SOFTWARE ARTICLE

CulebrONT: a streamlined long reads multi-assembler pipeline for prokaryotic and eukaryotic genomes

Julie Orjuela[1,2,3*†], Aurore Comte[2,3†], Sébastien Ravel[2,3], Florian Charriat[2,3], Tram Vi[2,4], François Sabot[1,3] and Sébastien Cunnac[2]

[*]Correspondence: julie.orjuela@ird.fr
[1]DIADE Unit, University of Montpellier, CIRAD, IRD, 911 Avenue Agropolis, F-34394 , Montpellier Cedex 5, France
[2]PHIM Plant Health Institute, Univ Montpellier, CIRAD, INRAE, Institut Agro, IRD, Montpellier, France
[3]IFB - South Green Bioinformatics Platform, Bioversity, CIRAD, INRAE, IRD, F-34398 Montpellier, France
Full list of author information is available at the end of the article
[†]Equal contributor

Abstract

Background: Using long reads provide higher contiguity and better genome assemblies. However, producing such high quality sequences from raw reads requires to chain a growing set of tools, and determining the best workflow is a complex task.

Results: To tackle this challenge, we developed CulebrONT, an open-source, scalable, modular and traceable Snakemake pipeline for assembling long reads data. CulebrONT enables to test on multiple samples multiple long reads assemblers in parallel, and can optionally perform, downstream assembly, circularization and polishing. It further provides a range of assembly quality metrics summarized in a final user-friendly report.

Conclusions: CulebrONT leverages the difficulties of assembly pipelines development, and allow even basic users to obtain high-quality assemblies.

Keywords: long reads; assembly; QC; reproducibility; snakemake

Background

The third-generation sequencing technologies, namely Oxford Nanopore Technologies (ONT) and Pacific Biosciences (PB), provide reads up to hundreds of thousands of bases in length. They can extend over repeats or structural variants, and thus result in higher contiguity and accuracy of genome assembly. Due to its low price, speed, portability and easy-access system, ONT is increasingly used worldwide, generally by laboratories sequencing their favorite organism with limited expertise in assembly methods. Nonetheless, assembly is not a quite pace: eukaryotic genomes assembly is a highly complex task (large genome sizes, high rates of repeated sequences, high heterozygosity levels and even polyploidy), and while prokaryotic genomes may appear less challenging, specific features such as circular DNA molecules, must be taken into consideration to achieve high quality assembly.

Numerous long-read assembly and post-assembly tools are available, relying on a large variety of approaches and algorithms, and many of them are frequently updated. However, even with this plethora of tools, there is no single silver bullet to genome assembly for all taxonomic groups. The systematic assessment of these assembly tools is necessary to properly exploit the data [1] but performing benchmarks to find the best combination of tools for a given dataset is highly complex, and require accessible, scalable, traceable and reproducible data processing

workflows. A few solutions such as Katuali [2] or CCBGpipe [3], were previously developed to tackle these issues, but they are dedicated to either eukaryotic or prokaryotic genomes, or provide a restricted choice of assemblers. Some of these are also difficult to parallelize in a HPC environment, or to update with the latest versions of software components. In addition, for the subsequent evaluation of the assemblies, it is critical to organize and aggregate the numerous Quality Control (QC) metrics generated by various tools, in order to facilitate comparisons. To solve these issues, we developed CulebrONT, a pipeline allowing users to easily assemble, circularize (if needed) and polish assemblies on multiple dataset with multiple alternative tools, while reporting various QC metrics for each assembly.

Implementation

CulebrONT uses the Snakemake [4] functionalities, enabling readability of the code, local and HPC scalability, reentry, reproducibility and modularity. CulebrONT assembles, circularises, polishes and corrects genome sequences from raw read sequences in fastq format, and provides QC metrics. To provide more flexibility to the user, individual tools are optional and include the most popular ones. Thus, CulebrONT can be used either to prototype and explore assemblies on new organisms or in assembly production. While originally developed for working on ONT data, CulebrONT can be used on PB data also (except nanopore-only tools).

Six recent community-validated assemblers are currently included (Canu [5], [6], Raven [7], MiniAsm [8] (coupled with minipolish [9] for an initial polishing), Shasta [10], smartDeNovo [11]), as well as several tools for polishing (Racon [12] and Pilon [13] for ONT and PB data, Medaka [14] and Nanopolish [15] for ONT data only). If requested, Circlator [16] or a dedicated internal tool can automatically circularise the primary output of assemblers that cannot handle circular molecules (see Supplementary information part 2.2 for more details). Each processing step in CulebrONT, from raw to final assembly, can be optionally assessed by various quality control tools (Busco [17], QUAST [18], BlobTools [19], KAT [20], Assemblytics [21], Samtools Flagstats [22] and Mauve [23]).

CulebrONT builds a workflow using the user setup file *config.yaml*. Data can come from a single sample as well as from multiple ones (one fastq file per sample in the data folder). Tools paths are imported from *tools_path.yaml*, and cluster resources are managed using the *cluster_config.yaml* file (Figure 1.1). The dedicated CulebrONT class checks configuration files, controls if data and software environments exist and ensures the global coherence of requested steps. This python class imports snakefiles to build a specific instance of the workflow (Figure 1.2). Upon execution (on HPC or a single machine, Figure 1.3), in addition to the expected output, individual rules will generate log files formatted in HTML (Figure 1.4). Recently, snakemake profiles have been also added for more precise HPC job management. CulebrONT relies on conda [24] and singularity [25] to optimize and simplify software installation, to secure environments and to ensure reproducibility.

Results

CulebrONT was developed in order to facilitate comparisons between primary assemblies, but also to check the effect of polishing on sequence accuracy. CulebrONT compiles information from these steps, but also the useful quality information calculated at each activated step. A output directory is generated, with a specific architecture (see supplemental file to details). For each sample, the output directory contains a sub-directory corresponding to each activated step on the *config.yaml* file (assembly, circularisation, polishing, correction, fixstart, and QC, for instance). In addition, a log directory contains execution information for all steps. CulebrONT generates an useful HTML report, found in the *FINAL REPORT* folder. In this report, summarized statistics and relevant information can be found, such as the configuration parameters used, the tools versions and the computational time for each steps and so on (Figure 2). These information can be useful for users that want to benchmark and identify the best pipeline for their data to be used.

In order to give an idea of how CulebrONT can help on benchmarking assembly pipelines, four ONT dataset on three different species were tested : two strains of *Xanthomonas sp.* (reads N50 24,037 bp and 20,653 bb; median 4,152 bp and 5,696 bp; 610 Mb and 1.8 Gb for VNO70 and VNO15, respectly), *Meloidogyne graminicola* nematoda (reads N50 9,372 pb; median 2,853 bp; 3.3 Gb) and a *Oryza sativa* dataset (N50 22,328 bp, median 13 544 bp, total 14,6 Gb).

On *Xanthomonas sp.* all available assemblers were tested on both samples VNO70 and VNO15. Good results were obtained by Canu and Miniasm, but the best were found for VN70 using Flye + 2 Racon rounds + Medaka correction with only 1 circularised contig (N50 4,917,887 pb and L50 1) obtained and a busco score of 94.2%. Busco results obtained on this sample can be found on the figure 2.A For VN15, Canu was the most performing one with 2 circularised contigs (N50 4,985,087 pb and L50 1) and with 99.2% of conserved orthologous genes found by Busco. More details can be found on the table 1).

Five assemblers were tested for *M. graminicola* (Flye, Shasta, Miniasm, Smartdenovo and Raven). At the best, 617 contigs were obtained on VN18 sample using Flye + 2 Racon rounds + Medaka, with a N50 of 402,454 pb and a L50 of 29 (see table 1).

On *Oryza sativa*, due to the larger genome length, only three assemblers were activated (Flye, Shasta and Raven), and only polishing with racon was performed. For DJ123 rice sample, Flye + two Racon rounds was the best solution, with a N50 of 14,895,042 pb and a L50 of 10 (for 12 chromosomes). An overview of a CulebrONT report is shown on the figure 2.B using rice as an example. More details can be found on table 1 and more reports generated by CulebrONT are available at: https://itrop.ird.fr/culebront_utilities/BMC_RESULTS.

Applications

CulebrONT has been also successfully used on many and various organisms, including bacteria (*Xanthomonas sp*), haploid fungi (*Pseudocercospora fijiensis*), green

algae (*Bathycoccus sp*), insects (*Drosophila sp* [26]), diploid grasses (*Paspalum sp* (O. Blanc, pers com) and other *Oryza sp*), as well as on allopolyploid plants (ABB triploid banana; M. Rouard, pers com).

Conclusion

In summary, CulebrONT simplifies the analysis of large-scale assemblies by allowing fast and reproducible processing of a single dataset or on a collection of samples simultaneously. The output facilitates the comparative evaluation of the assembly workflows while keeping a traceable record of the analyses.

Availability and requirements

Project name: CulebrONT pipeline
Project home page: https://github.com/SouthGreenPlatform/CulebrONT_pipeline
Documentation: https://culebront-pipeline.readthedocs.io/en/latest/
Operating system: Platform independent
Programming language: Python >=3.6
Other requirements: Snakemake >=5.10
License: GPLv3
Any restrictions to use by non-academics: GPLv3

List of abbreviations

N50 : the minimum contig length required to cover 50 percent of the assembled genome sequence.
L50 : The number of sequences evaluated at the point when the sum length exceeds 50% of the assembly size.

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Author's contributions
JO, SC and FS conceived the original idea. JO, AC, SR, FC, TV and SC written code source. Documentation was performed by JO, SR, FS and AC. JO and FS analysed data. JO, FS and SC drafted the manuscript. All authors read and approved the final manuscript.

Acknowledgements
Data processing was performed at the High-Performance Computing Cluster i-Trop of IRD. Authors thank N.Tando for administration support and the South Green Plateform.

Funding
This work was supported by PHIM and DIADE research units. AC was supported by a IRD engineer contract. TV was supported by PHIM funding and the CRP-Rice program. FC was supported by the MagMAX ANR-18-CE20-0016 grant awarded to Pierre Gladieux.

Author details
[1]DIADE Unit, University of Montpellier, CIRAD, IRD, 911 Avenue Agropolis, F-34394 , Montpellier Cedex 5, France. [2]PHIM Plant Health Institute, Univ Montpellier, CIRAD, INRAE, Institut Agro, IRD, Montpellier, France. [3]IFB - South Green Bioinformatics Platform, Bioversity, CIRAD, INRAE, IRD, F-34398 Montpellier, France. [4]Agricultural Genetics Institute, Vietnam Academy of Agricultural Sciences, Hanoi, Vietnam.

References

1. Murigneux, V., Rai, S.K., Furtado, A., Bruxner, T.J.C., Tian, W., Harliwong, I., Wei, H., Yang, B., Ye, Q., Anderson, E., Mao, Q., Drmanac, R., Wang, O., Peters, B.A., Xu, M., Wu, P., Topp, B., Coin, L.J.M., Henry, R.J.: Comparison of long-read methods for sequencing and assembly of a plant genome. GigaScience **9**(12) (2020). doi:10.1093/gigascience/giaa146. giaa146. https://academic.oup.com/gigascience/article-pdf/9/12/giaa146/35044370/giaa146.pdf
2. Katuali: https://nanoporetech.github.io/katuali/index.html (2020). https://nanoporetech.github.io/katuali/index.html
3. Liao, Y.-C., Cheng, H.-W., Wu, H.-C., Kuo, S.-C., Lauderdale, T.-L.Y., Chen, F.-J.: Completing circular bacterial genomes with assembly complexity by using a sampling strategy from a single minion run with barcoding. Frontiers in microbiology **10**, 2068–2068 (2019). doi:10.3389/fmicb.2019.02068. 31551994[pmid]
4. Köster, J., Rahmann, S.: Snakemake—a scalable bioinformatics workflow engine. Bioinformatics **28**(19), 2520–2522 (2012). doi:10.1093/bioinformatics/bts480. https://academic.oup.com/bioinformatics/article-pdf/28/19/2520/819790/bts480.pdf
5. Koren, S., Walenz, B.P., Berlin, K., Miller, J.R., Bergman, N.H., Phillippy, A.M.: Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. Genome Research **27**(5), 722–736 (2017). doi:10.1101/gr.215087.116. http://genome.cshlp.org/content/27/5/722.full.pdf+html
6. Kolmogorov, M., Yuan, J., Lin, Y., Pevzner, P.A.: Assembly of long, error-prone reads using repeat graphs. Nature Biotechnology **37**(5), 540–546 (2019). doi:10.1038/s41587-019-0072-8
7. Vaser, R., Šikić, M.: Raven: a de novo genome assembler for long reads. bioRxiv (2020). doi:10.1101/2020.08.07.242461. https://www.biorxiv.org/content/early/2020/08/10/2020.08.07.242461.full.pdf
8. Li, H.: Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences. Bioinformatics **32**(14), 2103–2110 (2016). doi:10.1093/bioinformatics/btw152. https://academic.oup.com/bioinformatics/article-pdf/32/14/2103/19567911/btw152.pdf
9. Wick, R., Holt, K.: Benchmarking of long-read assemblers for prokaryote whole genome sequencing [version 3; peer review: 4 approved]. F1000Research **8**(2138) (2020). doi:10.12688/f1000research.21782.3
10. Shafin, K., Pesout, T., Lorig-Roach, R., Haukness, M., Olsen, H.E., Bosworth, C., Armstrong, J., Tigyi, K., Maurer, N., Koren, S., Sedlazeck, F.J., Marschall, T., Mayes, S., Costa, V., Zook, J.M., Liu, K.J., Kilburn, D., Sorensen, M., Munson, K.M., Vollger, M.R., Monlong, J., Garrison, E., Eichler, E.E., Salama, S., Haussler, D., Green, R.E., Akeson, M., Phillippy, A., Miga, K.H., Carnevali, P., Jain, M., Paten, B.: Nanopore sequencing and the shasta toolkit enable efficient de novo assembly of eleven human genomes. Nature Biotechnology **38**(9), 1044–1053 (2020). doi:10.1038/s41587-020-0503-6
11. Liu, H., Wu, S., Li, A., Ruan, J.: Smartdenovo: A de novo assembler using long noisy reads. https://www.preprints.org/ (2020). doi:10.20944/preprints202009.0207.v1. https://www.preprints.org/manuscript/202009.0207/v1
12. Vaser, R., Sović, I., Nagarajan, N., Šikić, M.: Fast and accurate de novo genome assembly from long uncorrected reads. Genome research **27**(5), 737–746 (2017). doi:10.1101/gr.214270.116. 28100585[pmid]
13. Walker, B.J., Abeel, T., Shea, T., Priest, M., Abouelliel, A., Sakthikumar, S., Cuomo, C.A., Zeng, Q., Wortman, J., Young, S.K., Earl, A.M.: Pilon: An integrated tool for comprehensive microbial variant detection and genome assembly improvement. PLoS ONE **9**(11), 112963 (2014). doi:10.1371/journal.pone.0112963
14. Medaka: https://github.com/nanoporetech/medaka (2021). https://github.com/nanoporetech/medaka
15. Nanopolish: https://nanopolish.readthedocs.io/en/latest/index.html (2020). https://nanopolish.readthedocs.io/en/latest/index.html
16. Hunt, M., Silva, N.D., Otto, T.D., Parkhill, J., Keane, J.A., Harris, S.R.: Circlator: automated circularization of genome assemblies using long sequencing reads. Genome Biology **16**(1), 294 (2015). doi:10.1186/s13059-015-0849-0
17. Simão, F.A., Waterhouse, R.M., Ioannidis, P., Kriventseva, E.V., Zdobnov, E.M.: BUSCO: assessing genome assembly and annotation completeness with single-copy orthologs. Bioinformatics **31**(19), 3210–3212 (2015). doi:10.1093/bioinformatics/btv351. https://academic.oup.com/bioinformatics/article-pdf/31/19/3210/17086320/btv351.pdf
18. Gurevich, A., Saveliev, V., Vyahhi, N., Tesler, G.: QUAST: quality assessment tool for genome assemblies. Bioinformatics **29**(8), 1072–1075 (2013). doi:10.1093/bioinformatics/btt086
19. Laetsch, D., Blaxter, M.: Blobtools: Interrogation of genome assemblies [version 1; peer review: 2 approved with reservations]. F1000Research **6**(1287) (2017). doi:10.12688/f1000research.12232.1
20. KAT: The K-mer Analysis Toolkit, https://github.com/TGAC/KAT (2018). https://github.com/TGAC/KAT
21. Nattestad, M., Schatz, M.C.: Assemblytics: a web analytics tool for the detection of variants from an assembly. Bioinformatics **32**(19), 3021–3023 (2016). doi:10.1093/bioinformatics/btw369
22. Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., and, R.D.: The sequence alignment/map format and SAMtools. Bioinformatics **25**(16), 2078–2079 (2009). doi:10.1093/bioinformatics/btp352
23. Darling, A.C.E., Mau, B., Blattner, F.R., Perna, N.T.: Mauve: multiple alignment of conserved genomic sequence with rearrangements. Genome research **14**(7), 1394–1403 (2004). doi:10.1101/gr.2289704. 15231754[pmid]
24. Anaconda: Anaconda Software Distribution. Anaconda Inc. (2021). https://docs.anaconda.com/
25. Kurtzer, G.M., Sochat, V., Bauer, M.W.: Singularity: Scientific containers for mobility of compute. PLOS ONE **12**(5), 1–20 (2017). doi:10.1371/journal.pone.0177459
26. Mohamed, M., Dang, N.T.-M., Ogyama, Y., Burlet, N., Mugat, B., Boulesteix, M., Mérel, V., Veber, P., Salces-Ortiz, J., Severac, D., Pélisson, A., Vieira, C., Sabot, F., Fablet, M., Chambeyron, S.: A transposon story: From te content to te dynamic invasion of drosophila genomes using the single-molecule sequencing technology from oxford nanopore. Cells **9**(8), 1776 (2020). doi:10.3390/cells9081776. 32722451[pmid]

Figures

[scale=0.25]image-diagrammeculebront.png

Figure 1 Setting up parameters, building a workflow, executing and viewing results of CulebrONT. (1) Parameters are provided by the user in the *config.yaml*, *tools_path.yaml* and *cluster_config.yaml* files. (2) CulebrONT python class will read, transform and link rules to build the corresponding pipeline. (3) Each rule can be executed linearly or in parallel (HPC or multithreaded single machine). (4) Results are outputted in an HTML report.

[scale=0.25]image-report.png

Figure 2 Example of CulebrONT HTML reports. A. Busco statistics for 2 *Xanthomonas sp.* dataset B. Quast for the rice dataset.

Tables

Table 1 Standard results obtained using CulebrONT on various models and assemblers. The *"Organisms"* row describes information about samples, the *"Reads"* one describes reads quality, the *"Activated tools"* one shows used tools in CulebrONT, and a summary of the best results is presented in the *"Best workflow"* row.

Specie			*Xanthomonas oryzae*	*Xanthomonas oryzae*	*Meloidogyne graminicola*	*Oryza sativa*
Organism	Sample name		VOX070	VOX015	VN18	DJ123
	Kingdom		Bacteria	Bacteria	Nematoda	Plantae
	Genome Size		4.8M	4.8M	35M	380M
	Ht %		-	-	2%	0.20%
	Ploidy		haploid	haploid	haploid	diploid
Reads						
	Number		63,658	181,508	671,385	930,909
	N50 (pb)		24,037	20,653	9,372	22,328
	Total bases (pb)		621,636,147	1,869,930,611	3,283,238,647	14,599,973,374
	Depth		124X	370X	93X	38X
Activated tools						
	Assemblers					
			yes	yes	yes	yes
		Raven	yes	yes	yes	yes
		Miniasm	yes	yes	yes	no
		Smartdenovo	yes	yes	yes	no
		Shasta	yes	yes	yes	yes
		Canu	yes	yes	no	no
	Circulatisation		yes	yes	no	no
	Polishers					
		Racon	x2	x2	x2	x3
	Correcteurs					
		Medaka	yes	yes	yes	no
		Nanopolish	yes	yes	yes	no
		Pilon	no	no	yes	no
Best workflow			Flye+2xRacon+Medaka	Canu+2xRacon+Medaka	Flye+2xRacon+Medaka	Flye+3xRacon
	Busco score		C:98.4%[S:98.4%,D:0.0%]	C:99.2%[S:99.2%,D:0.0%]	C:65.9%[S:64.7%,D:1.2%]	C:95.1%[S:93.7%,D:1.4%]
	Largest contig (pb)		4,917,887	4,985,087	1,984,010	26,745,371
	Contigs number		1	2	617	261
	N50 (pb)		4,917,887	4,985,087	402,454	14,895,042
	L50		1	1	29	10

Additional Files

Additional file 1 — Additional information

Additional file descriptions of the CulebrONT implementation can be found on the article_CulebrONT_suplementary_file.tex document.

A

CulebrONT report

Rulegraph
Config file Parameters
Benchmark
BUSCO
QUAST
ASSEMBLYTICS
Versions

09 June, 2021

BUSCO

BUSCO sets are collections of orthologous groups with near-universally-distributed single-copy genes in each species

VXO70 VXO15

Search

Assembler	Steps	Complete	Single	Duplicate	Fragmented	Missing
A	All				All	
CANU	ASSEMBLY	91.1%	91.1%	0.0%	6.5%	2.4%
CANU	CORRECTION_MEDAKA	98.4%	98.4%	0.0%	0.8%	0.8%
CANU	CORRECTION_NANOPOLISH	85.5%	85.5%	0.0%	11.3%	3.2%
CANU	POLISHING_RACON	91.9%	91.9%	0.0%	6.5%	1.6%
FLYE	ASSEMBLY	96.8%	96.8%	0.0%	2.4%	0.8%
FLYE	CORRECTION_MEDAKA	98.4%	98.4%	0.0%	0.8%	0.8%
FLYE	CORRECTION_NANOPOLISH	86.3%	86.3%	0.0%	11.3%	2.4%
FLYE	POLISHING_RACON	88.7%	88.7%	0.0%	8.1%	3.2%
MINIASM	ASSEMBLY	91.1%	91.1%	0.0%	6.5%	2.4%
MINIASM	CORRECTION_MEDAKA	98.4%	98.4%	0.0%	0.8%	0.8%
MINIASM	CORRECTION_NANOPOLISH	87.1%	87.1%	0.0%	11.3%	1.6%
MINIASM	POLISHING_RACON	91.1%	91.1%	0.0%	6.5%	2.4%

Showing 1 to 12 of 12 entries

B

CulebrONT report

Rulegraph
Config file Parameters
Benchmark
QUAST
nerica4
chhomrongDhan
nericaL19
DJ123
Og132

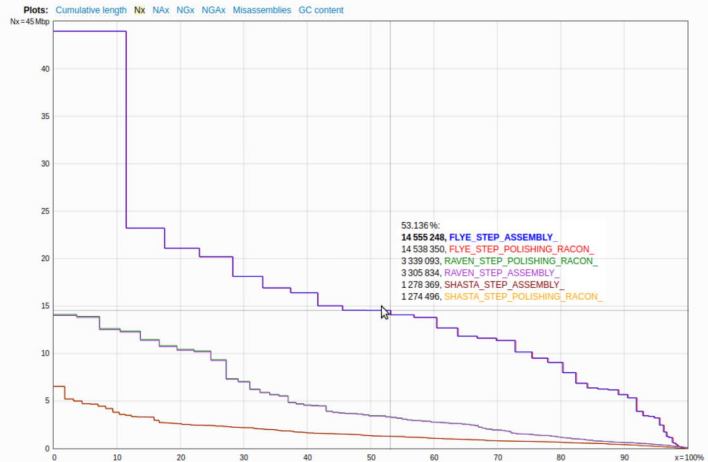
03 April, 2021

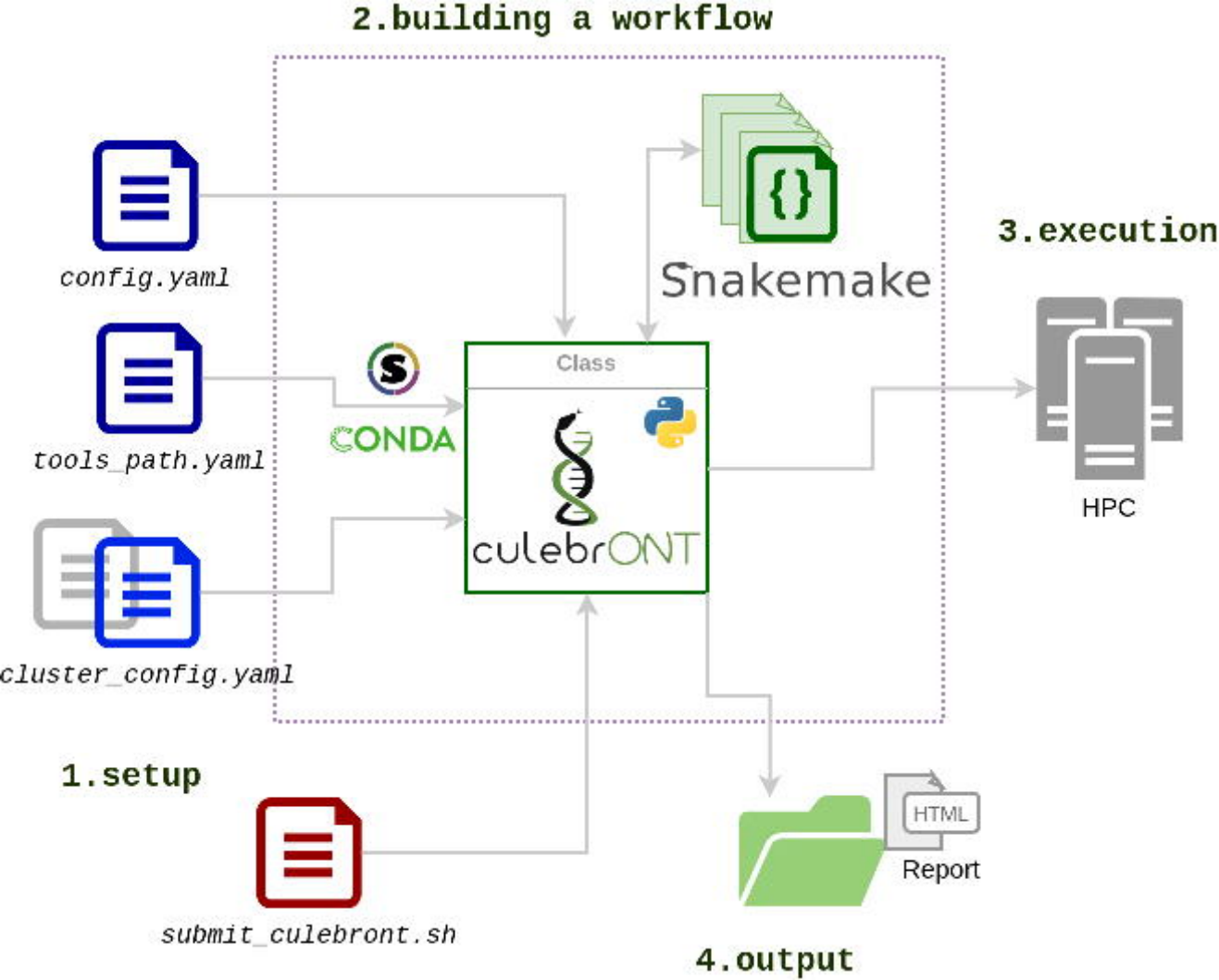
QUAST

QUAST is a good starting point to help evaluate the quality of assemblies. It provides many helpful contiguity statistics.

nerica4 chhomrongDhan nericaL19 DJ123 Og132

Open Quast report on new window




2.building a workflow
config.yaml
tools_path.yaml
cluster_config.yaml
1.setup
submit_culebront.sh
CONDA
Snakemake
Class
culebrONT
3.execution
HPC
HTML
Report
4.output