Association for Clinical Genetic Science
Part of the British Society for Genetic Medicine

Guidelines for development and validation of software, with particular focus on bioinformatics pipelines for processing NGS data.

Nicola Whiffin[1,2], Kim Brugger[3], Joo Wook Ahn[4]

[1] NIHR Cardiovascular Biomedical Research Unit, Royal Brompton & Harefield NHS Foundation Trust, London
[2] National Heart and Lung Institute, Imperial College, London
[3] Department of Molecular Genetics, Addenbrooke's Hospital, Cambridge
[4] Genetics Laboratories, Guy's and St Thomas' NHS Foundation Trust, London

Introduction

This document is intended to provide guidance for the development and validation of software used in clinical genetics, focusing on bioinformatics pipelines for next-generation sequencing (NGS) applications. It is not meant to replicate the large amount of literature and best practice guidelines that already exist around software development or validation of clinical genetics diagnostic tests [1–7]. It aims to provide some baseline guidance for bioinformaticians and stakeholders in the NHS. It should be noted that any data analysis pipeline is just one part of a far larger clinical process; it is therefore both dependent on the quality of raw data and limited by the tools that are available.

Definitions

Shall = requirement [8].
Should = recommendation, i.e. there may exist valid reasons in particular circumstances to not follow a recommendation but the full implications must be understood and carefully considered before choosing a different course [8].
Software = inclusive of everything from simple scripts to complex software packages.

Software/pipeline development

The principal of “write code for humans not machines” shall be adhered to [5], including the use of extensive commenting, informative variable names etc. Benefits of developing a "house-style” should be considered and if adopted, this shall be enforced. Annotation within code shall be used for the benefit of other developers and as memory aids for the initial developer, including documenting purpose and rationale. Ongoing support for end users shall be considered, particularly when designing user interfaces, and user guides shall be provided. Contingency planning and other risk assessments shall be performed for all software critical for diagnostic testing. Attempts should be made to publish developed software where it is possible/appropriate to do so (an online repository such as GitHub [9] and/or bioarxiv.org [10] is sufficient).

A system of version control, e.g. Git [11], shall be used for all software and/or scripts used to process diagnostic data. The version control system shall be used to record all changes, and informative commit messages shall be logged to aid traceability and audit. The versions of reference genome, annotation sources and other reference data shall also be recorded. Furthermore, the version control system shall allow recording releases/milestones so that the results of a diagnostic test can be unambiguously linked to the software and data sources that were used to generate them. The version control system shall be designed to work in a multiuser environment to aid teams of developers working together and to assist with transition of workforce.

High-level peer review of code shall be performed before any code is made live. This shall be carried out by a person other than the initial developer (assessed to be competent in this role) and documented. High-level code review shall include reviewing the purpose, the logic used to reach the goal and sanity checks of the output. A more in-depth system that includes raising issues (or change requests), implementing changes on a development branch, testing, performing code review at a line by line level, and then merging changes back to a master branch is preferable and should be implemented. An example of an in-depth system using Github [9] is shown in figure 1. In addition to improving quality, code review shall also be used to facilitate training, assess competency and review staff performance.

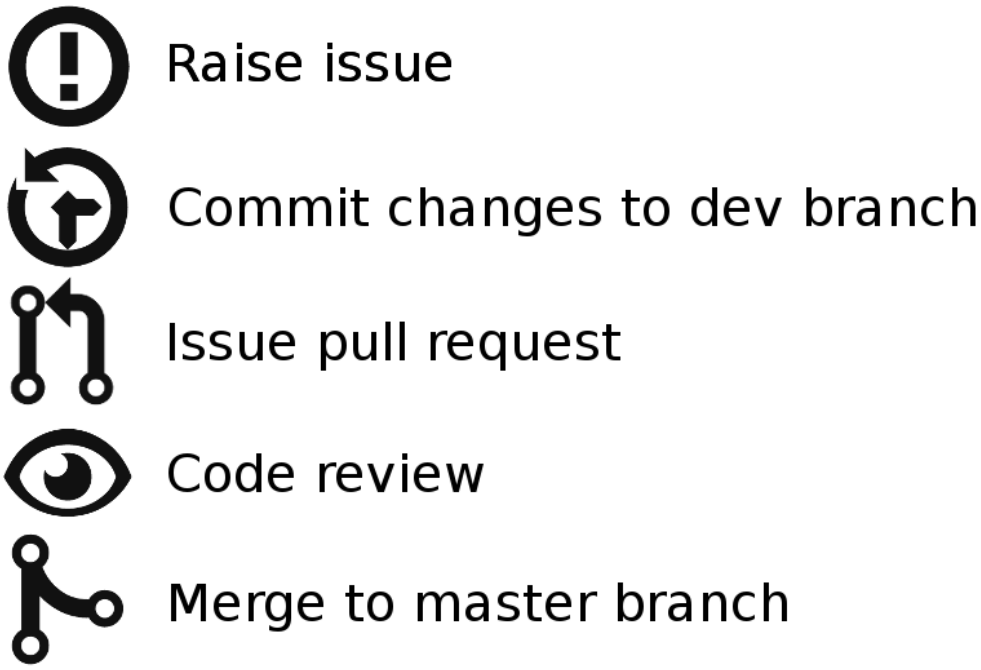


Figure 1: An example workflow for version control and code review using Git [11]. The code base is forked to create a development branch. Change requests are logged by raising issues, which should be granular. Each commit should address a single change and be accompanied by a meaningful commit message. Once ready for code review, the developer should send a pull request. This triggers code review and if meaningful commit messages have been used, this can be performed by the reviewer without assistance from the developer. If the pull request is accepted, the reviewer merges the new code into the master branch.

The use of external software

Diagnostic pipelines often contain a mix of software that has been developed in-house and externally, of which the latter can be further divided into open source, collaborative and commercial. When adopting to use external software, developers shall consider potential issues around documentation, support and updates, preferably using only heavily documented software with an active support network. Developers should avoid external, non-commercial software that has not been published following peer review. A system for regularly checking for software updates and bug fixes shall be adopted and documented. Any new version shall be validated before introduction into clinical service. The versions of any software used for diagnostic purposes shall be recorded and details of key software shall be included on the clinical report. The versions of reference genome, annotation sources and other reference data shall also be recorded.

Where external software is being used with settings different to those recommended or default, these shall be validated through a comparison with the recommended settings, using the validation dataset as described in the following section. Justification for use of the new settings shall be documented.

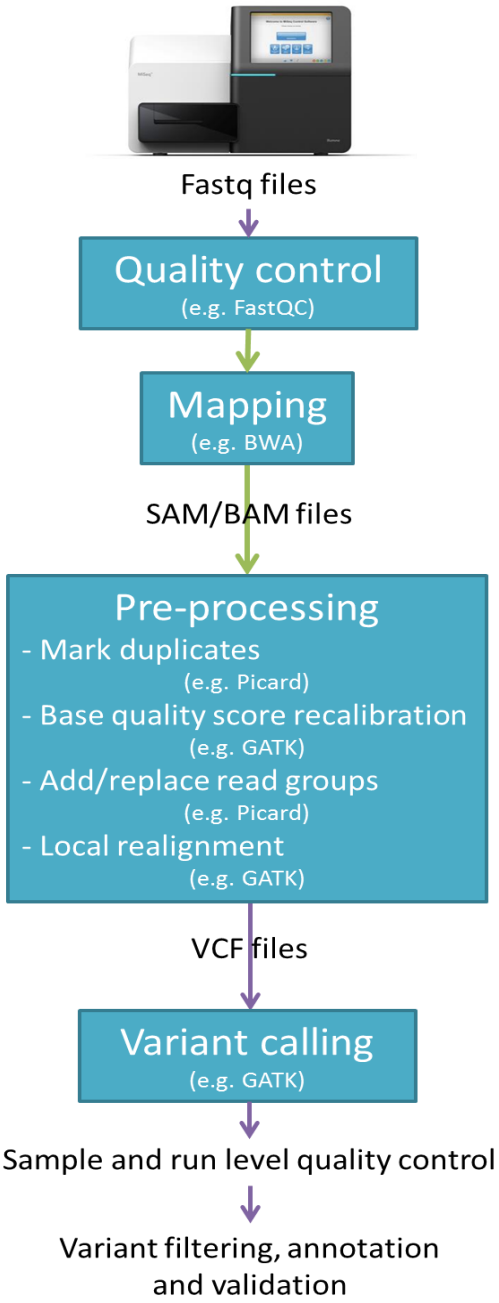
An example of a typical NGS analysis pipeline is shown in Figure 2. Table 1 details examples of commonly used software.

Table 1: Commonly used software for analysis of NGS data

Pipeline step	Software	URL
Quality control	FastQC	http://www.bioinformatics.babraham.ac.uk/projects/fastqc
	Picard Collect*Metrics	http://broadinstitute.github.io/picard
	QualiMap	http://qualimap.bioinfo.cipf.es
Mapping	BWA	http://bio-bwa.sourceforge.net
	Stampy	http://www.well.ox.ac.uk/project-stampy
	Bowtie	http://bowtie-bio.sourceforge.net
Pre-processing	Picard	http://broadinstitute.github.io/picard
	Samtools	http://samtools.sourceforge.net
	GATK[1]	https://www.broadinstitute.org/gatk
Variant calling	GATK Haplotype Caller[1]	https://www.broadinstitute.org/gatk
	GATK Unified Genotyper[1]	https://www.broadinstitute.org/gatk
	Platypus	http://www.well.ox.ac.uk/platypus
	VarScan2	http://varscan.sourceforge.net
	Samtools	http://samtools.sourceforge.net
	Scalpel	http://scalpel.sourceforge.net

[1] Note that at time of writing, versions above 2.3 require licensing for commercial use.

Figure 2: An example outline of a pipeline for analysis of NGS data. Shown are the main steps along with example software for performing each step. For further software examples see Table 1.



Validation of NGS pipelines

Extensive validation processes shall be incorporated into software development for clinical diagnostics. The example of single nucleotide variant (SNV) calling using a pipeline developed in-house for constitutional rare disease diagnostics is used here for demonstration purposes; however, the principles outlined here are also applicable to wider situations.

1. Initial validation of pipeline

This shall be a "dry" validation using truth data to assess the pipeline's output against the truth set. This can be done using the Genome in a Bottle reference data [12]. Users should be aware that there are potential biases in this data towards regions that are easier to sequence with current NGS technologies, and also a potential bias towards current variant callers (e.g. the GATK suite [13]). However, the data is useful as a baseline measure of sensitivity and specificity. A "wet" validation using Genome in a Bottle reference material 8398 [14] should also be considered.

The sensitivity of the pipeline shall be determined using clinical data, i.e. variants detected by Sanger sequencing as part of a diagnostic service which are then also identified with NGS. The resulting sensitivity should have a 95% confidence interval >0.95. This can be achieved if the NGS pipeline detects all 60 of 60 Sanger variants, with no false negatives [3][1]. In addition, detecting 300 of 300 will achieve a 95% confidence interval >0.99. These variants shall be derived from at least 10 individuals.

It should be noted that comparison of called variants against a truth set is a non-trivial task in cases of indels and complex variation, due to alternative representations of variants in vcf files. Tools such as hap.py [15] and vcfeval [16] may be useful for normalisation and comparison of vcf files. There is a benchmarking task team within the Global Alliance for Genomics and Health that is developing further tools and standards [17].

2. Further validation of pipeline

[1] As per the referenced paper and in line with other guidelines, the "rule of three" is used to derive figures of 60 and 300 for 95% confidence intervals >0.95 and >0.99 respectively. However, laboratories are able to calculate confidence intervals for sensitivity (e.g. https://www.medcalc.org/calc/diagnostic_test.php), which is particularly useful when sensitivity is <100%.

Prior to any changes being merged into production code, a round of validation shall be performed as per the initial validation detailed above. Therefore, a validation dataset should be maintained to standardise and simplify this process.

3. Validation following substantive changes to a pipeline

If substantive changes are made to a pipeline, e.g. implementing a new variant caller to improve detection of indels, the existing dataset for validation shall be assessed for relevance and addition of further data shall be considered.

Laboratories may want to consider using data derived from a haploid cell line, e.g. CHM1 [18–20], in some cases.

Summary

The guidelines presented here are intended to be useful to laboratories developing software for analysis of data generated by diagnostic tests, in particular bioinformatics pipelines for NGS data. It is important to note that these guidelines are for validation of software/pipelines and not the full diagnostic test; they should therefore be implemented in the context of other guidelines. The authors are aware that the field of clinical bioinformatics is in its infancy and moving quickly as efforts worldwide strive to advance its clinical utility, ranging from improving the data input all the way through to global data sharing initiatives. Therefore this document should be regularly reviewed and revised following publication.

References

1. Ellard S, Lindsay H, Camm N, Watson C, Abbs S, Wallis Y, Mattocks C, Taylor GR, Charlton R. Practice guidelines for Targeted Next Generation Sequencing Analysis and Interpretation. ACGS Best Practice Guidelines. 2014; Available: http://www.acgs.uk.com/committees/quality-committee/best-practice-guidelines/

2. Matthijs G, Ijntema H, Feenstra I, Souche E. Eurogentest Guidelines for Diagnostic Next Generation Sequencing Final Draft [Internet]. 2015. Available: http://www.eurogentest.org/index.php?id=645

3. Mattocks CJ, Morris MA, Matthijs G, Swinnen E, Corveleyn A, Dequeker E, et al. A standardized framework for the validation and verification of clinical molecular genetic tests. Eur J Hum Genet. 2010;18: 1276–1288.

4. Weiss MM, Van der Zwaag B, Jongbloed JDH, Vogel MJ, Brüggenwirth HT, Lekanne Deprez RH, et al. Best practice guidelines for the use of next-generation sequencing applications in genome diagnostics: a national collaborative study of Dutch genome diagnostic laboratories. Hum Mutat. 2013;34: 1313–1321.

5. Wilson G, Aruliah DA, Brown CT, Chue Hong NP, Davis M, Guy RT, et al. Best practices for scientific computing. PLoS Biol. 2014;12: e1001745.

6. Richards S, Aziz N, Bale S, Bick D, Das S, Gastier-Foster J, et al. Standards and guidelines for the interpretation of sequence variants: a joint consensus recommendation of the American College of Medical Genetics and Genomics and the Association for Molecular Pathology. Genet Med. 2015;17: 405–423.

7. Gargis AS, Kalman L, Bick DP, da Silva C, Dimmock DP, Funke BH, et al. Good laboratory practice for clinical next-generation sequencing informatics pipelines. Nat Biotechnol. 2015;33: 689–693.

8. Bradner S. Key words for use in RFCs to Indicate Requirement Levels. 1997; Available: https://tools.ietf.org/html/rfc2119

9. Build software better, together. In: GitHub [Internet]. [cited 8 Jul 2015]. Available: https://github.com/

10. bioRxiv.org - the preprint server for Biology [Internet]. [cited 8 Jul 2015]. Available: http://biorxiv.org/

11. Git [Internet]. [cited 8 Jul 2015]. Available: https://git-scm.com/

12. GIAB Reference Materials and Data | Advances in Biological and Medical Measurement Science [Internet]. [cited 2 Jun 2015]. Available: https://sites.stanford.edu/abms/content/giab-reference-materials-and-data/

13. GATK. In: GATK [Internet]. [cited 8 Jul 2015]. Available: http://broadinstitute.org/gatk/

14. NIST - SRM Order Request System RM 8398 - Human DNA for Whole-Genome Variant Assessment (Daughter of Utah/European Ancestry) [Internet]. [cited 2 Jun 2015]. Available: https://www-s.nist.gov/srmors/view_detail.cfm?srm=8398

15. Krusche P. hap.py. In: GitHub [Internet]. [cited 1 Apr 2016]. Available: https://github.com/Illumina/hap.py

16. Cleary JG, Braithwaite R, Gaastra K, Hilbush BS, Inglis S, Irvine SA, et al. Comparing Variant Call Files for Performance Benchmarking of Next-Generation Sequencing Variant Calling Pipelines [Internet]. bioRxiv. 2015. p. 023754. doi:10.1101/023754

17. GA4GH. Global Alliance for Genomics and Health Benchmarking Tools. In: GitHub [Internet]. [cited 1 Apr 2016]. Available: https://github.com/ga4gh/benchmarking-tools

18. Resolving the complexity of the human genome using single-molecule sequencing [Internet]. [cited 2 Jun 2015]. Available: http://eichlerlab.gs.washington.edu/publications/chm1-structural-variation/

19. Chaisson MJP, Huddleston J, Dennis MY, Sudmant PH, Malig M, Hormozdiari F, et al. Resolving the complexity of the human genome using single-molecule sequencing. Nature. 2015;517: 608–611.

20. The CHM1-NA12878 benchmark for single-sample SNP/INDEL calling from WGS Illumina data [Internet]. [cited 2 Jun 2015]. Available: http://hackersome.com/p/lh3/hapdip/