Biopet: towards Scalable, Maintainable, User-friendly, Robust and Flexible NGS pipelines

Peter van 't Hof[1], Wibowo Arindrarto[1], Wai Yi Leung[1], Sander van der Zeeuw[1], Sander Bollen[2], Jeroen Laros[3] and Leon Mei[1]

[1]Sequence Analysis Support Core, Leiden University Medical Center, The Netherlands
[2]Dept. Clinical Genetics, Leiden University Medical Center, The Netherlands
[3]Dept. Human Genetics, Leiden University Medical Center, The Netherlands
E-mail: h.mei@lumc.nl

1. Requirements of NGS data analysis pipelines

With the increasing scale of NGS (Next Generation Sequencing) datasets and projects, bioinformatics groups are facing a pressing need to build scalable and robust NGS data analysis production pipelines so that NGS data can be efficiently processed on a Big-Data infrastructure. Furthermore, introducing a service model on top of such pipelines requires clearly specified Standard Operation Procedures (SOP), including controlled file storage, backup policy, versions and parameters to support reproducibility of results. A year ago, we have organised a DTL focus meeting on this topic where the following list of requirements were elicited from the discussion for an ideal pipeline framework, i.e., SMURF principle.

- **Scalable**: framework can be easily deployed on a server or cluster and can utilise all available cores.
- **Maintainable**: framework should have a good support and reliable community and easy to understand code syntax.
- **User-friendly**: framework should be relatively easy to use by a non-expert with a short learning curve.
- **Robust**: framework can rerun part of the pipeline if certain step fails and can explicitly track all scripts and options for logging, report generation, or monitoring through a web page.
- **Flexible**: framework can support a modular implementation (even across different languages) to be DRY (Don't Repeat Yourself) and can provide a transparent control to manage script, file location and change parameters.

2. Current NGS pipeline frameworks

Several open source pipeline frameworks were adapted or developed in the field of bioinformatics, e.g., GNU makefile, Snakemake, Bpipe, MOA, Bcbio-nextgen, Molgenis-compute, GATK-queue, etc, while other on-going works are trying to provide more tailored solutions[1]. It has been demonstrated at several groups that Snakemake is a very good solution for building relatively simple pipelines. With ample experience on using GNU Makefiles but less satisfied with its peculiar syntax, at LUMC we were actively looking for a replacement of our Makefile based pipeline framework. Based on our requirements, Snakemake and Queue were selected for a thoroughly testing at our LUMC local infrastructure for both being robust and flexible. However, we have encountered several incompatibilities with Snakemake pipelines involving more complicate steps on our SGE cluster. On the other hand, GATK-Queue is proven to be more scalable and has a more plausible long term maintainability given its support from the Broad institute. It is worth noting that although GATK package is not a fully free and open source package, GATK-Queue is under a separate MIT license so that there is no restriction on using GATK-Queue to build pipelines even for commercial organisations. This is not a well known fact and hence could explain a not yet very high adoption rate of Queue as a pipeline framework.

3. Development of Biopet

At LUMC, we have developed a GATK-Queue based open source pipeline framework – biopet[2] (Bioinformatics Pipeline Execution Toolkit). We implemented all our commonly used NGS tools as Queue modules in the form of Scala classes. Together with those that are already supported in GATK-Queue like GATK variant-calling and Picard tools, we have a full set of NGS tools at our disposal as Scala classes that are further combined into pipeline functions. Besides meeting the aforementioned requirements, the biopet framework also offers the following advanced features.

- **Support live debugging**: because the use of Scala language, developers can actually run debug session to investigate bugs in an IDE or in a live run.
- **Test framework**: biopet contains both unit tests and pipeline functional tests to allow continuous and automated code quality and infrastructure system dependency monitoring.
- **Easy deployment**: the whole biopet codebase can be compiled and deployed as a single JAR to allow a maximum portability.

4. Summary

We presented our approach to reuse an existing best of breed solution to build scalable, robust, flexible and maintainable NGS pipelines. Our goal is to demonstrate the strength of this solution that can help ourselves and other bioinformatics groups to build more advanced and shareable NGS pipelines.

[1]https://github.com/pjotrp/bioinformatics

[2]https://git.lumc.nl/biopet/biopet