Decision

by François Rousset, 2019-07-16 13:06

Manuscript: https://doi.org/10.1101/638288

Major revision

I managed to obtain two reviews. One of the reviews highlights why this ms may be eventually worth recommending by PCI. Nevertheless, it also notes two important weaknesses, and the other review points additional important issues. I summarize these criticisms below to make clear the main revisions that appear required for the ms to be eventually recommended.

Reponse: We would like to thank both reviewers and the editor for reviewing and considering our manuscript. We feel that their highly constructive comments helped us to considerably improve our manuscript from the original version and we hope that have managed to address them accordingly.

From the first review:

"ACACIA might be advantageous to the existing programs / workflows, [but] this is not really fully tested in the manuscript": comparisons should be provided.

Reponse: We provide a full comparison between ACACIA and AmpliSAS where we compare across a range of settings and we now demonstrate that ACACIA consistently give higher allele calling accuracy and repeatability. As now mentioned both in the article and to our response to the first review, there are no other software (that we are aware of) that work as an automated genotyping workflow for complex multigene systems such as the MHC. All other software such as SESAME and jMHC allowed users to demultiplex sequences and to generate tables which contains sequence variants and the number of reads (added to discussion, lines 492-498). However they do not allow users to apply an automated workflow to distinguish artefacts from real allelic variants.

"The authors should either have run all settings in one study data-set or one setting in all data sets (or all combinations for all data sets)." Here the issue is : what can be concluded from the different analyses? I guess that the authors will be able to partially rebut this question, but it is not clear what is meant by "test" on l. 183 ("test ACACIA in wildlife species with unknown genotypes of varying CNV").

Reponse: As mentioned above we now presented the results across all settings (within reason) for both ACACIA and AmpliSAS and across datasets. We also acknowledge that we did not indeed “test” (we should have said apply instead of test) ACACIA *per se* using the dataset from a wildlife species with unknown genotypes of varying CNV. We merely motivated to provide an example with a “real” wildlife dataset. We have now also removed the dataset of the wildlife species at the suggestion of reviewer 1 since we agree that it distracts from the formal demonstration of the high performance of ACACIA as genotyping tool.

The second review highlights that ACACIA is not yet really a "pipeline" but rather an interactive script. Most importantly, it expresses concerns about the repeatability of the analyses. I concur with this review that reproducible(s) example(s) should be provided. This review also implies that the version described in the ms should be made permanently accessible. I see the point but I am not sure it is the best way to address the issue of reproducibility. An alternative view is that future versions should be tested against the results of the current version, which brings us back to the issue of providing reproducible examples.

Reponse: We completely agree that reproducible examples and formal versions of our pipeline and we believe we have now addressed this now fully in our detailed answer to reviewer 2 below. We also agree that our pipeline is an interactive script, however this fits with the definition of a bioinformatic pipeline that we have provided in the introduction (lines 101-104) and fits with how the term is used generally in the computer science and bioinformatic world (see for instance https://www.quora.com/What-are-pipelines-in-Bioinformatics or https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5429012/).

I hope the authors will be able to submit a revised version addressing all these points.

Response to reviewers:

Reviewer 1 (Helena Westerdahl):

Gillingham and Co-workers have developed a workflow to improve the genotyping of MHC class I and IIB genes in non-model organisms. The MHC genes are often highly duplicated and although the different gene copies are extremely polymorphic different alleles can be highly similar. High-throughput amplicon sequencing enables the characterization of the MHC genotype in individuals/species with extreme MHC gene copy numbers.

The first main step in the laboratory is a PCR that amplifies over the entire multi-locus of MHC I or MHC IIB (all gene copies are amplified at the same time) and then follows high-throughput amplicon sequencing, here using Illumina. Both these steps in the laboratory generate artefacts, hence MHC alleles that do not really exist. These artefacts need to be filtered away and this is the main topic of the present manuscript. How can we filter away artefactual MHC alleles while keeping all the true alleles? Several different research groups have been engaged in this filtering process and different laboratories use different programs / workflows. The present study provides a partly novel pipeline how to filter MHC data, ACACIA, it is user friendly and flexible according to the authors. ACACIA might be advantageous to the existing programs / workflows, this is not really fully tested in the manuscript but I think the ACACIA workflow seems excellent and I am very positive to this part of the manuscript!

Reponse: We are very grateful for the positive comment and we now hope that we have done a

better job at fully testing ACACIA as suggested (see below).

Gillingham and Co-workers use three different data sets that they filter using their workflow ACACIA:

- 241bp MHC class IIB that has been amplified using perfect chicken primers, in a PCR with 30 plus 10 cycles with 10 minutes of extension.
- 151bp MHC class IIB that has been amplified using poor chicken primers, in a PCR with 30 plus 10 cycles with 10 minutes of extension.
- 236bp MHC class I that has been amplified using primate primers, in a PCR with 25 plus 7 cycles without any 10 minute extension.

These three data sets differ in amplicon length, number of PCR cycles, PCR settings, target genes, amplicon read depths etc.. These differences are major and all have considerable effects on the outcome of the filtering. It is impossible, or at least very difficult, to draw any conclusions on findings considering artefacts / chimeras etc. between these data sets since a large number of the parameters differ between the data sets. The authors should either have run all settings in one study data-set or one setting in all data sets (or all combinations for all data sets).

Reponse: As suggested we now present the results of the different ACACIA pipeline settings (low_por and abs_nor) across all chicken datasets. The original version had already investigated the optimal settings for the optimal primer dataset, where all settings (or at least within reason) were tested for both AmpliSAS and ACACIA. However the results were only partly presented in the supplementary material. We now provide the results of this full comparison in the main body of the article. Please note that all other pipeline/workflow settings, including chimera detection, is the same across chicken datasets.

We agree that PCR settings for the lemur dataset were slightly different to the two chicken datasets (but were identical between the two chicken datasets) which make comparisons between the chicken datasets and the lemur dataset difficult. Following Prof. Westerdahl's suggestion (see below), we have now removed the lemur dataset from our manuscript and focus only on the chicken datasets for which all wet lab protocols and pipeline settings are the same (only primers and fragment length differ between the two datasets).

We fully agree that different amplicon lengths will result in considerable effects on the settings of the filtering, which is why it is one of our take home messages that optimal settings of any pipeline/workflow will be specific to a particular dataset.

However, in my opinion the authors have an excellent set-up with the chicken and the artificial MHC genotypes, very clever design, and my advice would be to focus your ACACIA manuscript on the chicken only and present your nice ACACIA workflow based on the chicken data with the perfect chicken primers. The MS would also improve if you compared the output and

repeatability of the ACACIA with several different MHC filtering pipelines. At present you only test AmpliSAS. I think you should compare your ACACIA pipeline with that of other filtering programs / workflows and explain how they differ and also what similarities they have, and why you choose the setting that you did.

Reponse: We are again very grateful for the positive comment regarding our experimental set-up. Following Prof. Westerdahl's suggestion, we have now focused on the chicken datasets and removed the lemur dataset from our manuscript. As mentioned above we now present results from different low_por and abs_nor settings of the ACACIA pipeline so that readers can easily see why we have chosen the settings for our datasets. We have removed the lemur dataset but kept the naïve dataset. We decided to keep the naïve dataset because we feel it provides an important example of the difficulty of genotyping unknown systems, of the importance of primer design and to ensure that we avoid tailoring the pipeline to a single dataset.

To our knowledge the only available software that works as a workflow is AmpliSAS , which why we have chosen to compare our results only with AmpliSAS. Other software such as jMHC and SESAME only allow users to visualise unique sequences and their frequency and do not function as a workflow. We now mention this in our manuscript (lines 492-498).

Different genotyping pipelines for complex gene families have been extensively reviewed by Prof. Westerdahl and colleagues in Biedrzycka et al. (2017) using AmpliSAS. However repeating the same approach as Biedrzycka et al. (2017) did but with our datasets and ACACIA seemed rather redundant to us, since the conclusions of Biedrzycka et al. (2017), namely that there is high agreement between genotyping methods provided there is sufficiently high sequencing coverage is very likely to be the same when using ACACIA. Furthermore, we felt that replicating Biedrzycka et al. (2017) with ACACIA would distract from the main purpose of the article, which is to describe the relationship between artefacts and CNV and to present a novel bioinformatic tool. However, as suggested, we have now provided an extensive section in the discussion which discuss in more detail the different workflows that other workgroups have used and compare them to ours (lines 499-544).

Finally, figure 2 reuses data, 2b plus 2c is equal to 2a, not so, moreover it is tricky to interpret and understand figure 2, think about rearranging this figure.

Reponse: We have now rearranged the figure as suggested by the reviewer.

Reviewer 2 (Thomas Bigot):

This article presents a workflow to improve multi-locus genotyping. They propose an experimental set-up and a pipeline named Acacia to perform the genotyping itself. They chose chicken as a model organism, and try to characterize sequences of MHC B Complex with their tool.

The manuscript is well-written.

According to my skills, I will focus this review on the pipeline and its bioinformatics aspects.

Reponse: Once again we would like to thank the reviewer for his high quality comments which considerably help us improve our workflow.

The ACACIA pipeline

Description in the article

- The introduction (L 273) should mention Biopython as a dependency;

Reponse: Done, also for the other dependencies.

- Some step were coded ad-hoc, even being non-trivial (e.g. Trimming low quality ends). The reason why well known methods wer not used should be briefly explained.

Reponse: The Python functions for trimming both primers and low-quality ends (step #2) are part of the ACACIA pipeline. Whenever possible, external tools were avoided in order to decrease dependency on further software and to simplify installation. This now explained in the manuscript

- Input data is not explained. In the documentation, three input files are listed. One of them is *A fasta file with 100+ sequences related to those that you expect to have sequenced. This file will be used to setup a local BLAST database.*. This description is not clear and BLAST is not mentionned in the manuscript.

Reponse: We have now completely redesigned the GitLab page, the organization of the pipeline code and the availability of example data as follows:

- GitLab Page: this is the central repository of ACACIA. The page now has a much more detailed description of the installation steps
- Installation: installing all necessary software was simplified significantly with the use of a dedicated Python virtual environment and the Bioconda channel, which takes care of the installation of all 3rd-party software.
- The Code: while all code was originally in one single file, it has now been split into several files in 1 root folder with 3 subfolders in order to improve readability, debugging and future developments.
- Example Data: both the manuscript and the README section of the GitLab page now point to a FigShare repository, in which the pipelines's source code and a series of example data (including the FASTA file mentioned, the results of a real sequencing run and a file with ACACIA settings) are publicly available. BLAST is now explained both in the GitLab and in the manuscript.

- FLASH and Pandas are used in the script but not mentionned in the manuscript.

Reponse: all external dependencies are now mentioned in the manuscript.

The pipeline itself

Reproducibility of the code

I have a major concern about reproducibility: the only code available is the master branch of the git repository. If a user downloads the pipeline in the future, nothing can tell the code available at this time corresponds to the one described in this article, and nothing guarantees the code is still available on Github. Hence, I strongly suggest to:

- create a release number of the code (eg v1.0) and indicate this number in the article;
- create an archive of this release and upload it to zenodo or figshare (or any repository of this kind);
- get a DOI from them, and indicate it in the pipeline description.

Dataset: reproducibility of the analysis

I wish I could test the pipeline, but no example dataset is provided. Moreover, the article describes the analysis of a peculiar one (Chicken HMC), so it should be included. I suggest to upload it (fastq data, primers, "well known sequences") at zenodo or figshare like explained just above, and indicate the DOI in the article, and in the documentation files as a testing procedure.

Reponse: we share the reviewers concern with reproducibility and we believe to have addressed this now with our documentation strategy. As suggested, we have now created a static snapshot of the pipeline (tag V1.0, https://gitlab.com/psc_santos/ACACIA/-/tags/V1.0), which will always be available through the GitLab alongside with future versions. As a precaution, all code is also deposited in the FigShare repository (with a corresponding D.O.I., https://doi.org/10.6084/m9.figshare.9952520). The ACACIA page at FigShare.org (https://figshare.com/projects/ACACIA/66485) is publicly searchable and available. The latter repository hosts, in addition to the source code, all the files necessary to repeat the analysis described in the paper:

- all raw FASTQ files
- the FASTA file used to generate a BLAST database
- the sequences of all primers
- a list of settings used for parameters such as quality and other filter thresholds.
- the exact ACACIA code used for this paper.

As a further effort towards repeatability (and also to outreach to users "bound" to Windows systems), we have made available, also though FigShare, the image of a virtual machine which has all software already installed and all data files in place. This image file can be installed in any computer, using the VirtualBox platform (https://www.virtualbox.org/), independent of the operating system.

Pipeline manager

The program is an interactive script, asking questions to the user who has to wait during the whole time of the analysis. No argument can be provided to the pipeline at the launching time. The files must be at certain places with certain names. Moreover, all the steps are performed in one run: if one step fails, it has to be restarted from the beginning.

This script should be transformed as a real pipeline, using a dedicated software. As authors seem to have a good level of Python, I suggest them to choose Snakemake (https://snakemake.readthedocs.io/en/stable/). It is a Python tool: each step code chunk could be simply copied/pasted in the Snakemake recipie.

Reponse: We have considered and discussed the pros and cons of SnakeMake exhaustively and have decided to redesign the pipeline's code, including our own pipeline manager, instead of adapting the code to SnakeMake. By doing that and ensuring repeatability (explained above), there was little benefit left from SnakeMake. Apart from the measures towards repeatability mentioned earlier, we did the following:

- redesigned the code to be more stable by handling all exceptions and testing user input (it is now much harder to be "thrown out" of the pipeline).
- redesigned the code to make it easy for users to repeat steps of the pipeline without having to rerun everything. This is accomplished by a "config.ini" file, which controls all variable settings and can be changed manually by users. Users can now run the entire workflow without interaction, if a full config.ini file is provided with all settings. It is now also easy to pause (or kill) the process at any point and resume the workflow at a later time. Instructions on how to do the latter or repeat individual steps of the pipeline are given in the GibLab page.

Other remarks

L 216, L 217: "Naive" and "naively" do have an umlaut in English.

Reponse: Done