5-2021

TruncTrimmer: A First Step Towards Automating Standard Bioinformatic Analysis

Z. Gunner Lawless
University of Arkansas, Fayetteville

Dana Dittoe
University of Wisconsin-Madison

Dale R. Thompson
University of Arkansas, Fayetteville

Steven C. Ricke
University of Wisconsin-Madison

TruncTrimmer: A First Step Towards Automating Standard Bioinformatic Analysis

An Undergraduate Honors College Thesis

in the

Department of Computer Science and Computer Engineering
College of Engineering
University of Arkansas
Fayetteville, AR
April, 2021

by

Z. Gunner Lawless

Running Title: TruncTrimmer

Title of Technical Report: TruncTrimmer: A First Step Towards Automating Standard Bioinformatic Analysis

Z. G. Lawless[1], D. Dittoe[2], D.R. Thompson[1], S. C. Ricke[2]

[1] Department of Computer Science & Computer Engineering, College of Engineering, University of Arkansas, Fayetteville, Arkansas 72701, USA

[2] Department of Animal and Dairy Sciences, University of Wisconsin Madison, Madison, Wisconsin 53706, USA

Corresponding Author: Dr. Dale R. Thompson; Department of Computer Science and Computer Engineering, University of Arkansas, Fayetteville, AR 72701; Phone: (479) 575-5090; Email: drt@uark.edu

Abstract

Bioinformatic analysis is a time-consuming process for labs performing research on various microbiomes. Researchers use tools like Qiime2 to help standardize the bioinformatic analysis methods, but even large, extensible platforms like Qiime2 have drawbacks due to the attention required by researchers. In this project, we propose to automate additional standard lab bioinformatic procedures by eliminating the existing manual process of determining the trim and truncate locations for paired end

sequences. We introduce a new Qiime2 plugin called *TruncTrimmer* to automate the process that usually requires the researcher to make a decision on where to trim and truncate manually after importing and demultiplexing sequences in the Qiime2 pipeline. By automating this process and removing the need for manual interaction by the researcher, this plugin provides another opportunity to automate another standard bioinformatic analysis procedure.

Keywords: *Qiime2, Bioinformatics, Microbiome, TruncTrimmer*

Introduction

In the research world of microbiology and food science, it is not uncommon to find that the lengthy analysis of bioinformatic information creates a bottleneck for various research projects within a lab. Biologists have spent years perfecting laboratory techniques to streamline lab processes related directly to sequencing and hands-on experimentation, but a knowledge gap still exists for many researchers when it comes to using computer systems to further their research. Nonetheless, bioinformatic analysis is still part of most labs' standard operating procedures and is still an integral part of conducting modern biological research.

The lab used to perform this work routinely uses the Illumina sequencing platform [1] to run sequences for various experiments to obtain microbiome information. After gathering sequence information, it uses the Qiime2 pipeline [2] as part of its standard operating procedures to perform bioinformatic analysis. The Qiime2 platform offers an expandable plugin system with extensive support for performing various sequence

analysis techniques from taxonomic classification to beta diversity analysis. This platform allows researchers to start with raw sequence data and perform analysis all the way to publication quality graphs and figures [2]. While this platform offers extensive abilities in its analysis capabilities, the bioinformatic analysis is a bottleneck. The bottleneck is created due to the extensive amount of manual interaction required by researchers while conducting bioinformatic analysis. While the analysis itself is mostly automated through Qiime2 plugins, prepping commands and making project specific judgement calls are still required of researchers to run the data standardly through the pipeline.

All labs have standard analysis operations that are performed on the majority of projects. If there were a way to automate these standard bioinformatic analysis procedures, it would create a more efficient research environment and reduce the data analysis bottleneck. To reduce the amount of time to perform the analysis, we propose that labs automate their standard bioinformatic analysis procedures through the use of standardized scripting to remove steps that require manual interaction. We introduce a new Qiime2 plugin called *TruncTrimmer* that automates the process of determining *trim* and *trunc* parameter values that are needed to run sequence data through the Qiime2 Dada2 plugin [3] that performs the trimming and truncating of a sequence run to remove noise introduced into the process. This is one of the few manual judgement calls required by researchers in labs when performing standard analysis of sequence data. By removing this manual step, we are one step closer to achieving a fully automated standard analysis process.

TruncTrimmer uses a sliding window algorithm to analyze sequence quality scores and determine where trim and trunc values should be set for the sequences being passed into Dada2 for filtration. Other software solutions seeking to accomplish similar goals differ in that they are not incorporated into Qiime2 plugins, they are abandoned software projects that are confusing to use, and they perform actual sequence trimming completely discarding sequence data where the trim and trunc values would be set [4][5]. The proposed solution was developed initially as a Qiime2 plugin allowing easy incorporation into the Qiime2 pipeline. The developed plugin also provides an intuitive well documented interface comparable to other Qiime2 plugins. Finally, *TruncTrimmer* does not perform the actual trimming of sequence data allowing users to save data storage space as other solutions make a copy of the sequencing data and then delete parts of the sequencing doubling the amount of storage required. Other options would require researchers to maintain trimmed and untrimmed sequence copies in case of the event where the researcher needs to go back to make manual adjustments to their trim and trunc values. The long-term goal of this plugin is to produce a fully automated standard analysis procedure to reduce the manual interaction and work required by researchers.

Materials and Methodology

In the lab where this work was performed, there are standard operating procedures used for all experiments. For this work, the first step is to extract DNA from the medium used in the project. The next steps are library preparation and sequencing of the V4 region of 16S rRNA on the Illumina Miseq platform [1]. After sequencing is

performed, the sequencer uploads sequence data to BaseSpace where demultiplexed reads can be downloaded before being imported into the Qiime2 pipeline [2]. It is at this point that all wet lab processes have been completed and bioinformatic analysis is ready to be conducted by the researcher using Qiime2.

The first step when using the Qiime2 pipeline is to import sequence data into a Qiime2 artifact. A Qiime2 artifact is a file standard implemented by Qiime2 to store information about the type of data stored in a file and the source of the data. This standard is how Qiime2 implements their integrated and automatic tracking of data provenance. In this work, the processes produce raw sequence information stored in the Casava 1.8 paired-end demultiplexed FASTQ format. This data is not in a format that can be directly used within the Qiime2 pipeline, so the data must be imported into a Qiime2 artifact with a semantic type of *PairedEndSequencesWithQuality*. Qiime2 semantic types are essentially class types developed for the Qiime2 pipeline to implement standard object-oriented programming practices with interfaces for various data types. In this case, the *PairedEndSequencesWithQuality* semantic type is a subclass of the *SampleData* semantic type which is used to store raw sequence information.

Once the data is imported into a Qiime2 artifact, the data is ready to be cleaned using the Dada2 Qiime2 plugin since the sequences have already been demultiplexed. The Dada2 plugin denoise-paired method will denoise the paired-end sequences, dereplicate them, and filter chimeras [3]. However, before the data can be run through Dada2, *trim* and *trunc* parameters for the forward and reverse sequence reads must be chosen by the researcher to pass to Dada2. The trim parameters signify where the

sequences should be trimmed due to low quality caused by the process. Trimming will remove the 5’ (5-prime) end of the input sequence, which contain the bases sequenced in the first cycles of the sequencing process. Likewise, the *trunc* parameters truncate the sequence reads at the 3’ (3-prime) end of the input sequences due to another decrease in quality. The truncated sequences will be the bases sequenced in the last cycles of the process.

Determining where to trim and truncate reads is typically a judgement call made by the researcher based upon sequence quality scores included with the sequences in the results. To help the researcher make this decision, a chart of box-and-whisker plots can be generated using the summarize function within the Qiime2 Demux plugin [6]. This plugin is typically used to demultiplex sequences for labs that need that functionality. However, in this work the summarize function takes the imported sequence data stored within the Qiime2 artifact and extracts a subsample of each read’s quality scores to generate a chart like the one in Figure 1.

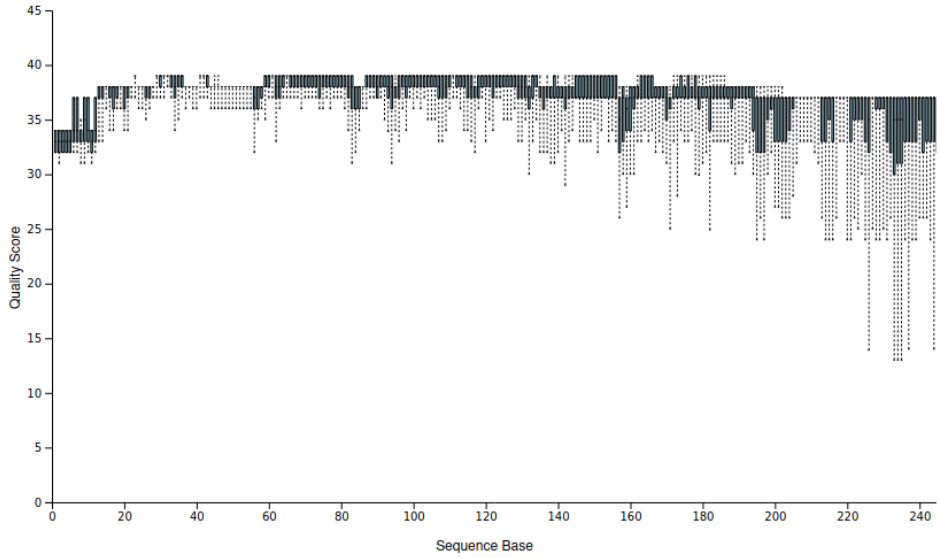

Figure 1: Sequence quality score box-and-whisker plots generated by q2-demux [6]

The researcher uses this chart to identify where quality scores begin to increase and decrease within the forward and reverse reads. The lower quality scores are a result of the 5' and 3' ends that are read in the first and last several cycles by the sequencer and should be removed to maintain quality sequence data. Once the researcher decides on *trim* and *trunc* parameter values, this information can be input into the Dada2 plugin to proceed with bioinformatic analysis.

Except for a few software solutions with weaknesses discussed in the introduction, the process of deciding *trim* and *trunc* values for sequence data is entirely manual and at the discretion of the researcher based upon experience. The *TruncTrimmer* plugin offers a revitalized way of automatically determining *trim* and *trunc* values to pass to Dada2 with the intent of providing a path to eventually automate all standard microbiome data analysis.

TruncTrimmer works by performing the same analysis a researcher would when deciding where to make *trim* and *trunc* cuts to sequences. In the lab where this work was performed, we try to maintain a quality score of 30 within our sequence data while primarily focusing on the bottom whisker and bottom box score values. We make the trims by determining where the quality scores have an initial increase in quality, signifying the 5' end of the sequence. Likewise, we make truncations based upon where the quality scores begin to dip below the desired value of 30, signifying the 3' end of the sequence.

Algorithm 1 TruncTrimmer Pseudocode

```
1: function TRUNCTRIMMER(S, ss, ws, tx, ty)        ▷ Where tx - trim threshold, ty - trunc threshold
2:     Qs = sample(S, ss)                          ▷ Where Qs - sample of qual scores, S - sequences, ss - sample size
3:     ws = Qs.length() * ws                       ▷ ws is window size and ws starts is originally a percentage
4:     trim = NULL
5:     trunc = NULL
6:     for i = 0 to (Qs.length() − ws) do
7:         wm = mean(Qs, i, i + ws)                ▷ Mean of quality score values within the window
8:         if trim is NULL then
9:             if wm > tx then
10:                if i = 0 then
11:                    for j = 0 to ws do
12:                        if Qs[j] > tx then
13:                            trim = j
14:                            break
15:                else
16:                    trim = i
17:        else
18:            if wm < ty then
19:                trunc = i + (ws/2)
20:                break
21:    if trunc is NULL then                       ▷ If trunc wasn't set by the end of window sliding we set it here.
22:        for k = 0 to ws do
23:            if Qs[Qs.length() − k] > ty then
24:                trunc = Qs.length() − k
25:                break
```

Figure 2: Basic TruncTrimmer Pseudocode

Using the same standards of a researcher, *TruncTrimmer* uses a sliding window algorithm to determine where the rise and falls of quality scores occur. The pseudocode for the algorithm is shown in Figure 2. The algorithm contains a series of parameters which are customizable, so other researchers can adjust the algorithm as necessary to better fit their lab's needs. Some of the customizable parameters include the quality score sample size, the window size of the sliding window, the *trim* threshold, and the *trunc* threshold. *TruncTrimmer* first extracts samples of quality scores for each read with a default sample size of 10,000 scores. Each sample is used to produce an array of values correlating to the values which would be displayed in a box-and-whisker plot. This is the same process used to generate the box-and-whisker chart by the summarize

function in the Qiime2 Demux plugin [6]. Next a window with the default size of 5% of the total raw sequence length is created. This window is then slid across the quality score values starting at position 0 looking for the average quality score value within the window to reach the threshold which defaults to 30. Once the *trim* threshold is met, the middle position within the window is marked as the *trim* location for those reads. The window then continues sliding across the sequence reads quality score values until the scores fall below the *trunc* threshold which defaults to 21. When the average quality score within the window no longer meets the truncate threshold, the position in the middle of the window is marked as the truncate value. If the average of the scores within the window never fall below the truncate threshold, the reads are iterated over in reverse order, starting at the last position, until a read is found that surpasses the *trunc* threshold. This location is then marked as the *trunc* location. When the first position is found surpassing the *trunc* threshold the position in front of it is marked as the *trunc* value. This process is repeated twice to find the *trim* and *trunc* values for both the forward and reverse reads of the sequences. This is essentially the same process that a researcher would conduct manually while moving the mouse over the box-and-whisker plot chart to determine *trim* and *trunc* values.

After determining where the trimming and truncating of the sequences should occur, *TruncTrimmer* then stores this information into a table-like format which can be stored in a Qiime2 artifact. By storing the information in this way, it will allow the researcher to manually review *trim* and *trunc* values if they desire. Additionally, it removes the need for duplicating raw sequences into trimmed sequences saving storage space for the researchers. It is not abnormal for sequence data to take up

gigabytes of data, so reducing duplicated sequence data makes a noticeable difference. Storing this information in a table also allows for this information to be used within the Qiime2 pipeline as an input. This is a key requirement so that this plugin can be used to further automate the standard operating procedures. The idea is that the file output by *TruncTrimmer* can be used as an input to Dada2 removing the need for the researcher to manually type in *trim* and *trunc* values.

TruncTrimmer is designed such that it can be used as soon as raw sequences are imported and demultiplexed as Qiime2 artifacts. *TruncTrimmer* takes the Qiime2 artifact containing the demultiplexed sequence data as input and outputs an additional Qiime2 artifact containing the table with the *trim* and *trunc* parameters in a format similar to the table in Figure 3.

# ID	trim	trunc
forward	1	250
reverse	1	198

Figure 3: TruncTrimmer example output table

From here the table will ideally be able to be used as input into the Qiime2 Dada2 plugin to specify the *trim/trunc* parameters then the standard bioinformatic analysis procedures using Qiime2 can proceed as normal.

Results

TruncTrimmer uses a sliding window algorithm to determine *trim* and *trunc* points based upon sequence quality scores, so researchers no longer have to manually complete this task. To analyze the effectiveness of *TruncTrimmer*, the *trim/trunc* selections made by an expert were compared to those made by *TruncTrimmer*. For this comparison, multiple quality score samples from three different datasets were used.

One dataset was a microbiome dataset that was used to develop an anaerobic in vitro turkey cecal model [7]. The other two datasets are microbiome sequence datasets being used in ongoing projects. Each of these datasets contained characteristics that were representative of different sequencing runs. The first dataset represented the ideal sequencing run, and it had very high and consistent quality scores from start to finish. The second dataset represented a common run where quality scores were generally high, but variance in quality scores based upon sample selection were common. The last dataset was from a sequencing run which resulted in below average, but usable, quality scores. For the last dataset, the quality scores from the reverse reads were low enough that the expert noted when scores from reverse reads are notably low it might be preferrable to omit the reverse reads from the analysis since analysis can still be performed accurately on forward reads alone.

In creating the comparison data, three sequence quality score box-and-whisker plots were developed for each dataset using the Qiime2 Demux plugin [6]. The expert's *trim/trunc* values for each of the datasets were averaged out to help account for variances due to different quality score sample selections. A similar process was performed to determine *trim/trunc* values output by *TruncTrimmer*. *TruncTrimmer* was

run three times for each of the datasets and the *trim/trunc* values were averaged out and compared with the expert results. The resulting average *trim/trunc* values for the three datasets can be seen in Figure 4.

Dataset	Expert				TruncTrimmer			
	Forward		Reverse		Forward		Reverse	
	Trim	Trunc	Trim	Trunc	Trim	Trunc	Trim	Trunc
Dataset 1	0	251	0	215	1	250	1	198
Dataset 2	0	243	0	233	1	237	1	223
Dataset 3	12	222	3	153	1	238	1	165

Figure 4: Expert and TruncTrimmer average trim/trunc values

Analyzing Figure 4, one can see that the average results produced by *TruncTrimmer* are comparable to those selected by the expert. The differences between the *trim*/*trunc* positions chosen by the expert versus the positions calculated by the *TruncTrimmer* algorithm are almost negligible considering the largest difference is on the order of 17 nucleotides (nts). The average standard deviation for the expert's *trim* and *trunc* values was 10.07, and the average standard deviation for *TruncTrimmer's trim* and *trunc* values was 0.20. In combination with the average standard deviations, average differences this small can likely be attributed to human error in selection and/or the differing quality samples used to determine *trim/trunc* values. It is not uncommon for some sequence runs to have a high variance and random drops in the quality scores on each read, so different samples can justify different *trim/trunc* locations. *Trim* and *trunc* locations are error tolerant since data cleaning and clustering techniques are used later in the bioinformatic process to organize sequences into operational taxonomic units (OTUs). A maximum average difference of 17 nts between the expert's selections and

TruncTrimmer's selections can be considered negligible because those 17 nts will likely not be needed to place a sequence into the correct OTU.

Since *TruncTrimmer* is able to reliably find *trim/trunc* values similar to those picked by an expert for the same datasets, it is reasonable to argue that *TruncTrimmer* should be incorporated into other labs' standard practices to save time. Other researchers can incorporate *TruncTrimmer* into their standard operation procedures and make parameter adjustments as necessary to fit the needs and preferences of their lab. Such parameter changes might be to quality score thresholds, window size, and sample size depending on what the researcher desires. The results produced here were generated using the *TruncTrimmer*'s default parameters and seem to produce reliable results.

Discussion

The initial motivation for developing the *TruncTrimmer* Qiime2 plugin was to remove the manual interaction required by the researcher to determine *trim* and *trunc* points within the sequence data. Many labs have a set of standard operating procedures that are followed for almost all projects. Researchers spend a large amount of time performing the same tasks repeatedly in the standard bioinformatic analysis portion of the project. For example, researchers run microbiome data from various projects through the same Qiime2 plugins the same way repeatedly. To increase the efficiency of analyzing the microbiome data, the idea was proposed to automate the standard Qiime2 procedures such that a researcher could preload a program with some initial project-related information then automatically run the data through the standard

Qiime2 pipeline. However, certain parts of the analysis require the researcher to observe the data in its current state to make decisions about how to proceed in the pipeline. One of major obstacles to overcome the need to the researcher's attention was the ability to automatically determine *trim* and *trunc* locations for the sequence data before it is run through the Dada2 plugin. A new Qiime2 plugin, *TruncTrimmer* is the proposed solution to this problem.

TruncTrimmer successfully automates the process of picking *trim* and *trunc* locations before cleaning sequence data with Dada2. The plugin makes decisions comparable to those of experienced researchers as shown in the results section. With the incorporation of *TruncTrimmer* into the lab's standard operating procedures, we are close to achieving a program that will fully automate standard bioinformatic analysis. While researchers will still be required to go back and make manual adjustments to the operations as needed, the overhead of the standard work will be minimized allowing researchers to spend more time on other tasks such as interpreting the meaning of the analyzed data. It is in every labs' interest to find ways to automate standard repetitive tasks. By automating such tasks, labs should be able to increase their work efficiency and output.

Conclusion

TruncTrimmer automates the task of selecting *trim* and *trunc* parameter values for input into the Dada2 Qiime2 plugin by using a sliding window algorithm to analyze sequence quality scores, and the results are comparable to those of seasoned researchers that manually choose the locations. With the incorporation of the

TruncTrimmer Qiime2 plugin into the standard pipeline, researchers will no longer have to manually generate and analyze quality score plots to determine where sequence data needs to trimmed and truncated. Automating this task removes one manual interaction required by the researcher when performing standard analysis thus making the entire process closer to full autonomy. The long-term goal is for researchers to incorporate this plugin into their standard pipelines and have their lab's standard Qiime2 microbiome bioinformatic analysis processes fully automated. This will reduce work overhead associated with running microbiome data through the Qiime2 pipeline allowing researchers to focus more time on other tasks.

Before the long-term goal of full autonomy can be achieved, there is still future work that needs to be completed. Currently, the Dada2 plugin is being modified to provide the option of passing the Qiime2 artifact output by the *TruncTrimmer* plugin in place of the manually input *trim* and *trunc* parameters. The *TruncTrimmer* plugin only supports and has been tested with paired-end sequence data at this time. Additionally, features such as the support for single-end sequence will need to be developed for the *TruncTrimmer* plugin, so other labs with different procedures can customize the plugin's functionality to meet their needs. With the motivation for developing the *TruncTrimmer* plugin being full automation for the standard bioinformatic analysis procedures, a future project is to complete the development of a set of scripts/programs which run sequence data from start to finish through the complete Qiime2 pipeline.

TruncTrimmer is only the first step towards automating standard processes related to the Qiime2 pipeline. More work is needed to achieve this goal, but

researchers can begin deploying this plugins functionality to their current processes now.

Acknowledgements: A portion of this work was supported by *Student Cross-Training Opportunities for Combining Food and Cybersecurity into an Academic Food Systems Education Program*, USDA National Institute of Food and Agriculture (NIFA), Higher Ed Challenge, Challenge Grants Program, under Grant Number 2018-70003-27663. The authors do not declare a conflict of interest.

Conflict of Interest: The authors do not declare a conflict of interest

References

[1] Kozich J.J., Westcott S.L., Baxter N.T. *et al.* Development of Dual-Index Sequencing Strategy and Curation for Analyzing Amplicon Sequence Data on the MiSeq Illumina Sequence Platform. *Applied and Environmental Microbiology* 79(17) 5112-5120 (2013). https://doi.org/ 10.1128/AEM.01043-13

[2] Bolyen, E., Rideout, J.R., Dillon, M.R. *et al*. Reproducible, interactive, scalable and extensible microbiome data science using QIIME 2. *Nature Biotechnology* **37**, 852–857 (2019). https://doi.org/10.1038/s41587-019-0209-9

[3] Callahan, B., McMurdie, P., Rosen, M. *et al.* DADA2: High-resolution sample inference from Illumina amplicon data. *Nat Methods* **13,** 581–583 (2016). https://doi.org/10.1038/nmeth.3869

[4] Marcel, M. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet journal* 17(1), 10-12 (2011). https://doi.org/10.14806/ej.17.1.200

[5] Joshi, N.A., Fass J.N. Sickle: A sliding-window, adaptive, quality-based trimming tool for FastQ files. Github repository (Version 1.33) (2011). https://github.com/najoshi/sickle

[6] Q2-D2, *et al*. q2-demux. Github repository (2017). https://github.com/qiime2/q2-demux

[7] Ricke, S., Feye, K.M, Dittoe, D.K. *et al*. Yeast fermentate-mediated reduction of Salmonella Reading and Typhimurium in an in vitro turkey cecal culture model. *Frontiers in Microbiology* (2021). doi: 10.3389/fmicb.2021.645301