Mito-AssemblyVis: Mitochondrial Genome Assembly Assessment Visualization

Armaghan Savar and Cecilia Yang

Abstract—Mitochondrial genomes (mitogenomes) are short and circular genomes commonly used in the development of DNA barcodes for environmental surveys, forensics, and anthropology studies. Advances in DNA sequencing technologies make it possible to assemble mitogenomes on a large scale, and thus, several mitogenome pipelines have been developed to facilitate mitogenome reconstructions from DNA reads. To choose high-quality assembled sequences, parameter tuning and assembly quality assessments are essential. Given there are several assessment tasks needed for assembly evaluations, we developed a multi-purpose website application for users to generate visualizations for their own curated datasets using three visual encodings: parallel coordinate plot, circos-based consistency plot, and parallel annotation plot.

Index Terms—DNA Sequencing, Genome Assembly, Mitochondrial Genome

1 Introduction

Genomes are DNA or RNA molecules that consist of nucleic acids (nucleotides). The nucleotide sequence contains valuable information that directly or indirectly mediates biological activities (e.g., protein productions/interactions, cellular responses) within organisms. Shotgun genome sequencing is a process of determining the genome sequence of an organism by randomly fragmenting the original genome sequence into small pieces of nucleic acid fragments called DNA reads. *de novo* assembly is a genome reconstruction process independent of a reference sequence. In order to reconstruct the original sequence *de novo*, a genome assembler is used to computationally join the DNA reads based on their overlapping regions [5].

Mitochondrial genomes (mitogenomes) are short and circular DNAs in cellular organelles called mitochondria. Mitogenomes are significantly more abundant than nuclear genomes per cell and are less prone to degradation thereby allowing forensic and environmental scientists as well as anthropologists to obtain DNAs from samples that consist of little biological material [2].

In the past, the high sequencing cost and the inaccessibility of DNA sequencing technologies led to a shortage of complete mitogenomes of some culturally and environmentally important animal species in public databases such as the National Center for Biotechnology Information (NCBI). Genome sequencing technologies have become much cheaper and more easily accessible, making it possible to assemble mitogenomes on a large scale. State-of-the-art mitogenome assembly pipelines such as GetOrganelle [6] and MitoZ [8] are often used for mitogenome reconstructions. A more recent mitogenome assembler tool called mtGasp has been developed by Cecilia Yang and other members of the Bioinformatics Technology Lab. To compare the mitogenome assembly results from different pipelines and ensure the accuracy of the assembled mitochondrial DNA sequences, conducting effective quality assessments of mitogenome assemblies is very crucial in the field.

Commonly used assembly quality assessment metrics include sequence length, genome coverage, number of gaps and misassemblies, mapped regions between two or more sequences, and genome annotation results, i.e., the total number of annotated genes, the orientation, and the location of the genes. Mitogenomes are approximately 16,000-20,000 base-pair long, so sequence length helps evaluate the completeness of the assembled sequence. Coverage indicates the number of DNA reads mapped to a given nucleotide in a reconstructed sequence [12]. Higher genome coverage is often associated with better genome assemblies, meaning the resulting reconstructed genome is close to its original form. The number of gaps and misassemblies reflect the overall contiguity and accuracy of the assembled genome. Genome annotation is a method that determines the location, orientation, and identity of specific genes. Potential genomic rearrangements and the completeness of assembled mitogenomes can be investigated by visualizing the mapped regions between the reconstructed and the reference genome and comparing their annotation information.

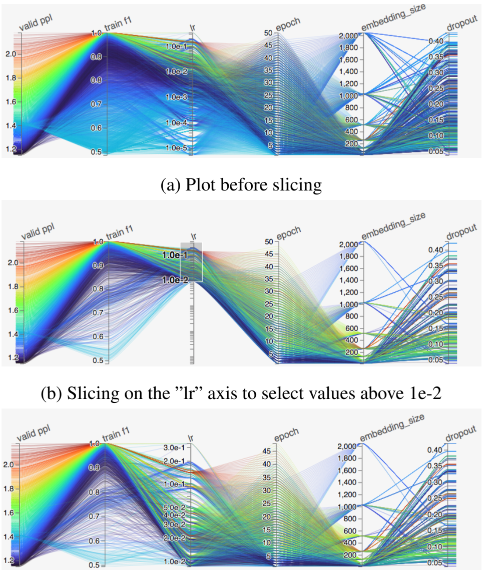
Visualizations are crucial in genome assembly quality assessments and effective communication of the results. Hence, we propose a web-based visualization interface that implements the published visualization packages/tools in the back end and allows users to upload their genome assembly data and visualize the genome assembly assessment results based on their specified design choices, which can be circos-based genome consistency plot, parallel coordinate plot, or annotation information plot. This proposed web interface not only offers a one-click solution to visualize the mitogenome assembly results but also facilitates the comparisons between the performance of different assembly pipelines. In this work, the assembly results of the aforementioned novel pipeline, mtGasp, and two other state-of-the-art assembly pipelines will be used to generate our visualization prototype.

2 Related Work

2.1 HiPlot Hyperparameter Tuning

HiPlot is an interactive visualization tool developed by Facebook, allowing users to create parallel coordinate plots to navigate through high-dimensional datasets [11]. This tool was initially designed for machine learning (ML) hyperparameter tuning visualization to help the field researchers view a large number of data points from multiple experiments simultaneously (Figure 1) in order to evaluate the performance of different hyperparameters such as learning rate or the number of epochs. This tool takes input data in *csv* format and generates a multi-dimensional visualization of different attributes. The interactive slicing feature of HiPlot enables efficient selection and filtering of the data subsets, allowing users to easily analyze the patterns within the selected set.

A genome assembly software may yield different outputs depending on the chosen set of parameters. In order to generate the desired reconstructed genome sequences, conducting different parameter sweeps is required for fine-tuning parameter values. When multiple assembly experiments are conducted, HiPlot can be utilized to display the quality metric values along with the corresponding parameters used, allowing bioinformaticians to visualize and efficiently select the optimal parameter(s) for mitogenome assembly.



(a) Plot before slicing

(b) Slicing on the ”lr” axis to select values above 1e-2

(c) Plot after slicing

Fig. 1: Demonstration of the slicing feature of HiPlot for Machine Learning (ML) Hyperparameter Tuning [11]. (a): The original parallel coordinate plot consists of 7000 experiment data points. Hyperparameters include ”valid ppl”, ”lr”, ”epoch”, ”embedding size” and ”dropout”. ”train f1” is the evaluation metric that reflects the accuracy of the ML model’s prediction. (b): Datapoints with a ”lr” above 1e-2 are selected through the slicing feature of HiPlot. (c): After slicing through the ”lr” axis, only the selected data points are shown

2.2 Icarus

Icarus is a visualization browser for contig-focused assembly evaluations. It can be used to display the output contigs generated by various assembler tools and their corresponding assembly metrics (e.g., NG50). A sliding window can be applied to select a genomic region of interest, allowing an in-depth assessment of assembly quality.

As shown in Figure 2, contigs are shown as boxes with varying widths which correspond to the contig sizes. Genome contiguity statistics including N50, NG50 and NG75 can be easily determined from the panel. Correct, misassembled and unaligned contigs can be differentiated by hue(green, red and grey, respectively). The zoomed-in version of this plot on the top of the panel allows users to explore smaller contigs that may be hidden in the zoomed-out format (bottom panel). This visualizer is applicable when multiple contigs are generated by the genome assembler. However, mitogenome assembly pipelines do not yield more than one DNA contig in most cases, so this visualizer has not been chosen for Mito-AssemblyVis.

2.3 Circos Plot

Circos is a Perl-based visualization software that provides a circular layout of genomics data. This tool is intended to visualize large-scale translocations or misassemblies in draft assemblies, but it can also be useful when trying to show structural agreements between different genomes. Given only a reference genome sequence and a pipeline-assembled sequence, this analysis is helpful for getting a quick qualitative view of the misassemblies in a genome assembly. One possible drawback of this method is that small and detailed misassemblies, possibly mediated by repeats, might not be visible. [7]

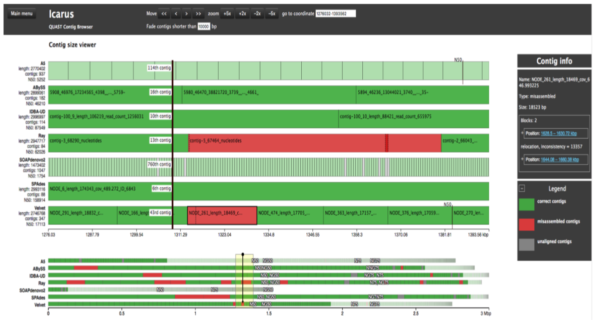


Fig. 2: Icarus Contig Size Viewer [9]: A published visualizer for Assembler Software Evaluation. Contig sets (shown as rectangular blocks) generated by different assemblers (e.g., ABySS, SPAdes) are shown. The plot at the top is the zoomed-in version of the bottom plot which shows the genome contiguity metrics such as N50 and NG50. Three hues are used to differentiate contigs. Green, red, and grey blocks represent correct contigs, misassembled contigs, and unaligned contigs, respectively. This visualization facilitates the performance evaluation of the built-in assemblers in the three assembly pipelines of interest.

2.4 Mitogenome Annotation Visualizer: MitoZ-Visualize

MitoZ-Visualize is a sub-feature of the MitoZ [8] which serves as a toolkit for animal mitogenome assembly, annotation, and visualization. When the “visualize” subcommand is activated, MitoZ produces a circular genome annotation plot displaying gene names, locations, orientations, GC (guanine-cytosine) content, and genome coverage (Figure 3). However, this tool only supports the visualization of one mitogenome sequence and does not enable the visual comparisons of multiple pipeline-assembled sequences with the ground-truth reference sequence.

3 Data and Task Abstraction

3.1 Task

Bioinformaticians run assembly pipelines to reconstruct mitogenome sequences from DNA sequencing reads. The resulting DNA assembled sequences (or contigs) and their quality metrics can be analyzed in order to determine the optimal experimental set-ups, e.g., pipeline(s) and set of parameters to use, for the mitogenome assembly pipelines. To assess the assembly qualities across multiple assemblies and determine the parameter specification that yields the optimal quality statistics and an assembled sequence with high similarity to the reference sequence, several visual encodings need to be established such that each of them supports a specific task in the multi-level assessment of the mitogenome assemblies.

Running assembly experiments computationally with different parameter settings would yield DNA sequences with varying quality levels and genomic properties. At the *de novo* assembly level, bioinformaticians often need to fine-tune the assembly parameters to get the best-assembled genomic sequence based on their pre-determined evaluation criteria. For the assessment of mitogenome assemblies, commonly used evaluation metrics are the number of gaps and misassemblies, the number of reconstructed sequences found in the assembly output file, the length of the reconstructed sequence(s), and the genome fraction. The key parameter to tune up is the *k*-mer size. Once the quality metrics are calculated and stored in tabular datasets, researchers would want to visualize how parameter setting affects the assembly results by analyzing the correlation between the parameter settings and the outputs. Therefore, the visual encoding for the assembly-level assessment needs to display all of the quality metrics from multiple experiments on the same plot and avoid overlapping marks. Since there are several existing mitogenome assembly pipelines used by bioinformaticians, researchers may also want to compare the assembly results of different pipelines. After determining the most acceptable parameters for each pipeline

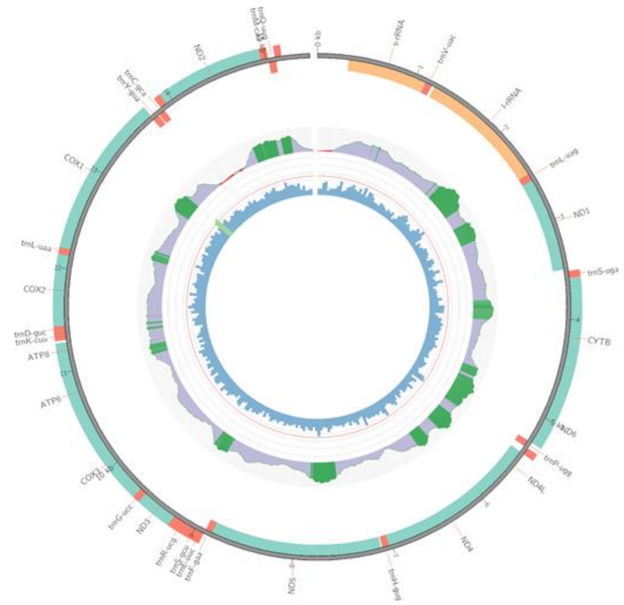


Fig. 3: MitoZ Mitogenome Annotation Visualization [8]: Demonstration of the MitoZ visualizer. The outermost circle consists of the annotated genes and the inner two layers display the GC-content and genome coverage distribution

from the assembly-level task depending on the target quality metrics, the sequence assembled using the chosen parameter setting will be used to compare with the "best" sequences assembled by other pipelines. A reference sequence will be chosen as a ground truth for quality comparisons. To compare the pipeline-generated sequences with the reference, researchers would need to analyze the mapped regions between an assembled sequence and a reference, how identical the reconstructed sequence is compared to the ground truth reference genomic sequence, and whether a complete set of genes with the desired characteristics, e.g., location, size and strand orientation of the genes, similar to the annotated genes of the reference can be found on the reconstructed sequence(s). If there are genes found missing from the reconstructed mitogenomes when compared to a reference, a researcher would need to consider the sequencing depth (or genome coverage) levels at different base pair (or nucleotide) positions which gives more insight into how accurate the assembly is since higher coverage means the pipeline is more confident with the assembly output in a specific region.

3.2 Data

The target animal for this prototype work is the southern sea otter which is a culturally and environmentally important animal species in Canada. The raw DNA read data of a southern sea otter sample (SRR8597300) is downloaded from NCBI, the DNA reads are assembled into mitogenome sequences using three different mitochondrial genome assembly pipelines GetOrganelle [6], MitoZ [8] and the novel in-house pipeline called mtGasp using 3 different *k*-mer settings (*k*= 61, 81, 91). The resulting reconstructed sequences are stored in 3 different *fasta* files. The *fasta* file format is a commonly used text-based format to store sequence descriptions, also known as sequence headers, and nucleic or amino acid sequence data. The quality metrics of the assembly output sequences, e.g., number of gaps, number and types of misassemblies, sequence length, and genome fraction are calculated using Quast [4] and stored in a tabular dataset. While the number of gaps and misassemblies are included in the input tabular dataset, they were not presented in the resulting parallel coordinate visualization because the line marks would overlap with each other at these two attribute axes if included. A reference sequence with the accession number NC009692.1 is downloaded from NCBI and used as the ground truth for genome similarity and gene annotation comparisons. Sequence similarities and mapped regions between the query sequences and the reference are determined automatically after submitting the fast-formatted sequences to the GGisy-based pipeline [13]. The locations, strands, and names of the mitogenome genes are determined by MitoS [3] and stored in annotation files with *fas* format. Genome coverage levels are determined by mapping the DNA sequencing reads to the reconstructed mitogenomes and calculating the total number of reads mapped to each nucleotide position and stored in text *txt* files. A summary of the attributes discussed above can be found in Table 1.

4 Solution

We have created a multi-view interactive visualization tool that implements three types of visual encodings: 1) Parallel coordinate plot for hyperparameter tuning, 2) GGisy-based consistency plot for mapped region visualization, and 3) Parallel genome annotation information visualization, to support the three levels of mitogenome assembly assessments discussed in the task abstraction section.

4.1 Parallel Coordinate Plot for Hyperparameter Tuning

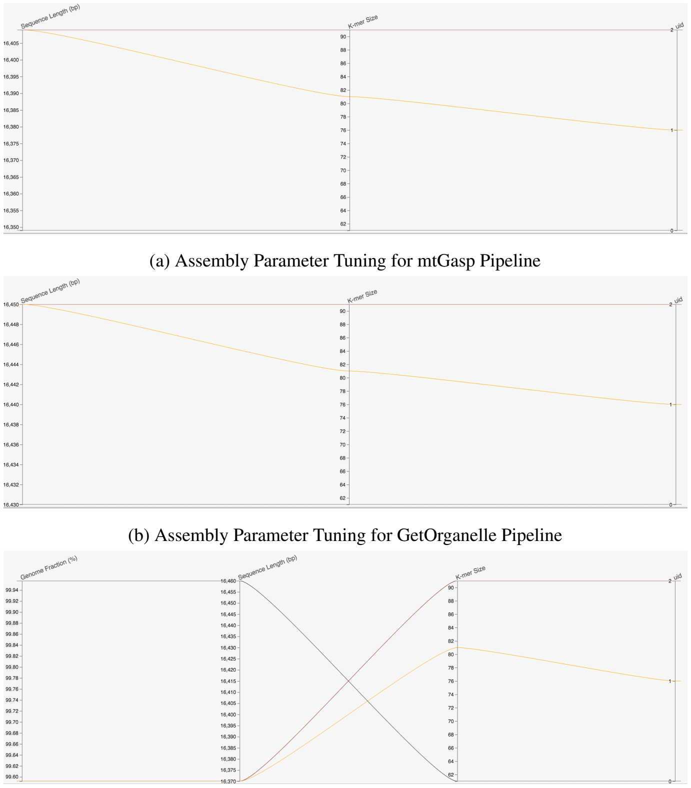
HiPlot (Figure 4) is an interactive visualization tool that, given experimental data points, helps us understand which parameters influence the metrics we want to optimize. However, this tool has never been used in the task of genome assembly parameter optimization. The HiPlot tool can be used to illustrate the relationships between different pipeline parameters, such as the k-mer sizes (61, 81, 91), and the final assembly results (e.g., sequence length, genome coverage) using parallel coordinate plots. This multi-dimensional data visual will help bioinformaticians to tune up the mitogenome assembly pipeline to obtain optimal results using the best assembly parameters. The parameter tuning results for 3 pipelines (mtGasp, GetOrganelle and MitoZ) are shown in Figure 4, some attributes (number of gaps, number of misassemblies, number of sequences) in Table 1 are not shown in the parallel coordinates plots, this is because HiPlot automatically filters out the attributes (e.g., Average Gap Size, Number of Gaps) that do not have unique values to avoid overlapping line marks thereby allowing users to pay more attention to the key metrics that help select the optimal mitogenome assembly resulted from the 3 k-mer settings (61, 81, 91). Hiplot interactive paralelle coordinate plot was chosen because it uses line marks to reveal the relationships between multiple attributes and its interactive slicing feature and automatic filtering of the overlapping line marks enables users to easily observe the patterns across multiple experiments and determine the optimal parameter for the target pipeline.

4.2 GGisy-based Consistency Plot for Mapped Region Visualization

The GGisy-based genome circos plot is used to visualize the high-level structural agreements between the pipeline-assembled mitogenomes and the reference sequence as well as the similarity levels. In Figure 5, the resulting GGisy-based plots show the mapped genomic regions connected by the red ribbon between three pipeline-reconstructed sequences (MitoZ, mtGasp, and GetOrganelle-assembled sequences) and the aforementioned reference sequence. The query sequences are shown on the left, and the reference can be seen on the right of each plot. Moreover, The color saturation reflects the identity levels between the two sequences. Unmapped regions, i.e., regions where the sequence similarity is below 50 percent are shown as blank areas within the circular area. The high color contrast between the ribbon color and the unmapped regions allows users to easily determine the structural agreements between the two sequences. The color saturation of the ribbon helps users to identify the similarity levels. Since only one mitogenome sequence was generated in each of the mitogenome assemblies we ran for this DNA sequencing read sample, the saturation does not serve the purpose of differentiating multiple assembly sequences but it would be useful when more than one mitogenome sequences are generated by the pipeline. Compared to the linear-view sequence similarity comparisons, the circular view is more effective at revealing the size differences in mapped vs. unmapped regions. In addition, circos plots are widely used in bioinformatics literature; hence, many end-users have already estab-

Table 1: Summary of Attributes

Variable	Type	Description	Possible Values
K-mer Size	Quantitative	K-mer size used in de novo assembly	3 possible values, k=61, 81, 91
Sequence Length	Quantitative	Length of assembled sequence	1-length of the reconstructed mitogenome sequence (bp)
Number of Gaps	Quantitative	Type of assembly quality metric	0-length of the reconstructed mitogenome sequence (bp)
Number of Misassemblies	Quantitative	Type of assembly quality metric	0-length of the reconstructed mitogenome sequence (bp)
Number of Sequences	Quantitative	Type of assembly quality metric	0-infinity
Sequence Type	Categorical	Type of input sequence	4 possible values: sequences generated by 3 pipelines, reference sequence
Gene Positions	Quantitative	Start (or end) position for an annotated gene	1-length of the reconstructed mitogenome sequence
Positions of Mapped Regions	Quantitative	Start (or end) position for a mapped region between assembly and reference	1-length of the reconstructed mitogenome sequence
Identity Percent	Quantitative	The level of similarity between the query sequence and reference sequence	0-100
Genome Coverage	Quantitative	Type of assembly quality metric	0-infinity



(a) Assembly Parameter Tuning for mtGasp Pipeline

(b) Assembly Parameter Tuning for GetOrganelle Pipeline

(c) Assembly Parameter Tuning for MitoZ Pipeline

Fig. 4: Hiplot: Multi-dimensional Visualization of the Quality Metrics from the assemblies generated by various sets of k-mer sizes. Please note that HiPlot only displays the attributes (e.g., sequence length, genome coverage) that have a distinct value from each k-mer assembly run to avoid overlapping line marks at the same attribute axis. As a result, a few attributes listed in Table 1 are omitted in the final HiPlot visualizations. (A): The result for mtGasp pipeline. (B): The result for GetOrganelle pipeline. (C): The result for MitoZ pipeline.

lished skills for analyzing sequence agreements in such circular-view visualizations.

4.3 Parallel Genome Annotation Information Visualization

The parallel genome annotation visualization is implemented to enable multi-sequence comparisons of the gene annotations and the visualization of the genome coverage distributions. As shown in Figure 6, four annotation tracks (results from GetOrganelle, MitoZ, mtGasp, and the reference) and the three corresponding genome coverage tracks are used to illustrate the genome annotation results along with the confidence levels reflected by the genome coverage distributions. In each annotation track, each arrow-shaped mark represents an annotated gene, and the width of the arrow corresponds to the gene size (bp). The direction and color of the arrow indicate the gene strand orientation, and the thin lines connecting the arrow boxes represent the DNA regions that do not contain annotated genes. Also, the x-axis scale (0-16kb) reveals the overall length of the mitogenome, and the genome coverage level distributions across nucleotide positions are color encoded in blue. This parallel linear view is chosen to facilitate visualizations of the gene properties, which are the location, orientation, and size of the genes, and easy comparisons across multiple sequences. This linear view is favored over the previously proposed circular view (Figure 7) because the linear sequences can be easily aligned in parallel to the same scale axis, making it easier to determine gene sizes and locations according to the genome-scale axis at the bottom of the viewer.

5 Implementation and Milestones

The first phase of the project, i.e., the generation of the input data consists of the following steps: 1) Extracting target species sequencing reads from possible databases, 2) Generating the assemblies using the aforementioned pipelines, and 3) Parsing the resulting assembly outputs and the data annotation files. The Python programming language and the Bash Unix shell and command language are used for the above steps. As for the visualizations, we have implemented Mito-AssemblyVis using Python libraries HiPlot [11], GGisy [13] and pyGenomeViz [10]. Back-end scripts for the integration of the three visual encodings are written in Python, and front-end web pages are constructed using HTML, CSS, and Javascript. Lastly, the Python-based web development package Flask [1] is used to generate the final website application for user interaction and display the final results. The estimated and actual hours for the milestones of the project are outlined in Table 2.

6 Results

The objective of this work was to create a multi-purpose visualization tool that supports different stages of assembly quality assessment including assembly software fine-tuning, sequence similarity analysis, and annotation visualization. The final web application tool successfully generates visualizations displaying the assembly results using three idioms (parallel coordinate plot, circos plot, and parallel annotation plot). Mito-AssemblyVis has 3 types of interfaces including a front page (Figure 8a), data upload pages (Figure 8b,c,d), and results display pages (Figures 4, 5, 6). Users may click on one of the visualization idioms from the front page depending on the type of input data and

Table 2: Expected and actual completion dates and required time for project tasks.

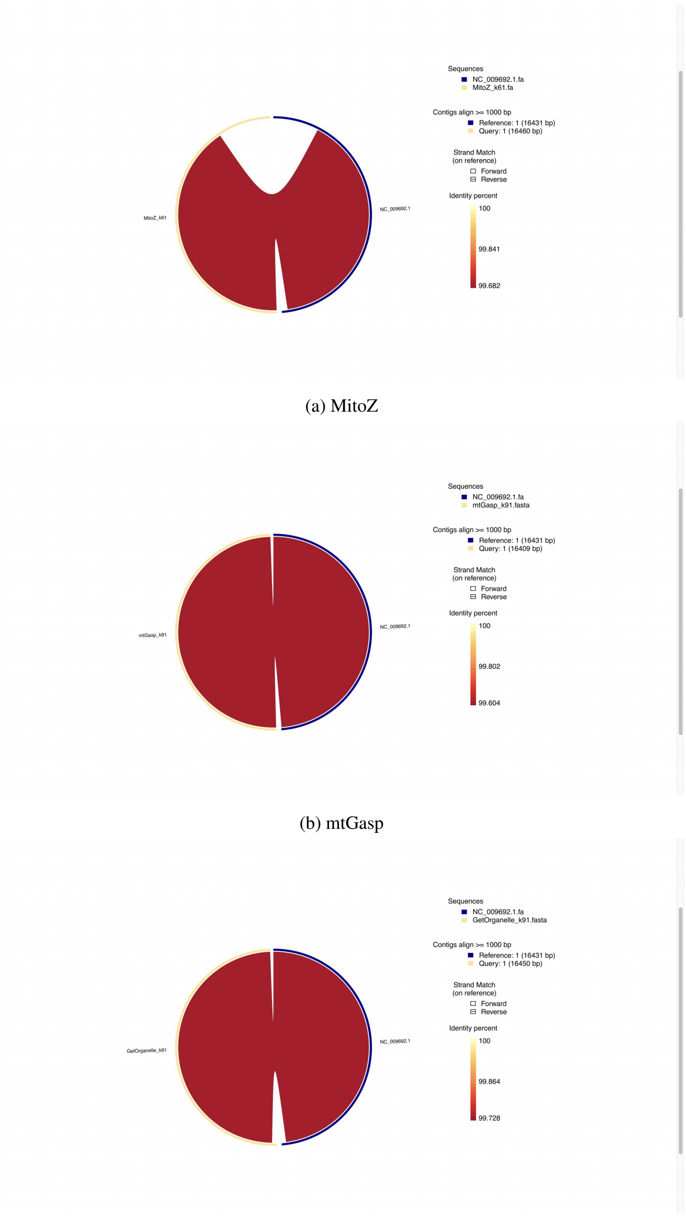
Task	**Est. Completion Date**	**Actual Completion Date**	**Est. Time (hr/person)**	**Actual Time (hr/person)**	**Assignments**
Project Pitch	Sep 28, 2022	Sep 28, 2022	3	5	Armaghan and Cecilia
Preparing the Annotation Results of One Species Sample Assembly (Sea Otter)	Oct 3, 2022	Oct 2, 2022	1	2	Cecilia
Parsing the Sea Otter Annotation Results	Oct 7, 2022	Oct 6, 2022	2	4	Armaghan
Pre-Proposal Meeting	Oct 12, 2022	Oct 12, 2022	-	-	Armaghan and Cecilia
Running the MitoZ and GetOrganelle Assembly Tools	Oct 14, 2022	Oct 15, 2022	8	10	Armaghan and Cecilia
Completion of the Project Proposal	Oct 21, 2022	Oct 21, 2022	5	5	Armaghan and Cecilia
Completion of Assembly Data Generation and Having Installed the Needed Packages	Oct 31, 2022	November 2, 2022	20	25	Armaghan and Cecilia
Having the Initial Version of Visualizations Ready	Nov 8, 2022	Nov 10, 2022	20	20	Armaghan and Cecilia
Written Updates Based on the Improvements and Changes in the Project	Nov 15, 2022	Nov 15, 2022	6	6	Armaghan and Cecilia
Peer Project Reviews	Nov 16, 2022	Nov 16, 2022	4	4	Armaghan and Cecilia
Finalize the Gene Annotation Plot	Nov 22, 2022	Nov 29, 2022	20	15	Armaghan and Cecilia
Post-Update Meeting	Nov 23, 2022	Nov 23, 2022	-	-	Armaghan and Cecilia
Finalize the FrontEnd Code (HTML + CSS)	Nov 30, 2022	Dec 11, 2022	20	15	Cecilia
Finalize the BackEnd Code (Python + Flask)	Dec 10, 2022	Dec 11, 2022	20	25	Armaghan
Finalize the Visualizations, Implementations and Analysis	Dec 10, 2022	Dec 11, 2022	5	6	Armaghan and Cecilia
Finalize the Project Presentation	Dec 14, 2022	Dec 13, 2022	7	10	Armaghan and Cecilia
Finish and Submit the Final Paper	Dec 16, 2022	Dec 16, 2022	8	11	Armaghan and Cecilia

specific tasks users have. Researchers may be interested in finding out the optimal set of parameters used in their target assembly pipelines that lead to the best result. Here, the HiPlot tool can be utilized. After running their assembly pipelines with the hyper-parameters to be evaluated, the user will generate and upload a tabular dataset in *csv* format that summarizes the parameters and the quality metrics to the HiPlot data upload interface (Figure 8b). For example, considering mitogenome assemblies, most often, sequence length along with the number of misassemblies or gaps are evaluated. For our novel assembly pipeline, hyper-parameters such as different *k*-mer sizes help users to select the optimal assembly based on the above quality metrics. Depending on the assembly pipeline, the target hyper-parameters can change. Next, imagine a bioinformatician or clinical user has run an assembly pipeline followed by an annotation step. Now, they have access to the gene positions and the corresponding strand orientations on the genome. If they are interested to compare the annotation results with those of other assembly pipelines, pyGenomeViz annotation, our proposed linear-view multi-assembly annotation visualization can become handy. By uploading the output of the different annotation information files in *fas* formats containing the name of the genes, starting positions, ending positions, and the strands to the pyGenomeViz data uploading interface, the users will be able to visually compare the extracted information for the target assembly tools of interest or only a single one of them. The total number of pipeline-assembled sequences to compare can be specified in the data input box (Figure 8d), and up to five sequences excluding the reference can be uploaded all at once. This parallel genome annotation visualization also has interactive features that allow users to modify colors, zoom in/out for in-depth navigation as well as see the hover effects showing the exact start and end positions of the annotated genes and their orientations (Figure 6). Moreover, researchers may need to visually compare the assembled genomes with a reference to search for highly similar regions. Hence, they can upload two *fasta* files, one containing the mitogenome assembly sequence, and the other storing the reference sequence. A circular plot with a ribbon linking the mapped regions will be shown, and the color saturation of the ribbon will indicate the similarity percentage (Figure 5).

7 Discussion and Future Work

The HiPlot input dataset we curated for each pipeline only includes three experiments resulting from the *k*-mer settings (*k*=61, 81, 91) due to time constraints. The y-axes of a few attributes were automatically omitted by HiPlot to avoid displaying overlapped line marks at the y-axis coordinates. This automatic filtering may be useful for dimensionality reduction when there are a large number of data points (e.g., Figure 1), but it may lead to the loss of important attributes when the dataset is small. For GGisy-based circos plot, as there is only one sequence yielded from each assembly run, the left side of the circos plot contains one contig sequence hence a single ribbon is shown in the resulting circos visualizations (Figure 5). In future studies, we intend to explore the visualizations for multi-contig visualization using GGisy circos plot and show more details such as small misassemblies, possibly mediated by repeats. In addition, a color gradient from low saturation to high saturation would be better suited to reflect the low to high identity levels in the GGisy circos plot. When implementing the parallel annotation plot using the pyGenomeViz package [10], there are a few technical limitations we fail to overcome. Although the final visualization has some interactive features (e.g., color modifications, hover effects, zoom-in/zoom-out), the mouseover texts cannot be manually modified because they are hard-coded in the pyGenomeViz package. Currently, the mouseover text when hovering over the genes (shown as arrows) only displays the gene orientation and the gene start/end positions but does not include the gene names. To label the genes with their corresponding names, the name texts are added to the gene annotation tracks with a rotation of 45 degrees and a very small font size. In order to determine the gene identities, users would have to zoom in on certain regions of the gene annotation track to read the name labels. If the user intends to save the final visualization as a static graph (.png or .svg files), the gene labels will be too small to read. After talking to a domain expert, a potential solution for this is to color-code the genes using different hues and include a legend for the gene color coding on the right side of the figure. As the current version of pyGenomeViz does not allow users to add customized legends, we will explore other annotation visualization options for more customizable features. An intrinsic limitation the parallel annotation visual encoding has is poor scalability, the genome annotation results can be difficult to compare when there are more than 5 sequences displayed in the same channel. Therefore, other visualization methods need to be considered for large-scale annotation data. Additionally, we weren't able to add y-axis and x-axis ticks, and labels to the genome coverage distributions. As a result, the exact genome coverage value cannot be determined from this plot, only the overall distribution patterns can be observed.

Although this web-based application makes it less difficult to upload datasets and create visualizations without excessive coding, it may not be ideal when working with larger genomic datasets. A desktop application may be a better solution than a web interface. This proposed tool has only been tested on one set of assembly data generated from a single whole genome sequencing (WGS) library due to limited time. In the future, the generalizability of this tool will be investigated by using different sets of mitogenome results from other WGS libraries.

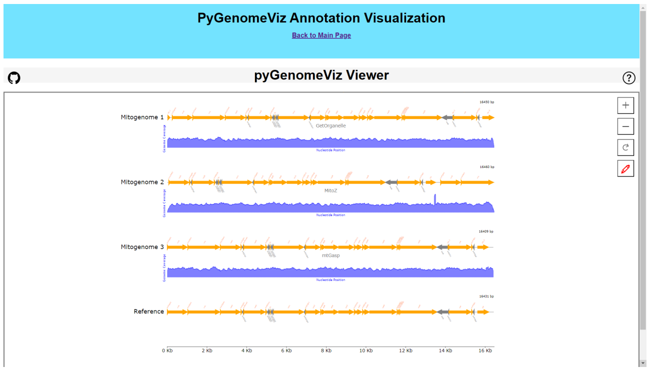


(a) MitoZ

(b) mtGasp

(c) GetOrganelle

Fig. 5: GGisy-Based Genome Assembly Circos Plot: Exploration of the relationships between reference genome sequence (right) and pipeline-reconstructed sequence (left). Regions with at least 50 percent similarity are highlighted by the wide ribbons within the circles. The color saturation reflects the similarity levels. (A): The result for the MitoZ pipeline, (B): The result for the novel mtGasp pipeline, (C): The result for the novel GetOrganelle pipeline.


PyGenomeViz Annotation Visualization

Back to Main Page

pyGenomeViz Viewer

Fig. 6: Parallel Mitogenome Annotation Visualization: This plot includes 4 genome annotations tracks (3 pipeline-assembled sequences and 1 reference sequence) and 3 genome coverage tracks. Genes found on the forward strand are shown as orange right-pointing arrows; Genes found on the reverse strand are shown as grey left-pointing arrows.

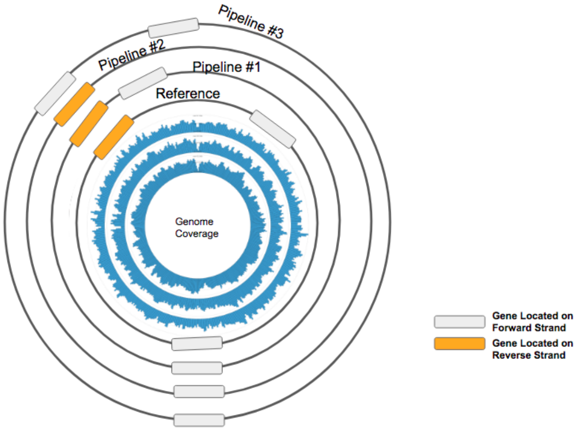
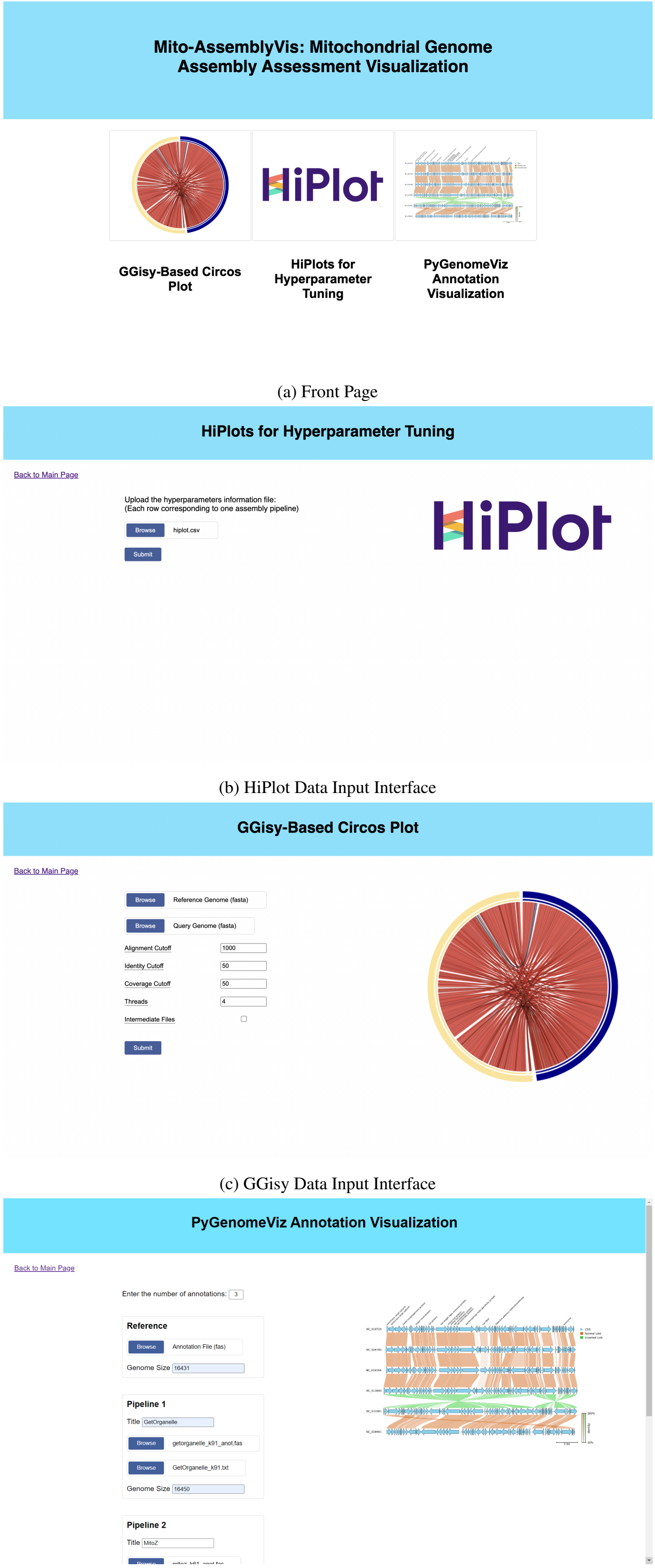

Pipeline #3
Pipeline #2
Pipeline #1
Reference
Genome
Coverage
Gene Located on Forward Strand
Gene Located on Reverse Strand

Fig. 7: Circular Multi-Layer Genome Annotation Plot: The 4 outer layers (from innermost to outermost) are the reference mitogenome and 3 pipeline-constructed sequences. The size of the annotated genes and their locations relative to the reference genome are also shown. Genes found on different strands (forward, reverse) are indicated by two hues (grey, orange). By comparing the annotated genes on reconstructed genomes with the reference, missing genes or gene relocations can be observed. The 3 inner blue layers (from innermost to outermost) are genome coverage from pipelines 1, 2, and 3, respectively.



Mito-AssemblyVis: Mitochondrial Genome Assembly Assessment Visualization

GGisy-Based Circos Plot
HiPlot
HiPlots for Hyperparameter Tuning
PyGenomeViz Annotation Visualization

(a) Front Page

HiPlots for Hyperparameter Tuning

Back to Main Page

Upload the hyperparameters information file:
(Each row corresponding to one assembly pipeline)

hiplot.csv

(b) HiPlot Data Input Interface

GGisy-Based Circos Plot

Back to Main Page

Reference Genome (fasta)

Query Genome (fasta)

Alignment Cutoff 1000

Identity Cutoff 50

Coverage Cutoff 50

Threads 4

Intermediate Files ☐

(c) GGisy Data Input Interface

PyGenomeViz Annotation Visualization

Back to Main Page

Enter the number of annotations: 3

Reference

Annotation File (fas)

Genome Size 16431

Pipeline 1

Title GetOrganelle

getorganelle_k91_anot.fas

GetOrganelle_k91.txt

Genome Size 16450

Pipeline 2

Title MitoZ

mitoz_k61_anot.fas

(d) pyGenomeViz Data Input Interface

Fig. 8: Mito-AssemblyVis Front Page and User Data Input Interfaces

8 Conclusion

We developed a web-based multi-purpose visualization tool for three levels of mitogenome assembly assessment including parameter fine-tuning, assembly sequence similarity comparison, as well as genome annotation analysis. Users can upload their data to the data input interfaces and generate the corresponding visualizations. The parallel coordinate plot created using the HiPlot package allows users to visualize the relationships between different assembly quality metrics and select optimal parameter settings. The GGisy-based circos plot displays the sequence identity levels and structural agreements between assembly sequences and the reference sequence. Finally, the pyGenomeViz parallel annotation plot is used to visualize the annotation results of multiple reconstructed sequences and the reference. The genome coverage distributions can also be visualized through pyGenomeViz plot. Overall, this visualization tool offers a one-click solution for visualizing assembly metrics data, gene annotation results, and reference/assembly sequence comparisons, facilitating effective mitogenome assembly quality assessments.

References

[1] Flask. `https://flask.palletsprojects.com/en/2.2.x/`. Accessed: 2022-12-14.

[2] M. Alexeyev, I. Shokolenko, G. Wilson, and S. Ledoux. The maintenance of mitochondrial dna integrity–critical analysis and update. *Cold Spring Harb Perspect Biol*, 5(5):a012641, 2013.

[3] M. Bernt, A. Donath, F. Jühling, F. Externbrink, C. Florentz, G. Fritzsch, J. Pütz, M. Middendorf, and P. F.Stadler. Mitos: Improved de novo metazoan mitochondrial genome annotation. *Molecular Phylogenetics and Evolution*, 69(2):313–319, 2013.

[4] A. Gurevich, V. Saveliev, N. Vyahhi, and G. Tesler. Quast: quality assessment tool for genome assemblies. *Bioinformatics*, 29(8):1072–1075, 2013.

[5] N. H. G. R. Institute. Shotgun sequencing. `https://www.genome.gov/genetics-glossary/Shotgun-Sequencing`. Accessed: 2022-12-13.

[6] J.-J. Jin, W.-B. Yu, J.-B. Yang, Y. Song, C. W. DePamphilis, T.-S. Yi, and D.-Z. Li. Getorganelle: a fast and versatile toolkit for accurate de novo assembly of organelle genomes. *Genome biology*, 21(1):1–31, 2020.

[7] M. Krzywinski, J. Schein, I. Birol, J. Connors, R. Gascoyne, D. Horsman, S. J. Jones, and M. A. Marra. Circos: an information aesthetic for comparative genomics. *Genome research*, 19(9):1639–1645, 2009.

[8] G. Meng, Y. Li, C. Yang, and S. Liu. Mitoz: a toolkit for animal mitochondrial genome assembly, annotation and visualization. *Nucleic Acids Research*, 47(11):e63, 2019.

[9] A. Mikheenko, G. Valin, A. Prjibelski, V. Saveliev, and A. Gurevich. Icarus: visualizer for de novo assembly evaluation. *Bioinformatics*, 32(21):3321–3323, 2016.

[10] moshi4. pygenomeviz. `https://github.com/moshi4/pyGenomeViz`.

[11] F. Research. Hiplot: High-dimensional interactive plots made easy. `https://ai.facebook.com/blog/hiplot-high-dimensional-interactive-plots-made-easy/`. Accessed: 2022-12-13.

[12] D. Sims, I. Sudbery, N. E. Ilott, A. Heger, and C. P. Ponting. Sequencing depth and coverage: key considerations in genomic analyses. *Nature Reviews Genetics*, 15:121–132, 2014.

[13] S. Valenzuela. Ggisy. `https://github.com/Sanrrone/GGisy`, 2017.