

User manual for Phylobook

Contents:

- Overview
- Installation and Configuration
- Using Phylobook – viewing and annotating trees.
- Uploading data to Phylobook using the built-in pipeline
- Preprocessing data and making it available within Phylobook

Overview:

As the volume of sequence data from variable pathogen genomes increases, means of analyzing, annotating and extracting specific taxa and taxon groups for study becomes more difficult. To meet these challenges for moderately large datasets (e.g., hundreds to thousands of taxa) the “Phylobook” tool was developed. Phylobook displays phylogenetic tree data adjacent to highlighter plots showing the position of mutations in the alignment. A key feature of Phylobook is that it allows the user to identify lineages and recombinants within a given dataset, annotate them and then export selected subsets of sequences for downstream analysis. Lineage identification can be aided using annotations created by one or more clustering methods.

Installation and Configuration:

Phylobook is distributed as a Docker container. A container is a standard unit of software that packages code and all its dependencies so that the application runs quickly and reliably from multiple computing environments, including Mac, Windows and Linux systems. Docker containers run under a “Docker engine”, a virtual machine that is available on most common operating systems.

- 1) Install **Docker** on your host server: <https://docs.docker.com/get-docker/>. Note that **Docker Compose** is an additional tool that is needed and it is automatically included with Mac and Windows downloads of Docker. However, Linux users will need to add it manually. You can do this by running the following command after your Docker installation is complete.

```
sudo pip install docker-compose
```

(pip = python installer)

- 2) Install Git on your computer.
 - a. On a Mac, if you have previously installed the developer tools, Git is already installed. If you have not previously installed the developer tools you can obtain Xcode from the app store and run the installer.

- b. On Windows, git can be obtained at <https://gitforwindows.org/>
 - c. On Linux, Git is specific for the specific version of Linux but most common options can be found at <https://git-scm.com/download/linux>.
- 3) Create 2 new folders. One is the folder into which Phylobook will be installed, the other is the folder that will contain the data that Phylobook uses. These folders can be anywhere on the host computer.
- 4) Install Phylobook. In a terminal window, navigate to the install folder (/Applications/Phylobook in this example) and type the command:

```
git clone https://github.com/MullinsLab/phylobook.git
```

This will install phylobook in the install directory. If you are on a mac and used the example folder name of /Applications/Phylobook, this step will create a folder named /Applications/Phylobook/phylobook and it will populate it with the phylobook software.

- 5) Create an environment file. Change directory to the newly created phylobook folder by typing

```
cd phylobook
```

Create the .env file by copying the .env.Template file to .env using the command:

```
cp .env.TEMPLATE .env
```

Note: On most computers, files that start with a “.” are hidden from display by default so if you wish to see or edit the file with a GUI based editor (like textedit), you will need to tell your computer to show hidden files (the Mac OS command in the folder you wish to display hidden files is: cmd + shift + “.”). If you are comfortable with a command line editor, you can edit .env in the terminal window.

- 6) Configure Phylobook by editing the .env file for your own environment. Phylobook can be set up to run locally (e.g., just on your laptop or desktop) or it can be run as an internet connected server. As an internet connected server, the login can be configured to be either local, Single Sign-On (aka, “SSO”), or dual, i.e., local and SSO.

Note: SSO authenticates users using an institutional or corporate user authentication service and is most useful if every intended user belongs to the same organization. SSO must be configured to connect to your organization’s Open Authorization server and configuration will likely require assistance from a local IT professional. Local authorization will require a username and password for each user and those usernames/passwords will be local to the Phylobook server. Dual login allows for a mix of local usernames/passwords and SSO. This is most convenient when some of the users are from the same organization and other users are collaborators from external organizations. Since SSO configuration is institution specific, the following describes configuration as a local server.

If you are comfortable with a command line editor, you can edit .env in the terminal window.

Some lines (e.g., DEBUG, SERVER_NAME, and EMAIL_*, should only be edited by developers or to change passwords). The lines that need to be edited in the .env file are:

- SECRET_KEY=PUT_A_UNIQUE_PASSWORD_HERE

The password can be anything the user desires. If you need help generating a strong password, a number of websites can be used for that purpose. For example, either <https://djecrety.ir/> or <https://randomkeygen.com/> can be used to generate a strong password. The secret key is used by the Django web framework.

- DJANGO_ALLOWED_HOSTS=localhost 127.0.0.1

If phylobook will be run local to your computer or from behind a reverse proxy such as NGINX or Apache (not covered here), you can leave this line unchanged. Otherwise, it should be completed with the server and domain name.

- PROJECT_PATH=

The PROJECT_PATH directory is the primary storage location for project data within Phylobook. PROJECT_PATH should be set to the desired location of the data directory. In our mac example, this might be /Applications/Phylobook/Data

- DB_USER=postgres
- DB_NAME=postgres
- DB_PASS=

In most cases, the DB_USER and DB_NAME can be left as PostgreSQL. However, if you have another server that is your research group's primary database server AND you wish to store the database outside of the Docker container, you may wish to configure Phylobook to use that alternate database. We do not recommend this. DB_PASS should be set to a unique password. Other configuration changes may be needed for your specific environment but just those mentioned above are sufficient to get Phylobook running on your local computer.

- EMAIL_HOST=outlook.office365.com
- EMAIL_USE_TLS=1
- EMAIL_PORT=587
- EMAIL_HOST_USER=
- EMAIL_HOST_PASSWORD=

Passwords can be changed on the Admin panel (the "(Admin)" link at the top of each Phylobook page) by the administrator. In addition, Phylobook has the ability to send password reset emails if a user has forgotten their password. In order for this to work, the .env file needs to contain valid email settings, including the login credentials called for above. Collecting these credentials is outside the scope of this manual.

Note: Although not recommended, but if you chose LOGIN_TYPE=SSO or LOGIN_TYPE=dual in config.env, then you must create settings/saml.py by copying settings/saml.py.TEMPLATE :

```
cp settings/saml.py.TEMPLATE settings/saml.py
```

Edit settings/saml.py and add your institution's SAML configuration and certificates:

```
nano settings/saml.py
```

7) Build and deploy the docker containers by issuing the following command:

```
docker compose up -d --build
```

After you have issued that command, you should see phylobook running under docker (viewable in the Docker Desktop), e.g., see **Figure 1**. If this does not appear automatically, simply launch the Docker app.

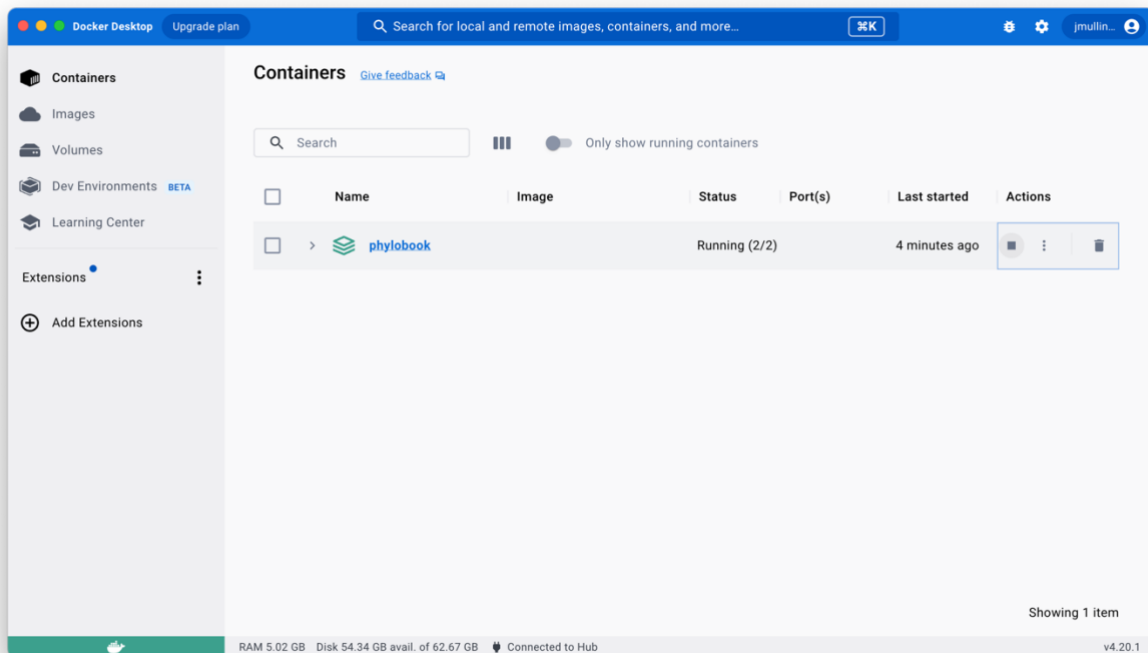


Figure 1. Docker desktop.

Note: Depending on the location of the PROJECT_PATH directory, you may need to make the directory available to Docker as a shared folder. Do this by going to Settings > Resources > File Sharing in the Docker desktop and adding the directory as shown in **Figure 2**.

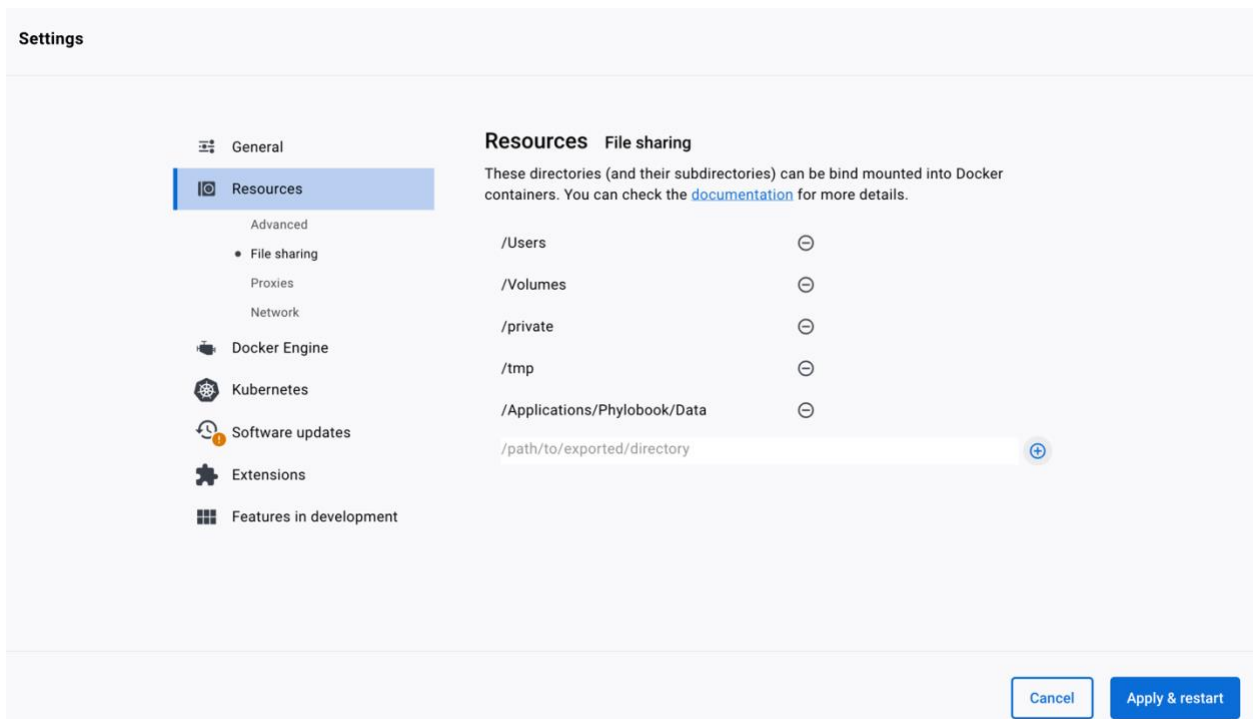


Figure 2. A view of Docker Desktop showing how to share a local directory with a Docker container.

8) The following command will perform initial database migrations:

```
docker exec -it phylobook python manage.py migrate --settings=phylobook.settings.prod
```

9) The following command will create the initial superuser/admin for the phylobook system:

```
docker exec -it phylobook python manage.py createsuperuser --settings=phylobook.settings.prod
```

After executing this command you will be asked for a username and a password. These credentials will be used for the initial login to Phylobook.

10) In a web browser, login to the server <http://localhost:8000/> using your super user account credentials . In the upper right corner of the browser window, there is an "Admin" link. Click the link and you will enter the Administrator module.

11) Prior to using Phylobook for analysis, data files must be placed in the PROJECT_PATH directory and Phylobook. Later in this manual we will discuss how to generate data files for use in Phylobook. For now, we will show how to setup Phylobook using sample data provided as part of the Phylobook installation. Sample data files are provided as .tar.gz files – open these files to expand into folders.

Note: Multiple datasets may exist in each project folder. Each dataset will be represented by multiple files that start with the dataset name and that end with names associated with the

type of file. While the Phylobook pipeline produces a large number of files (**Figure 3**), only three files are required for each project, with additional files being optional (**Table 1**).

▼ HIV_DEMO	🕒 Today at 12:11 PM	--	Folder
▼ run_phymI_logs	🕒 Today at 12:11 PM	--	Folder
run_phymI.log	🕒 Mar 2, 2023 at 1:45 PM	892 bytes	Log File
DEMO.log	🕒 Mar 2, 2023 at 1:45 PM	891 bytes	Log File
DEMO.phy_phymI_tree.txt.svg	🕒 Mar 2, 2023 at 2:43 PM	38 KB	SVG image
DEMO.json	🕒 Mar 2, 2023 at 2:43 PM	322 bytes	JSON Document
DEMO_highlighter.png	🕒 Mar 2, 2023 at 1:45 PM	545 KB	PNG image
DEMO_highlighter.fasta	🕒 Mar 2, 2023 at 1:45 PM	176 KB	BEdit Document
DEMO_highlighter.txt	🕒 Mar 2, 2023 at 1:45 PM	21 KB	text
DEMO_highlighter_untrimmed.png	🕒 Mar 2, 2023 at 1:45 PM	556 KB	PNG image
DEMO.phy_phymI_tree.txt_nexus.tre	🕒 Mar 2, 2023 at 1:45 PM	5 KB	FigTree tree file
DEMO.phy_phymI_tree.txt_newick.tre	🕒 Mar 2, 2023 at 1:45 PM	2 KB	FigTree tree file
DEMO.phy_pwcoldist.txt	🕒 Mar 2, 2023 at 1:45 PM	43 KB	text
DEMO.phy_log.txt	🕒 Mar 2, 2023 at 1:45 PM	231 bytes	text
DEMO.phy_phymI.txt	🕒 Mar 2, 2023 at 1:45 PM	81 KB	text
DEMO.phy_phymI_tree.txt	🕒 Mar 2, 2023 at 1:45 PM	2 KB	text
DEMO.phy_phymI_stats.txt	🕒 Mar 2, 2023 at 1:45 PM	3 KB	text
DEMO.phy	🕒 Mar 2, 2023 at 1:45 PM	172 KB	Alignment file
DEMO.fasta	🕒 Mar 2, 2023 at 1:43 PM	174 KB	BEdit Document
DEMO.cluster.km9clusters	🕒 Dec 7, 2022 at 4:06 PM	656 bytes	Document
DEMO.cluster.km8clusters	🕒 Dec 7, 2022 at 4:06 PM	656 bytes	Document
DEMO.cluster.km7clusters	🕒 Dec 7, 2022 at 4:06 PM	656 bytes	Document
DEMO.cluster.km6clusters	🕒 Dec 7, 2022 at 4:06 PM	656 bytes	Document
DEMO.cluster.km5clusters	🕒 Dec 7, 2022 at 4:06 PM	656 bytes	Document
DEMO.cluster.km4clusters	🕒 Dec 7, 2022 at 4:06 PM	656 bytes	Document
DEMO.cluster.km3clusters	🕒 Dec 7, 2022 at 4:06 PM	656 bytes	Document
DEMO.cluster.km2clusters	🕒 Dec 7, 2022 at 4:06 PM	656 bytes	Document
DEMO.cluster.km10clusters	🕒 Dec 7, 2022 at 4:06 PM	657 bytes	Document
DEMO.cluster.GAPclusters	🕒 Dec 7, 2022 at 4:06 PM	656 bytes	Document
▼ WA_SARS_CoV2Spike	🕒 Today at 11:13 AM	--	Folder
▼ run_phymI_logs	🕒 Today at 11:13 AM	--	Folder
run_phymI.log	🕒 Mar 1, 2023 at 12:35 PM	1 KB	Log File
WA_SARS-CoV2_SpikeAA.log	🕒 Mar 1, 2023 at 12:35 PM	1 KB	Log File
WA_SARS-CoV2_SpikeAA.phy_phymI_tree.txt.svg	🕒 Mar 1, 2023 at 2:52 PM	105 KB	SVG image
WA_SARS-CoV2_SpikeAA.json	🕒 Mar 1, 2023 at 2:52 PM	489 bytes	JSON Document
WA_SARS-CoV2_SpikeAA_highlighter.png	🕒 Mar 1, 2023 at 12:36 PM	766 KB	PNG image
WA_SARS-CoV2_SpikeAA_highlighter.fasta	🕒 Mar 1, 2023 at 12:36 PM	187 KB	BEdit Document
WA_SARS-CoV2_SpikeAA_highlighter.txt	🕒 Mar 1, 2023 at 12:36 PM	14 KB	text
WA_SARS-CoV2_SpikeAA_highlighter_untrimmed.png	🕒 Mar 1, 2023 at 12:36 PM	786 KB	PNG image
WA_SARS-CoV2_SpikeAA.phy_phymI_tree.txt_nexus.tre	🕒 Mar 1, 2023 at 12:36 PM	11 KB	FigTree tree file
WA_SARS-CoV2_SpikeAA.phy_phymI_tree.txt_newick.tre	🕒 Mar 1, 2023 at 12:36 PM	5 KB	FigTree tree file
WA_SARS-CoV2_SpikeAA.phy_pwcoldist.txt	🕒 Mar 1, 2023 at 12:35 PM	488 KB	text
WA_SARS-CoV2_SpikeAA.phy_log.txt	🕒 Mar 1, 2023 at 12:35 PM	280 bytes	text
WA_SARS-CoV2_SpikeAA.phy_phymI.txt	🕒 Mar 1, 2023 at 12:35 PM	494 KB	text
WA_SARS-CoV2_SpikeAA.phy_phymI_tree.txt	🕒 Mar 1, 2023 at 12:35 PM	6 KB	text
WA_SARS-CoV2_SpikeAA.phy_phymI_stats.txt	🕒 Mar 1, 2023 at 12:35 PM	2 KB	text
WA_SARS-CoV2_SpikeAA.phy	🕒 Mar 1, 2023 at 12:34 PM	183 KB	Alignment file
WA_SARS-CoV2_SpikeAA.fasta	🕒 Mar 1, 2023 at 12:33 PM	185 KB	BEdit Document
readme	🕒 Jun 29, 2023 at 10:37 AM	2 KB	Document

Figure 3. Pipeline output files present in the HIV_DEMO and WA_SARS_CoV2Spike folders.

Table 1. User files used by Phylobook

File type	Requirement	File naming convention	Source
Image of phylogenetic tree	Required	DatasetName.phy_phymI_tree.txt.svg	Pipeline
Image of highlighter plot	Required	DatasetName_highlighter.png	Pipeline
Alignment FASTA	Required	DatasetName_highlighter.fasta	Pipeline
Cluster Assignments	Optional	DatasetName.cluster.clusters	Clustering algorithm https://github.com/MullinsLab/ClusteringForPhylobook

12) *Making a project visible within the phylobook server:* Once data is available in the PROJECT_PATH directory, the Admin tool in Phylobook is used to make that project available within the system. After logging in (step 11 above), you will see an "Admin" link in the upper right corner of the browser window. Click the link and you will enter the Administrator module (**Figure 4**). The admin module allows an administrator to add projects, project categories, users, and groups to the system.

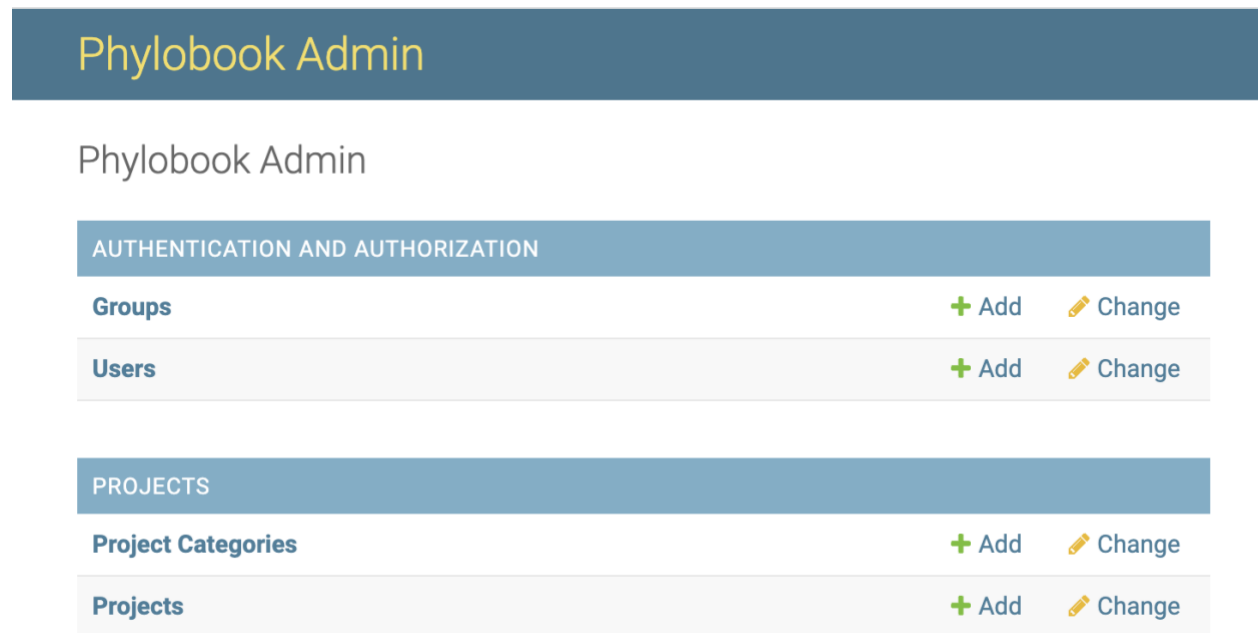


Figure 4. The Phylobook Administration module.

To add a project to Phylobook, click on the Add link next to “Projects” and type the EXACT name of the folder in the Project path directory. For example, to make the sample folder HIV_DEMO available you would enter it in the dialog below and SAVE (**Figure 5**).

Add project

Name:

HIV_DEMO

Category:



Save and add another

Save and continue editing

SAVE

Figure 5. Adding a project to the Phylobook server

Follow the same process to add the WA_SARS_CoVSpikes dataset. Once added a project will appear in the Projects list in the admin module and on the landing page within Phylobook.

For users who have a large number of projects stored within Phylobook, the landing page can become a bit cluttered with a very long list of projects. The concept of "Project categories" allows projects to be grouped on the landing page by user defined categories, which then behave as subfolders on the landing page. Note that this only affects the display on the landing page for Phylobook. All projects remain in the PROJECT_PATH directory.

After a project has been created, permissions for that project can be assigned for users within the admin module. Only administrators will see all projects.

- 13) **Setting Access Permissions:** Permissions are managed at the level of a project, with all project contents having the same permissions. Once created, users and groups can be assigned permissions within the "Object Permissions" for any given Project object. This is performed by clicking a Project in the list of Projects and then clicking "Object Permissions".

Select the user or group to which you want to assign permissions and click the "Manage user" or "Manage group" button.

Set the permissions for a user or group. Phylobook uses the default permissions provided by a custom permission object in Django. Of these permissions, only two matter to Phylobook: "Can change project" and "Can view project". If a user or group has "Can change project" permission, then they are able to have access to the full set of annotation tools and can edit and save changes. Users or groups with "Can view project" permission are only able to view the projects contents and have no ability to save any new information. Save any permission changes and then click "VIEW SITE" in the upper right corner to return to the main Phylobook site project list.

- 14) **Other configuration considerations:** Phylobook can have fairly large POST requests when dealing with large samples. E.g. the webserver may need to ship/receive large amounts of data. If you are using a reverse proxy server it should be configured to accept POST requests of at least 2M. Note: A reverse proxy server, also known as a surrogate server, is a server

that acts as an intermediary between a client and web servers. It appears to the client as a web server, but instead forwards the client's requests to one or more web servers. Reverse proxy servers are often put into place by systems administrators to a network's performance, security, and reliability. Hence you may need to work with your local systems administrator in order to assure that the reverse proxy server is configured in a way that will allow Phylobook to operate properly.

Using Phylobook – viewing and annotating trees

Initial viewing and navigation of the data:

Assuming you have followed the installation and configuration steps above, you should have 2 projects within Phylobook – HIV_DEMO and WA_SARS_CoVSpikes. Log in to Phylobook and select the project HIV_DEMO. You should see a display that looks like the image in **Figure 6**.

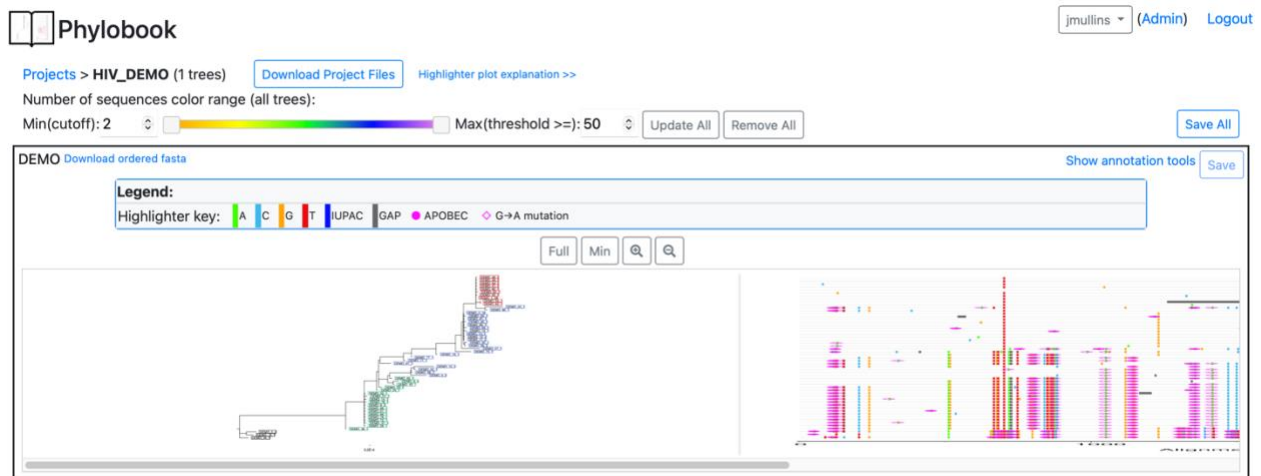


Figure 6. Initial view of the HIV_DEMO project.

Clicking on the “Full” button will expand the view. The magnifying glass The upper left corner shows the current project being displayed and the number of trees within that project. If multiple trees are present in the project, multiple panels similar to that in **Figure 6** will be displayed. Each panel corresponds to a dataset within a project consisting of a set of sequences represented by a phylogenetic tree and a highlighter plot. The highlighter plot shows sequence variation relative to a master sequence. The master sequence is marked with a (m) just to the right of the sequence names in the highlighter plot. Highlighter plots are created using the Los Alamos National Laboratories (LANL) Highlighter tool found at https://www.hiv.lanl.gov/content/sequence/HIGHLIGHT/highlighter_top.html/.

Just to the right of the project name is the “Download Project Files” button. This will download all files associated with this project. This can be used to archive data at a given state of analysis or to share data with other Phylobook installations. The demo data provided through our GitHub site was downloaded from our own Phylobook server. Just to the right of the Download Project Files button is a link to LANL’s page that explains the highlighter application.

Within the phylogenetic tree, sequence names may be surrounded by colored boxes that indicate sequence groupings, e.g., lineage designations. As will be discussed later, groupings can be edited within Phylobook and sequences within a group can be exported for further analysis.



Figure 7. View of the HIV_DEMO project after pressing the “Full” button and selecting “Show annotation tools”.

Annotations

Annotation tools can be shown or hidden by clicking on the Show/Hide Annotation tools text in the upper right of the window. Trees can be labeled by abundance of sequences within the population, proposed lineage groups, information encoded within the sequence name (such as date or tissue type) and individual markers can be placed on the tree and associated with a notes field. The annotation tools availability is context specific. For example, the “annotate sequence boxes by cluster” tool will only appear when clustering data is available and the “Color sequence names by field” tool will only appear when the sequence names have fields available.

Labeling sequences by abundance within a population:

When sequences are produced from a population of a pathogen (say HIV in this example), it is common that many sequences are identical to each other. Our standard practice is to collapse identical sequences into one sequence prior alignment and creation of the phylogenetic tree. When we do this, the sequence names are modified to end with an underscore followed by the number of identical sequences that were collapsed into this particular sequence ID. For example in the DEMO_HIV data, the 8th sequence down on the tree (DEMO_1_28) is the result of collapsing 28 identical sequences into one ID – DEMO_1. The default in Phylobook is that all sequence names end with _n where n is the number of sequences that were collapsed into

each sequence ID. At the top of the project page is a tool labeled “Number of sequences color range (all trees):”. This tool can be used to apply a colored square just to the left of each sequence name that was created from collapsing identical sequences. The color of the square indicates the number of sequences that were collapsed. Using this tool, ALL of the trees within a given project can be annotated with the same color range. Within each individual sample, a identical tool is available. This tool only labels the tree for that sample. **Figure 7** shows the DEMO tree after the sequences have been marked by abundance within the sample. Note that after this annotation was added, the sample is marked by a red border. This indicates that the newly created annotation has not yet been saved. Hitting the save button will store the annotation and change the color of the border back to black. Annotation is stored within the SVG file that represents the tree.



Figure 7. Sequences marked by abundance and annotation tools shown. For each sequence in the tree that represents more than one sequence in the dataset, a small colored square has been placed to the left of the sequence name in the tree. The color of the square indicates the abundance of the sequence in the original dataset. Sequences with no square represent a single sequence in the dataset. A red border indicates that the changes have not yet been saved.

Showing/hiding annotation tools:

Each sample within a project can be annotated independently of other samples in the project. In the up righthand corner of the border for each sample is a link labeled “Show annotation tools”. Clicking on the link will show the annotation tools that are available for that sample and will change the link to “Hide annotation tools”. Figure 7 shows the annotation tools that are available for the DEMO dataset within the HIV_DEMO project. Note that more or fewer

annotation tools will be available depending on the nature of the data. For example, as we will see later, the tree can also be labeled by data that is encoded within the sequence name when such information exists. In the DEMO dataset, that information is not present and hence that annotation tool is not available. Similarly, the “Annotate sequence boxes by cluster” tool is only available when clustering information is available in the dataset.

Labeling sequences by potential lineages/clades using clustering data:

For the DEMO dataset, the “Annotate sequence boxes by clusters” is available as clustering data is in the project directory. One of the primary applications of Phylobook is to create and store individual lineages within a dataset. Lineage annotation can be created in a semi-automated fashion with an external algorithm that assigns sequences to groupings or “clusters”. When data is loaded into Phylobook, one or more initial lineage designations can be provided for each dataset. These consist of files in the dataset with names in the format of DataSetName.cluster.AlgorithmName. For example in the DEMO dataset in the HIV_DEMO project, there are files named DEMO.cluster.km2clusters, DEMO.cluster.km3clusters, etc. Each of these is a text file that contains sequenceID, LineageID for all of the sequences in the dataset (where lineage ID is an integer). In the DEMO example, the lineage designations of type “km#clusters” were created by running a kmedoid algorithm on the nucleotide distance matrix for all samples in the dataset. Kmedoid was run with “K”= 2-10 and a lineage designation file was saved for each value of K. Clustering will be discussed in more detail later in the manual.

When such files are available, the “Annotate sequence boxes by clusters” tool will be available and the pulldown menu “Cluster” will contain the algorithm name (which is parsed from the file name). For example, **Figure 8** shows the pulldown menu with “km7clusters” selected. Selection of this algorithm annotates the tree with colored triangles at the end of each sequence name. The color of the triangle corresponds to an initial lineage designation. Hitting the “Apply to Labels” button will transfer these tentative lineage designations to boxes that correspond to saved lineage designations in Phylobook. **Figure 9** shows the result of this process.



Figure 8. Using the “Annotating sequence boxes by cluster” tool. In this example, the km7clusters algorithm has been selected and each sequence in the tree is labeled by a colored triangle at the end of the sequence name. The triangle color corresponds to the initial lineage designation.



Figure 9. The DEMO data set after km7cluster designations have been Applied to Labels. To simplify the image, the cluster designations (triangles in Figure 8) have been turned off by selecting “None” in the clustering pulldown menu and the annotation tools have been hidden. Note the change in the lineage designations (indicated by the colored boxes around the sequence names in the tree).

After applying the K-medoid designations to the data, lineages (sequence groupings) are indicated by the colored boxes around the sequence names in the trees. For the most part, selecting km7 makes biological sense as the sequence groupings correlate mostly with what appears by eye to be separate lineages. For example 9 of the first 10 sequences have been grouped into one lineage as indicated by the red boxes around the sequence names. One sequence (DEMO_56_1) has been left out of this grouping and inspection of the highlighter plot shows a large gap in this sequence relative to others in the grouping. Then there is a grouping of sequences indicated by the green boxes that appears to be a separate lineage but there are two sequences (DEMO_25_1 and DEMO_66_1) that appear to be recombinants. At the bottom of the tree, there are two unique lineages (lime green and light blue boxes). In between, there are a number of sequences that appear to be recombinants of the other lineages. In such complicated trees, automating lineage assignments via clustering (or other algorithms) is likely to result in some errors so Phylobook provide a mechanism for manual editing of lineages.

Manual editing of lineages:

Lineages can be manually edited by dragging the mouse to highlight sequence names in the tree that one wishes to assign to a different lineage. Figure 10 shows the menu that pops up when

this is done. In this example DEMO_56_1 has been selected. By clicking on the red box, it will be assigned to the same lineage as the other sequences labeled by red boxes.

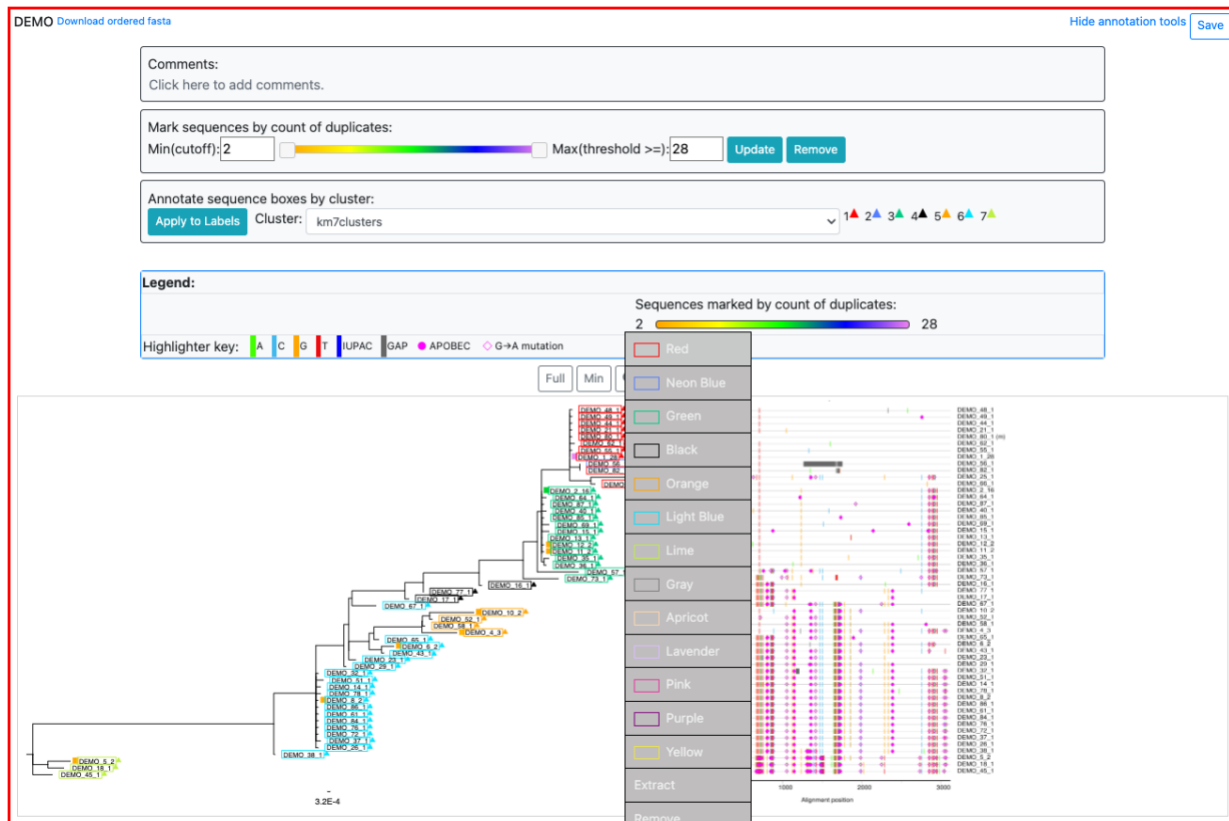


Figure 10. Manual editing of lineage designations. Dragging the mouse around (a) sequence name(s) in the tree brings up a box that allows one to assign the selected names to a new or different lineage.

Figure 11, shows the result of additional manual editing to place all likely recombinant sequences in a separate “lineage” designated by orange boxes. In addition, Figure 11 shows the ability to add comments to the dataset.

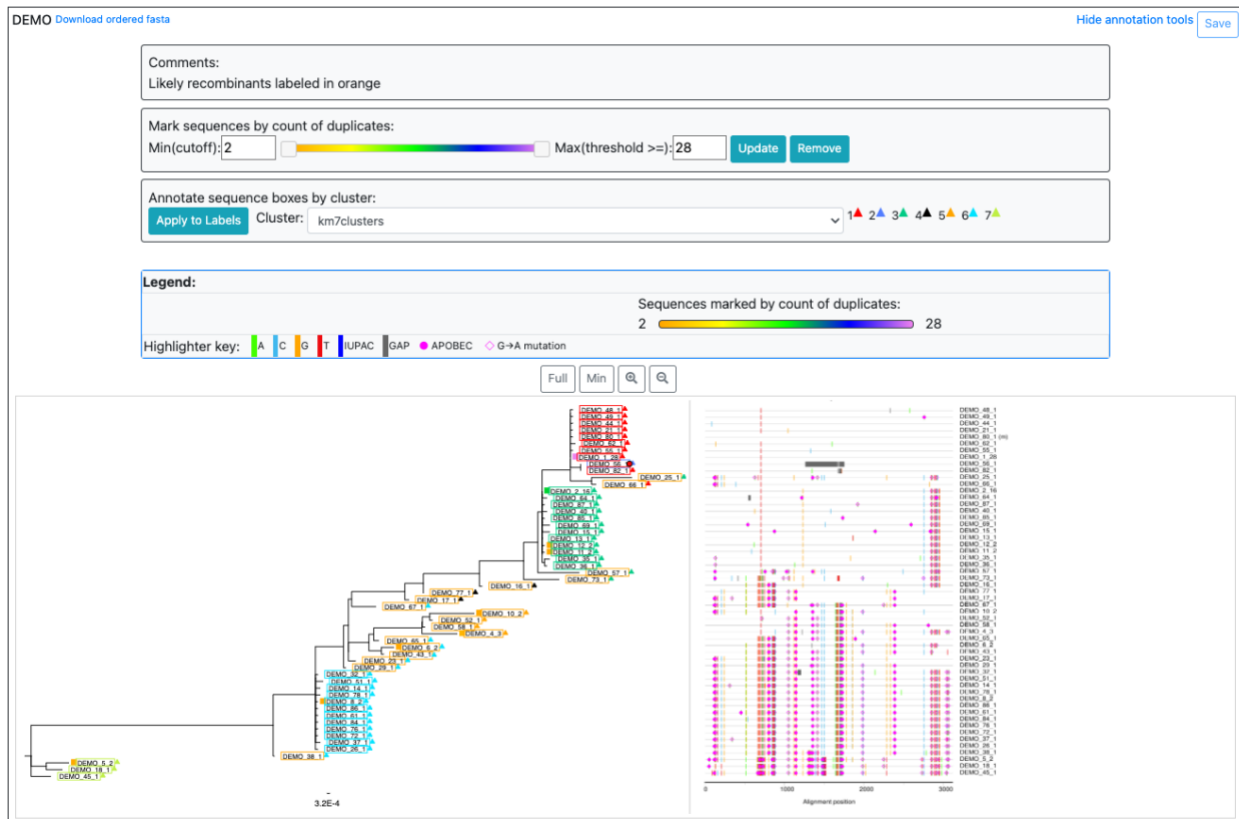


Figure 11. The DEMO dataset after manual editing of lineage designations. Note that a comment has been added to indicate that lineage designations in Orange are sequenced deemed to be likely recombinants by the editor.

Exporting sequences in lineage groups:

After sequences have been assigned to lineage groups, they may be exported for downstream analysis. **Figure 10** shows the menu used to manually assign sequence groups to lineages. At the bottom of that same menu is a button labeled “Extract”. Clicking on Extract pulls up the menu shown in **Figure 12**. Extracted sequences can be copied and pasted into another application for downstream analysis.

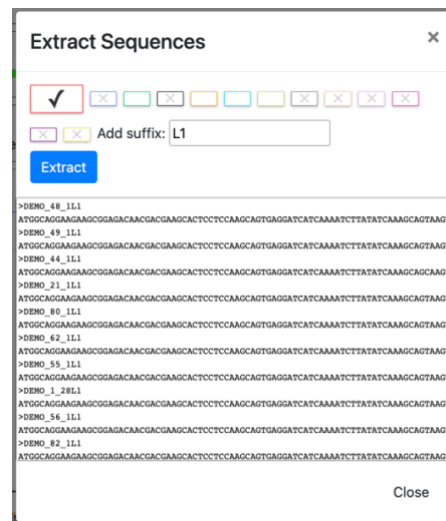


Figure 12. The menu for extracting sequences associated with selected lineages. In this case, the “red” lineage was selected and the system was told to add “L1” (for lineage 1) to the sequence names. The extract button was clicked and the window was populated with the selected sequences that have been renamed as requested. These sequences can then be copied and pasted into another application for downstream analysis.

Annotation by fields encoded in the sequence names:

In some cases it is useful to annotate the tree to indicate associations with other parameters such as time, tissue source, partner relationships etc. Phylobook provides for annotation using additional data encoded within the sequence name. In this annotation method, the font color of the sequence names in the tree is colored based on the values in an annotation field. As was discussed above, by default we encode the number of identical sequences in a sample in the sequence name with an underscore n at the end of the sequence name where n is the number of sequences that were collapsed into each sequence in a dataset. Additional information can be encoded within the sequence names delimited by additional underscores.

Phylobook parses the sequence names in each dataset to identify fields that are delimited by underscores and counts the number of variants for each field. For each field that has variation in it where the number of different variants is ≤ 10 , Phylobook allows the user to designate a font color for each variant. Since it is difficult to visually distinguish more than about 10 font colors, the feature is limited to annotations with 10 or fewer variants. For example, in the WA_SARS-CoV2_SpikeAA sample data set, the sequence names contain information about the date of sampling. In this data set, the sequence names have the form SequenceID_x_Year_Month_n. The Year field contains the values 2020, 2021, 2022 and 1 (1 is the Wuhan reference sequence which was collected in 2020) and the month field contains values that run from 1-12. In such a dataset, an additional annotation tool becomes active (see **Figure 13**). Fields that can be used to color sequence filenames are highlighted in teal boxes.

Color sequence names by field:
MW813275_1_2021_3_1

Figure 13. Annotation by values encoded in the filename. When appropriate fields in the filename exist, Phylobook will identify fields that are delimited by underscores and identify the number of variants within each such field. In cases where the number of variants is ≤ 10 , the annotation tool “Color sequence names by field:” will appear followed by a representation of a sequence name - MW813275_FieldA_FieldB_3_1 in this example. In this case FieldA and FieldB are highlighted with teal boxes indicating that either field can used to color the sequence filenames.

Clicking on an eligible field in the sequence name representation brings up a dialogue box similar to that shown in **Figure 14**. In this case, we have selected the field year values (except for the reference sequence which is from 2020). We have then used the pulldown menus in the dialog box to assign the black font to all sequences from 2020, the magenta font to all sequences from 2021 and the light blue font to all sequences from 2022. The net result of this color assignment is show in figure 15. As expected sequences from 2022 (mostly omicron variants) cluster together at the bottom of the tree while sequences from 2020 (mostly similar to the early Wuhan sequences) cluster near the top of the tree and 2021 sequences are mostly in the middle of the tree.

Annotate sequence names for: MW813275_1_1_3_1

Value	Color
1	black
2020	black
2021	magenta
2022	light blue

Close Color Sequences

Figure 14. The dialogue box used to assign colors to sequence names based on a field in the sequence name.



Figure 15. WA_SARS-CoV2_SpikeAA sequences color coded by year. Black corresponds with sequences obtained in 2020, blue → 2021 and orange → 2022. Note: other annotations (lineage designations) present in the sample dataset have been removed in this diagram for clarity.

Annotation with colored dot placed on the tree:

Phylobook also allows the user to place small, colored dots at any location within the tree image. Control clicking within the image brings up the annotation tool shown in Figure 16. The user can select a colored dot and drag it around to place it in any desired location on the tree. When combined with a comment, this can be useful to label specific branches within the tree or other features of interest.



Figure 16. Annotation of the tree with colored dots. Control clicking outside the tree brings up the box shown above. The user can select a colored dot from within this box and place it anywhere on the tree by dragging it to a desired location.

Preprocessing data and making it available within Phylobook:

Now that you have Phylobook installed and have familiarized yourself with its use, the next step is to populate it with your own data. Table 1 (above) contains the files that need to be present for each dataset within a project.

In addition to the files listed in Table 1 one may wish to store additional related files in the Phylobook file system. For example, when a given sample contains multiple identical sequences, we typically collapse the sequences in the fasta file to just the unique sequences and we add an annotation to the sequence names to indicate number of sequences that have been collapsed in the new fasta file. To collapse sequences, we use the `sequence_collapsing` program that can be found at https://github.com/MullinsLab/sequence_collapsing. The input to this program is a fasta file that contains redundant sequences and the output is a new fasta file containing the collapsed sequences PLUS an additional “namefile” that contains the mappings between sequence names in the original fasta file and the collapsed fasta file. We typically upload the namefile to Phylobook to store it with the rest of the data.

There are three ways to populate Phylobook projects with data: 1) Upload the data directly through the web interface, 2) Run the fasta files through our standalone pipeline and upload to the server via sFTP or 3) Run the fasta files through various algorithms of your choosing and then upload the appropriate output to the server via sFTP. For the majority of Phylobook users, we anticipate that the first method will suffice.

Uploading data via the web interface:

The standard install of Phylobook now includes the default data processing pipeline in the docker container (added in June of 2024). This feature makes it easy for end users to upload data. The Projects landing page now contains a button “Upload Project Files” (see below).



Upload Project Files

Projects

Clicking on the “Upload Project Files” button leads to the dialog shown below.

Projects

New Project [Add to Project](#)

New Project Name:

☒ Process .fasta files with the pipeline on upload

Drop files here to upload

[Upload files](#)

If the “Process .fasta files with the pipeline on upload” box is selected, the expected input is fasta files that have been previously aligned. One can either drag fasta files into the region labeled “Drop files here to upload” or one can click in that region. Clicking in that region will bring up the normal file selection tool for the operating system on the user’s computer. Once all the fasta files of interest have been selected, they can be uploaded. Note that files can be uploaded to a new project or an existing project. If files are uploaded to an existing project, access to those files is determined by the prior user/group permissions settings for that project. If files are uploaded to a new project, the pipeline creates a new project but user and group permissions must be set separately through the admin tool.

If the ““Process .fasta files with the pipeline on upload” box is NOT selected, the expected input is the collection of files indicated in Table 1 above and, if desired the namefile from collapsing sequences for each dataset.

Pre-processing data with the standalone pipeline and uploading data via the web interface or sFTP:

Before discussing the use of the standalone pipeline, it’s worth discussing the rationale for it’s continued existence. E.g. why have a standalone pipeline the essentially reproduces what is integrated within Phylobook? There are two primary reasons for this: 1) The pipeline processes (in particular the sequence alignments) can be computationally intensive and might be better performed on a different computer than the server and 2) The components of the pipeline can be potentially replaced with different components by users who wish to use different algorithms. Regardless of how the files are produced, they can be uploaded using the web interface or manually using sFTP to the server.

The Phylobook pipeline can be installed either directly on your computer or it can be installed within a Docker container. We strongly recommend you install it via a Docker container and will provide detailed instructions for that install below.

1. These instructions assume that you have already installed Docker and Git as described above.

2. Create a directory to house the pipeline.
3. In a terminal window, change to the desired directory and execute the command:

```
git clone https://github.com/MullinsLab/phylobook\_pipeline.git
```

This will create a directory called "phylobook_pipeline" in your designated directory.

Note: If you are installing on a Mac or a Unix operating system, you will then need to modify the docker-compose.yml file to have your userid and groupid. This will assure that files created by the pipeline will be associated with your userid and groupid. To make this modification first discover your userid and groupid by issuing the following terminal command

```
id
```

In the example output from this command below, the userid is 501 and the groupid is 20.

```
jamesmullins@iMac phylobook_pipeline % id
```

```
uid=501(jamesmullins) gid=20(staff)
```

```
groups=20(staff),12(everyone),61(localaccounts),79(_appserverusr),80(admin),81(_appserveradm),98(_lpadmin),33(_appstore),100(_lpoperator),204(_developer),250(_analyticsusers),395(com.apple.access_ftp),398(com.apple.access_screensharing),399(com.apple.access_ssh),400(com.apple.access_remote_ae)
```

4. Using a text editor open the docker-compose.yml file and edit line 12 to delete the # sign (making it no longer a comment) and change it to the correct user and group ID.
5. In the terminal, switch to the phylobook_pipeline directory and enter

```
cd phylobook_pipeline
```

6. Now build the new container using the command:
























```
docker compose up -d --build
```

This should create the new docker container for the pipeline. In docker desktop, you should see a display similar to the image below.

Containers [Give feedback](#)

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. [Learn more](#)

☐ Only show running containers

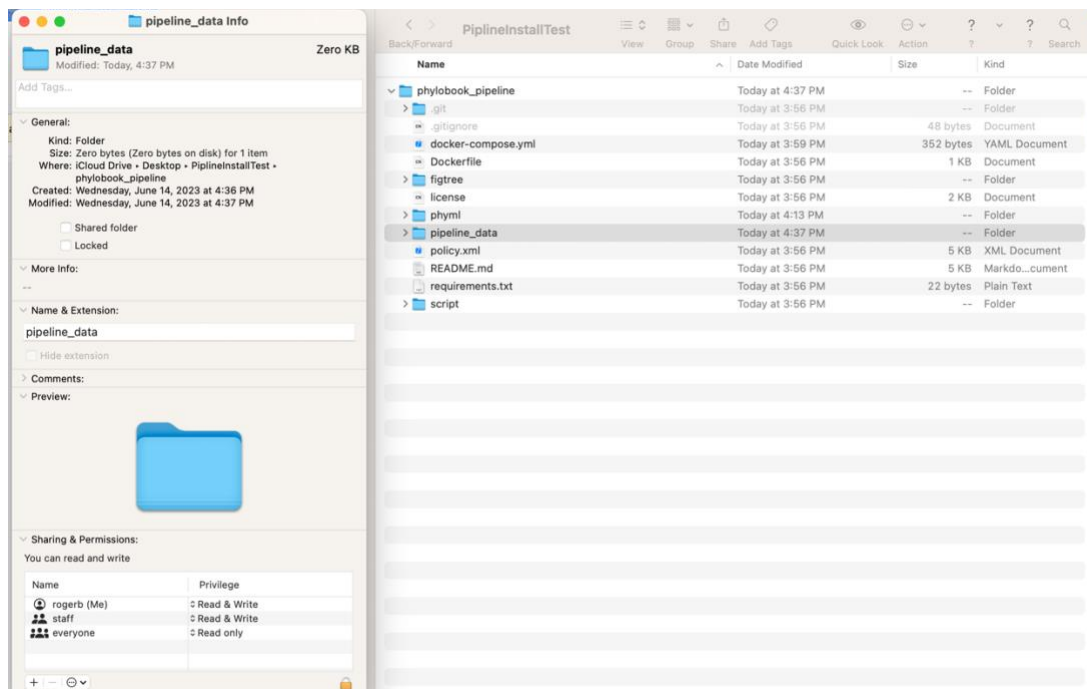
<input type="checkbox"/>	Name	Image	Status	Port(s)	Started	Actions
<input type="checkbox"/>	 phylobook_project	-	Running (2/2)			  
<input type="checkbox"/>	 phylobook 2e7cbe3b1646 	phylobook_project-web	Running	8000:8000	20 seconds ago	  
<input type="checkbox"/>	 db-1 c174f9730e78 	postgres:11	Running		20 seconds ago	  
<input type="checkbox"/>	 phylobook_pipeline	-	Running (1/1)			  
<input type="checkbox"/>	 phylobook_pipeline e31cd71a7275 	phylobook_pipeline-phylobook	Running		12 minutes ago	  

7. With the pipeline running inside the docker container, issue the command:

```
docker exec -it phylobook_pipeline bash
```

This will put you into a terminal inside of the docker and you can operate the pipeline from there. Control-d will return you to the command line on your local computer's OS.

8. The phylobook pipeline operates on a working directory that contains one or more sequence alignment fasta files. Those fasta files must be stored in a directory to which the pipeline has access. Under the default setup, the only portion of the host computer's file system that is visible within the Docker container is the phylobook_pipeline directory (and subfolders therein). Anything that gets created inside the docker's phylobook_pipeline directory will show up in the host machine. To facilitate this, the docker container includes a directory under the phylobook_pipeline directory, pipeline_data. (see image below).



9. Now you must populate the data folder with one or more fasta files on which the pipeline will operated. These fasta files are contain sequence alignments of your own data. A sample data file named `WA_SARS-CoV2_SpikeAA.fasta` comes in the data directory, and can be found at https://github.com/MullinsLab/phylobook_pipeline if it needs to be redownloaded later. Leave this file in the working directory, or replace it with one of your own. For example:

▼ phylobook_pipeline	Today at 4:43 PM	--	Folder
> .git	Today at 3:56 PM	--	Folder
.gitignore	Today at 3:56 PM	48 bytes	Document
docker-compose.yml	Today at 3:59 PM	352 bytes	YAML Document
Dockerfile	Today at 3:56 PM	1 KB	Document
> figtree	Today at 3:56 PM	--	Folder
license	Today at 3:56 PM	2 KB	Document
> phylml	Today at 4:13 PM	--	Folder
▼ pipeline_data	Today at 4:43 PM	--	Folder
WA_SARS-CoV2_SpikeAA.fasta	Mar 1, 2023 at 12:33 PM	185 KB	TextEdit
policy.xml	Today at 3:56 PM	5 KB	XML Document
README.md	Today at 3:56 PM	5 KB	Markdo...cument
requirements.txt	Today at 3:56 PM	22 bytes	Plain Text
> script	Today at 3:56 PM	--	Folder

10. Then issue the command

```
python3 /phylobook_pipeline/script/phylobook.py -d
/phylobook_pipeline/pipeline_data -t aa
```

This command will run the script `phylobook.py` on the fasta files in `/phylobook_pipeline/pipeline_data`. Note that in this example, there was only a single fasta file on which to operate and it contained an amino acid sequence

alignment. If the directory contains (a) nucleic acid sequence alignment(s), then the command should end with nt instead of aa.

11. After the pipeline has run, the folder will be populated with all of the required files for phylobook. In addition, a number of intermediate results files and a log will be created (see image below). These files can now be moved to a project directory folder in phylobook.

> run_phymlog	Today at 5:20 PM	--	Folder
WA_SARS-CoV2_SpikeAA_highlighter_untrimmed.png	Today at 5:20 PM	787 KB	PNG image
WA_SARS-CoV2_SpikeAA_highlighter.fasta	Today at 5:20 PM	187 KB	TextEdit
WA_SARS-CoV2_SpikeAA_highlighter.png	Today at 5:20 PM	768 KB	PNG image
WA_SARS-CoV2_SpikeAA_highlighter.txt	Today at 5:20 PM	14 KB	Plain Text
WA_SARS-CoV2_SpikeAA.fasta	Today at 5:18 PM	183 KB	TextEdit
WA_SARS-CoV2_SpikeAA.phy	Today at 5:18 PM	183 KB	TextEdit
WA_SARS-CoV2_SpikeAA.phy_log.txt	Today at 5:19 PM	292 bytes	Plain Text
WA_SARS-CoV2_SpikeAA.phy_phymlog_stats.txt	Today at 5:19 PM	2 KB	Plain Text
WA_SARS-CoV2_SpikeAA.phy_phymlog_tree.txt	Today at 5:19 PM	6 KB	Plain Text
WA_SARS-CoV2_SpikeAA.phy_phymlog_tree.txt_newick.tre	Today at 5:19 PM	5 KB	Dendro...ocument
WA_SARS-CoV2_SpikeAA.phy_phymlog_tree.txt_nexus.tre	Today at 5:19 PM	11 KB	Dendro...ocument
WA_SARS-CoV2_SpikeAA.phy_phymlog_tree.txt.svg	Today at 5:19 PM	67 KB	Scalabl...s Image
WA_SARS-CoV2_SpikeAA.phy_phymlog.txt	Today at 5:19 PM	494 KB	Plain Text
WA_SARS-CoV2_SpikeAA.phy_pwcoldist.txt	Today at 5:19 PM	488 KB	Plain Text

12. Notes:

- a. The pipeline will operate on all the files in the pipeline data folder sequentially. It assumes that all files are either nucleotide alignments OR amino acid alignments. I cannot handle a mixture of both. It also assumes that ALL files are alignment files. So after running the pipeline, you must move the files out of the pipeline data directory prior to doing another run.
- b. At present, the pipeline does not handle fasta files that have been wrapped at 80 characters. Please assure your files are appropriately formatted.
- c. The highlighter plot uses the first sequence in the fasta alignment file as the reference sequence. E.g. sequence variation relative to the FIRST sequence in the file is shown in the highlighter plot. If the sequences in the fasta file have been created by collapsing identical sequences into representative sequences, out standard practice has been to put the most abundant sequence first in the fasta file.
- d. Several files that are not needed by Phylobook are also produced by the pipeline (for example the distance matrix, the newick tree file etc). We typically store these files in the Phylobook project directory as we occasionally find them useful for other analyses.