SETTING UP A BIOINFORMATICS QC PIPELINE

BRIAN MCCONEGHY

BIOINFORMATICS SPECIALIST

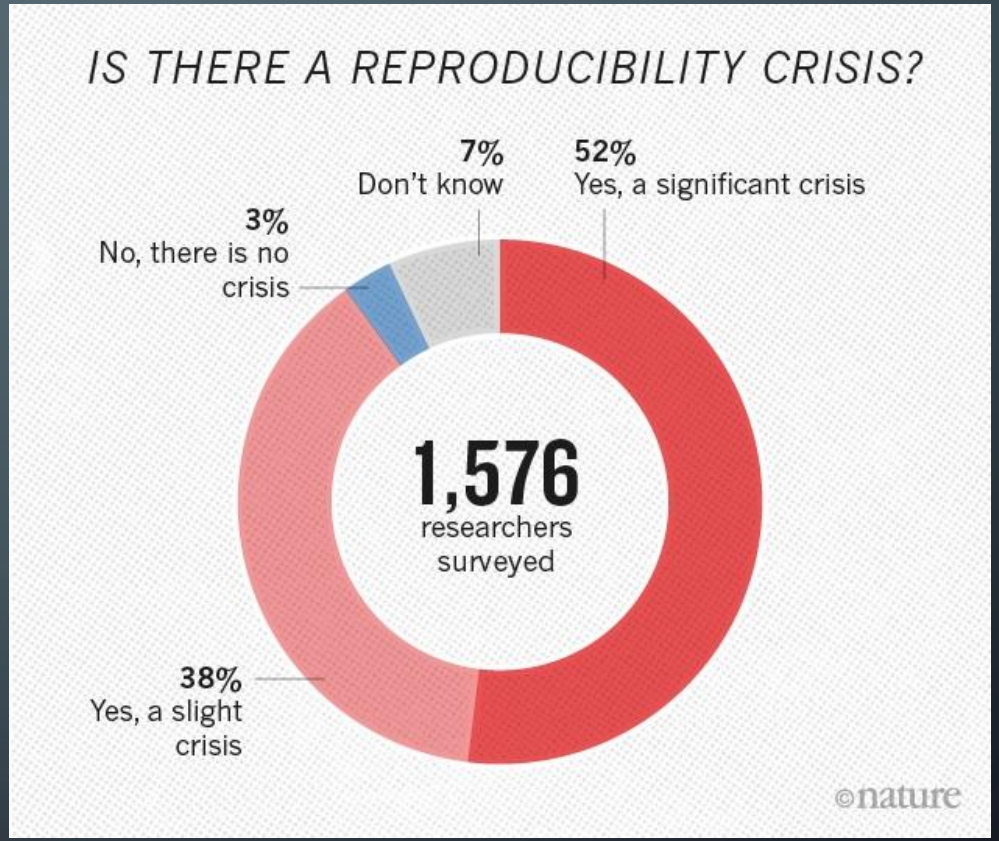
SEQUENCING AND BIOINFORMATICS CONSORTIUM, UBC

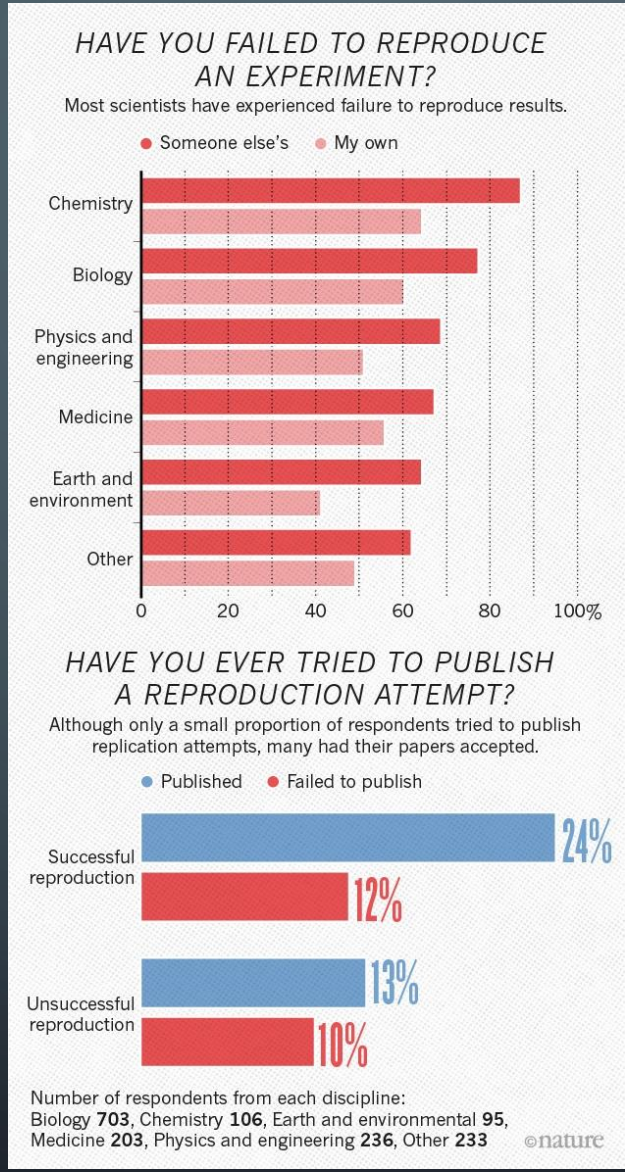
OFFICE OF THE VICE-PRESIDENT, RESEARCH & INNOVATION

WESTGRID WEBINAR 2019-11-13

"More than 70% of researchers have tried and failed to reproduce another scientist's experiments, and more than half have failed to reproduce their own experiments."

Baker, M. (2016). 1,500 scientists lift the lid on reproducibility. *Nature*, 533(7604), 452-454. doi:10.1038/533452a





Baker, M. (2016). 1,500 scientists lift the lid on reproducibility. *Nature*, 533(7604), 452-454. doi:10.1038/533452a

Background

Pipelining Tools

Writing the Pipeline

Metrics to Track

Implementation

Conclusions

Background

Pipelining Tools

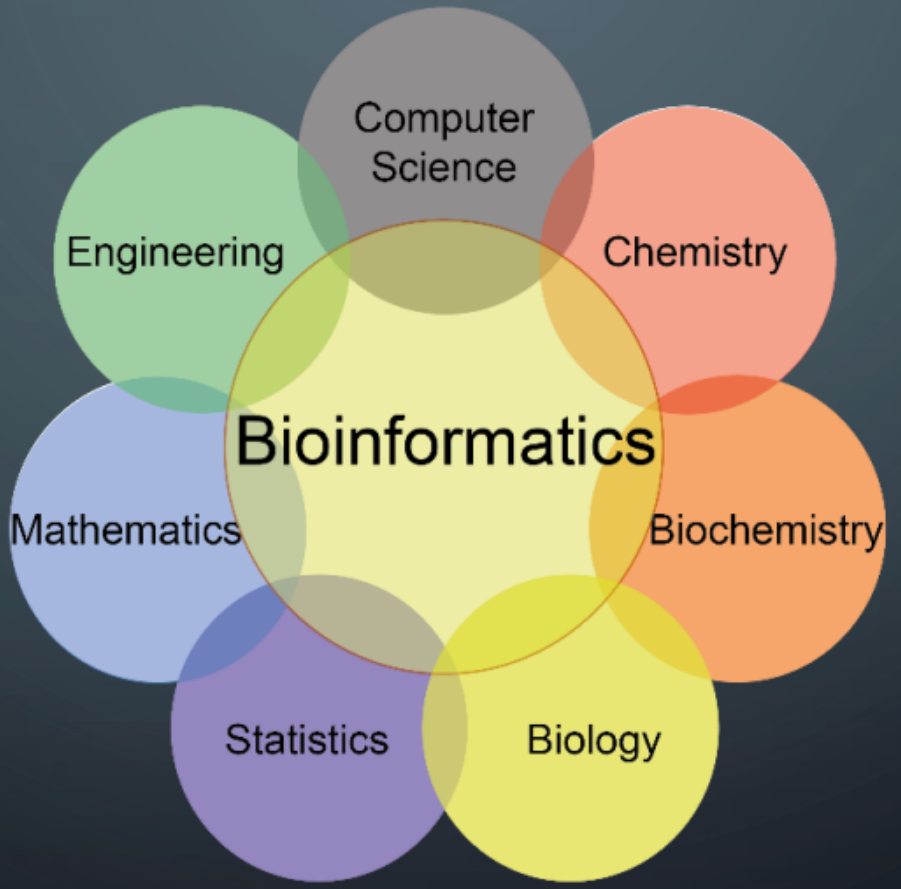
Writing the Pipeline

Metrics to Track

Implementation

Conclusions

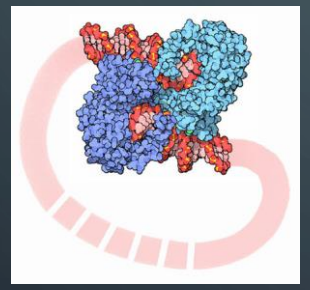
WHAT IS BIOINFORMATICS?



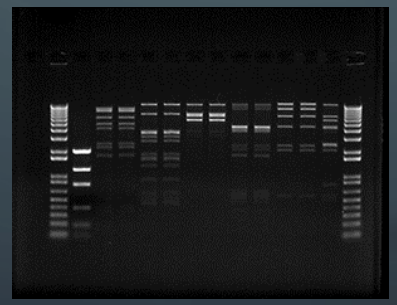
NEXT GENERATION SEQUENCING

Sample (input) QC → Sample Preparation → Sample (library) QC → Sequencing → Data!

https://www.nextadvance.com/



https://pdb101.rcsb.org/motm/84



www.thermofisher.com

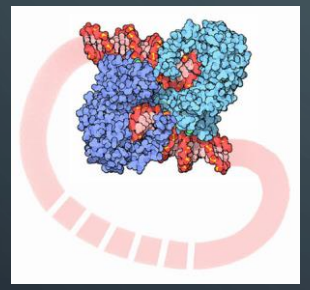
www.Illumina.com

https://www.makeuseof.com/tag/best-linux-server-operating-systems/

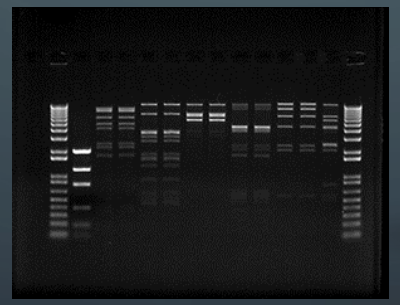
NEXT GENERATION SEQUENCING

Sample (input) QC → Sample Preparation → Sample (library) QC → Sequencing → Data!

https://www.nextadvance.com/



https://pdb101.rcsb.org/motm/84



www.thermofisher.com

www.Illumina.com

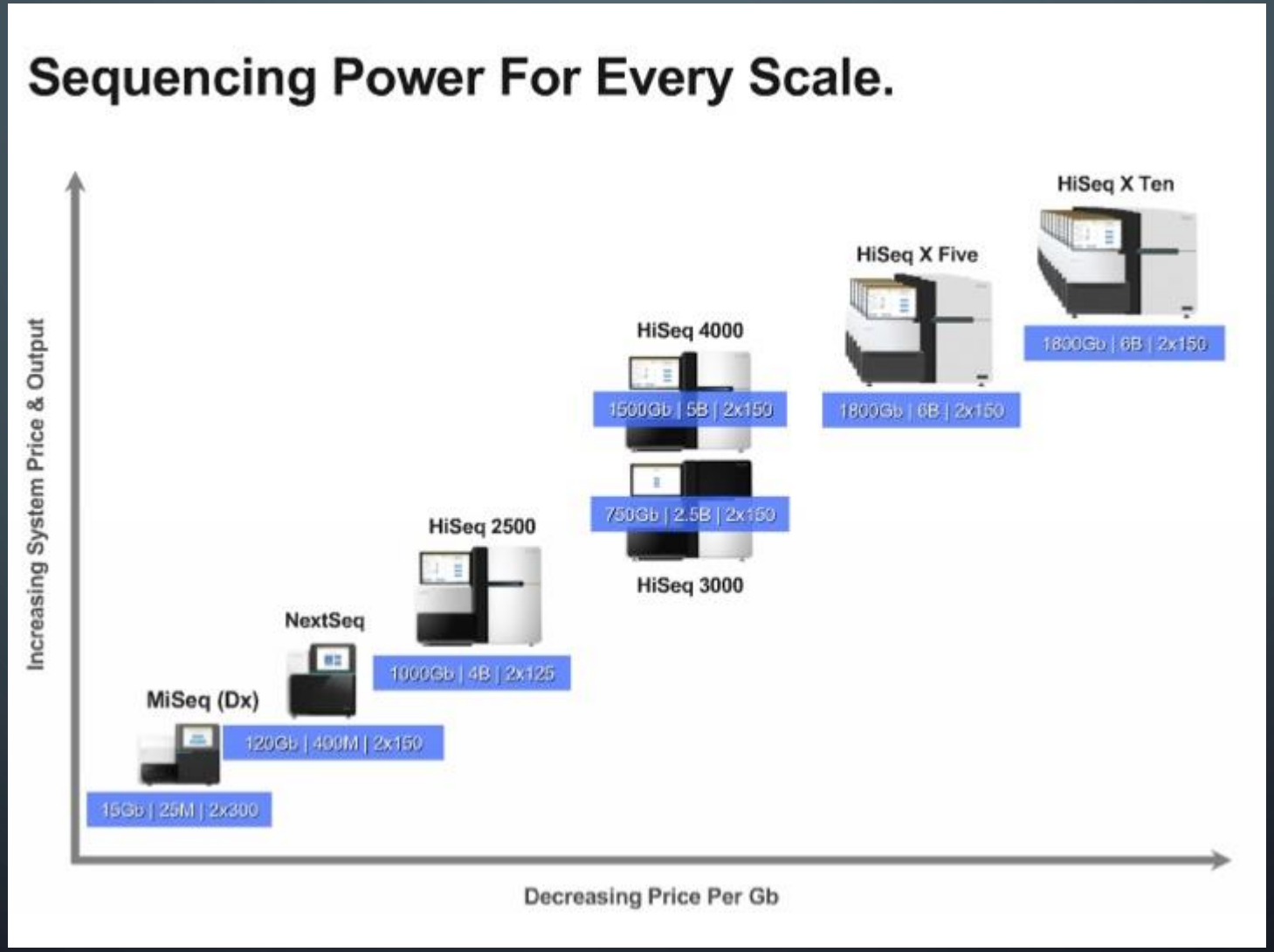
https://www.makeuseof.com/tag/best-linux-server-operating-systems/

WHAT *IS* NEXT GEN SEQUENCING DATA, EXACTLY?

- High-throughput sequencing technology
  - Generates **millions** of 'reads'
  - Reads are just strings of G's, A's, T's, and C's (with associated quality values)

```
@NB999999:999:ZZZZZZZZ9:1:11101:16570:1094 1:N:0:1
AAAGCNGCTGAATTGTTCGCGTTTACCTTGCGTGTACGCGCAGGAAACACT
+
AAAAA#A6EA66EEEAEEE/E//EAE/E//EEEEEEEEAAEEEAEEEEAEE
```

phiX 174 control DNA



https://bloggenohub.files.wordpress.com/2015/01/slide1.jpg

Background

Pipelining Tools

Writing the Pipeline

Metrics to Track

Implementation

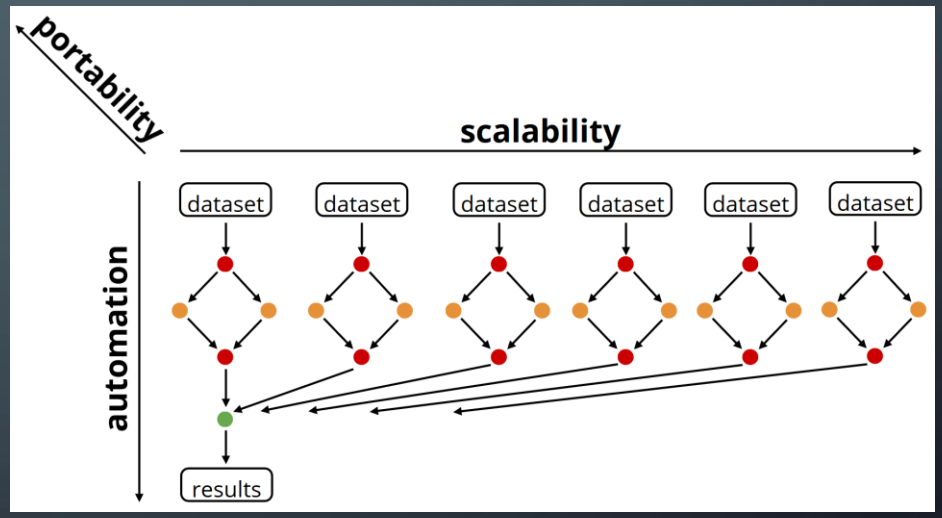
Conclusions

WHAT AND WHY

- Workflow management system
- Reproducible and scalable data analysis
- Rules, inputs, and outputs

NEEDS

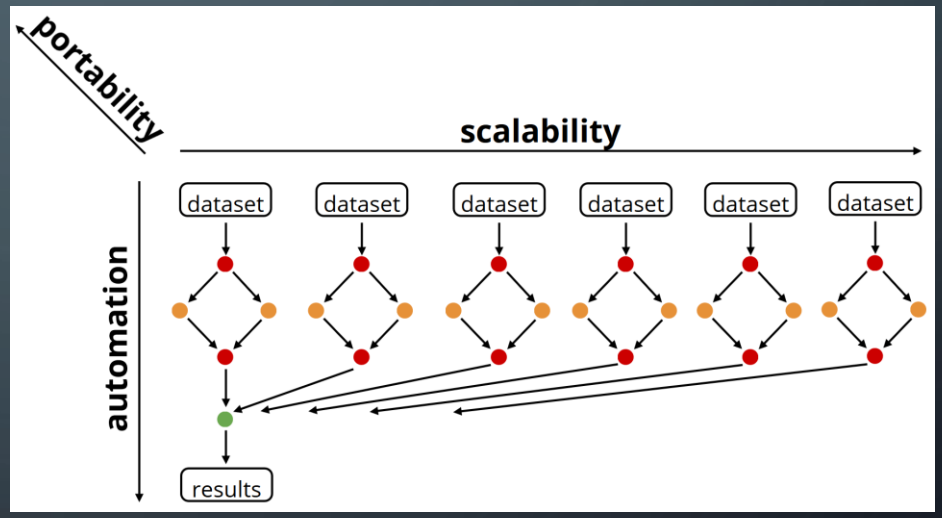
- Reproducible
- Scalable
- Efficient (Parallelizable)
- Portable
- Automated



https://slides.com/johanneskoester/snakemake-short#/3

WANTS

- Ease of development
- Unix-compatible
- FREE



https://slides.com/johanneskoester/snakemake-short#/3

COMPARISON

- Galaxy
- Ruffus
- Snakemake

COMPARISON

- Galaxy
- Ruffus
- **Snakemake**

Background

Pipelining Tools

Writing the Pipeline

Metrics to Track

Implementation

Conclusions

RESOURCES - HARDWARE

- Cedar - Compute Canada
  - 58,416 Cores
  - 306,306 GB of RAM
  - 10TB scratch space

https://medium.com/monplan/how-we-automated-deployments-and-testing-with-bitbucket-pipelines-bb478c12c55f

RESOURCES - PEOPLE

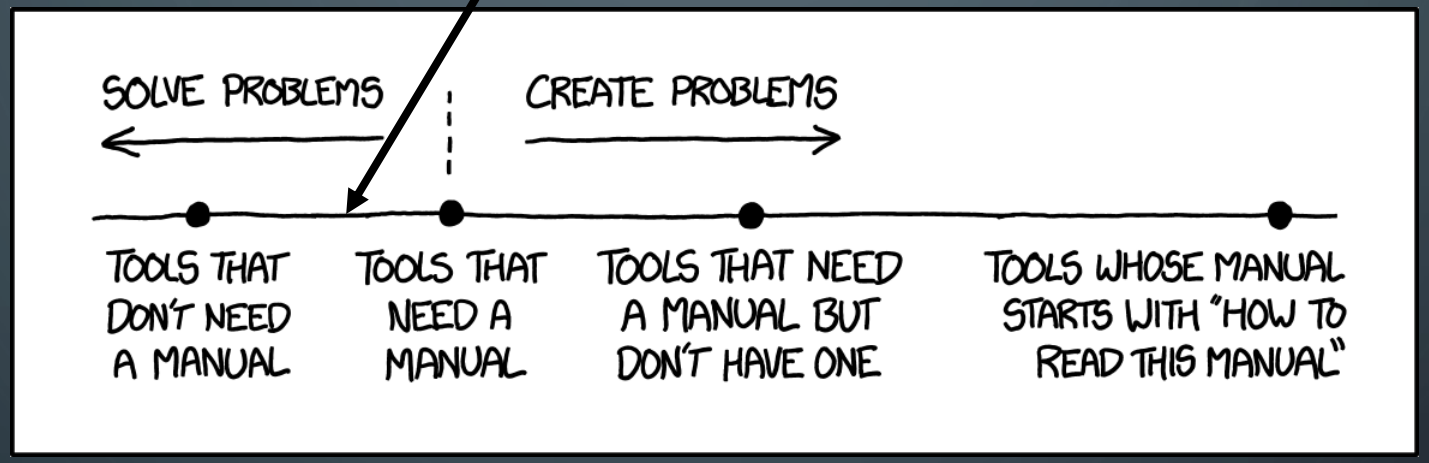
- Advanced Research Computing (ARC)
  - Jamie Rosner
  - Venkat Mahadevan
- VP Research & Innovation (VPRI)
  - Dr. Helen Burt

https://medium.com/monplan/how-we-automated-deployments-and-testing-with-bitbucket-pipelines-bb478c12c55f

SNAKEMAKE

- Decompose workflow into rules
- Rules define how to obtain output files from input files
- Snakemake infers dependencies and execution order

SNAKEMAKE



https://xkcd.com/1343/


And now
for something
completely TECHNICAL ...

SIMPLICITY!

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

GENERALIZE RULES WITH NAMED **WILDCARDS**

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

SPECIFY MULTIPLE INPUTS (AND OUTPUTS) REFER BY **INDEX**

```
rule sort_and_annotate:
    input:
        "path/to/{dataset}.txt",
        "path/to/annotation.txt"
    output:
        "{dataset}.sorted.txt"
    shell:
        "paste <(sort {input[0]}) {input[1]} > {output}"
```

CAN SPECIFY MULTIPLE INPUTS (AND OUTPUTS), AND REFER BY **NAME**

```
rule sort_and_annotate:
    input:
        a="path/to/{dataset}.txt",
        b="path/to/annotation.txt"
    output:
        "{dataset}.sorted.txt"
    shell:
        "paste <(sort {input.a}) {input.b} > {output}"
```

USE PYTHON WITHIN RULES

```
rule sort:
    input:
        a="path/to/{dataset}.txt"
    output:
        b="{dataset}.sorted.txt"
    run:
        with open(output.b, "w") as out:
            for l in sorted(open(input.a)):
                print(l, file=out)
```

A **REAL** RULE

- Used in DNA QC pipeline

```
rule bwa_mem_map_reads:
    input:
        get_trimmed_reads
    output:
        temp('mapped/{sample}-{unit}.sorted.bam')
    log:
        'logs/bwa_mem/{sample}-{unit}.log'
    params:
        index = get_genome_index,
        rg = get_read_group_bwa
    threads: 46
    shell:
        '(bwa mem -t {threads} {params.rg} {params.index} {input} | '
        'samtools sort -T $SLURM_TMPDIR/ -o {output} -) 2> {log}'
```

JOB EXECUTION

- A job only executes if:
  1. output file is the target requested and does not exist
  2. output file needed by another executed job (i.e. is an input to another job) and does not exist
  3. input file is newer than the output file
  4. input file will be updated by other job
  5. execution is forced

CLUSTER EXECUTION

- Can set up pipeline **profiles**
- Execute DAG by way of cluster job submission
- Configuration file
  - Max jobs at a time
  - CPUs
  - MEM
  - General (per profile) or granular (per rule)

Background

Pipelining Tools

Metrics to Track

Writing the Pipeline

Implementation

Conclusions

METRICS TO TRACK

- Adapter trimming
- Duplicate Rate
- % Aligned (for genomes we can map to)
- Insert size
- Coverage
- Error rate
- GC Content
- For RNA, specifically:
  - Strand specificity (% correct strand)
  - 5'-3' bias
  - % rRNA
  - Intron-exon ratio

Background

Pipelining Tools

Writing the Pipeline

Metrics to Track

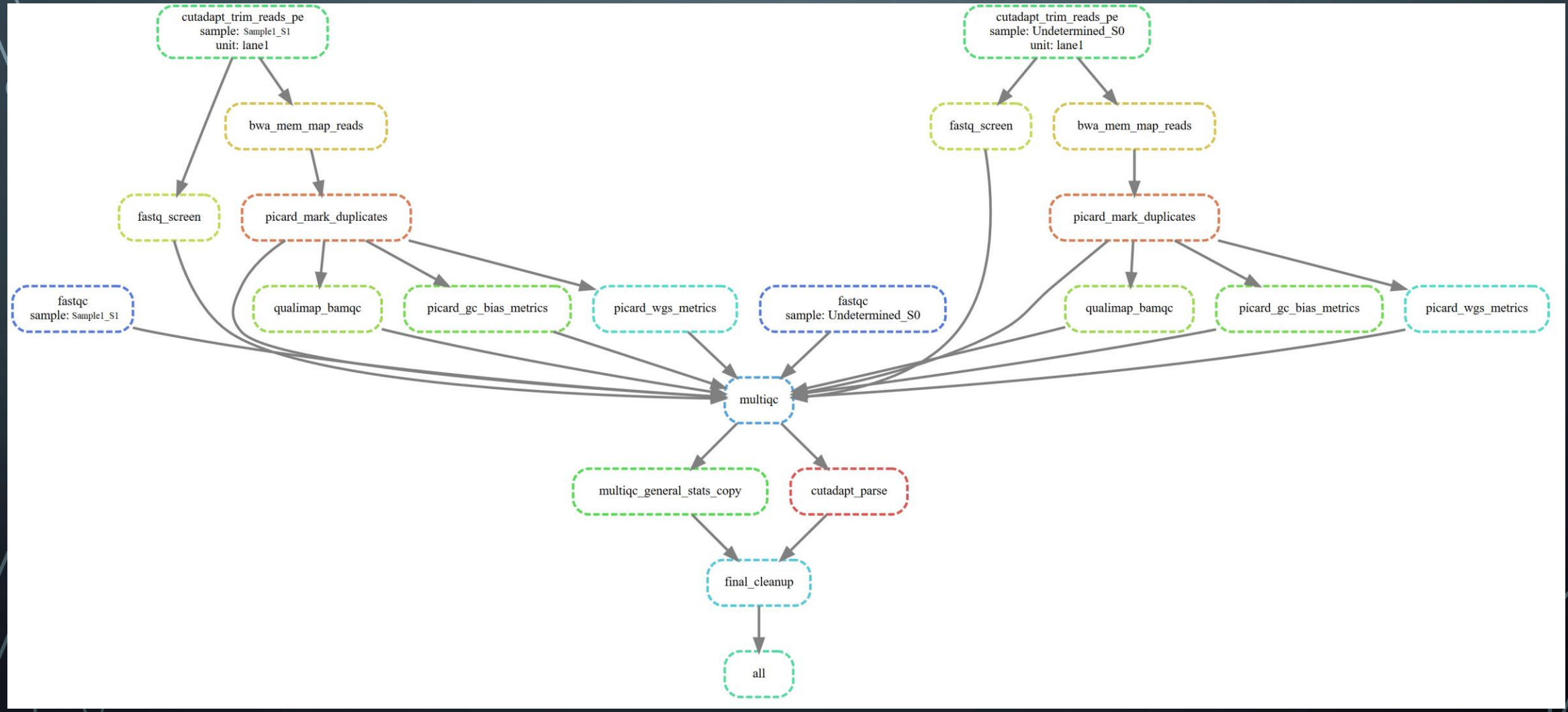
Implementation

Conclusions

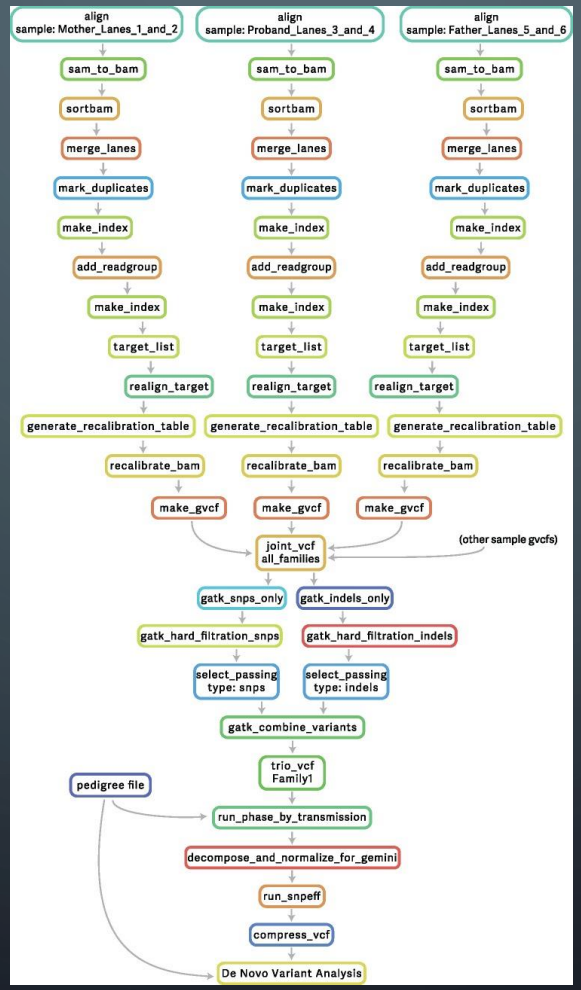
IMPLEMENTATION

- Conda environment
- Version controlled - **GitHub**
- SBC has 3 QC pipelines (combined into 1, dynamically determined)
  - Paired-end DNA QC
  - Single-end DNA QC
  - Paired-end RNA QC

DNA QC WORKFLOW – 2 SAMPLES



WORKFLOWS CAN BE COMPLEX



Leipzig, J. (2016). A review of bioinformatic pipeline frameworks. *Briefings in Bioinformatics*. doi:10.1093/bib/bbw020

Background

Pipelining Tools

Writing the Pipeline

Metrics to Track

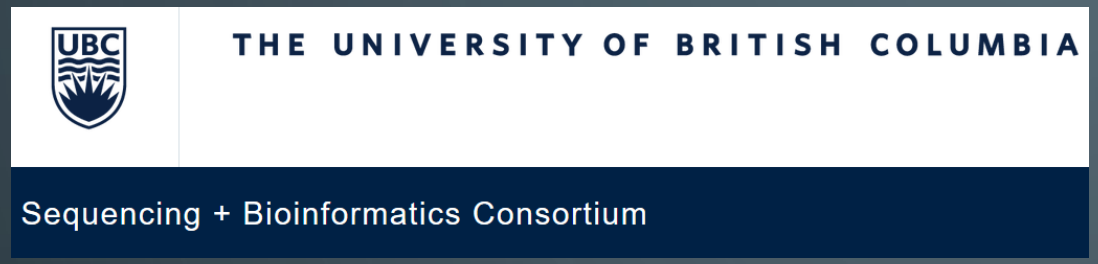
Implementation

Conclusions

CONCLUSIONS

- Snakemake satisfied all needs of the SBC and is simple to work with
- The complex metrics the SBC is most interested in are being tracked, in an automated fashion
- Implementation allows for reproducible, scalable, flexible, trackable QC

THANK YOU



Link to GitHub with workshop instructions:

https://bit.ly/2Xf4HN6