

Community based approaches to scaling variant calling pipelines

Brad Chapman

Bioinformatics Core, Harvard School of Public Health

<https://github.com/chapmanb>

<http://j.mp/bcbiolinks>

10 December 2013

Complex, rapidly changing pipelines

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Warning: the material on this page is considered out of date by the GSA team.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Warning: the material on this page is considered out of date by the GSA team.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



Mark_DePristo Posts: 150 Administrator, GSA Official Member admin
July 2012 edited April 24 in [Methods and Workflows](#)

HaplotypeCaller now so sensitive, it cries at the movies

We know you don't want to miss a single true variant, so for this release, we've put a lot of effort into making the HaplotypeCaller more sensitive. And it's paying off: in our tests, the HaplotypeCaller is now more sensitive than the UnifiedGenotyper for calling both SNPs and indels when run over whole genome datasets.

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>

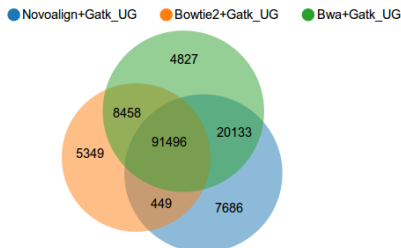
Quality differences between methods

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 concordance view on our "Illumina 100bp Paired End 30x Coverage" data set.

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



<http://www.bioplanet.com/gcat>

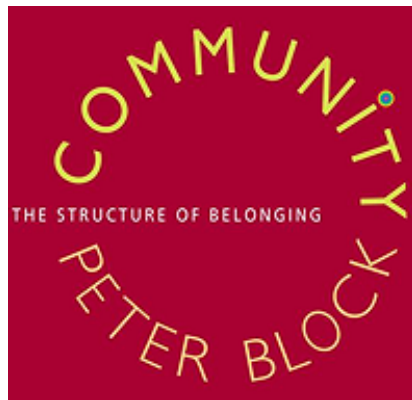
Scaling on full ecosystem of clusters



Platform LSF

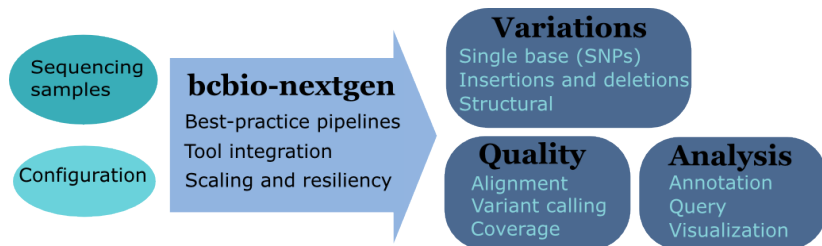
Torque

Solution



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

Overview



Development goals

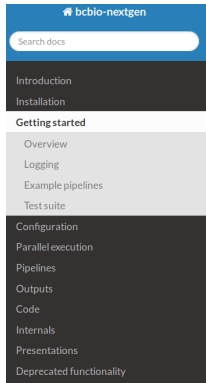
- Community developed
- Quantifiable
- Scalable

Automated Install

Bare machine to ready-to-run pipeline, tools and data

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

Community: documentation



Docs » Getting started

[Edit on GitHub](#)

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**



Unwatch 17 Unstar 62 Fork 30






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

1,720 commits 2 branches 11 releases 12 contributors

 branch: master bcbio-nextgen / 

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 usag...	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

Code

Issues 15


Pull Requests 1

Pulse

Graphs

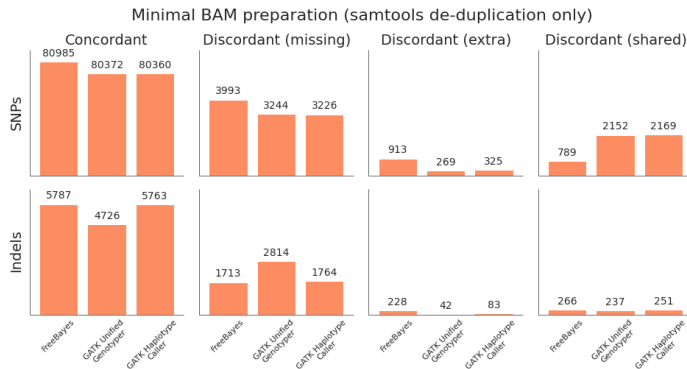
Network

Settings

HTTPS clone URL
<https://github.com> 

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

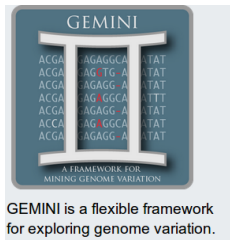
Quantify quality



- Reference materials: <http://www.genomeinabottle.org/>
- Quantification details: <http://j.mp/bcbioeval2>

- Unit tests for implementation and methods
- Expand to:
 - Cancer tumor/normal
<http://j.mp/cancer-var-chal>
 - Family/population calling
 - Structural variations

Analysis: GEMINI



GEMINI: a flexible framework for exploring genome variation

Overview

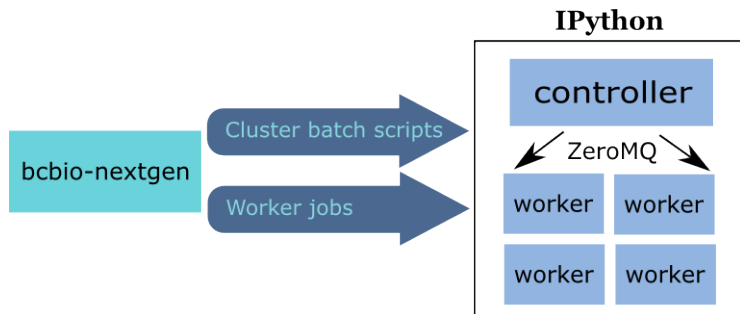
GEMINI (GEnome MINing) is designed to be a flexible framework for exploring genetic variation in the context of the wealth of genome annotations available for the human genome. By placing genetic variants, sample genotypes, and useful genome annotations into an integrated database framework, **GEMINI** provides a simple, flexible,

Rory Kirchner

Aaron Quinlan

<http://quinlanlab.org/tutorials/cshl2013/gemini.html>

Scaling overview



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

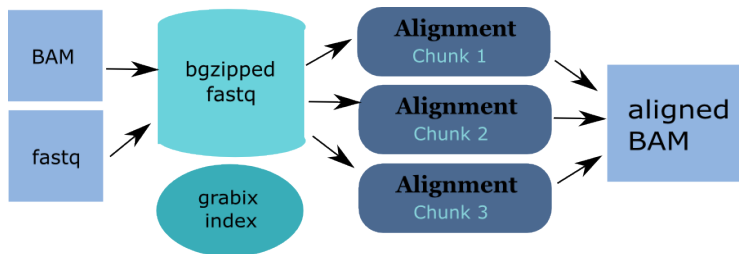
Current target environment

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

Scaling wins

- Split alignments
- Split by genome regions
- Take advantage of multicore algorithms
- Manage memory
- Avoid IO

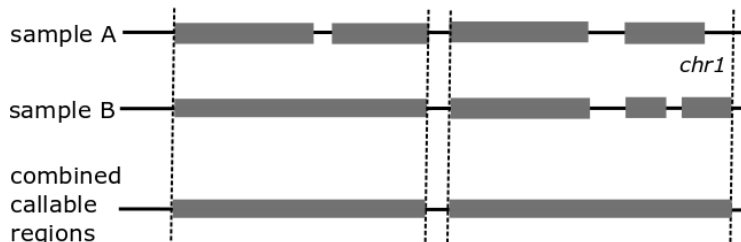
Alignment parallelization



<https://github.com/arq5x/grabix>

Variant calling and BAM preparation parallelization

Selection of genome regions for parallel processing



Multicore parallelization

BAM manipulation

Sambamba

<https://github.com/lomereiter/sambamba>

Prep analysis database (SQLite)

GEMINI

<https://github.com/arq5x/gemini>

Memory usage

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
```

Pipes and streaming algorithms

```
("{bwa} mem -M -t {num_cores} -R '{rg_info}' -v 1 "  
  "{ref_file} {fastq_file} {pair_file} "  
  "| {samtools} view -b -S -u - "  
  "| {samtools} sort -@ {num_cores} -m {max_mem} "  
  "- {tx_out_prefix}")
```

Dell Active Infrastructure for HPC Life Sciences

High Performance Computing

- > Dell Advantage
- > Strategy
- > Products & Services
- > Resource Library

"With diseases like neuroblastoma, hours matter. Our new Dell HPC cluster allows us to do the processing we need to get a meaningful result in a clinically relevant amount of time."

— Jason Corneveaux, Bioinformatician, Neurogenomics Division, the Translational Genomics Research Institute ¹

High performance for high-volume genomics research

Processing complex genomic data sets requires massive compute power, storage and network capabilities. Getting the balance right is critical to success, but without proper support and expertise, it can take months to integrate the necessary computing components and tune them for maximum performance and efficiency.

Glen Otero, Will Cottay

<http://dell.com/ai-hpc-lifesciences>

Evaluation details

System

- 400 cores
- 3Gb RAM/core
- Lustre filesystem
- Infiniband network

Samples

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

Timing: Alignment

Step	Time	Processes
Alignment preparation	13 hours	BAM to fastq; bgzip; grabix index
Alignment	30 hours	bwa-mem alignment
BAM merge	7 hours	Merge alignment parts
Alignment post-processing	6 hours	Calculate callable regions

Timing: Variant calling

Step	Time	Processes
Post-alignment BAM preparation	6 hours	De-duplication
Variant calling	18 hours	FreeBayes
Variant post-processing	2 hours	Combine variant files; annotate: GATK and snpEff

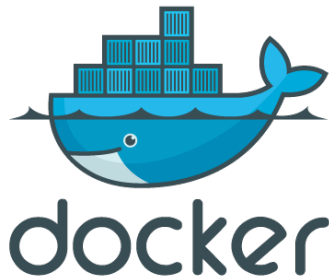
Timing: Analysis and QC

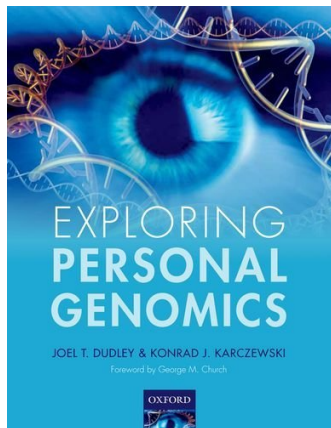
Step	Time	Processes
BAM merging	6 hours	Combine post-processed BAM file sections
GEMINI	3 hours	Create GEMINI SQLite database
Quality Control	5 hours	FastQC, alignment and variant statistics

Timing: Overall

- 4 days for 60 samples
- ~2 hours per sample at 400 cores
- In progress: optimize for single samples

Virtualization and reproducibility





<http://exploringpersonalgenomics.org/>

Summary

- Community developed pipelines > challenges
- Focus
 - Community: easy to install and contribute
 - Assessing quality: good science
 - Scalability
 - Parallelization
 - Diagnose bottlenecks
- Widely accessible

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

<http://j.mp/bcbiolinks>