Distributed multi-platform variant calling with bcbio and the Common Workflow Language

Brad Chapman
Bioinformatics Core, Harvard Chan School
https://bcb.io
http://j.mp/bcbiolinks

25 January 2018

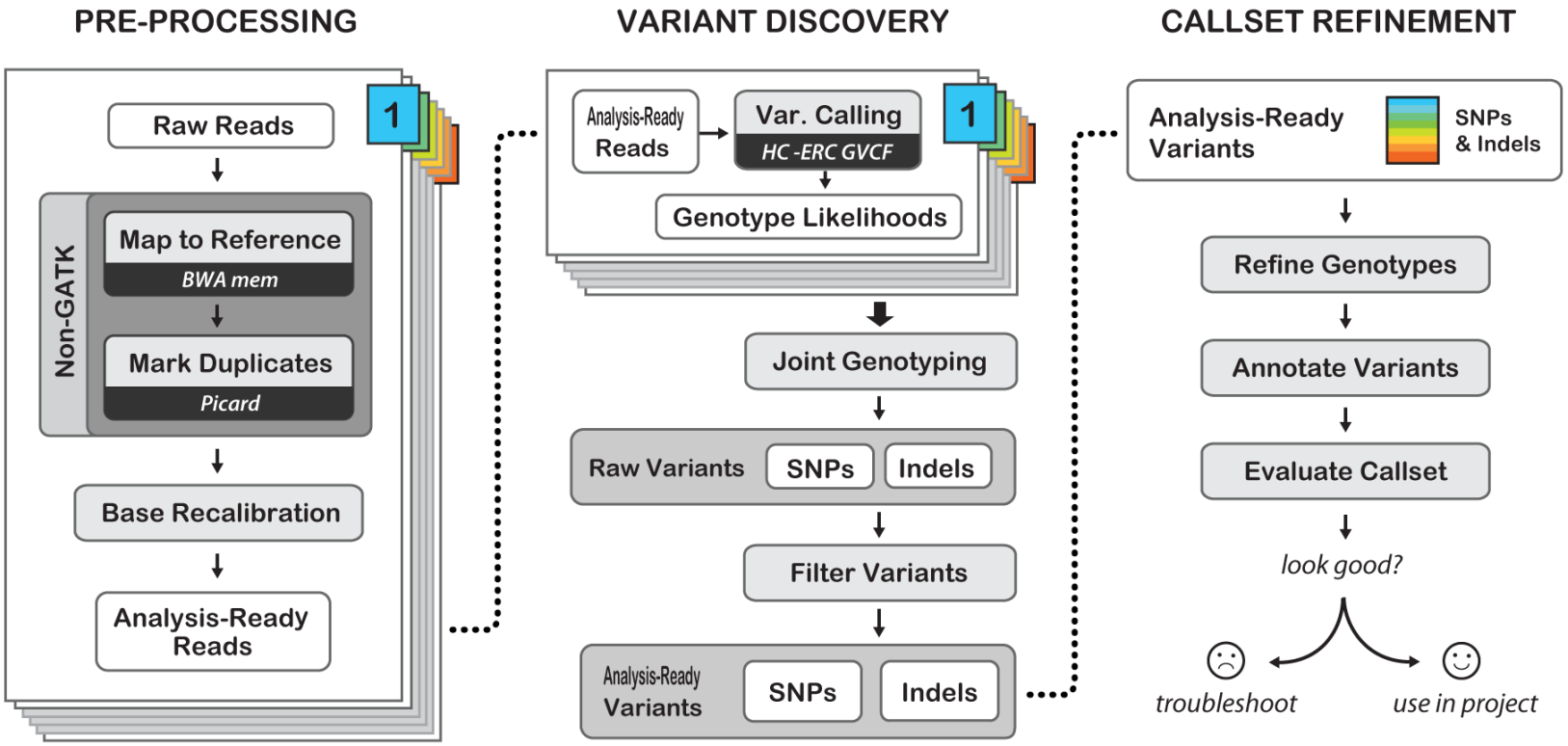
Overview

- Barriers to building analysis pipelines
- bcbio: open source community development
- Common Workflow Language (CWL) and Workflow Description Language (WDL): assembly language for workflows
- Practical CWL with bcbio: HPC, DNAnexus, Arvados, SevenBridges
- Scaling and parallelization
- GA4GH: Automating validation and multi-platform testing

Takeaways

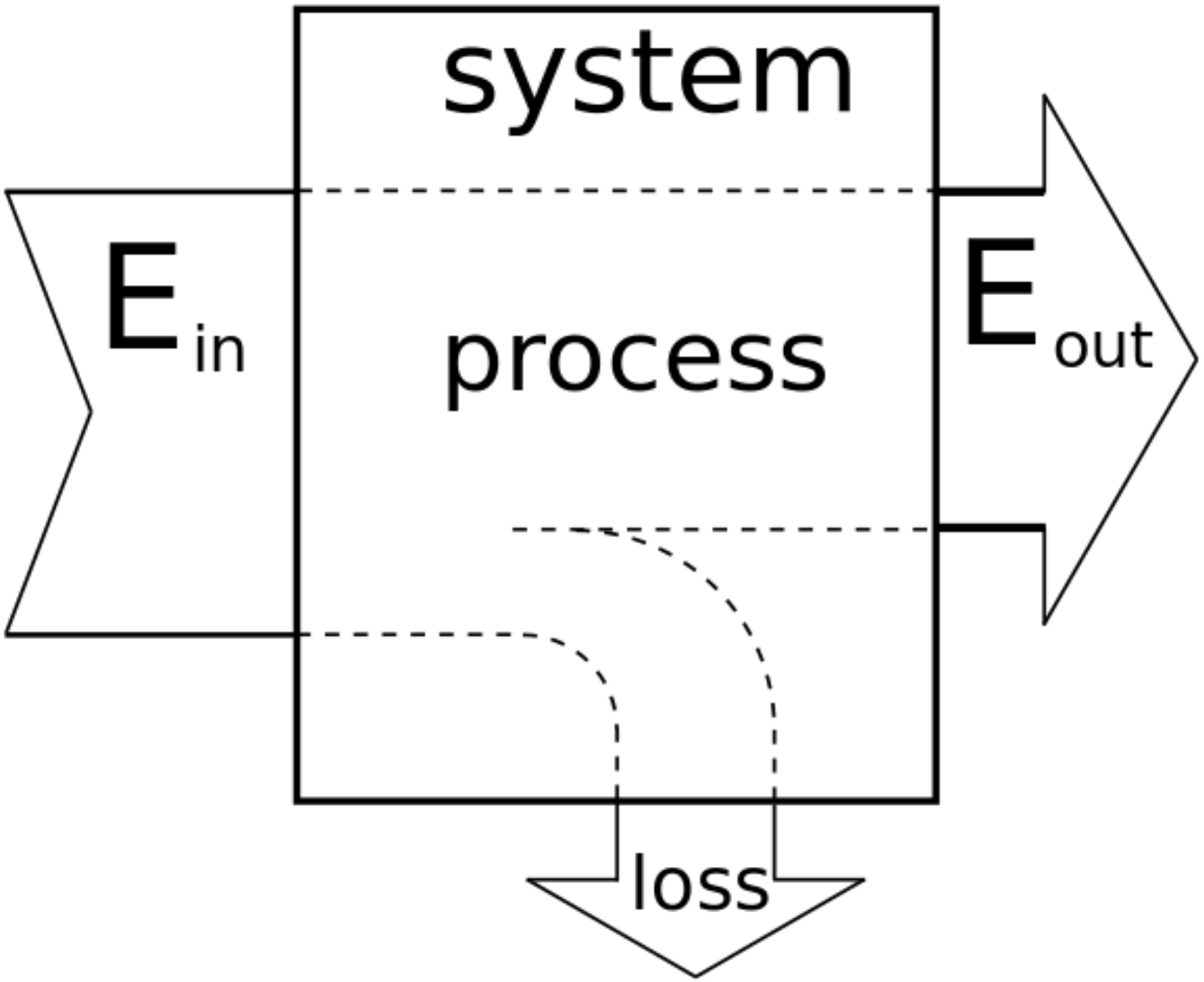
- Science = collaboration and re-use
- Need interoperable workflow abstractions
- We can build better things together

You want to build a variant calling pipeline



Best Practices for Germline SNPs and Indels in Whole Genomes and Exomes - June 2016

https://software.broadinstitute.org/gatk/best-practices/



https://commons.wikimedia.org/wiki/File:Efficiency_diagram_by_Zureks.svg

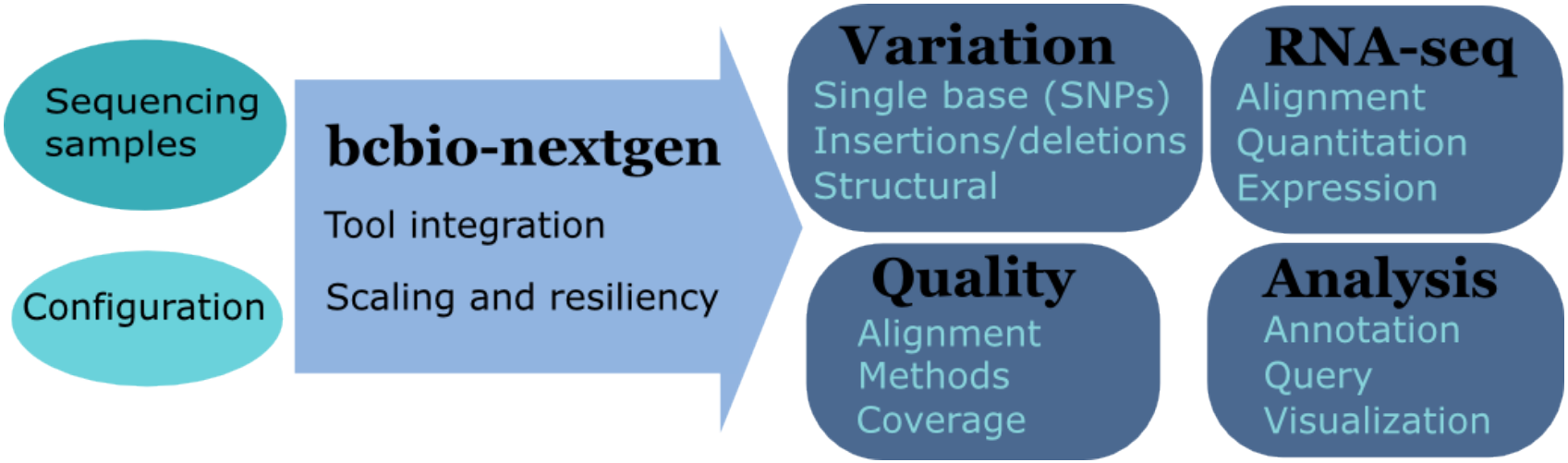
Barriers to implementing yourself

- Changing tools
- Feature support burden
- Multi-platform interoperability
- Validation

Build open source communities

https://gccbosc2018.sched.com/

Overview



https://github.com/chapmanb/bcbio-nextgen

Supported analysis types

- ⊟ **Pipelines**
  - ⊟ **Germline variant calling**
    - Basic germline calling
    - Population calling
  - Cancer variant calling
  - Structural variant calling
  - RNA-seq
  - single-cell RNA-seq
  - smallRNA-seq
  - ChIP-seq

https://bcbio-nextgen.readthedocs.org/en/latest/contents/pipelines.html

We made a pipeline – so what?

There have been a number of previous efforts to create publicly available analysis pipelines for high throughput sequencing data. Examples include Omics-Pipe, bcbio-nextgen, TREVA and NGSane. These pipelines offer a comprehensive, automated process that can analyse raw sequencing reads and produce annotated variant calls. However, the main audience for these pipelines is the research community. Consequently, there are many features required by clinical pipelines that these examples do not fully address. Other groups have focused on improving specific features of clinical pipelines. The Churchill pipeline uses specialised techniques to achieve high performance, while maintaining reproducibility and accuracy. However it is not freely available to clinical centres and it does not try to improve broader clinical aspects such as detailed quality assurance reports, robustness, reports and specialised variant filtering. The Mercury pipeline offers a comprehensive system that addresses many clinical needs: it uses an automated workflow system (Valence) to ensure robustness, abstract computational resources and simplify customisation of the pipeline. Mercury also includes detailed coverage reports provided by ExCID, and supports compliance with US privacy laws (HIPAA) when run on DNANexus, a cloud computing platform specialised for biomedical users. Mercury offers a comprehensive solution for clinical users, however it does not achieve our desired level of transparency, modularity and simplicity in the pipeline specification and design. Further, Mercury does not perform specialised variant filtering and prioritisation that is specifically tuned to the needs of clinical users.

http://www.genomemedicine.com/content/7/1/68

Sustainability

A piece of software is being sustained if people are using it, fixing it, and improving it rather than replacing it.

http://software-carpentry.org/blog/2014/08/sustainability.html

Complex, rapidly changing baseline functionality

Whole genome, deep coverage v1

Best Practice Variant Detection with the GATK v2

RETIRED: Best Practice Variant Detection with the GATK v3

Best Practice Variant Detection with the GATK v4, for release 2.0 [RETIRED]

Mark_DePristo Posts: 153 Administrator, GSA Member admin

July 2012 edited February 4 in Methods and Workflows

The Best Practices have been updated for GATK version 3. If you are running an older version, you should seriously consider upgrading. For more details

GATK 4.0 will be released Jan 9, 2018

Posted by Geraldine_VdAuwera on 16 Oct 2017

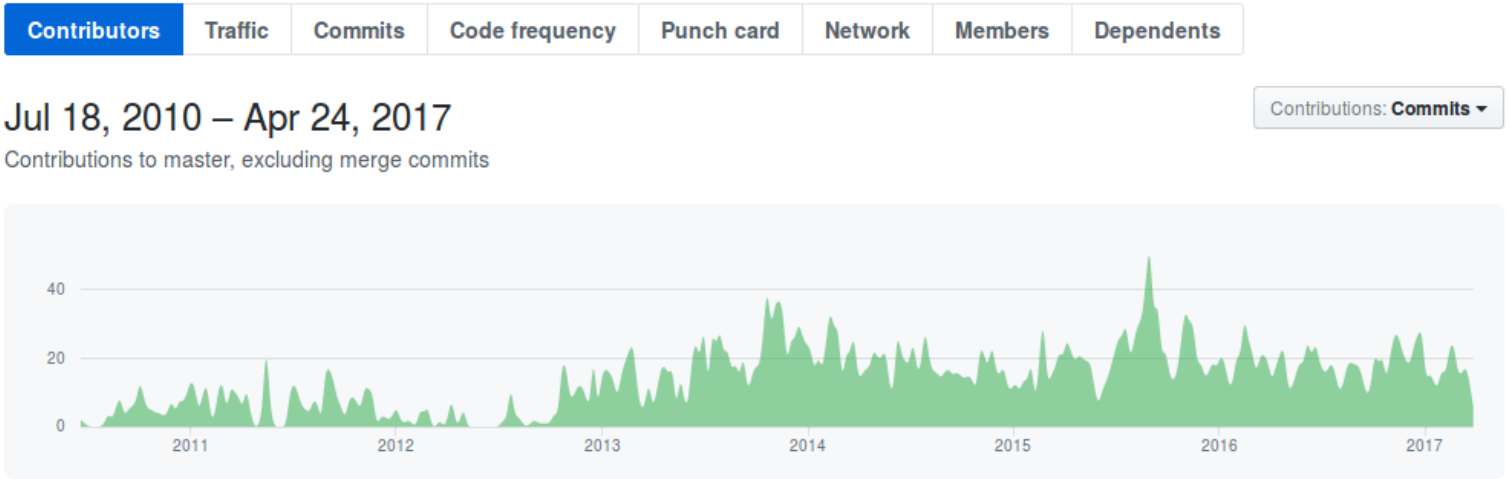
Feature support burden

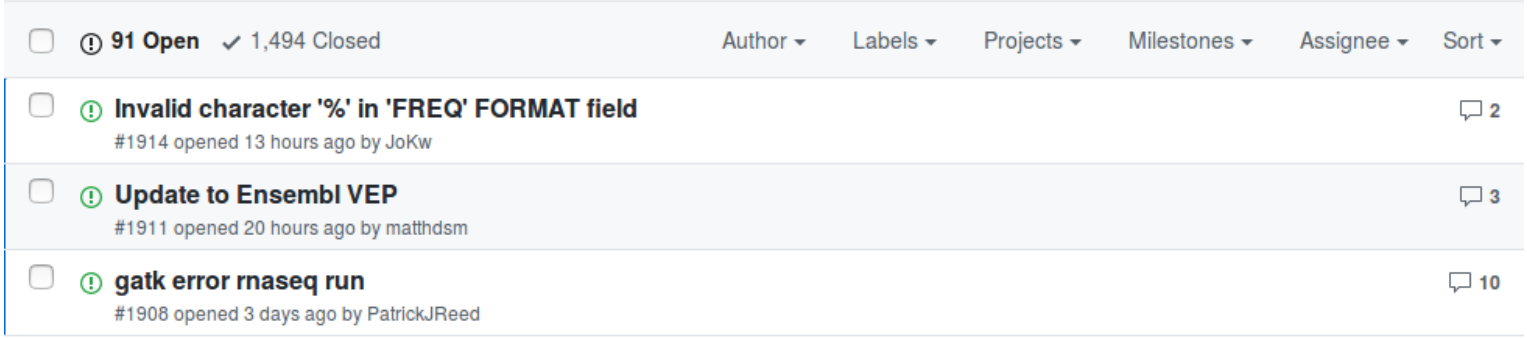
Table 1: Comparison of Nextflow with other workflow management systems

Workflow	Nextflow	Galaxy	Toil	Snakemake	Bpipe
Platform[a]	**Groovy/JVM**	**Python**	**Python**	**Python**	**Groovy/JVM**
Native task support[b]	Yes (any)	No	No	Yes (BASH only)	Yes (BASH only)
Common workflow language[c]	No	Yes	Yes	No	No
Streaming processing[d]	Yes	No	No	No	No
Dynamic branch evaluation	Yes	?	Yes	Yes	Undocumented
Code sharing integration[e]	Yes	No	No	No	No
Workflow modules[f]	No	Yes	Yes	Yes	Yes
Workflow versioning[g]	Yes	Yes	No	No	No
Automatic error failover[h]	Yes	No	Yes	No	No
Graphical user interface[i]	No	Yes	No	No	No
DAG rendering[j]	Yes	Yes	Yes	Yes	Yes
Container management					
Docker support[k]	Yes	Yes	Yes	No	No
Singularity support[l]	Yes	No	No	No	No
Multi-scale containers[m]	Yes	Yes	Yes	No	No
Built-in batch schedulers[n]					
Univa Grid Engine	Yes	Yes	Yes	Partial	Yes
PBS/Torque	Yes	Yes	No	Partial	Yes
LSF	Yes	Yes	No	Partial	Yes
SLURM	Yes	Yes	Yes	Partial	No
HTCondor	Yes	Yes	No	Partial	No
Built-in distributed cluster[o]					
Apache Ignite	Yes	No	No	No	No
Apache Spark	No	No	Yes	No	No
Kubernetes	Yes	No	No	No	No
Apache Mesos	No	No	Yes	No	No
Built-in cloud[p]					
AWS (Amazon Web Services)	Yes	Yes	Yes	No	No

http://www.nature.com/nbt/journal/v35/n4/full/nbt.3820.html

Community: sustainability and support





https://github.com/chapmanb/bcbio-nextgen

Infrastructure Goals

- Local machines
- Clusters: SLURM, SGE, Torque, PBS, LSF
- Clouds: Amazon, Google, Azure
- Clinical environments
- User interface for researchers
- Integrate with LIMS
- Accessible to the general public

Mike Lin Retweeted

DNAnexus, Inc. @dnanexus · 13 Jun 2013
#BigData Parking: "There's no reason **to move data** outside the #cloud. You can do **analysis** right there." ow.ly/m14Ke **#genomics**

Stuart Watt @morungos · 4 Mar 2014
Big upcoming change in **genomics**: **data** sets are now too large **to** download for **analysis**. **Move** code **to** the **data**, not vice versa #ibcretreat2014

Rob Schaefer @CSciBio · Jul 17
huge problem: moving **analysis** to the data, not the other way around. **@ewanbirney** #ISAG2017 #BigData

Aaron Quinlan
@aaronquinlan

Following

This is the only way genomic research can scale.

Javier Quilez @jaquol
Laura Clarke: do not download the data, bring the analysis to the data @laurastephen #gi2017

6:54 PM · 1 Nov 2017

Why do we transfer data around?

- Lots of work to setup and configure an analysis
- Hard to port scalable analysis to new environment

Many great workflow systems: Nexflow

```
#!/usr/bin/env nextflow

cheers=Channel.from "Bonjour","Ciao","Hello","Hola"

process sayHello {
  input:
  val x from cheers

  """
  echo $x world!
  """
}
```

Nextflow

Data-driven computational pipelines

Nextflow enables scalable and reproducible scientific workflows using software containers. It allows the adaptation of pipelines written in the most common scripting languages.

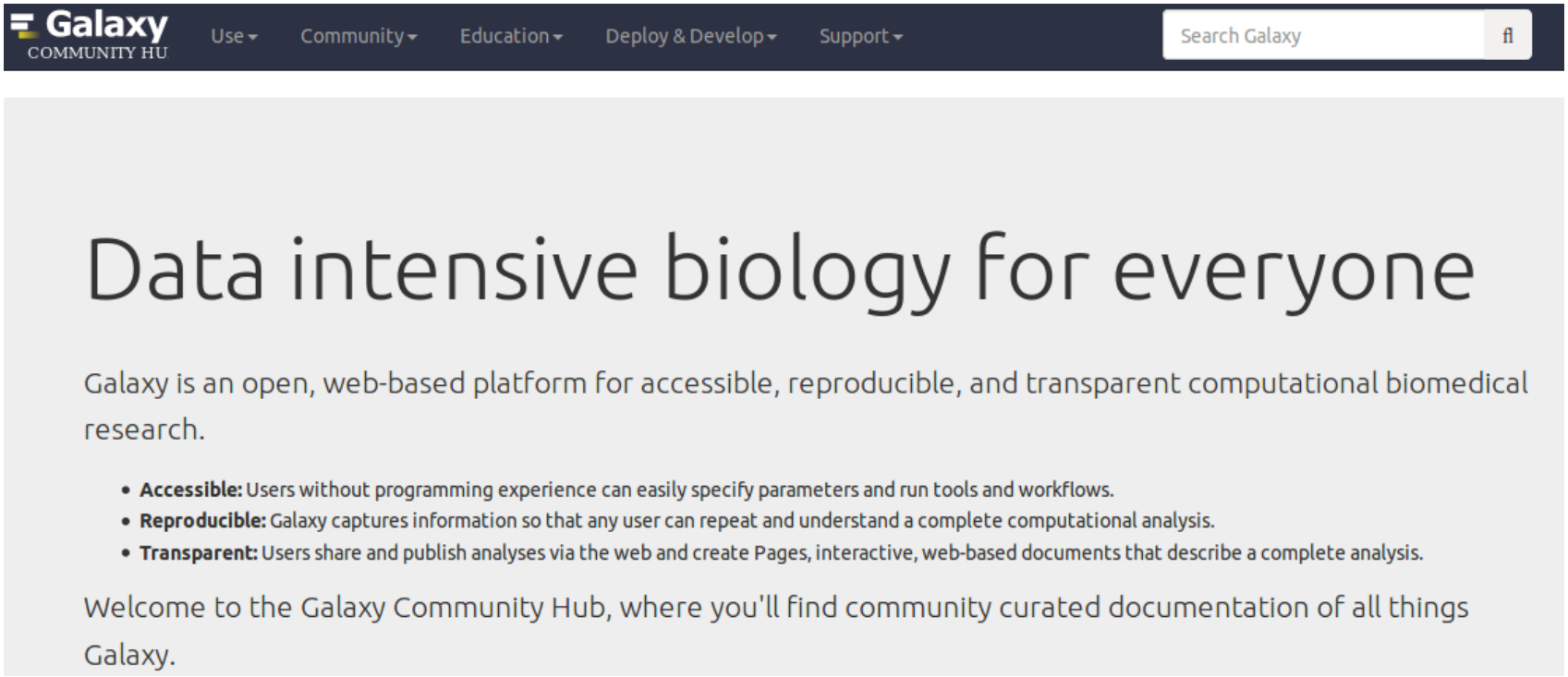
Its fluent DSL simplifies the implementation and the deployment of complex parallel and reactive workflows on clouds and clusters.

Find out more

Zero config | Polyglot | Concurrency | Scale easily

http://nextflow.io/

Many great workflow systems: Galaxy



http://galaxyproject.org/

Many great workflow systems: Snakemake

Snakemake Tutorial

This tutorial introduces the text-based workflow system Snakemake. Snakemake 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, creating a DAG (directed acyclic graph) of jobs that can be automatically parallelized.

Snakemake sets itself apart from existing text-based workflow systems in the following way. Hooking into the Python interpreter, Snakemake offers a definition language that is an extension of Python with syntax to define rules and workflow specific properties. This allows to combine the flexibility of a plain scripting language with a pythonic workflow definition. The Python language is

https://snakemake.readthedocs.io

But, many workflow systems

Existing Workflow systems

Michael R. Crusoe edited this page 8 hours ago · 141 revisions

Computational Data Analysis Workflow Systems

An incomplete list

176. Reflow: a language and runtime for distributed, integrated data processing in the cloud https://github.com/grailbio/reflow
177. Resolwe: an open source dataflow package for Django framework https://github.com/genialis/resolwe
178. Yahoo! Pipes (historical) https://en.wikipedia.org/wiki/Yahoo!_Pipes
179. Walrus https://github.com/fjukstad/walrus
180. Apache Beam https://beam.apache.org/
181. CLOSHA https://closha.kobic.re.kr/ https://www.bioexpress.re.kr/go_tutorial http://docplayer.net/19700397-Closha-manual-ver1-1-kobic-korean-bioinformation-center-kogun82-kribb-re-kr-2016-05-08-bioinformatics-workflow-management-system-in-bio-express.html

https://github.com/common-workflow-language/common-workflow-language/wiki/Existing-Workflow-systems

We'll never agree on one system

- Advantages and disadvantages to each
- Familiarity and teaching
- Personal preference

So we can't easily share workflows

- Single workflow system allows coordinated groups
- Create barrier to sharing externally
- Hard to mix and match components between workflow environments
- How can we do better?

Better abstractions = more interoperability

https://bcbio-nextgen.readthedocs.io/en/latest/contents/cwl.html

Common Workflow Language (CWL)

Workflow	pipeline-se-narrow.cwl		
Sub-workflow 1	01-qc-se.cwl		
Step 1	extract.cwl	extract.py	
Step 2	count.cwl	count.py	
Step 3	fastqc.cwl	fastqc	
Sub-workflow 2	02-trim.cwl		
...			

http://www.commonwl.org/

https://f1000research.com/slides/5-1617

Workflow Description Language (WDL)

http://openwdl.org/

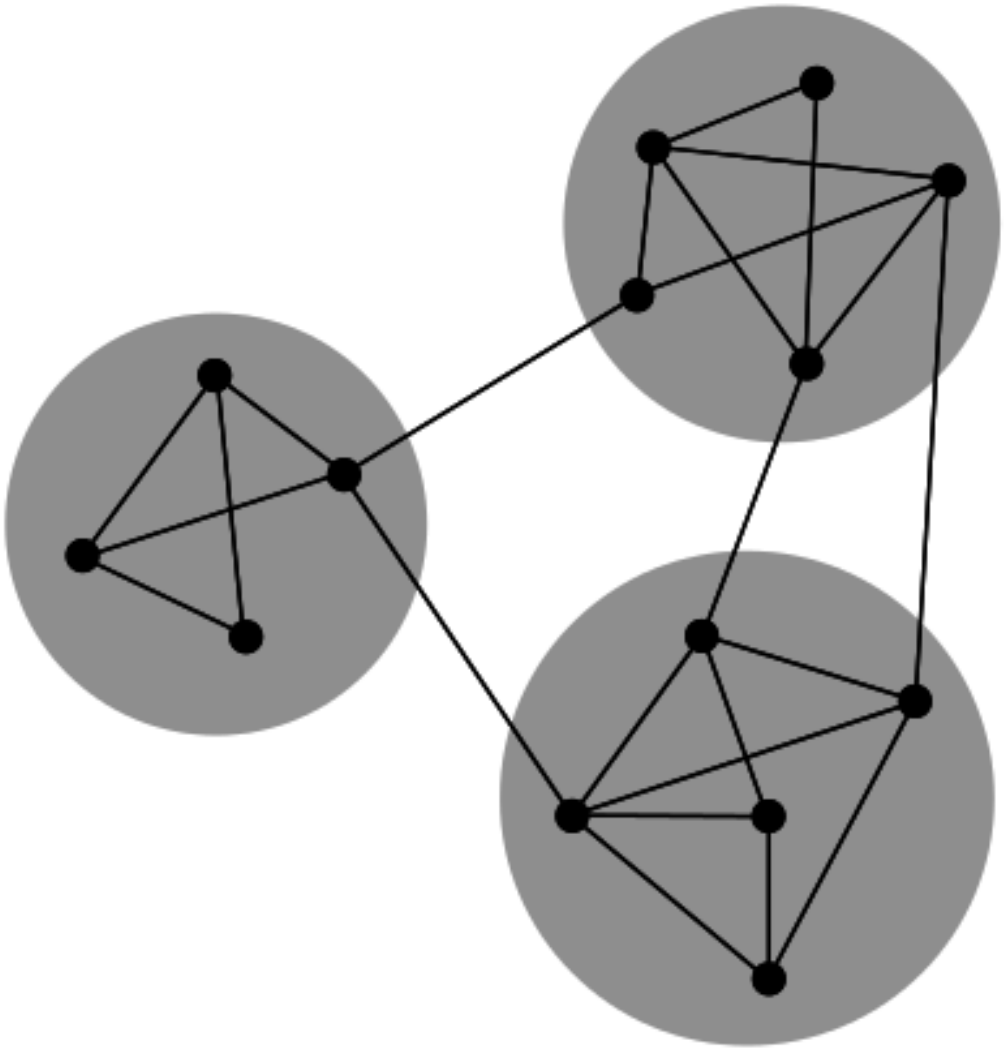
Why use a workflow abstraction?

- Integrate with multiple platforms
  - Rabix Bunny – local
  - Toil – HPC
  - Arvados
  - DNAnexus
  - Seven Bridges + Cancer Genomics Cloud
- Stop maintaining bcbio specific infrastructure
- Focus on hard biological problems

Unique goals with CWL

- Multiple concurrent production environments
  - HPC
  - External vendors (DNAnexus, SevenBridges, Arvados)
  - Direct on Cloud (AWS, GCE, Azure)
- Coordinated release and update process
  - Workflow
  - Tools in containers
  - Reference data

Connections



By jham3 - Own work, CC BY-SA 3.0,

https://commons.wikimedia.org/w/index.php?curid=17125894

CWL in bcbio

- Start with high level configuration file
- Generate CWL
- Run, on any infrastructure that supports CWL
  - Generated CWL
  - Docker or local bcbio installation
  - Genome data

https://bcbio-nextgen.readthedocs.io/en/latest/contents/cwl.html

bcbio-vm: CWL wrapper

- bcbio-like interface integrating with external tools
- Install wrapper plus supported runners

```
conda install -c conda-forge -c bioconda bcbio-nextgen-vm
```

https://github.com/chapmanb/bcbio-nextgen-vm
https://bioconda.github.io/

Template: describe your analysis

```
details:
 - algorithm:
     aligner: bwa
     recalibrate: true
     variantcaller: gatk-haplotype
     tools_on: [gatk4, gvcf]
   analysis: variant2
   variant_regions: Exome-AZ_V2_pluschr20-hg38.bed
genome_build: hg38
```

https://github.com/bcbio/bcbio_validation_workflows

Define your samples

```
samplename,description,batch,validate
NA12878_R1.fq.gz;NA12878_R2.fq.gz,NA12878,gj1,
  hg38/validation/giab-NA12878/truth_small_variants.vcf.gz
NA24385_R,NA24385,gj1,
  hg38/validation/giab-NA24385/truth_small_variants.vcf.gz
NA24631_R,NA24631,gj1,
  hg38/validation/giab-NA24631/truth_small_variants.vcf.gz
```

Local or shared filesystem environment

```
local:
  ref: biodata/collections
  inputs:
    - biodata/regions
    - biodata/giab/na12878
    - biodata/giab/na24385
    - biodata/giab/na24631
resources:
  default:
    cores: 8
    memory: 3500M
    jvm_opts: [-Xms750m, -Xmx3500m]
```

Generate CWL for local or HPC run

```
PNAME=giab-joint
bcbio_vm.py template --systemconfig bcbio_system.yaml \
  joint-template.yaml $PNAME.csv
bcbio_vm.py cwl --systemconfig bcbio_system.yaml \
  $PNAME/config/$PNAME.yaml
```

Run multicore on single machine with Rabix Bunny

```
bcbio_vm.py cwlrun bunny $PNAME-workflow
```

https://github.com/rabix/bunny

Run distributed on SLURM cluster with Toil

```
export TOIL_SLURM_ARGS="-t 0-12:00 -p short"
bcbio_vm.py cwlrun toil --no-container $PNAME-workflow \
  -- --batchSystem slurm
```

http://toil.readthedocs.io

Arvados – Veritas and Curoverse

```
arvados:
  reference: 9127147c168e27e26738524cbd3a59c6+1633
  input: [a1d976bc7bcba2b523713fa67695d715+464]
resources:
  default:
    cores: 8
    memory: 3500M
    jvm_opts: [-Xms750m, -Xmx3500m]
```

https://arvados.org/

Generate CWL and run on Arvados

```
bcbio_vm.py template \
  --systemconfig bcbio_system_arvados.yaml \
  $PNAME-template.yaml $PNAME.csv
bcbio_vm.py cwl \
  --systemconfig bcbio_system_arvados.yaml \
  $PNAME/config/$PNAME.yaml
bcbio_vm.py cwlrun arvados $PNAME-workflow -- \
  --project-uuid qr1hi-j7d0g-7t73h4hrau3l063
```

SevenBridges and the Cancer Genomics Cloud

```
sbgenomics:
  project: bchapman/sgdp-recalling
  reference: bchapman/biodata-hg38
resources:
  default:
    cores: 8
    memory: 3500M
    jvm_opts: [-Xms750m, -Xmx3500m]
```

https://www.sevenbridges.com/

CGC: biological reference data

biodata-hg38

Dashboard | Files

Search file names and description | BED.GZ, FA, FAI, VCF.GZ | Sample ID: All | Task ID: All | Tags: All | + | Clear filters

File name	Size	Ref
truth_small_variants.vcf.gz	214.2 KB	-
ref-transcripts.fa	283.0 MB	-
hg38_transcriptome.fa	306.5 MB	-
hg38.fa.fai	150.6 KB	-
hg38.fa	3.0 GB	-
hapmap_3.3.vcf.gz	59.2 MB	-
gdc-viral.fa.fai	4.9 KB	-
gdc-viral.fa	1.8 MB	-
exac.vcf.gz	3.0 GB	-
esp.vcf.gz	132.7 MB	-
dbsnp-147.vcf.gz	3.3 GB	-

https://cgc.sbgenomics.com/u/bchapman/biodata-hg38/

DNAnexus

```
dnanexus:
  project: giab-joint
  ref:
    project: bcbio_resources
    folder: /reference_genomes
  inputs:
    - /data/input
resources:
  default:
    cores: 8
    memory: 3500M
    jvm_opts: [-Xms750m, -Xmx3500m]
```

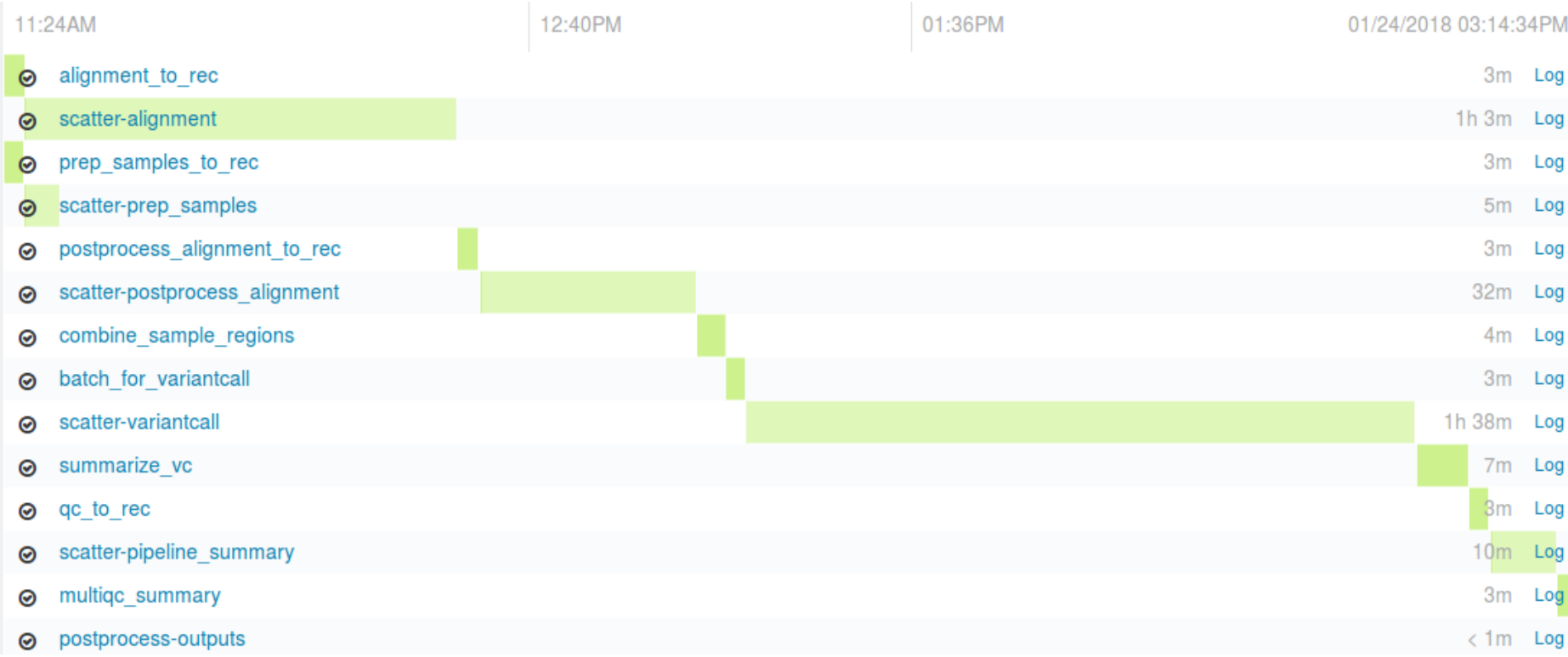
https://platform.dnanexus.com

dx-cwl: compile to DNAnexus workflow language

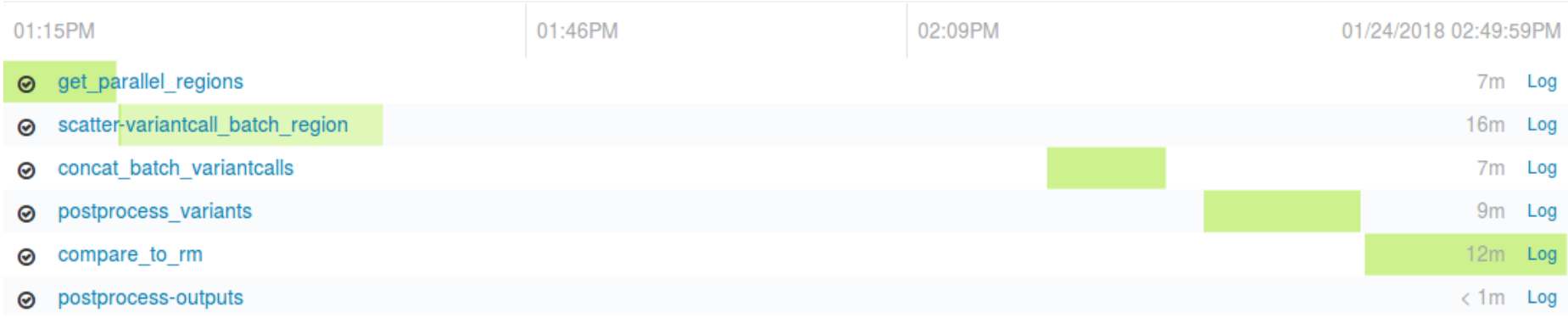
```
dx-cwl compile-workflow PNAME-workflow/main-PNAME.cwl \
  --project PROJECT_ID --token AUTH_TOKEN

dx-cwl run-workflow /dx-cwl-run/main-PNAME/main-PNAME \
       /PNAME-workflow/main-PNAME-samples.json \
       --project PROJECT_ID --token AUTH_TOKEN
```

https://github.com/dnanexus/dx-cwl

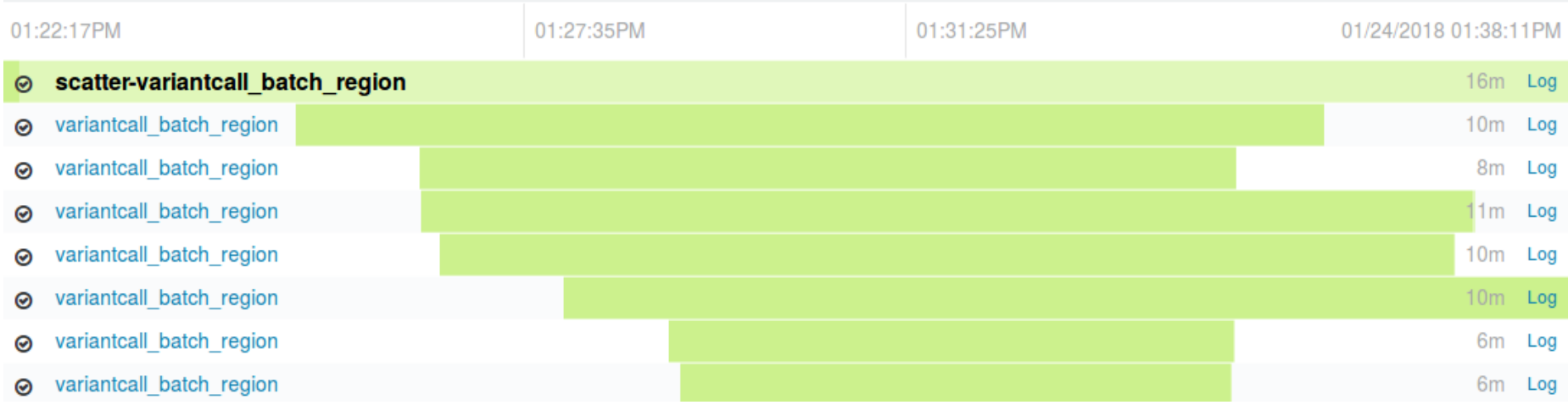
DNAnexus monitoring: align, variant call, QC



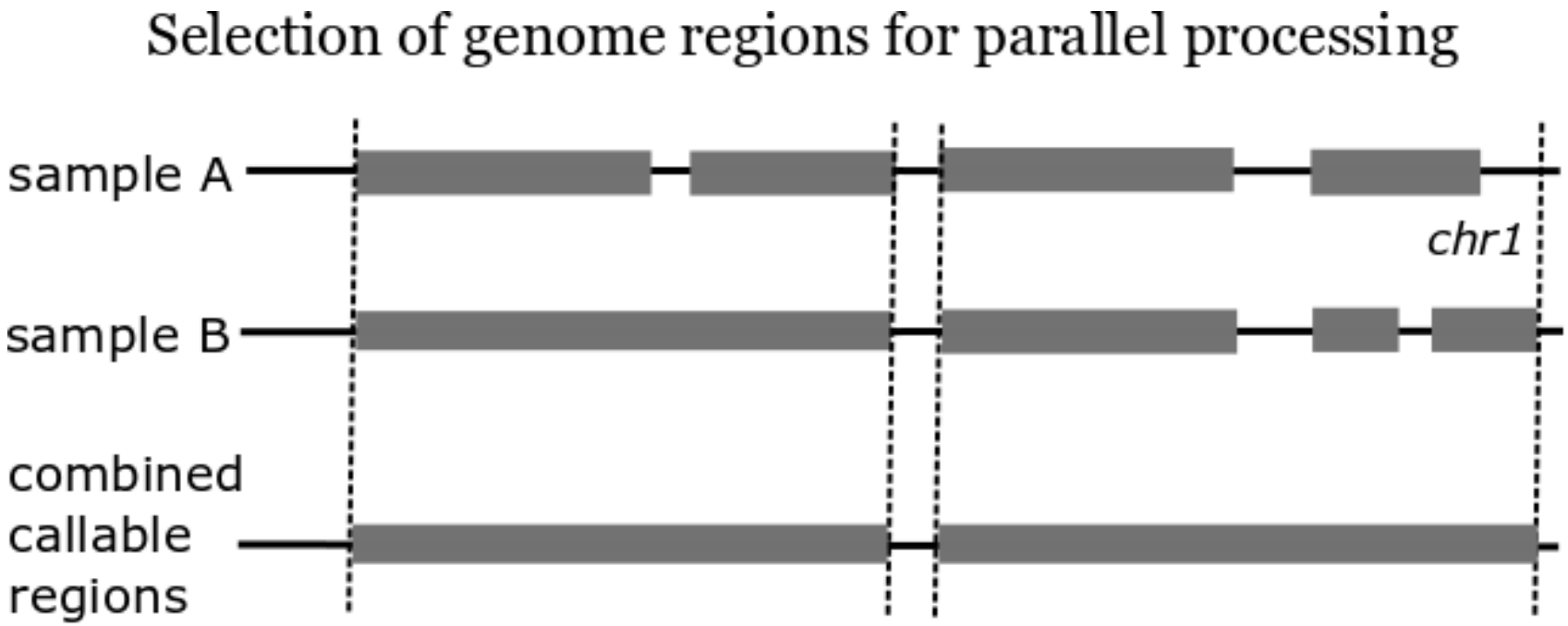
Subworkflow parallelization: per sample or batch



Variant calling parallelization: per region



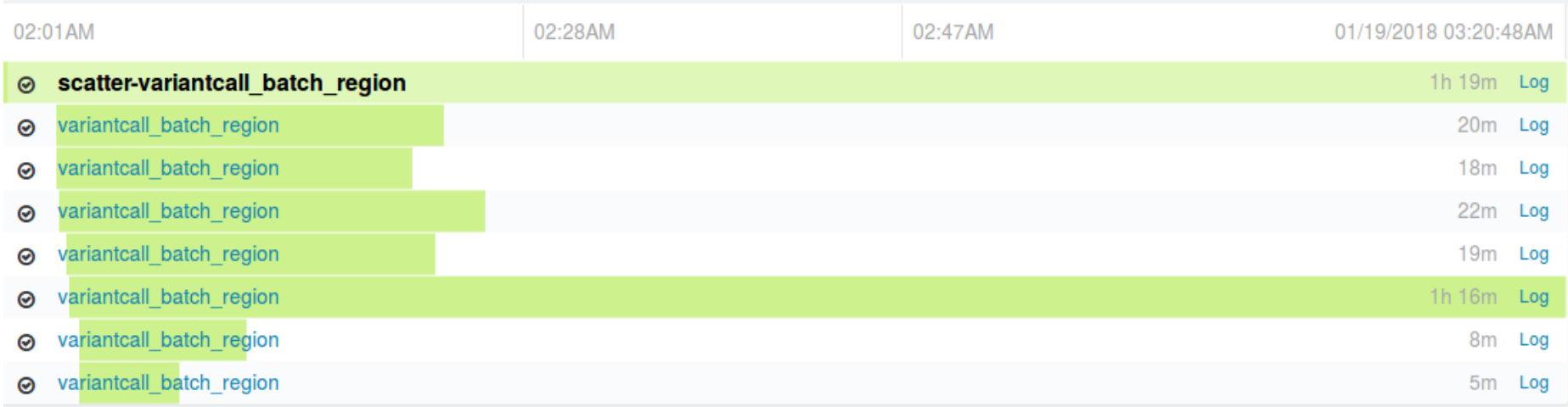
Region splitting approach



Region problem: long tail jobs

03:06PM	04:41PM	05:50PM	01/09/2018 07:50:59PM	
scatter-variantcall_batch_region			4h 44m	Log
variantcall_batch_region			23m	Log
variantcall_batch_region			24m	Log
variantcall_batch_region			23m	Log
variantcall_batch_region			22m	Log
variantcall_batch_region			4h 40m	Log
variantcall_batch_region			10m	Log
variantcall_batch_region			5m	Log

Region improvement: multicore Spark parallelization



Region improvement

Avoid long running jobs on single core

- Use multicore support when available (GATK4 HaplotypeCallerSpark, Strelka2, Sentieon)
- Avoid calling on non chr1-22,X,Y,MT chromosomes
- Maximum coverage downsampling of collapsed and simple sequence repeats
- Trimming of low quality reads at 3' ends

https://blog.dnanexus.com/2018-01-16-evaluating-the-performance-of-ngs-pipelines-on-noisy-wgs-data/

Value of validation

- Integration tests for pipelines
- Unbiased algorithm comparisons
- Baseline for improving methods
- Automated tests for platforms

Reference materials

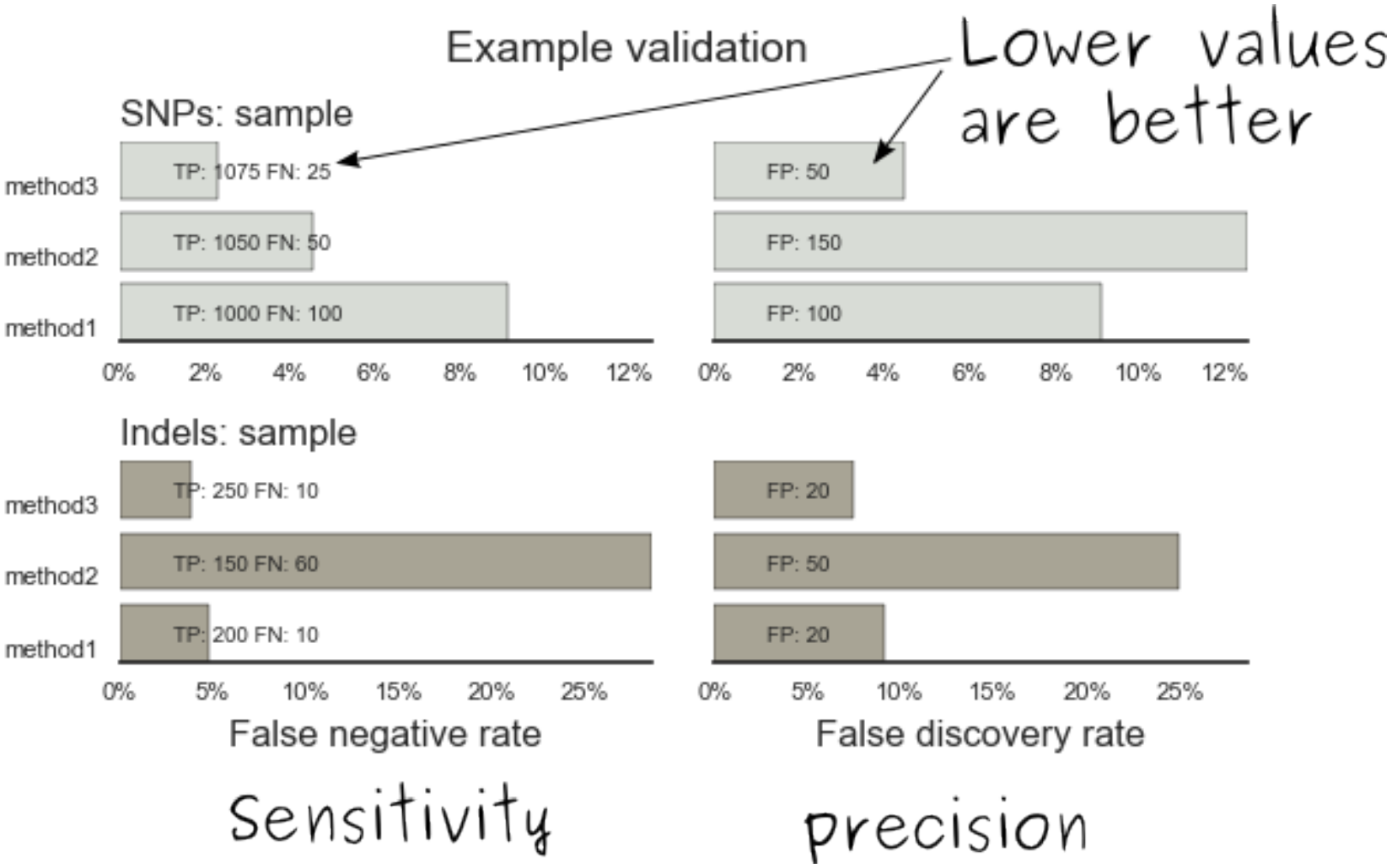
Genome in a Bottle Consortium

Global Alliance for Genomics & Health

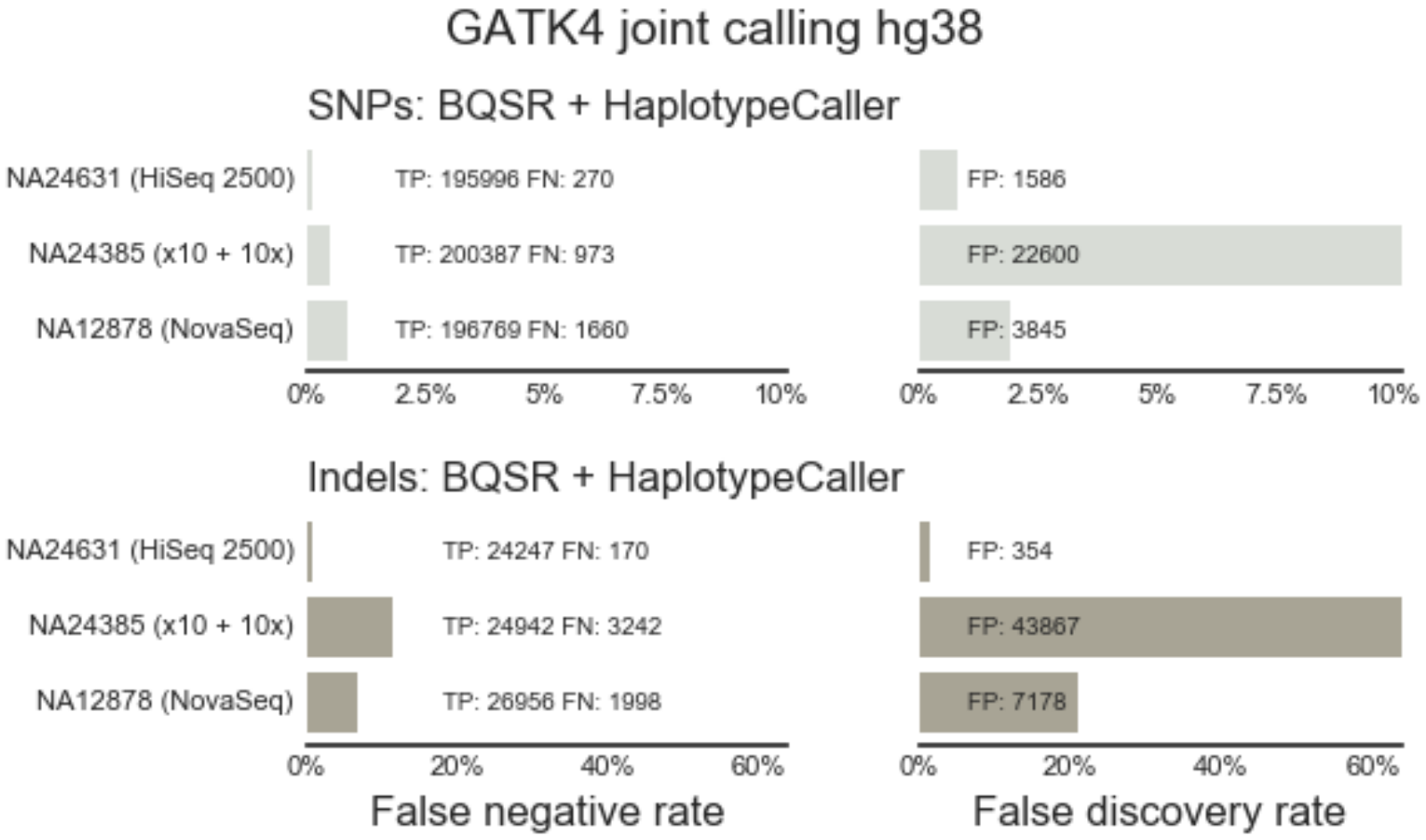
ICGC-TCGA DREAM Mutation Calling challenge

http://www.genomeinabottle.org/
http://ga4gh.org/#/benchmarking-team
https://www.synapse.org/#!Synapse:syn312572

Validation graphs



NA12878, NA24385, NA24631 GATK4 joint calling



https://github.com/bcbio/bcbio_validations/tree/master/gatk4

Need continuous integration process

- Automate testing across multiple platforms
- Test new workflow definitions
- Test new tools and algorithms
- Transparent process

GA4GH: workflow coordination

https://www.synapse.org/#!Synapse:syn8507133/wiki/415976

GA4GH: next steps

- Automation of validation
- Workflow Execution Service (WES)
- Shared API for running CWL/WDL workflows
- Contributors welcome

https://github.com/ga4gh/workflow-execution-schemas

Takeaways

- Science = collaboration and re-use
- Workflow abstractions allow interoperability
- We can build better things together

Summary

- Challenges of building analysis workflows
  - Changing tools
  - Feature support burden
  - Multi-platform interoperability
  - Validation
- bcbio open source community development
- Practical CWL with bcbio: HPC, DNAnexus, Arvados, SevenBridges
- Scaling and parallelization
- GA4GH: Automated multi-platform validation

http://bcb.io