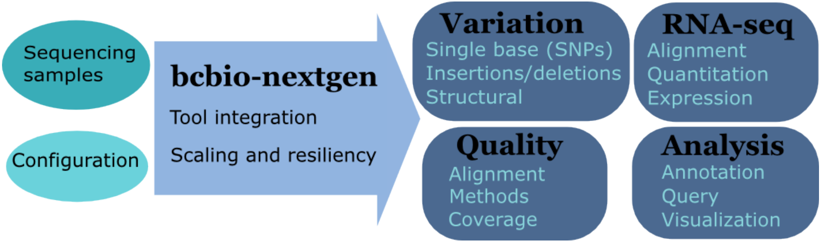
Validated, scalable, community developed variant calling and RNA-seq analysis

Brad Chapman
Bioinformatics Core, Harvard School of Public Health
https://github.com/chapmanb/bcbio-nextgen
http://j.mp/bcbiolinks

3 June 2014

Overview



Sequencing samples

Configuration

bcbio-nextgen

Tool integration

Scaling and resiliency

Variation

Single base (SNPs)
Insertions/deletions
Structural

RNA-seq

Alignment
Quantitation
Expression

Quality

Alignment
Methods
Coverage

Analysis

Annotation
Query
Visualization

Development goals

- Community developed
- Quantifiable
- Configurable
- Scalable
- Reproducible

Content-free descriptions

- Pipeline
- Best-practice
- Framework

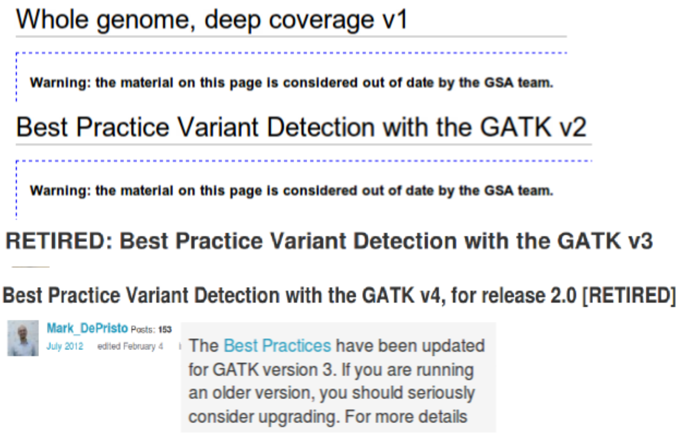
Uses

- Aligners: bwa-mem, novoalign, bowtie2
- Variantion: FreeBayes, GATK, MuTecT, SnpEff, VEP, GEMINI
- RNA-seq: tophat, STAR, cufflinks, HTSeq
- Quality control: fastqc, bamtools, RNA-SeQC
- Manipulation: bedtools, bcftools, biobambam, sambamba, samblaster, samtools, vcflib

Provides

- Validation – outputs + automated evaluation
- Tool integration
- Multi-platform support
- Scaling

Complex, rapidly changing tools



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Warning: the material on this page is considered out of date by the GSA team.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Warning: the material on this page is considered out of date by the GSA team.

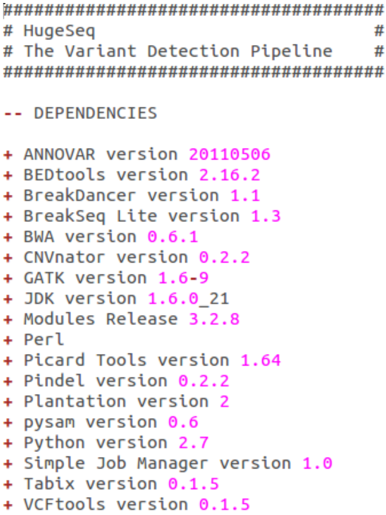
RETIRED: Best Practice Variant Detection with the GATK v3

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

Large number of specialized dependencies



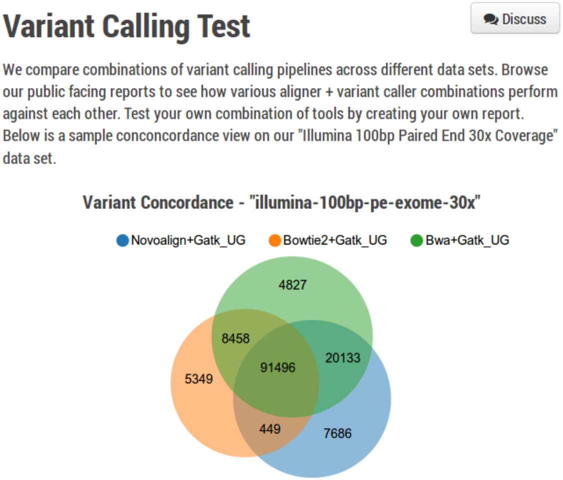
```
#####################################
# HugeSeq                           #
# The Variant Detection Pipeline    #
#####################################

-- DEPENDENCIES

+ ANNOVAR version 20110506
+ BEDtools version 2.16.2
+ BreakDancer version 1.1
+ BreakSeq Lite version 1.3
+ BWA version 0.6.1
+ CNVnator version 0.2.2
+ GATK version 1.6-9
+ JDK version 1.6.0_21
+ Modules Release 3.2.8
+ Perl
+ Picard Tools version 1.64
+ Pindel version 0.2.2
+ Plantation version 2
+ pysam version 0.6
+ Python version 2.7
+ Simple Job Manager version 1.0
+ Tabix version 0.1.5
+ VCFtools version 0.1.5
```

https://github.com/StanfordBioinformatics/HugeSeq

Quality differences between methods



Variant Calling Test

Discuss

We compare combinations of variant calling pipelines across different data sets. Browse our public facing reports to see how various aligner + variant caller combinations perform against each other. Test your own combination of tools by creating your own report. Below is a sample conconcordance view on our "Illumina 100bp Paired End 30x Coverage" data set.

Variant Concordance - "illumina-100bp-pe-exome-30x"

http://www.bioplanet.com/gcat

Scaling on full ecosystem of clusters

DRMAA
Distributed Resource Management Application API — www.drmaa.org

openlava
Open Source Workload Management

GRID ENGINE

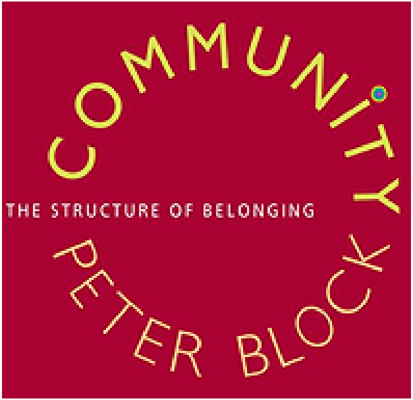
slurm
workload manager

amazon web services™

Platform LSF

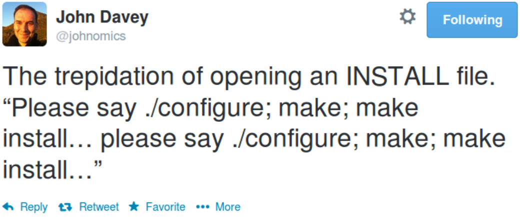
Torque

Solution



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

Community: installation



John Davey
@johnomics

The trepidation of opening an INSTALL file. “Please say ./configure; make; make install… please say ./configure; make; make install…”

Automated Install

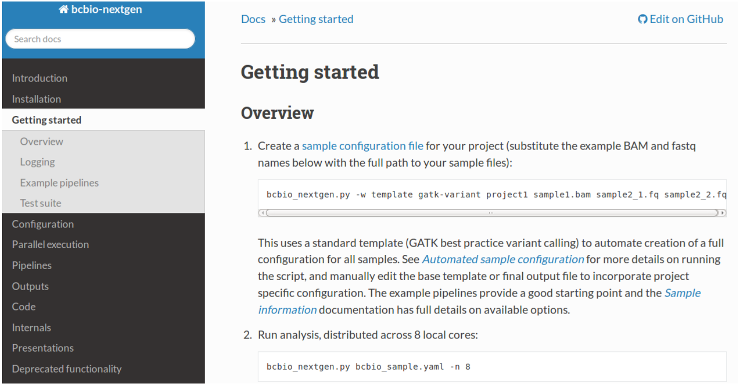
Bare machine to ready-to-run with tools and data

- CloudBioLinux: http://cloudbiolinux.org
- Homebrew: https://github.com/Homebrew/homebrew-science
- Conda: http://j.mp/py-conda

Easier install

Docker

Community: documentation



Getting started

Overview

1. Create a sample configuration file for your project (substitute the example BAM and fastq names below with the full path to your sample files):

```
bcbio_nextgen.py -w template gatk-variant project1 sample1.bam sample2_1.fq sample2_2.fq
```

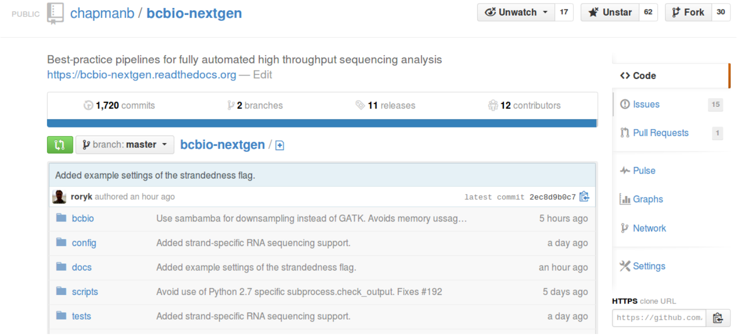
   This uses a standard template (GATK best practice variant calling) to automate creation of a full configuration for all samples. See *Automated sample configuration* for more details on running the script, and manually edit the base template or final output file to incorporate project specific configuration. The example pipelines provide a good starting point and the *Sample information* documentation has full details on available options.

2. Run analysis, distributed across 8 local cores:

```
bcbio_nextgen.py bcbio_sample.yaml -n 8
```

https://bcbio-nextgen.readthedocs.org

Community: contribution



PUBLIC chapmanb / bcbio-nextgen

Best-practice pipelines for fully automated high throughput sequencing analysis
https://bcbio-nextgen.readthedocs.org — Edit

Added example settings of the strandedness flag.

roryk authored an hour ago — latest commit 2ec8d9b0c7

bcbio	Use sambamba for downsampling instead of GATK. Avoids memory ussag…	5 hours ago
config	Added strand-specific RNA sequencing support.	a day ago
docs	Added example settings of the strandedness flag.	an hour ago
scripts	Avoid use of Python 2.7 specific subprocess.check_output. Fixes #192	5 days ago
tests	Added strand-specific RNA sequencing support.	a day ago

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

Validation

Tests for implementation and methods

- Currently:
  - Family/population calling
  - RNA-seq differential expression
  - Structural variations
- Expand to:
  - Cancer tumor/normal

    `http://j.mp/cancer-var-chal`

Example evaluation

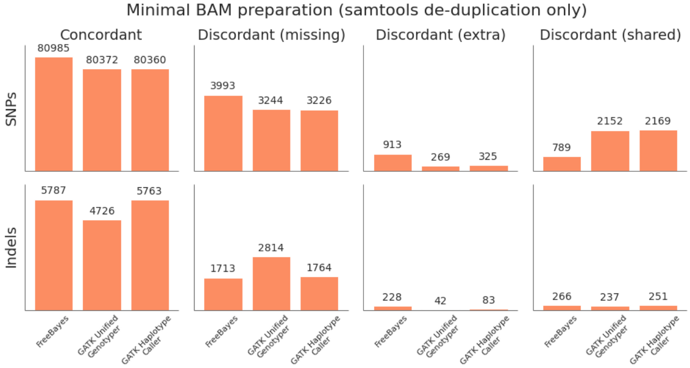
- Variant calling
  - GATK UnifiedGenotyper
  - GATK HaplotypeCaller
  - FreeBayes
- Two preparation methods
  - Full (de-duplication, recalibration, realignment)
  - Minimal (only de-duplication)

Reference materials

Genome in a Bottle
Consortium

http://www.genomeinabottle.org/

Quantify quality



- Quantification details: http://j.mp/bcbioeval2

Validation enables scaling

- Little value in realignment when using haplotype aware caller
- Little value in recalibration when using high quality reads
- Streaming de-duplication approaches provide same quality without disk IO

Configuration overview

- High level abstraction
- Adjust by intent, rather than command line
- Domain specific language
- YAML configuration file

Getting started

- Start with examples

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

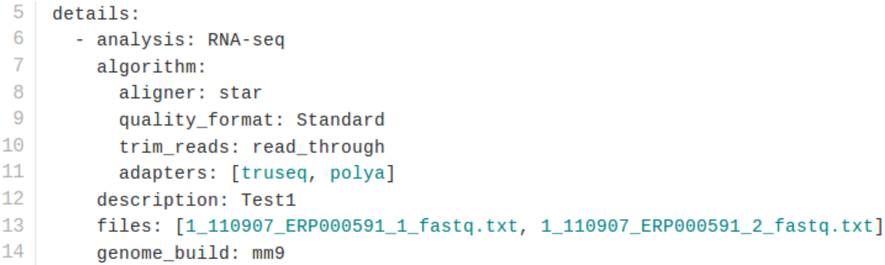
- Automatically generate configuration

  https://bcbio-nextgen.readthedocs.org/en/latest/contents/configuration.html#automated-sample-configuration

- Parameter documentation

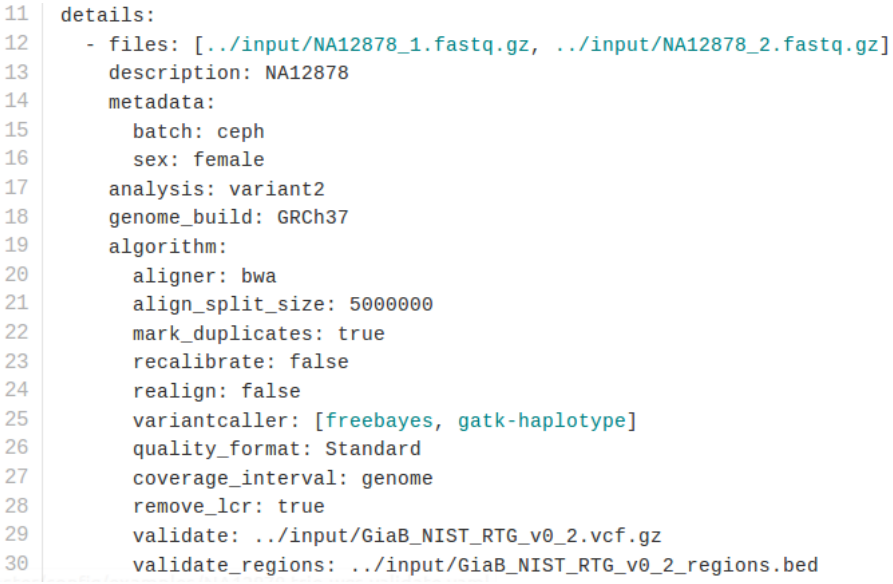
  https://bcbio-nextgen.readthedocs.org/en/latest/contents/configuration.html#algorithm-parameters

Example – RNA-seq



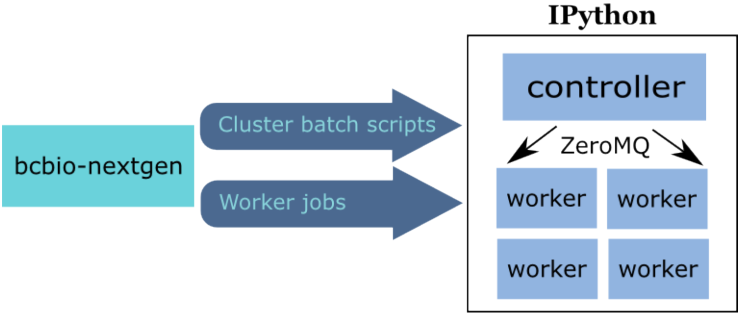
```
details:
  - analysis: RNA-seq
    algorithm:
      aligner: star
      quality_format: Standard
      trim_reads: read_through
      adapters: [truseq, polya]
    description: Test1
    files: [1_110907_ERP000591_1_fastq.txt, 1_110907_ERP000591_2_fastq.txt]
    genome_build: mm9
```

Example – variant calling



```
details:
  - files: [../input/NA12878_1.fastq.gz, ../input/NA12878_2.fastq.gz]
    description: NA12878
    metadata:
      batch: ceph
      sex: female
    analysis: variant2
    genome_build: GRCh37
    algorithm:
      aligner: bwa
      align_split_size: 5000000
      mark_duplicates: true
      recalibrate: false
      realign: false
      variantcaller: [freebayes, gatk-haplotype]
      quality_format: Standard
      coverage_interval: genome
      remove_lcr: true
      validate: ../input/GiaB_NIST_RTG_v0_2.vcf.gz
      validate_regions: ../input/GiaB_NIST_RTG_v0_2_regions.bed
```

Scaling overview



- Infrastructure details: http://j.mp/bcbioscale
- IPython: http://ipython.org/ipython-doc/dev/parallel/index.html

Current target environment

- Cluster scheduler
  - SLURM
  - Torque
  - SGE
  - LSF
- Shared filesystem
  - NFS
  - Lustre
- Local temporary disk
  - SSD

Configuration to batch scripts

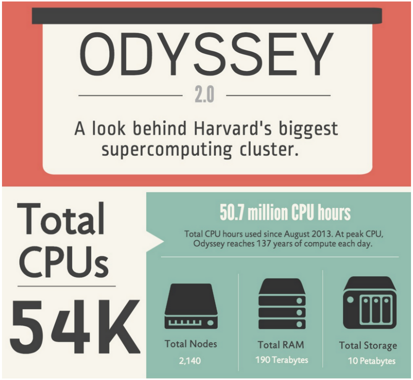
Configuration

```
bwa:
  cmd: bwa
  cores: 16
samtools:
  cores: 16
  memory: 2G
gatk:
  jvm_opts: ["-Xms750m", "-Xmx2750m"]
```

Batch file

```
#PBS -l nodes=1:ppn=16
#PBS -l mem=45260mb
```

Intel + Harvard FAS Research Computing



ODYSSEY

2.0

A look behind Harvard's biggest supercomputing cluster.

Total CPUs

54K

50.7 million CPU hours

Total CPU hours used since August 2013. At peak CPU, Odyssey reaches 137 years of compute each day.

Total Nodes	Total RAM	Total Storage
2,140	190 Terabytes	10 Petabytes

James Cuff, John Morrissey, Kristina Kermanshahche

https://rc.fas.harvard.edu/

Evaluation details

System

- 560 cores
- 4Gb RAM/core
- Lustre filesystem
- Infiniband network

Samples

- 75 samples
- 30x whole genome (100Gb)
- Illumina
- Family-based calling

Timing: Alignment

Step	Time	Processes
Alignment preparation	9.5 hours	BAM to fastq; bgzip; grabix index
Alignment	31 hours	bwa-mem alignment samblaster deduplication
BAM merge	5.5 hours	Merge alignment parts
Post-processing	11 hours	Calculate callable regions

Timing: Variant calling

Step	Time	Processes
Variant calling	30 hours	FreeBayes
Variant post-processing	5 hours	Combine variant files; annotate: GATK and snpEff

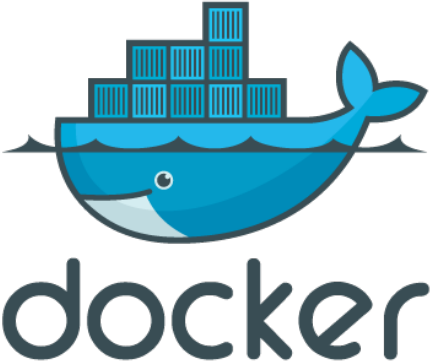
Timing: Analysis and QC

Step	Time	Processes
GEMINI	5 hours	Create GEMINI SQLite database
Quality Control	2.5 hours	FastQC, alignment and variant statistics

Timing: Overall

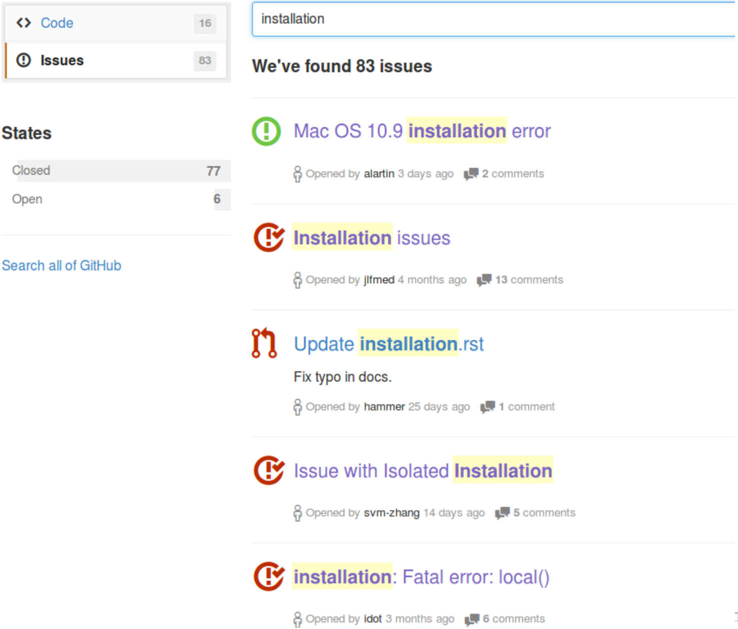
- 100 hours, ~4 days for 75 samples
- ~1 1/2 hours per sample at 560 cores
- In progress: optimize for single samples

Reproducible environment



http://docker.io

Consistent support environment



installation

We've found 83 issues

States

Closed	77
Open	6

Search all of GitHub

Mac OS 10.9 **installation** error

Opened by alartin 3 days ago 2 comments

Installation issues

Opened by jlfmed 4 months ago 13 comments

Update **installation**.rst

Fix typo in docs.

Opened by hammer 25 days ago 1 comment

Issue with Isolated **Installation**

Opened by svm-zhang 14 days ago 5 comments

installation: Fatal error: local()

Opened by idot 3 months ago 6 comments

Docker benefits

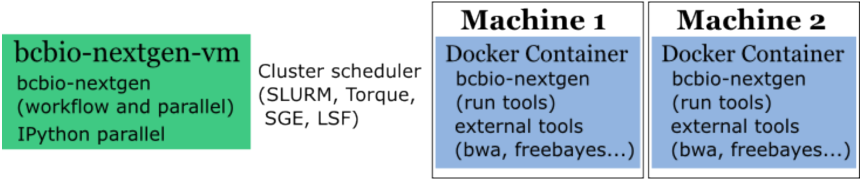
- Fully isolated
- Reproducible – store full environment with analysis (~1Gb)
- Improved installation – single download + data

bcbio with Docker

- External Python wrapper
  - Installation
  - Start and run containers
  - Mount external data into containers
  - Parallelize
- All analysis tools inside Docker

https://github.com/chapmanb/bcbio-nextgen-vm
http://j.mp/bcbiodocker

Docker HPC parallelization



bcbio-nextgen-vm

bcbio-nextgen
(workflow and parallel)
IPython parallel

Cluster scheduler
(SLURM, Torque,
SGE, LSF)

Machine 1

Docker Container
bcbio-nextgen
(run tools)
external tools
(bwa, freebayes...)

Machine 2

Docker Container
bcbio-nextgen
(run tools)
external tools
(bwa, freebayes...)

Consistent scaling environment

Amazon challenges

- Cost – spot instances
- Disk – local scratch, no EBS
- Organization – no shared filesystems, S3 push/pull
- Data – reconstitute on minimal machines
- Security – encryption at rest

Clusterk `http://clusterk.com/`

Summary

- Community development > challenges
- Easy to install, learn and contribute
- Validated
- Configurable
- Scalability
- Reproducibility and virtualization

https://github.com/chapmanb/bcbio-nextgen
http://j.mp/bcbiolinks