Evolutionary sequence analysis and visualization with Wasabi

Andres Veidenberg and Ari Löytynoja

Keywords: evolutionary sequence analysis, reproducible research, data visualization, web application

Running title: Wasabi

Affiliation: Andres Veidenberg (corresponding author) and Ari Löytynoja

Institute of Biotechnology, University of Helsinki, Helsinki, Finland

e-mail: andres.veidenberg@helsinki.fi

Abstract

Wasabi is an open-source, web-based graphical environment for evolutionary sequence analysis and visualization, designed to work with multiple sequence alignments within their phylogenetic context. Its interactive user interface provides convenient access to external data sources and computational tools, and is easily extendable with custom tools and pipelines using a plugin system. Wasabi stores intermediate editing and analysis steps as workflow histories and provides direct-access web links to datasets, allowing for reproducible, collaborative research and easy dissemination of the results. In addition to shared analyses and installation-free usage, the web-based design allows Wasabi to be run as a cross-platform, stand-alone application, and makes its integration to other web services straightforward.

This chapter gives a detailed description and guidelines for the use of Wasabi's analysis environment. Example use cases will give step-by-step instructions for practical application of the public Wasabi, from quick data visualization to branched analysis pipelines and publishing of results. We end with a brief discussion of advanced usage of Wasabi, including command-line communication, interface extension, offline usage, and integration to local and public web services. The public Wasabi application, its source code, documentation and other materials are available at http://wasabiapp.org

1 Introduction

In evolutionary sequence analysis, phylogenetic trees and multiple sequence alignments are tightly linked. Many analyses use one in the inference of the other, e.g. a guidetree to infer an alignment, or an alignment to infer a phylogenetic tree. Some sequence aligners (e.g. PRANK [1] and PAGAN [2]) and many downstream evolutionary analysis tools and pipelines (e.g. CodeML [3], EPO [4]) combine phylogenetic and sequence data to infer parameters attached to specific nodes of input trees. Some parameters relate only to the tree while others, like ancestral sequences, are associated both on the tree and the input alignment. To get the necessary context for drawing conclusions, such parameters should be displayed together with both input datasets.

Wasabi [5] was designed to work with complex phylogenetic datasets, displaying each sequence next to the corresponding tree node and maintaining the link through tree edits and downstream analysis steps. In addition to data visualization, Wasabi integrates external programs, editing tools, data management, and related functions into a user-friendly graphical interface, providing a comprehensive environment for phylogenetic sequence analysis. While there are other software packages with a more versatile tools selection (e.g. Mega [6], ETE [7]), Wasabi is characterized by its web-based implementation. Use of modern web technologies allows Wasabi to provide, among other features, installation-free access, fine-grained customizations, secure linking to datasets, and a plugin system for extending its functionality.

The first two sections below provide an overview of Wasabi and an example workflow of a Wasabi analysis. They describe Wasabi in a public web service configuration (as used in http://wasabiapp.org), where a central analyses database is accessed via user accounts and sharing URLs. After that we briefly discuss alternative setups of Wasabi, its modifications with plugins and other advanced topics. Throughout the chapter, *italics* is used to mark the terms found in the Wasabi interface, and underline for typed text, filenames and web addresses. Consecutive actions (typically mouse clicks) are linked with arrows (→). Some paragraphs are supplemented with a note section to add details to the main text.

2 Overview of the user interface

Wasabi's graphical user interface is arranged to a horizontal toolbar placed on top of the visualization area with sections for rendering phylogeny, taxa names and multiple sequence alignment (see Fig. 1). This layout is adjustable: the top toolbar can be contracted or collapsed to maximize the visualization space (use *Tools* → *Settings*) and the vertical divider lines between the visualization sections are draggable to change their relative width. Most of the interface elements (e.g. buttons, icons or text with dotted underline) reveal a tooltip describing the associated function when pointed with a mouse cursor for a few seconds. Colored text indicates links that open sections (blue links) or windows (red links) within the Wasabi interface.

The top toolbar includes buttons for drop-down menus, visualization zoom level, undo/redo and notifications. Most of the tasks in Wasabi are done via dialog windows listed in the toolbar menus. To reduce interface clutter, only the applicable tasks are visible. Fort example, the *Data* menu initially shows just the *Import* tool. After a dataset is imported, the menu is expanded with *Export* and *Info* tools. Logging to a user account (see the example workflow below) adds *Analysis library*, *Save* and *Share* options.

The *Import* tool accepts tree and/or sequence data input from local or remote sources. It auto-detects common file standards (FASTA, ClustalW [8], Phylip [9], Newick [10], NHX [11], NEXUS [12], HSAML [13], PhyloXML [14]) but can also handle unknown data formats. The input sequence type (DNA/RNA/protein) is set automatically but, if needed, it can be manually corrected in the *Info* tool. The *Import* window includes a file drop area and selector button for local files, a dedicated section for importing phylogenomic datasets from the Ensembl database [15] and a multi-source text field accepting data files from a web address, Wasabi dataset ID or raw text data. The right side of the text field is accompanied by a clickable triangle to expand the text field and a "plus"-marked button for importing multiple files. Once a dataset has been imported, it can be converted to another format in the *Export* tool (supports FASTA, Phylip, Newick, NHX, HSAML and NEXUS).

The *Tools* menu lists the integrated command-line analysis programs, built-in data editing tools and system settings. The selection of available tools depends on the installed plugins and the type of input data imported to Wasabi. At the time of writing and using the default plugins, the following tools appear when the currently open dataset includes:

- sequences: PRANK [1, 5], PAGAN [2] and MAFFT [16] sequence aligners, FastTree [17] tree inference method and *Hide gaps* tool;
- a tree: *Edit tree* tool;
- both sequences and a tree: CodeML [3] selection model tester.

Adding custom programs to Wasabi is described in the plugins section. The built-in *Hide gaps* tool allows masking, collapsing or removing gap-rich or conserved sequence alignment columns. *Edit tree* tool is useful for fine-grained tree modifications, including collapsing/removing specific taxa, adding annotations (e.g. branch colors) or preparing the tree for running CodeML branch-site models [18]. The *Settings* window contains (depending on the Wasabi configuration) up to 30 adjustable preferences, including autosave, color schemes for visualization and the user account management. Many of the options are concealed in collapsed sections that, like elsewhere in Wasabi interface, are marked with triangle-shaped text bullets. The hidden content can be revealed (or re-hidden) with a mouse click on the bulleted text line.

Each time the user saves an imported dataset (*Data* → *Save*) or runs an analysis program (*Tools* menu), a data snapshot is stored to the user account. Together with other snapshots this forms a workflow track in the *Analysis library*. If the input dataset was imported from an external source, the snapshot is stored as the first step of a new workflow; otherwise the save location can be set either to continue or to branch off from the input analysis step. The stored analysis histories are listed in the *Analysis library* window. The analysis step currently open is marked with a white background while read-only shared analyses have dashed borders (see step 8 in the tutorial). A click on the arrowhead button of any analysis step reveals the subsequent step on the analysis path, while a click on the breadcrumb path bar or the back button takes a step towards the root. The *Ladderized* and *Compact* layout modes in the library window, found under the gear button or *Tools* → *Settings*, are useful when the list of stored analyses grows. Each analysis step includes info fields that can be revealed by clicking the black triangle. Some info fields can only be read by hovering its icon or title text (e.g. the date stamp from the clock icon, or the launch parameters from the program name), while others can be clicked and edited (e.g. the descriptive name) or open further options (a click on the dataset ID allows accessing the stored files). One can remove an analysis step with its *Modify* button or import the default output file by clicking *Open* (one can switch the default file by opening it in the file list). In addition, the info icon allows displaying and editing a free-text annotation and the link icon shows the dedicated sharing link. While Wasabi automatically creates a workflow of subsequent analysis steps, root level analysis steps can be collected into larger analysis collections by dragging them by their left side and dropping onto other analyses.

When the imported dataset includes both a phylogenetic tree and sequences with matching taxon names, sequences in the alignment area are displayed next to their position in the phylogenetic tree. Ancestral sequences are hidden by default but can be revealed via drop-down menu by clicking any tree node. The menu also gives access to the node metadata and allows modifying the connected subtree (show/hide/remove/recraft/reroot). Ancestral nodes can also be dragged to relocate them or to remove specific clades. Individual annotation labels and coloring displayed on the tree can be defined in the *Settings* window or modified with the *Edit tree* tool. Specific sequence alignment columns can be masked, collapsed or removed by dragging a selection box spanning the chosen sites, followed by the selected task in the right-click menu. Collapsed sequence sites are indicated with red markers on the ruler bar running along the top edge of the alignment box. The ruler also serves as a dragging handle for panning around the alignment (as an alternative to using arrow keys, mouse scroll wheel or the scrollbars).

In addition to the visualization and built-in tools, the user interface wraps complex functionality like the analysis database and background processes that are perhaps best described with a practical analysis workflow.

3 Example workflow

3.1 Introduction

This tutorial is significantly updated and expanded version of the workflow published in the original Wasabi article [5] that verified the findings of a study [19] linking snow leopard's high altitude adaptation to amino-acid changes in the hypoxia-related gene EGLN1. In short, multiple sequence alignment of EGLN1 is created by merging homologous sequences from Ensembl database with the study data, then realigned with two alternative methods, cleaned, and finally tested for signals of positive selection. Every analysis step with intermediate results is automatically visualized, stored to analysis history, and accessible through the web via sharing URLs.

Although the tutorial includes a detailed list of steps to cover most of the tools and functionality available in Wasabi, in practice the workflow is fairly simple and straightforward: the introduction video featured on the Wasabi homepage fits most of the process in less than 2 minutes (see Fig. 2 for overview). Also, you can pick and choose individual tutorial steps to form shorter tasks. For example, Wasabi is often used for quickly visualizing a sequence alignment file by dropping it to the importer (see step 1), as an interactive tree editor (step 5), versatile file converter (step 2), or to make a dataset available across the web (step 8).

3.2 Setup

Open the Wasabi application by clicking the launch button on http://wasabiapp.org (or go directly to http://was.bi). When visiting Wasabi for the first time (or using the web browser in incognito mode), the *Create account* notification will show up on the top toolbar. Wasabi user accounts allocate a 100MB server space for storing datasets and running background jobs. You can dismiss the notification when using Wasabi for just visualizing, editing and exporting datasets. For enabling full functionality (used in the tutorial from step 2 onward), click the notification button and fill in your email address. Wasabi sends a message to this address when the account is created, about to expire (after 30 days of the last visit), or when a background job has finished (optional). Alternatively, you can opt for a temporary account (valid for 1 day) without entering an email address.

Note that, after clicking *Create account*, Wasabi's web address has changed (to the form was.bi/yourUserID). This address is a direct link to your Wasabi user account and allows you to open your analysis library on any internet-connected device. Please keep the address for future reference, as otherwise you will lose the access to your stored datasets after the Wasabi window has been closed. Here are some suggestions for storing and retrieving your account URL:

- Write the address down or bookmark it in the web browser.
- Enable "Remember me on this computer" in the confirmation window. The web browser will automatically redirect to the account address on subsequent Wasabi launches.
- Locate the link in the email message that Wasabi sent you when the account was created.

In addition to the user account, the tutorial assumes that you have a query file ready to be used in step 4. Download it from http://wasabiapp.org/download/wasabi/other/EGLN1_bigcats.fas.

3.3 Instructions

Step 1: Import EGLN1 gene sequences

First, open the import tool (*Data* → *Import*) to download a GeneTree [20] dataset with EGLN1 homologs (see Note 1). In the Ensembl section, choose "Gene tree" from the left-hand selection, click *Import options*, type human and EGLN1 to the *species* and *gene name* fields, choose cDNA, and click *Get ID*. When the GeneTree ID appears to the top input field, click *Import*. After a brief moment, the tree and sequences are rendered to the visualization area and the import window disappears.

Step 2: Reduce the dataset

In this step, the set of included EGLN1 sequences is reduced to mammalian species. Locate the most recent common ancestor of the mammals clade by hovering the mouse cursor over the tree inner nodes until the label displays *Mammals* (see Note 2). Click the node to reveal a pop-up menu. Select *Remove nodes* → *Keep only subtree*.

Optional: Store a data snapshot

At any point during the tutorial, the currently open dataset can be browsed, modified, downloaded in a desired file format (*Data* → *Export*), or stored to the *Analysis Library* (*Data* → *Save*). Although optional (the next step stores the current data as input), a snapshot of the current dataset is handy for a couple of reasons: it serves as the root step for the following analysis pipeline, and gets a dedicated URL for sharing it in the web and to other Wasabi accounts (see step 8). When a background job is run (tutorial steps 3, 4 and 7), an analysis snapshot (including input, output and metadata) is automatically added to the analysis history in the library. Whenever the dataset state is not stored to the database, it's indicated on the toolbar *(unsaved).* The undo button allows reversing data edits up to the last snapshot.

Step 3: Realign with PRANK

Since the end goal of the workflow is to study positive selection, it's recommended to realign the mammalian EGLN1 sequences with an aligner designed for evolutionary analyses. Click *Tools* → *PRANK aligner* and type a descriptive name for this analysis step (e.g. "Prank realignment"). Next, open the *Alignment options* → *Fine tuning* section and tick *align as codons*. Click *Start alignment*. Click the notification button on the toolbar to check the status of the running background jobs. When the PRANK alignment has finished, click *Open* to import the results.

Optional: Realign with MAFFT

The EGLN1 sequences could also be aligned with another aligner to create an alternative starting point for the rest of the workflow (see Note 3). This would be useful to e.g. estimate the sensitivity of the positive selection tests to the choice of the alignment method. Start the MAFFT realignment (*Tools→MAFFT aligner*) with default settings and let it run in the background. This will create two independent realignments (PRANK and MAFFT version) from the same input dataset, creating a branching point in the analysis path. Go straight to step 4 to complete the rest of the tutorial, then return here to continue with the alternative analysis branch. Wasabi will take care of recording both workflows.

Unlike PRANK, MAFFT does not output a guidetree with taxa names matching the output alignment. Since the next tutorial step needs a reference phylogeny, a new tree needs to be built for the MAFFT alignment. Make a new tree with *Tools→FastTree*, open the resulting dataset and continue with the next tutorial step.

Step 4: Add more sequences

Next, the EGLN1 alignment is extended with homologs from the species studied in the snow leopard paper (tiger, lion and snow leopard). Click *Tools→PAGAN aligner* and drag the EGLN1_bigcats.fas file (from the setup step) to the query file drop area. Edit the name field for a better description. This setup will use PAGAN to extend the currently open alignment with the sequences from the query file. After clicking *Start* *alignment*, the notification button and *Status overview* window will show the progress of the PAGAN execution (see Note 4).

Step 5: Cleanup

After the PAGAN alignment has finished, open the results and then browse the imported alignment. Remove low-quality sequences (showing long stretches of missing data) and paralogs (species duplicates) by dragging the taxa name out of the tree and release it to the trashbin-marked alignment area (or click taxa name → *Remove leaf*). The placement of the added sequences depends on the reference alignment quality and sequence similarity. Check the location of the big cats and recraft if needed: drag and drop to the leopard/lion ancestral node to the cat branch. Next, trim out gappy alignment columns (uninformative for the following selection tests): click *Tools* → *Hide gaps*, adjust the sequence rows threshold to a low value (e.g. 4%) and click *Apply*. Then, right-click the alignment area → *Remove hidden columns* to delete the collapsed gap sites. If you rearranged the tree, the alignment does not match it anymore and needs to be updated. Click the *Realign* notification on the toolbar, tick *use codon model*, and click *Update alignment*. After the realignment, the imported dataset will update itself and a data snapshot is added to the analysis library workflow path.

Step 6: Browse

Click *Tools* → *Translate* and select *codons*. Browse the sequence alignment and locate the AAG→ATG substitution (Lys→Met, at around site 40 on the alignment ruler) in the snow leopard (see Note 5). EGLN1 has been annotated as a hypoxia-related gene and the source study linked this amino-acid change to snow leopard's adaptation to high-altitude environment. The conclusion is supported by the identical substitution at the same alignment position in the alpaca. (Alpaca was not part of the original study but was included here from the GeneTree.) A quantitative confidence score can be added to this finding with a positive selection test.

Step 7: Test for positive selection

The site-wise positive selection test implemented in CodeML calculates (amongst other metrics) an estimate of selection (d_N/d_S ratio) and its confidence score (p-value) for each site in the input multiple sequence alignment. The results file, however, is excessively detailed and makes eyeballing for relevant info a time-consuming task (especially when processing multiple genes). In addition, the validity of the column scores needs to be confirmed by comparing the overall fit of the positive selection model to an alternative model. Therefore, Wasabi's toolset includes a script that parses CodeML result files, performs likelihood ratio tests [22] for the compared model pairs and extracts statistically significant column scores. The script (like the rest of integrated programs) can be chained to a pipeline to feed the results from one analysis step (model testing) to the next one (results parsing).

Click *Tools* → *CodeML*. Type a new analysis step name. Select *Multiple site ratios*. Click *Edit options*, select *single ratio* for the branch model (see Note 6). Expand *Site model* and tick models 1 (nearly neutral) and 2 (positive selection). Now, without closing the CodeML window, open *Tools* → *CodeML tester* (or *Add a step* → *CodeML tester*). Note that the CodeML section is collapsed and the parser program is added as a second step, forming a pipeline. Optionally, tick "*send an email when the pipeline finishes*". This setup will run CodeML to test the two selected models against the input alignment data and to calculate the site scores, followed by the CodeML results parser. Click *Run pipeline*.

Step 8: Review and publish

Follow the pipeline progress in the *Status overview* window and click the *Open* button once it appears. This time, the results are not in the imported tree and alignment (that originate from step 5), but in the CodeML parser report file. To access it, open *Analysis library* and locate the active analysis step (marked with white background and labeled with the name from step 7). Click the analysis ID to see the list of files, hover model_tests.csv and click the revealed *View* button. The file content is opened in a separate window. If everything went as planned, the report is expected to indicate that the data support the presence of positive selection (model M2 passed the hypothesis test) at the previously noted substitution site (dn/ds ratio >1 with p-value <5%).

At this point, you have verified the Lys→Met change in EGLN1 gene from the snow leopard study and improved the confidence of the finding by including more sequence data and running a statistical test. You have also collected a detailed record of the analysis process. You can go back and examine each step in the *Analysis library*, visualize the intermediate results, check the analysis program parameters and input files, or split the analysis path to alternative branches (see step 3 for an example branching point).

To share your findings with the academic community, click on the blue link icon on any workflow step in the library. The resulting *Share data* window gives you the option to either share only the output dataset of the selected step, or together with the subsequent workflow. You can then distribute the displayed sharing URL, e.g. via email or social media (see Note 7). The link will launch Wasabi on any internet-connected device, open the shared dataset and (if enabled) will add the included workflow as a read-only copy to the recipient's *Analysis library*. This allows the recipient to view and work with the received datasets without affecting the source, store the modifications and send back the updated version via another sharing URL.

In addition to online communication, the sharing links can be used in scientific publishing. For example, the Wasabi article [5] includes an image of the EGLN1 alignment with the snow-leopard specific substitution, accompanied by a sharing URL (http://was.bi?id=usecases). When the reader clicks the link, Wasabi is launched, visualizing the EGLN1 alignment at the same position as depicted in the figure. The reader can then browse the rest of the alignment and all of the steps in the analysis workflow. A neat attribute of the shared workflow is that when we modify it (perhaps to fix an error), the distributed copies are also automatically updated in the recipient libraries. Similar sharing link, together with representative dataset and annotations, can also be created for our tutorial workflow. Start with an empty analysis collection (*Analysis library* → gear button → *New collection*). Drag the workflow (e.g. its root step) into the new collection. Click the collection's sharing icon and set the data snapshot from step 6 as the default dataset in the "*Upon import...*" selection menu. Update the annotation (click the collection's info icon) with a free-text description and links to reference articles [19]. Your tutorial workflow is now ready and wrapped, and should look quite similar to our version: http://was.bi?id=tutorial.

4 Advanced topics

4.1 Under the hood

Wasabi is built upon modern web technologies, consisting of the main application (written in Javascript), a server component (written in Python) and third-party programs (plugins). The modular design allows for cross-platform support (including mobile devices), different setup configurations and extensibility. For example, Wasabi can be launched on a local computer as a desktop application, run on a server computer to provide a web service, or integrated into an existing web page.

Wasabi is installed by downloading its files from http://wasabiapp.org/downloads. The application can be used without the server module by opening the index.html file: this allows for import, visualization, editing and export of datasets. The full functionality with external tools and the analysis library is enabled by launching the server script (wasabi_server.py). Wasabi is then available from the server's local address (by default http://localhost:8000). When Wasabi is running on a computer reachable from the internet, it can be provided as a web service, allowing for quick access, data sharing and central updates. Wasabi's server module uses job scheduling and user accounts for managing system resources and randomized IDs for data security. Wasabi running modes, user account quotas and other application parameters can be edited in the server module settings file (wasabi_settings.cfg).

The analysis library uses a file-based database, where the folder structure represents analysis paths with metadata stored in meta.txt files. This allows direct reading or writing to the analysis library in a file browser or on the command line. An existing data folder can be defined as an analysis library in the settings file. The Wasabi interface only shows folders with a meta.txt file (that, at the minimum, should include the ID and name fields).

The Wasabi application communicates with its server component and the outside world with URL commands an be used to integrate Wasabi with other tools and websites. For example, a command-line pipeline can use wget [23] to retrieve or write files to a remote analysis library, or open a web browser to visualize a dataset with locally installed Wasabi. Also, Wasabi links are an easy way to add a visualizer to a web-based alignment database. The URL parameters in the links can be used to provide a customized visualizer, e.g. to disable specific functions, display bootstrap values or hide the toolbar. See http://wasabiapp.org/rest for documentation and examples.

Since Wasabi is a web application, it can be added to an existing web service like any other web page: by including Wasabi's HTML, CSS and javascript files, and linking to the Wasabi's URL where needed. In addition, Wasabi's appearance and functionality can be extensively customized by editing the style.css and script.js files. Examples of web services with integrated and customized Wasabi include the Silva rRNA database (http://www.arb-silva.de), the ConSurf conservation profile database (http://consurfdb.tau.ac.il), and the Ensembl genome browser (http://ensembl.org).

In comparison to a native compiled application, a downside of a web-based implementation is the performance cost arising from the web browser overhead. Wasabi uses several optimization strategies to scale well even with large input datasets. For example, the sequence data is rendered to static but graphics-card accelerated canvas elements that are expanded piece-by-piece as new alignment regions are scrolled into the viewport. Since the imported dataset is loaded to memory, the maximum dataset size is limited mainly by the RAM available in the computer. We have tested Wasabi on a regular laptop with up to 1-gigabyte data files, showing performance on par with native visualization programs [5]. A favorable side effect of the web browser environment is that, with the continuous development of javascript engines, the performance of Wasabi improves over time even without contributions from the future code optimizations.

4.2 Plugins

Wasabi utilizes a plugin system to integrate external tools to its graphical interface. A plugin is a JSON-formatted [24] description of a command-line program that Wasabi uses to communicate with the program and to construct a graphical user interface for launching the tool. Fig. 3 depicts a JSON specification for a python script with two input parameters. The script is integrated to Wasabi by dropping it together with the JSON file to the plugins folder (see Note 8). The command-line program now appears in the *Tools* menu and has gained a graphical interface. The interface also allows chaining the plugins to form analysis pipeline that can be stored together with the filled parameters for later reuse. As demonstrated by the CodeML interface, the plugin API [25] allows building complex interfaces with embedded instructions, alternative parameter sets, user input-dependant default values, etc. The JSON files for current Wasabi tools are found in the plugins folder and the full API documentation is available at http://wasabiapp.org/plugins.

5 Future directions

Wasabi has grown from its beginnings as a capable cross-platform sequence alignment visualizer to an extendable analysis platform for evolutionary sequence analyses, with a distinctive user-friendly interface and easy access across the web. While there is no shortage of directions for improvement, some features planned for the upcoming Wasabi updates include a substring search, image file export, reference sequence support, and a visualization track for plotting site-wise graphs or annotations. Other improvements under consideration aim to reduce Wasabi's memory footprint and simplify integration to web content.

Wasabi's plugin system allows using standardized program descriptions for adding command-line tools to the graphical interface, significantly reducing the workload and know-how required for implementing the integration. Although the plugin system was built for Wasabi, it could be useful for many other programs and provide them with a dynamic graphical interface. To help in that, Wasabi's plugin system is now provided as a separate javascript library, the universal interface generator Pline. With that, one can quickly build graphical interfaces to command-line tools and use them on any web page or as a standalone desktop application. Each plugin file should only be written once, and then updated together with the target program. To that end, an online repository for sharing and reusing JSON files is available on the Wasabi homepage.

Notes

1. This procedure imports a dataset from Ensembl database. The same input dataset could have been fetched in the import tool via other routes, e.g. by dragging a prepared datafile (perhaps from a previous run of this tutorial) to the file drop area, typing http://rest.ensembl.org/genetree/member/id/ENSG00000135766 to the bottom section input field (this address uses Ensembl REST API [21]), or by filling the same input with the dataset ID g7TDxl to import a copy from our Wasabi account.
2. Dragging the tree/alignment divider to the right and using the vertical scrollbar will help to browse big trees.
3. An analysis path can be split at any time from a selected step in the *Analysis library*. For example, if you stored the input dataset in step 2 (you can also do it now since the dataset is still open), you can skip the branching option for now and return to this step later. For that, open the data snapshot (from step 2) in the library window and continue the tutorial with the MAFFT alignment step. Likewise, you don't have to finish the tutorial in one go. You can close and later relaunch Wasabi, open a stored analysis step and resume from there, or even switch between the alternative analysis paths while completing the remaining steps.
4. You can terminate a running background process with the *Kill* button. A terminated (or failed) process will show a *Delete* button: clicking that removes the generated files. You can also skip the *Open* button and move the files directly to the *Analysis library* for inspection (click the gear icon → *Move to library*). Also,

hovering the *program name* will show its launch parameters and clicking the blue *feedback* text line reveals the full output log.

5. After you have located the substitution site, it’s a good idea to save a data snapshot with the *zoom level* and *alignment position* enabled in the *Store visualization* settings section. Next time you (or a colleague using the sharing URL from step 8) opens the stored dataset, the alignment will automatically move to the correct position without having to spend time to relocate the substitution.
6. Hover the mouse over an input field to see the associated command-line parameter. The adjacent info icon or dotted text reveals the relevant documentation.
7. If you are using Wasabi datasets in a scientific publication, make sure that the sharing links will stay permanently accessible. The public Wasabi (http://wasabiapp.org) is an academically funded, free-of-charge service. At the time of writing, we provide free user accounts and keep user data for a minimum of 30 days but we cannot guarantee long-term storage of external datasets. However, you can easily install Wasabi locally (in a server configuration) to share your datasets permanently.
8. It's recommended to add a copy of the target program to the plugins folder instead of using a system-wide command, since future program updates may break the plugin compatibility.

References

1. Löytynoja A (2014) Phylogeny-aware alignment with PRANK. Methods Mol Biol 1079:155–170

2. Löytynoja A, Vilella AJ, Goldman N (2012) Accurate extension of multiple sequence alignments using a phylogeny-aware graph algorithm. Bioinformatics 28:1684–1691

3. Yang Z (2007) PAML 4: phylogenetic analysis by maximum likelihood. Mol Biol Evol 24:1586–1591

4. Paten B, Herrero J, Beal K, et al (2008) Enredo and Pecan: genome-wide mammalian consistency-based multiple alignment with paralogs. Genome Res 18:1814–1828

5. Veidenberg A, Medlar A, Löytynoja A (2016) Wasabi: An Integrated Platform for Evolutionary Sequence Analysis and Data Visualization. Mol Biol Evol 33:1126–1130

6. Kumar S, Stecher G, Li M, et al (2018) MEGA X: Molecular Evolutionary Genetics Analysis across Computing Platforms. Mol Biol Evol 35:1547–1549

7. Huerta-Cepas J, Dopazo J, Gabaldón T (2010) ETE: a python Environment for Tree Exploration. BMC Bioinformatics 11:24

8. Larkin MA, Blackshields G, Brown NP, et al (2007) Clustal W and Clustal X version 2.0. Bioinformatics 23:2947–2948

9. Baum BR (1989) PHYLIP: Phylogeny Inference Package. Version 3.2. Joel Felsenstein. The Quarterly Review of Biology 64:539–541

10. Felsenstein J (2004) Inferring Phylogenies. Sinauer Associates Incorporated

11. Zmasek CM NHX - New Hampshire eXtended, version 2.0. http://phylosoft.org/NHX/. Accessed 19 Aug 2019

12. Maddison DR, Swofford DL, Maddison WP (1997) NEXUS: an extensible file format for systematic information. Syst Biol 46:590–621

13. Löytynoja A HSAML format. http://wasabiapp.org/software/hsaml_format/. Accessed 19 Aug 2019

14. Han MV, Zmasek CM (2009) phyloXML: XML for evolutionary biology and comparative genomics. BMC Bioinformatics 10:356

15. Zerbino DR, Achuthan P, Akanni W, et al (2018) Ensembl 2018. Nucleic Acids Res 46:D754–D761

16. Katoh K, Standley DM (2013) MAFFT multiple sequence alignment software version 7: improvements in performance and usability. Mol Biol Evol 30:772–780

17. Price MN, Dehal PS, Arkin AP (2010) FastTree 2--approximately maximum-likelihood trees for large alignments. PLoS One 5:e9490

18. Zhang J, Nielsen R, Yang Z (2005) Evaluation of an improved branch-site likelihood method for detecting positive selection at the molecular level. Mol Biol Evol 22:2472–2479

19. Cho YS, Hu L, Hou H, et al (2013) The tiger genome and comparative analysis with lion and snow leopard genomes. Nat Commun 4:2433

20. Vilella AJ, Severin J, Ureta-Vidal A, et al (2009) EnsemblCompara GeneTrees: Complete, duplication-aware phylogenetic trees in vertebrates. Genome Res 19:327–335

21. Yates A, Beal K, Keenan S, et al (2015) The Ensembl REST API: Ensembl Data for Any Language. Bioinformatics 31:143–145

22. Yang Z, Nielsen R, Goldman N, Pedersen AM (2000) Codon-substitution models for heterogeneous selection pressure at amino acid sites. Genetics 155:431–449

23. GNU Wget. https://www.gnu.org/software/wget/wget.html. Accessed 19 Aug 2019

24. The JSON Data Interchange Syntax. http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf. Accessed 19 Aug 2019

25. Wasabi plugin API. http://wasabiapp.org/about/wasabi-plugin-api/. Accessed 19 Aug 2019

Figure captions:

Fig. 1 Wasabi with an imported analysis library dataset. Here, some ancestral sequences have been revealed, the *Tools* menu is open and the *Analysis library* shows the imported analysis step together with its workflow path, annotation text and the file list.

Fig. 2 Overview of the example workflow, represented by Wasabi interface cutouts. Circled numbers indicate corresponding tutorial steps.

Fig. 3 A minimal example of a plugin JSON file (top) and the resulting Wasabi interface window (bottom).