Published online -

TOGGLe, a flexible framework for easily building complex workflows and performing robust large-scale NGS analyses

Christine Tranchant-Dubreuil [1,6,*], Sébastien Ravel [2,6], Cécile Monat [1,6,†], Gautier Sarah [3,6], Abdoulaye Diallo [1,‡], Laura Helou [1], Alexis Dereeper [5,6], Ndomassi Tando [1,6], Julie Orjuela-Bouniol [4], and François Sabot [1,6,*]

[1]DIADE IRD, University of Montpellier, 911 Avenue Agropolis, 34934 Montpellier Cedex 5, France [2]BGPI CIRAD, INRA, TA A-54/K, Campus International de Baillarguet,34398 Montpellier Cedex 5,France [3]AGAP CIRAD, INRA, SupAgro, TA A-108/03, 1000 Avenue Agropolis, 34398 Montpellier Cedex 5, France [4]ADNid-QualTech, 830 Avenue du campus agropolis Baillarguet, 34980 Montferrier-sur-Lez, France [5]IPME IRD, University of Montpellier, 911 Avenue Agropolis, 34934 Montpellier Cedex 5, France [6]South Green Bioinformatics Platform, BIOVERSITY,CIRAD,INRA,IRD,SupAgro, Montpellier, France
[†]Present address: IPK Leibniz Institute of Plant Genetics and Crop Plant Research (IPK) Gatersleben, Seeland, Germany
[‡]Present address: SEQONE, 80 avenue Augustin Fliche, IRMB - CHRU SAINT-ELOI, 34090 Montpellier, France

ABSTRACT

The advent of NGS has intensified the need for robust pipelines to perform high-performance automated analyses. The required softwares depend on the sequencing method used to produce raw data (e.g. Whole genome sequencing, Genotyping By Sequencing, RNASeq) as well as the kind of analyses to carry on (GWAS, population structure, differential expression). Workflows have to be generic and scalable, and should meet the biologists needs.

Here, we present the new version of TOGGLe (Toolbox for Generic NGS Analyses), a simple and highly flexible framework to easily and quickly generate pipelines for large-scale second- and third-generation sequencing analyses, including multi-sample and multi-threading support. TOGGLe is a workflow manager designed to be as effortless as possible to use for biologists, so the focus can remain on the analyses. Pipelines are easily customizable and supported analyses are reproducible and shareable. TOGGLe is designed as a generic, adaptable and fast evolutive solution, and has been tested and used in large-scale projects on various organisms. It is freely available at http://toggle.southgreen.fr/, under the GNU GPLv3/CeCill-C licenses) and can be deployed onto HPC clusters as well as on local machines.

Keywords

Data-intensive analysis, Workflow manager, NGS pipeline, parallel computing, high-performance computing, reproducible research

INTRODUCTION

Advances in Next-Generation Sequencing (NGS) technologies have provided a cost-effective approach to unravel many biological questions, and revolutionized our understanding of Biology. Nowadays, any laboratory can be involved in large-scale sequencing projects, delivering astonishing volumes of sequence data. Although NGS are powerful technologies, they shifted the paradigm from data acquisition to data management, storage and *in fine* biological analyses (1). This intensifies the need for robust and easy-to-use pipelines to perform high-performance automated analyses (2). However, available pipelines depend on the sequencing method used to generate raw data and on the type of analyses to perform (variant calling, GWAS, differential gene expression,...) (3, 4, 5, 6).

To enable processing of large datasets, numerous pipelines frameworks are available, either web-based or command-lines based. The formers, Galaxy (7, 8) or Taverna (9) for instance, can be easily understood by lab biologists as they are available through a graphical web interface. Accessible to non-programmers, they allow adding additional workflows, picking up various pre-packed software, but with limited access to their options. In addition, managing more than a dozen of samples, or having access to different versions of the same software, are not possible without having a high-level access level to the host servers. Thus, they are generally used for small-scale analyses, prototyping or training (10). The second type of frameworks, such as Snakemake (11), Bpipe (12), Ruffus (13) or ClusterFlow (14), targets programmers and expert users. Many of theses tools rely on non-standard, adapted programming languages (e.g. derived either from Python, Bash or Make). Even if bioinformaticians can write

*To whom correspondence should be addressed. Email: christine.tranchant@ird.fr francois.sabot@ird.fr

© 2017 The Author(s)

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/2.0/uk/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

complex pipelines using those frameworks, they require high-level skills in programming and are not suitable for lab biologists.

TOGGLe (Toolbox for Generic NGS Analyses) is an alternate solution mixing advantages from the two kinds of workflow managers, offering a robust and scalable bioinformatic framework for a wide range of NGS applications. Compared to the previous version (version 2 (15)), the current TOGGLe (version 3) is a framework based on command-line, and no more hard-coded pipelines, aimed to both biologists and bioinformaticians. TOGGLe is highly flexible on the data type, working on sequencing raw data (Illumina, PacificBiosciences...), as well as on various other formats (e.g. SAM, BED, VCF) (Figure 1). Carrying out analyses does not require any programming skills, but only basic Linux ones. With TOGGLe, scientists can create robust, customizable, reproducible and complex pipelines, through an user-friendly approach, specifying the software versions as well as parameters.

DESIGN AND IMPLEMENTATION

Input data formats and sample IDs

There is no limit to the number of samples, as soon as all the files are in the same unique directory and of the same format. Input data format can be either FASTA, FASTQ (paired-end, single-end and mate-pair; first-, second- and third-generation), SAM, BAM, BED, GFF or VCF, either plain or compressed (*i.e.* gzip).

Sample IDs and read groups are automatically generated by TOGGLe using the file read name, and no dedicated nomenclature or external sample declaration is needed. For pair-end/mate-pair FASTQ data, no specific name or directory organization is required for pairs to be recognized as such.

Running a TOGGLe pipeline

TOGGLe workflows can be launch from start-to-end with a single command-line, with three mandatory arguments:

1. the input directory containing files to analyze,
2. the output directory that will contain results generated by TOGGLe,
3. the configuration file.

According to the workflow (e.g. a reads mapping step upon a reference), a transcriptome or genome reference sequence, an annotation file or a key file (for demultiplexing) may be also provided.

Configuration file

The configuration file plays a central role in TOGGLe, by enabling users to generate highly flexible and customizable workflows in a simple way. Indeed, this basic text file is composed of different parts allowing to build the workflow, to provide software parameters, to compress or remove intermediate data, and to set up a scheduler if needed (see Figure 2 for more details).

Building Workflow Steps composing the pipeline (e.g. aligning reads upon a reference genome, calling variants, assembling) and their relative order are defined after the *$order* tag. Each line consists of the step number followed by an equal symbol then by the software's name (e.g. 1=FastQC).

If the step number is lower than 1000, the analysis step is carried out for each sample separately, while the step is performed as a global analysis with the results of all the samples for a value higher or equal to 1000 (see Figure 2 and 3).

Providing software parameters and external tools usage The syntax for setting software parameters is identical to that used by each software using the command line. These parameters can be provided after the line composed of the symbol *$* followed by the software name (also called tag line; see Figure 2). If no software parameter is provided, the default ones are used. TOGGLe will handle itself the input and output files as well as the references. Users can also use any software not included in TOGGLe with the *generic* tag followed by the command-line.

Compressing or removing intermediate data As analyses generates a large amount of data, we included the possibility to gzip compress or to remove some intermediate data (*$compress* and *$cleaner* tags).

Setting up jobs scheduler When analyzing on high performance computing (HPC) systems, TOGGLe runs seamlessly with either LSF, MPRUN, SLURM or SGE jobs schedulers. In addition, users can provide specific environment variables to be transferred to the scheduler (such as the paths or modules to be loaded). Finally, node data transfer is automatically managed by TOGGLe when requested by user.

Workflows Management

The core of TOGGLe is the *toggleGenerator.pl* script which (*i*) reads the configuration file, (*ii*) generates pipeline scripts, and then (*iii*) executes them as parallel or global analyses (see Figure 4). Basically, *toggleGenerator.pl* acts as a Make-like tool, compiling blocks of code (themselves allowing the execution of the different tools) to create the requested pipeline. It allows the developers to easily add any new tool without having to modify the main code.

Platforms, Installation and Customization

TOGGLe currently runs on any recent GNU/Linux system (Debian Lenny and more, Ubuntu 12.04 and more, and CentOS 6 and more were tested). TOGGLe was developed to be straightforward to install in several ways : manually (git clone) or through a bash script.

A unique file (*localConfig.pm*) needs to be filled at installation to ensure the integration of the whole software list (path and version; installed separately). However, the whole set of integrated tools is not required to run TOGGLe: one can use it only with SAMtools (16, 17) for instance, and does

not need to install the other tools. More detailed information on the different installation procedures can be found at the TOGGLe website (http://toggle.southgreen.fr)

RESULTS

Analyses and post-analysis tools integrated in TOGGLe

Developed in Perl, TOGGLe incorporates more than 120 functions from 40 state-of-the-art open-source tools (such as BWA (18), Abyss (19) or FastME (20)) and home-made ones (Table 1 and Supplementary Table S1), making it the command-line workflow manager with the highest number of integrated tools by now (Figure 1; Table 2).

Once installed, a large array of NGS tools are ready to use with TOGGLe for various type of analyses: input data QC control, cleaning FASTQ, mapping, post-mapping treatment, SNP calling and filtration, structural variation analyses, assembly (genome and transcriptome). A more detailed list of tools implemented is listed in Table 1 and Supplementary Table S1.

Post-analysis tools are available in the current version of TOGGLe, for population genetics, genomic duplication, phylogeny or transcriptomics (Table 1 and Supplementary Table S1).

A tool targeting both biologists and bioinformaticians

Ease of use TOGGLe drastically simplifies NGS analyses (such as SNP calling, differential expression for RNA, *in silico* assembly) for biologists. Workflows can be easily set up in a few minutes through a unique configuration file, and can be executed through a short command line. In addition, TOGGLe offers access to all parameters without restrictions (or name change) proposed by each software. Finally, users can provide any reference files, without any additional step to add them, at the opposite of ClusterFlow (14) for instance.

As prototyping optimized workflows (software order and parameters) requires a good knowledge of the tools to be used, numerous pre-defined validated configuration files are available on our website (http://toggle.southgreen.fr/) for various type of analyses.

In addition, the effective output files naming convention used by TOGGLe as the well-organized file tree make it easy for the user to identify the different dataset produced (see Supplementary Figure 1).

Ease of development and evolution: Simpler is Clever TOGGLe is designed as a set of separated modules/functions and blocks of code, simplifying code integration and evolution. Each module is written either to run bioinformatic softwares or to ensure functionalities for a specific purpose (such as checking input file formats). The block files are composed of codes implementing a single function at a time. These blocks are then concatenated together following the user pipeline specifications by *toggleGenerator.pl*, to provide a dedicated script pipeline.

This code modularity as well as testing and development processes adopted in TOGGLe prevents the regressions and bugs, facilitating maintenance in a collaborative environment.

Production and development versions, and speed of releasing Two versions of TOGGLe are available, the production one and the development version. The former (http://toggle.southgreen.fr) is validated on a large set of tested data, while the latter (http://github.com/SouthGreenPlatform/TOGGLE-DEV/), more up-to-date, is not completely validated and may provoke errors. Nevertheless, the development version is merged to the production version on a regular basis.

A robust bioinformatics framework

As TOGGLe was developed initially for performing data-intensive bioinformatics analyses, our main aim was to build a robust workflow framework without sacrificing the simplicity of use and the ease of development (Table 2).

Pipeline and data sanity controls Numerous automatic controls are carried out at different levels as transparent actions : validation of the workflow structure defined by the user (checking if the output file format by step n is supported by the step n+1), format control on input data provided by user, checking format of intermediate data.

Missing but requested steps for ensuring the pipeline running (such as indexing reference) are added automatically if omitted.

Reproducibility and traceability TOGGLe ensures that all experimental results are reproducible, documented as well as ready to be retrieved and shared. Indeed, results are organized in a structured tree of directories: all outputs are sorted into separate directories grouped by analyses type (sample or global analyses) and by workflow step (see supplementary figure 1).

All parameters, commands executed as well as software versions are kept in logs and a PDF report. Files such as the pipeline configuration and reference data are duplicated in the output folder, in which are also produced the specific scripts used for the analyses. The original input files, at the opposite to reference and configuration files, are not duplicated to reduce disk usage. Finally, a PDF report for the whole analysis is produced, providing global and visual informations for each sample at each step of the workflow. This report provides a diagram of the workflow, the parameters used (configuration file, softwares version,...) and summary statistics for the key files generated by the pipeline.

Error tracking and reentrancy All errors and warnings encountered are recorded in logs; hence, finding the origin of an error is simplified. If a sample provokes an error, this sample will be ignored for the rest of the analysis, and the failing reported in the error logs. Moreover, as TOGGLe can start from any format or tool, restarting jobs from the failure point is easy and avoid to re-launch the whole analysis.

Large numbers of sample analyzed There is no true limits to the number of samples or the sequencing depth of a project that TOGGLe can take in charge. TOGGLe was already used on hundreds of samples jointly (up to 3,000 by now), from different types of assays (RNAseq, GBS, WGS,...), different analyses (polymorphism detection, read count), and

on different organisms (Table 3). The only observed current limits are the data storage and the computing capacities available.

HPC and parallel execution TOGGLe supports different job scheduling systems and can be efficiently run either on a distributed computing system or on a single server (default behaviour).

Documentation

Installation, quick and complete user manuals, screencasts and a complete developer documentation are available on our website http://toggle.southgreen.fr. In addition, we also provide pre-packed configuration files for different types of classical analyses.

DISCUSSION & PERSPECTIVES

Nowadays, data processing and analyses are the main bottleneck of scientific projects based on the high-throughput sequencing technologies (1). To deal with this issue, various workflow managers were proposed, with different philosophies. GUI tools (such as Galaxy(7, 8)) provide a web interface to easily and quickly design workflows, being developed for users without any programming knowledge. While very useful for prototyping pipelines with a limited data set and for performing small-scale analyses, they are less suitable to carry out large scale analyses or to create highly flexible pipelines (e.g. access to the whole set of software options). In contrast, many of the recent published workflow managers are aimed to bioinformaticians (e.g. Bpipe (12)), based on an extension of a programming language, allowing to design highly customizable and complex workflows through a command-line interface. However, this efficiency comes at the expense of the ease of use.

TOGGLe is an alternative solution taking advantages of these two kinds of tools. Based on a command-line interface, it is suitable for bioinformaticians as well as biologists. Through a unique text file, it offers a simple way to build highly complex and completely customizable workflows for a broad range of NGS applications, from raw initial data to high-quality post-analyses ones. This ease of use allows to greatly reduce hands-on time spent on technical aspects and to focus on the analyses themselves.

Thus, to improve user experience, TOGGLe performs, in a transparent way, different actions requested to ensure the success of the analysis: automatic reference indexing, data format checking, workflow structure validation, and so on. In addition, the unique possibility offered by TOGGLe to carry out analyses at different levels (per samples then global) reduces the difficulty of use, the manipulation time, and the errors linked to file manipulations (Table 2).

TOGGLe was used on numerous sequencing projects with high number of samples, or high depth sequencing or both (Table 3), computing capacities and data storage being the only observed limits. It was shown to be highly adaptable to various biological questions as well as to a large array of computing architecture and data.

TOGGLe has still some features to be enhanced, as a complete reentrancy after a failed run is not yet possible and request the user to modify the data organization before relaunching. We are thus working on a new management version allowing to re-launch directly without any other manipulations. Moreover, the current version of the configuration file is not exportable to another workflow system using CWL (Common Workflow Language), and it would be interesting to also provide in TOGGLe a translator to CWL. Finally, the current grain of parallelization is the sample level (as for ClusterFlow or Bpipe), and a lot of computation time can also be saved at that point. We are exploring the way to use embarrassingly-parallel approaches (splitting samples in numerous subsamples), in order to run optimized pipelines on highly distributed infrastructures.

As NGS analyses and sequencing technologies are continuously evolving, the modularity and the ease of development of our tool is a true advantage to stay up-to-date. For instance, the two-year old previous version of our software, TOGGLe v2 (15), integrated only 50 tools and was hard-coded. By now, more than 120 tools are available with the current TOGGLe v3, the pipeline creation system is much more flexible and robust, and new tools are regularly under integration to widen the scope of analyses possible using TOGGLe (phylogeny,metagenomics, pangenomics).

AVAILABILITY

The source code is freely available at http://toggle.southgreen.fr, under the double license GNU GPLv3/CeCiLL-C.

The TOGGLe website comprises a comprehensive step-by-step tutorial to guide the users to install and run the software. An online issue request is also available under GitHub of the South Green platform(21) (http://github.com/SouthGreenPlatform/TOGGLE/issues/) to report bugs and troubleshooting.

- Project home page: http://toggle.southgreen.fr
- Code repository: https://github.com/SouthGreenPlatform/TOGGLE
- Operating system: GNU/Linux (CentOS, Debian, Ubuntu)
- Programming Language: Perl 5.16 of higher (5.24 recommended)
- License: GNU GPLv3/CeCill-C

FUNDINGS

CM was supported by grants from ANR (AfriCrop project #ANR-13-BSV7-0017) and NUMEV Labex (LandPanToggle #2015-1-030-LARMANDE). All authors are supported by their respective institutes.

ACKNOWLEDGEMENTS

The authors thank users (especially Emira Cherif, Laurence Albar, Remi Tournebize) for testing the first phases of this new version. We also thank Mohamed Kassam for his feedbacks on the TOGGLe usage, and Mathieu Rouard for his useful comments on the manuscript.

REFERENCES

1. Muir, P., Li, S., Lou, S., Wang, D., Spakowicz, D. J., Salichos, L., Zhang, J., Weinstock, G. M., Isaacs, F., Rozowsky, J., and Gerstein, M. (Mar, 2016) The real cost of sequencing: scaling computation to keep pace with data generation. *Genome Biology,* **17**(1), 53.
2. Leipzig, J. (2017) A review of bioinformatic pipeline frameworks. *Briefings in Bioinformatics,* **18**(3), 530–536.
3. Djebali, S., Wucher, V., Foissac, S., Hitte, C., Corre, E., and Derrien, T. Bioinformatics Pipeline for Transcriptome Sequencing Analysis pp. 201–219 Springer New York New York, NY (2017).
4. Wu, X., Kim, T.-K., Baxter, D., Scherler, K., Gordon, A., Fong, O., Etheridge, A., Galas, D. J., and Wang, K. (2017) sRNAnalyzera flexible and customizable small RNA sequencing data analysis pipeline. *Nucleic Acids Research,* p. gkx999.
5. Maarala, A. I., Bzhalava, Z., Dillner, J., Heljanko, K., and Bzhalava, D. (2017) ViraPipe: Scalable Parallel Pipeline for Viral Metagenome Analysis from Next Generation Sequencing Reads. *Bioinformatics,* p. btx702.
6. Blawid, R., Silva, J., and Nagata, T. (2017) Discovering and sequencing new plant viral genomes by next-generation sequencing: description of a practical pipeline. *Annals of Applied Biology,* **170**(3), 301–314.
7. Afgan, E., Baker, D., vandenBeek, M., Blankenberg, D., Bouvier, D., Čech, M., Chilton, J., Clements, D., Coraor, N., Eberhard, C., Grüning, B., Guerler, A., Hillman-Jackson, J., VonKuster, G., Rasche, E., Soranzo, N., Turaga, N., Taylor, J., Nekrutenko, A., and Goecks, J. (jul, 2016) The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update. *Nucleic Acids Research,* **44**(W1), W3–W10.
8. Goecks, J., Nekrutenko, A., Taylor, J., and Galaxy Team, T. (2010) Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biology,* **11**(8), R86.
9. Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, M., Carver, T., Glover, K., Pocock, M. R., Wipat, A., and Li, P. (nov, 2004) Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics,* **20**(17), 3045–3054.
10. Mulder, N. J., Adebiyi, E., Alami, R., Benkahla, A., Brandful, J., Doumbia, S., Everett, D., Fadlelmola, F. M., Gaboun, F., Gaseitsiwe, S., Ghazal, H., Hazelhurst, S., Hide, W., Ibrahimi, A., Jaufeerally Fakim, Y., Jongeneel, C. V., Joubert, F., Kassim, S., Kayondo, J., Kumuthini, J., Lyantagaye, S., Makani, J., Mansour Alzohairy, A., Masiga, D., Moussa, A., Nash, O., Ouwe Missi Oukem-Boyer, O., Owusu-Dabo, E., Panji, S., Patterton, H., Radouani, F., Sadki, K., Seghrouchni, F., Tastan Bishop, Ö., Tiffin, N., and Ulenga, N. (2016 Feb, 2016) H3ABioNet, a sustainable pan-African bioinformatics network for human heredity and health in Africa.. *Genome Res,* **26**(2), 271–7.
11. Köster, J. and Rahmann, S. (2012) Snakemake–a scalable bioinformatics workflow engine.. *Bioinformatics (Oxford, England),* **28**(19), 2520–2.
12. Sadedin, S. P., Pope, B., and Oshlack, A. (jun, 2012) Bpipe: a tool for running and managing bioinformatics pipelines. *Bioinformatics,* **28**(11), 1525–1526.
13. Goodstadt, L. (nov, 2010) Ruffus: a lightweight Python library for computational pipelines. *Bioinformatics,* **26**(21), 2778–2779.
14. Ewels, P., Krueger, F., Käller, M., and Andrews, S. (dec, 2016) Cluster Flow: A user-friendly bioinformatics workflow tool. *F1000Research,* **5**, 2824.
15. Monat, C., Tranchant-Dubreuil, C., Kougbeadjo, A., Farcy, C., Ortega-Abboud, E., Amanzougarene, S., Ravel, S., Agbessi, M., Orjuela-Bouniol, J., Summo, M., and Sabot, F. (2015) TOGGLE: toolbox for generic NGS analyses. *BMC Bioinformatics,* **16**(1), 374.
16. Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., and Durbin, R. (2009) The Sequence Alignment/Map format and SAMtools. *Bioinformatics,* **25**(16), 2078–2079.
17. Li, H. (nov, 2011) A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data.. *Bioinformatics (Oxford, England),* **27**(21), 2987–93.
18. Li, H. and Durbin, R. (jul, 2009) Fast and accurate short read alignment with Burrows-Wheeler transform.. *Bioinformatics (Oxford, England),* **25**(14), 1754–60.
19. Robertson, G., Schein, J., Chiu, R., Corbett, R., Field, M., Jackman, S. D., Mungall, K., Lee, S., Okada, H. M., Qian, J. Q., Griffith, M., Raymond, A., Thiessen, N., Cezard, T., Butterfield, Y. S., Newsome, R., Chan, S. K., She, R., Varhol, R., Kamoh, B., Prabhu, A.-L., Tam, A., Zhao, Y., Moore, R. A., Hirst, M., Marra, M. A., Jones, S. J. M., Hoodless, P. A., and Birol, I. (nov, 2010) De novo assembly and analysis of RNA-seq data. *Nature Methods,* **7**(11), 909–912.
20. Lefort, V., Desper, R., and Gascuel, O. (2015) FastME 2.0: A Comprehensive, Accurate, and Fast Distance-Based Phylogeny Inference Program. *Molecular Biology and Evolution,* **32**(10), 2798–2800.
21. SouthGreen (2016 Nov, 2016) The South Green portal: a comprehensive resource for tropical and Mediterranean crop genomics. *Current Plant Biology,* **7-8**, 6–9.
22. Andrews, S. (2010) FastQC: a quality control tool for high throughput sequence data. *http://www.bioinformatics.babraham.ac.uk/projects/fastqc,*.
23. Lab, H. (2010) FASTX Toolkit. *http://hannonlab.cshl.edu/fastx_toolkit/index.html,*.
24. Martin, M. (may, 2011) Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet.journal,* **17**(1), 10.
25. Didion, J. P., Martin, M., and Collins, F. S. (August, 2017) Atropos: specific, sensitive, and speedy trimming of sequencing reads. *PeerJ,* **5**, e3720.
26. Catchen, J., Hohenlohe, P. A., Bassham, S., Amores, A., and Cresko, W. A. (jun, 2013) Stacks: an analysis tool set for population genomics. *Molecular Ecology,* **22**(11), 3124–3140.
27. Li, H. and Durbin, R. (mar, 2010) Fast and accurate long-read alignment with Burrows-Wheeler transform.. *Bioinformatics (Oxford, England),* **26**(5), 589–95.
28. Kim, D., Pertea, G., Trapnell, C., Pimentel, H., Kelley, R., Salzberg, S. L., Mortazavi, A., Williams, B., McCue, K., Schaeffer, L., Wold, B., Pei, B., Sisu, C., Frankish, A., Howald, C., Habegger, L., Mu, X., Harte, R., Balasubramanian, S., Tanzer, A., Diekhans, M., Reymond, A., Hubbard, T., Harrow, J., Gerstein, M., Roberts, A., Trapnell, C., Donaghey, J., Rinn, J., Pachter, L., Trapnell, C., Pachter, L., Salzberg, S., Wu, T., Nacu, S., Grant, G., Farkas, M., Pizarro, A., Lahens, N., Schug, J., Brunk, B., Stoeckert, C., Hogenesch, J., Pierce, E., Dobin, A., Davis, C., Schlesinger, F., Drenkow, J., Zaleski, C., Jha, S., Batut, P., Chaisson, M., Gingeras, T., Wang, K., Singh, D., Zeng, Z., Coleman, S., Huang, Y., Savich, G., He, X., Mieczkowski, P., Grimm, S., Perou, C., MacLeod, J., Chiang, D., Prins, J., Liu, J., Zhang, Z., Harrison, P., Liu, Y., Gerstein, M., Kalyana-Sundaram, S., Kumar-Sinha, C., Shankar, S., Robinson, D., Wu, Y., Cao, X., Asangani, I., Kothari, V., Prensner, J., Lonigro, R., Iyer, M., Barrette, T., Shanmugam, A., Dhanasekaran, S., Palanisamy, N., Chinnaiyan, A., Chen, R., Mias, G., Li-Pook-Than, J., Jiang, L., Lam, H., Miriami, E., Karczewski, K., Hariharan, M., Dewey, F., Cheng, Y., Clark, M., Im, H., Habegger, L., Balasubramanian, S., O, M., Xing, J., Zhang, Y., Han, K., Salem, A., Sen, S., Huff, C., Zhou, Q., Kirkness, E., Levy, S., Batzer, M., Jorde, L., Levy, S., Sutton, G., Ng, P., Feuk, L., Halpern, A., Walenz, B., Axelrod, N., Huang, J., Kirkness, E., Denisov, G., Lin, Y., MacDonald, J., Pang, A., Shago, M., Stockwell, T., Tsiamouri, A., Bafna, V., Bansal, V., Kravitz, S., Busam, D., Beeson, K., McIntosh, T., Remington, K., Abril, J., Gill, J., Borman, J., Rogers, Y., Frazier, M., Scherer, S., Strausberg, R., Langmead, B., Salzberg, S., Kim, D., Salzberg, S., Langmead, B., Trapnell, C., Pop, M., Salzberg, S., Griebel, T., Zacher, B., Ribeca, P., Raineri, E., Lacroix, V., Guigo, R., and Sammeth, M. (2013) TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome Biology,* **14**(4), R36.
29. Langmead, B., Trapnell, C., Pop, M., and Salzberg, S. L. (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology,* **10**(3), R25.
30. Langmead, B. and Salzberg, S. L. (mar, 2012) Fast gapped-read alignment with Bowtie 2. *Nature Methods,* **9**(4), 357–359.
31. Philippe, N., Salson, M., Commes, T., and Rivals, E. (mar, 2013) CRAC: an integrated approach to the analysis of RNA-seq reads. *Genome Biology,* **14**(3), R30.
32. Grabherr, M. G., Haas, B. J., Yassour, M., Levin, J. Z., Thompson, D. A., Amit, I., Adiconis, X., Fan, L., Raychowdhury, R., Zeng, Q., Chen, Z., Mauceli, E., Hacohen, N., Gnirke, A., Rhind, N., di Palma, F., Birren, B. W., Nusbaum, C., Lindblad-Toh, K., Friedman, N., and Regev, A. (jul, 2011) Full-length transcriptome assembly from RNA-Seq data without a reference genome.. *Nature biotechnology,* **29**(7), 644–52.
33. Pertea, G., Huang, X., Liang, F., Antonescu, V., Sultana, R., Karamycheva, S., Lee, Y., White, J., Cheung, F., Parvizi, B., Tsai, J., and Quackenbush, J. (mar, 2003) TIGR Gene Indices clustering tools (TGICL): a software system for fast clustering of large EST datasets..

Bioinformatics (Oxford, England), **19**(5), 651–2.
34. Wysoker, A., Tibbetts, K., and Fennell, T. (2013) Picard tools, http://picard.sourceforge.net. ,.
35. Van der Auwera, G. A., Carneiro, M. O., Hartl, C., Poplin, R., Del Angel, G., Levy-Moonshine, A., Jordan, T., Shakir, K., Roazen, D., Thibault, J., Banks, E., Garimella, K. V., Altshuler, D., Gabriel, S., and DePristo, M. A. (2013) From FastQ data to high confidence variant calls: the Genome Analysis Toolkit best practices pipeline.. *Current protocols in bioinformatics / editoral board, Andreas D. Baxevanis ... [et al.],* **43**, 11.10.1–33.
36. DePristo, M. A., Banks, E., Poplin, R., Garimella, K. V., Maguire, J. R., Hartl, C., Philippakis, A. A., del Angel, G., Rivas, M. A., Hanna, M., McKenna, A., Fennell, T. J., Kernytsky, A. M., Sivachenko, A. Y., Cibulskis, K., Gabriel, S. B., Altshuler, D., and Daly, M. J. (may, 2011) A framework for variation discovery and genotyping using next-generation DNA sequencing data.. *Nature genetics,* **43**(5), 491–8.
37. McKenna, A., Hanna, M., Banks, E., Sivachenko, A., Cibulskis, K., Kernytsky, A., Garimella, K., Altshuler, D., Gabriel, S., Daly, M., and DePristo, M. A. (sep, 2010) The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Research,* **20**(9), 1297–1303.
38. Breese, M. R. and Liu, Y. (feb, 2013) NGSUtils: a software suite for analyzing and manipulating next-generation sequencing datasets. *Bioinformatics,* **29**(4), 494–496.
39. Anders, S., Pyl, P. T., and Huber, W. (jan, 2015) HTSeq–a Python framework to work with high-throughput sequencing data.. *Bioinformatics (Oxford, England),* **31**(2), 166–9.
40. Trapnell, C., Williams, B. A., Pertea, G., Mortazavi, A., Kwan, G., van Baren, M. J., Salzberg, S. L., Wold, B. J., and Pachter, L. (may, 2010) Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature Biotechnology,* **28**(5), 511–515.
41. Quinlan, A. R. and Hall, I. M. (mar, 2010) BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics,* **26**(6), 841–842.
42. Danecek, P., Auton, A., Abecasis, G., Albers, C. A., Banks, E., DePristo, M. A., Handsaker, R. E., Lunter, G., Marth, G. T., Sherry, S. T., McVean, G., and Durbin, R. (aug, 2011) The variant call format and VCFtools. *Bioinformatics,* **27**(15), 2156–2158.
43. Cingolani, P., Platts, A., Coon, M., Nguyen, T., Wang, L., Land, S., Lu, X., and Ruden, D. (2012) A program for annotating and predicting the effects of single nucleotide polymorphisms, SnpEff: SNPs in the genome of Drosophila melanogaster strain w1118; iso-2; iso-3. *Fly,* **6**(2), 80–92.
44. Chen, K., Wallis, J. W., McLellan, M. D., Larson, D. E., Kalicki, J. M., Pohl, C. S., McGrath, S. D., Wendl, M. C., Zhang, Q., Locke, D. P., Shi, X., Fulton, R. S., Ley, T. J., Wilson, R. K., Ding, L., and Mardis, E. R. (sep, 2009) BreakDancer: an algorithm for high-resolution mapping of genomic structural variation. *Nature Methods,* **6**(9), 677–681.
45. Ye, K., Schulz, M. H., Long, Q., Apweiler, R., and Ning, Z. (nov, 2009) Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads.. *Bioinformatics (Oxford, England),* **25**(21), 2865–71.
46. Djedatin, G., Monat, C., Engelen, S., and Sabot, F. (2017) DuplicationDetector, a light weight tool for duplication detection using NGS data. *Current Plant Biology,* **9-10**(Supplement C), 23 – 28 Special issue on Plant Development.
47. Frichot, E., Mathieu, F., Trouillon, T., Bouchard, G., and François, O. (2014) Fast and Efficient Estimation of Individual Ancestry Coefficients. *Genetics,* **196**(4), 973–983.
48. Chang, C. C., Chow, C. C., Tellier, L. C., Vattikuti, S., Purcell, S. M., and Lee, J. J. (2015) Second-generation PLINK: rising to the challenge of larger and richer datasets. *GigaScience,* **4**(1), 1–16.
49. Gilbert, D. Readseq: Read and reformat biosequences. *http://iubio.bio.indiana.edu/soft/molbio/readseq/java/,*.
50. Pinel-Galzi, A., Dubreuil-Tranchant, C., Hbrard, E., Mariac, C., Ghesquire, A., and Albar, L. (2016) Mutations in Rice yellow mottle virus Polyprotein P2a Involved in RYMV2 Gene Resistance Breakdown. *Frontiers in Plant Science,* **7**, 1779.

ADDITIONAL FILES

1 – List of tools currently integrated (Table S1)

2 – Example of results file tree created by TOGGLe (Figure S1)

FIGURES AND TABLES

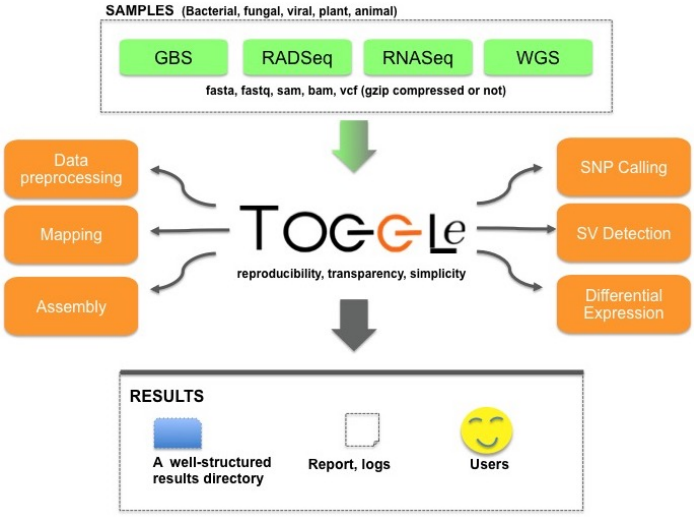

Figure 1. Overview of TOGGLe, a flexible framework for performing large-scale NGS analyses.

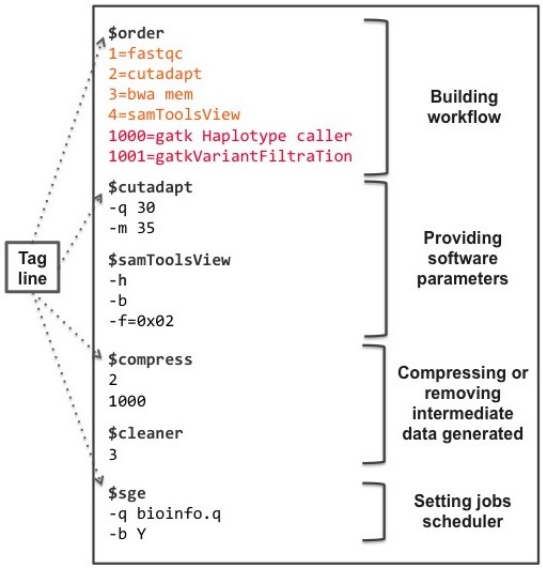

Figure 2. Configuration file to detect polymorphisms, composed of 4 parts : (1) "Building workflow" section : the first 4 analysis (steps lower than 1000, in orange) will be carried out separately for each sample (parallel analyses, in pink) whereas the last two steps will be performed as a global common analysis; (2) "Software parameters"; (3) "Data management"; and (4) "Scheduler management" section

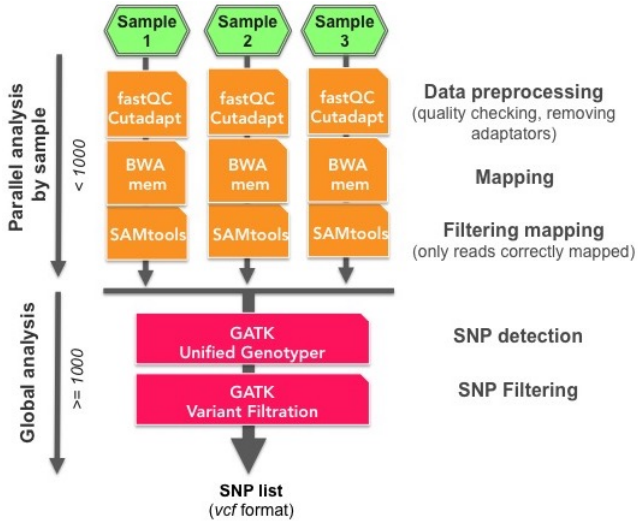

Figure 3. Overview of a TOGGLe pipeline for SNP detection from 3 samples. This figure shows the different steps performed either as parallel (in orange) or as global analysis (in pink). See the corresponding configuration file in Figure 2.

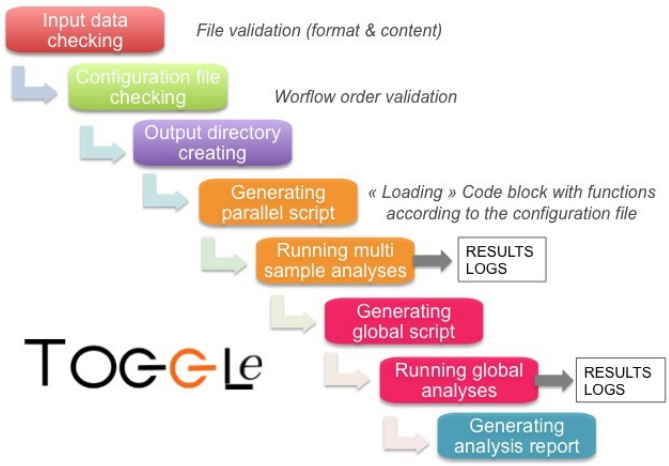

Figure 4. Schematic diagram of the different actions performed by *toggleGenerator.pl*.

Type of analyses	Software names
Data Preprocessing	FastQC (22), FASTX-Toolkit (23), Cutadapt (24), Atropos (25), checkFormat tools *
Demultiplexing	Stacks (26)
Mapping	BWA (18, 27), TopHat2 (28), Bowtie (29), Bowtie2 (30), CRAC (31)
Assembly	Trinity (32), TgiCl (33), Abyss (19)
SAM/BAM Management and analysis	Picard-Tools (34), SAMtools (16, 17), GATK (35, 36, 37), NGSutils (38)
ReadCount tools	HTSeq (39), cufflinks (40)
SNP calling and VCF management	GATK (35, 36, 37), BEDtools (41), vcftools (42), SnpEff (43)
Structural variant detection	BreakDancer (44), Pindel (45), DuplicationDetector (46)
Post-Calling data analysis	BEDtools (41), sNMF (47), FastME (20), plink (48), readseq (49)
Generic command line	Any bash command *

Table 1. Bioinformatic softwares integrated in TOGGLe. *: home-made tools for data format control and generic command-line.

Features	TOGGLe v3	TOGGLe v2 (15)	Bpipe (12)	Clusterflow (14)	SnakeMake (11)
Biologist use	+++	++	-	+	-
Ease of pipeline design for biologists	+++	-	+	++	-
Transparent supplementary actions	+++	-	-	-	+
Pipeline configuration language	Text	Text	Bash-like	Tabulated format	Python
Integrated tools	120	50	None*	31	None*
Sanity check	+++	++	+	+	-
Software version control	+++	+	+	-	+
Reproducibility	+++	++	++	++	++
Re-entrancy	+	-	++	+	+++
Schedulers integration	+++	+	+++	+++	++
Parallel-then-Global analyses	+++	++	-	-	+
Easy to install	+++	++	+	++	++
Documentation	+++	+	++	++	+++
Release frequency	+++	-	++	+	+++

Table 2. Comparison with some available command-line workflows managers. Release frequency was estimated from previous release date for each tool.
*: no tools are already included in Bpipe and SnakeMake, users have to include them through coding by themselves.

NGS type	Organism	Type	Analysis type	Description
GBS	Date Palm and *Phoenix sp.*	Plant	Polymorphism detection	242 samples
RNASeq	*Arabidopsis thaliana*	Plant	Differential expression	36 samples, 14 millions reads per sample
	Pacaya	Plant	Transcriptome Assembly	20 samples, 34 millions reads per sample
			Polymorphism detection	
			Differential expression	
	Magnaporthe oryzae	Fungus	Differential expression	6 samples, 90 millions reads
	Burkholderia gladioli	Bacteria	Differential expression	6 samples, 90 millions reads
WGS	*Coffea canephora*	Plant	Polymorphism detection	12 samples, 55x mean
	African Rice	Plant	Duplication detection	16 samples, 35x mean (46)
	African and Asian Rices	Plant	Polymorphism detection	350 samples, 35x mean
	Rice Yellow Mottle Virus	Virus	Polymorphism detection	50 samples, 8,000x mean (50)
	Leichmanii	Animal	Polymorphism detection	15 samples, 20x mean
	Magnaporthe oryzae	Fungus	Polymorphism detection	52 samples, 20x mean
	Micosphaerella fijensis	Fungus	Polymorphism detection	160 samples, 50x mean
CaptureSeq	Rice	Plant	Polymorphism detection	200 targets (genes), 100 samples
	Palm	Plant	Polymorphism detection	150 targets (800 exons), 120 samples
	Prunus persica	Plant	Polymorphism detection	540 targets (exons), 16 samples
	Human	Animal	Polymorphism detection	241 targets (8800 exons), 48 samples

Table 3. List of selected projects using TOGGLe.