PhyDGET

James B. Pease

5 November, 2024

Contents

Version 1.1.0

Introduction

Authors

James B. Pease (http://www.peaselab.org)

Contributors:

David de la Cerda

What is PhyDGET?

Phylogenetic Differential Gene Expression Tool (PhyDGET) is a method for analyzing the changes in transcriptome-wide expression levels gene by gene on a phylogeny. PhyDGET is a merger in method and thinking between Phylogenetic Comparative Methods and Differential Gene Expression. PhyDGET first performs a log2 counts-per-million transformation and filters low-coverage genes to prepare them for phylogenetic testing. PhyDGET then parallelizes the passage of these data to BayesTrait-v3, which tests each gene's expression level as a quantitative trait evolving on the tree. Using BayesTrait's likelihood ratio test framework, you can specify a range of branch rate-shifting models to test what genes' expression levels are changing on the targeted branches. Note that this is NOT a traditional differential expression framework using linear regression models with control vs. treatment. The goal of PhyDGET is to estimate species-specific expression levels for each gene and examine them phylogenetically as quantative traits.

How do I cite PhyDGET?

Citation forthcoming.

Please also include the URL https://www.github.com/peaselab/phydget in your methods section where the program is referenced.

You should also cite BayesTraits (see info at URL below)

Getting Started

Requirements

- Python 3.x (2.x will not work) https://www.python.org/downloads/
- Numpy for Python3 http://www.numpy.org
- Scipy for Python3 https://www.scipy.org
- BayesTraits (V3+) https://www.evolution.reading.ac.uk/SoftwareMain.html (We have tested versions V3 through V4.1.3 and all should work identically for PhyDGET)

Installation

No installation of PhyDGET itself is required, the scripts should work as long as the Requirements (above) are installed. The repository can be cloned or downloaded as a .zip file from GitHub.

```
git clone https://www.github.com/peaselab/phydget
```

Preparing your data

Phylogeny

An ultametric phylogenetic tree in Nexus format should be used. The tree will be passed to BayesTraits directly, so we recommend consulting the BayesTraits manual further for additional details about the preparation of phylogenetic trees for that software. We recommend preparation of trees using the *ape* package from R. https://cran.r-project.org/web/packages/ape/

RNA-Seq Count File

Data file should be a tab-separated file with a single header line and gene names in the first column.

Basic usage

PhyDGET can be run with all flags in the command line:

```
python3 phydget.py --data DATAFILE.tsv --tree TREEFILE.nwk --out
OUTPUT.txt --models MODEL1:S1  --samples S1:S1a,S1b,S1c --samples
S2:S2a,S2b,S2c ...
```

or by placing command line flags in a plain-text file, with one line per flag.

```
python3 phydget.py JOBFILE
```

Example of jobfile:

```
# PhyDGET Job Command File
--data DATAFILE.tsv
--tree TREEFILE.nwk
--threads 4
--transform log2cpm
--out OUTPUT.txt
--bt-exec BayesTraitsV3
--models M1:S1
--models M2:S2
--models M12:S1+S2
```

```
--samples S1:S1a,S1b,S1c
--samples S2:S2a,S2b,S2c
--samples S3:S3a,S3b,S3c
--samples S4:S4a,S4b,S4c
```

Note: Lines startwith with # will be ignored and can be used for keeping notes or metadata in the job file.

- Individual sample names (S1a, S1b, S2a, etc.) must match headers in the `--data` counts table.
- Sample names (S1, S2, etc.) must match the tips of `--tree` phylogeny and specified names in the `--model` parameters (see below).

Specifying Models

The model syntax of PhyDGET accommodates one or more alternative rate categories for quantitate trail value shifts on a branch. There is no need to specify the null model, it will run automatically. Note that for all models, the `--sample` must be provided (see Single Terminal Branch below for syntax).

Single Terminal Branch

For a single alternative rate category on a terminal branch, the syntax is:

```
--model MODEL1:S1 --sample S1:S1a,S1b,S1c --sample S2:S2a,S2b,S2c
--sample S3:S3a,S3b,S3c --sample S4:S4a,S4b,S4c
```

This will attach an alternative rate category for the branch leading to taxon `S1`. This species label must match your phylogeny tip label exactly.
The labels corresponding to each species must be included as several `--sample` lines in the format shown above. These sample labels must match the headers on the expression tabular data file exactly. All species on the tree provided must have a `--sample` entry.

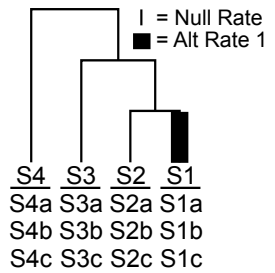

Figure 1: Model Example 1: Single Terminal Branch

Single Ancestral Branch

For a single alternative rate category on ancestral branch, the syntax is:

`--model MODELNAME:S1+S2+S3`

This places the most recent common ancestor branch of S1, S2, and S3 in the alternative rate category. The `+` signs are used to separate the taxa jointly specifying their most recent common ancestor.

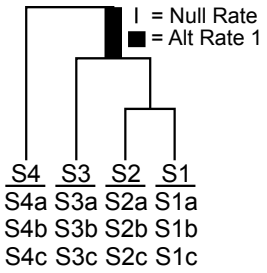

Figure 2: Model Example 2: Single Ancestral Branch

Multiple Branches, Single Rate

`--model MODELNAME:S1+S2+S3,S1`

This places the S1+S2+S3 ancestral branch and the S4 terminal branch both under a single common alternative rate category. The `,` separates different branches that are under the same rate.

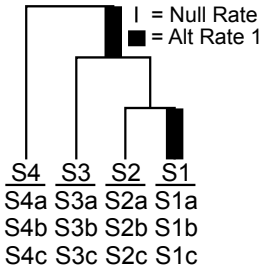

Figure 3: Model Example 3: Multiple Branches under a single rate

Multiple Branches, Multiple Rates

`--model MODELNAME:S1,S2:S3`

This places the S1 and S2 branches under one alternative rate category, and the S3 branch under a separate alternative rate category. The `:` separates the branch or branches under different rate categories.

Transform

The options for transformation (`--transform`) are:

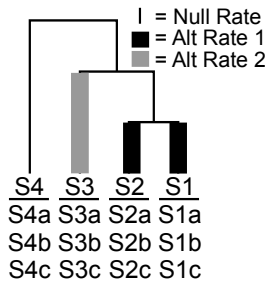

Figure 4: Model Example 3: Multiple Branches under multiple rate

- `log2cpm` (default): Divide each value by the total number of counts per sample times 10^6 (counts-per-million), then log2 transform.
- `log2`: Logarithm base 2 transfomation without sample-size normalization.
- `cpm`: Divide each value by the total number of counter per sample times 10^6 (counts-per-million).
- `none`: Do not transform the data. Not recommended unless you data are already log-transformed by your own method, or you are using data other than raw expression data.

BayesTraits Parameters

The specifics of the parameters for BayesTrait are listed in the BayesTrait manual.

Program Parameters

---

phydget

PhyDGET: Phylogenetic Differential Gene Expression Tool
Author: James B. Pease

Parameters

`--data` (required) = input expression data filepath (csv format) (type=file path, default=None)

`--out` (required) = output file path (csv format) (type=file path, default=None)

`--tree` (required) = input tree file path (Nexus format) (type=file path, default=None)

`--bt-burnin/--btburnin` = BayesTrait number of burn-in steps. (type=integer, default=1000000)

`--bt-exec/--btexec` = BayesTrait executable path (type=None, default=BayesTraitV3)

`--bt-iter/--btiter` = BayesTrait number of iterations usedper stone in the stepping stone sampling. (type=integer, default=10000000)

`--bt-priors-alpha/--btpriorsalpha` = BayesTrait distribution type and prior range for alpha. (type=None, default=('uniform', -10, 30))

`--bt-priors-sigma/--btpriorssigma` = BayesTrait distribution type and and prior range for sigma^2. (type=None, default=('uniform', 0, 60))

`--bt-priors-vrbl/--btpriorsvrbl` = BayesTrait distribution and prior range for variable rates branch length differential. (type=None, default=('sgamma', 1.1, 1.0))

`--bt-stoneiter/--btstoneiter` = BayesTrait number of iterations usedper stone in the stepping stone sampling. (type=integer, default=20000)

`--bt-stones/--bt-stones` = BayesTrait number of stones usedin the stepping stone sampling. (type=integer, default=200)

`--keep-files/--keepfiles` = Keep all temporary files (flag, default=False)

`--temp-dir/--tempdir` = temporary folder for files (type=None, default=PhyDGETtmp)

`--temp-prefix/--temp-prefix` = Temporary Directory Prefix (type=None, default=PhyDGETRun)

`--test-gene/--testgene` = Enter exacty gene name from first column ofinput csv file to do a test run on a single gene. (type=None, default=None)

`--threads` = Number of threads for parallelization (type=integer, default=2)

`--tip-values/--tipvalues` = Values to place at the tips (see manual for details). (type=None, default=all) Choices: ('all', 'amean', 'hmean', 'gmean', 'median', 'middle')

`--transform` = Data transformation type (see manual for details). (type=None, default=log2cpm) Choices: ('none', 'log2', 'cpm', 'log2cpm')

`--verbose` = extra screen output (flag, default=False)

Version History

- **0.3.0:** First public release
- **1.1.0:** Major Update: (1) Fixed disparity when running from command-line versus using a job file (2) changes to options for priors to make them more specific and customizable to alpha, sigma, and VRBL (3) added additional modes for placing data at the tips including median, mean, etc. in addition to the default of placing all at the tips (4) fixed a minor bug

in the script for non-transformed datasets (5) Renamed the misleading "bestMargin" to "bestL-secondL" to reflect that it is the difference in likelihood between the best and second-best likelihood

License