**DeepHyb Manual**

**1. Converting Phylip Files to CNN-Ready Datasets**

First, a sequential Phylip format file with header information removed (i.e., excluding lines specifying the number of individuals or sites) is required as input. This Phylip file must contain sequences from at least four species.

Prior to use, install HyDe to assist with labeling the training dataset. HyDe is available at: <https://github.com/pblischak/HyDe>.

Execute the script via the command line:  
python msa\_sitepattern\_counter.py

The script accepts the following command-line arguments to configure phylogenetic data processing with k-mer and depth parameters:

Command-Line Arguments

--phy\_folder  
Type: string  
Default: phy  
Specifies the folder containing input .phy files.

--kmer\_range  
Type: integers (space-separated)  
Default: 3 4 5  
Defines the range of k-mer lengths to include in the analysis. For example, use --kmer\_range 2 3 6 to analyze k-mers of lengths 2, 3, and 6.

--depth\_range  
Type: integers (space-separated)  
Default: 0 1 2  
Sets the range of depth values to be analyzed, provided as space-separated integers.

--extended\_range  
Type: integers (space-separated)  
Default: 1 2  
Specifies the range of extended parameters to incorporate into the analysis.

--out\_name  
Type: string  
Default: out  
Names the outgroup species used in phylogenetic calculations.

--num\_cores  
Type: integer  
Default: None  
Indicates the number of CPU cores to use for parallel processing. If not specified, all available cores will be utilized.

--output\_folder  
Type: string  
Default: output\_jsons  
Designates the directory where output JSON files (containing analysis results) will be saved.

Upon input of the Phylip file, JSON files containing four features and two target label sets will be generated. These are detailed as follows:

(1) HyDe site patterns: Using the outgroup’s bases as a reference, each base of species1 was compared to the outgroup (same = outgroup-like, otherwise = other). Bases of species2 were compared to both the outgroup and species1; bases of species3 were compared to the outgroup, species1, and species2. This yielded 15 site patterns (1×15-dimensional data), denoted 15-summary-site-patterns, also Feature (1).

(2) Base combinations per site: 4⁴ = 256 combinations (1×256-dimensional feature), which can be summed to form HyDe site patterns, denoted 256-one-base-site-patterns, also Feature (2).

(3) K-mer patterns (k > 1): K-mer length × 15-dimensional features, denoted 75-summary-kmer-site-patterns, also Feature (3).

(4) Individual-specific k-mer counts: Counts of consecutive 3-base combinations (4³ = 64 per individual) for four individuals, yielding 4×64 = 256-dimensional features, denoted 256-seq-kmer-patterns, also Feature (4).

.

Target label sets: Two distinct sets are generated:

* Hybridization Status (2 categories)
* Parent-Offspring Identification (64 categories)

Additional intermediate files (required input and output files for HyDe) will also be generated.

**2. Training and Testing the CNN Model**

A subset of the dataset is allocated for training and testing, with default directories named train and test. The default patience value for early stopping (training terminates if the model’s loss fails to decrease over consecutive epochs) is 15,000.

To use the framework that combines all four features, simply execute:  
python DeepHybCNN.py

The trained model file will be saved in the model folder by default, and loss history will be stored in the loss\_history folder.

For frameworks using individual features separately, execute the following scripts respectively:  
DeepHybCNN15d.py, DeepHybCNN256d.py, DeepHybCNN15m5d.py, and DeepHybCNN64m4d.py.

**3. Analyzing Feature Contributions with SHAP**

To analyze feature contributions using SHAP, run:  
python shap.py

For SHAP contribution analysis of individual features, execute the following scripts respectively:  
shap15d.py, shap256d.py, shap15m5d.py, and shap64m4d.py.