

Supplementary Material

Software Requirements:

Python Version 2.7

Biopython Version 1.60

Python Modules: httpLib, urllib2, etc2 (An Environment for Tree Exploration (ETE), Optional)

ARB Version 5.1

Goal: Importing alignment, phylogenetic tree and meta-data from external resources into ARB.

Motivation: Exploratory Analysis and Primer design in ARB.

Dataset: Major Facilitator Superfamily (MFS) Protein Family (440 sequences)

Available from the NCBI Conserved Domain Database (CDD)

<http://www.ncbi.nlm.nih.gov/Structure/cdd/cddsrv.cgi?uid=212314>

MFS: "...a large and diverse group of secondary transporters that includes uniporters, symporters, and antiporters. MFS proteins facilitate the transport across cytoplasmic or internal membranes of a variety of substrates including ions, sugar phosphates, drugs, neurotransmitters, nucleosides, amino acids, and peptides."

Procedure:

1. Download the MFS dataset from CDD. Two files: 1 for alignment and 1 for tree.

<http://www.ncbi.nlm.nih.gov/Structure/cdd/cddsrv.cgi?hs1f=1&uid=cd06174&#seqhrch>

*Note: Set the Type Selection to 'top listed sequences' and the Aligned Rows to All 440 Rows before downloading the alignment file via the Structure View button.

The screenshot displays the NCBI Conserved Domain Database (CDD) interface for the cd06174 Sequence Cluster. The left sidebar contains the 'Structure' section with 'Aligned Rows' set to 'All 440 rows' and the 'Hierarchy' section with 'Interactive Display' selected. The central area shows the 'cd06174 Sequence Cluster' with a phylogenetic tree and a sequence alignment view. The 'Sub-family Hierarchy' section on the right shows 'cd06174 MFS'. The 'Sequence Alignment' section at the bottom has 'Type Selection' set to 'top listed sequences'.

Supplementary Material

2. Download/Install CDTree tool for dataset visualization and extraction of tree.

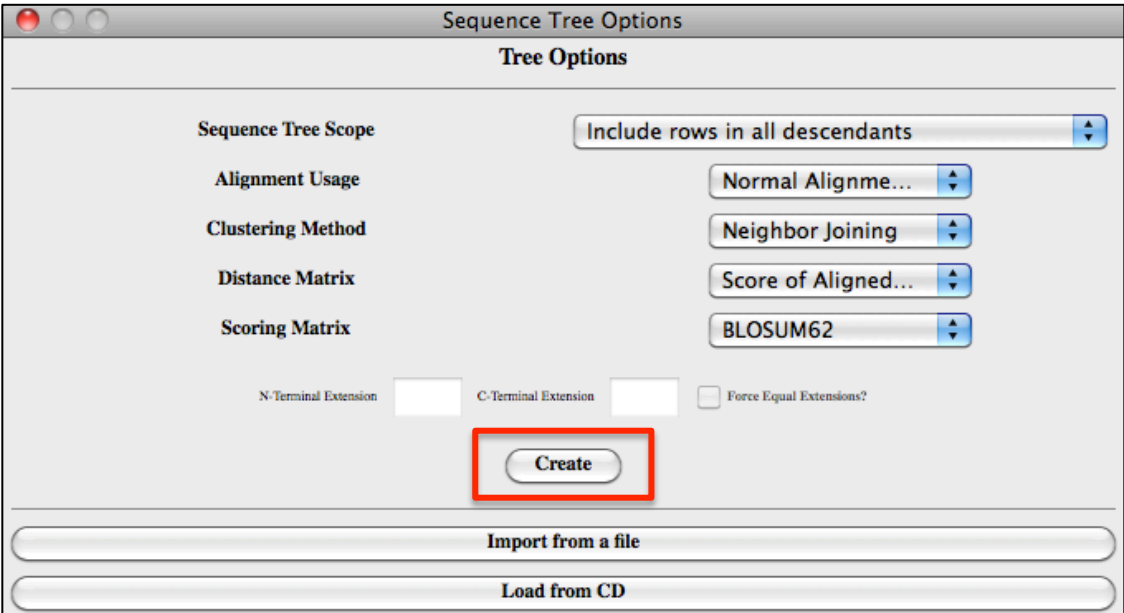
<http://www.ncbi.nlm.nih.gov/Structure/cdtree/cdtree.shtml>

3. Open MFS dataset project (*cddsrv.cn4*). Export MFS phylogenetic tree from CDTree:

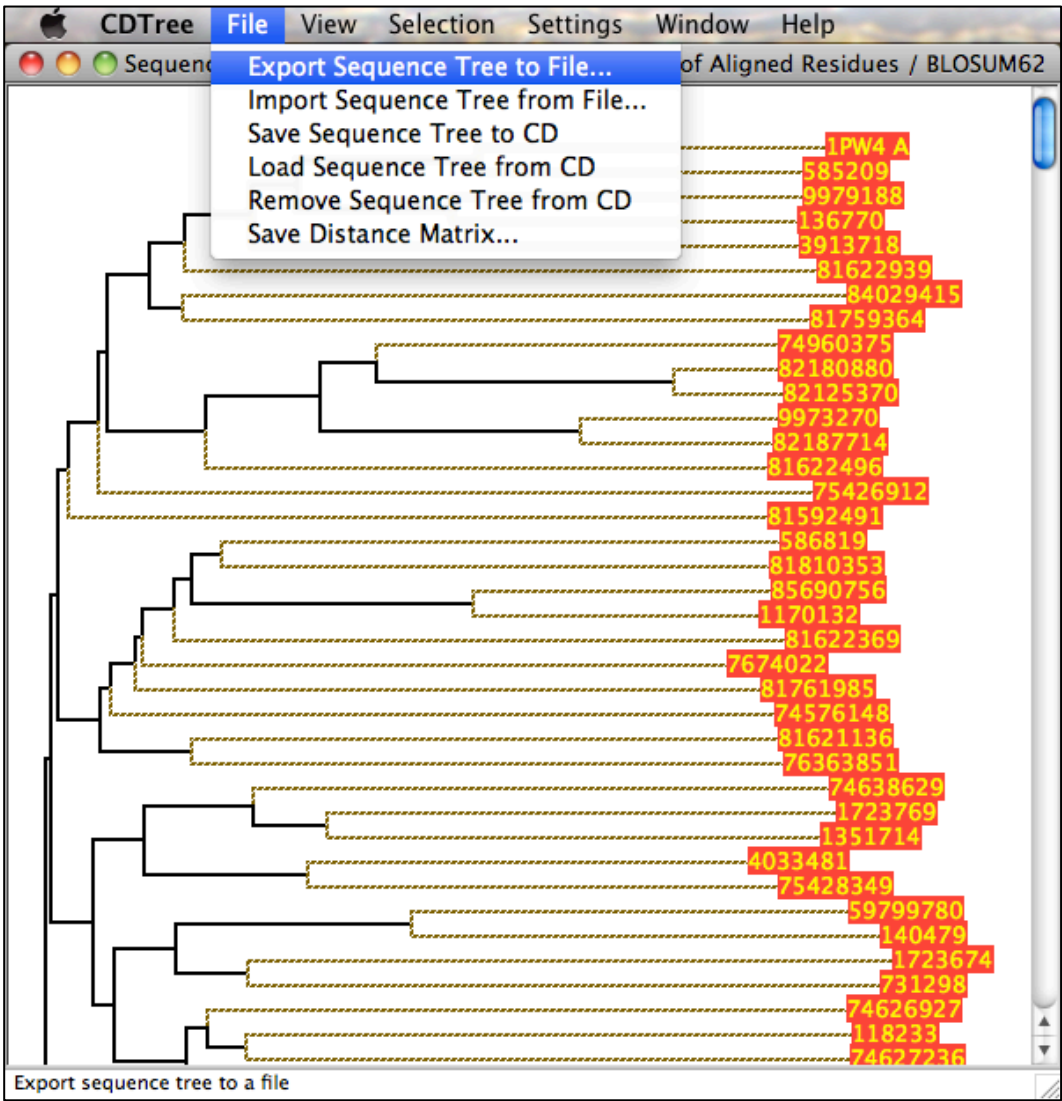
a. Right click/Select Sequence Tree

Supplementary Material

26 b. Create Tree

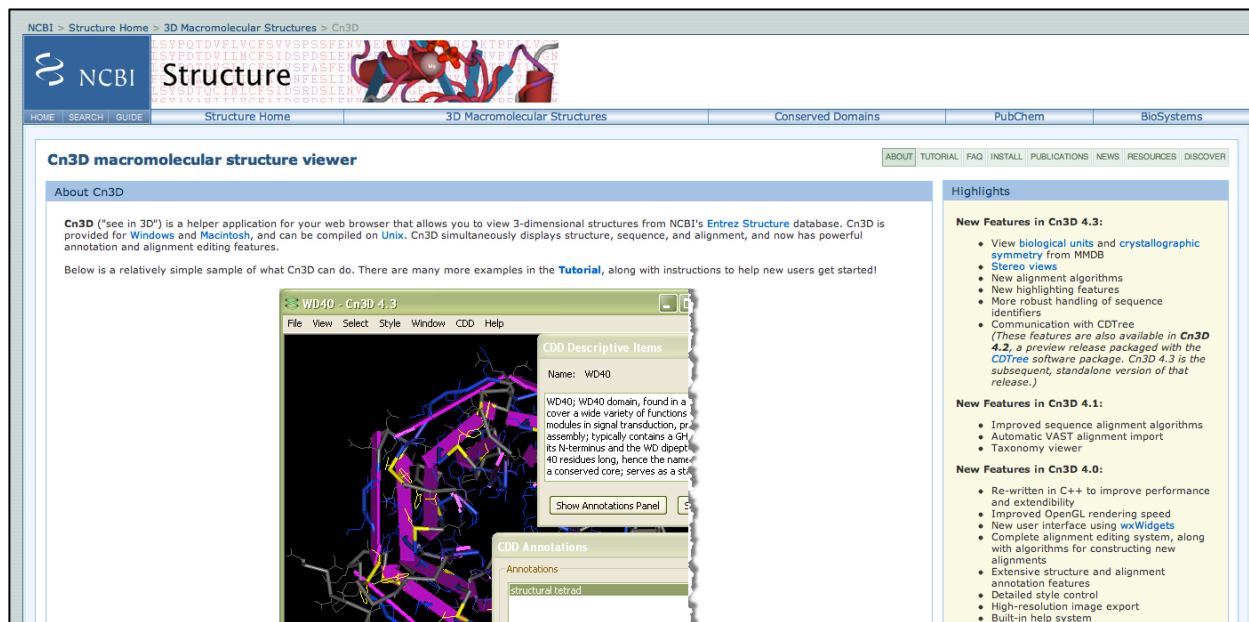


27
28 c. Export Sequence Tree: Save as *MFS.tree*

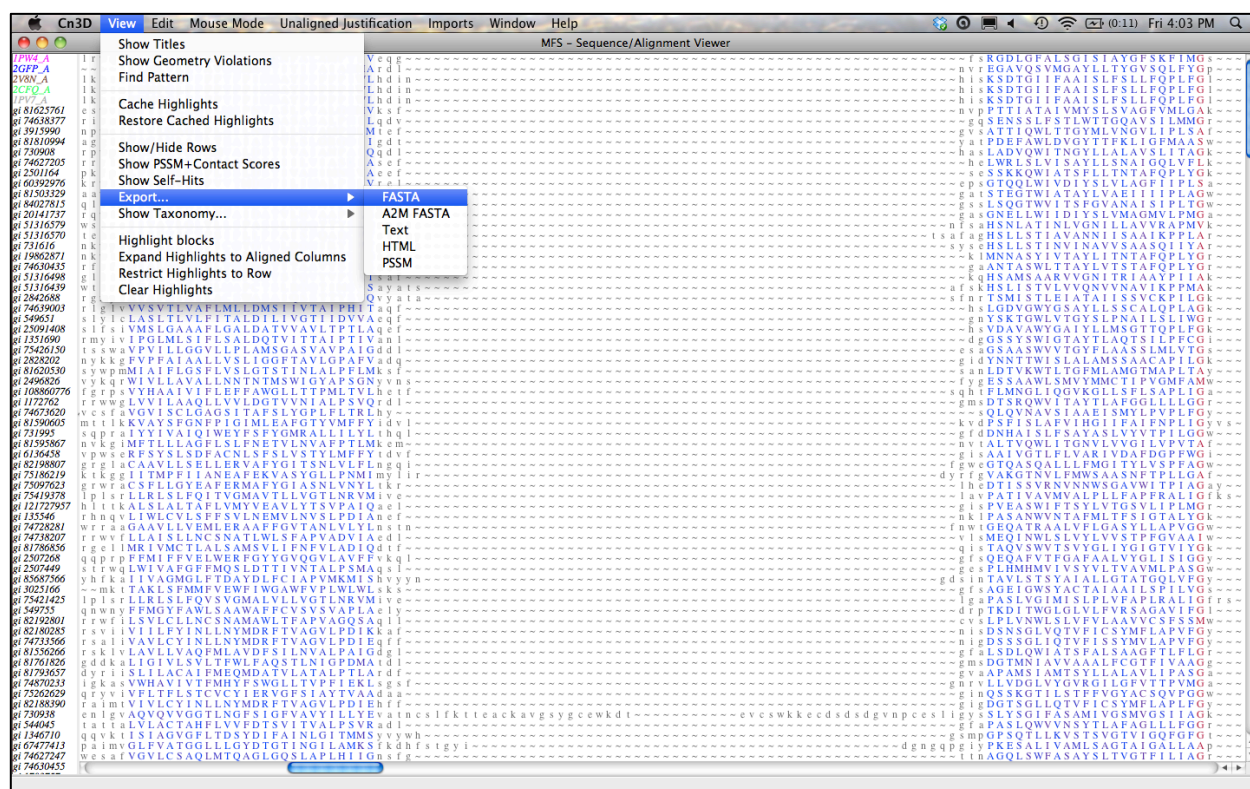


Supplementary Material

- Download/Install/Run Cn3D tool for exporting MFS alignment.
<http://www.ncbi.nlm.nih.gov/Structure/CN3D/cn3d.shtml>



- Open MFS dataset project (*cddsrv.cn3*). Export alignment: Save as *MFS_Align.fasta*



Supplementary Material

6. Assemble the Database -- Collect Meta-Data and Sequences

From a terminal, change the directory to the **GetDatabase** folder. Run the script **buildDatabase.sh** (e.g. `./buildDatabase.sh`) to obtain the GenBank files for each sequence in the dataset. Note that this script may only be run between the hours of 9pm-5am EST as per NCBI requirements. Once all of the GenBank files have been downloaded the script will proceed to construct the custom database. The completed database (**MFS_metaData.txt**) may be found at **GetDatabase/MFS/Output/** along with the names of the acquired meta-data fields (**MFS_Field_labels.txt**). This second file is required for building the ARB import filter described in the following section. Note that there are additional files in the Output directory that are not used as part of this tutorial.

The **buildDatabase.sh** script contains a python script (**NextractGB.py**) for constructing the custom database from the downloaded GenBank files. The **NextractGB.py** is also capable of reconstructing the entire NCBI taxonomy and using this information to populate taxonomy field labels within the custom database. To enable this feature you must uncomment lines 89-100 in the **NextractGB.py** file. Note that this feature requires at least 6GB of RAM (disabled by default). The supporting files and functions for this feature is contained in the **GetDatabase/NCBI_Taxonomy/** directory.

buildDatabase.sh dependencies:

- Python version 2.7
- Biopython version 1.60
- Python modules `httplib`, `urllib2`

NCBI_Taxonomy dependencies (additional):

- `taxid2lineageModule.py` module (script in **NCBI_Taxonomy**)
- `ete2` python module (An Environment for Tree Exploration (ETE))

<http://pypi.python.org/pypi/ete2/>

`ete2` dependencies:

Python Modules:

- `numpy`, `scipy`, `pyqt4`, `mysql`, `lxml`

Supplementary Material

7. From the directory containing *MFS_Field_labels.txt*, run the *buildFilter.py* (e.g. python buildFilter.py) script to generate the ARB import filter for the custom database. Use the *MFS_Field_labels.txt* as the input file and name the output, *MFS_import_filter.ift*. Place it in the ARB directory: *arb/lib/import/* (Typically, this directory is located at */opt/local/share/arb/lib/import/* in OSX).
8. We need to rename the headers in the alignment file and the leaves on the tree to the respective sequence's unique ID. This will ensure that we can link the meta-data, sequences, alignment and tree all together in ARB.
 - a. For the alignment:
 - i. Run the *getAccession.py* script from the directory containing *MFS_metaData.txt*. This will produce an alignment file (*MFS_UID.fasta*) that replaces the sequence headers with the respective unique ID.
 - b. For the tree labels:
 - i. We will use the Interactive Tree of Life (iTOL) to revise our tree labels.
<http://itol.embl.de/>

ITOL

INTERACTIVE TREE OF LIFE

Welcome to iTOL!

Interactive Tree Of Life is an online tool for the display and manipulation of phylogenetic trees. It provides most of the features available in other tree viewers, and offers a novel circular tree layout, which makes it easy to visualize mid-sized trees (up to several thousand leaves). Trees can be exported to several graphical formats, both bitmap and vector based. [more...](#)

News

- Version 2.1 introduces a new dataset type, [circles](#). Recent additions also include the ability to assign dataset values with internal nodes. These will be displayed only when associated clades are collapsed. Check the [Help pages](#) for details.
- Second iTOL article was published in 2011 NAR Web server issue ([abstract](#), [full text PDF](#)).

ITOL account login

Logged in as [sessinger](#). [Logout](#)

[Firefox toolbar](#)
If you are using Mozilla Firefox to access iTOL, try our [Firefox toolbar](#)

Recent changes

Version 2.2

- multiple binary datasets will be automatically spread to prevent overlap
- If labels are present in a color strip dataset, a legend containing the labels will be added to exported trees

Version 2.1.1

- branches can be marked with [custom labels](#)
- Dataset legends are optional in exported trees

Version 2.1

- New dataset type: [circles](#)
- Several datasets (binary, bar chart, multi-value bar chart, protein domains) support display of values assigned with internal nodes of the tree. These values will only be displayed when associated clades are collapsed.

Version 2.0.1

- Pie chart position on each branch can be defined: [example](#)

Version 2.0

- New dataset type: [connections](#)

Version 1.9

- [custom information](#) can be displayed in node popups; basic HTML and CSS are supported, with links and images

Version 1.8.3

- trees can be rerooted when exported through batch access

Version 1.8.2

- name based node filtering in the automatic NCBI tree generator

Version 1.8.1

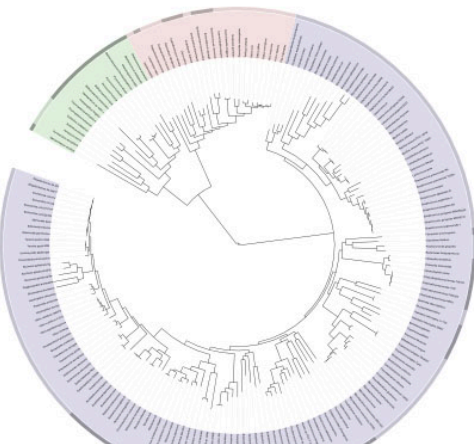
- protein IDs/ACCs (Uniprot and NCBI) supported in the automatic NCBI tree generator

Version 1.8

- new tree display mode: [unrooted](#)
- branches can be automatically colored based on bootstrap values

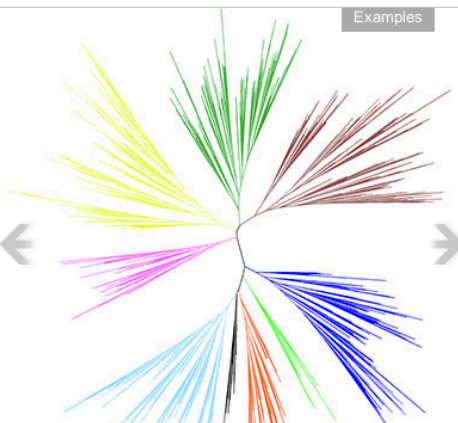
[Full version history](#)

The Tree Of Life



Various iTOL generated tree images

Examples



Large tree in unrooted display mode

87

ii. Login and upload the tree file from step 3c: *MFS.tree*

Project 1

No description.

Tree

Description

Datasets

Saved views

Inserted

Features

There are no trees associated with this project

Upload a new tree to this project

Add a new project...

Citations: Letunic and Bork (2006) *Bioinformatics* **23**(1):127-8 and Letunic and Bork (2011) *Nucleic Acids Res* doi: [10.1093/nar/gkr201](https://doi.org/10.1093/nar/gkr201)

ITOL

INTERACTIVE TREE OF LIFE

Upload a tree into the project 'Project 1' on workspace 'MFS'

Use this form to upload your own phylogenetic tree. It should be in plain text, in one of supported formats (Newick, Nexus or PhyloXML) . Please check [our help pages](#)

You can either paste your tree into the box, or upload a file using the file selector below. Don't forget to select the correct tree format.

Paste or type the tree:

Paste your tree here...

Upload a file which contains your tree:

Choose File

No file chosen

Tree format:

Newick

Make sure the file is plain text, and contains only trees in the selected format.

Optional information

Tree name:

MFS

If you don't specify a name, a numeric ID will be used instead

Tree description:

can also be set/changed later in your account page

Advanced options [\(show\)](#)

Upload

(If you're uploading extra data with your tree, fill the dataset section below before clicking 'Upload')

ITOL

INTERACTIVE TREE OF LIFE

Multiple trees succesfully uploaded

Tree upload summary:

The following table shows the uploaded trees and their corresponding projects. Use the links next to each tree to interactively assign color ranges (opens in a new window). You can also automatically assign taxonomy to multiple trees by ticking the 'Assign' checkbox and clicking the 'Automatically assign taxonomy' button below the table.

Name	Project	Automatic taxonomy assignment	Other options
MFS	Project 1	<input type="checkbox"/> Assign	Define color ranges
14411815766877813468824281	Project 1	<input type="checkbox"/> Assign	Define color ranges
Select all			

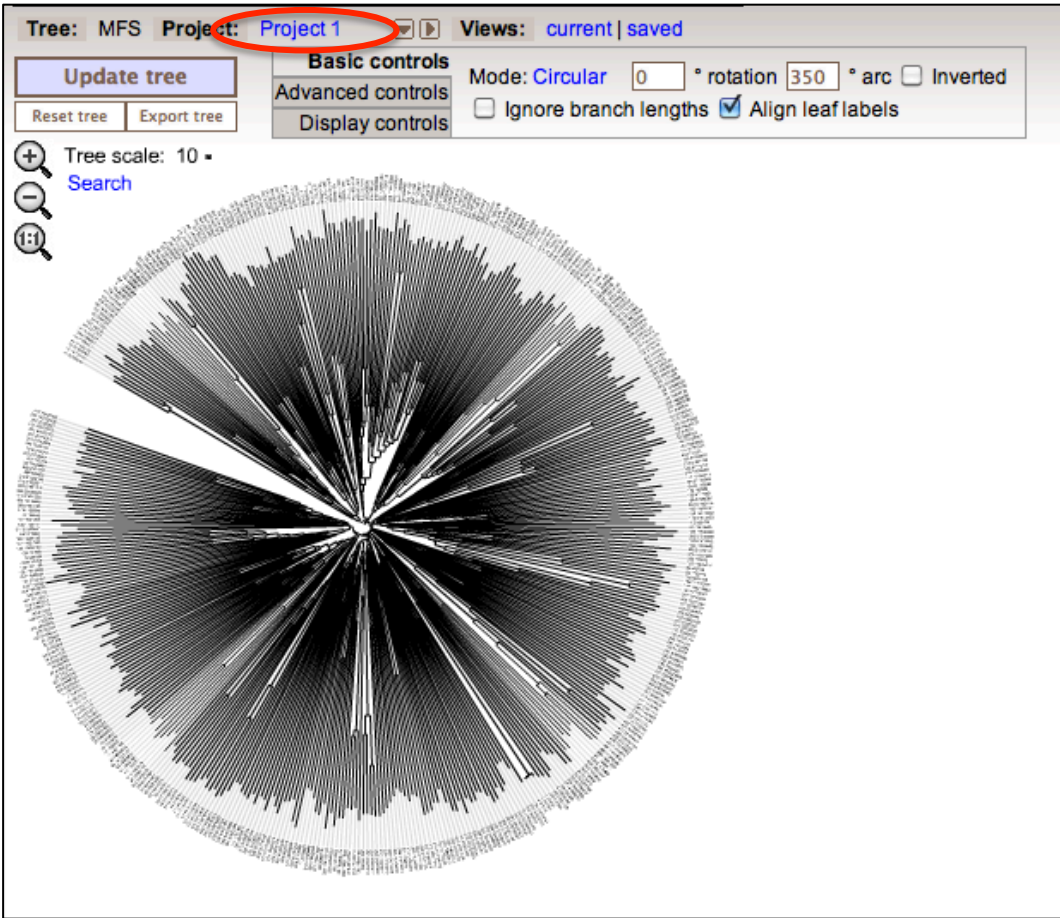
Automatically assign taxonomy

Back to your personal page

88

Supplementary Material

89 Verify tree then return back to Project 1.



- 90
- 91
- 92 iii. Rename the leaves using the file *TreeLabels_Mapped_New.txt*
- 93 *This file was created using the *rename_tree_leaves.py* script (see Appendix A).

Project 1

No description.

Tree	Description	Datasets	Saved views	Inserted	Features
MFS	Uploaded tree	-	-	2012-09-06	<input checked="" type="checkbox"/> % <input checked="" type="checkbox"/> L
14411815766877813468824281	Uploaded tree	-	-	2012-09-06	<input checked="" type="checkbox"/> % <input checked="" type="checkbox"/> L

Upload a new tree to this project

Add a new project...

- Rename tree
- Edit tree
- Root mid-point
- Duplicate tree
- Re-label leaves
- Assign taxonomy
- Edit colors
- Add a dataset

Allows you to edit the leaf labels. You can also upload a file and re-label the whole tree at once.

Citations: Letunic and Bork (2006) *Bioinformatics* **23**(1):127-8 and Letunic and Bork (2011) *Nucleic Acids Res* doi: 10.1093/nar/gkr244

Supplementary Material

99

Use the file *TreeLabels_Mapped_New.txt* here.

Editing leaf labels for tree 'MFS'

You can either use the form below to directly edit the labels, or upload a file with label definitions. The file should be in plain text and tab separated, with one label definition per line. First field should contain the leaf ID and the second one the new label.

Note: The labels are independent from actual leaf IDs in the tree. If you add new datasets to this tree, the dataset file must use the original leaf IDs and not the labels defined here.

Option 1: Upload label definitions

Label definition file:

Choose FileNo file chosen

Upload file

Option 2: Edit labels directly

Edit the individual labels and click 'Submit changes' when finished. Use the letter shortcuts to quickly find the individual entries.

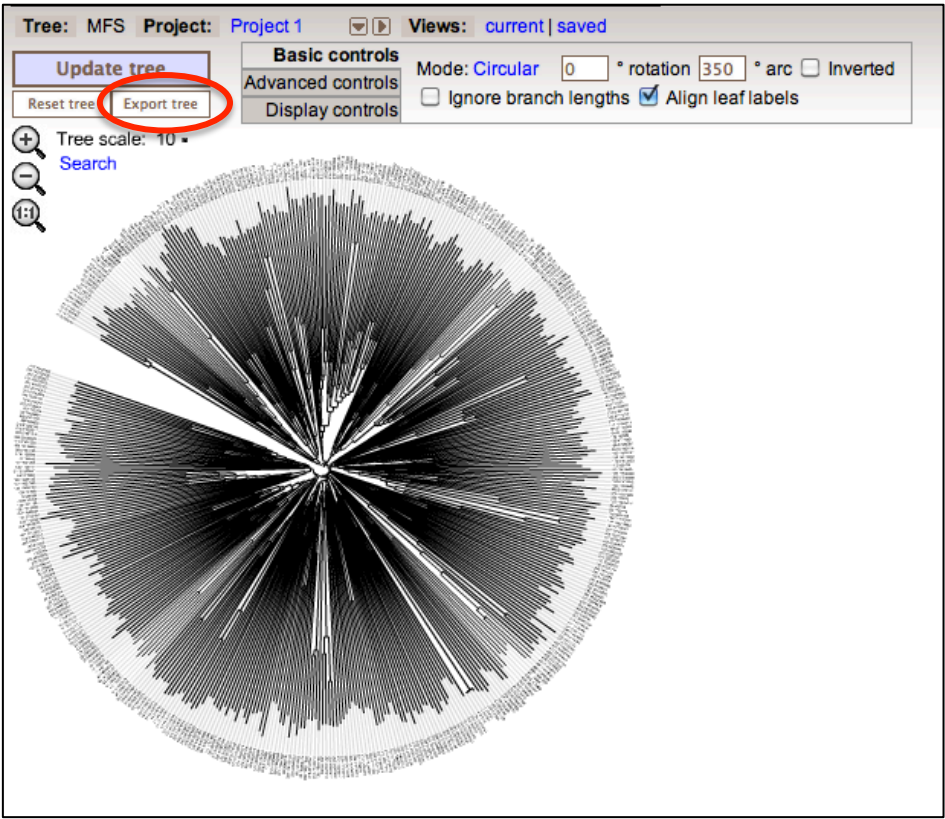
0123456789

Leaf ID	Label
0	<div>top</div>
0_1PW4A	<div>0_1PW4A</div>
1	<div>top</div>
100_167008994	<div>100_167008994</div>
101_2829802	<div>101_2829802</div>
102_74957183	<div>102_74957183</div>
103_74947906	<div>103_74947906</div>
104_2498056	<div>104_2498056</div>
105_48428688	<div>105_48428688</div>
106_82184762	<div>106_82184762</div>
107_187663979	<div>107_187663979</div>
108_74938574	<div>108_74938574</div>
109_13634050	<div>109_13634050</div>
10_74627205	<div>10_74627205</div>

100

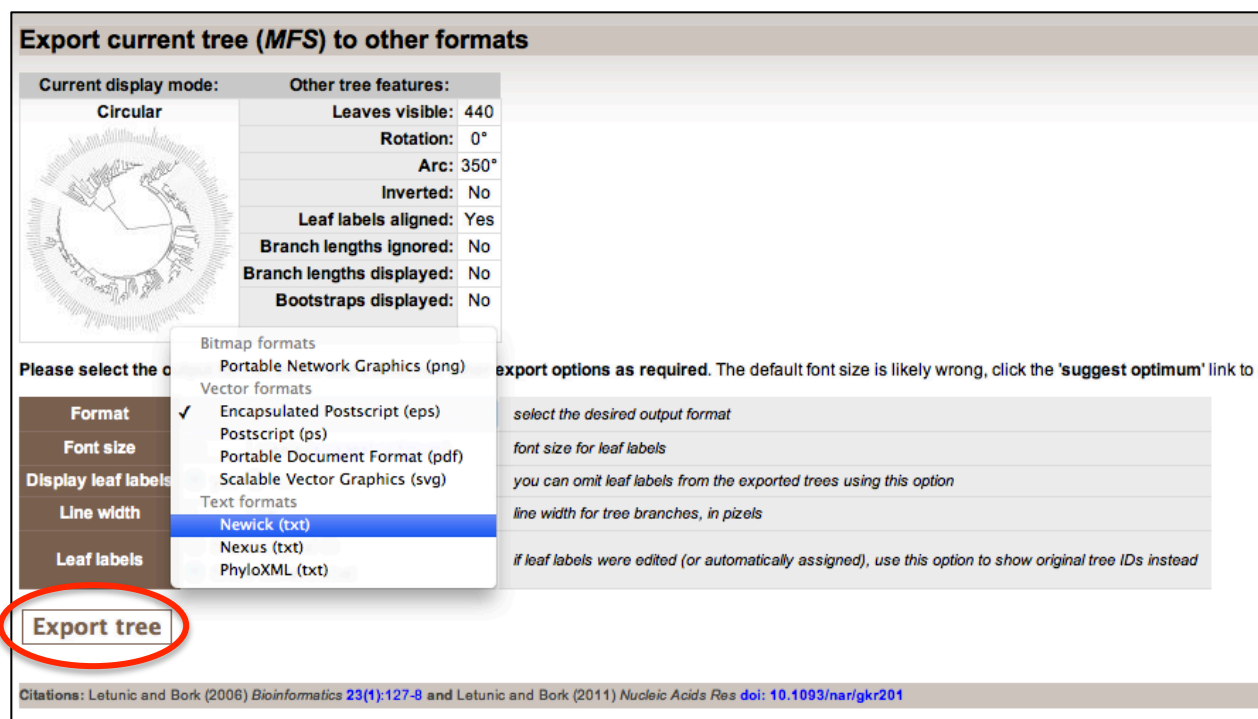
101

iv. Return to the tree view to export the tree with the newly assigned labels.

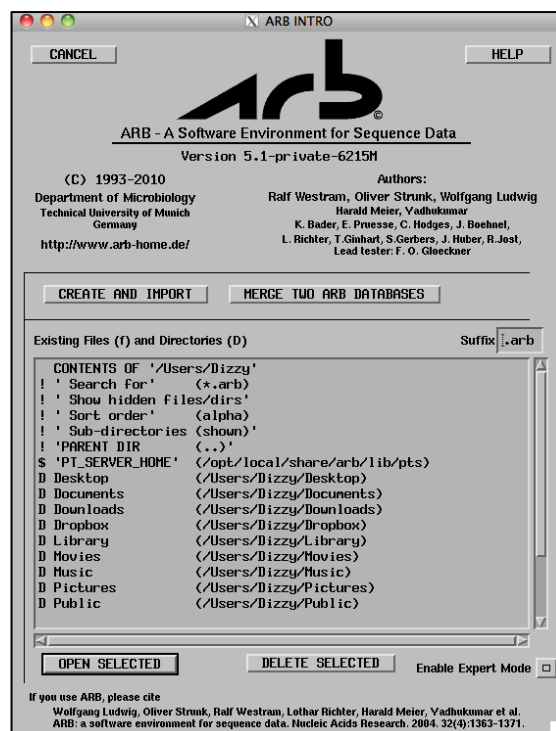


Supplementary Material

102 Export the tree in Newick format and save as *MFS_UID.tree*.



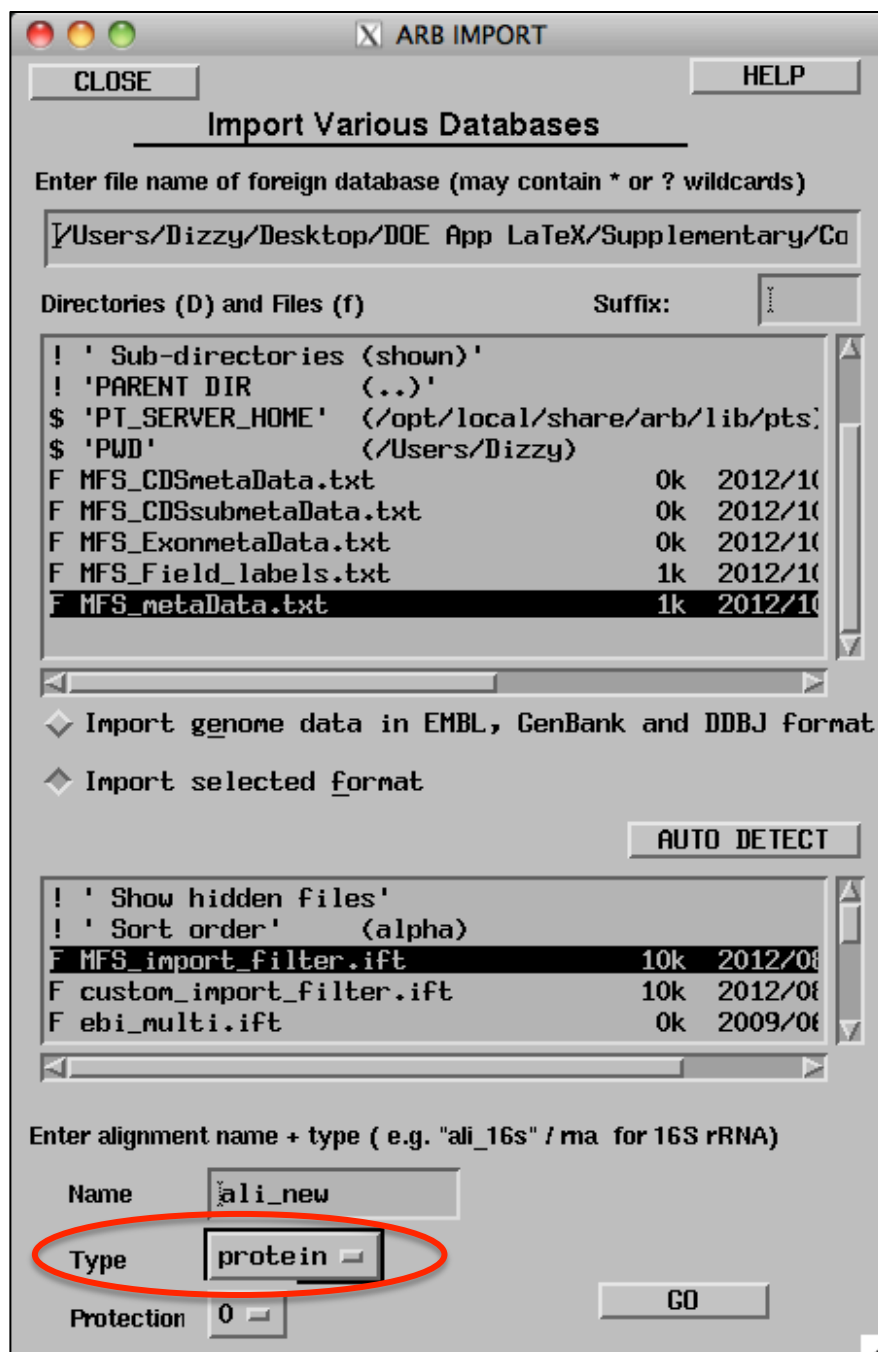
- 103
- 104
- 105 9. At this point we have four files and are ready to import them into ARB:
 - 106 a. Tree file with unique IDs: *MFS_UID.tree*
 - 107 b. Alignment file with unique IDs: *MFS_UID.fasta*
 - 108 c. Database file containing sequences with meta-data: *MFS_metaData.txt*
 - 109 d. ARB import filter: *MFS_import_filter.ift*
- 110
- 111 10. Start ARB from the command line. ARB is available at: <http://www.arb-home.de/>.



Supplementary Material

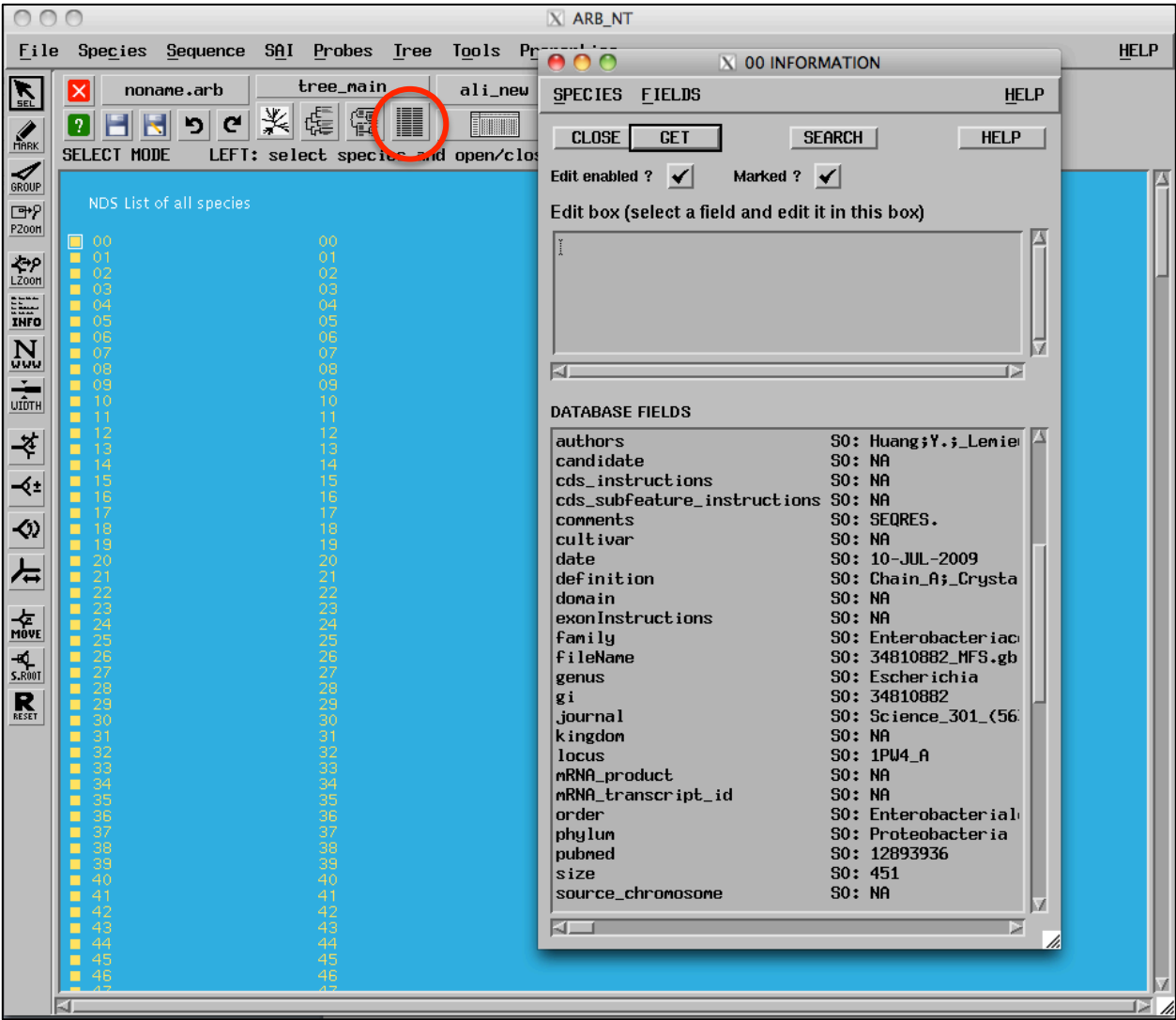
11. **Create and Import** the ARB database. Navigate to the directory containing the sequences and metadata `.../MFS/Output/`. Choose the ***MFS_metaData.txt*** file and use the ***MFS_import_filter.ift*** created in step 7. Change **Type** to **Protein**.

Note: If prompted, select 'Format (all)'. Also, if prompted choose 'Use found names'.



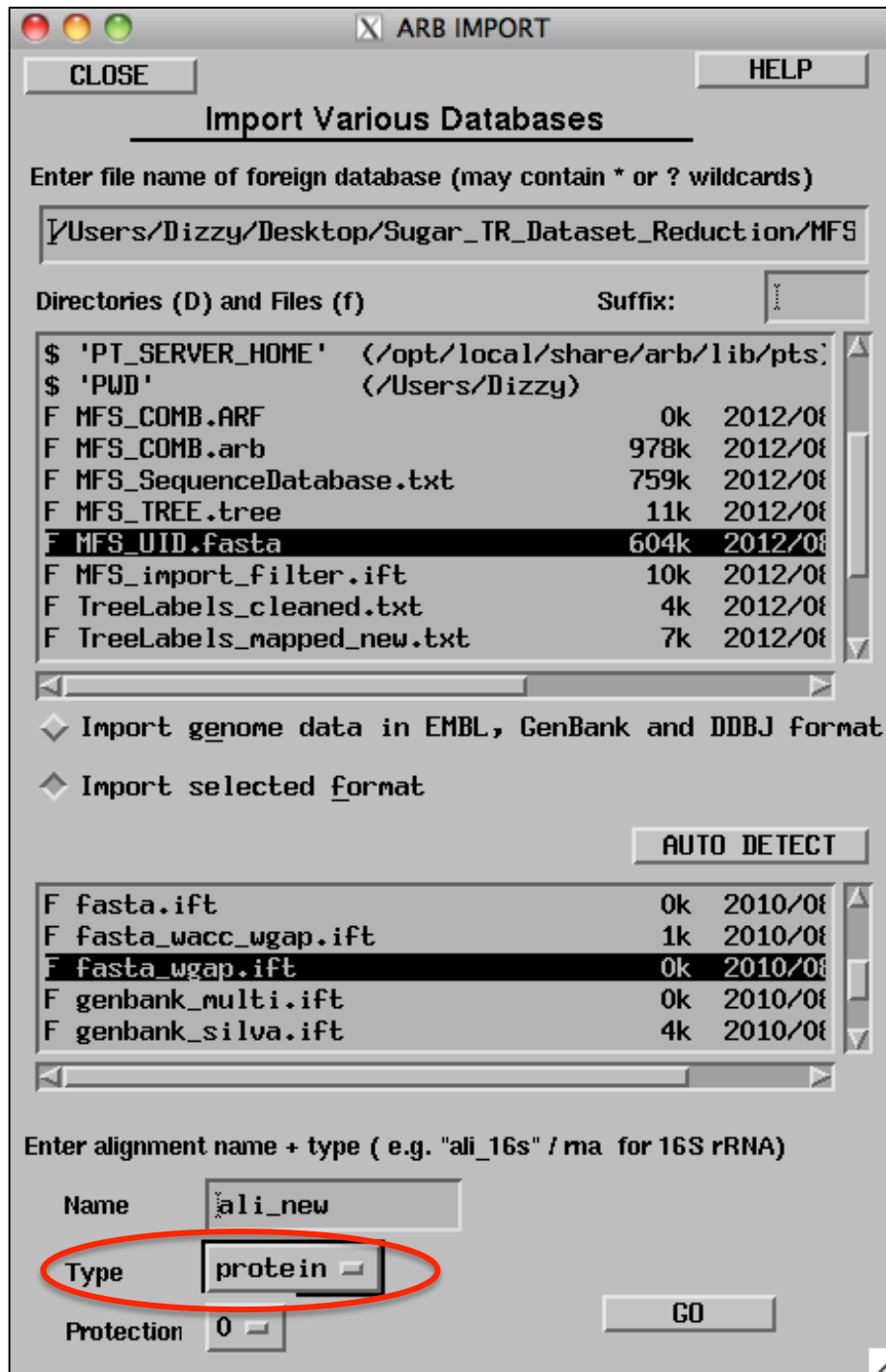
Supplementary Material

12. Verify the sequences/meta-data has been imported. Save the database as **DB_Meta.arb**.
Close ARB.
Note: You may need to select the list-view button to observe the list of all species.



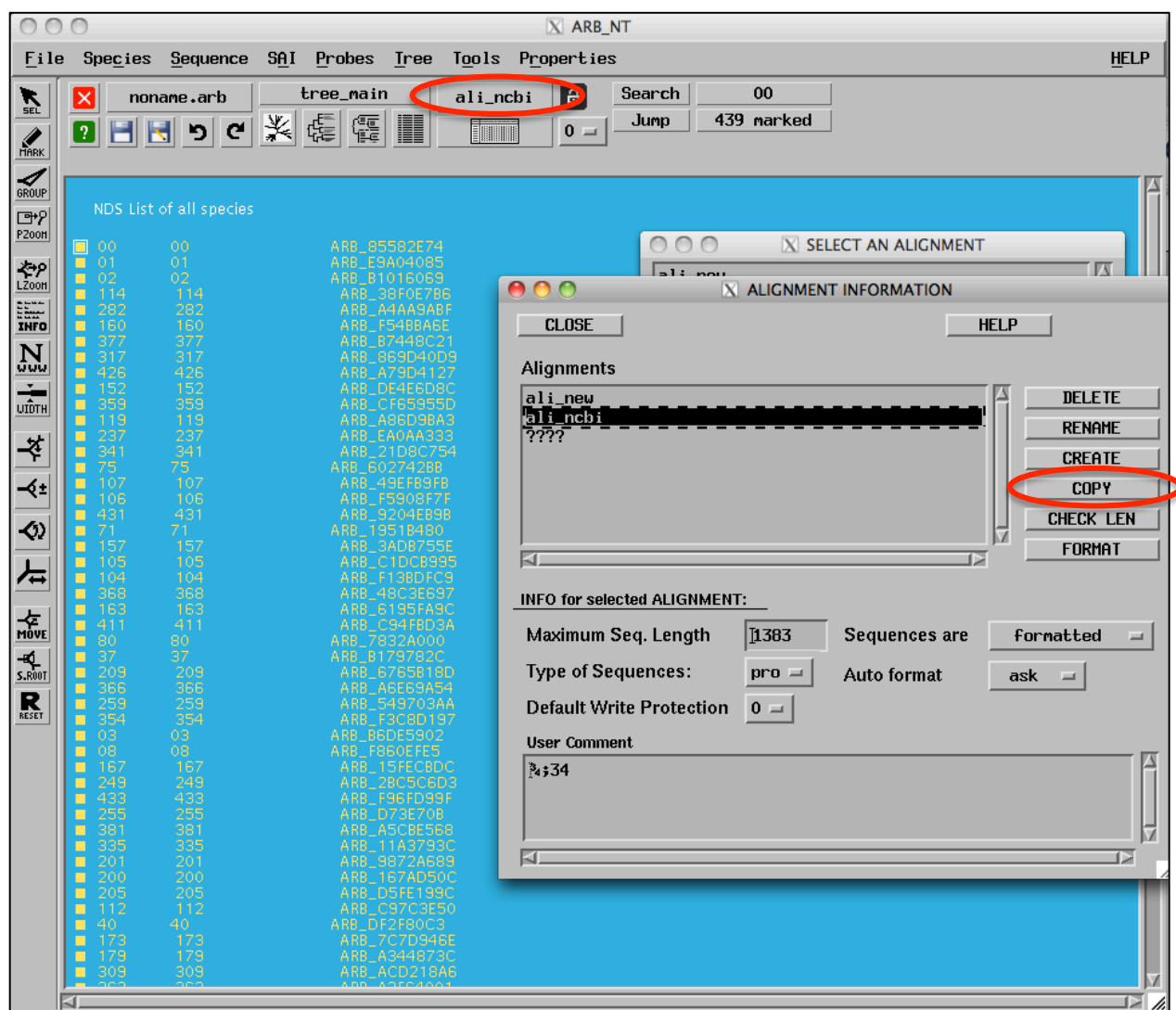
Supplementary Material

13. Open ARB. Import the alignment file. Use the *fasta_wgap.ift* filter this time. Change **Type** to **Protein**.
Note: If prompted, select 'Format (all)'. Also, if prompted choose 'Use found names'.



Supplementary Material

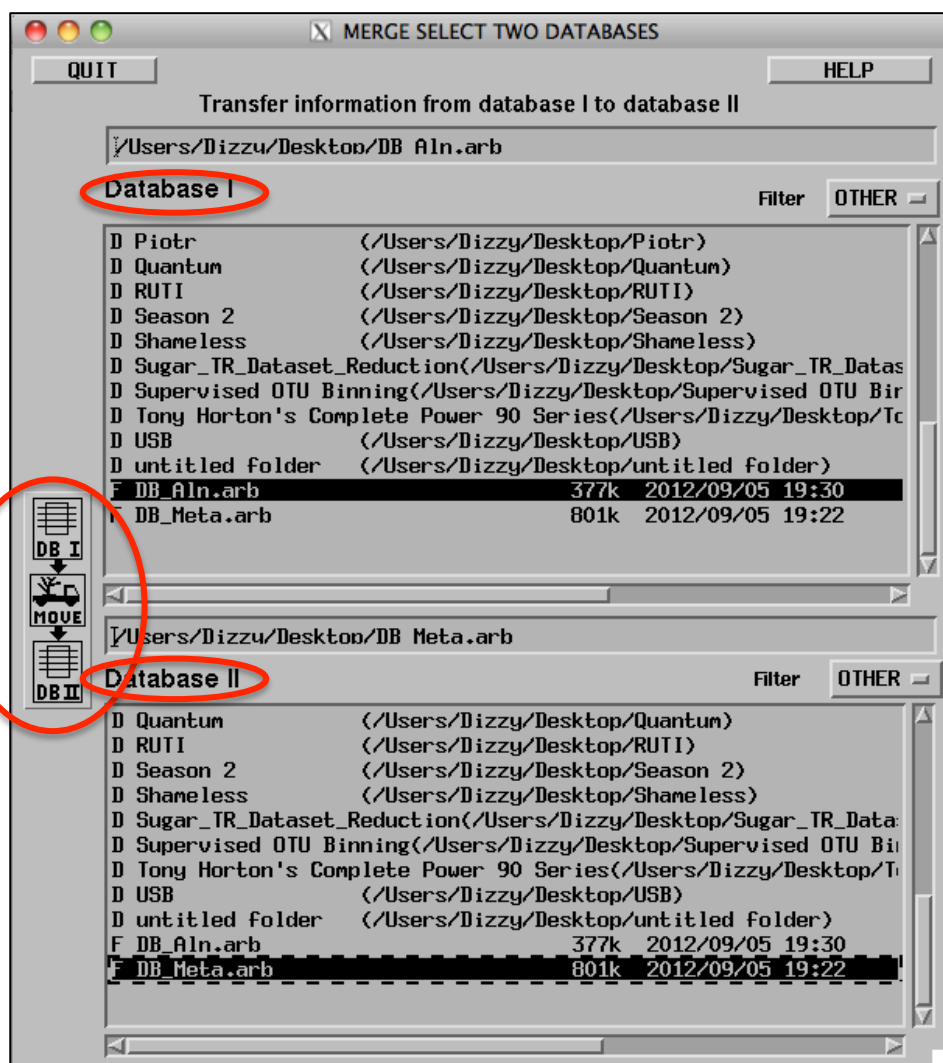
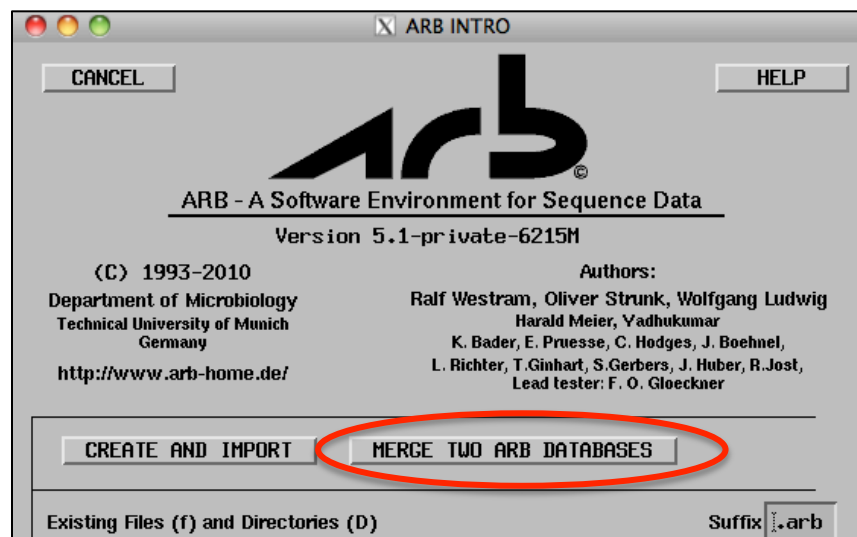
- 139 14. Create a new alignment field by copying *ali_new*. Name the new field *ali_ncbi*.



- 140
- 141 15. Save the database as *DB_Aln.arb*.
- 142
- 143
- 144
- 145
- 146
- 147
- 148

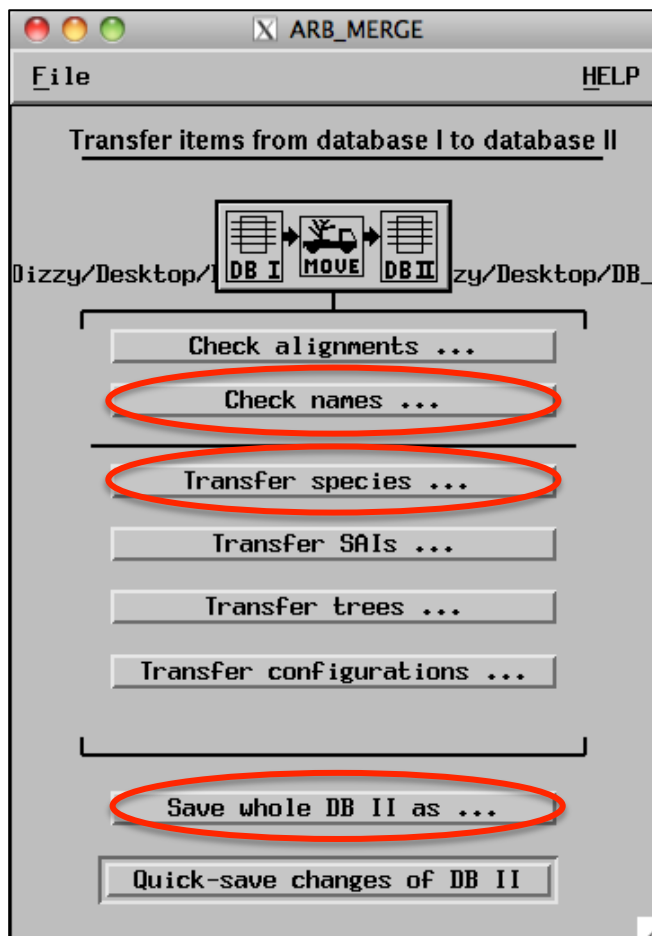
Supplementary Material

16. Merge both ARB databases so that all meta-data, sequences and alignments are linked.
 - a. Open ARB and select the merge database feature.
 - i. Set database I to **DB_Aln.arb**
 - ii. Set database II to **DB_Meta.arb**

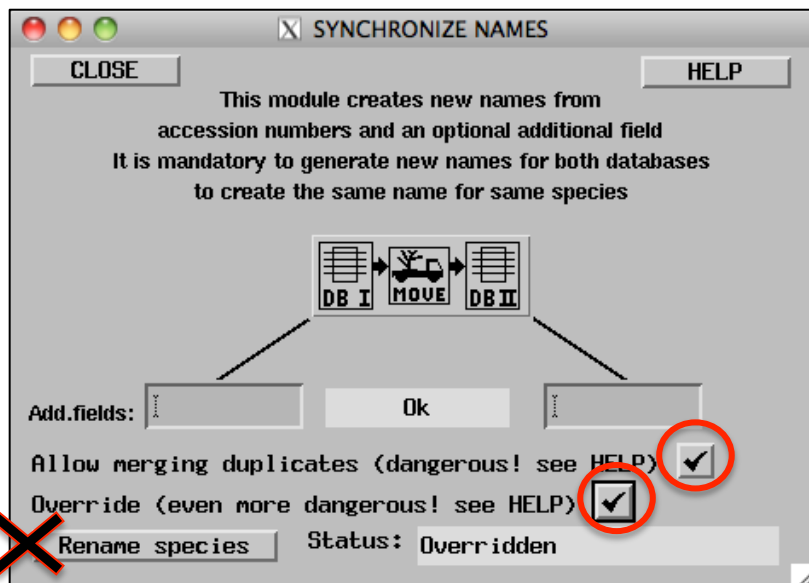


Supplementary Material

- 154
155
156
157
- b. Check names...
 - i. Check Allow Merging
 - ii. Check Override
- Note: Do NOT click the Rename Species button.*

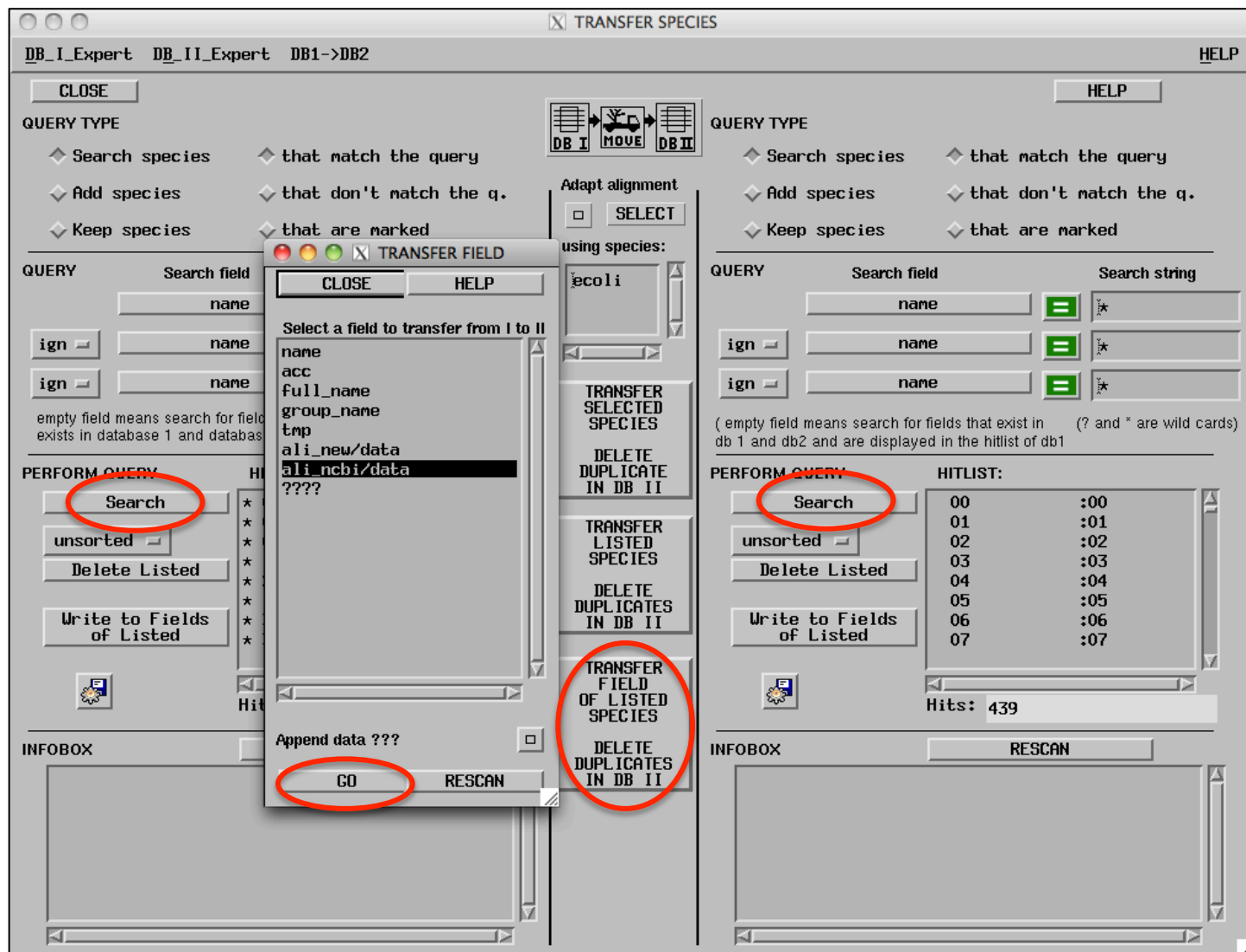


158



Supplementary Material

- 159 c. Transfer Species...
- 160 i. Mark All Species in both DBI and DBII (via Search)
- 161 ii. Click Transfer field of listed species...
- 162 iii. Select name of alignment to transfer: *aln_ncbi/data*
- 163 iv. Click Go.



- 164 d. Close Transfer Species. Save the merged database as *MFS_COMB.arb*.
- 165

166

167

168

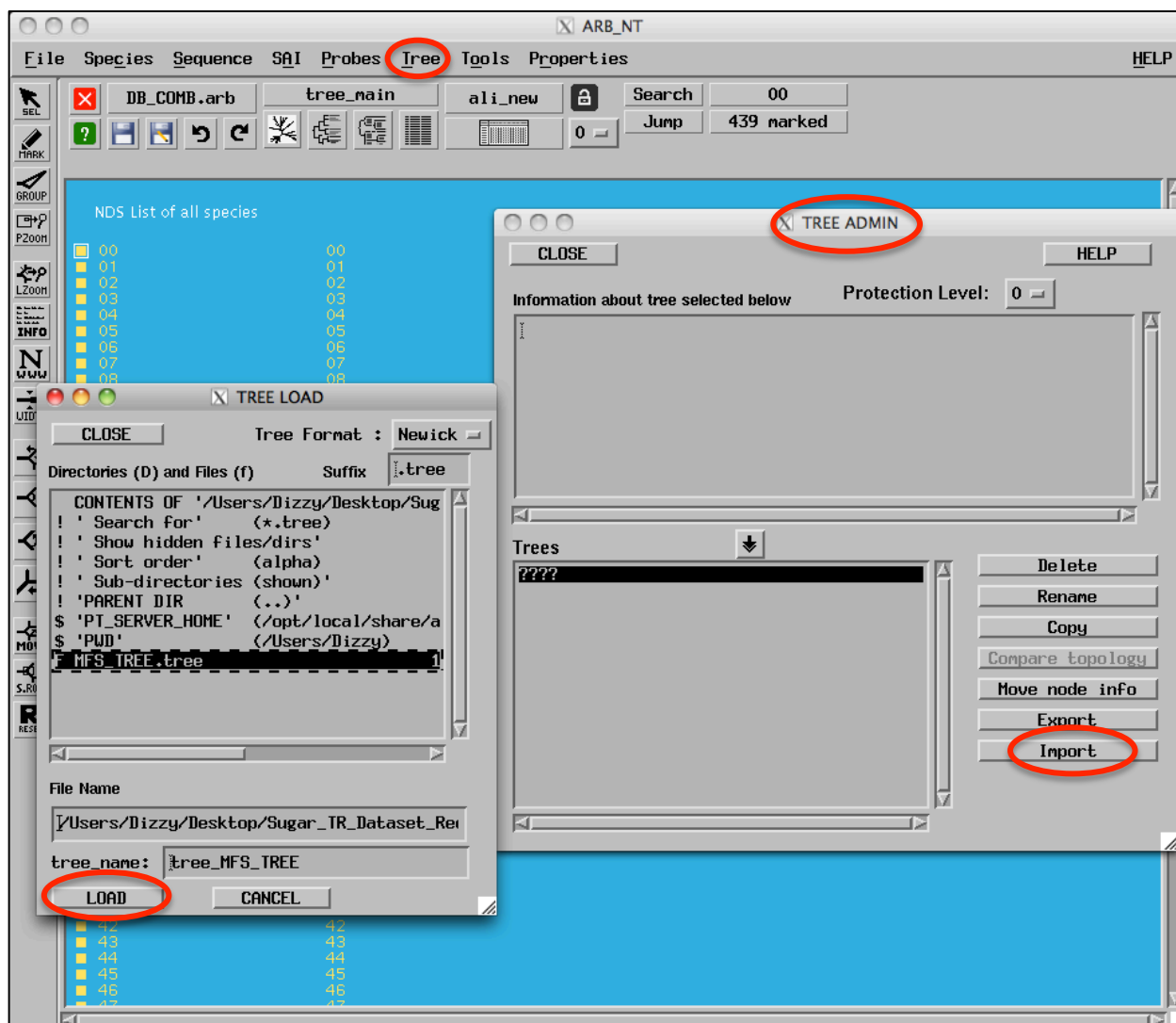
169

170

Supplementary Material

17. Import tree to ARB

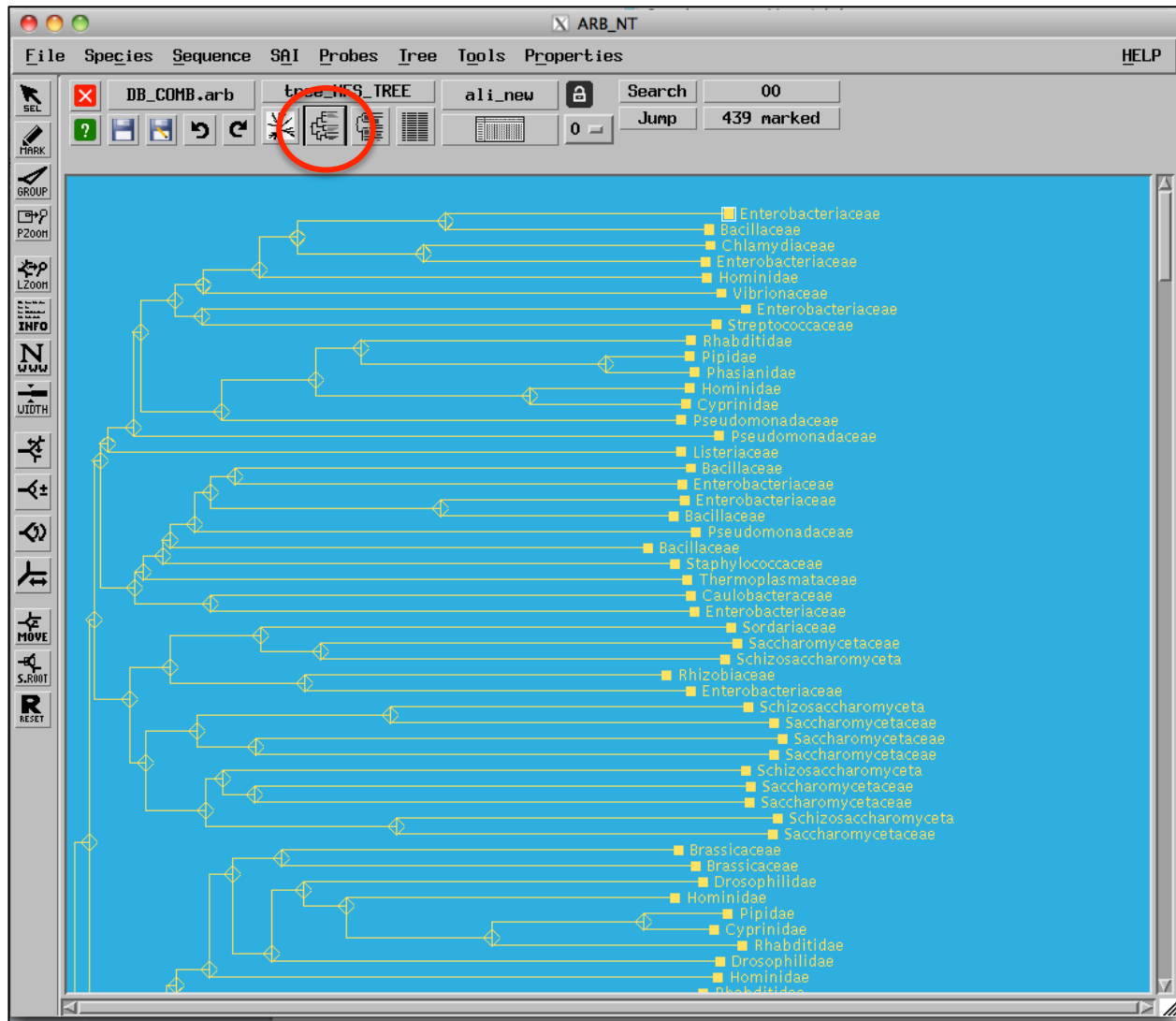
- Open ARB and the MFS_COMB database.
- Select Tree/Tree Admin/
- Import MFS_UID.tree from step 8.



18. Save the database again so the tree is stored. The import process is complete!

Supplementary Material

184 Tree view using meta-label **Family** to investigate taxonomic relationship of leaves on tree. The
 185 leaf labels may be changed to display any of the meta-data choices via the Node Display Setup
 186 (NDS) under the Tree menu. To view different leaf labels on the tree go to: Tree → NDS (Node
 187 Display Setup).



Supplementary Material

Appendix A - Renaming Tree Labels

The tree produced via the MFS CDD database contains leaves that are named using an index followed by an underscore followed by the identifier displayed on the tree (e.g. 2_2V8N|A). Note that the index is hidden when viewing the tree in *CDTree*. We need to rename the leaves so that they match the unique identifier in our custom database (i.e. MFS_metaData.txt). This is necessary so that we can link the tree with the alignment and the meta-data once they are all imported into ARB.

Renaming the tree leaves:

1. Navigate to the 'Editing Leaf Labels for Tree MFS' page on the iTOL website as described in step 8, b, iii.
2. Highlight, then copy the current leaf labels on this page to a text file. Manually inspect this file to ensure that only leaf names have been copied. Delete anything else that was added to the file. Save this file as *TreeLabels_Orig.txt*
3. The MFS CDD dataset has a slight naming inconsistency between the alignment identifiers and the tree leaves. The tree does not contain the '|' character, but for parsing purposes this must be inserted into the *TreeLabels_Orig.txt* file so that it is consistent with the alignment file. There are five instances where '|' must be inserted:
0_1PW4|A 1_2GFP|A 2_2V8N|A 3_2CFQ|A 4_1PV7|A
4. Place *TreeLabels_Orig.txt*, *MFS_UID.fasta* and *MFS_Align.fasta* in the same directory as *rename_tree_leaves.py*. Run the *rename_tree_leaves.py* script to produce the new tree labels file, *TreeLabels_Mapped_New.txt*. This file is formatted so that each line specifies one leaf name: the original followed by a tab and then the new one (i.e. the unique ID).
5. Use the *TreeLabels_Mapped_New.txt* as directed in step 8, b, iii.