Title	Building a community menagerie of automated variant validations
Authors	*Brad Chapman*, Rory Kirchner, Lorena Pantano, Shannan Ho Sui, Ward Vandewege, Sasha Wait Zaranek, Justin Johnson, Oliver Hofmann
Affiliations	Harvard Chan School Bioinformatics Core, Veritas Genetics, Curoverse Research, AstraZeneca Oncology, The University of Melbourne Centre for Cancer Research
Contact	bchapman@hsph.harvard.edu
Availability	https://github.com/bcbio/bcbio-nextgen, https://github.com/bcbio/bcbio_validation_workflows
Documentation	https://bcbio-nextgen.readthedocs.org/en/latest/contents/cwl.html
License	MIT

bcbio (https://github.com/chapmanb/bcbio-nextgen) is an open, community effort to develop validated and scalable variant calling, RNA-seq, CHiP-seq and small RNA analyses. bcbio runs across a wide variety of platforms from full stack cloud providers to local high performance computing environments by leveraging Common Workflow Language (CWL: http://www.commonwl.org/) and the GA4GH iteroperability standards (https://github.com/ga4gh/wiki/wiki),

bcbio integrates production ready analysis pipelines with automated validations using reference materials developed by communities like Genome in a Bottle (http://jimb.stanford.edu/giab/) and the International Cancer Genome Consortium (https://icgc.org/). This creates a multi-plaform test suite for method comparisons across a wide variety of biological analyses, as well as a baseline for adjusting and improving existing methods.

We'll highlight four useful validations from our collection of workflows:

- GATK4: an evaluation of the new open source GATK4 variant caller against GATK3 and other recent methods like strelka2. https://github.com/bcbio/bcbio_validations/tree/master/gatk4
- DeepVariant and CHM reference materials: compares Google's Neural Network based caller against other standard methods, including comparisons on an orthogonal haploid/diploid based truth set to evaluate training bias. https://github.com/bcbio/bcbio_validations/tree/master/deepvariant
- Value of trimming in low frequency somatic variant detection: explores the impact of 3' quality and low complexity trimming on runtime and quality, helping remove bottlenecks in whole genome variant calling. https://github.com/bcbio/bcbio_validations/tree/master/somatic_trim
- Structural variant detection sensitivity for long and short reads: identifies limits of detection for short read methods based on comparisons with resolvable large scale events in long reads. https://github.com/bcbio/bcbio_validations/tree/master/NA24385_sv

The goal of this work is to coordinate with the bioinformatics community to build automated variant test suites that run on any platform of choice. Regular automated builds: ensure that tools are always functional across platforms, create a historical account of method performance, evaluate new methods, and give reseachers a robust and validated way to run variant analyses to answer difficult biological questions.