Bioinformatics pipeline for revealing tumour heterogeneity

Mustafa Anıl Tuncel
Mustafa Anıl Tuncel
Software Engineer @ ETH Zürich

Research interests

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

<table>
<thead>
<tr>
<th>Region 1</th>
<th>Region 2</th>
<th>Region 3</th>
<th>Region 4</th>
<th>Region 5</th>
</tr>
</thead>
</table>

![Diagram of trees representing structural mutations with regions and mutations indicated.]
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

Before

root

R1 : +1
R2 : -1, R3 : +1
R4 : -1
R5 : +2

After

root

R1 : +1
R2 : -1, R3 : +1
R4 : -1
R5 : +2

R1 : +1
R3 : -1

R1 : +1
R3 : -1
Add / remove node

Before

After
Condense / split node

Before

After

Department of Biosystems Science and Engineering

Bioinformatics pipeline for revealing tumour heterogeneity. | Mustafa Anıl Tuncel | 10.07.19 | 11
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

```yaml
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)

```plaintext
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
Snakefile

```python
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 = '_' + 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

```json
{
    "analysis_prefix": "HELALEVELELVEA_scD_Ar1v1.6",
    "ref_genome_version": "GRCh37",
    "ref_genome_path": "/path/to/reference/genome",
    "fastqs_path": "~/path/to/fastqs",
    "analysis_path": "~/path/to/analysis",
    "cellranger_dna":
        {
            "local_cores": 24,
            "local_mem": 384,
            "mem_per_core": 16,
            "mem": "16384",
            "time": "1438"
        }
}
```
Cluster execution

- Configurable for LSF/BSUB scheduler
- Allows scaling without changing the workflow
HDF5

- 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

HDF = Hierarchical Data Format
HDF5 wrappers in Python

h5py is a thin, pythonic wrapper around the HDF5

pip install h5py
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

anlbey
/in/aniltuncel
anlbey
mtuncel@ethz.ch