Title	Promoting platform interoperability with portable bcbio workflows
Authors	*Brad Chapman*, Rory Kirchner, Lorena Pantano, Peter Amstutz, Alexander Zaranek, Shannan Ho Sui, Oliver Hofmann
Affiliations	Harvard Chan School Bioinformatics Core (http://bioinformatics.sph.harvard.edu/), Curoverse (https://curoverse.com/), Wolfson Wohl Cancer Research Centre (http://www.gla.ac.uk/researchinstitutes/cancersciences/ics/facilities/wwcrc/)
Contact	bchapman@hsph.harvard.edu
Availability	https://github.com/chapmanb/bcbio-nextgen
License	MIT

Running multi-step bioinformatics analyses requires coordinating software and data across a wide variety of heterogeneous computational resources. We've actively developed bcbio (https://github.com/chapmanb/bcbio-nextgen) for the past six years as a open, community built approach to developing variant calling, RNA-seq and small RNA analyses. The complexity of supporting scalable parallel workflows has become a barrier to allowing bcbio to interact with other open source platforms.

bcbio previously used a parallelization framework build on IPython parallel (https://ipyparallel.readthedocs.org) that runs on both local compute infrastructure (https://bcbio-nextgen.readthedocs.org/en/latest/contents/parallel.html) and on cloud resources (https://bcbio-nextgen.readthedocs.org/en/latest/contents/cloud.html). This approach unintentionally isolated bcbio development. For example, we could not easily deploy bcbio on community developed systems like Galaxy (https://galaxyproject.org/) due to different approaches to running compute jobs. This incompatibility results in duplication of effort as bcbio develops and tests system specific parallel code, while communities like Galaxy need to re-implement validated and tested analyses available in bcbio.

We re-engineered bcbio's internal workflow representation to use the Common Workflow Language (CWL: http://www.commonwl.org/). By using this community standard, users choose an infrastructure that matches their usage requirements. First build a bcbio parallel workflow directly from existing sample description files (https://bcbio-nextgen.readthedocs.org/en/latest/contents/cwl.html), then choose the appropriate run environment. A clinical lab requiring full data provenance could run the generated bcbio CWL using Arvados (https://arvados.org/). Research teams with local compute could use an alternative engine like Toil (https://github.com/BD2KGenomics/toil). By interoperating with other workflows, bcbio supplements existing infrastructure and analysis development within each system.

We'll discuss the challenges of migrating to CWL versus the benefits of being able to integrate within multiple platforms. bcbio is now a better architected, more portable set of validated tools and workflows to help scientists answer biological questions. We focus on developing analysis methods and validations while CWL supporting tools focus on other essential functionality like run tracking, multi-architecture support, resource usage assessment, provenance and data management. We hope to promote the continued exploration of ways to re-use and cooperate more effectively as an open source community.