Author's Response To Reviewer Comments

Close

Reviewer reports:

Response: We thank the referee for the comments and for their time dedicated to this manuscript.

Reviewer #1: In this manuscript, the authors describe the usability and the performance of public cloud computing services and their batch job execution services for the transcriptome annotation pipeline. The authors also compare the services of two major public cloud vendors, Google Cloud Platform (GCP), and Amazon Web Service (AWS). The results show that the two cloud providers can provide a similar experience on its operation, workflow execution time, and payment for execution. The performed experiments follow the modern best practice of data analysis using tools and frameworks for better reproducibility, including Docker container, the Common Workflow Language (CWL), and Jupyter Notebook. This article can be a good example of a reproducible study with open data and open-source software.

This report is very significant with the practical statistics, which can be a helpful reference for all the cloud use cases in biomedical data analysis. The title states that this study focuses on the transcriptome annotation, but the output provides insight for all the cloud use cases. I suggest the authors change the title because the current one looks the best practice valid only for the transcriptome annotation in the wide variety of genomic data analysis.

Response: We thank the referee for their comment and suggestions. It is true that the results of this study provide insight for many computational biology workflows but our experiments were limited to the transcriptome annotation process, specifically to the BLAST searches that are the core of the annotation. We think that the best practices and conclusions of this study should remain limited to the cloud transcriptome annotation process.

Below are minor comments for the manuscript.

Page 3, the last paragraph:
The authors claim as "little has been published describing cloud costs and implementation best practices". However, there is a study implemented software to monitor the runtime metrics of a given workflow, and support the cost estimation for the executions on the cloud (https://doi.org/10.1093/gigascience/giz052). This article describes the cost for the normal EC2 instance and does not mention the batch execution services, yet it provides additional information to the readers. Please consider introducing this in the background section as a related study. (Disclaimer: I am the first author of this article)

Response: We agree with the referee that this study provides additional information about the use and cost of computational biology workflows in the cloud. A new paragraph was added to the Background section: “The utilization of cloud environments for computational biology experiments is increasing [14-17], however, little has been published estimating cloud costs and implementation best practices. A recent work published by Ohta at al. [18] presents a tool named CWL-metrics that collects runtime metrics of Docker containers and workflow metadata to analyze workflow resource requirements. This study presents a cost estimation for the execution on the cloud for AWS EC2 instances, but does not mention the cloud batch system for users to submit thousands of jobs to the cloud.”

Page 12, Best practices
The authors describe here that they recommend CWL as a workflow language. I think the authors should introduce the reason they chose CWL at the Method section where they first mention CWL (Page 4, Transcriptome Annotation Workflow). The authors also should provide a more practical reason to choose CWL because CWL is not only one framework that has portability and scalability. For example, having multiple workflow runners or its syntax easily parsed and connected to the resources such as runtime metrics can be a reason to choose CWL in this use case.

Response: We agree with the referee about the inclusion of more information about CWL and the reason why it was selected for our study. We added a new section to the manuscript named: Common Workflow Language

Workflow dependencies
Both AWS batch and Google LifeScience allow users to specify only one container per one batch job. This means the user needs to install the workflow runner (in this article cwltool) and the tools to process the data (e.g., BLAST) in a single container. However, the current best practice for bioinformatics is to separate the containers for each tool. Thus, in most cases, users cannot reuse the containers they use for a normal computing environment to the cloud batch system. The authors need to mention the limitation that one needs to create a container for cloud batch systems. Another possible solution would be to run sibling containers from the main batch job container, though I do not know if it is feasible with AWS and GCP.

Response: The referee's comment is correct. Docker-in-Docker, the process to execute docker containers inside of another Docker container is not allowed in both GCP and AWS. We add a new section to the manuscript named GCP and AWS batch system limitations to address these limitations.

Software and framework related to the cloud batch services Many workflow languages and runners are supporting AWS batch and GCP. For example, Nextflow (https://www.nextflow.io/docs/latest/awscloud.html), Cromwell (https://cromwell.readthedocs.io/en/stable/backends/Google/), or Snakemake (https://snakemake.readthedocs.io/en/stable/executing/cloud.html#executing-a-snakemake-workflow-via-tibanna-on-amazon-web-services) can run the workflows via AWS batch or GCP. Task Execution Service (TES) of the Global Alliance for Genomics and Health (GA4GH) Cloud working group is also a framework to utilize the cloud batch system (https://github.com/ga4gh/task-execution-schemas). Some tools that run workflows on the cloud using the ETL framework (https://doi.org/10.21105/joss.01069) or cloud batch services (https://github.com/DataBiosphere/dsub) are also available. These may not be directly relevant to this study, but it would be helpful for readers to understand how the cloud batch services are being used by the researchers.

Response: We agree with the referee about the inclusion of more information about workflow managers. We added a new section to the manuscript named: Common Workflow Language.

Reviewer #2: Reviewer's report
Title: Transcriptome annotation in the cloud: complexity, best practices and cost. transcriptome data
Reviewers: Qiao Xuanyuan and Lucas B. Carey

Response: We thank the referees for their comments and time dedicated to this manuscript.

Reviewer's report:
The authors provide a perspective on transcriptome annotation in the cloud by presenting a comparative study of the two main public cloud providers, aiming to help the reader determine the proper cloud platform and its utilization in their research. The main strength of this paper is that it addresses a question that little has been studied before—the cloud cost estimates and implementation best practices. This is useful not only for labs doing transcriptome annotation, but, because all code is provided and very well commented, it might be of use for labs dealing with big data & distributed computation problems in general.
The manuscript is well written, and I have only a few minor concerns on presentation and the information that is provided.

1. The tested transcriptome data need to be clarified. I read the Data partitioning code for the creation of 20 FASTA files of the different query sizes (Fig 3). The variation in processing time among these files is consistent across clusters, and is therefore presumably due to differences in query sets in each file. This is surprising, as 10,000 is a large number. Are the differences because the sequence records were created sequentially from the input fasta file, with some genes having large numbers of hits? Would subsetting transcripts randomly result in more uniform processing times? There is almost two-fold variation in processing time between query files.

Response: The referee's comment is correct. The variation in processing time for queries with the same number of transcripts was due to the sequentially subsampling approach. We have modified the Data

Partitioning notebook to create a random sampling of the transcriptome file. The notebook now shows the statistics for each file created. All measured parameters, mean, standard deviation, minimum length, 25%, 50% and 75% quarters show similar values for all files. Modified text was added in the beginning of the Results and Discussion section “From the Opuntia pool of transcripts, we analyzed three sizes of query files: 2,000, 6,000, and 10,000 transcripts in each input query file. Two experiments were executed. First 20 FASTA files (input files for the workflow) for each query size were randomly created, see notebook “01 - Data Partitioning”. Each of these files were submitted independently as jobs to the batch systems on each cloud provider. For the second experiment, 120,000 transcripts were randomly selected and then partitioned in files with 2,000, 6,000, and 10,000 transcripts to analyze the relationship between query size, runtime and cost.”

2. Please include a figure showing the distribution of transcript lengths for Opuntia streptacantha, and write that it is the prickly pear cactus. Presumably timing depends on the transcript lengths and on the number of BLAST hits.

Response: The referee’s comment about that the timing certainly depends on the transcript length and the number of BLAST hits. We added a cell to the 01 - Data Partitioning notebook that shows the transcriptome length distribution and its statistic metrics.

3. It is difficult to draw the conclusions from the way the data are plotted in Figure 4. The author concluded that "Reducing the number of transcripts per input file will reduce the total running time but will also increase the cost of the project as more instances will be in used."
In addition to the raw data boxplot, it is better to show how the time and cost scale with query size, or with the number of instances. (The number of instances equals query size divided by total transcripts). Controlling the total transcripts and making instances as variable may be helpful in balancing the interpretation and data.
The plot below shows the relationship between the number of CPUs and time, and query size and time. (data collected using https://automeris.io/WebPlotDigitizer/). It also provides direct-viewing evidence to support the author's conclusion "AWS is more suitable for large data analysis groups to establish a set of queues and compute environments for multiple pipelines." Unlike the boxplot in the current manuscript, this figure also shows the differences in scaling between 16, 32 & 64 vCPU nodes.

Please add graphs showing query file sizes vs time (as below) and query file sizes vs cost. As well as cost vs time. These are the important take-home messages from the manuscript, but it is difficult to extract this information from the current figures.

Response: We agree with the referee’s comment that the relationship between query size, time and cost was not well described in the way that the results were presented in Figure 4. We add a new experiment where we processed 120,000 transcripts using the three query sizes. The new Figure 4 shows the relationship between the time, cost and query size for processing a fixed number of transcripts. Additional text describing this experiment was added to the Result and Discussion section: “Figure 4 shows the time and cost of processing 120,000 transcripts using second generation 64 vCPUs instances on each cloud provider. Reducing the number of transcripts per input file reduces the total run time but will also increase the cost of the analysis as more instances will be used. BLAST databases are transferred to more instances spending, on average, 10 minutes for each instance. For example, our experiment with the 10,000-query size processes all transcripts in about 105 minutes with a total cost of 59.37USD using 12 instances (GCP, N2, 64 CPU). Processing the same number of transcripts with a query size of 2,000 costs122.36USD with all transcripts processed in 43 minutes using 60 instances (GCP, N2, 64 CPU). “

Optional (not necessary) suggestions:
The figures could be improved to give a clearer visualization of the data. For Figure 3a, will adding a secondary Y-axis regarding money and draw a line plot be better to show the relationship between total time and cost? For Figure 3b&c&d, drawing a component bar chart could allow the readers to compare the job-dependent time among various configurations and demonstrates the proportionality at the same time.

Response: We thank the referee for this suggestion. In our opinion, as the data we are plotting here is not continuous data, drawing a line between points in Figure 3a could imply the idea of continuity. For Figures 3b, 3c, and 3d drawing bars could hide the variability of the runtime for each input file that is associated with instance and network performance. We prefer to keep the figure as it is now.

Reviewer #3: The authors provide a comparison of two cloud-based solutions for running BLAST-based transcriptomics analysis.
With cloud-based solutions becoming more popular in science, I think this comparison, along with the practical recommendations provided in this manuscript will be interesting to readers.

Response: We thank the referee for their comments and time dedicated to this manuscript.

Some suggestions for enhancements below:

1) The authors mention that there are numerous genomics companies in the space of cloud based biocomputations, citing reference 14 which discusses some of the legal responsibilities of groups doing such cloud based analyses. However, the authors do not connect this discussion back to the use of GCP or AWS, where users would need to obtain similar gaurantees of data security which may not be possible to obtain. Thus a comparison of pricing against the private firms which provide similar services would be interesting to see if they provide specific guarantees that affect researchers with data that requires specific legal requirements.

Response: We thanks the referee for the comment. This study was executed under the NIH's STRIDES initiative using the currently available cloud provider partners: AWS and GCP. We do not consider that a comparison between public and private cloud providers and their legal responsibilities is within the scope of this study. However, we think that mentioning some of the most important private cloud providers is necessary. Accordingly, we rephrased the paragraph as:

In addition, private genomic cloud providers, for instance DNAnexus (www.dnanexus.com), DNAstar (www.dnastar.com), Seven Bridges (www.sevenbridges.com) and SciDAP (scidap.com), also are in the market and offer cloud-based genomics frameworks. These commercial cloud providers make the execution of computational biology experiments easier offering command line and web-based interfaces designed for genomic data analysis.

2) I really like the inclusion of a best practices section with practical recommendations, and would love to see this section expanded:
a) In the first point the authors state: "We recommend CWL because the resulting product is portable and scalable,and it can be executed across a variety of computational environments as dissimilar as personal laptops or the cloud". However, these features are not exclusive to CWL, and solutions such as NextFlow and SnakeMake (and probably others) would also fit this description (and both of these also offer point 2 Conda and containerization). Please elaborate on your recommendation to include discussion of other workflow management systems, and explain in more detail why you would recommend CWL over these other solutions.

Response: We agree with the referee about the inclusion of more information about CWL and workflow managers. We added a new section to the manuscript named: Common Workflow Language.

b) In point 5, you recommend that users "Execute a small test in the cloud to find the best instance type for a workflow". Do you have any further practical recommendations about how users can best go about this? E.g. how does one define a "small test run" from a full datasets, and how can they predict how this will scale up to the full analysis and determine the most suitable machine types?

Response: We agree with the referee that defining a "small test run" may be difficult and it is intrinsically determined by the data and the workflow to be used. Our intention with this recommendation was to alert users that the cloud is a completely different environment than local workstations or on premise clusters. Users should test different cloud services and configurations before submitting a huge number of jobs. We edited that recommendation to:

5. Cloud computing behaves differently than local workstations or on premise clusters. Users should define and execute small tests with their data and workflow before submitting large jobs. Testing different cloud services and configurations could help to reduce the runtime and cost for the whole analysis.

3) It could be nice to expand the section about the Jupyter notebooks, and how these are being used, perhaps with some screenshots of results, and/or a small schematic showing that (if I understand correctly): the user interacts with the Jupyter notebook on their local machine, which in turn configures the cloud resources and starts the CWL workflow on the cloud, and then fetches the relevant results back and analyses them and displays results to the user. I think a schematic to this effect would be helpful for less technical readers and this in combination with some screenshots of the analysis results in Jupyter will increase the appeal of your work to research scientists.

Response: We agree with the referee about expanding the Jupyter notebook section. We added more description to it. The notebooks are available on Github for reading and browsing. Adding more figures to the manuscript would increase its size and complexity. Jupyter notebooks are very popular and we think that less technical readers could easily find documentation about Jupyter notebooks without any problem.

4) In the conclusion the authors state “In our opinion, the choice of a cloud platform is not dependent on the workflow but, rather, on the specific details of the cloud provider”. However, I don’t believe the authors can make this statement having tested only a singe workflow. So please rephrase the conclusion, or compare performance of different workflows covering a range of different characteristics (e.g. one that is memory-intensive, one that is CPU-intensive, and one that requires a lot of data transfer) and showing whether this conclusion holds, or whether some of the “specific details of the cloud provider” may make it more or less suitable for certain types of workflows.

Response: We agree with the referee that the phrase was general. We rephrased to this:

In our opinion, for BLAST based workflows, the choice of cloud platform is not dependent on the workflow but, rather, on the specific details and requirements of the cloud provider (e.g. NCBI maintains updated copies of the very large genetic sequence databases, such as nr, RefSeq and SRA, on both GCP and AWS). These choices include the accessibility for institutional use, the technical knowledge required for effective use of the platform services, and the availability of open-source frameworks such as application programming interfaces (APIs) to deploy the workflow.

5) AWS, Google, and Azure are probably the "big 3" providers that most readers will l know about, and they might wonder why Azure was not included here and how it would compare. I understand that the authors cannot compare all providers, but it may be useful to at least mention Azure in the introduction where different providers are mentioned, and briefly explain if there were any specific reasons why you chose to compare AWS and GCP, and whether the same methodology could also be applied to Azure.

Response: We thank the referee for the comment. This study was executed under the NIH’s STRIDES initiative using the current available cloud provider partners: AWS and GCP. We don’t have access to Azure, therefore, we cannot extrapolate the conclusions of this study to Azure or any other cloud provider.

Close