

# Community development of validated variant calling pipelines

Brad Chapman<sup>1</sup>, Rory Kirchner<sup>1</sup>, Oliver Hofmann<sup>1</sup> and Winston Hide<sup>1,\*</sup>

<sup>1</sup>Bioinformatics Core, Harvard School of Public Health, Department of Biostatistics, Boston, MA USA

Correspondence\*:

Winston Hide

Harvard School of Public Health, 655 Huntington Avenue Building II, Room 415  
Boston, Massachusetts 02115 USA, [whide@hsph.harvard.edu](mailto:whide@hsph.harvard.edu)

Bioinformatics and genetic variation exploration

## ABSTRACT

Exploratory and translational research relies on accurate identification of genomic variants from populations, families and cancer tumor/normal pairings. However, rapidly changing best practice approaches in alignment and variant calling, coupled with large data sizes, make it a challenge to develop scalable, accurate pipelines. Coordinated community development overcomes these challenges by sharing testing and updates across groups relying on the same infrastructure.

bcbio-nextgen is a distributed multi-architecture pipeline that automates variant calling, validation and organization of results for query and analysis. It creates an easily installable and widely runnable infrastructure from best-practice tools, coupled with an integrated methodology for assessing variant quality. We use the bcbio-nextgen framework to provide comparisons of variant calling methods and pipelines and identify key bottlenecks for scaling to large collections of whole genome sequencing data.

The open-source, community-developed framework is freely available from <https://github.com/chapmanb/bcbio-nextgen>.

**Keywords:** Next Generation Sequencing, Variant detection, Quality control and validation, Community development, Open source software

## INTRODUCTION

bcbio-nextgen provides a variant calling pipeline to accurately detect single nucleotide polymorphisms (SNPs) and insertions/deletions (indels) from high throughput sequencing data. It utilizes multiple best practice approaches for alignment, alignment post-processing and variant calling, provides an integrated mechanism to assess variant quality and interfaces with downstream tools for variant analysis. Practically, it installs with a single command on multiple computing architectures, scales to large whole genome analyses, and is community developed. The goal is to provide a platform for moving from raw sequencing data to high-quality variant calls that evolves as algorithms and sequencing technologies change.

The pipeline utilizes existing algorithms, wrapping them in an easy to use and scalable way. It provides software programming interfaces to enable new tools and currently builds on a large number of reusable software packages:

- 29 • Alignment: bwa (**Li and Durbin**, 2010), bwa-mem (**Li**, 2013) and novoalign (**novoalign**, 2013)
  - 30 • BAM alignment processing: samtools (**Li et al.**, 2009), bamtools (**bamtools**, 2013), Picard (**Picard**,  
31 2013), sambamba (**sambamba**, 2013) and pysam (**pysam**, 2013)
  - 32 • Interval manipulation: BedTools (**Quinlan and Hall**, 2010) and pybedtools (**Dale et al.**, 2011)
  - 33 • Variant calling: GATK (**DePristo et al.**, 2011) and FreeBayes (**Garrison and Marth**, 2012)
- 34 The pipeline reports variant calling results both in standard VCF format and as a ready to query database,  
35 making results analyzable by both bioinformaticians and biologists familiar with SQL and command line  
36 tools. snpEff (**Cingolani et al.**, 2012) predicts effects associated with identified variation and GEMINI  
37 provides the SQLite database associating variants with a wide variety of genome annotations (**Paila et al.**,  
38 2013).
- 39 Three major components of bcbio-nextgen differentiate it from both existing tools like HugeSeq (**Lam**  
40 **et al.**, 2012) and customized in-house scripts:
- 41 • Quantifiable: It validates variant calls against known reference materials developed by the Genome  
42 in a Bottle consortium (**Zook et al.**, 2013). Automating scoring and assessment of calls allows  
43 identification of improvements or regressions in variant identification as calling pipelines evolve.  
44 Incorporation of multiple variant calling approaches from Broad's GATK best practices (**Van der**  
45 **Auwer et al.**, 2002) and the Marth lab's FreeBayes caller (**Garrison and Marth**, 2012) enables  
46 informed comparisons between algorithms.
  - 47 • Scalable: bcbio-nextgen handles large population studies with hundreds of whole genome samples by  
48 parallelizing on a wide variety of schedulers (LSF, SGE, Torque, SLURM) and multicore machines.
  - 49 • Community developed: Due to the focus on solving the problems of setting up and maintaining a  
50 complex analysis pipeline, multiple sequencing centers and research laboratories use bcbio-nextgen.  
51 We actively encourage contributors to the code base and make it easy to get started with a fully  
52 automated installer and updater that prepares all third party software and reference genomes.

## VALIDATION

53 Alignment and variant calling algorithms are both diverse and rapidly changing. Tools quickly become  
54 outdated and new approaches provide improved resolution and speed. This continuous change requires  
55 flexible pipelines that can incorporate new methods and iterate rapidly in response to updated tools. It also  
56 requires integrated methods for ensuring that variant calling accuracy improves with new changes, and for  
57 evaluating new methodologies against established best practice.

58 bcbio-nextgen includes an automated approach to validate calling methods against known reference  
59 materials. A high quality NA12878 reference genome developed by the Genome in a Bottle consortium  
60 (**Zook et al.**, 2013) provides a baseline dataset for comparison. The evaluation dataset is a NA12878  
61 clinical exome contributed by EdgeBio. The process of retrieving the data and running the evaluation is  
62 fully documented (<https://bcbio-nextgen.readthedocs.org/en/latest/contents/testing.html#exome-with-validation-against-reference-materials>).  
63

64 As an example of the usefulness of having this integrated validation system, we evaluated three different  
65 current variant callers:

- 66 • FreeBayes (v0.9.9.2-18): A haplotype-based Bayesian caller, filtering calls with a hard filter based on  
67 depth and quality (DP ≥ 5; QUAL ≥ 20).
- 68 • GATK UnifiedGenotyper (2.7-2): GATK's Bayesian caller, with calls filtered using GATK's  
69 recommended hard filters.

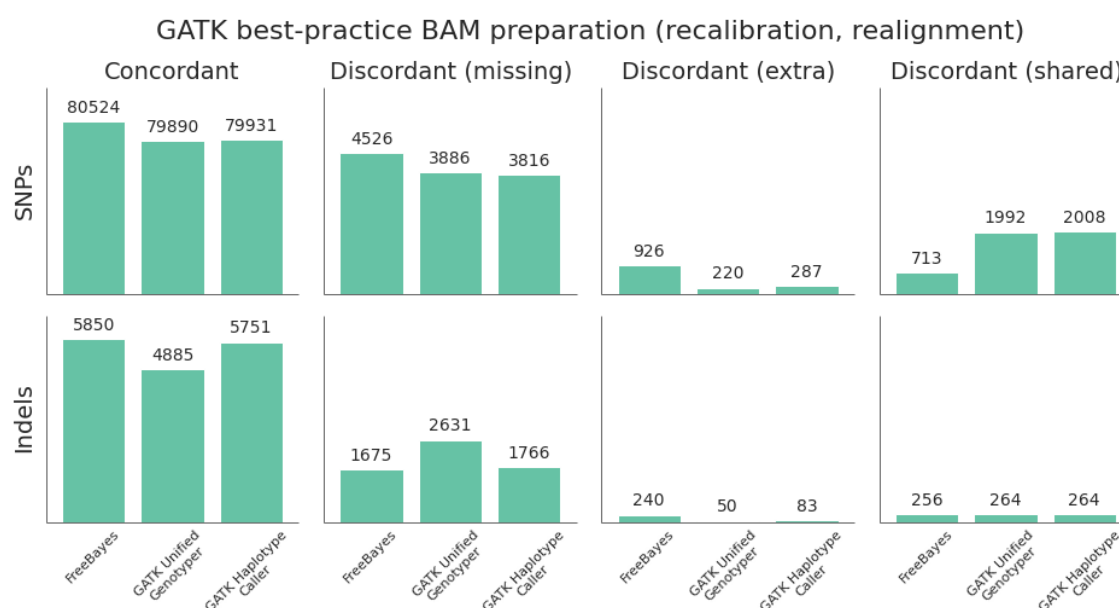
70 • GATK HaplotypeCaller (2.7-2): A GATK variant caller which local haplotype re-assembly around  
 71 variant regions. We also filtered these calls using recommended hard filters.

72 Following alignment with bwa-mem (0.7.5a), we additionally post-processed the BAM files with two  
 73 methods:

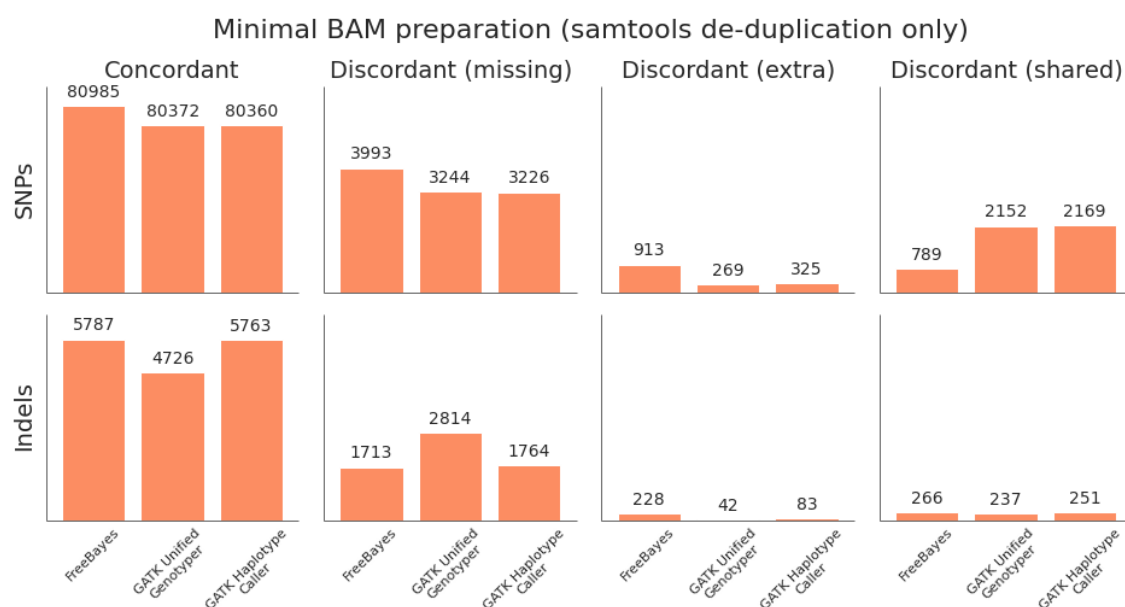
- 74 • GATKs best practices (2.7-2): This involves de-duplication with Picard MarkDuplicates, GATK base  
 75 quality score recalibration and GATK realignment around indels.  
 76 • Minimal post-processing: with de-duplication using samtools rmdup and no realignment or  
 77 recalibration.

78 Figures 1 and 2 show comparisons to the Genome in a Bottle reference materials for the GATK best  
 79 practice and minimal BAM preference methods, respectively.

80 The comparison identified three areas to consider for future variant calling. First, FreeBayes outperforms  
 81 the GATK callers on both SNP and indel calling. The most recent versions of FreeBayes have  
 82 improved sensitivity and specificity which puts them on par with GATK HaplotypeCaller. Second,  
 83 GATK HaplotypeCaller is all around better than the UnifiedGenotyper. In previous GATK versions,  
 84 UnifiedGenotyper performed better on SNPs and HaplotypeCaller better on indels, but the recent  
 85 improvements in GATK 2.7 have resolved the difference in SNP calling. Finally, skipping base  
 86 recalibration and indel realignment had almost no impact on the quality of resulting variant calls when  
 87 using a realigning caller. While GATK UnifiedGenotyper suffers during indel calling without recalibration  
 88 and realignment, both HaplotypeCaller and FreeBayes perform as good or better without these steps.



**Figure 1.** Validation results for three variant calling methods using GATK best practice post-alignment preparation (de-duplication, realignment around indels and base quality recalibration). Realigning variant callers (GATK HaplotypeCaller and FreeBayes) have equal sensitivity and specificity in calling SNPs but provide improved resolution of indels. FreeBayes performs on par with GATK HaplotypeCaller methods. Discordant variants are in 3 categories. Missing: not present in evaluation but present in reference (potential false negatives), Extra: present in evaluation but not in reference (potential false positives). Shared: present in both but different due to heterozygote/homozygote call or alleles called.



**Figure 2.** Validation results for three variant calling methods using minimal BAM post-alignment preparation methods (only de-duplication). Results are equivalent to those seen using the more time intensive GATK best practice approach when using realigning variant callers (GATK HaplotypeCaller and FreeBayes).

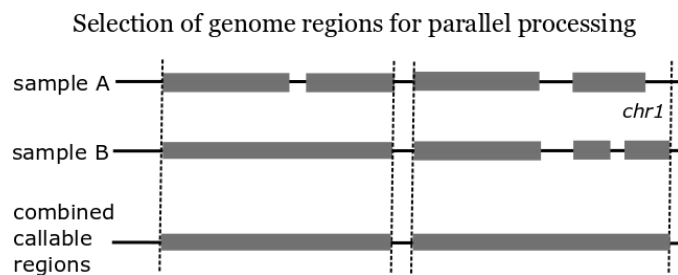
89 The main benefit of validation is to enables experiments that quantitatively assess widely held  
 90 approaches. We expect best practices to change with new releases and algorithms, and the automated  
 91 assessment mechanism allows bcbio-nextgen to track and adapt to continuously improving tools.

## SCALING

92 The second differentiating feature of bcbio-nextgen is the ability to scale to handle large population whole  
 93 genome datasets on a wide variety of architectures. The pipeline runs in parallel on single multicore  
 94 machines or on clusters with a shared filesystem and scheduler. It utilizes the general purpose IPython  
 95 parallel infrastructure (**IPython**, 2013), which supports multiple schedulers including LSF, SGE, SLURM,  
 96 and Torque. This infrastructure allows jobs to adapt to increased scale or system changes without adjusting  
 97 the underlying configuration or code.

98 To utilize large cluster architectures, bcbio-nextgen parallelizes processing at multiple steps:

- 99 • Alignment – Block gzipping (bgzip) and indexing of sequencing reads with grabix (**grabix**, 2013)  
 100 allows alignment processing in blocks. The split size for each alignment is configurable to match  
 101 available processing cores.
- 102 • Alignment post-processing – Following alignment the pipeline assesses callable regions in each  
 103 sample, identifying regions with no coverage to use as analysis breakpoints. Each chromosomal  
 104 region between regions of no coverage provides an independent section for parallel processing  
 105 (Figure 3).
- 106 • Variant calling – Variant processing parallelizes using the same chromosomal blocks identified  
 107 during alignment post-processing. For population and family based calling, variant calls occur  
 108 simultaneously for all batched samples in a region.



**Figure 3.** Identification of shared no coverage regions between multiple samples. Each no coverage region breaks the genome into chunks allowing parallel processing.

**Table 1.** Processing times for 60 whole genome Illumina samples (30x) on 400 cores

Step	Time	Processes
Alignment preparation	24 hours	BAM to fastq; bgzip; tabix index
Alignment	36 hours	bwa-mem alignment and BAM merging
Alignment post-processing	9 hours	Calculate callable regions
Post-alignment BAM preparation	12 hours	De-duplication
Variant calling	23 hours	FreeBayes
Variant post-processing	2 hours	Combing variant files; annotate with GATK and snpEff
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
Total	5 days	

- Identification of callable regions, BAM merging and indexing, quality control, and GEMINI database preparation – All of these steps allow shared memory parallel processing, which the pipeline enables by launching cluster jobs with multiple cores on a single machine.

Within each independently running parallel process, bcbio-nextgen controls memory usage and disk IO to maximize the throughput of multiple simultaneous processes. An input configuration files specifies available memory usage for programs that allow memory restrictions, and expected memory usage for those that do not. These inputs allow an accurate estimate of memory consumption and bcbio-nextgen avoids overscheduling jobs relative to available memory on each machine. Similarly, simultaneous disk IO on shared filesystems is a common bottleneck during processing. bcbio-nextgen minimizes this by use of streaming piped processing steps where supported by the underlying tools. As an example, the alignment steps converts output into standard sorted BAM files via use of unix pipes, avoiding writing intermediates to disk.

These scaling approaches enable simultaneous processing of large whole genome population samples. As an example, Table1 shows timing results from running 60 whole genome Illumina samples with 30x coverage through a full alignment, variant calling, analysis and quality control pipeline. The example uses Dell’s Active Infrastructure (Dell, 2013) with 400 cores, 3Gb of memory per core and a Lustre filesystem connected on an Infiniband network. Processing 60 samples in 5 days is an effective time of 2 hours/sample when processing large families. Single whole genome samples typically process in less than a day with 100 cores.

## COMMUNITY DEVELOPMENT

A final unique aspect of the pipeline is a strong focus on community development and broad usability. We believe that good quality scalable variant calling is a shared problem faced in multiple research laboratories, core facilities and companies. By pooling resources, a community developed framework overcomes the inherent difficulties associated with maintaining and extending rapidly changing pipelines.

To achieve wide usability, bcbio-nextgen installs on multiple unix-based operating systems and cluster types. An automated installer built on CloudBioLinux (**Krampis et al.**, 2012) installs both the bcbio-nextgen Python framework as well as all associated tools and pre-indexed genomic data. Installation is fully documented (<https://bcbio-nextgen.readthedocs.org/en/latest/contents/installation.html>) and uses the same automated process to provide updates for new versions of the pipeline and tools. It includes a full test suite as well as example exome and genome datasets for ensuring correct installation and scaling (<https://bcbio-nextgen.readthedocs.org/en/latest/contents/testing.html>).

## FUTURE WORK

By removing installation and infrastructure integration hurdles, bcbio-nextgen has an active user community with regular contributions from outside our core group. We continue to actively develop the framework to increase the scope and currently active projects include:

- Coverage – Assessment of coverage in gene regions of interest, allowing identification of regions without effective coverage for calling.
- Structural variation – Detection of large scale events (duplications, deletions and inversions) as well as identification of copy number variations.
- Cancer tumor/normal – Integration and evaluation of cancer-specific paired callers.
- RNA-seq – Identify differentially expressed transcripts and evaluate performance of aligners and transcript resolution methods.
- Cloud computing – Enable native support for cloud providers such as Amazon, making the pipeline readily usable by researchers without local compute infrastructure.
- Reproducibility and provenance – Provide versioned, locally isolated Linux containers using Docker to improve the ability to trace and re-run analyses.
- Accessibility – Interface with web-based biologist targeted front ends such as Galaxy (**Goecks et al.** (2010); **Giardine et al.** (2005); **Blankenberg et al.** (2010)).

In summary, bcbio-nextgen provides an automated pipeline to identify and validate genomic variations in high throughput sequencing data. The pipeline scales to handle large population studies by minimizing computational bottlenecks and integrating with multiple cluster architectures. The pipeline is open-source, documented and we welcome community contributions.

## ACKNOWLEDGMENTS

Thanks to John Morrissey and James Cuff at Harvard Faculty of Arts and Science Research Computing, and Glen Otero and William Cottay at Dell Life Sciences for compute infrastructure and support. Thank you to the open-source community for feedback, reports, code fixes and contributions: Miika Ahdesmaki, Luca Beltrame, Guillermo Carrasco, Peter Cock, Mario Giovacchini, Jakub Nowacki, Brent Pedersen, James Porter, Valentine Svensson, Paul Tang, Roman Valls, and Kevin Ying. Justin Johnson and David Jenkins at EdgeBio contributed the NA12878 evaluation exome dataset.



166 *Funding* Funding for variation evaluation provided by the Archon Genomic X Prize foundation.

## REFERENCES

- 167 bamtools (2013), <https://github.com/pezmaster31/bamtools>
- 168 Blankenberg, D., Kuster, G. V., Coraor, N., Ananda, G., Lazarus, R., Mangan, M., et al. (2010), Galaxy: A
- 169 web-based genome analysis tool for experimentalists, in *Current Protocols in Molecular Biology* (John
- 170 Wiley & Sons, Inc.)
- 171 Cingolani, P., Platts, A., Wang, L. L., Coon, M., Nguyen, T., Wang, L., et al. (2012), A program for
- 172 annotating and predicting the effects of single nucleotide polymorphisms, SnpEff: SNPs in the genome
- 173 of *drosophila melanogaster* strain w1118; iso-2; iso-3, *Fly*, 6, 2, 80–92, doi:10.4161/fly.19695, PMID:
- 174 22728672
- 175 Dale, R. K., Pedersen, B. S., and Quinlan, A. R. (2011), Pybedtools: a flexible python library for
- 176 manipulating genomic datasets and annotations, *Bioinformatics (Oxford, England)*, 27, 24, 3423–3424,
- 177 doi:10.1093/bioinformatics/btr539, PMID: 21949271
- 178 Dell (2013), <http://dell.com/ai-hpc-lifesciences>
- 179 DePristo, M. A., Banks, E., Poplin, R., Garimella, K. V., Maguire, J. R., Hartl, C., et al. (2011), A
- 180 framework for variation discovery and genotyping using next-generation DNA sequencing data, *Nature*
- 181 *genetics*, 43, 5, 491–498, doi:10.1038/ng.806, PMID: 21478889
- 182 Garrison, E. and Marth, G. (2012), Haplotype-based variant detection from short-read sequencing,
- 183 *arXiv:1207.3907 [q-bio]*
- 184 Giardine, B., Riemer, C., Hardison, R. C., Burhans, R., Elnitski, L., Shah, P., et al. (2005), Galaxy: A
- 185 platform for interactive large-scale genome analysis, *Genome Research*, 15, 10, 1451–1455, doi:10.
- 186 1101/gr.4086505, PMID: 16169926
- 187 Goecks, J., Nekrutenko, A., Taylor, J., and \$author.lastName, a. f. (2010), Galaxy: a comprehensive
- 188 approach for supporting accessible, reproducible, and transparent computational research in the life
- 189 sciences, *Genome Biology*, 11, 8, R86, doi:10.1186/gb-2010-11-8-r86, PMID: 20738864
- 190 grabix (2013), <https://github.com/arq5x/grabix>
- 191 IPython (2013), <http://ipython.org/>
- 192 Krampis, K., Booth, T., Chapman, B., Tiwari, B., Bicak, M., Field, D., et al. (2012), Cloud
- 193 BioLinux: pre-configured and on-demand bioinformatics computing for the genomics community,
- 194 *BMC Bioinformatics*, 13, 42, doi:10.1186/1471-2105-13-42, PMID: 22429538 PMCID: PMC3372431
- 195 Lam, H. Y. K., Pan, C., Clark, M. J., Lacroute, P., Chen, R., Haraksingh, R., et al. (2012), Detecting
- 196 and annotating genetic variations using the HugeSeq pipeline, *Nature Biotechnology*, 30, 3, 226–229,
- 197 doi:10.1038/nbt.2134
- 198 Li, H. (2013), Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM,
- 199 *arXiv:1303.3997 [q-bio]*
- 200 Li, H. and Durbin, R. (2010), Fast and accurate long-read alignment with BurrowsWheeler transform,
- 201 *Bioinformatics*, 26, 5, 589–595, doi:10.1093/bioinformatics/btp698, PMID: 20080505
- 202 Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., et al. (2009), The sequence
- 203 Alignment/Map format and SAMtools, *Bioinformatics (Oxford, England)*, 25, 16, 2078–2079, doi:10.
- 204 1093/bioinformatics/btp352, PMID: 19505943
- 205 novoalign (2013), <http://www.novocraft.com/main/index.php>
- 206 Paila, U., Chapman, B. A., Kirchner, R., and Quinlan, A. R. (2013), GEMINI: integrative exploration
- 207 of genetic variation and genome annotations, *PLoS Comput Biol*, 9, 7, e1003153, doi:10.1371/journal.
- 208 pcbi.1003153
- 209 Picard (2013), <http://picard.sourceforge.net/>
- 210 pysam (2013), <https://code.google.com/p/pysam/>
- 211 Quinlan, A. R. and Hall, I. M. (2010), BEDTools: a flexible suite of utilities for comparing genomic
- 212 features, *Bioinformatics (Oxford, England)*, 26, 6, 841–842, doi:10.1093/bioinformatics/btq033,
- 213 PMID: 20110278
- 214 sambamba (2013), <https://github.com/lomereiter/sambamba>

- 215 Van der Auwera, G. A., Carneiro, M. O., Hartl, C., Poplin, R., del Angel, G., Levy-Moonshine, A., et al.  
216 (2002), From FastQ data to high-confidence variant calls: The genome analysis toolkit best practices  
217 pipeline, in Current Protocols in Bioinformatics (John Wiley & Sons, Inc.)  
218 Zook, J. M., Chapman, B., Wang, J., Mittelman, D., Hofmann, O., Hide, W., et al. (2013), Integrating  
219 sequencing datasets to form highly confident SNP and indel genotype calls for a whole human genome,  
220 *arXiv:1307.4661 [q-bio]*

## FIGURES

- 221 Figures go here after review.