



SETTING UP A BIOINFORMATICS QC PIPELINE

BRIAN MCCONEGHY

BIOINFORMATICS SPECIALIST

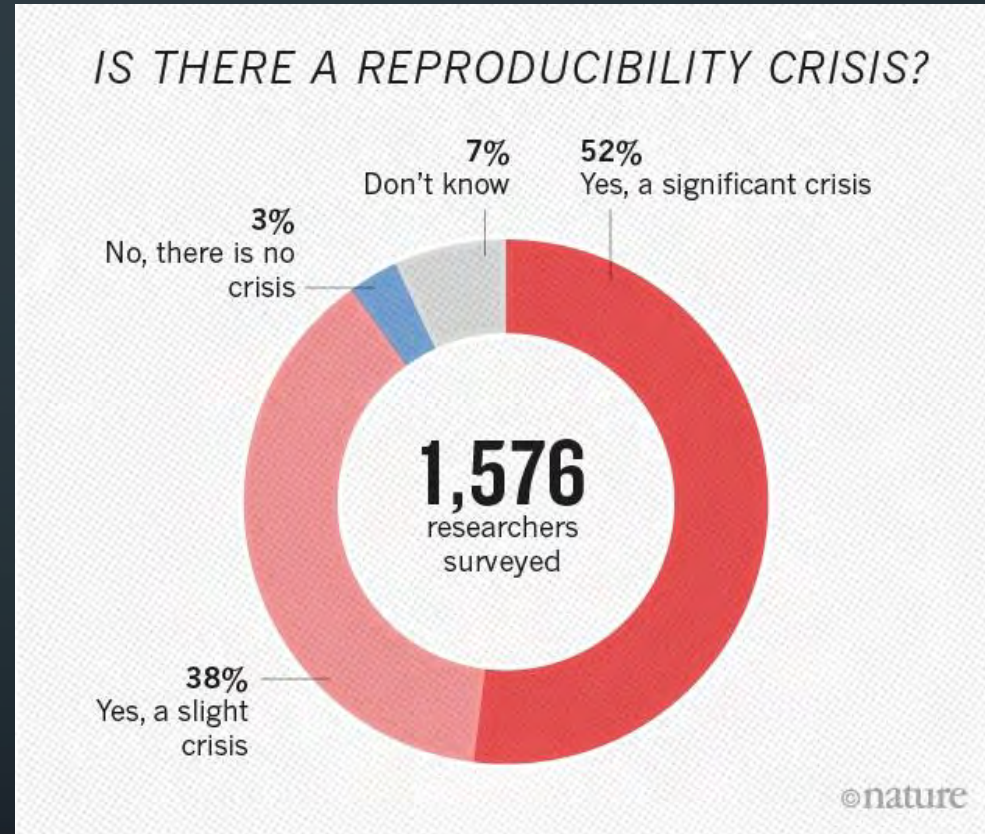
SEQUENCING AND BIOINFORMATICS CONSORTIUM, UBC

OFFICE OF THE VICE-PRESIDENT, RESEARCH & INNOVATION

BCNET CONFERENCE – 2019-05-02

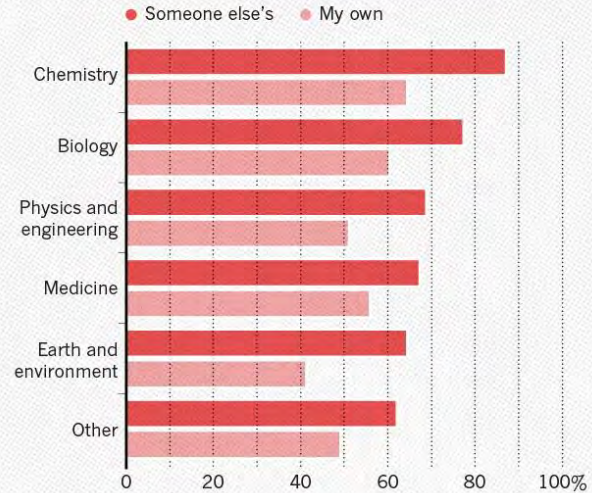
“More than 70% of researchers have tried and failed to reproduce another scientist's experiments, and more than half have failed to reproduce their own experiments.”

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



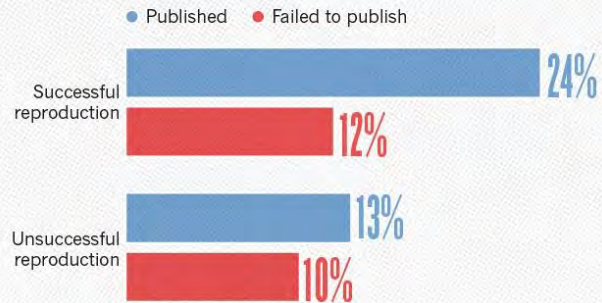
HAVE YOU FAILED TO REPRODUCE AN EXPERIMENT?


Most scientists have experienced failure to reproduce results.



HAVE YOU EVER TRIED TO PUBLISH A REPRODUCTION ATTEMPT?

Although only a small proportion of respondents tried to publish replication attempts, many had their papers accepted.



Number of respondents from each discipline:
Biology 703, Chemistry 106, Earth and environmental 95,
Medicine 203, Physics and engineering 236, Other 233 



Background

Pipelining Tools

Writing the Pipeline

Metrics to Track

Implementation

Conclusions





Background

Pipelining Tools

Writing the Pipeline

Metrics to Track

Implementation

Conclusions



BACKGROUND – NEXT GENERATION SEQUENCING

Data!

Sample (input) QC

Sample Preparation

Sample (library) QC

Sequencing



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

BACKGROUND – NEXT GENERATION SEQUENCING

- 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:ZZZZZZZ9:1:11101:16570:1094 1:N:0:1  
AAAGCNGCTGAATTGTTTCGCGTTTACCTTGCGTGTACGCGCAGGAAACACT  
+  
AAAAA#A6EA66EEEEEEE/E//EAE/E//EEEEEEEEAAEEEEEEEEEEE
```

phiX 174 control DNA



Background

Pipelining Tools

Writing the Pipeline

Metrics to Track

Implementation

Conclusions



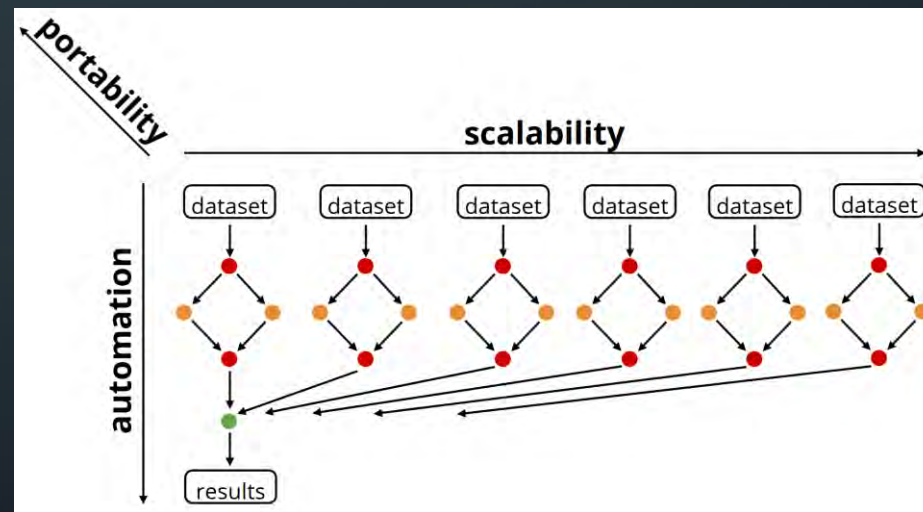
PIPELINING TOOLS – WHAT AND WHY

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



PIPELINING TOOLS – NEEDS AND WANTS

- Needs:
 - Reproducible
 - Scalable
 - Efficient (Parallelizable)
 - Portable
 - Automated
- Wants:
 - Ease of development
 - Unix-compatible
 - FREE



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

PIPELINING TOOLS - COMPARISON

- Galaxy
- Ruffus
- Nextflow
- Snakemake

PIPELINING TOOLS - COMPARISON

- Galaxy
- Ruffus
- Nextflow
- **Snakemake**



Background

Pipelining Tools

Writing the Pipeline

Metrics to Track

Implementation

Conclusions



WRITING THE PIPELINE - RESOURCES

- Cedar - Compute Canada
 - 58,416 Cores
 - 306,306 GB of RAM
- Advanced Research Computing (ARC)
 - Jamie Rosner
 - Venkat Mahadevan
- Documentation (lots of documentation...)



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

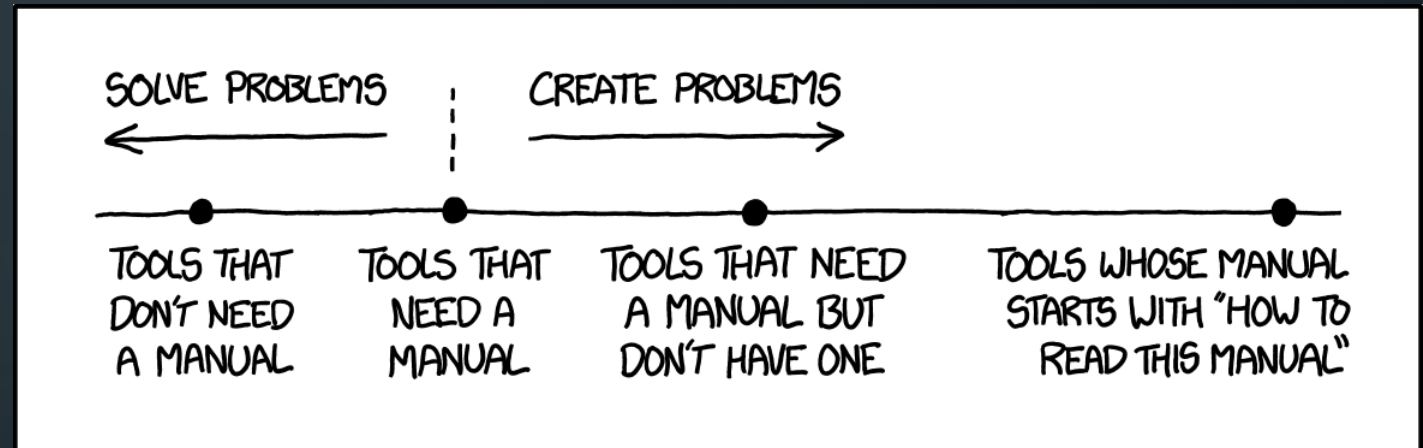
WRITING THE PIPELINE - SNAKEMAKE

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

WRITING THE PIPELINE - EXAMPLE RULE

- SIMPLICITY!

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



<https://xkcd.com/1343/>

WRITING THE PIPELINE - EXAMPLE RULE

- Generalize rules with named wildcards

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

WRITING THE PIPELINE - EXAMPLE RULE

- Can specify multiple inputs (and outputs), and 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}"
```

WRITING THE PIPELINE - EXAMPLE RULE

- 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}"
```

WRITING THE PIPELINE - EXAMPLE RULE

- Can use python within rules (using the run directive)

```
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)
```

WRITING THE PIPELINE – A REAL RULE

- 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=config['ref']['genome'],
        rg=get_read_group
    threads: 46
    shell:
        '(bwa mem -t {threads} {params.rg} {params.index} {input} |
        samtools sort -T $SLURM_TMPDIR/ -o {output} -) 2> {log}'
```

WRITING THE PIPELINE – 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

WRITING THE PIPELINE – CLUSTER EXECUTION

- Can set up pipeline profiles
- Execute DAG by way of cluster job submission
- Config to determine all parameters, such as maximum jobs submitted at a time, CPUs, MEM
 - Can be 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

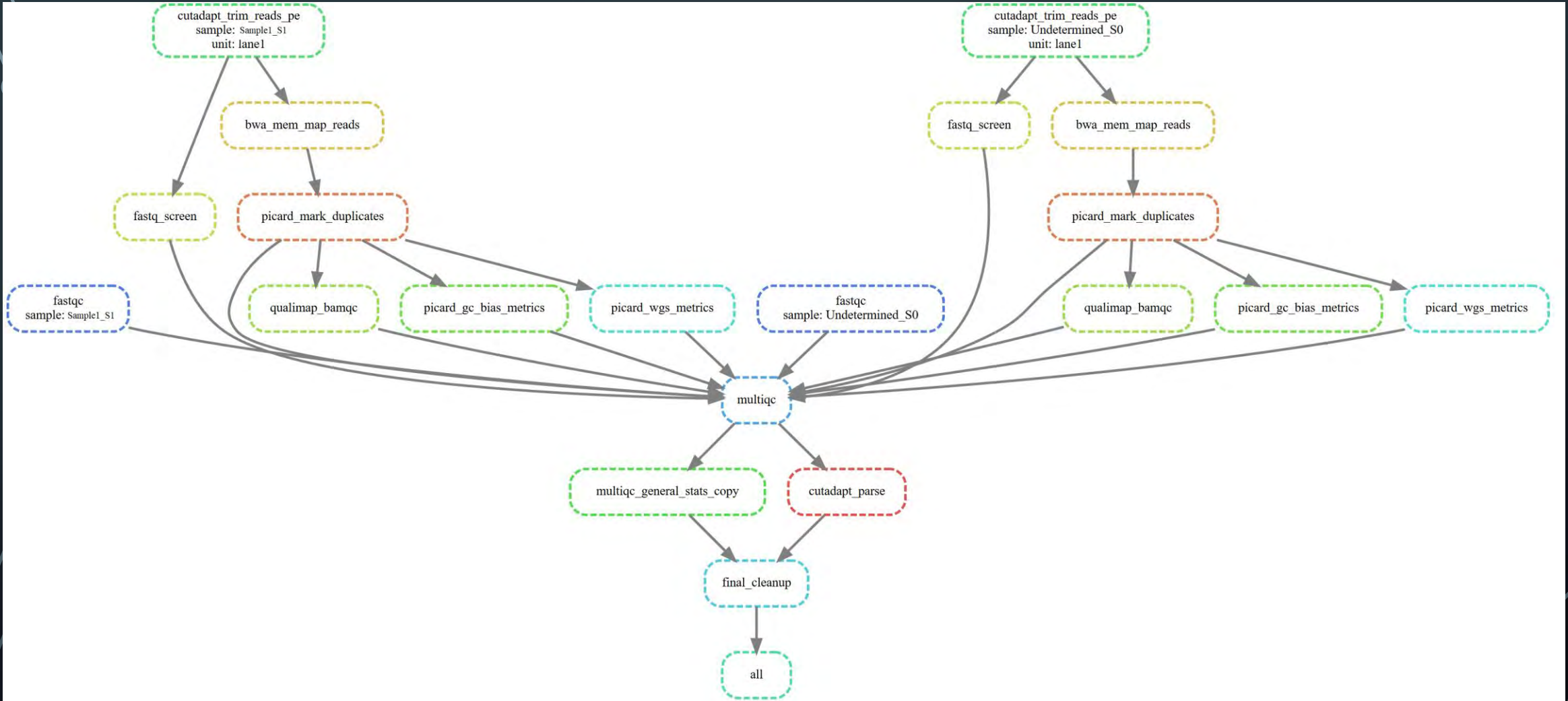
Conclusions



IMPLEMENTATION

- Conda environment
- Version controlled with GitHub (privately)
- Set up 3 pipelines so far:
 - Paired-end DNA QC
 - Single-end DNA QC
 - Paired-end RNA QC

DNA QC WORKFLOW – 2 SAMPLES



WORKFLOWS CAN BE COMPLEX





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



THE UNIVERSITY OF BRITISH COLUMBIA

Sequencing + Bioinformatics Consortium