A Guide to Using AdaptML

by Lawrence David (ldavid@mit.edu)

Guide Version 1.0.0.
Last updated: 05/24/08

AdaptML is a software package that will automatically partition groups of gene sequences according to both their genetic and ecological similarity. For a full description of AdaptML's design and capabilities, please see (ref 1) and its supplementary online materials.

1. **Installing AdaptML**:

   1.1. AdaptML will run on any computer capable of interpreting the Python programming language. Simply download and install either a Windows, Mac OS X, or *nix version of Python more recent than Python 2.5.

   1.2. Download and install the NumPy and SciPy packages

   1.3. Download AdaptML here: http://almlab.mit.edu/AdaptML/adaptml.tar.gz

2. **Running AdaptML**: The AdaptML software package functions in several parts. First, AdaptML.py is used to learn the habitats and their ecological makeup in the input dataset. Second, JointML.py assigns each extant and ancestral strain to one of the inferred habitats. Third, the software package iToL is used to visualize the results of the AdaptML analysis.

   2.1. *Habitat learning*

   2.1.1. The syntax for calling the habitat learning component of AdaptML is: *python AdaptML.py tree=input_tree.file init_hab_num=16 outgroup=5S_OUTGROUP write_dir=./ collapse_thresh=0.10 converge_thresh=0.001 rateopt=avg*

   2.1.1.1. **tree**: An input phylogenetic tree that incorporates ecological data into sequence filenames. Due to Newick file idiosyncrasies it is recommended that PhyML be used to generate input trees. Gene sequences should be named according to the following format: "*EcologyID_SequenceID*" where *EcologyID* is a string shared in common by all sequences with identical ecology and *SequenceID* is a unique identifier such that no two sequences with the same ecology information share the same *SequenceID*. Note that for naming conventions in 2.2, it is recommended that the *EcologyID* should have one character position for each ecological variable and a different character for each possible instance of that variable For instance, if sequences are drawn from either High-Light and Low-Light environments, and either 1m or 5m of depth, it is recommended that the 4 *EcologyIDs* be: H1, L1, H5, & L5. If clonal sequences exist in the sequence dataset, care should be taken that the subtree containing those sequences does not have subtrees monophyletic for a particular *EcologyID*. Doing so may lead to spurious habitat inference. An example input tree can be found in /example/vibrio.hsp60.tree.

2.1.1.2. **init_hab_num**: The number of random habitats AdaptML will initialize with. If the ultimate number of inferred habitats is equal to this initial number, try re-running AdaptML with more initial habitats.

2.1.1.3. **outgroup**: The outgroup sequence for the input tree. Note that this name should have an *EcologyID* (can just pick one of the *EcologyIDs* used in the dataset).

2.1.1.4. **write_dir**: Directory to write output to.

2.1.1.5. **collapse_thresh**: Threshold value for collapsing redundant habitats. Value should range between (0,1). Default value is 0.10. Higher values will lead to fewer habitats being inferred.

2.1.1.6. **converge_thresh**: Threshold value for declaring habitat distributions to have converged. Value should range from (0,1). Default value is 0.001.

2.1.1.7. **rateopt**: Method for inferring *mu* or the average habitat transition rate. Default is ‘avg’, which is a fast, approximative method. A more precise, but also more time-consuming option is ‘num’, which uses SciPy’s numerical optimization toolbox.

2.1.2. The output files:

2.1.2.1. **habitat.matrix**: Defines the inferred probability that for a given habitat, an isolate adapted to that habitat will be observed in a given ecological niche. Also referred to as the “emission probability matrix” in (ref 1 SoM).

2.1.2.2. **.mu.file**: The inferred average habitat transition rate.

2.1.2.3. **stats.file**: Inference process statistics.

2.2. *Cluster Assignment*

2.2.1. Once the habitats have been inferred, habitat assignment can proceed: *python JointML.py tree=vibrio.hsp60.tree outgroup=5S_OUTGROUP habitats=habitat.matrix mu=mu.val write=./ color=color.file*

2.2.1.1. **tree**: Same as inputted in 2.1.1.

2.2.1.2. **outgroup**: Same as inputted in 2.1.1

2.2.1.3. **habitats**: Habitats inferred in 2.1.2.1.

2.2.1.4. **mu**: Habitat transition rate inferred in 2.1.2.2.

2.2.1.5. **write**: Output directory.

2.2.1.6. **color**: File specifying visualization colors for both leaves and ancestral nodes on phylogenetic tree. Leaves sharing identical ecology will have the same radial bar plots; ancestral nodes sharing the same ancestral assignment will also have uniform colors. To specify bar plot components, identify *EcologyID* character position in column 1 (character position begins at 1) and desired character in column 2. To specify habitat colors, put an 'H' in column 1 and identify the habitat number in column 2. Columns 3, 4, & 5 define R, G, & B integer values from (0,255). Example file provided in /example/color.file. Each column should be single-space delimited.

2.2.2. The output files:

2.2.2.1. **habitat.file**: The habitat assignment of each sequence.

2.2.2.2. **itol.tree**: Phylogenetic tree with habitat assignments for each ancestral sequence.

2.2.2.3. **full.file**: Data file for iToL visualization (see 2.3).

2.3. *Visualization*

2.3.1. Using iToL: AdaptML was developed with the iToL visualization package in mind. To use this tree depiction software:

2.3.1.1. Visit http://itol.embl.de/ and click on the Data Upload tab.

2.3.1.2. Choose to upload the itol.tree file from 2.2.2.2.

2.3.1.3. In Dataset 1, choose to upload the full.file data file from 2.2.2.3.

2.3.1.4. Assign this dataset a display label.

2.3.1.5. Select the comma field delimiter and the multi-value bar chart or pie chart options.

2.3.1.6. Click upload.

References:

(1) Dana E. Hunt*, Lawrence A. David*, Dirk Gevers, Sarah P. Preheim, Eric J. Alm, and Martin F. Polz. Resource Partitioning and Sympatric Differentiation Among Closely Related Bacterioplankton. Science 23 May 2008: 1081-1085.