# Building Reproducible Bioinformatics Workflows with Snakemake

Ryan C. Thompson
Salomon Lab
The Scripps Research Institute

August 26, 2016

# What is Reproducible Research?

- Seems obvious, but precise definition is not necessarily agreed upon (like most of science)

# What is Reproducible Research?

- Seems obvious, but precise definition is not necessarily agreed upon (like most of science)
- At the very least, you should be able to **easily re-run your analysis from beginning to end**, with all the same parameters and options. Ideally with a minimum number of manual steps.

# What is Reproducible Research?

- Seems obvious, but precise definition is not necessarily agreed upon (like most of science)
- At the very least, you should be able to **easily re-run your analysis from beginning to end**, with all the same parameters and options. Ideally with a minimum number of manual steps.
- Something like this:

```
git clone https://github.com/username/reproducible-repo
cd reproducible repo
./run-entire-workflow.sh
# (Wait a few hours/days/weeks...)
open paper.pdf
```

# What is Reproducible Research?

Bonus points for being:

- **Portable** - runs for other people & on other machines
- **Parallelizable** - runs on a cluster, on AWS, etc.
- **Auditable** - Record software versions, parameters, commands, data versions, etc. at runtime for later retrieval
- **Maintainable** - Easy to make changes/additions to the pipeline
- **In version control** - Easy to tell from the history exactly what version of the pipeline was run on a given date

# Why Reproducible Research?

- Allow others to verify your work
- Easily answer reviewer skepticism of methods
- Provide an example to others for how to do the analysis

# Why Reproducible Research?

- Allow others to verify your work
- Easily answer reviewer skepticism of methods
- Provide an example to others for how to do the analysis

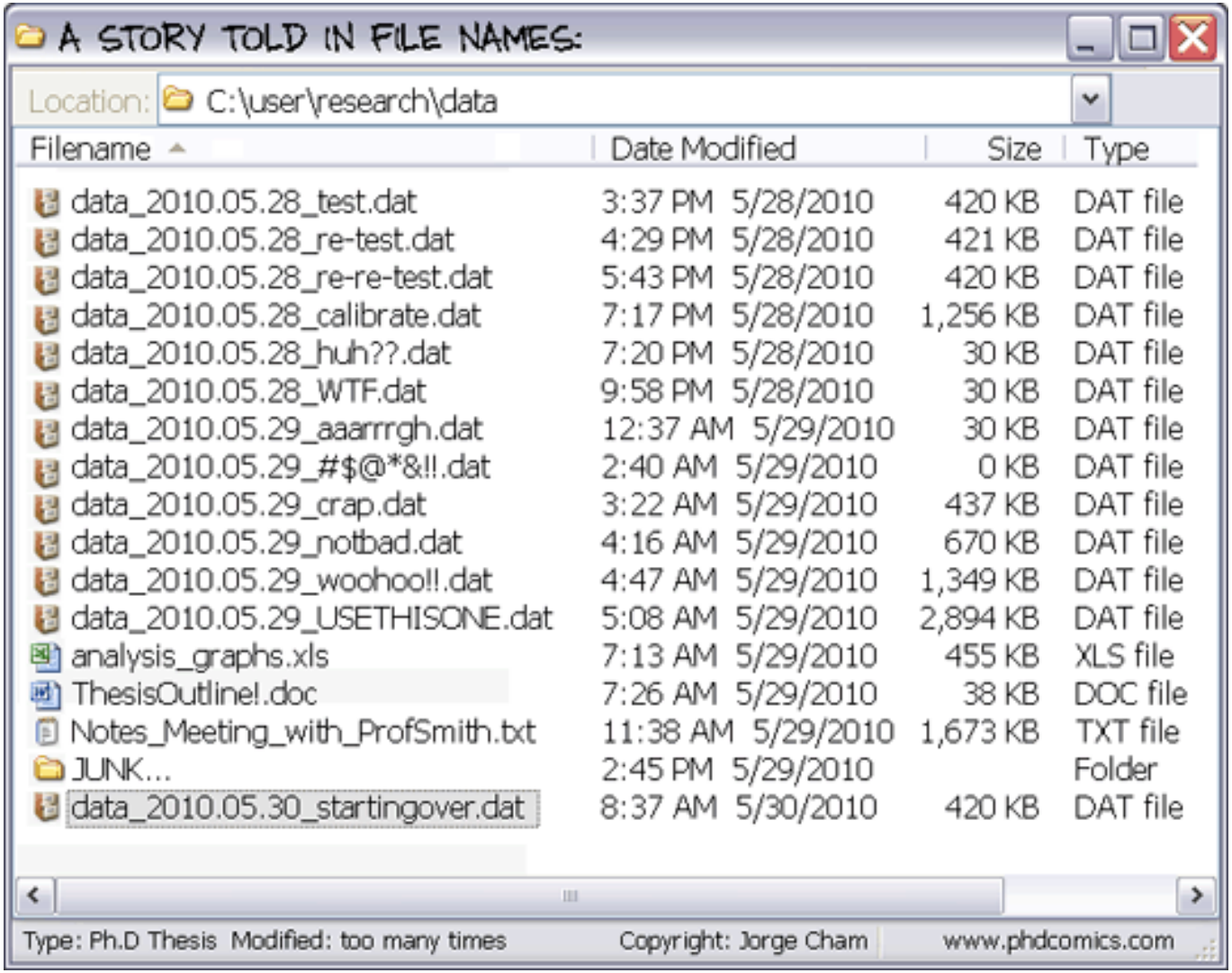
Ok, sure, but what's in it for *me*, right now?

# Why Reproducible Research? (For selfish people)

Do any of these questions sound familiar?

- How did I generate this file?
- Which script generated this file?
- Which version of the results is this file?
- Which of these files is the new results, and which one is the old?
- Why have the results changed between these two files?
- Which version of the program did we use to generate these results?
- Can I compare results between these two files?
- Can you re-do the analysis on hg38 instead of hg19?
- Can you re-do the analysis with a different aligner?
- Shouldn't you use the `--do-what-I-actually-wanted` option in the first step of the pipeline?

# Why Reproducible Research? (For selfish people)



# Why Reproducible Research? (For selfish people)

Without a reproducible workflow:

- Output files become precious and irreplacable over time, as software versions change or you forget how you generated them.
- Wanting to avoid messing with them discourages you from even trying to re-run earlier steps to ensure they still work.
- Different steps in the workflow run on different dates, with software upgrades in between, mean that your final results may be never even be reproducible by any one software version.

# Why Reproducible Research? (For selfish people)

Without a reproducible workflow:

- Output files become precious and irreplacable over time, as software versions change or you forget how you generated them.
- Wanting to avoid messing with them discourages you from even trying to re-run earlier steps to ensure they still work.
- Different steps in the workflow run on different dates, with software upgrades in between, mean that your final results may be never even be reproducible by any one software version.

With a reproducible workflow:

- Output files are disposable; the only cost to reproduce them is time.
- Ease of re-generating results encourages experimentataion and testing, since you can always get back to where you were.
- When the workflow is finished and you're ready to publish, you can easily re-run the entire thing at once with a consistent set of software versions, configurations, etc.

# Why Reproducible Research? (For selfish people)

More bonuses:

- You're going to have to clean up the data before you publish it anyway. No manual steps mean the data will almost certainly be cleaner and more organized to begin with.
- Your paper is going to get a *lot* more citations if you provide code that people can adapt to their own purposes.

# The Data

Genes and Immunity (2016), 1–15

www.nature.com/gene

**ORIGINAL ARTICLE**

## Promoter H3K4 methylation dynamically reinforces activation-induced pathways in human CD4 T cells

SA LaMere, RC Thompson, HK Komori, A Mark and DR Salomon

- RNA-seq and ChIP-Seq data released as GEO accession **GSE73214**

# The Workflow(s)

Workflow will be split into two parts:

- One part for building all the aligner indices, collecting gene annotation data, and all the other tasks that relate only to the reference:
  `https://github.com/DarwinAwardWinner/hg38-ref`
- Second part for the actual data processing & analysis: fetching reads from SRA, aligning, counting, differential expression/binding, etc.:
  `https://github.com/DarwinAwardWinner/CD4-csaw`

## Snakemake – A scalable bioinformatics workflow engine

Johannes Köster [1,2*], Sven Rahmann [1]

[1]Genome Informatics, Institute of Human Genetics, University of Duisburg-Essen
[2]Paediatric Oncology, University Childrens Hospital Essen

# The Tool

**Snakemake – A scalable bioinformatics workflow engine**

Johannes Köster [1,2*], Sven Rahmann [1]

[1]Genome Informatics, Institute of Human Genetics, University of Duisburg-Essen
[2]Paediatric Oncology, University Childrens Hospital Essen



- Similar to `make`, but written in Python
- Arbitrary wild cards in filenames, not just suffixes like `make`
- Arbitrary Python code to build the workflow steps (called rules)
- Each step can be any combination of Python code, shell commands/pipelines, & R code (via rpy2 module)

# Snakemake goodies

- Auto-deletes incomplete output files of failed/cancelled runs
- Auto-creates output directories
- Arbitrary wildcards in file names. E.g. `aligned/rnaseq_{aligner}_{genome}_{transcriptome}/{samp`
- Same workflow can run nearly unmodified on a cluster or cloud
- Remote file access (read & write where applicable): HTTP, FTP, SFTP, S3, etc.

# Basic Snakemake rule

```
rule sort:
    input:
        'path/to/dataset.txt'
    output:
        'dataset.sorted.txt'
    shell:
        'sort {input} > {output}'
```

# More complicated Snakemake rule

```
rule align_rnaseq_with_star_single_end:
    '''Align fastq file with star'''
    input: fastq='fastq_files/{sample}.fq.gz',
           index_sa='STAR_index_{genome}_{txome}/SA',
           transcriptome_gff='{txome}.gff3',
    output: bam='aligned/rnaseq_star_{genome}_{txome}/{sample}/Aligned.sortedByCoord.out.bam',
            splice_junctions='aligned/rnaseq_star_{genome}_{txome}/{sample}/SJ.out.tab',
            transcriptome_bam='aligned/rnaseq_star_{genome}_{txome}/{sample}/Aligned.toTranscriptome.out.bam',
            gene_counts='aligned/rnaseq_star_{genome}_{txome}/{sample}/ReadsPerGene.out.tab',
    params: temp_sam='aligned/rnaseq_star_{genome}_{txome}/{sample}/Aligned.out.sam',
    threads: 8
    run: [CODE]
```

# Wildcards

Example:

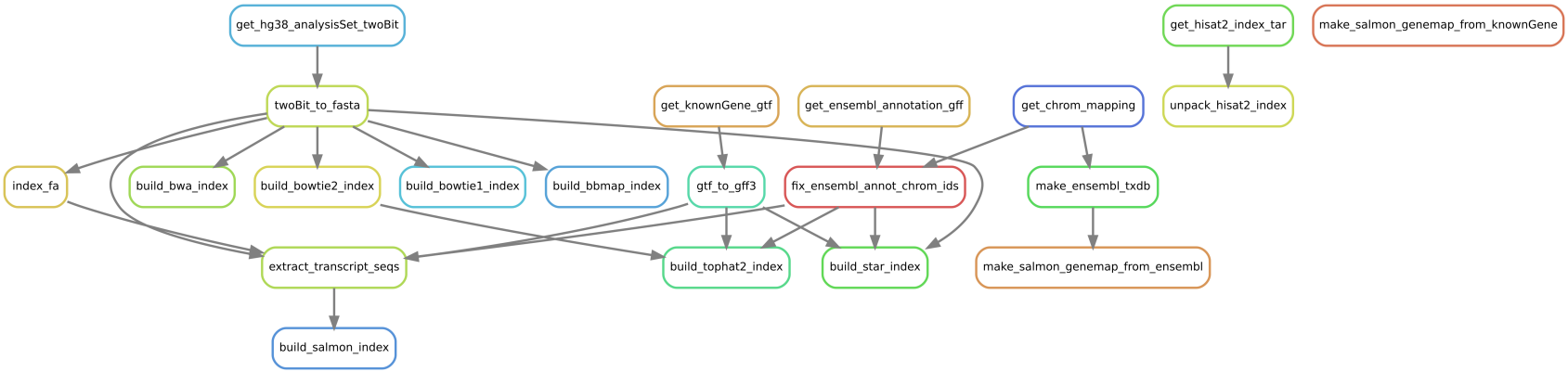
- Wildcards:
  - `sample`
  - `genome`
  - `txome`
- Input:
  - `fastq_files/{sample}.fq.gz`
  - `STAR_index_{genome}_{txome}/SA`
  - `{txome}.gff3`
- Output in `aligned/rnaseq_star_{genome}_{txome}/{sample}`:
  - `Aligned.sortedByCoord.out.bam`
  - `SJ.out.tab`
  - `Aligned.toTranscriptome.out.bam`
  - `ReadsPerGene.out.tab`

# Wildcards

Example:

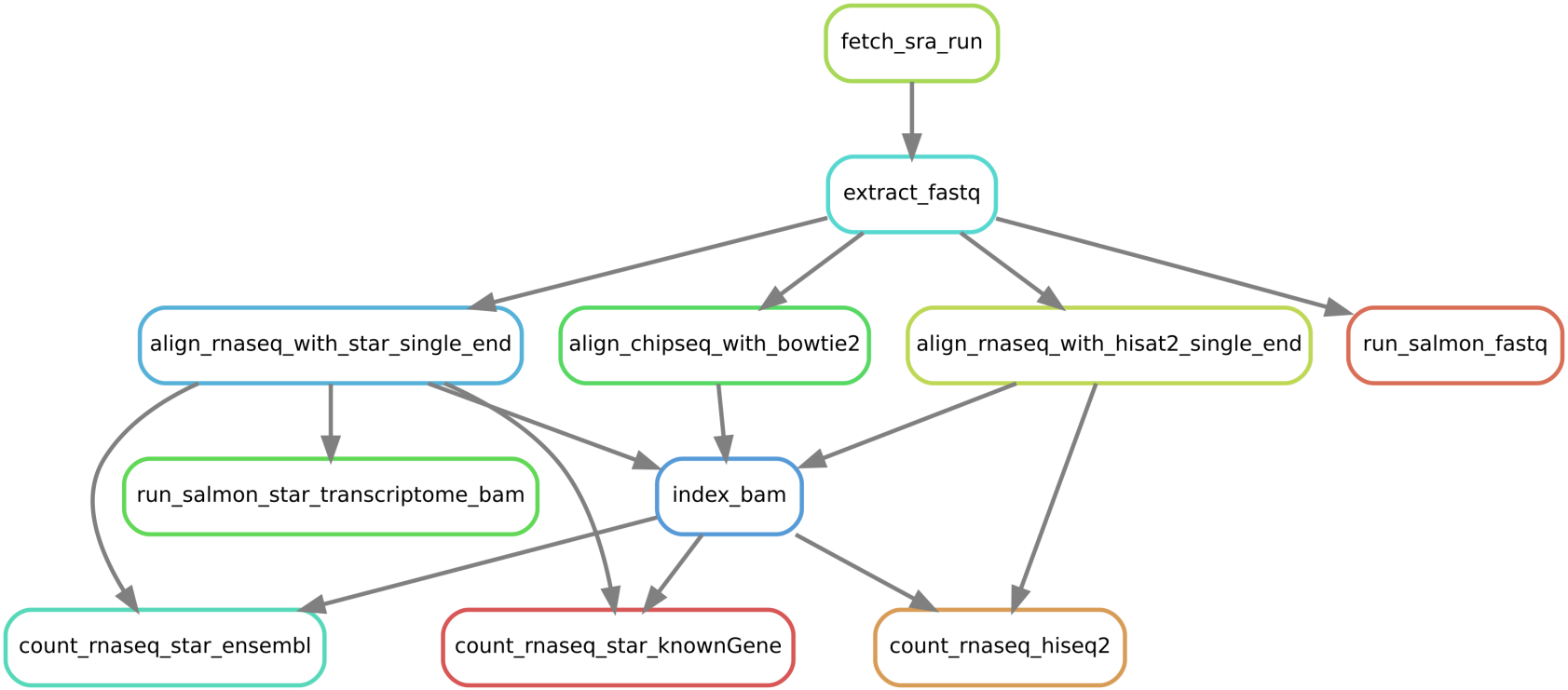
- Wildcards:
  - `sample='Sample1'`
  - `genome='hg38'`
  - `txome='ensembl.85'`
- Input:
  - `fastq_files/Sample1.fq.gz`
  - `STAR_index_hg38_ensembl.85/SA`
  - `ensembl.85.gff3`
- Output in `aligned/rnaseq_star_hg38_ensembl.85/Sample1`:
  - `Aligned.sortedByCoord.out.bam`
  - `SJ.out.tab`
  - `Aligned.toTranscriptome.out.bam`
  - `ReadsPerGene.out.tab`

# Workflow 1: Preparing indices and annotations for hg38




`https://github.com/DarwinAwardWinner/hg38-ref`

# Workflow 2: Data analysis




`https://github.com/DarwinAwardWinner/CD4-csaw`