Reply to recommenders comments on round #1 following submission to PCI Genomics for "CulebrONT: a streamlined long reads multi-assembler pipeline for prokaryotic and eukaryotic genomes" (https://www.biorxiv.org/content/10.1101/2021.07.19.452922v3) by Julie Orjuela, Aurore Comte, Sébastien Ravel, Florian Charriat, Tram Vi, Francois Sabot, Sébastien Cunnac

Dear PCI Genomics recommender and reviewers,

We would like first to thank you for your positive and encouraging comments on our manuscript and its underlying ressources. We greatly appreciated the various suggestions for improving the manuscript and the set of utilities included in CulebrONT.

As detailed below in our replies to individual recommenders comments (in blue), we have made the required modifications to both our manuscript and the https://culebront-pipeline.readthedocs.io documentation to clarify their content and promote a better understanding by the reader.

After careful consideration of the feedback from the reviewing process, we hereby submit a revision of the initial version in biorxiv https://www.biorxiv.org/content/10.1101/2021.07.19.452922v4. .

We are sincerely grateful to the recommenders for taking the necessary time and efforts to critically assess our work and provide very useful comments.

Ms. Julie Orjuela

Reviews

Reviewed by Benjamin Istace, 15 Mar 2022 09:54

I read the manuscript with great interest. The authors describe a new pipeline named "CulebrONT" that they developed in order to be able to test multiple genome assemblers at once. The pipeline also performs optional steps like the polishing and the circularization and outputs QC metrics that are often used to assess the quality of genome assemblies. I personally think that this type of pipeline is very useful to the community, as it aggregates the most commonly used tools in order to improve the ease of use for the end-user. I only have very minor concerns that I would like the authors to address if they agree with me.

Thank you very much for your positive feedback and the numerous interesting suggestions.

Introduction - line 30-31: "Oxford Nanopore Technologies (ONT) and Pacific Biosciences (PB), provide reads up to hundreds of thousands of bases in length" - While I agree with this statement for ONT reads, PacBio reads are generally around 15kb and I don't think I have ever seen a read larger than 30kb. Would you have a reference for this?

We agree with the reviewer here, it was a typing mistake. We modified the text accordingly: *“Pacific Biosciences (PB) and Oxford Nanopore Technologies (ONT), provide reads up to 25kb in length, and even hundreds of thousands of bases for ONT.”*

I also rapidly tested t they developed in order to be able to test multiple genome assemblers at once. The pipeline also performs optional steps like the polishing and the circularization and outputs QC metrics that are often used to assess the quality of genome assemblies. I personally think that this type of pipeline is very useful to the community, as CulebrONT on a small yeast genome and I also have some suggestions:

- R is not specified as a dependency but it is required to install the CulebrONT PyPI package (it is required by RPy2, which is a dependency of Pandas). I think that stating that it is required would be a good idea because it produces weird error messages otherwise.

We agree with your comment; indeed, the more recent versions of RPy2 now require R as a dependency. We modified the documentation into the requirements section to incorporate these changes https://culebront-pipeline.readthedocs.io/en/latest/INSTALL.html#requirements.

- during the installation step (install_cluster), I chose to use the singularity environment. I think that it would be a good thing to indicate in the docs that images will be downloaded in CulebrONTs install directory. Indeed, it was installed in my home for testing purposes and completely filled it up. It was an easy fix to create a virtualenv in a more spacious filesystem but seeing it mentioned somewhere would be better I think.

We never encountered this problem, and we thank the reviewer for pointing it out. We modified our documentation in the “steps-for-local-installation” section to clarify how much free disk space is required for the whole CulebrONT installation. https://culebront-pipeline.readthedocs.io/en/latest/INSTALL.html#steps-for-local-installation

Globally, the pipeline is relatively easy to use and configure with helpful messages.

Some general comments/suggestions, with no impact on the result of the review:

- The inclusion of Smartdenovo in the list of assemblers is a good point, as we often get good results with it but it is a lesser-known software. You could also take a look at Necat, which is a new assembler that often leads to pretty great assemblies of complex genomes.

In the next releases we plan to offer the option to supply an “external assembly” as fasta input into the quality part of CulebrONT. This “external assembly” will be added to the set of the CulebrONT ones and provided as input to the Quality Control section of our pipeline. Users will then be able to compare the “external assembly” from an assembler not included in CulebrONT (e.g. Necat) to the ones from the pipeline.

We are also planning to add new assembly and others quality tools, but we are wondering if we should not keep the tools that are still in development: typically Necat has not been released for 2 years

- I don't want to seem like I am pushing my own tool but for the polishing step with short reads, we developed Hapo-G which specifically handles heterozygous genomes while still

doing great with homozygous/haploid ones. It's just a comment, I won't take it personally if it is not considered for this pipeline.

Indeed, we understand and agree on the importance of haploidization tools for large genomes. Nevertheless, in the current context, adding this tool in CulebrONT requests large changes in the software structure that cannot be performed in a reasonable amount of time. It is however planned for future releases (i.e. probably during 2023) to include the haplotyping possibilities in CulebrONT.

- Merqury is another tool that is very practical to assess the quality of an assembly. It is used with Illumina short reads and compares the k-mers that are in the assembly to the ones of the Illumina reads. It then gives a Q-score to the assembly based on shared k-mers.

We agree with the reviewer, Merqury is by now one of the main standard QC tools. We have thus implemented it in the last release, 2.1.0 version of CulebrONT associated with this resubmission.

- I am a big fan of Singularity and containers in general so seeing them included in a pipeline makes me very happy.

We agree with the reviewer, and thank him for the comment. The immutability (or almost) of Singularity environments, coupled with their impermeability compared to Conda or other similar systems and the possibility to use it in HPC, made this technology the best choice for us (after many tests!).

Reviewed by Valentine Murigneux, 20 Apr 2022 16:29

The manuscript describes the software tool culebrONT, whose goal is to help benchmark assembly pipeline. The introduction clearly explains the motivation of the pipeline development. This is a very useful tool that should be useful to many in the genome assembly community, who can be easily overwhelmed by a growing number of tools available and the fact that no tool performs best for every sample dataset. To my knowledge, there is no similar workflow/software currently available in the community. The pipeline aims to solve common challenges for the user to install different tools prior to running them and comparing their results. Raw data and the source code are available to the reader. The pipeline is extremely well documented, illustrated and currently well maintained with an active Github webpage. A useful feature of the software is the Html report generated containing results, multiple graphes and the version of the tools.

We are very grateful for your positive comments and thoughtful suggestions. We address them individually below.

I have a few questions and suggestions:

-line 14" Implementation

CulebrONT uses Snakemake [4] functionalities, enabling readability of the code,

local and HPC scalability, reentry, reproducibility and modularity. "

I am not familiar with snakemake functionalities therefore it could be useful to provide a few details on each aspect for the reader.

We have modified the corresponding paragraph of the introduction in order to very briefly explain those concepts and how the CulebrONT's implementation embodies them.

-Following up on the previous suggestion , I was looking for more details about the "modular" aspect of the pipeline. How easy is it for a user to add a new tool to the pipeline, e.g. a new assembler or polisher? Can a user do it thanks to the modular aspect of the pipeline and its open source status?

We understand the reviewer's comment: here the term modular in this case does not refer to the easy integration of a tool in the pipeline but rather to the choice given to the user to activate or deactivate steps in the pipeline. While based on SnakeMake, introducing a new tool that can behave correctly with all the different steps in CulebrONT is not so easy. We changed the text accordingly to clarify our idea.

- Same question for a new version of a tool.

Can the user choose to use a new version of any tool, i.e. a more recent than the one listed on this page? https://culebront-pipeline.readthedocs.io/en/2.0.1/ABOUT.html#assembly

The tool versions described in this page are the minimal versions that we validated in CulebrONT, the last ones we tested being the ones integrated in the Singularity image. However, excepting for modification of option writing or outputs, there is no reason for a new, more recent tool version to function. We modified the manual accordingly to this

-the scalable aspect of the pipeline could be illustrated by a few examples. I wonder if examples could come from the "Application" section which contains several use cases from "personal communication" especially plants which require more computational resources. Is it possible /useful to provide more details here.

Here, scalability is defined as the possibility to easily manage multiple assemblies and ways to obtain these assemblies. Indeed, CulebrONT allows you to launch assemblies and correction/polishing for many samples without being bored by files and flow management. As an example, we were able to launch more than 40 different bacterial genomes at once, and more than 10 plant (rice) ones. True computational scalability, i.e. the real time, will depend on many parameters: tools, options, hardware infrastructure… Thus, as an example, a raw miniasm bacterial assembly for 1 sample will be really fast on a single laptop computer, while a Canu one for 10 rices will require the availability of an HPC cluster and lasts for days…

-The manuscript does not contain a discussion section. The authors could comment on future developments/improvements planned for the pipeline if there are any. How and how often are the authors planning to maintain/ update/ improve the pipeline?

We agree with the reviewer that this information was missing. We now added a Discussion section dealing with this matter.

- The report includes the run time for each step of the pipeline. Is there an easy way for snakemake to also include the computational resources used e.g. memory/CPU ?

We thanks the reviewer for this nice idea, and we have included it in the discussion as an area for future CulebrONT development

- Table 1: the legend does not mention if those examples are exclusively from ONT data?

Yes, and we corrected the legend according to that

-Table 1: the busco score for the nematode sample is quite low 65%, is there an explanation?

We suspect here that the BUSCO db for nematodes may not be perfectly adapted for this nematode. In addition, the initial data coverage was low and thus may explain the low BUSCO score.

-The background section mentions past research in the field and available software. culebrONT aims at providing a workflow chaining different tools to facilitate genome assembly and compare different assembly results. Although restricted to prokaryotic genomes, previous benchmarkings of long read assemblers could be cited in the introduction (e.g. https://f1000research.com/articles/8-2138, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7730629/, https://www.nature.com/articles/s41598-020-70491-3) as well as a workflow for bacterial genome assembly using long read sequencing published in 2021 (https://bmcgenomics.biomedcentral.com/articles/10.1186/s12864-021-07767-z). CulebrONT includes a lot of similar tools as included in those publications. CulebrONT provides the advantages of reporting the results of several combination of tools to facilitate their comparison.

We agreed with the reviewer and add several more references in the introduction