Community development of validated variant calling pipelines

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

21 January 2014

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Warning: the material on this page is considered out of date by the GSA team.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Warning: the material on this page is considered out of date by the GSA team.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

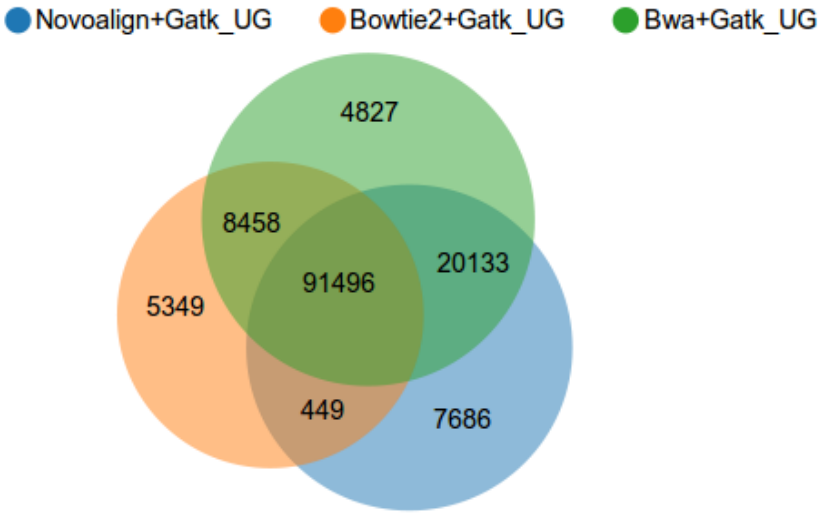
Quality differences between methods

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

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



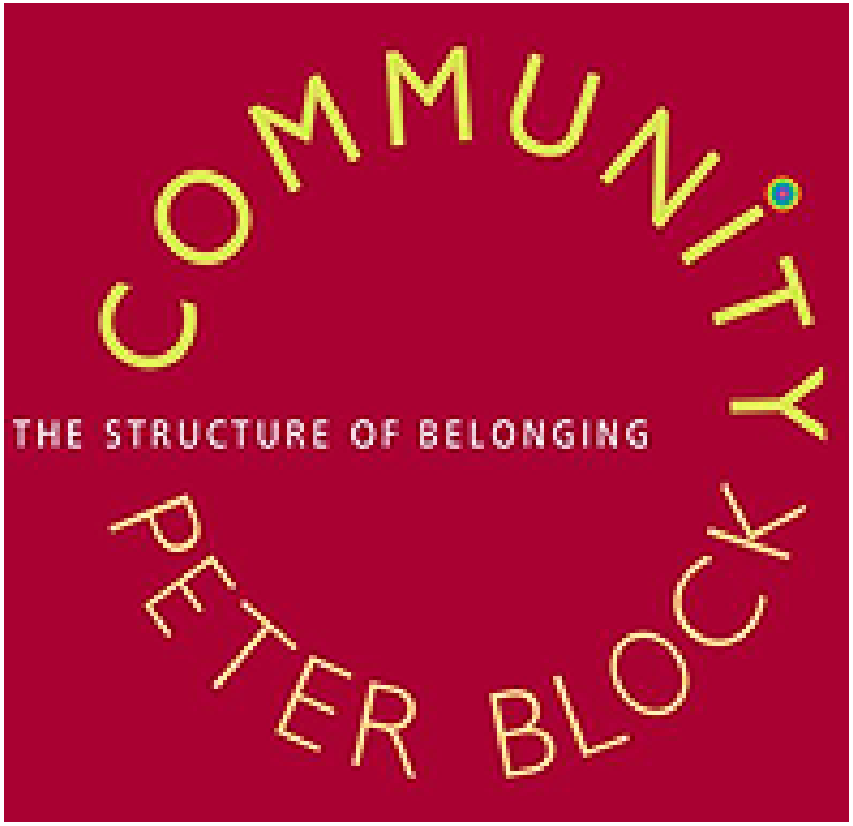
http://www.bioplanet.com/gcat

Scaling on full ecosystem of clusters

Platform LSF

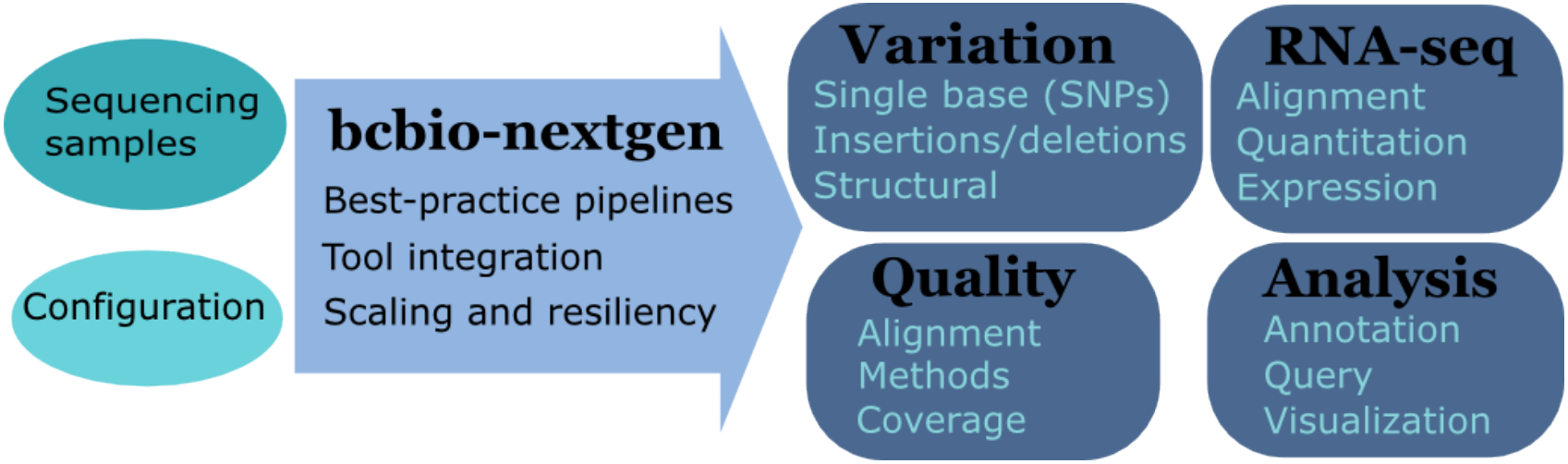
Torque

Solution



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

Overview


Sequencing samples
Configuration
bcbio-nextgen
Best-practice pipelines
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

Uses

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

Provides

- Best practice analysis pipelines
- Tool integration
- Multi-platform support
- Scaling

Development goals

- Community developed
- Quantifiable
- Scalable
- Reproducible

Community: installation

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

Coming soon

Docker

Community: documentation

bcbio-nextgen

Search docs

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 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

Code
Issues 15
Pull Requests 1
Pulse
Graphs
Network
Settings

HTTPS clone URL
https://github.com/

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

Validation

- Tests for implementation and methods
- Currently:
  - Germline variant calling
  - RNA-seq differential expression
- Expand to:
  - Cancer tumor/normal
    http://j.mp/cancer-var-chal
  - Family/population calling
  - Structural variations

Example evaluation

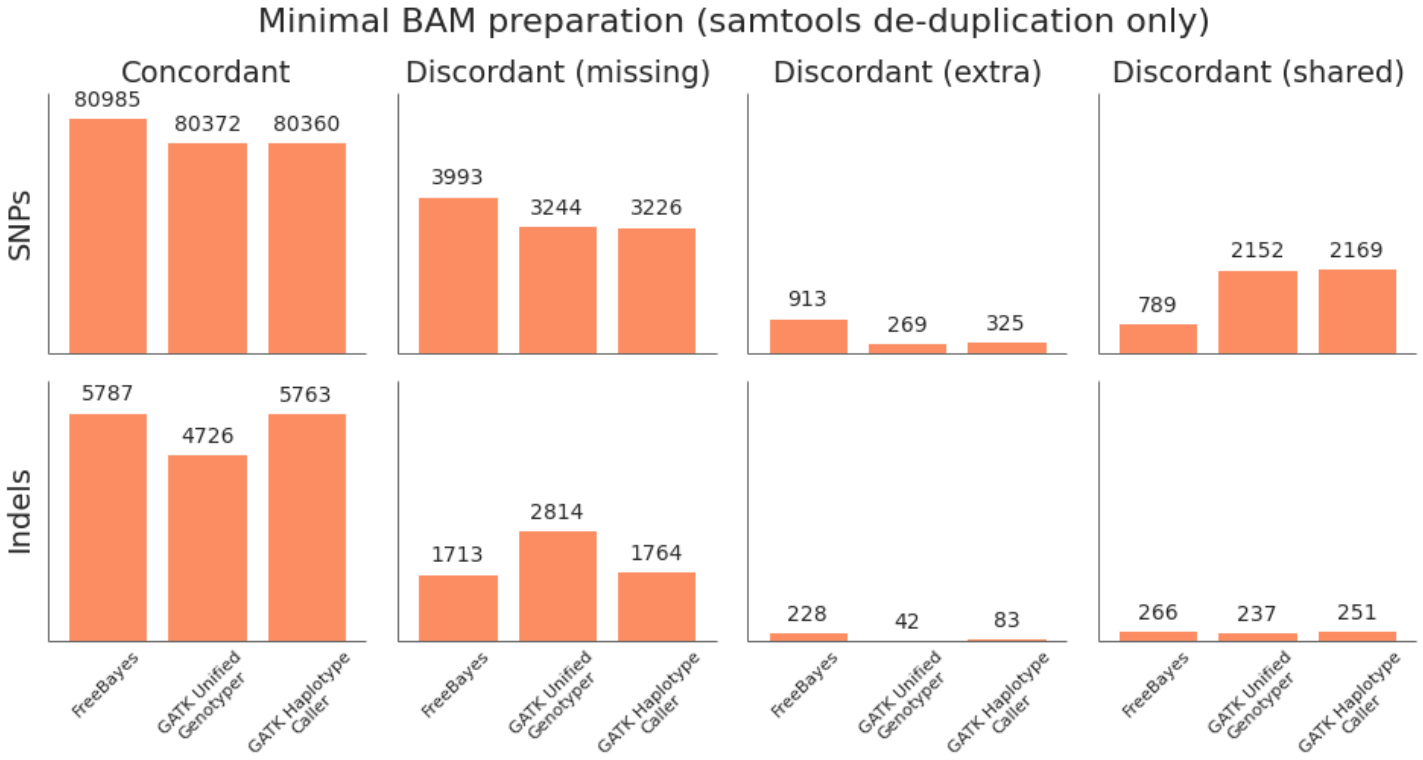
- Three variant callers
  - GATK UnifiedGenotyper
  - GATK HaplotypeCaller
  - FreeBayes
- Two preparation methods
  - Full (de-duplication, recalibration, realignment)
  - Minimal (only de-duplication)

Reference materials

Genome in a Bottle Consortium

http://www.genomeinabottle.org/

Quantify quality



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

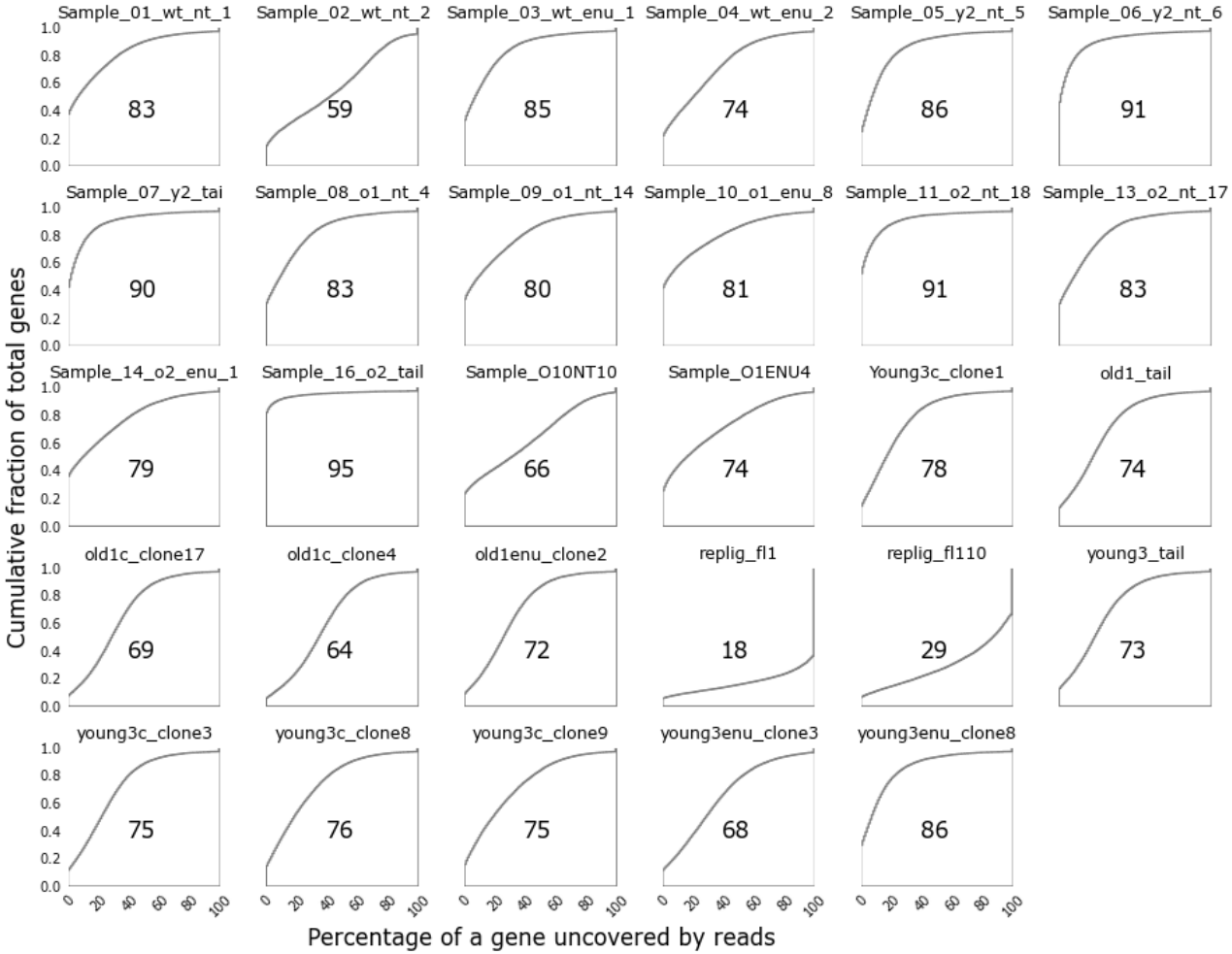
Results

- Haplotype aware callers better than UnifiedGenotyper
- FreeBayes performs on par with GATK HaplotypeCaller
- Little value in realignment when using haplotype aware caller
- Little value in recalibration when using high quality reads

Known unknowns

- Coverage: summarize what you can't assess
- Structural: large, complex rearrangements

Coverage plots



Analyze: GEMINI

GEMINI is a flexible framework for exploring genome variation.

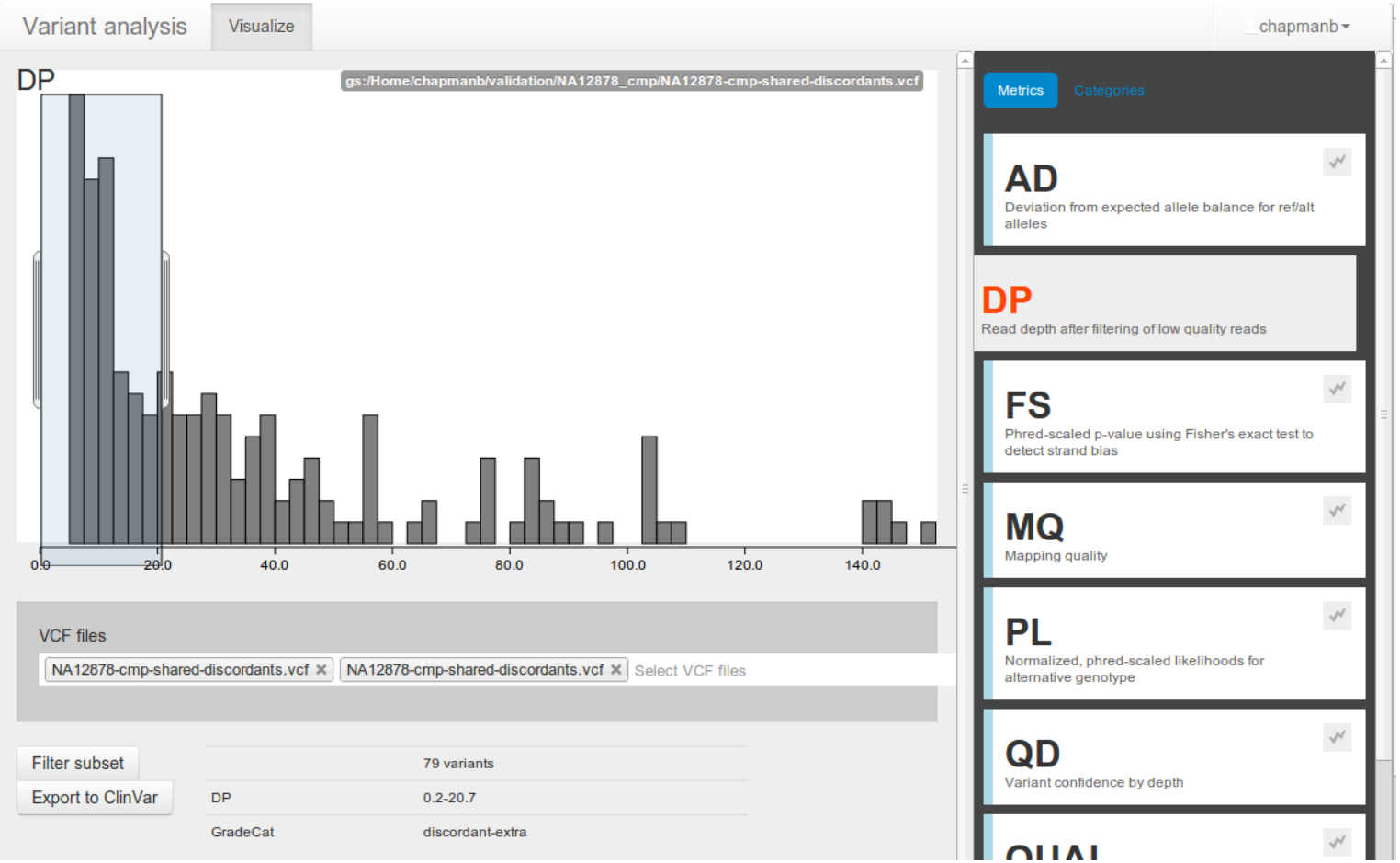
GEMINI: *a flexible framework for exploring genome variation*

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

Visualize



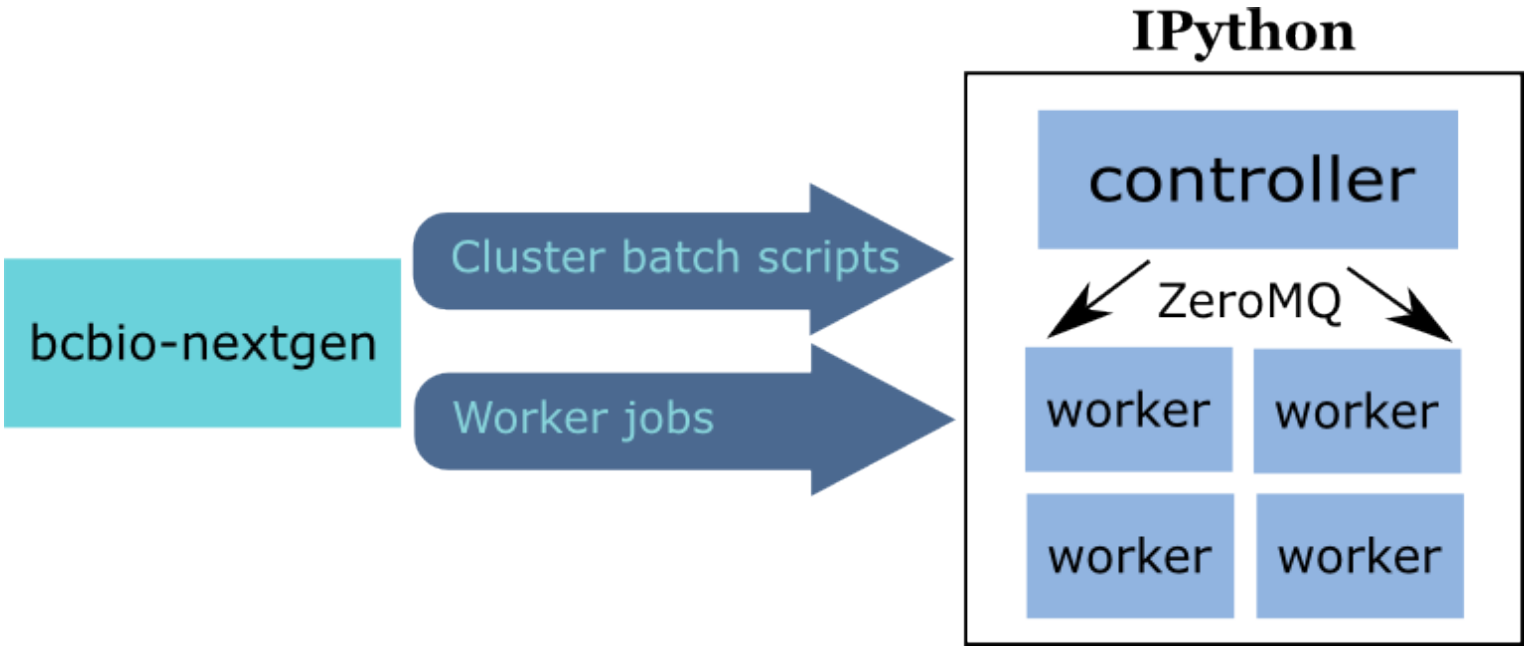
https://github.com/chapmanb/o8

RNA-seq evaluation

- Multiple differential expression callers
- External RNA Controls Consortium (ERCC) spike in analysis
- SEQC – 1000 qPCR genes
  http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE49712l

Rory Kirchner
https://github.com/roryk/bcbio.rnaseq

Scaling overview



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

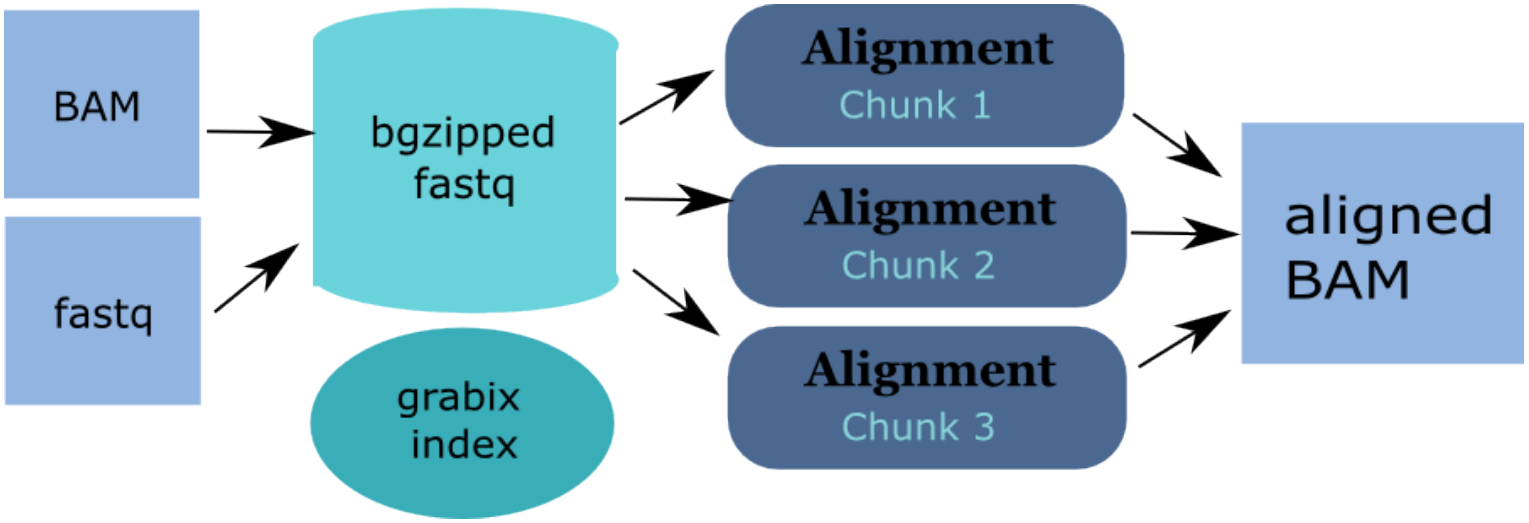
Current target environment

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

Scaling improvements

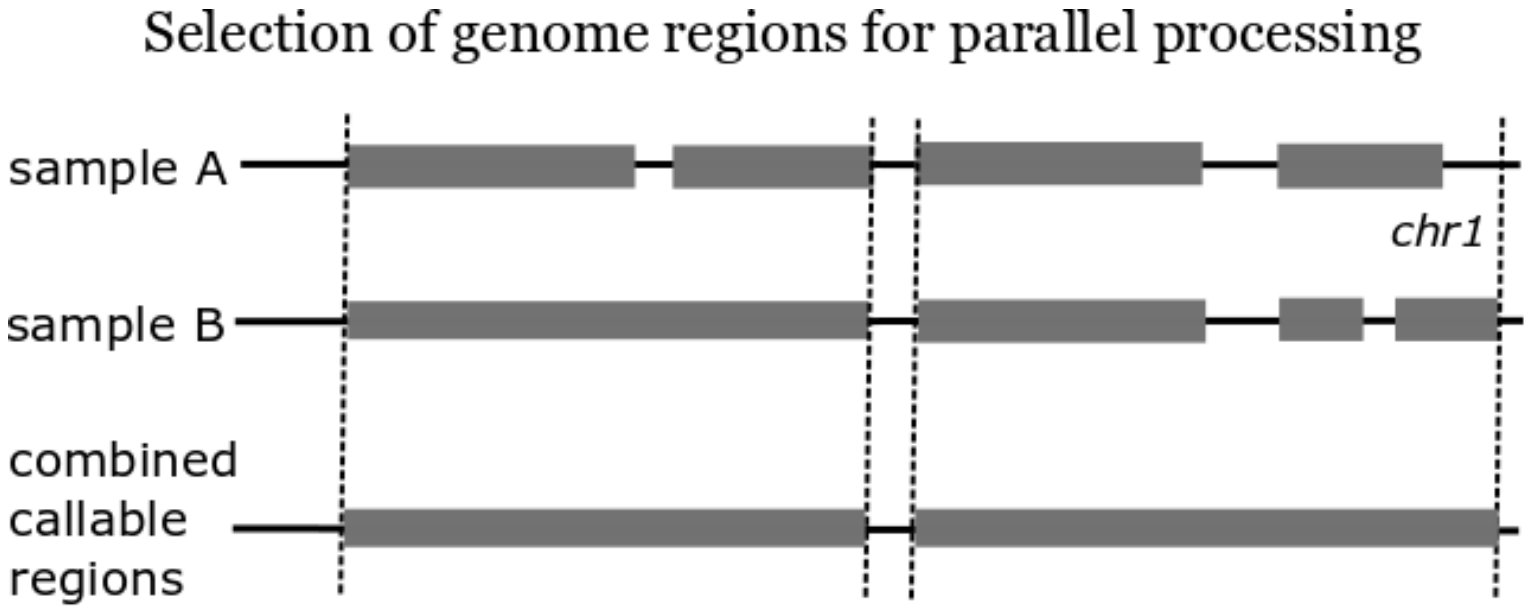
- Split alignments
- Split by genome regions
- Take advantage of multicore algorithms
- Manage memory
- Avoid IO

Alignment parallelization



https://github.com/arq5x/grabix

Variant calling and BAM preparation parallelization



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
```

Filesystem IO

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 System

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

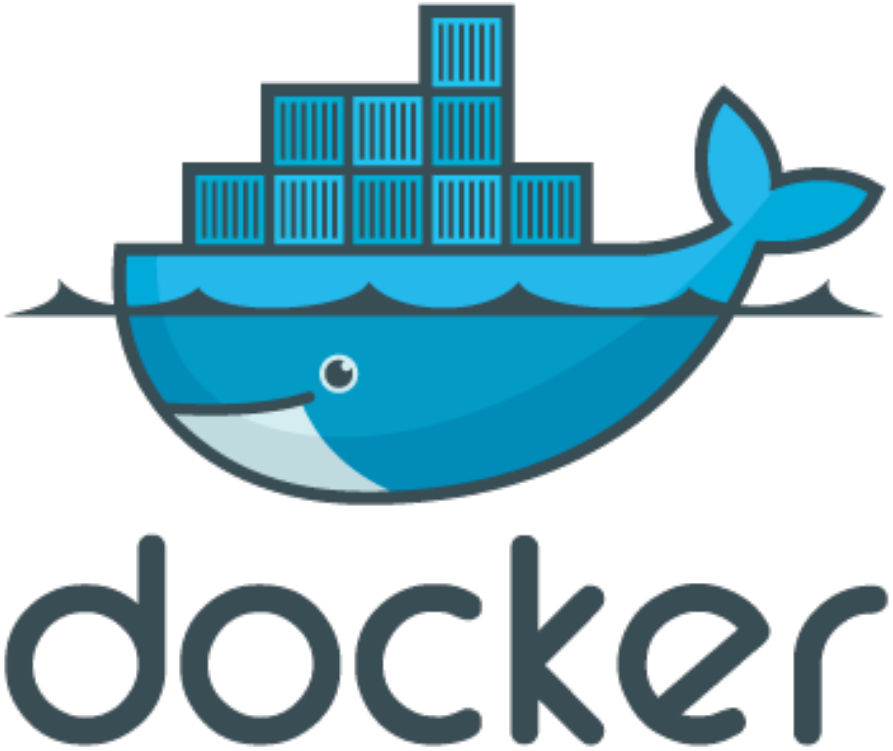
Timing: Analysis and QC

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

Reproducible environment



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

Docker benefits

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

Program provenance

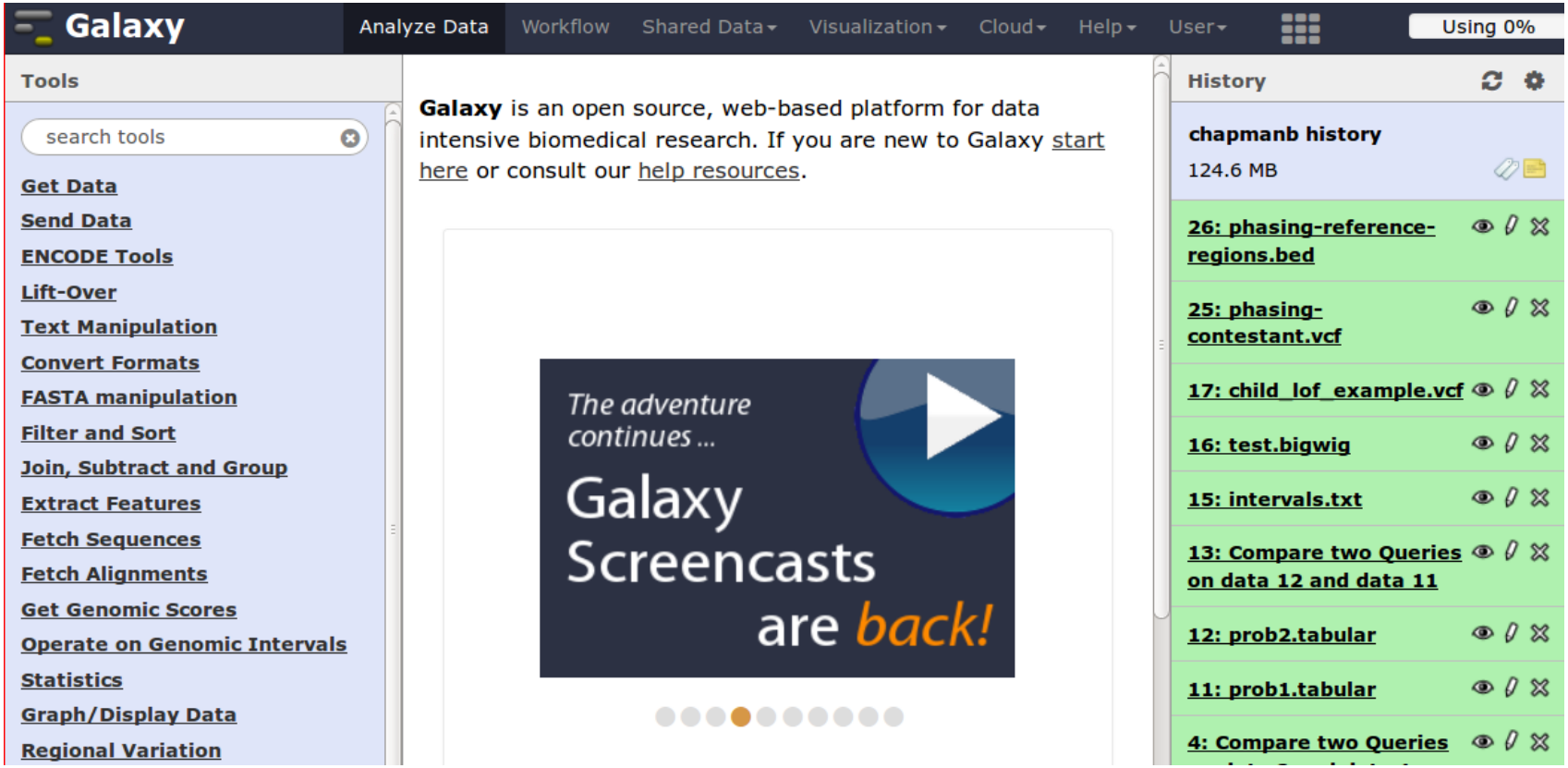
Arvados is a free and open source bioinformatics platform for genomic and biomedical data.

Store | Organize | Compute | Share

https://arvados.org/
https://curoverse.com/

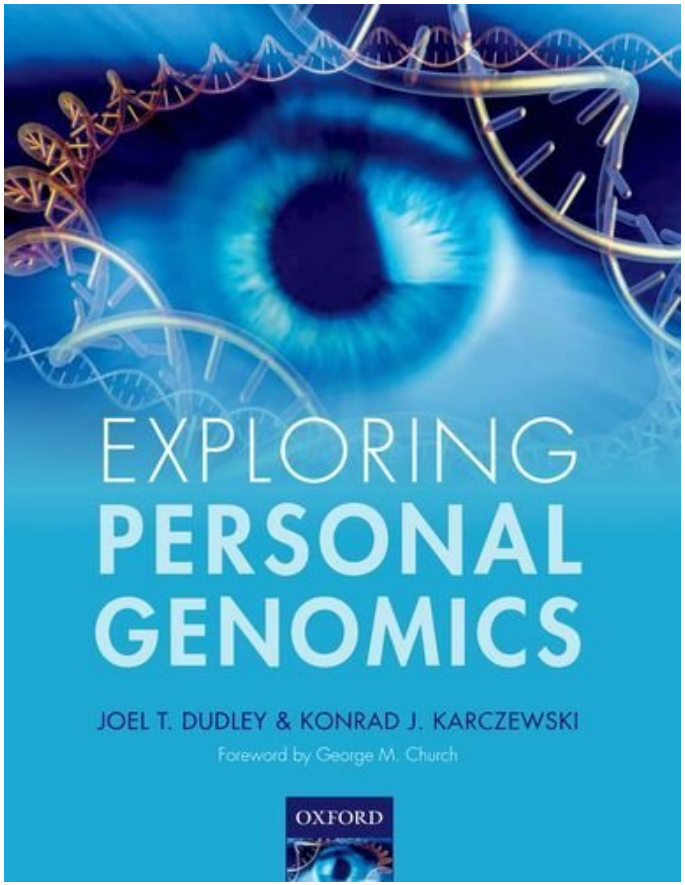
Scalable environment

Integrated



https://usegalaxy.org/

Accessible



http://exploringpersonalgenomics.org/

Summary

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

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