Dear Prof. Benson,

Thank you for the opportunity to revise my manuscript “DNAvisualization.org: a serverless web tool for DNA sequence visualization”. I appreciate the reviewers’ insightful comments and timely responses.

I have carefully reviewed the concerns raised and have revised my manuscript accordingly. Following this letter are point-by-point responses to each of the reviewers’ comments.

I hope that this revised manuscript is now suitable for publication and look forward to hearing from you in due time.

Warm regards, Benjamin Lee, on behalf of all authors

Referee: 1

Comments for the Author

> Overall the web site works as described and I see no problems with it (using Chrome and Firefox on Mac OSX). The graphics are nice, the tools to inspect the graphs are functional. The tools are not original but this is not the point, these are nicely implemented and appropriately referenced to prior work, though the discussion of these could be expanded upon.
>
> A comparison to other methods should be expanded upon. Of the 17 references cited there is little discussion of their content. For instance Page 1 “a variety of methods have been proposed” (references 3-12) !! This really does little justice to the prior work or description thereof. Generally speaking if a reference is included then something substantial from that paper should be mentioned - if only a sentence for each.

All visualization methods referred to in the paper have now been described and compared to each other. This section now reads: "One common technique is to map each nucleotide to a vector and connect those vectors tip-to-tail to represent a DNA sequence. For example, the Gates method uses up, down, left, and right vectors of length one to represent Ts, As, Cs, and Gs, respectively, while the Yau method uses vectors along a unit circle to represent the bases. Others, such as Qi and its derivative Squiggle algorithm are based on mapping a binary representation of the sequence to upward- and downward-oriented vectors for 1s and 0s, respectively. In contrast, other algorithms such as @qiNovel2DGraphical2007 and @randicCompact2DGraphical2003 are based on tablature, with the x coordinate corresponding to base number and the y coordinate to a specific nucleotide or dinucleotide, respectively."

Intro Page 2

> It is stated that methods used are those for which “Visualizations which may be unambiguously transformed back into the DNA sequence” Following on from this when downloading a zoomed section of the graph (something of interest to the user) I would suggest also including the DNA sequence within the x-axis range with the bases colour coded to match the graph. (you could also think about aligning these DNA sequence but that might be too much to ask?).

This feature is not able to be implemented due to the lack of existing software capable of inverting DNA sequence transformations and the fact that raw DNA sequences are not stored, thereby necessitating transformation inversion. However, this is a valuable suggestion that will most certainly be implemented in a future version of this website.

Interface Page 2.

> Include a short description for each of the 5 visualization methods used /referenced rather than bunching all refs to a single sentence. Why were these chosen over some of the other methods referenced.

The reason for the inclusion of these specific methods has been clarified as being those implemented in the previously published `Squiggle` package:

> Just a thought - when looking at a zoomed region, it would be nice to jump to another visualization method of the same x-axis region. The zoom out function is nice.

This feature has been implemented.

Page 3 Implementation

> First sentence refers twice to serverless - it this redundant (minor edit)

The sentence now reads "The web tool is built using a novel architecture, with computing, as well as data storage and selective retrieval, done in an entirely serverless manner."

> The section on 'FaaS method reads a little like an advertisement for Amazon services. I think a less biased discussion of other providers of such methods should be included. “generous free tier” is too opinionated. ‘Lambda’s pricing is very affordable” ??? Need to be more objective and less biased in this regard. Are there other providers on which such an architecture could be run?

With respect to being more objective about the Lambda's free tier and pricing (as well as competitors), this paragraph has been rewritten to read: "At the time of this writing, there are a variety of serverless computing platforms including (but certainly not limited to) Amazon Web Services (AWS) Lambda, Google Cloud Functions, and Microsoft Azure Functions, each of which differ in terms of factors such as supported programming languages, startup latency, and pricing structure. DNAvisualization.org is built atop AWS Lambda due to its permanent free tier that, at the time of this writing, allows for one-million free function invocations totaling up to 3.2 million seconds of compute time per month, which is anticipated to easily meet the demand for the site. In the event that the free tier is exceeded, AWS Lambda's pricing is $0.20 per million function invocations and 1.667×10^{-5} dollars per GB-second of computation at the time of this writing." Additionally, the following sentence has been added about S3 pricing: "At the time of this writing, the price of storage in S3 is $0.023 per GB per month (for the first 50 TB of data), with S3 Select priced at $0.002 per GB of data scanned and $0.0007 per GB of data returned."

In regard to other providers on which this architecture would work, the following has been added to the discussion: "While this website was implemented using AWS, this architecture is not exclusive to AWS. Google Cloud Platform (GCP) offers both serverless computing and serverless data querying via their BigQuery platform. Although both GCP and AWS have similar offerings, it is nontrivial to change cloud service providers due to each service provider's use of a proprietary application programming interface (API). The issue of vendor lock-in via proprietary APIs is one of the open problems in serverless computing, although open-source tools such as the Serverless Framework (https://www.github.com/serverless/serverless) and Apache OpenWhisk (https://openwhisk.apache.org) show promise for ameliorating this issue."

Figure 2

> I would suggest it is “A diagram demonstrating…” rather than “A sequence diagram demonstrating…” The legend is not really sufficient to describe the Figure. For instance why do some arrows have x and others not? Some yellow boxes are aligned with vertical lines some are free standing - is this meaningful? Overlap of lilac (opt) over gray box - does that imply something? Some text is in [] brackets - what does this imply. The “initial sequence transformations” and “sequence querying” mentioned in legend are not clearly labeled in figure - which is which? I assume it’s the boxes marked as loop.

This figure has been completely remade for greater clarity. Instead of using a sequence diagram, this figure now features a graphical representation of the architecture with a much more robust legend, while capturing the same information as before. The “initial sequence transformations” and “sequence querying” mentioned in legend are now clearly labeled, along with each of the steps. Color-coded parallel lines are now used to demonstrate parallelism, rather than the previous loop boxes.

Page 7 Discussion

> When discussing the limitation of the architecture it would be good to discuss a clear-cut bioinformatic process or approach that would be problematic - if is this what you are implying? Again in the Conclusion reference is made to “not all applications are amenable” - this is too vague and an example discussed (in Discussion?) would help clear this up.

The following sentences have been added to the discussion to address this point: "An example of a web server that cannot trivially be ported to use an entirely serverless architecture is MISTIC2 [@MISTIC2], a tool for protein coevolution analysis, whose job durations can be as long as five hours (supplementary figure 1). This job length is significantly longer than any current serverless offering allows a single function invocation to run."

> Also I am unsure of how to “respect” a memory constraint. I would rephrase this statement.

This sentence has been reworded to read: "In addition, a function's memory use may not exceed a predefined limit, which can range from the scale of megabytes to several gigabytes and be specified by the user."

> With the Randic (Qi) dinucleotide approach. Would it be better to implement a codon (trinucleotide) approach with degeneracy (20 options rather than the 16 dinucleotides). From a users point of view this may be more biologically relevant.

Simply put, yes, it would be better to implement a codon approach with degeneracy that the Qi method. This is an insightful method that would likely capture more biological meaning. However, as per the NAR web server instructions ("new methods that have not been previously validated in a separate publication" are discouraged), no new novel visualization methods were introduced. With respect to the addition of novel methods, the website has a section ("Have an idea for another way to turn a DNA sequence into a two-dimensional visualization? Let us know over on the Squiggle repository and we'll be happy to work with you on implementing it") that refers users with novel ideas to propose it to the underlying transformation library atop which the web service is built for inclusion.

> Also in the Qi graph, it might be a good idea to label the consensus line explicitly rather than refer to it as one of the input sequence identifiers. E.g. with the test data the consensus (overlapping) orange line is labeled Chimpanzee.

With the Qi graph, there is no consensus line. That the topmost line is the only one visible is a pernicious example of overfitting. Hovering over a legend item results in that sequence being the overlapping (topmost) line. A clarification regarding this issue has been added to the instructions.

Referee: 2

Comments for the Author

> In "DNAvisualization.org: an entirely serverless web tool for DNA sequence visualization", the authors present an open-source website that displays uploaded DNA sequences using the previously published Squiggle program. The website is free to use and does not require registration.
>
> Uploaded DNA sequences are plotted in a 2-D space, with DNA sequence position along the X axis; the Y position for each DNA base is calculated according to the user's selection of algorithm. The main algorithm is Squiggle, which uses glyphs of different shapes for the DNA bases A, C, G and T. The glyphs are connected at their endpoints, with the result that the line representing a DNA sequence will tend to move upward in GC-rich sequence and downward in AT-rich sequence. Similar DNA sequences have similar patterns of upward and downward movement. Four older algorithms are also offered: Yau (unit vector glyphs), Yau-BP (vector glyphs normalized to constant x width), Randic (tablature-like), and Qi (tablature-like but for dinucleotide pairs).

> The website works as described. An example DNA sequence file containing the HBB gene sequence from several species is provided as an example input, which makes it very easy to try out the website without first having to find suitable input. The algorithm is selected before the FASTA sequences are uploaded; after the initial rendering, it is possible to zoom to a smaller region of the sequence, reset to the original range, or alter the image title and caption. It would be nice to also be able to change the algorithm after the initial rendering; as far as I can tell, the website must be reloaded or revisited in order to select a different algorithm.

For the initial submission, changing the visualization algorithm required removing all the sequences from the chart by using the trashcan icon and then reuploading the FASTA files or pasting the sequences. This approach is obviously less than ideal. Changing the algorithm after initial rendering poses a trilemma: store the uploaded raw FASTA sequence and transform on demand (slow due to the need to pull from S3 and wastes storage space if the user doesn't end up using the feature), performing the transformations preemptively without specification by the user (uses up to 5x more computing resources, which is wasted if not used by the user), or store the transformed sequence and transform on demand (most space efficient option but also insures the I/O penalty of pulling from S3 as well as the computational cost of inverting the transformation).

Thus, to change the algorithm after initial rendering, an alternate solution was found. Instead of the user selecting a single visualization method before uploading a FASTA file, the user is now able to select multiple visualization methods they are interested in. After uploading DNA sequence(s), the user may then click on a button to rerender the graph with an alternative visualization method.

> The github repository is well organized and the code is very readable.
>
> In my opinion, the title does not need to include the word "entirely".

The word "entirely" has been removed from the title.

> The manuscript is well written. It puts this work in context of other 2D graphical representations of DNA sequences; it would be better still if it compared 2D graphical representations to other more concrete representations such as sequence alignments, 2D similarity dotplots, and genome browsers which can display GC content, alignments, repetitive sequence annotations and many other types of features and data along the DNA sequence.

To better place this work in the context of browser-based DNA sequence exploration tools, the following section has been added to the introduction: "Numerous tools, ranging from genome browsers [@genomeBrowsers] to multiple sequence alignment viewers [@MSAViewer] and dot plot visualizers [@yassEnhancing] have been developed to enable interactive browser-based visualization of DNA sequences, alignments, and annotations. A different approach to addressing this problem is to convert DNA sequences directly into two-dimensional visualizations that capture some aspect of the biological information contained within, without relying on external information such as annotations."

> I spotted a couple typos:
>
> Abstract: "a variety publication-ready" -- probably missing an "of"

The abstract now reads "a variety of publication-ready".

> Implementation: "the amount of scanned" -- extra "of" or missing "data"?

The implementation now reads "the amount of data scanned during querying".

Referee: 3

Comments for the Author

> This work hosts a server less site to visualize large DNA sequences. Using FaaS framework in AWS, the site can query large sequences while zooming into the graph. While the work is interesting, The technicality can significantly be improved by accommodating the followings;

> 1. Is there any time/space complexity analysis for sequence and file level parallelization? As the former parses each sequences, so there should be trade-off of Memory/time complexities between file and sequence level parsing. As parse level sequencing seems the main contribution for limited amount of memory, any analytical insights need to be added here. comparing the technical benefits/ complexity analysis with other state of the art e.g., DNASonification.org would be great.

With respect to the time and space complexity of the different types of parallelism, the following has been added to the discussion: "While this tradeoff results in increased memory use by the client, which must load and parse the FASTA files in memory, and greater cost because pricing is by both the function invocation and the total amount of computation (which remains the same, as the total number of bases that must be transformed does not change), it enables greater throughput by more effectively leveraging parallelism."

> 2. the descriptions in "Implementation" section need to be modified, please refer AWS or FaaS framework. At a first glance, the description seems a new variant of FaaS framework is proposed and implemented for the purpose of this work.

Links to AWS Lambda, Google Cloud Functions and Microsoft Azure functions have been added to refer to the various platforms for serverless computing. In addition, links to the Serverless Framework and Apache OpenWhisk have been added to refer to open-source serverless frameworks.

> 3. The boxes for client, AWS Lambda and AWS S3 at the below of Figure 2 can be removed as they are already on top.

In response to the points raised by the first reviewer, Figure 2 has now been completely remade. As such, there are no more boxes on the bottom (or the top) of the figure.

> The followings are related to 'usability test' for the site; Most of the functionalities work except the followings;
>
> 1. drop anywhere on page does not work

Without knowing the specific details of the browser version, this bug is not able to be reproduced. Drag-and-drop is powered by FileReader.js, which has full support for IE, Chrome, Firefox, and Opera as well as support for Safari 10+ (as per https://developer.mozilla.org/en-US/docs/Web/API/File). That drag-and-drop does not work but the website as a whole does work suggests that the issue is not specifically with the File API, but more information would be needed to fully isolate and fix this issue. As a stopgap measure, feature detection for the File API has been implemented, which alerts the user if their browser is incompatible with the site and provides a list of browsers and versions which are compatible.

> 2. plot option does not vanishes automatically, if I want to change title and subtitle of the same figure twice or thrice and so on. It vanishes only for the first time.

This bug has been fixed.

> 3. Sequence can't be tracked and modified while I want to see the visualization of modified PASTA sequence using "Paste FASTA function"

The ability to modify pasted FASTA sequences and descriptions has been implemented.

> 4. In Safari, it does not work. While click on "Load an example file", the site freezes. Just shows the loading spinner

The website was, in fact, developed exclusively using MacOS Mojave v10.14.2 (presently using v10.14.3) and Safari v12.0.2 (presently using v12.0.3) and has had the "load an example file" functionality verified. Further information, such as the version of MacOS and Safari in use as well as any errors printed to the Javascript console, would be helpful to fix this issue. If you are comfortable using GitHub (through an anonymous account), filing an issue would allow for more discussion, as the bug is presently not reproducible.

> 5. Integrating two default sample files - one for small (already available), the other for large FASTA sequences - may be a good idea.

Three sample files of varying sizes are now available for testing and download: HBB (already available), human titin (as an example of a very large gene), and *Bacillus subtilis* (as example of a model bacterial genome).