MyESL: A software for evolutionary sparse learning in molecular phylogenetics and genomics

Supplementary Information

A list of MyESL commands and associated descriptions.

Command	Description
Model Options	
<file name>	A text file containing full paths for all gene sequence alignments in FASTA format.
<tree file name>	A phylogenetic tree in the Newick format with clade labels. If the response file is not provided using the text format, MyESL processes the text file to create the response for the ESL model.
--clade_list <text_file>	A text file containing the list of all clades for which clade models are built. Each line in the file contains clade labels.
--gen_clade_list <lower, upper>	For use when the input phylogeny does not contain any clade names. It automatically generates clade labels for all internal nodes whose descendant species count is within the specified **lower** and **upper** bounds. By default, both values are set to 0, which results in the assignment of clade labels to **all internal nodes** in the phylogeny.
--auto_name_length <int>	Specifies the number of characters to extract from species names when generating internal node labels for clades. The default is **3**, meaning the first three letters of species names will be used to construct clade labels.
--clade_size_cutoff_lower <int>	It enables users to automatically generate candidate clades for model building by specifying a lower and upper bound on clade size (both numeric). Clades contaseveralber of species within the specified size range will be selected, and ESL models will be built for each of them. ESL models will not be built for other clades containing species more or fewer than the specified cut-off.
--clade_size_cutoff_upper <int>	
--class_bal <method>	For selecting the strategy for class balancing: ● **up**: Upsampling the smaller class to match the size of the larger class. ● **down**: Downsampling the larger class to match the smaller class ● **weighted**: Applying inverse class weights during model training ● **phylo**: Phylogeny-aware balancing that selects training samples based on evolutionary relationships
--method	Specifies the type of loss function or regression to be used in model

Command	Description
<method>	training. The default option is logistic. ● logistic: which performs sparse group lasso or lasso regression with logistic loss and is suitable for binary response variables (e.g., +1/–1). ● leastr: applies sparse group lasso or lasso regression using least squares loss, and should be used when the response variable is continuous, ranging from $-\infty$ to $+\infty$.
--lambda1 <float [0-1]>	Specifies the site sparsity penalty parameter that controls the inclusion of features (e.g., sites) in the clade model. The value must be a float between 0 and 1, where higher values apply stronger penalization, resulting in a sparser model that selects fewer sites.
--lambda2 <float [0-1]>	Specifies the group sparsity penalty parameter that controls the inclusion of feature groups (e.g., genes) in the clade model. The value must be a float between 0 and 1, where higher values apply stronger penalization, resulting in a sparser model that selects fewer groups.
--no_group_penalty	Enables monolevel Lasso analysis by disabling the group-level sparsity penalty. This directive must be used in combination with --lambda1 to ensure that sparsity is applied only at the individual feature level, not at the group (e.g., gene) level.
--lambda1_grid <float, float, float [0-1]>	Specifies a range of feature-level sparsity parameters (lambda1) for building multiple clade models, one model for each parameter. This directive takes three float values: the starting value, the maximum value, and the step size. All values should be between 0 and 1. The parameter values increase linearly from the start to the maximum using the specified step, allowing exploration of different levels of feature sparsity.
--lambda1_grid <float, float, float [0-1]>	Specifies a range of feature-level sparsity parameters (lambda1) for building multiple clade models, one model for each parameter. This directive takes three float values: the starting value, the maximum value, and the step size. All values should be between 0 and 1. The parameter values increase linearly from the start to the maximum using the specified step, allowing exploration of different levels of feature sparsity.
--grid_rmse_cutoff <float>	This directive is to be used when lambda grids are used with multiple models. This helps to reduce the number of models to build and which models to use for aggregation.
--grid_acc_cutoff <float [0 1)>	Sets the minimum training accuracy threshold for retaining models during grid search. Any model with accuracy below this cutoff will be discarded.
--grid_summary_only	This directive retains only the summary of all models generated during the grid search and discards the results of individual models.
--min_group <num>	Specifies the minimum number of groups that must be retained in each model during grid search. Models with fewer groups will be discarded.

Command	Description
Data Preprocessing	
--include_singletons	By default, MyESL removes all conserved sites and singleton variants (sites with only one differing sequence). Use this option to retain singleton sites during clade model construction if they are relevant to your analysis.
--bit_ct <num>	Excludes sites where mutations are observed fewer than <num> times across all sequences. This helps filter out extremely rare variants that may be uninformative or noisy.
--data_type <type>	Specifies the type of input sequence data MyESL will process. Supported options are: ● nucleotide: For DNA sequences ● protein: For amino acid (AA) sequences ● molecular: For datasets containing both DNA and protein sequences ● universal: Allows any IUPAC characters, additional symbols, and numeric values (0–9), treating each unique character, including numbers, as distinct. This mode is case-sensitive, unlike the others, which are not. Choose the appropriate type based on the format and diversity of your input data.
Model Output	
--output <name>	An output directory name can be defined by users. This directory will be created in the current working directory. If no output directory name is defined by users, a directory named "**output**" will be created by default.
--gene_display_limit <num>	Specifies the maximum number of gene groups (columns) to display in the model grid generated by MyESL. The default is 20, or fewer if the number of selected groups is less than 20.
--species_display_limit <num>	Specifies the maximum number of taxa (columns) from the clade of interest to display in the model grid generated by MyESL. The default is 20, or fewer if the number of selected taxa is less than 20.
--gene_display_cutoff <num>	Defines the number of genes or groups (columns) to display in the model grid. The default is 20, or fewer if fewer than 20 groups are selected.
--m_grid <row,col>	Sets the number of rows and columns in the model grid displayed by MyESL. The default is 20 for both dimensions, or fewer if the available rows or columns are less than 20.
--stats_out <PGHS>	Specifies the types of sparsity statistics to output from the model. Users can include any combination of the following: **P:** Site-level (position) sparsity scores **G**: Gene or group sparsity scores **H**: Hypothesis sparsity score **S**: Species prediction scores

Command	Description
Miscellaneous	
--DrPhylo	Enables DrPhylo analysis for a specified clade, either defined by clade ID using a phylogeny in a NEWICK format or via a response file in text format. DrPhylo builds multiple sparse models across combinations of site and group sparsity parameters. By default, it explores values from 0.1 to 0.9 (in steps of 0.1), generating 81 models. Users can customize this range using the --lambda_grid directive. To reduce computation, DrPhylo skips models that include fewer than three genes. It outputs summary statistics, including PSS, GSS, and HSS scores, as well as a model grid. The default grid size is 20×20 (20 rows for species and 20 columns for genes), which can be adjusted using the --m_grid <row, col> directive.
--subsets <int>	Enables subset-based analysis by dividing feature groups (e.g., genes) into multiple subsets. Models are trained on each subset, and informative groups are selected from each. A final model is then built using the combined set of selected groups. If --subsets 0, MyESL automatically determines the optimal number of subsets based on the system's computational capacity.
Prediction	
MyESL provides a separate pipeline "**MyESL_model_apply.exe**" for applying the ESL model to predict traits or determine the clade membership of new species. To use this feature, users must provide sequence alignments that are aligned with the training alignments.	
<file name>	This positional input refers to the text file generated in the result directory of a MyESL analysis. The file, starting with "**MyES_model**", contains the ESL model, specifically, the estimated weights of the selected features.
<file name>	This is a positional directive. In the second position, provide a text file listing the full paths to all sequence alignments that will be used for prediction.
--response <file name>	Specifies a text file where each line contains the name of a species for which clade membership or trait phenotype predictions will be performed.
--output <name>	Specifies the name of the directory where all results will be stored. If no name is provided, the output will be saved in a directory named "output" by default.
--gene_display_limit <int>	Sets the maximum number of genes to display in the grid image generated for the species specified with the --response directive.
--m_grid <int,int>	Sets the number of rows and columns in the model grid displayed by MyESL. The default is 20 for both dimensions, or fewer if the available rows or columns are less than 20