Tree House Explorer Documentation

Contents

Introduction

Genomes are mosaics of evolutionary histories that reflect ancient signatures of species divergence, as well as incomplete lineage sorting (ILS) or gene flow. In recent years, rapid advancements in massively parallel sequencing technologies have enabled researchers to collect large volumes of genome-scale phylogenetic data for many species. Understanding how and why phylogenetic signal varies across species genomes can yield powerful insights into evolutionary histories and adaptive evolution. By integrating diverse data types with local genealogies, users can differentiate genetic variation that is consistent with the species tree from that stemming from natural selection, ILS, or gene flow *(1-4)*. While several tools have been developed for the independent analysis of genomic data (e.g., IGV, UCSC Genome Browser, etc.) and phylogenetic tree visualization (Iroki, FigTree, etc.), a tool that can simultaneously analyze phylogenetic signal variation with other chromosomal and gene-based annotations has yet to be developed for the field of phylogenomics. Tree House Explorer (THEx) is a novel genome browser that allows users to integrate phylogenomic data and genomic annotations into a single interactive platform for combined analysis. THEx allows users to visualize genome-wide variation in evolutionary histories as well as genetic divergence on a chromosome-by-chromosome basis, with continuous sliding window comparisons to gene annotations, recombination rates, GC-content, and other user-specified, highly customizable feature annotations. THEx provides a new resource for interactive data visualization in phylogenomics and a novel approach to analyze and interpret the diverse evolutionary histories woven throughout genomes.

What Can Tree House Explorer Do?

Tree House Explorer offers a collection of dashboards that visualize phylogenetic and genomic data concurrently. Each dashboard provides highly interactive graphs that allow for smooth browsing through data at chromosome or genome-wide levels. In addition to being interactive, each dashboard provides an array of graph customization that help create publication-quality figures. Each graph comes with a built-in snapshot feature located in the top right corning in the toolbar that allows you to download a graph with a click of a button. Depending on the graph, users can also find other tools in the toolbar that provide other means of interacting with each graph. For Unix and MacOS systems, THEx comes with a command line suite called THExBuilder that provides pipelines and tools to generate, modify, and analyze THEx input files, allowing for further customization and control over your data.

THExBuilder

Purpose

THExBuilder is a suite of pipelines and tools designed to generate, modify, and analyze THEx input files. It provides a customized solution for generating Tree Viewer input files. It also contains a single command pipeline for calculating raw p-distance values from multiple-sequence fasta files in non-overlapping sliding windows for Signal Tracer. THExBuilder was designed to be flexible and scalable, taking advantage of multiprocessing and partitioned workflows. For example, the Tree Viewer Pipeline is designed to be run either with a configuration file or by command line arguments, as well as being able to run the pipeline step-by-step or from start-to-finish with a single command. This provides the user the freedom and flexibility to tailor their runs and track the progress through pipelines with ease.

Usage

In addition to being a novel interactive genome browser, THEx comes with a built-in command line suite that contains tools and pipelines that help simplify the process of building and manipulating input files for the dashboards. There are two main pipelines, one for Tree Viewer and one for Signal Tracer. To use THExBuilder, activate the conda environment that THEx is installed within and enter the command _thexb_ to begin. This command will bring up the help section, providing an overview of all the options and available tools.

THExBuilder's output is contained in a single THExBuilderOutput directory. By default, the output directory is written to the current working directory, so if you do not provide an output location be sure to use the same working directory as you move through the pipelines. Within the output directory you will find the intermediate output files for each stage of the different pipelines as well as log files to help track what has been performed, to what files, in what order, and with what input parameters.

Input Files

All of the pipelines for THExBuilder start with multiple-sequence alignment fasta files, organized within a single directory. Each fasta file ideally represents the multiple-sequence alignment of a single chromosome, but scaffold level assemblies can be used too with the caveat that each sequence will be treated as independent sequences and there is no way to link scaffolds into pseudo-chromosomes within the pipeline or application. Below is a basic example of the directory structure and file structure of the multiple-sequence fasta file.

MultiAlignmentDir/

```
| -- chr1.fasta

| -- chr2.fasta

| -- chX.fasta
```

<u>Example file</u>: (chr1.fasta)

```
> Sample 1

AGTGCTAGC…GTTC

> Sample 2

AGTGTCCTA…GCTT
> Sample 3
AGTGTCGTA…GCTT
```

Tree Viewer Pipeline

The Tree Viewer pipeline is designed to take multi-alignment fasta files, run them through several filtration steps, and then through IQ-Tree to generate per-window maximum likelihood phylogenies. These phylogenies along with their chromosome-window coordinates are put together in a Tree Viewer file where the topologies can then be binned and labeled for viewing in Tree Viewer. The examples below show how to run each stage with and without a configuration file, providing the user flexibility to run the pipeline with all arguments conveniently located in a single file. It is recommended to use a configuration file as this helps improve reproducibility and also reduces the length of the commands, but the option is left to the user. Since the configuration file has to be built with a specific format, you can generate a blank configuration file running "thexb --tv_config_template".

Start-to-Finish Run

The Tree Viewer pipeline is designed to be flexible, providing the user complete control over how the pipeline is ran. Although not recommended, one can run the entire Tree Viewer pipeline start to finish by providing a configuration file and calling the "--tv_all" argument. This will run through each of the pipeline steps described below, but will not stop after each step is completed. Although the option to run the pipeline start-to-finish is available, it is recommended to run each step individually to inspect and validate the intermediate results.

Command Example

```
$ thexb –tv_all -c config.ini
```

Fasta Windowing

The fasta windowing stage will take a directory containing multi-alignment fastas (ideally whole chromosomes) and will return a directory of subdirectories, labeled by the original sequence names (i.e., chromosome names), where each subdirectory contains the windowed fasta files. Note that this step can return thousands of files when using a small window size (i.e., 5kb for a 2.5Gb genome), so be mindful when opening these directories in a file browser.

Command Example

```
$ thexb --minifastas -c config.ini
```

or

```
$ thexb --minifastas --window_size 10kb -i dir_of_multi_alignment_files/
```

Descriptions:

--window_size = Window size [bp/kb/mb] (default: 100kb)
-i = Input file path (Cannot be used when running Tree Viewer pipeline with configuration file)

Trimal Gap Trimming

The Trimal (*5*) stage of the pipeline uses the gap-threshold feature that removes regions of a window where a taxon has missing data that exceeds the provided threshold for allowed gaps. A second threshold is provided to remove resulting sequences that do not meet a minimum sequence length.

Command Example

```
$ thexb --trimal -c config.ini
```

or

```
$ thexb --trimal –trima_gapl_threshold 0.1 --trimal_min_seq_len 1kb -i
THExBuilderOutput/windowed_fastas
```

Descriptions:

--trimal_gap_threshold = Percentage of alignment with no gaps (i.e., removes sequence with gaps in 10% or more of the sequences

--trimal_min_seq_len = Minimum sequence length to retain window in study. If above, window is excluded

-i = Input file path (Cannot be used when running Tree Viewer pipeline with configuration file)

Pairwise Distance + Coverage Filter

The pairwise distance and coverage filter walks through each window of the genome and conducts a sliding sub-window analysis with a step size of *n*-bp to mask local regions of extreme divergence that may be caused by misalignment or undetected paralogy. It also masks entire windows where a single taxon's base coverage (valid bases include A, T, G, and C) is below the provided threshold. This step was integrated from Foley et al. 2021 (*6*) and updated to run on Python 3.

Command Example

```
$ thexb --pw_filter -c config.ini
```

or

```
$ thexb --pw_filter --pw_subwindow_size 100bp --pw_step 10bp --pw_min_seq_len 1000bp --
pw_min_pdist 0.15 --pw_zscore_cutoff 2 --pw_seq_coverage 0.9 -i
THExBuilderOutput/trimal_filtered_windows/
```

Descriptions:

--pw_subwindow_size = Sub-window size (default: 100bp)

--pw_step 10bp = Sub-window step size (default: 10bp)

--pw_min_seq_len = Minimum length of sequence in a given window to be retained (default: 1000bp)

--pw_max_pdist = Maximum p-distance for a given taxon to be considered non-extreme. Any sequence above this value in a given sub-window will be masked. Note this value is dataset and taxon dependent! (default: 0.15)

--pw_zscore_cutoff = Maximum number of standard deviations away from the mean (calculated as z-score) allowed before entire window is masked. (default: 2)

--pw_seq_coverage = Minimum valid, non-missing sequence coverage. Individual sequences below this value are masked. (default: 0.9)

-i = Pathway to directory containing the output from the Trimal stage
(i.e., THExBuilderOutput/trimal_filtered_windows/)

IQ-Tree: Maximum Likelihood Phylogeny Inference

IQ-Tree (*7-8*) is used to infer maximum likelihood phylogenies providing the basis for the distribution of phylogenetic signal visualized in Tree Viewer. The filtered fasta files generated by the pairwise filtration step is passed to IQ-Tree and the resulting Newick files are collected and organized into the basis of the Tree Viewer input file. IQ-Tree creates several output files per-window, but we are only interested in the trees so the other data can be saved or discarded.

There are two approaches built into the Tree Viewer pipeline for the IQ-Tree analysis. The first is to have THExBuilder call IQ-Tree for each filtered window by using the "--iqtree" command, a model of sequence evolution, and number of bootstrap replications. The second approach is to take the filtered fasta files from the pairwise filtration step and run IQ-Tree externally on a different local machine, server, or cluster. If you choose to run IQ-Tree externally, you will need to organize the resulting ".treefile" Newick trees into a single directory and pass the directory to the "-iqtree_external" command rather than the "--iqtree" command. You can also optionally create sub-directories per chromosome and organize the ".treefile" by chromosome and pass it to the "--iqtree_external" command, but it is not required.

Command Example

```
$ thexb --iqtree -c config.ini
```

or

```
$ thexb --iqtree --iqtree_model "GTR*H4" --iqtree_bootstrap 1000 \
-i THExBuilderOutput /pairwise_filtered_windows/
```

Descriptions:

--iqtree_model = Any valid model of nucleotide sequence evolution supported by IQ-Tree
--iqtree_bootstrap = Number of bootstrap replicates to run
-i = Pathway to directory containing the output from the Pairwise Distance + Coverage Filter step

Topobinner

Topobinner is a tool used to organize and label equal tree topologies based on RF-distance. Trees are treated as equal topologies when their RF-distance is equal to 0 (*9-10*). This is what allows you to visualize discordant topologies across the genome and identify regions of interest. The trees are binned and then labeled by their whole genome frequency, meaning the most frequent topology in the genome will be labeled Tree1, then Tree 2 for the second most frequent topology, and so on. Once this step is completed, an updated Tree Viewer file with binned topologies will be generated and placed in the THExBuilderOutput directory ready for visualization in Tree Viewer. It should also be noted that users can also pass a Tree Viewer input file produced by the *File Pruning* export option (see Tree Viewer export options section for more details) in Tree Viewer to bin new trees that have been pruned into a subset from a larger Tree Viewer input file. No special steps need to be taken to run this, simply provide the Tree Viewer input file as you would normally.

Command Example

```
$ thexb --topobinner -c config.ini
```

or

```
$ thexb --topobinner -i TreeViewer_file_with_blank_TopologyID_col.xlsx
```

Descriptions:

-i = Tree Viewer input file from IQ-Tree step or File Pruning export option with TopologyID column blank

TOPOBINNER ALTERNATIVE:

An alternative to using topobinner is PhyBin (https://github.com/rrnewton/PhyBin). PhyBin is not hosted on conda, so you would need to download the program from GitHub and install it externally.

It is important that you use PhyBin's "--bin" option and not it’s clustering algorithm. When you run PhyBin, it will generate binned results in a new output directory named either phybin_output or whatever you provide as an output location. Passing the output directory generated by PhyBin and a Tree Viewer file with the TopologyID column blank to THExBuilder's "--phybin_external" command will produce a complete Tree Viewer input file just like topobinner.

Command Example

```
$ thexb --phybin_external -i pathway_to_phybin_output_directory/ --tv_file_name
./THExBuilderOutput/TreeViewer_input_file.xlsx
```

Signal Tracer Pipeline

The Signal Tracer pipeline is a single script that calculates raw, uncorrected p-distance between a reference and other taxon of one or more multi-alignment fasta files. There are two ways to run the pipeline, you can provide a single multi-alignment fasta file or a directory of multi-alignment fasta files. The directory of files is typically the input directory used for the entire Tree Viewer pipeline that is ideally a multi-alignment fasta file for each chromosome. Since some chromosomes can be quite large, this process may take several hours to run, if not a day or two depending on the system configuration being used (i.e., processor, amount of memory, etc.). The only two additional arguments that are needed to run the pipeline are the window size and missing data threshold. The missing data threshold is a value between 0.0 and 1.0 that indicates the maximum amount of missing data (gaps or masked sequence) allowed for a single taxon, and if that threshold is surpassed, then the values for all taxa in the window are set to NULL. The output from this pipeline is formatted to be directly uploaded into the Signal Tracer dashboard for quick visualization!

Command Example

```
$ thexb --pdistance --pdist_threshold 0.75 --window_size 100kb
```

Descriptions:

--pdist_threshold = Maximum frequency of missing data allowed in a single taxon before the entire window is masked. (value: 0.0-1.0)

--window_size = Window size [bp/kb/mb] (default: 100kb)

Tree Viewer

Purpose

Tree Viewer is a dashboard designed for window-based approaches to visualizing phylogenetic signal across a reference genome alongside additional data types like recombination rate, GC-content, and gene annotations. Through simultaneous visualization of phylogenetic signal and additional data types, users are offered an all-in-one experience for identifying and understanding the implications of phylogenetic signal variation and its underlying genomic context. Through Tree Viewer, users are able to make more impactful findings and gain a better understanding of their data all in one place. Download the example data sets from GitHub and refer to Li et al. 2019 for an example clade that illustrates Tree Viewer (*2*).

Input Files

There are two required input files to run Tree Viewer: 1) Tree Viewer input file and 2) chromosome length BED file. These two files provide the required information to visualize phylogenetic signal across the genome in proper scale to the chromosome length. Additional window-based data types can be added to the Tree Viewer input file providing, the ability to visualize a multitude of data types concurrently. These additional data types are added as a new column in the Tree Viewer file keeping your data succinct and organized.

Tree Viewer Input File

The input file for Tree Viewer is designed to be simple to make and even easier to incorporate new window-based data. Tree Viewer takes a tab or comma delimited file where the first four columns are Chromosome, Window (i.e., 100,000 - which covers bases 1-100,000 for 100kb windows), NewickTree, and TopologyID. The first four columns are required and must have the appropriate headers in the order given in the example input below. Tree Viewer accepts four different file extensions (.csv, .tsv, .txt, .xlsx) for the input file. Note there are column and per-cell limitations to Excel (.xlsx) files, so large datasets may be better off in a flat file format like .csv, .tsv, or .txt.

Chromosome	Window	NewickTree	TopologyID
Chr1	1000	(A,(B,C));	Tree1
Chr2	2000	(B,(A,C));	Tree2
Chr3	3000	(C,(A,B));	Tree3
Chr4	4000	(A,(B,C));	Tree1

Additional numeric and categorical data (i.e., GC-content, recombination rate low/high regions, etc.) can be easily incorporated into your Tree Viewer file by adding an additional column on the right-most side of the Tree Viewer input file. When you load your Tree Viewer file into a new session, the additional data features will load as options in the “Additional Data" dropdown in the “Single Chromosome” tab of the toolbar.

Chromosome	Window	NewickTree	TopologyID	GC-content
Chr1	1000	(A,(B,C));	Tree1	0.35
Chr2	2000	(B,(A,C));	Tree2	0.46
Chr3	3000	(C,(A,B));	Tree3	0.33

Chromosome Lengths BED File

The second file required to run Tree Viewer is a BED file (.bed) of chromosome lengths. Ensure that the chromosome length bed file contains all chromosomes that are found in the Tree Viewer main input file, otherwise you will be prompted with an error and asked to update your BED file with the missing data. It is also important that you create your file with headers as shown below.

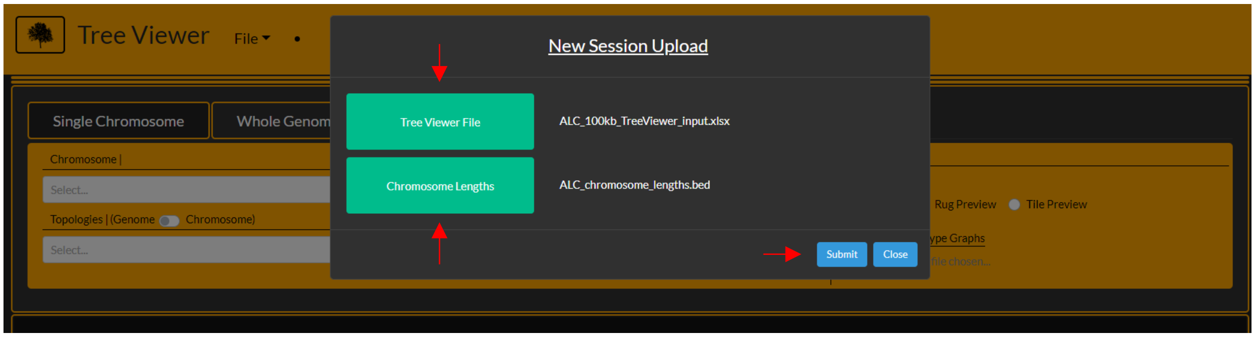
Chromosome	Start	End
Chr1	0	1000
Chr2	0	2000
Chr3	0	3000
Chr4	0	4000

Standardized File Types

Tree Viewer currently only offers the ability to load GFFv3/GTF gene annotation files. Provided one of these files, users can investigate underlying genes, coding regions, and other annotations concurrently with phylogenetic signal and all other user provided window-based data types. More standard data types will be added in future updates.

Usage

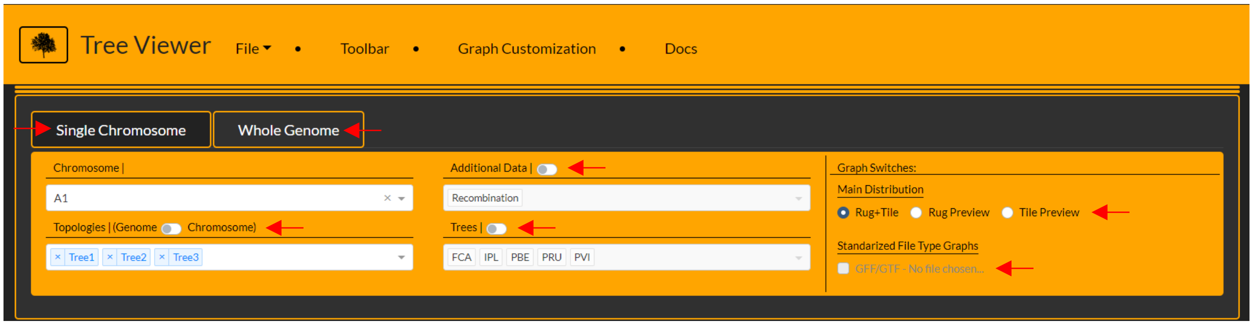
Upon opening Tree Viewer, a pop-up will appear prompting you to select two input files. The two required input files are a Tree Viewer input file and a chromosome length BED file (formats are described in input file structure section). Clicking each button will bring up a file system browser allowing you to navigate to the location of your input files. When a file is selected, the upload button will turn green if the file type is valid, or red if it is invalid. After selecting both a Tree Viewer input file as well as a chromosome length BED file, you click submit. Tree Viewer then compares the input files and returns an error message if one or both files are incorrectly formatted. If both files pass the validation checks, they will automatically be uploaded into the session and the toolbar dropdowns will be populated with the data provided in the input files.



Tree Viewer File ALC_100kb_TreeViewer_input.xlsx

Chromosome Lengths ALC_chromosome_lengths.bed

The dashboard is separated into two sections, the “Single Chromosome” tab and “Whole Genome” tab. Within the “Single Chromosome” tab, the main distribution graph is the master graph for this view, meaning all zooming and panning performed on the master graph will update the x-axis range for all other additional data graphs being shown. This allows for smooth, simultaneous investigation of phylogenetic signal alongside additional data types like recombination rates, GC-content, and gene annotations without the need to align the data yourself. “Whole Genome” view allows for a large-scale overview of the phylogenetic signal, making it easy to pick out interesting chromosomal regions to investigate in further detail in the “Single Chromosome” view.



Chromosome | A1

Topologies | (Genome ○ Chromosome) Tree1 Tree2 Tree3

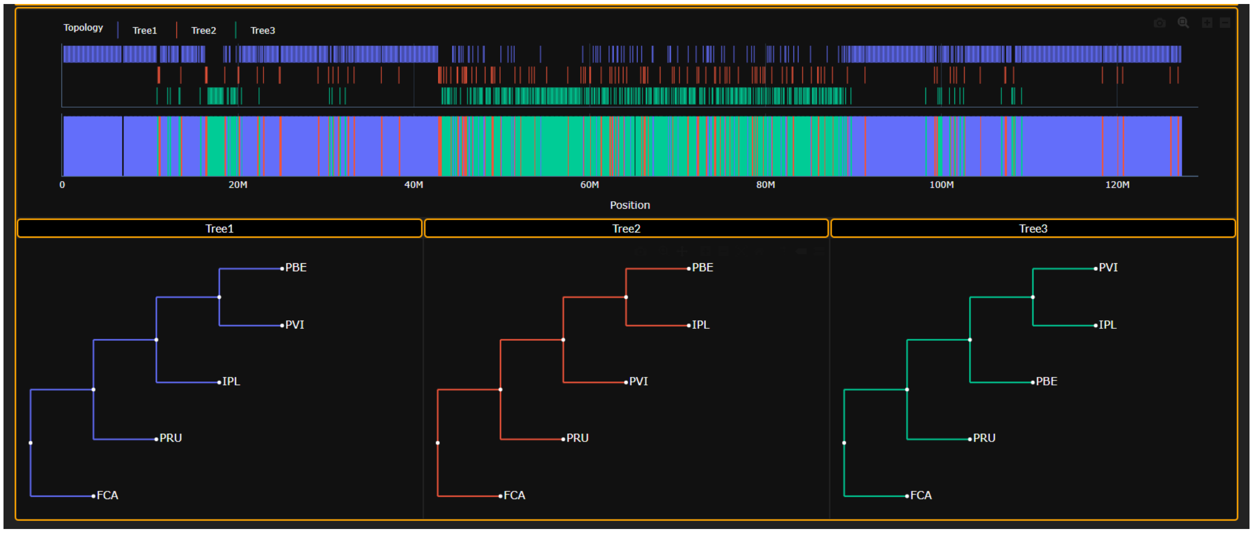
Additional Data | ○ Recombination

Trees | ○ FCA IPL PBE PRU PVI

● Rug+Tile ○ Rug Preview ○ Tile Preview

□ GFF/GTF - No file chosen...

“Single Chromosome” view provides the most interactive place to explore your phylogenetic signal. This is where you will add/remove topologies, load additional data types into graphs, and visualize tree topologies. The dropdowns for the topologies, additional data, and tree taxa allow the user to customize which data are being shown at a given time. You may plot one topology or all topologies, with the caveat that as one increases the number of topologies, this complicates visualization and analysis. By default, Tree Viewer orders the topologies by their genome-wide frequencies. However, it may be useful to look at them by chromosome frequency, and there is a switch next to the Topologies header that allows you to change between these alternatives. Turning the Additional Data switch to “On” will load all additional data types that are selected within its respective dropdown, allowing you to load multiple data types at once. Lastly, turning on the Trees switch will load simple representations of the tree topologies selected in the Topologies dropdown. Currently, Tree Viewer plots the trees without branch lengths since branch lengths can change from window-to-window. By selecting/removing taxa names from the Tree’s dropdown, you can prune the tree topology to zoom in on a specific clade or set of taxa. Note that this function does not re-bin topologies, it simply visually prunes the trees. If you are wanting to prune and re-bin topologies, see the File Pruning function in the Tree Viewer export options section.



On the right most side of the single chromosome toolbar, there are a few extra options. Under *Graph Switches* you will find toggles for the main topology graph as well as a switch for GFF/GTF gene annotation files. The main distribution options allow you to independently preview the rug plot and tiled histogram chromosome displays. These previews can be exported using the snapshot button in the top-right of each graph. To upload a GFF/GTF file, go to “File - > GFF/GTF” and upload the file in the same manner as the main input files. Once the file is loaded into the session, the GFF/GTF switch will activate allowing you to load the GFF/GTF graph.

Walk-Through: Start New Session + New Load Input Files

1. Click "*File + New Session*" – A pop-up will appear titled "New Session Upload".
2. Click "*Tree Viewer Input*" button – navigate to and select Tree Viewer input file.
3. Click "*Chromosome Lengths*" button – navigate to and select Chromosome Length BED file.
4. Assuming both input buttons have turned green, click "Submit".

Walk-Through: Load GFF/GTF Files

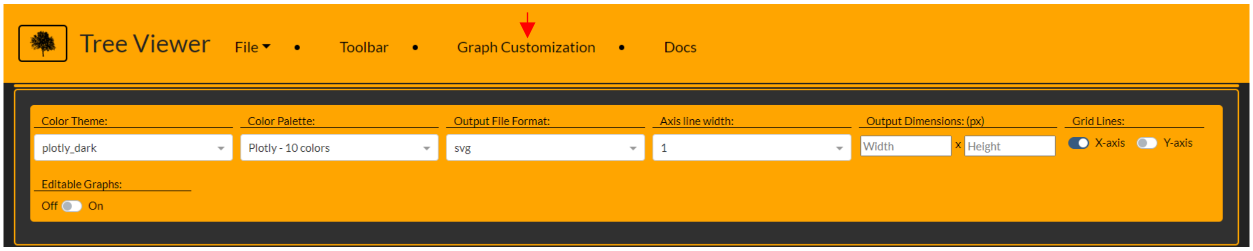
1. Click "*File + GFF/GTF*" – A pop-up will appear titled "Alternative Input Files".
2. Click "*GFF/GTF File*" button – navigate to and select GFF/GTF input file.
3. Once the button turns green, click "Submit" – you should see the GFF/GTF switch now active.

Walk-Through: Zooming + Panning

1. Zoom in by clicking and dragging across a region of interest.
2. You can zoom in or out in steps by using the plus and minus buttons in the toolbar at the top right corner of the graph.
3. Reset the x-axis range by double clicking the graph.
4. Pan by clicking and dragging the x-axis of the graph

Graph Customization

Graph Customization allow you to customize the look of the topology or data distribution graphs. Dropdowns provide lists of themes, color palettes, gridline toggles, and several output formats to choose from. Making changes to the graph customizations will update all graphs currently loaded on the browser. Through these customizations, one can create publication-quality graphs that can be dropped directly into your manuscript. We have allowed users the choice to specify the output dimensions in pixels, however leaving the inputs blank will revert each graph to default dimensions that are optimized for a word document or .pdf file.



Color Theme: plotly_dark

Color Palette: Plotly - 10 colors

Output File Format: svg

Axis line width: 1

Output Dimensions: (px) Width x Height

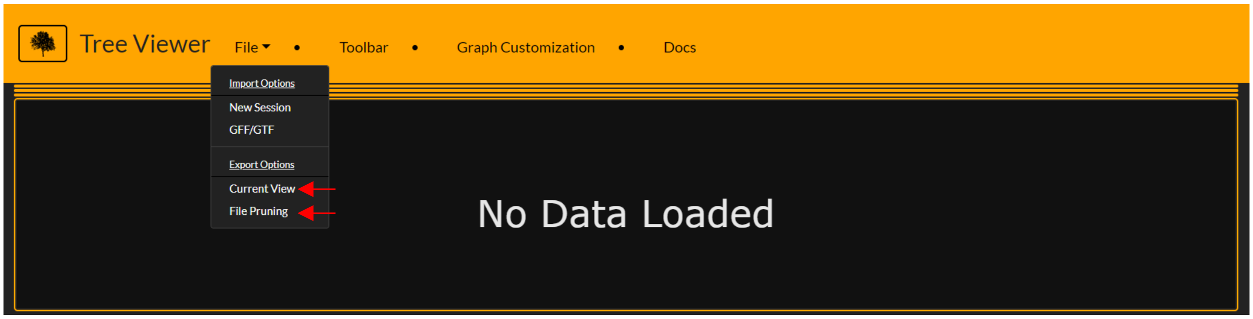
Grid Lines: ● X-axis ○ Y-axis

Editable Graphs: Off ○ On

Tree Viewer Export Options

Tree Viewer offers two export options: current view and file pruning. The current view option allows you to extract local information for a chromosomal region. For example, you are in single chromosome view and looking at Chromosome 1 and you zoom into range 1.5Mb-2Mb. If you click the current view export option, a download prompt will appear allowing you to choose a place to download a new Tree Viewer file that only contains the information for the current range. If you also have a GFF/GTF gene annotation file loaded into the session, it will also extract the information from the given range.

The file pruning export option allows you to select a subset of taxa and create a new Tree Viewer input file with a pruned set of Newick trees. This process will also clear the TopologyID column, enabling you to re-bin the new tree topologies. There is an option to run the binning process before downloading the new file, but please note that re-binning large trees across large genomes (e.g, >2 Gbp) may require considerable run time and it will lock your session until it completes. In such circumstances we recommend running the --topobinner command in THExBuilder.



Walk-Through: Current View Export

1. Click "*File + Current View*".
2. A file-system prompt will appear and ask you to select a location and file name. Once you have selected a location and file name, click "*Save*".
    a. If a GFF/GTF file is loaded into the session, then a second file-system prompt will appear to download a subset of the GFF/GTF file too.

Walk-Through: File Pruning

1. Click "*File + File Pruning*" – A pop-up will appear titled "Tree Viewer File Pruning".
2. Select taxa to retain in the Tree Viewer input file subset using the dropdown.
3. Check "*run Topobinner"* if you wish to run topobinner prior to downloading the file subset.
4. Finally, click "*Submit*" and wait for a file-system prompt to appear to choose the location to download the new subset file.

Signal Tracer

Purpose

Signal Tracer (ST) is a simple dashboard that visualizes window-based values calculated from multiple-sequence alignments. It provides an easy way to investigate variation in variables like genetic distance, branch length, or divergence times, at single chromosome and whole genome levels. Signal Tracer was originally developed to validate the phasing of F1 hybrid long-reads using the trio-binning approach for single haplotype genome assembly references. By visualizing genetic distance of reference and non-reference species, we were able to visually validate that there were no regions that indicate potential improper phasing. If you would like to learn more about this process, refer to our paper (*11*).

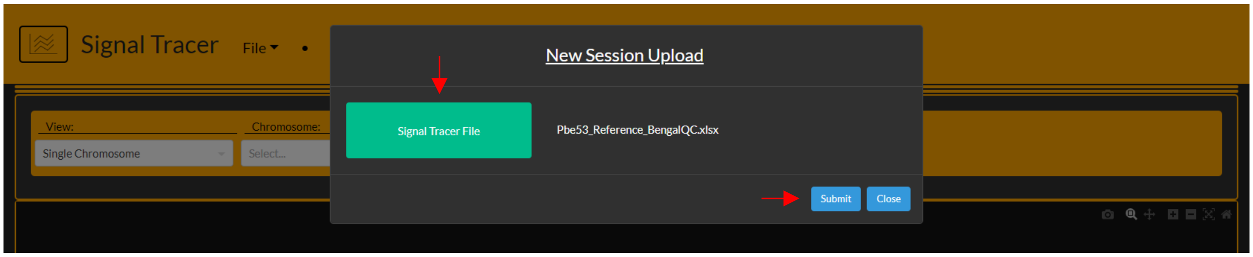
Input File Structure

The input file for Signal Tracer is a tab or comma delimited file consisting of four columns: *Chromosome*, *Window*, *Sample*, and *Value*. The *Value* column can only contain numerical values, not categorical values. Ensure the headers of your input file match the headers listed in italics above, an error message will appear if they do not match exactly.

Chromosome	Window	Sample	Value
Chr1	100	Tiger1	0.015
Chr1	100	Tiger2	0.25
Chr2	200	Tiger1	0.011
Chr2	200	Tiger2	0.008

Usage

Upon entering Signal Tracer, a new session pop-up will appear prompting you to upload your data. Click submit once the “Load File” button turns green. From there you can switch between chromosomes or change to whole genome view and explore your data.



View: Single Chromosome

Chromosome: Select...

Pbe53_Reference_BengalQC.xlsx

This dashboard was originally designed to visualize raw genetic distances variation across the genome, so THExBuilder comes will a “--pdistance” command to generate input files for Signal Tracer from multi-alignment fasta files. Visit the Signal Tracer Pipeline in the THExBuilder documentation for more information.

Walk-Through: Load Input Files

1. Click "*File + New Session*" – A pop-up will appear titled "New Session Upload".
2. Click "*Signal Tracer File*" button – navigate to and select Tree Viewer input file.
3. Assuming the input button turned green, click "Submit".

Walk-Through: Load GFF/GTF Files

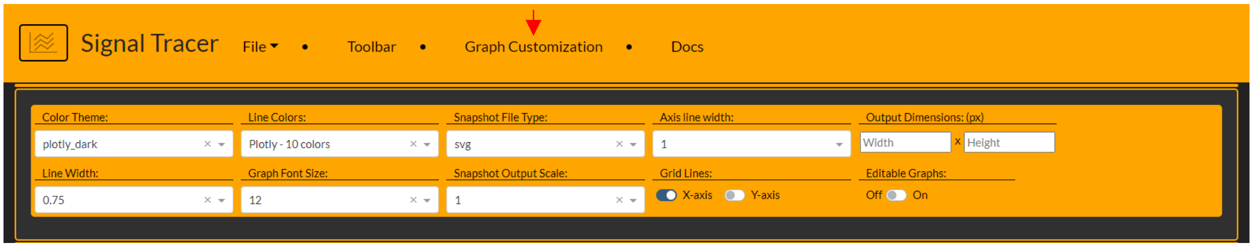
1. Click "*File + GFF/GTF*" – A pop-up will appear titled "Alternative Input Files".
2. Click "*GFF/GTF File*" button – navigate to and select GFF/GTF input file.
3. Once the button turns green, click "*Submit*" – you should see the GFF/GTF switch now active.

Walk-Through: Zooming + Panning

1. Zoom in by clicking and dragging across a region, then release.
2. You can zoom in or out in steps by using the plus and minus buttons in the top right corner of the graph.
3. Reset the view by double clicking the graph.
4. Pan by clicking and dragging the x-axis of the graph

Graph Customization

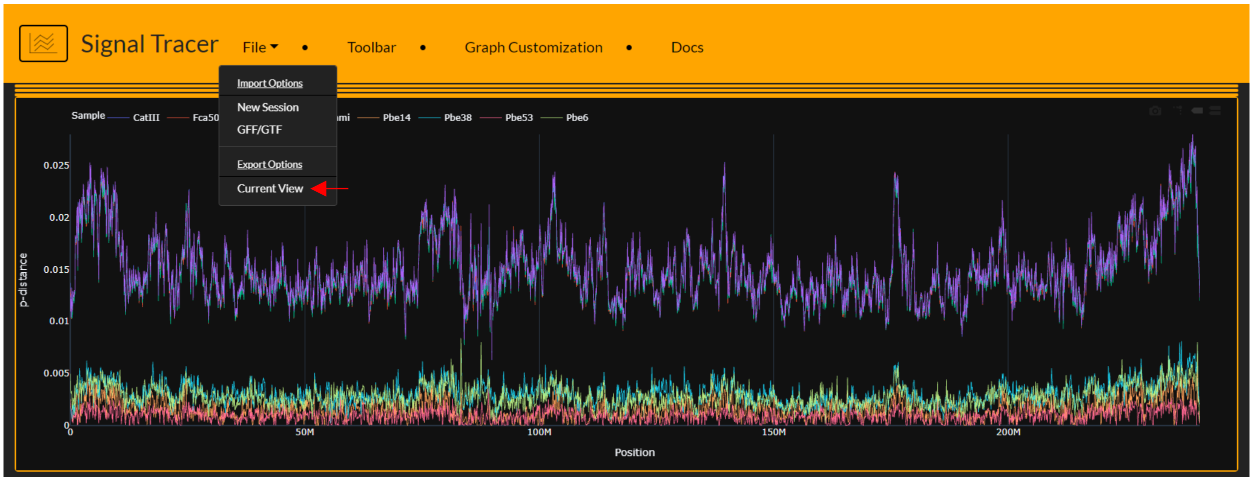
Graph Customization allow you to customize the look of the chromosome graphs in an easy-to-use way. Dropdowns provide lists of themes, color palettes, gridline toggles, and several output formats to choose from. Making changes to the graph customizations will update all graphs currently loaded on the browser. Through these customizations, one can create publication-quality graphs that can be dropped directly into your in-progress manuscript. Although Signal Tracer allows users to specify the output dimensions in pixels, leaving the inputs blank will revert each graph to default dimensions that are optimized for a word document or .pdf file.



Color Theme: plotly_dark
Line Colors: Plotly - 10 colors
Snapshot File Type: svg
Axis line width: 1
Output Dimensions: (px) Width x Height
Line Width: 0.75
Graph Font Size: 12
Snapshot Output Scale: 1
Grid Lines: ● X-axis ○ Y-axis
Editable Graphs: Off ○ On

Signal Tracer Export Options

Signal Tracer offers a single export option that allows you to extract local information from your input file. Similar to Tree Viewer, zooming into a region of a chromosome in single chromosome view and selecting “File -> Current View” will bring up a download prompt where you can specify the file name and download location.



Walk-Through: Current View Export

1. Click “*File + Current View*”.
2. A file-system prompt will appear and ask you to select a location and file name. Once you have selected a location and file name, click “*Save*”.
    a. If a GFF/GTF file is loaded into the session, then a second file-system prompt will appear to download a subset of the GFF/GTF file too.

References

1. Edelman, N. B. *et al.* Genomic architecture and introgression shape a butterfly radiation. *Science (80- .).* 366, 594–599 (2019).

2. Li, G., Figueiró, H. V, Eizirik, E., Murphy, W. J. & Yoder, A. Recombination-Aware Phylogenomics Reveals the Structured Genomic Landscape of Hybridizing Cat Species. *Mol. Biol. Evol.* 36, 2111–2126 (2019).

3. Nelson, T. C. *et al.* Ancient and recent introgression shape the evolutionary history of pollinator adaptation and speciation in a model monkeyflower radiation (*Mimulus* section Erythranthe). *PLOS Genet.* 17, e1009095 (2021).

4. Small, S. T. et al. Radiation with reticulation marks the origin of a major malaria vector. Proc. Natl. Acad. Sci. 202018142 (2020) doi:10.1073/pnas.2018142117.

5. trimAl: a tool for automated alignment trimming in large-scale phylogenetic analyses. Salvador Capella-Gutierrez; Jose M. Silla-Martinez; Toni Gabaldon. Bioinformatics 2009 25: 1972-1973.

6. Foley et al. Zoonomia Phylogenetic Analysis (*unpublished*)

7. L.-T. Nguyen, H.A. Schmidt, A. von Haeseler, B.Q. Minh (2015) IQ-TREE: A fast and effective stochastic algorithm for estimating maximum likelihood phylogenies.. Mol. Biol. Evol., 32:268-274.

8. S.M. Crotty, B.Q. Minh, N.G. Bean, B.R. Holland, J. Tuke, L.S. Jermiin, A. von Haeseler (2019) GHOST: Recovering historical signal from heterotachously-evolved sequence alignments. Syst. Biol., in press. https://doi.org/10.1093/sysbio/syz051

9. Robinson, D. F. & Foulds, L. R. Comparison of phylogenetic trees. Math. Biosci. 53, 131–147 (1981).

10. ETE 3: Reconstruction, analysis and visualization of phylogenomic data. Jaime Huerta-Cepas, Francois Serra and Peer Bork. Mol Biol Evol 2016; doi: 10.1093/molbev/msw046

11. Bredemeyer, K. R. *et al.* Ultracontinuous Single Haplotype Genome Assemblies for the Domestic Cat (Felis catus) and Asian Leopard Cat (Prionailurus bengalensis). *J. Hered.* 112, 165–173 (2021).

CITATION

If you use Tree House Explorer in your analysis, please cite it using the citation below.

<CITATION GOES HERE>