

Interoperable community developed variant calling with bcbio and the Common Workflow Language

Brad Chapman

Bioinformatics Core, Harvard Chan School

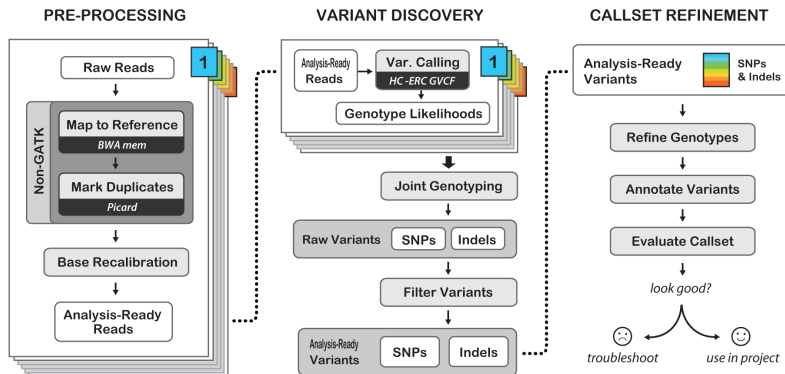
<https://bcb.io>

<http://j.mp/bcbiolinks>

1 May 2017

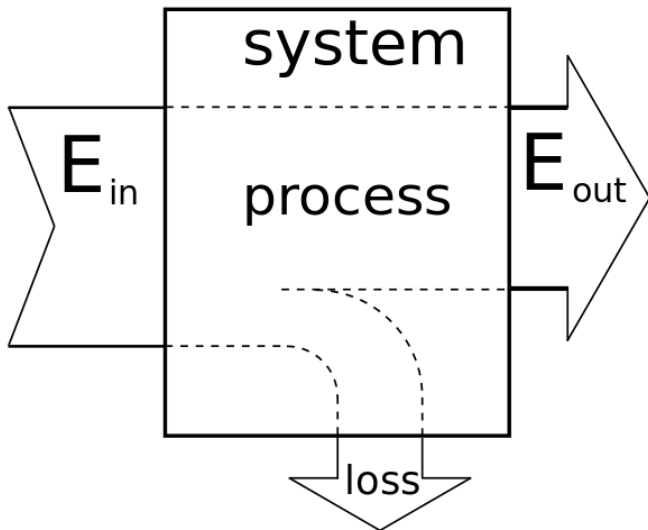
- Barriers to building analysis pipelines
- bcbio: open source community development
- Interoperable infrastructure on the Cancer Genomics Cloud

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/>

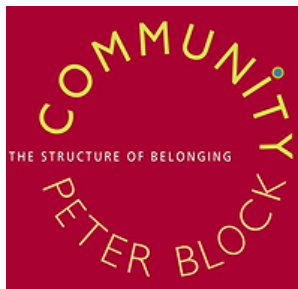


https://commons.wikimedia.org/wiki/File:Efficiency_diagram_by_Zureks.svg

Barriers to implementing yourself

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

Build open source communities

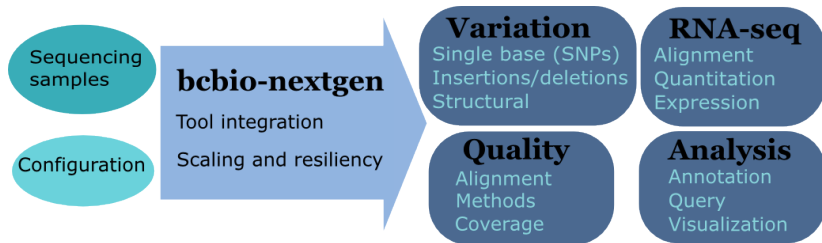


<http://www.amazon.com/Community-Structure-Belonging-Peter-Block/dp/1605092770>



http://www.open-bio.org/wiki/BOSC_2017

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>

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



Genome in a Bottle
Consortium



Global Alliance
for Genomics & Health

ICGC-TCGA DREAM Mutation Calling challenge

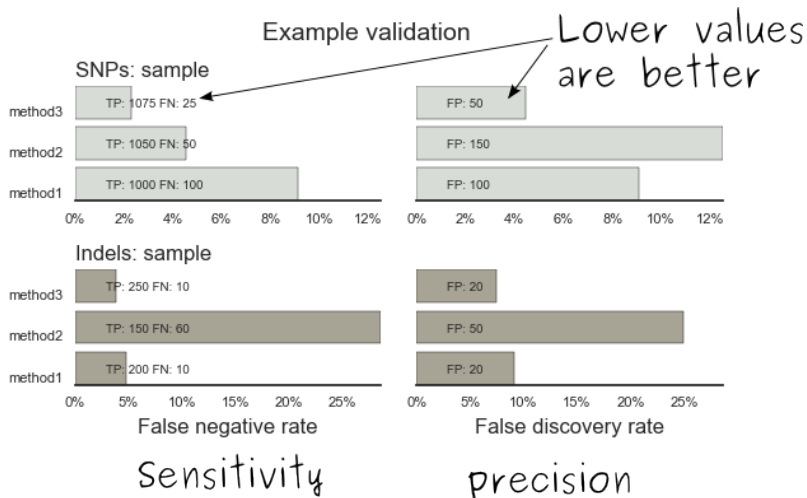
<http://www.genomeinabottle.org/>

<http://ga4gh.org/#/benchmarking-team>

<https://www.synapse.org/#!Synapse:syn312572>

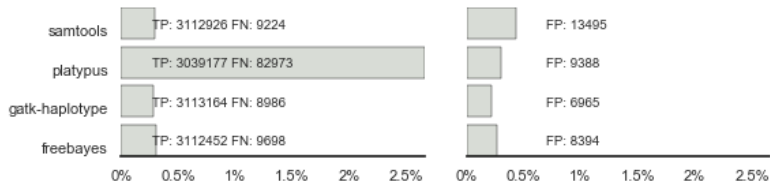
- Collaboration with GATK methods development
- Compare HaplotypeCaller to other methods
- Germline validation
- Genome in a Bottle reference materials
 - NA12878 – Caucasian
 - NA24385 – Ashkenazim Jewish
 - NA24631 – Chinese

Validation graphs

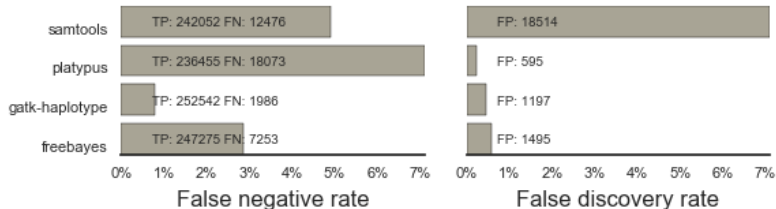


NA12878: Genome in a Bottle whole genome validation

SNPs: bwa

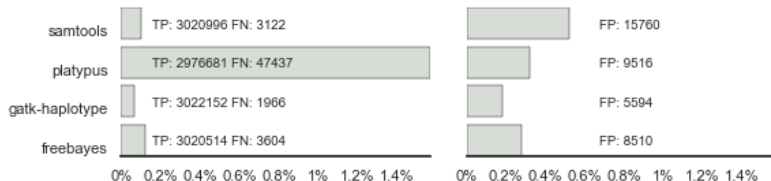


Indels: bwa

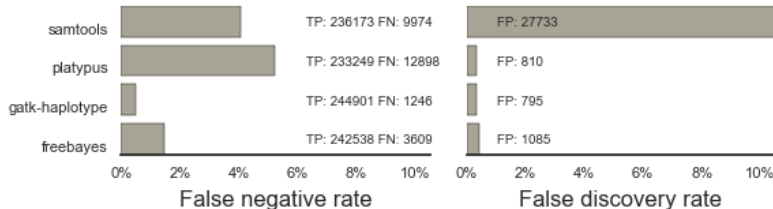


NA24385: Genome in a Bottle whole genome validation

SNPs: bwa



Indels: bwa



Validation results

- Good performance for GATK HaplotypeCaller
- Other good performing callers: FreeBayes
- Consistency across diverse samples
- Identify potential problem areas for tuning
 - samtools Indel false positive rates
 - Platypus SNP sensitivity
- PrecisionFDA: <https://precision.fda.gov/>

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>

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

Warning: the material on this page is considered out of date by the GSA team.

Best Practice Variant Detection with the GATK v2

Warning: the material on this page is considered out of date by the GSA team.

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
July 2012 edited February 4

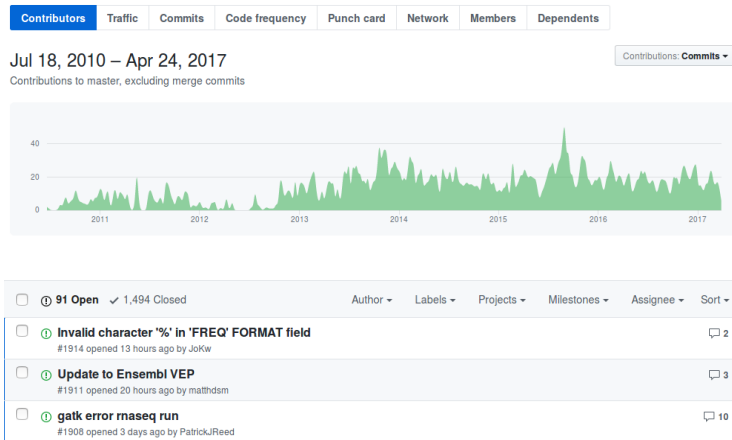
The [Best Practices](#) have been updated for GATK version 3. If you are running an older version, you should seriously consider upgrading. For more details

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 fallover ^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 schedulersⁿ | | | | | |
| 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 |

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

Challenge: many communities

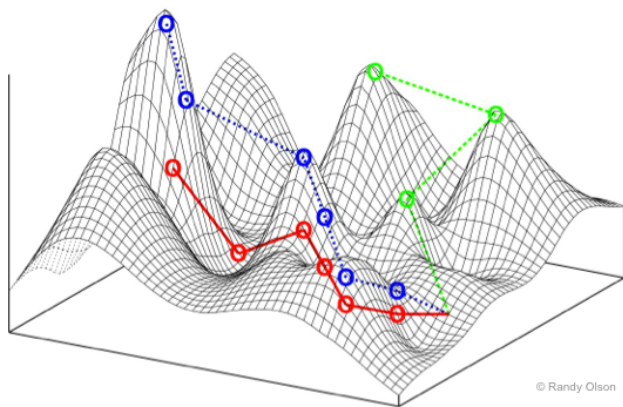


<http://www.cancer-genomics-cloud.org/>

<http://www.cbioportal.org/>

<https://www.synapse.org/>

Challenge: open source communities not yet optimal



https://en.wikipedia.org/wiki/Fitness_landscape

Better abstractions = more interoperability






COMMON
WORKFLOW
LANGUAGE



<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>

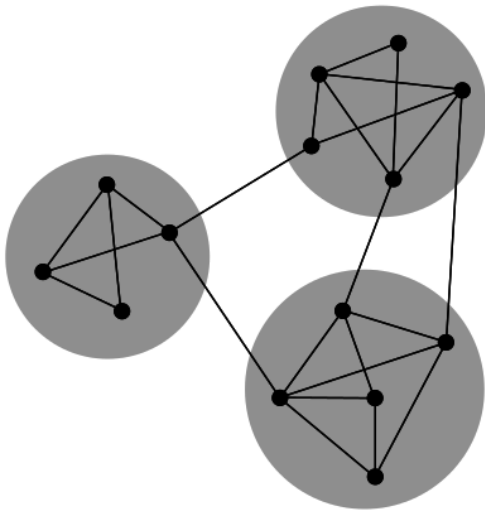
- Start with high level configuration file
- Generate CWL
- Run CWL:
 - Any infrastructure that supports CWL
 - Generated CWL
 - Docker or local bcbio installation
 - Genome data

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

Why use a workflow abstraction?

- Integrate with multiple platforms
 - Cancer Genomics Cloud + Seven Bridges
 - Toil
 - Arvados
 - DNAnexus
 - Galaxy
 - Nextflow
 - Cromwell
- Stop maintaining bcbio specific infrastructure
- Focus on hard biological problems

Connections




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

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

Practical example

- Use bcbio to build CWL that works on the Cancer Genomics Cloud
- Publicly available Simons Genome Diversity Project sample


CGC: project and files

 Projects ▾ Data ▾ Public Apps Public projects ▾

SGDP recalling

Dashboard Files

File extension: All ▾ Sample ID: All ▾ Task ID: All ▾ Tags: All ▾ + Clear filters

| <input type="checkbox"/> ▾ | File name | Size | Reference genome | Sample ID |
|----------------------------|---|----------|------------------|-----------|
| <input type="checkbox"/> |  LP6005443-DNA_H08.srt.aln.bam | 108.4 GB | hs37d5 | DNK05 |

<https://cgc.sbgenomics.com/u/bchapman/sgdp-recalling>

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/>

bcbio: describe your analysis

```
- analysis: variant
  genome_build: hg38
  algorithm:
    aligner: bwa
    mark_duplicates: true
    recalibrate: false
    realign: false
    variantcaller: [gatk-haplotype, freebayes, vardict]
    ensemble:
      numpass: 2
    svcaller: [lumpy, manta, cnvkit]
```


- Build CWL with references to CGC data
- Upload to CGC:
 - CWL as App
 - Sample information with App as Task
- Run same pipeline with CWL
 - Toil: local HPC environment

https://github.com/bcbio/bcbio_validation_workflows

- Challenges of building analysis workflows
 - Validation
 - Changing tools
 - Feature support burden
 - Multi-platform interoperability
- bcbio open source community development
- Common Workflow Language interoperable infrastructure
- Practical example with Cancer Genomics Cloud

<http://bcb.io>