bcbio usage recommendations

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

1 March 2019

Goals

- Recommendations for bcbio runs
- Science: Variant calling
  - Small variants: germline and somatic
  - CNVs
  - Structural variations
  - Heterogeneity
- Practical aspects
  - Parallelization/CWL
  - Cloud/Hosted

Caveats

- Personal opinions
- Lots of choices in bcbio
- Will point out where likely to change over time

Germline small variants

- GATK4 HaplotypeCaller

```
variantcaller: [gatk-haplotype]
```

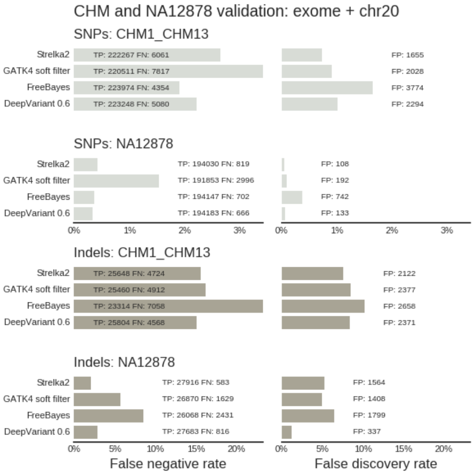
- Joint calling for scaling

```
tools_on: [gvcf]
```

Other germline options

- strelka2
- DeepVariant
- FreeBayes
- Octopus

Germline small variant validations



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

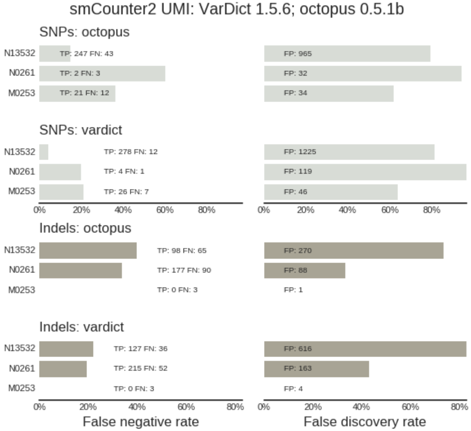
Somatic small variants: tumor only

- VarDict

```
variantcaller: [vardict]
```

- Other option: octopus. Performs well, not as fast in complex/deep regions

Somatic: tumor only validations



https://github.com/bcbio/bcbio_validations/tree/master/somatic-lowfreq

Somatic: tumor and normal

- VarDict
- MuTect2
- Strelka2
- Octopus

Somatic: tumor and normal

- Ensemble available
- Costs: runtime and non-standard output VCFs

```
variantcaller: [vardict, mutect2,
                strelka2, octopus]
ensemble:
  numpass: 2
```

https://bcbio-nextgen.readthedocs.io/en/latest/contents/configuration.html#ensemble-variant-calling

Somatic: tricky cases

- Low frequency variants in high depth panels: VarDict
- Similar suggestions to tumor only, tweak minimum allele fraction

```
variantcaller: [vardict]
min_allele_fraction: 0.5
```

CNVs: somatic, tumor only

- seq2c: uses other samples in project as background
- CNVkit: uses flat background

CNVs: somatic, tumor only PoN

- For tumor only with process matched normals
- Recommend generating a Panel of Normals (PoN)
- Supported
  - GATK4 CNV
  - seq2c
  - CNVkit

CNVs: tumor/normal

- GATK4 CNV
- seq2c
- CNVkit

CNVs: germline

- Work in progress
- GATK4 GermlineCNV will be recommendation
- CNVkit works now if you have case/control

https://github.com/bcbio/bcbio-nextgen/issues/2245

Structural variants – using one caller

- Manta
  - Best at reducing false positives
  - Not the most sensitive but will capture clear events

```
svcaller: [manta]
```

Structural variants – more sensitivity

- Lumpy
  - More sensitive, at the cost of additional false positives
  - Larger scale/complex events like fusions
  - Pair with prioritization

```
svcaller: [lumpy]
```

Structural variant prioritization

- Focus around genes of interest
- Summarize from multiple callers
- Provides useful practical filter

```
svcaller: [manta, lumpy]
svprioritize; cancer/civic
```

https://bcbio-nextgen.readthedocs.io/en/latest/contents/configuration.html#structural-variant-calling

Heterogeneity overview

- Estimation of purity/ploidy
- Allele specific copy number calling
- HLA Loss of heterozygosity
- LOH + amplification
  - Disease specific genes of interest (from CIViC)

Heterogeneity options

- TitanCNA:
  - tumor/normal
  - exome or bigger
- PureCN
  - tumor/normal
  - panels or bigger

https://github.com/bcbio/bcbio_validations/tree/master/TCGA-heterogeneity

Heterogeneity details

Inputs:

- Variant calls from VarDict
- CNV calls from GATK4 CNV

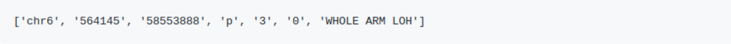
Heterogeneity configuration

```
algorithm:
 variantcaller: [vardict]
 svcaller: [gatk-cnv, purecn, titancna]
 svprioritize: cancer/civic
metadata:
 disease: lung
```

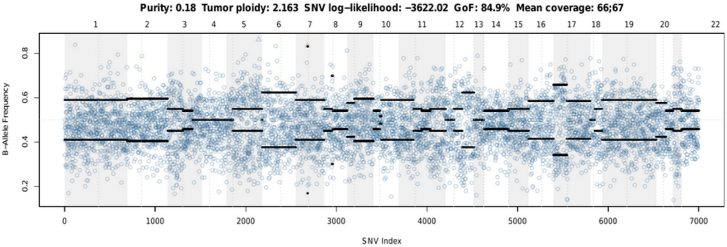
Heterogeneity output: genes

```
LOH:
  CDKN2A: LOH
  HLA: ’no’
amplification:
  AKT2: ’no’
  EGFR: amplification
ploidy: ’1.6921836479328’
purity: ’0.84’
```

Heterogeneity output: plots



```
['chr6', '564145', '58553888', 'p', '3', '0', 'WHOLE ARM LOH']
```



Heterogeneity future work

- PureCN can support tumor-only panels
- Need panel of normals for proper CNV calls
- Pending additional work on PoN calling

Practical running suggestions

When to run bcbio with what where and why

2 options:

- Original/old approach
  - multicore for local
  - IPython on cluster
- Common Workflow Language (CWL)
  - Cromwell runner: local + cluster + GCP
  - Hosted support: DNAnexus, Arvados, SevenBridges

Common Workflow Language

- Future (and present) of bcbio runs
- bcbio generates a workflow and supplies tools + implementation
- Supports Docker (and in the future Singularity)
- Uses external runners: Cromwell

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

Original bcbio runner

- Runs single/few samples multicore on single machines
- Uses IPython for distribution on local clusters
- Easier to debug than equivalent CWL runs right now

Cloud, single machine

- Traditional bcbio runner with multicore
- Spin up single machine, attach external stable EBS volume
- Use ansible to attach and provision
- Bigger machine size during runs, take down when finished

https://github.com/bcbio/bcbio-nextgen/tree/master/scripts/ansible

Cloud, multiple machines

- Common Workflow Language
- Google Cloud Platform, Google Pipelines API
- Cromwell runner
- bcbio Docker containers

https://bcbio-nextgen.readthedocs.io/en/latest/contents/cloud.html#docs-cloud-gcp

Hosted cloud

- Common Workflow Language
- Platform specific runners
  - Arvados
  - DNAnexus
  - SevenBridges
- Public genome resources available

https://bcbio-nextgen.readthedocs.io/en/latest/contents/cwl.html#running-on-arvados-hosted-cloud

Summary of recommendations

- Science: Variant calling
    - Small variants: germline and somatic
    - CNVs
    - Structural variations
    - Heterogeneity
- Practical aspects
    - Parallelization/CWL
    - Cloud/Hosted