ETH zürich

Bioinformatics pipeline for revealing tumour heterogeneity

Mustafa Anıl Tuncel

europython BASEL July 8-14 2019

Mustafa Anıl Tuncel

Software Engineer @ ETH Zürich

Research interests

- Data analysis workflows
- Bioinformatics
- Machine learning
- Recommender systems

anilbey

/in/aniltuncel

anilbey

Outline

- Background
  - Biology prior
  - Single cell sequencing technologies
  - Mutations on DNA
- DNA mutation trees
  - Tree model
  - MCMC moves
- Pipeline
  - Snakemake
  - HDF5

What is a cell?

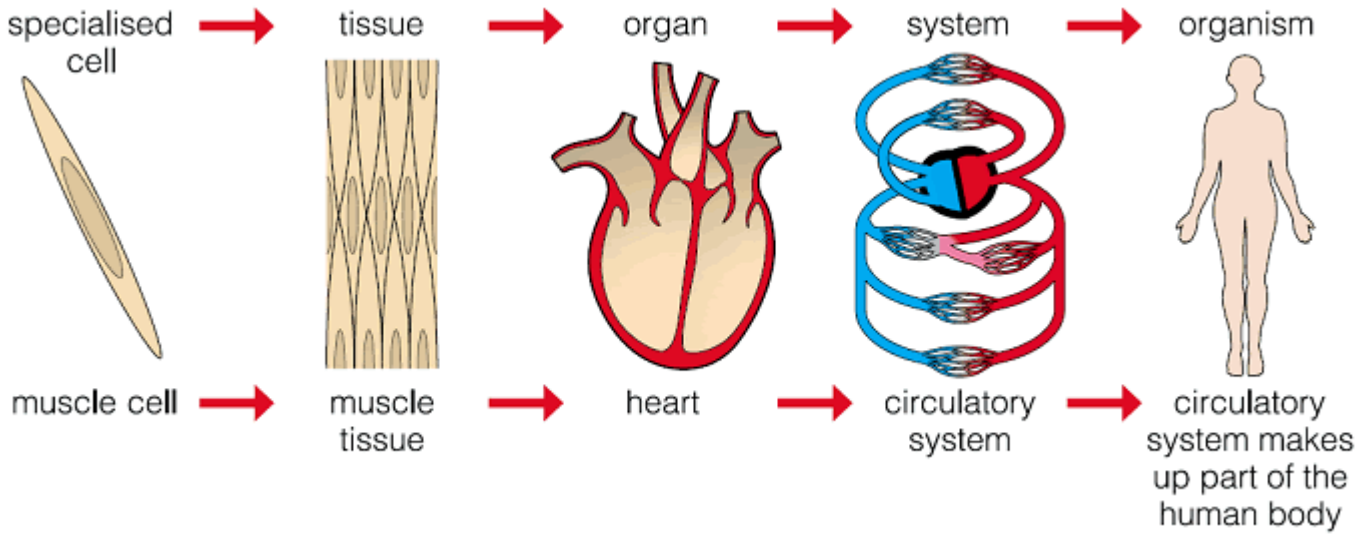
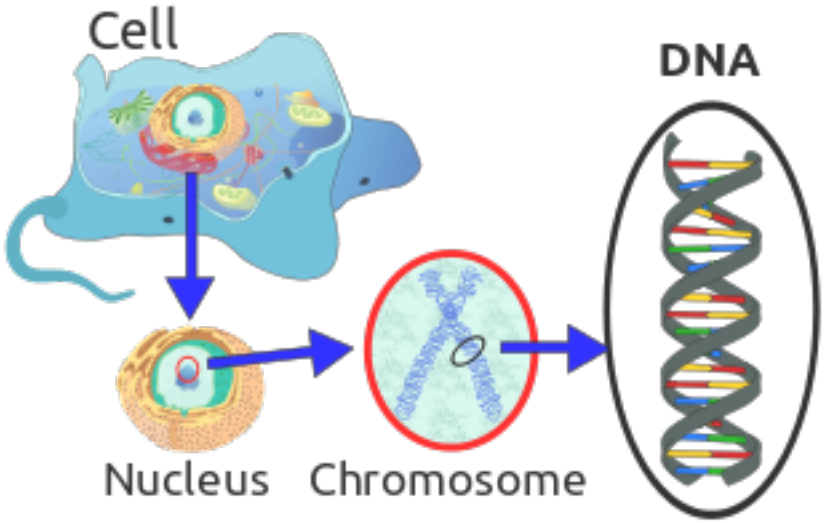


Figure 1. Representation of cell, tissue, organ, system and organism. Retrieved from https://www.colscol.com/body-system/

DNA from single cells



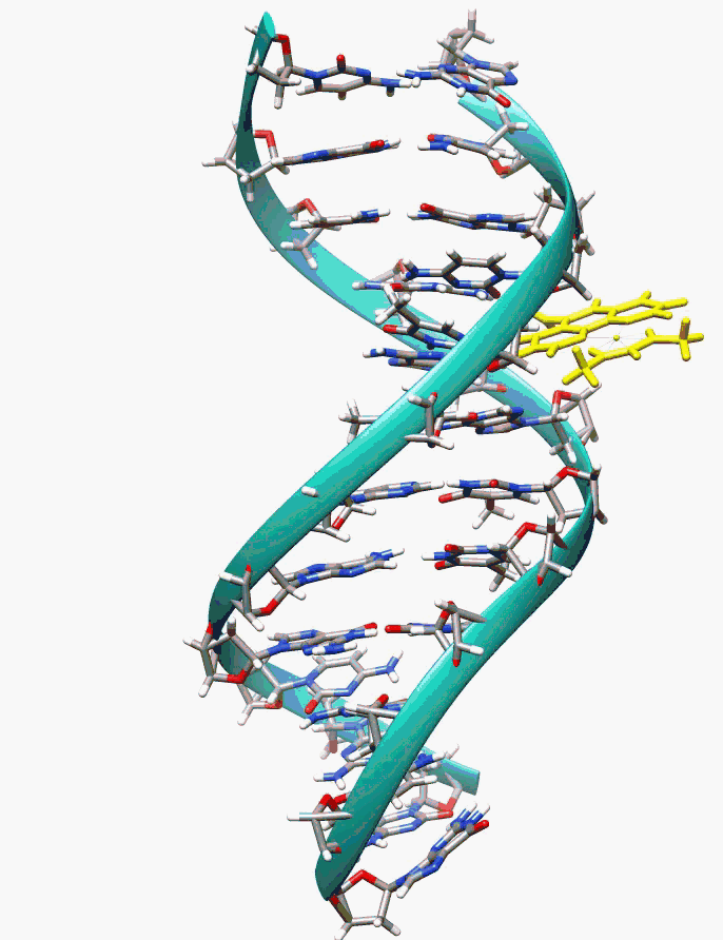
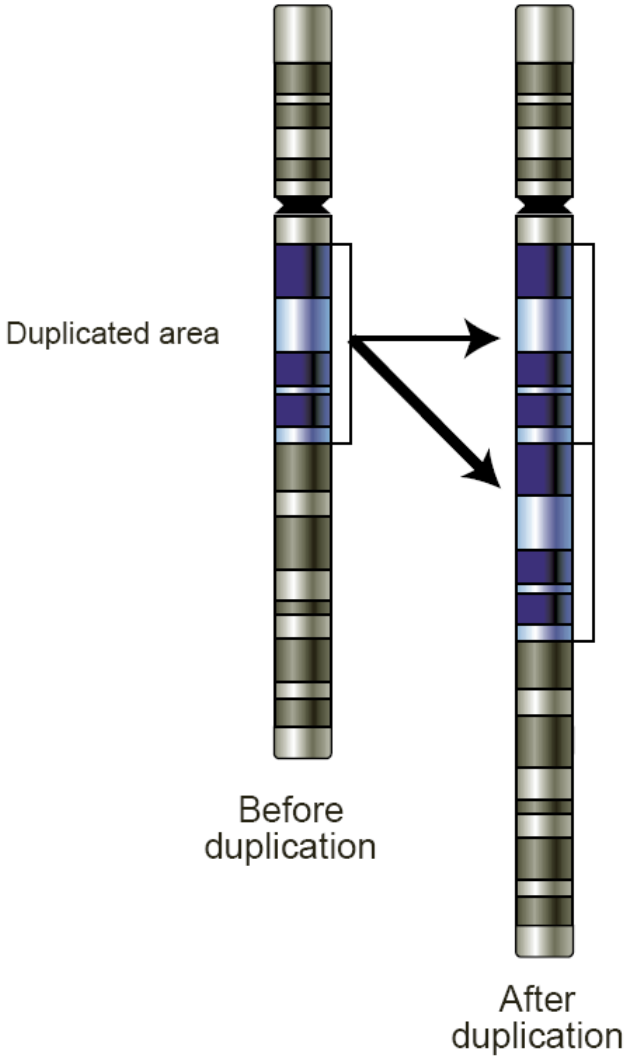


Figure 2. DNA structure. Retrieved from https://www.interleucina.org/

Structural mutations on DNA

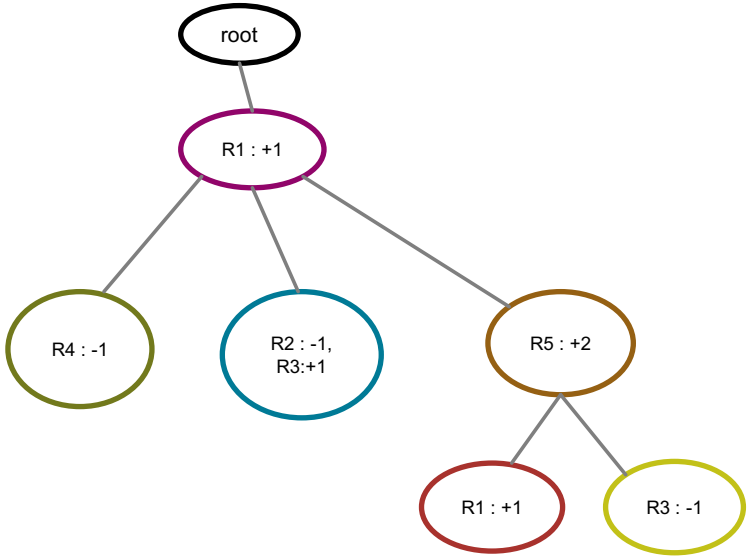
- Copy number variations
  - Deletion
  - Duplication

- Mutations from DNA of single cells
- Heterogeneous
- Have ancestors, children, siblings

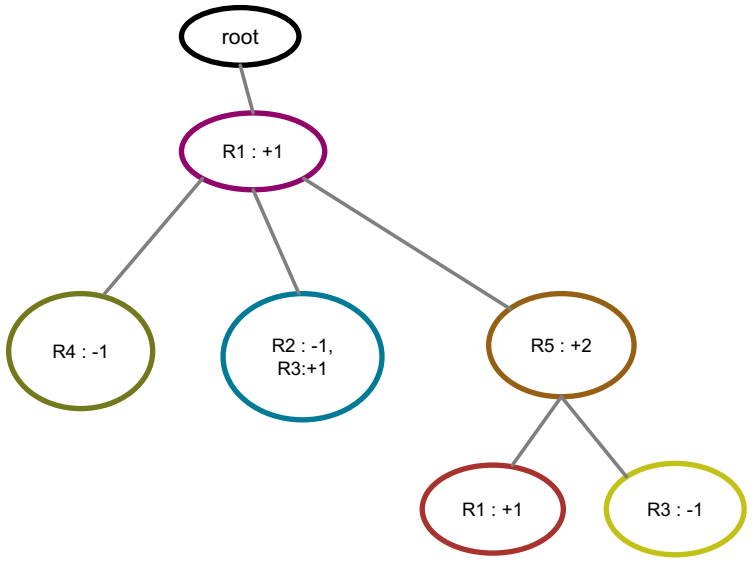


Trees to represent structural mutations

DNA:

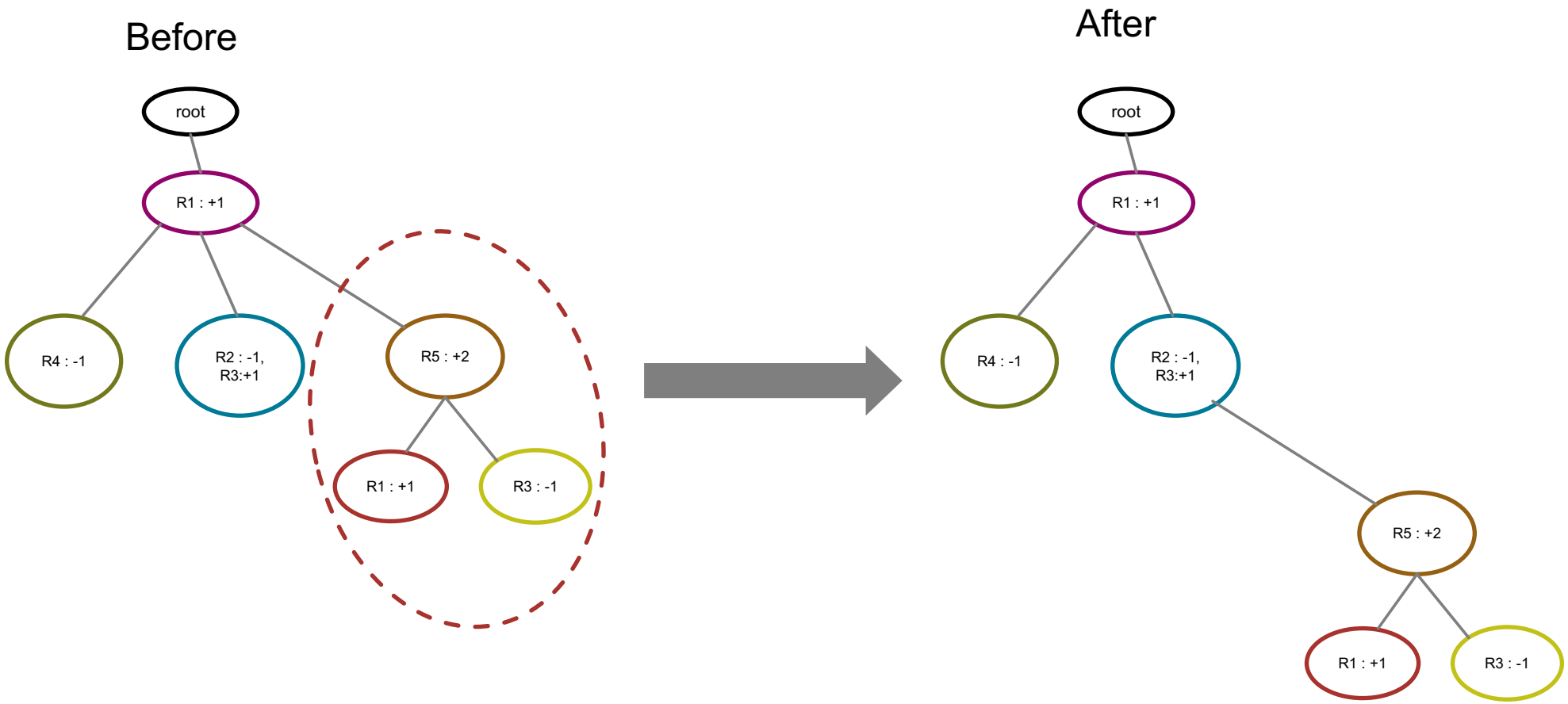


Learning the tree

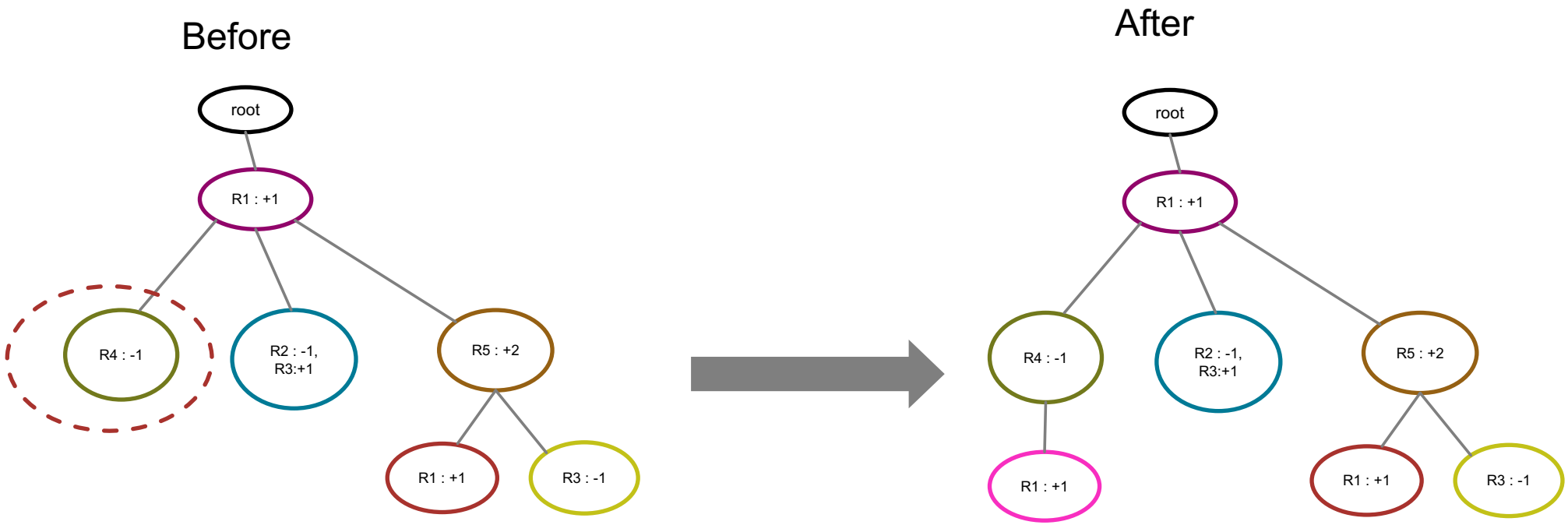


- Dirichlet-multinomial model with overdispersion
- We target maximising the tree posterior with an MCMC scheme
    - Prune-reattach
    - Label swap
    - Add/remove events
    - Add/remove node
    - Condense/split node
    - Genotype preserving prune-reattach

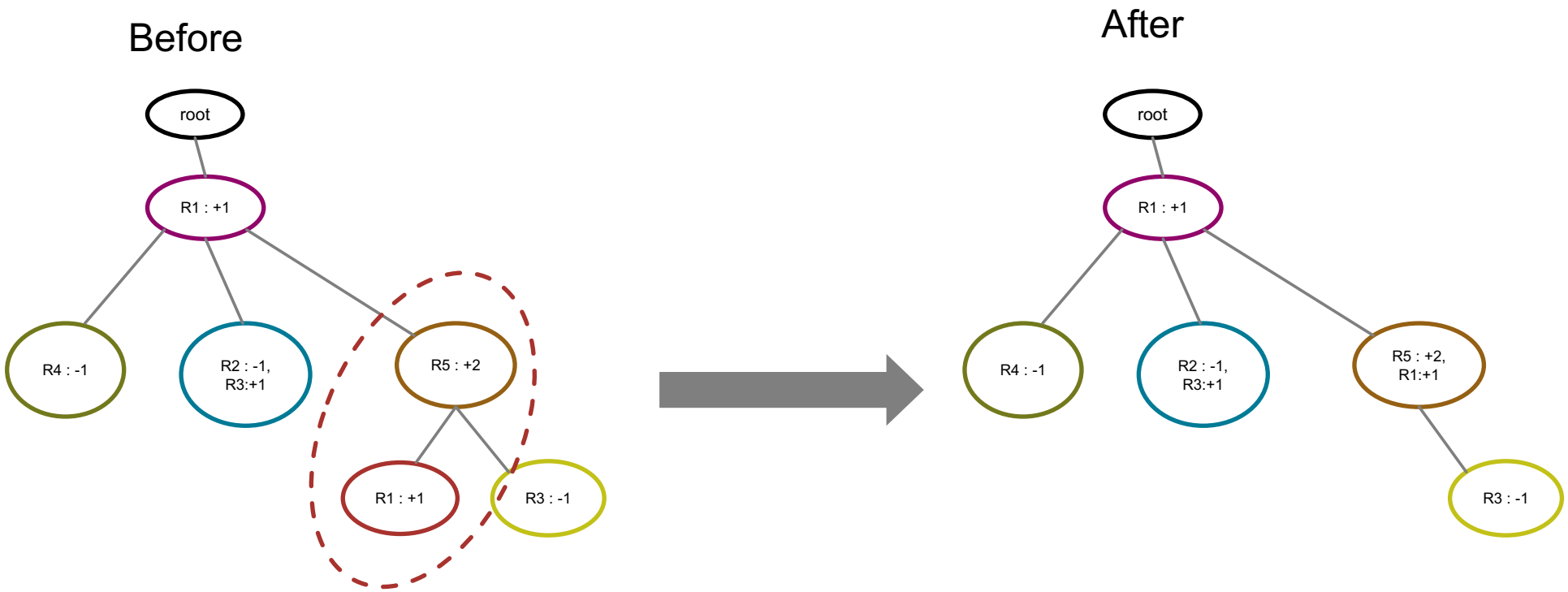
Prune-reattach



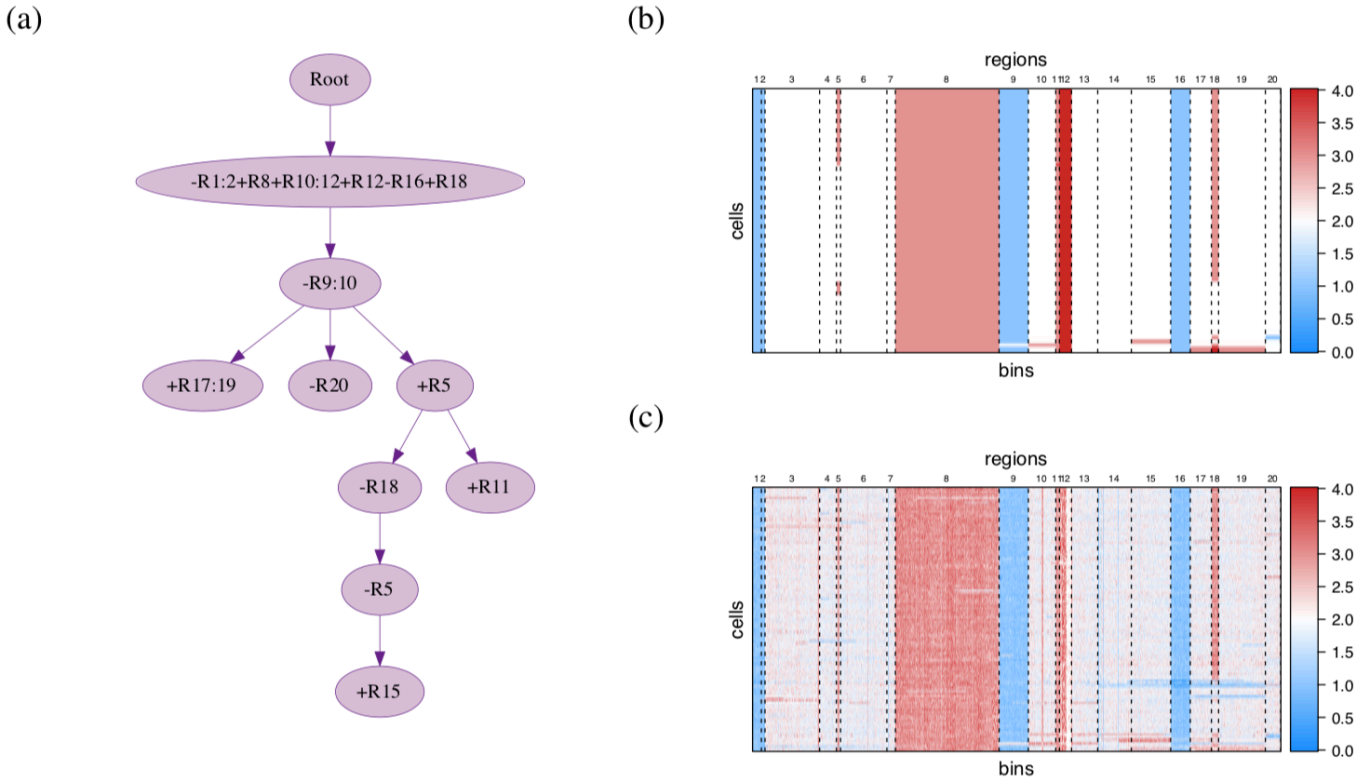
Add / remove node



Condense / split node



Tree learned from mouse data



Inferred tree (a) and copy number profiles (b) for the first 20 regions (3047 bins) of the real sequencing data For comparison the normalised counts per bin are drawn in (c).

What else is required?

- Reproducibility in research
- Scalability
- Support for Multiple programming languages
- Multi processing
- Cluster execution
- Resources management
- Statistics about resource usages

Workflow management system

Python + GNU Makefile = Snakemake

Snakemake

- A Pythonic workflow management system
- Extends the Python syntax
- Follows the GNU make paradigm
  - Workflows are defined in terms of rules that define how to create output files from input files
  - Dependencies between the rules are determined automatically
- Benefits from Python libraries
- Automated logging of the status
- Suspend/resume workflow

- A general-purpose workflow management system for any discipline

```
pip install snakemake
```

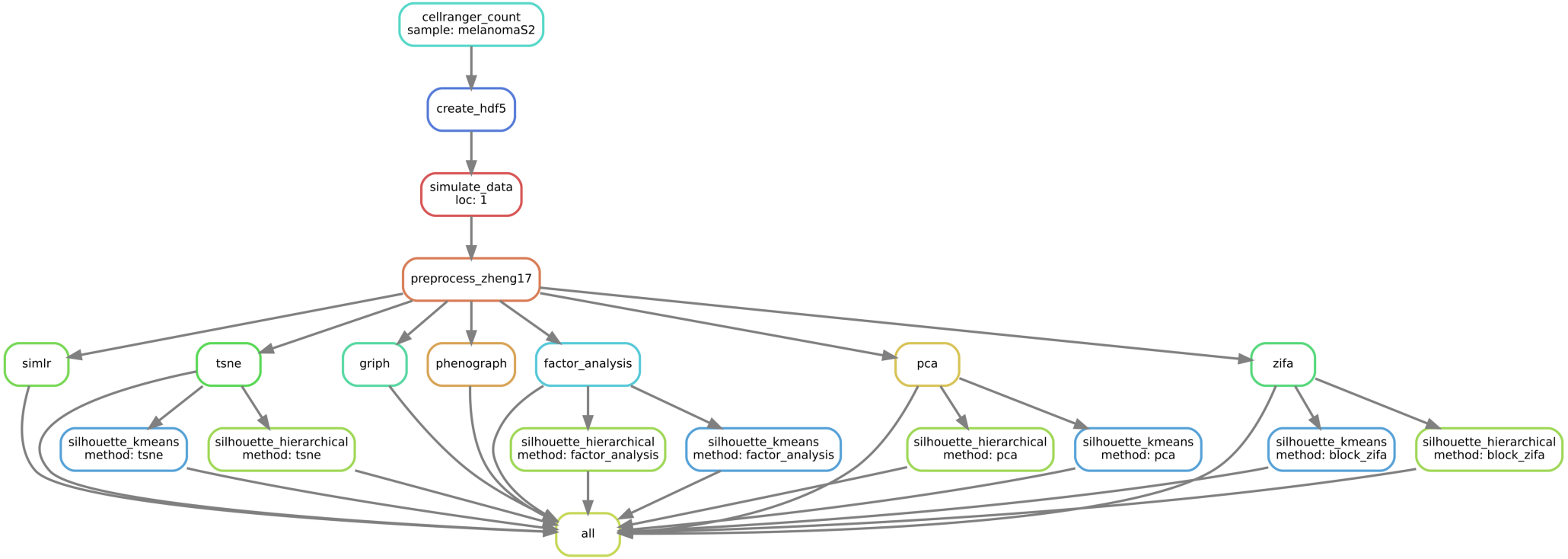
Example: read mapping

```
rule bwa_map:
    input:
        "data/genome.fa",
        "data/samples/A.fastq"
    output:
        "mapped_reads/A.bam"
    shell:
        "bwa mem {input} | samtools view -Sb - > {output}"
```

Example: read mapping (generalised)

```
rule bwa_map:
    input:
        "data/genome.fa",
        "data/samples/{sample}.fastq"
    output:
        "mapped_reads/{sample}.bam"
    shell:
        "bwa mem {input} | samtools view -Sb - > {output}"
```

DAG of jobs


cellranger_count
sample: melanomaS2
create_hdf5
simulate_data
loc: 1
preprocess_zheng17
simlr
tsne
griph
phenograph
factor_analysis
pca
zifa
silhouette_kmeans
method: tsne
silhouette_hierarchical
method: tsne
silhouette_hierarchical
method: factor_analysis
silhouette_kmeans
method: factor_analysis
silhouette_hierarchical
method: pca
silhouette_kmeans
method: pca
silhouette_kmeans
method: block_zifa
silhouette_hierarchical
method: block_zifa
all

Snakefile

```
import glob
import os
from pathlib import Path
from secondary_analysis import SecondaryAnalysis

fastqs_path = config['fastqs_path']
analysis_path = config['analysis_path']

def rename_fastq(s_name):
    split_name = s_name.split('_')
    new_name = '_'.join(split_name[6:7]+split_name[-4:])
    return new_name

rule rename_fastqs:
    input:
        rules.merge_files.output.done
    output:
        "rename_fastqs_done.txt"
    run:
        merged_fastqs_path = fastqs_path + "/merged/"
        print(merged_fastqs_path)
        fastqs_dir = merged_fastqs_path
        for filename in os.listdir(fastqs_dir):
            if filename.startswith("MERGED_BSSE") and filename.endswith('.gz'):
                print("old name: " + filename)
                print("new name: " + rename_fastq(filename))
                os.rename(fastqs_dir+filename, fastqs_dir+rename_fastq(filename))
        Path('rename_fastqs_done.txt').touch()
```

Config file

```
1   {
2     "analysis_prefix": "HELAVELAVELVELA_scD_Ar1v1.6",
3     "ref_genome_version": "GRCh37",
4     "ref_genome_path": "/path/to/reference/genome",
5     "fastqs_path": "~/path/to/fastqs",
6     "analysis_path": "~/path/to/analysis",
7     "cellranger_dna":
8     {
9       "local_cores": 24,
10      "local_mem": 384,
11      "mem_per_core": 16,
12      "mem":"16384",
13      "time":"1438"
14    }
15  }
16
17
```

Cluster execution

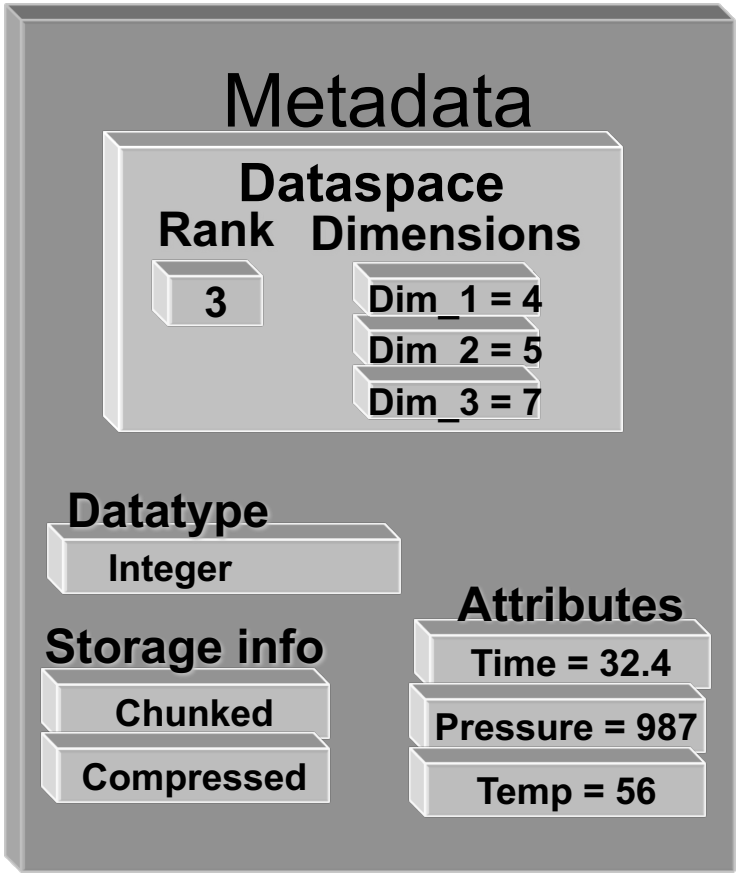
```
bsub -J snake_job_name  -W 23:59 -R "rusage[mem=16000]" "snakemake -s self_benchmark_snakefile.py --stats
./simulations.stats --cluster 'bsub -M {params.mem} -n {threads} -W {params.time} -R "rusage[
mem={params.mem},scratch={params.scratch}]"' -j 24 -p -k --latency-wait 300"
```

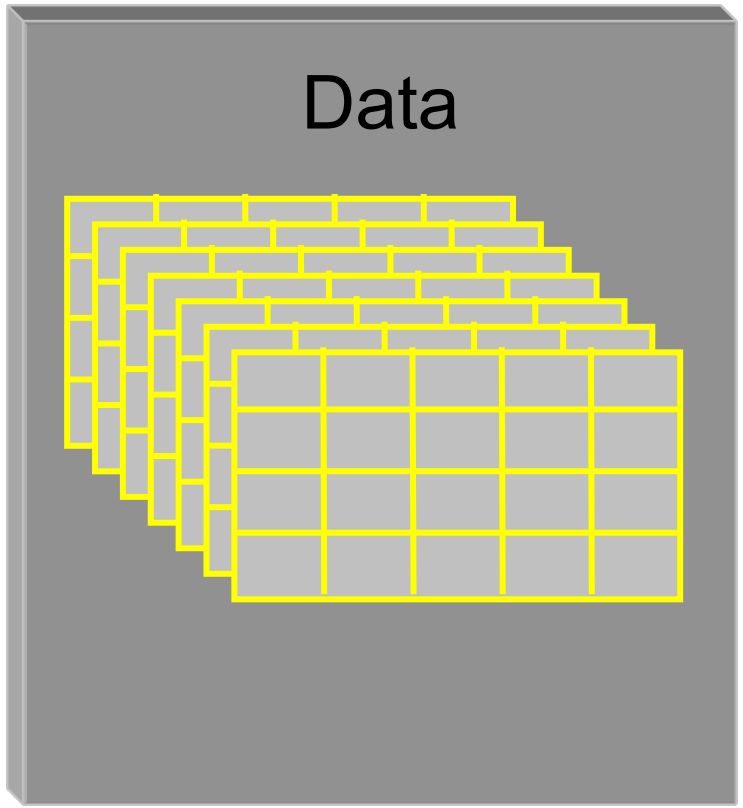
- Configurable for LSF/BSUB scheduler
- Allows scaling without changing the workflow

HDF5

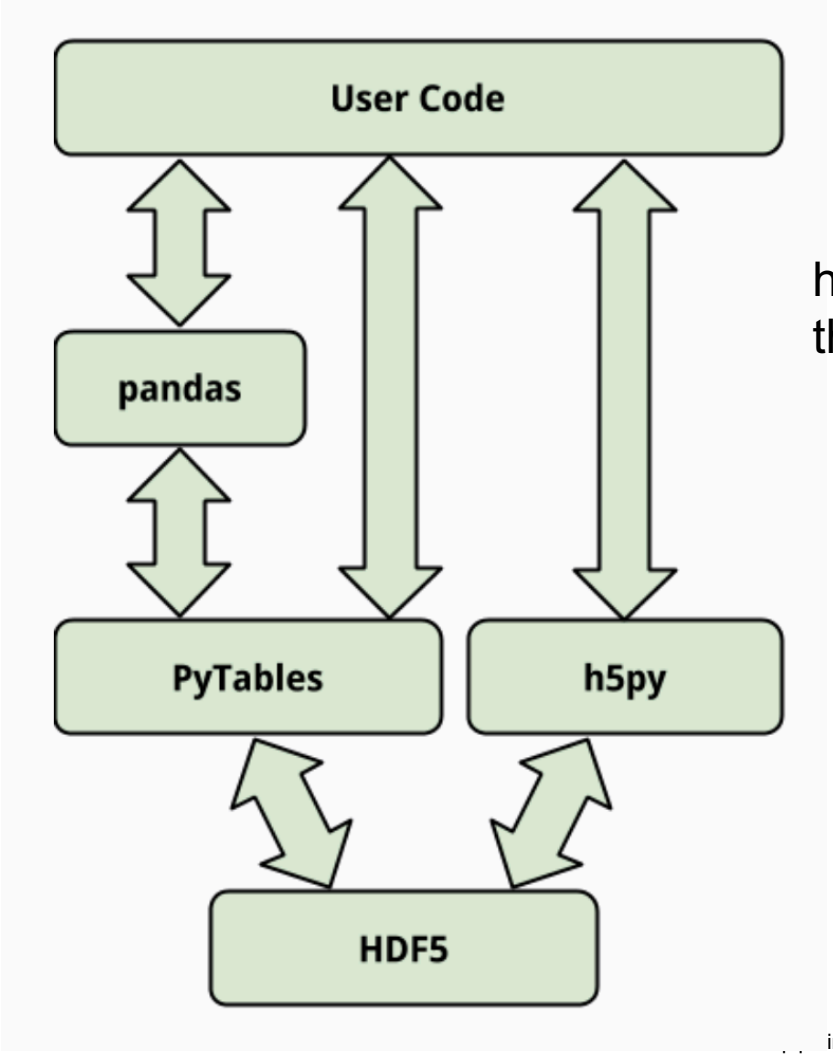
HDF = **H**ierarchical **D**ata **F**ormat

- **Hierarchical data format v5**
- Binary files
- Easy to manage multiple datasets
- Keeps metadata with data
- Fast I/O operations & storage space optimization (compressed binary files)
- Platform/language independent
- Self describing
- No need to load whole data





HDF5 wrappers in Python



h5py is a thin, pythonic wrapper around the HDF5

```
pip install h5py
```

```
pip install tables
```

Outline

- Background
  - Biology prior
  - Single cell sequencing technologies
  - Mutations on DNA
- DNA mutation trees
  - Tree model
  - MCMC moves
- Pipeline
  - Snakemake
  - HDF5

Future work

- Publish the method
  - Compare to clustering methods
  - Evaluate on simulated data
  - Show results on real data
- Wrap up the workflow as a Python package
  - Do the C++ bindings
  - Open source it

Thank you!

Mustafa Anıl Tuncel
Software Engineer

ETH Zurich

anilbey
/in/aniltuncel
anilbey
mtuncel@ethz.ch