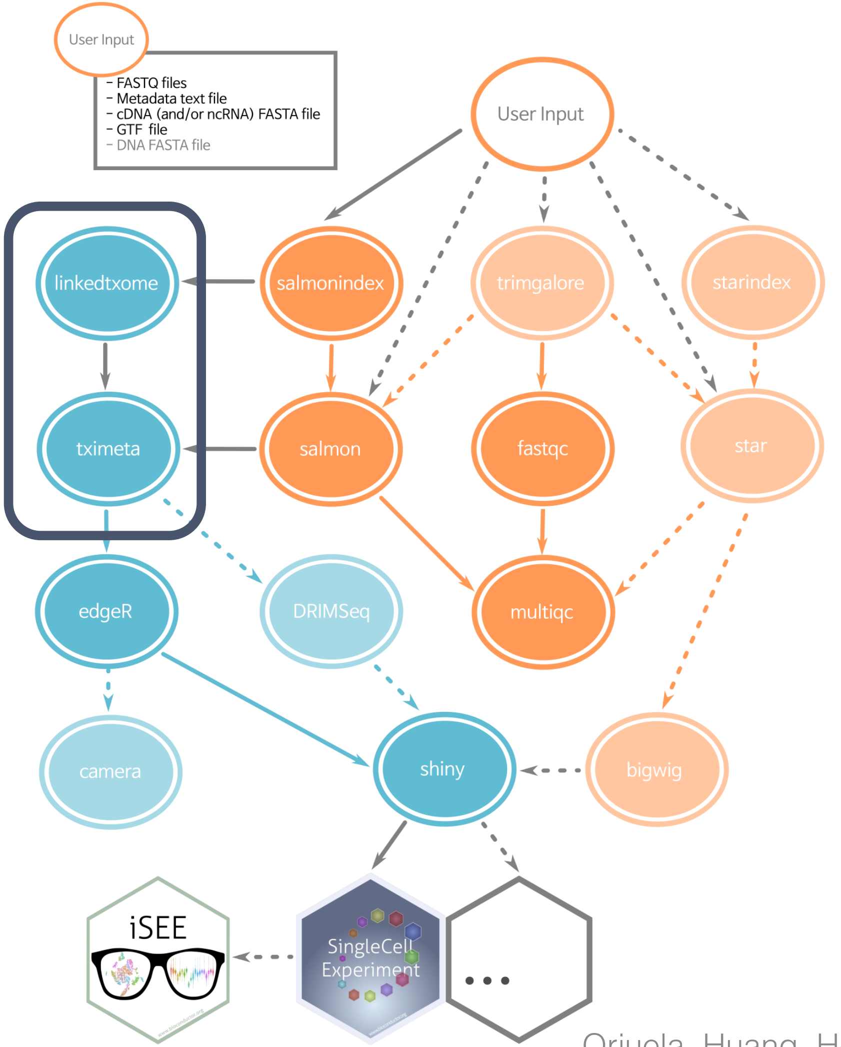
Transcript abundance estimation



Orjuela, Huang, Hembach et al., G3 2019

ARMOR workflow

Data import into R



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

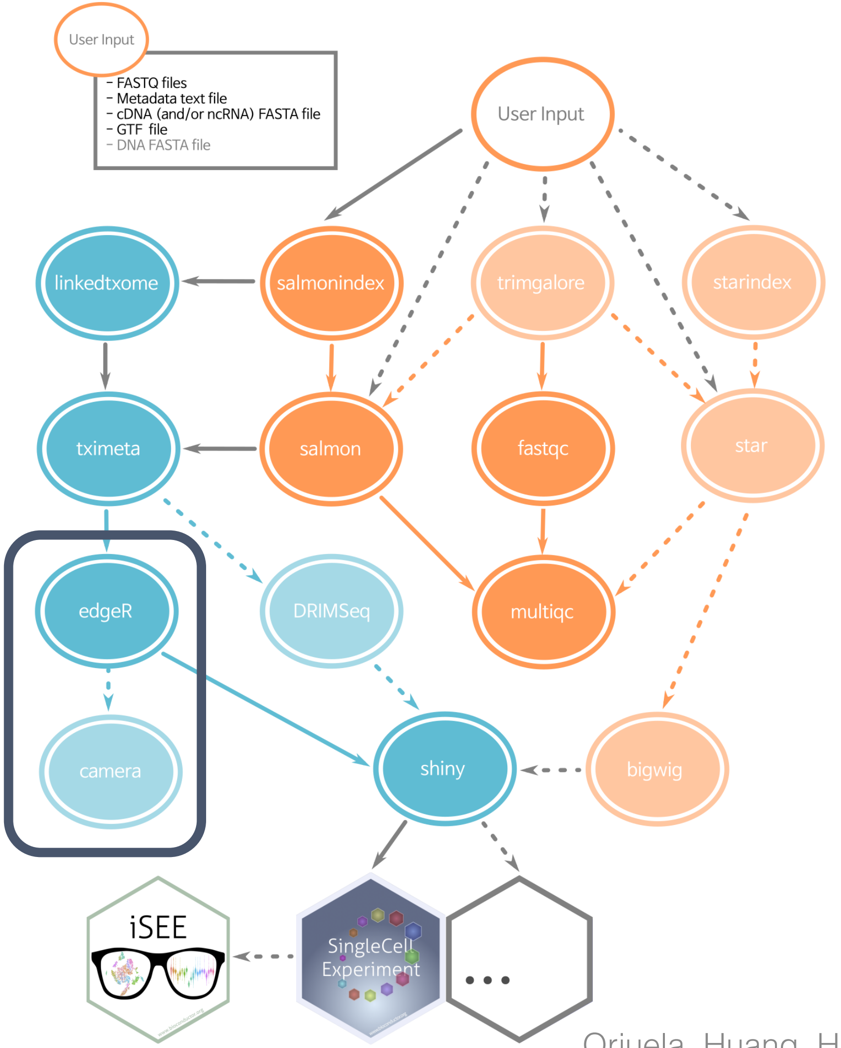
iSEE

SingleCell Experiment

...

Orjuela, Huang, Hembach et al., G3 2019

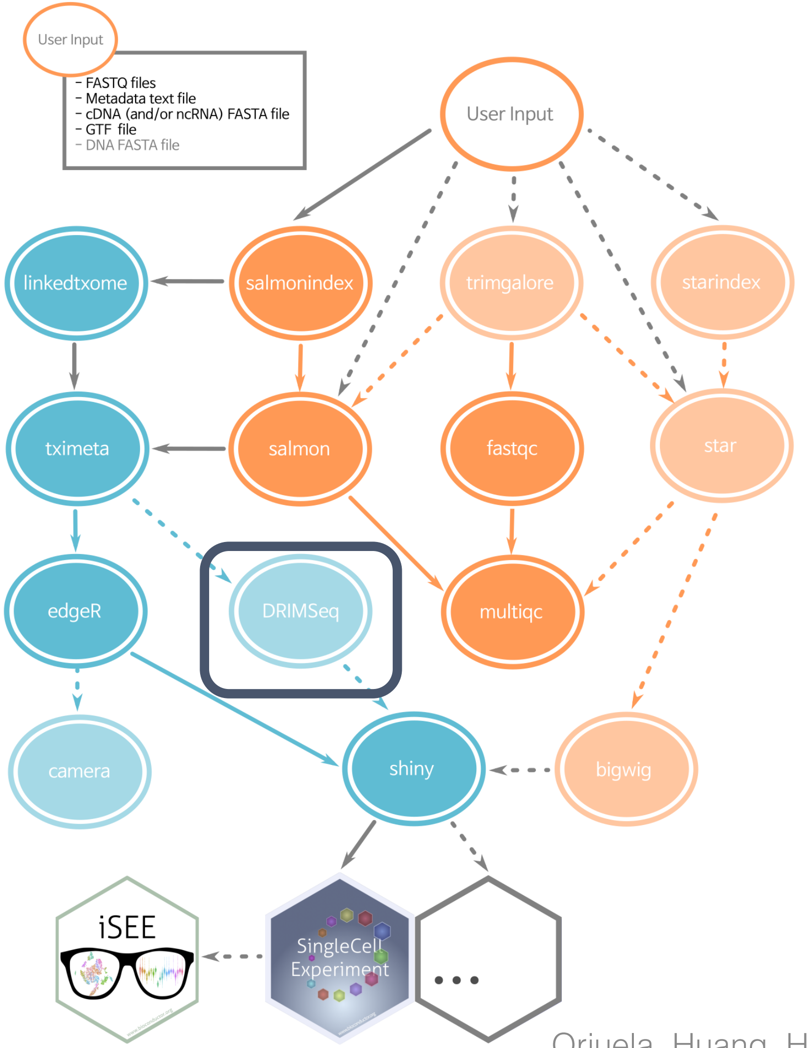
ARMOR workflow



Differential gene expression analysis & gene set enrichment

Orjuela, Huang, Hembach et al., G3 2019

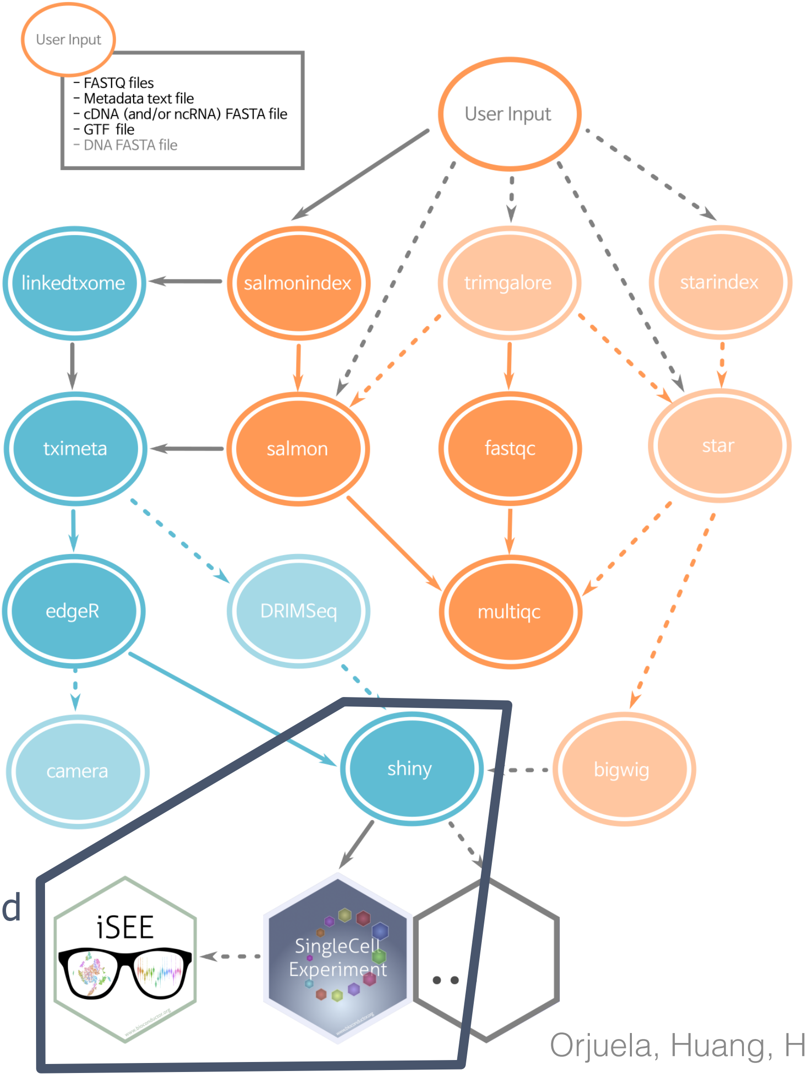
ARMOR workflow



Differential transcript usage

Orjuela, Huang, Hembach et al., G3 2019

ARMOR workflow

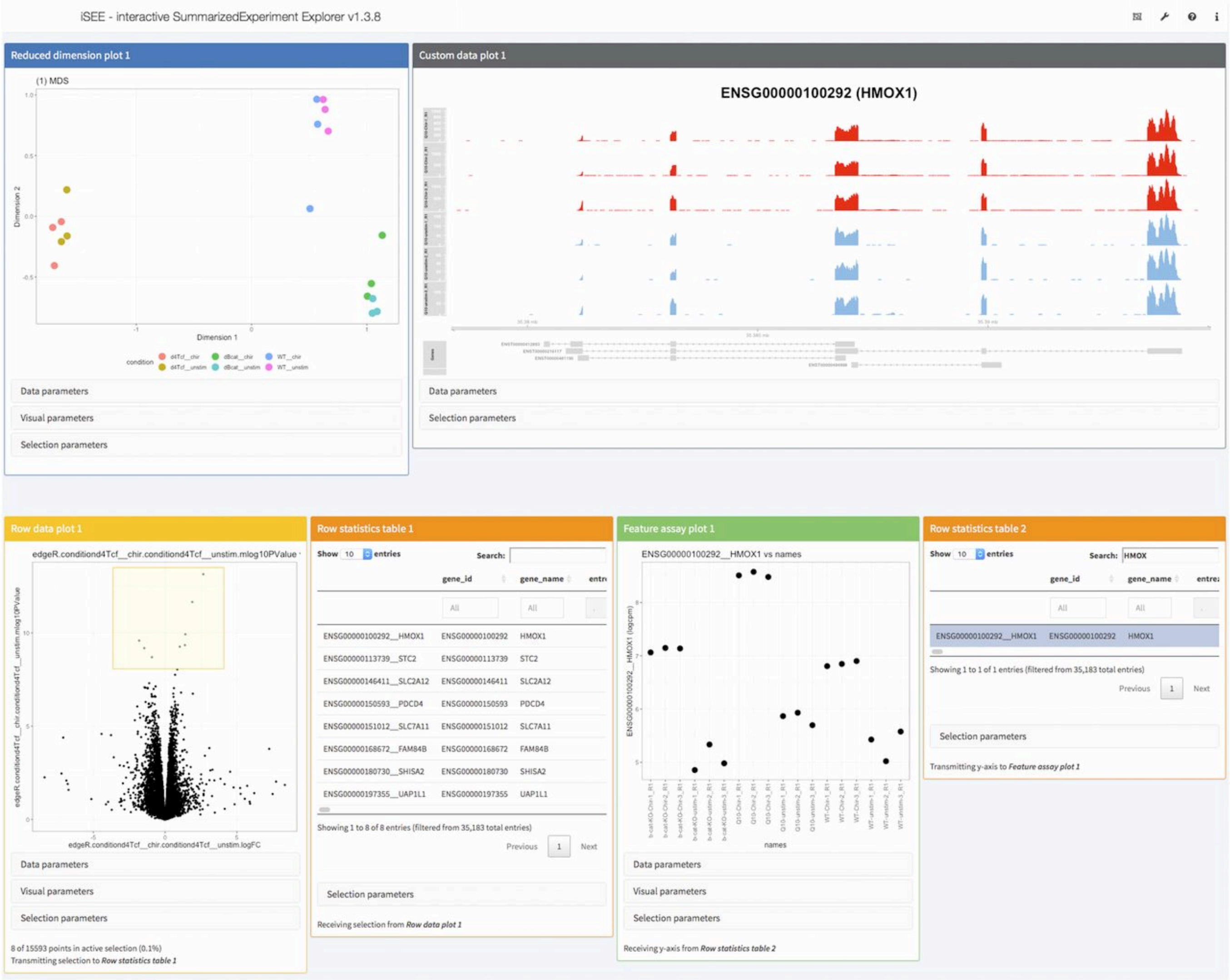


Data export and visualisation

Orjuela, Huang, Hembach et al., G3 2019

iSEE visualization of the ARMOR output

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



8 of 15593 points in active selection (0.1%)

Transmitting selection to *Row statistics table 1*

Row statistics table 1

Show 10 entries Search:

	gene_id	gene_name	entr
	All	All	
ENSG00000100292__HMOX1	ENSG00000100292	HMOX1	
ENSG00000113739__STC2	ENSG00000113739	STC2	
ENSG00000146411__SLC2A12	ENSG00000146411	SLC2A12	
ENSG00000150593__PDCD4	ENSG00000150593	PDCD4	
ENSG00000151012__SLC7A11	ENSG00000151012	SLC7A11	
ENSG00000168672__FAM84B	ENSG00000168672	FAM84B	
ENSG00000180730__SHISA2	ENSG00000180730	SHISA2	
ENSG00000197355__UAP1L1	ENSG00000197355	UAP1L1	

Showing 1 to 8 of 8 entries (filtered from 35,183 total entries)

Receiving selection from *Row data plot 1*

Receiving y-axis from *Row statistics table 2*

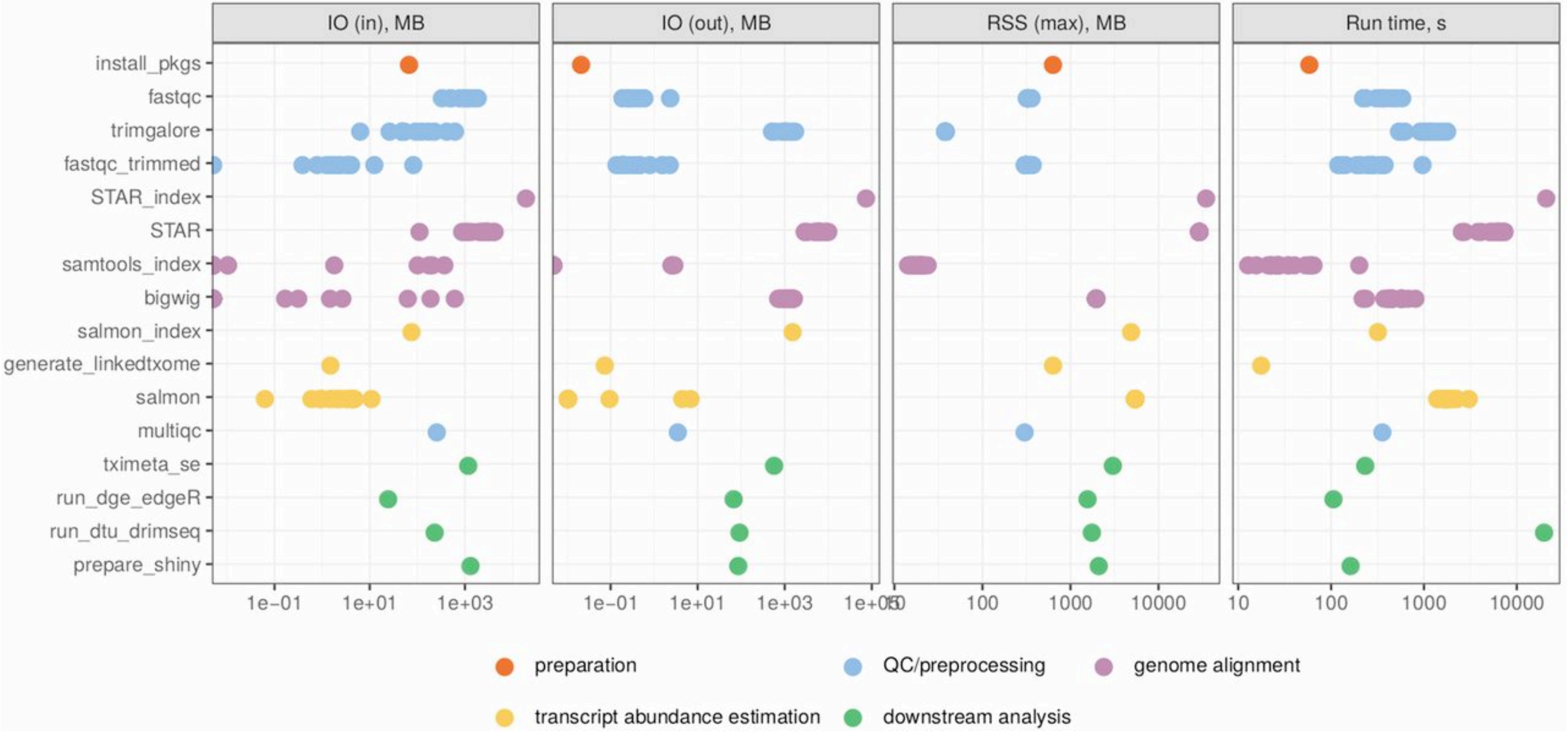
Row statistics table 2

Show 10 entries Search: HMOX

	gene_id	gene_name	entre:
	All	All	
ENSG00000100292__HMOX1	ENSG00000100292	HMOX1	

Showing 1 to 1 of 1 entries (filtered from 35,183 total entries)

Transmitting y-axis to *Feature assay plot 1*



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

Notes

- If you use renku, select at least 2GB of memory (STAR needs a lot).
- Fork one of these renku projects and start an environment:

https://renkulab.io/projects/rok.roskar/sta426hs2020 or
https://renkulab.io/projects/mark.robinson/armor_bioc311

- You might want to relax the filtering of lowly expressed genes (edgeR-dge.Rmd).
- DRIMSeq might fail because there are not enough genes (you can disable it in config.yaml).