Supplementary Figures

Figure S1: Quantifying the degree of paraphyly among OTUs. The example tree illustrates how the Polyphyly Index (PI) is calculated. Circles at the leaves of the tree are colored to show the OTU to which the sequence belongs. Circles at internal nodes are used to indicate the common ancestor of all sequences in the OTU of the corresponding color. Monophyletic OTUs (OTU-0 and OTU-4) have a paraphyly index of zero, while OTUs with little phylogenetic coherence (OTU-1 and OTU-2) have PI values closer to one. OTUs that are close to being monophyletic but whose common ancestor has one or two descendants not belonging to the OTU (OTU-3) will have a low, but non-zero PI.

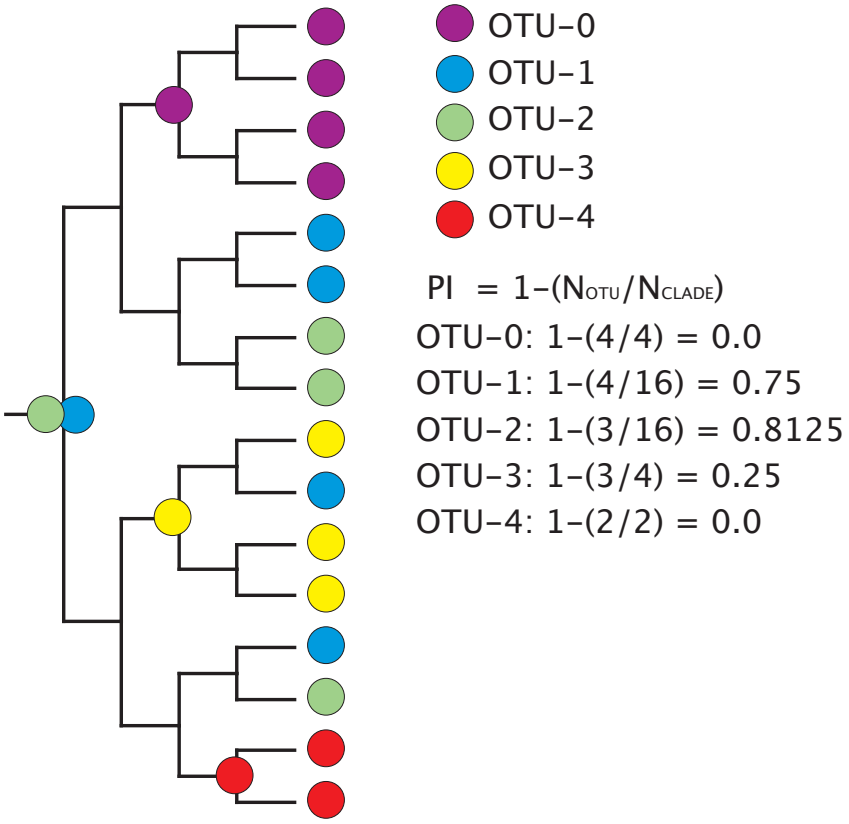


Figure S2: Microdiversity is consistent with the Stable Ecotype Model. The graphs display the sequence diversity curves (plots of the number of OTUs at different sequence identity thresholds) for each major genus analyzed in this study. The 16S rRNA OTUs from the skin dataset are shown in A-E: (A) *Corynebacterium*; (B) *Propionibacterium*; (C) *Staphylococcus*; (D) *Aquabacterium;* (E) *Diaphorobacter*. The curves for the *Vibrio hsp60* sequences (F) and *Synechococcus psaA* sequences (G) are also shown. Note that the scaling is not identical in all graphs, most notably parts F and G, due to the more rapid evolutionary rate of protein-coding genes compared to 16S rRNA.

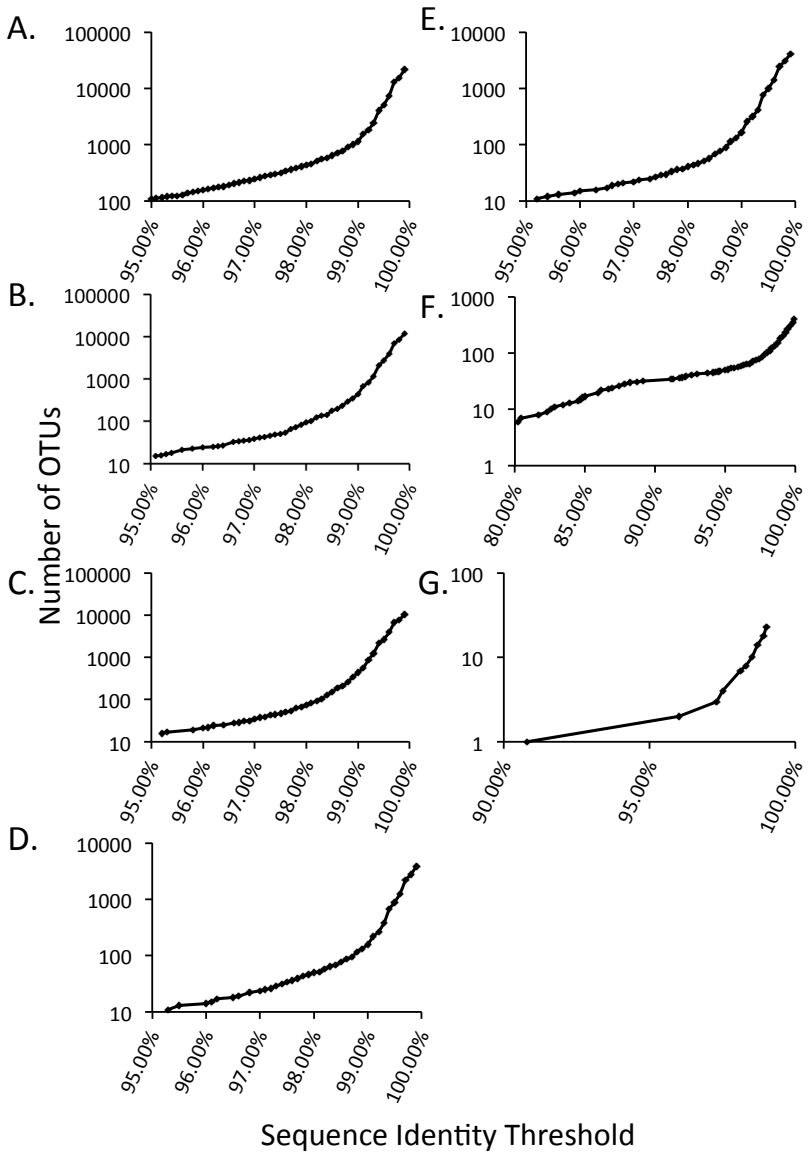


Figure S3: Paraphyly Index at different OTU cut-offs. Formatting of these plots is identical to that of Figure 1 in the main text. As with the 99% OTUs (Figure 1), many of the 97% OTUs (A) and 99.5% OTUs (B) show a high degree of paraphyly regardless of the OTU size.

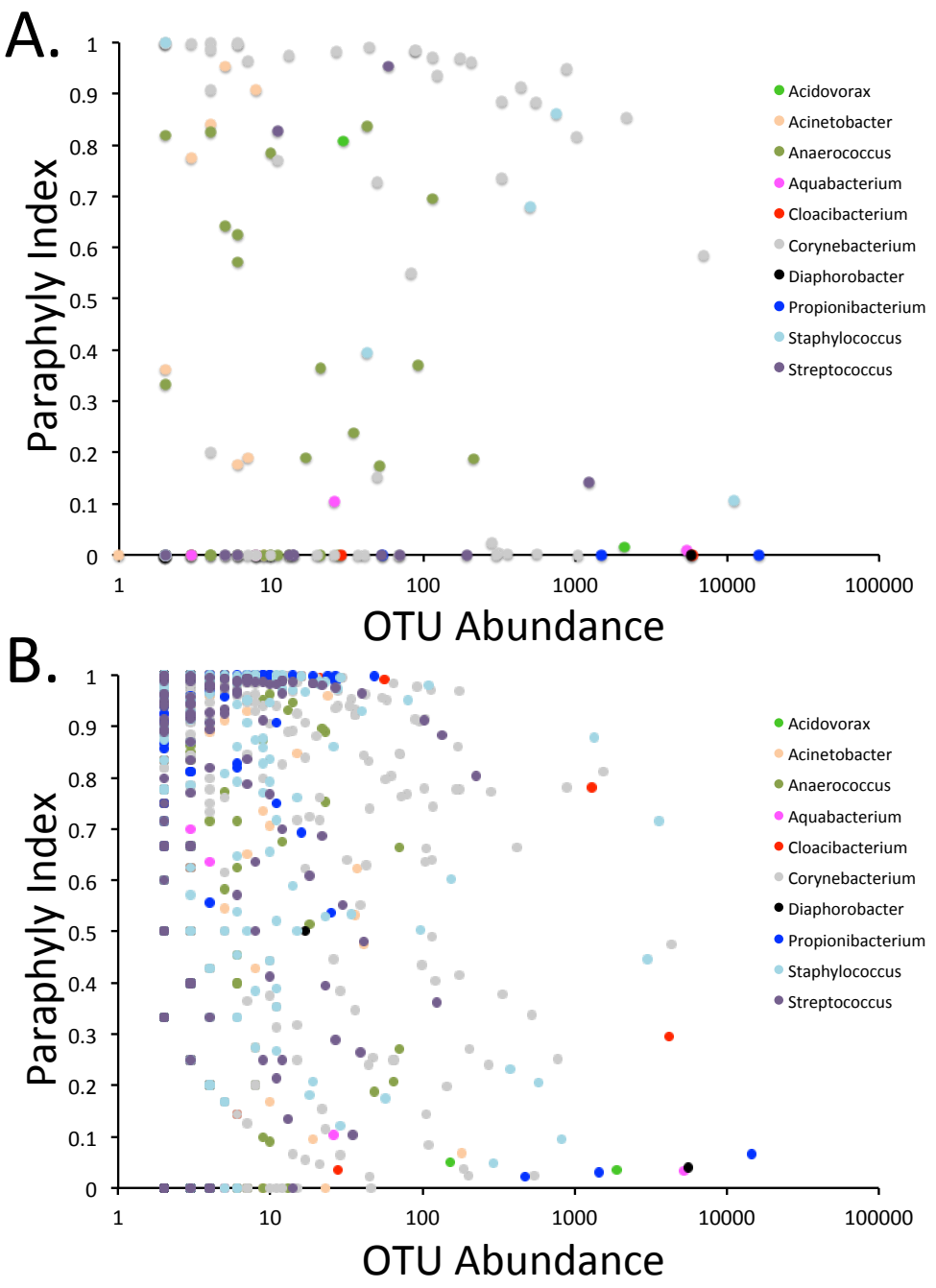


Figure S4: Variation in rates of periodic selection between clades of three genera.

Maximum-likelihood trees display the major subclades of the genera *Diaphorobacter* (A) and *Aquabacterium* (B) from the skin dataset, and of marine *Vibrio* (C). Leaves and branches within these clades have been replaced with triangles. The height of the triangle indicates the number of sequences represented, and the width represents the distance from the ancestral node to the tip of the longest branch. Clades are shaded to reflect statistically significant differences in the rates of periodic selection estimated by complete ES for each clade. The rate categories are relative within the genus, and should not be compared between genera. The mean rate for each category is shown in parentheses.

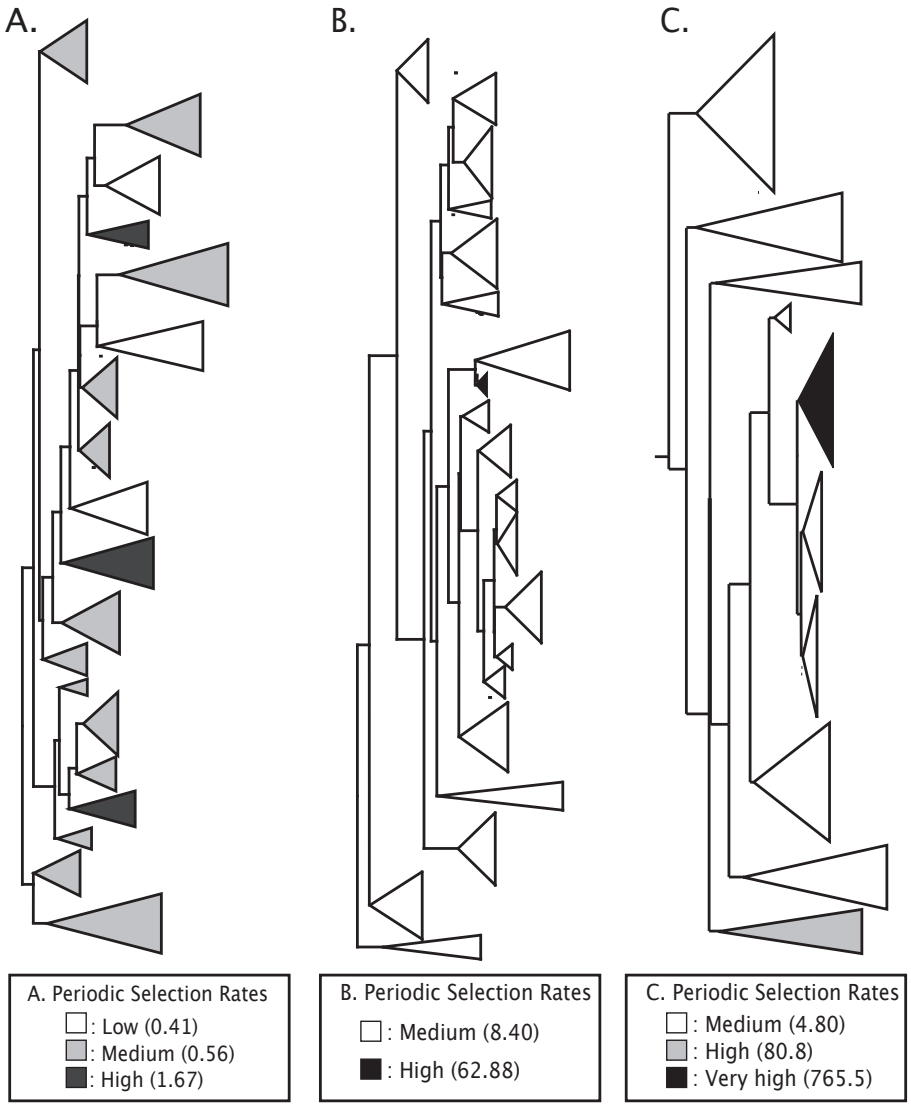


Figure S5: Variation in rates of ecotype formation between clades of three genera.

Maximum-likelihood phylogenies display the major subclades of the genera *Diaphorobacter* (A) and *Aquabacterium* (B) from the skin dataset, and of marine *Vibrio* (C). Formatting is exactly the same as in Figure S4, except that clade shading here reflects statistically significant differences in the rates of ecotype formation.

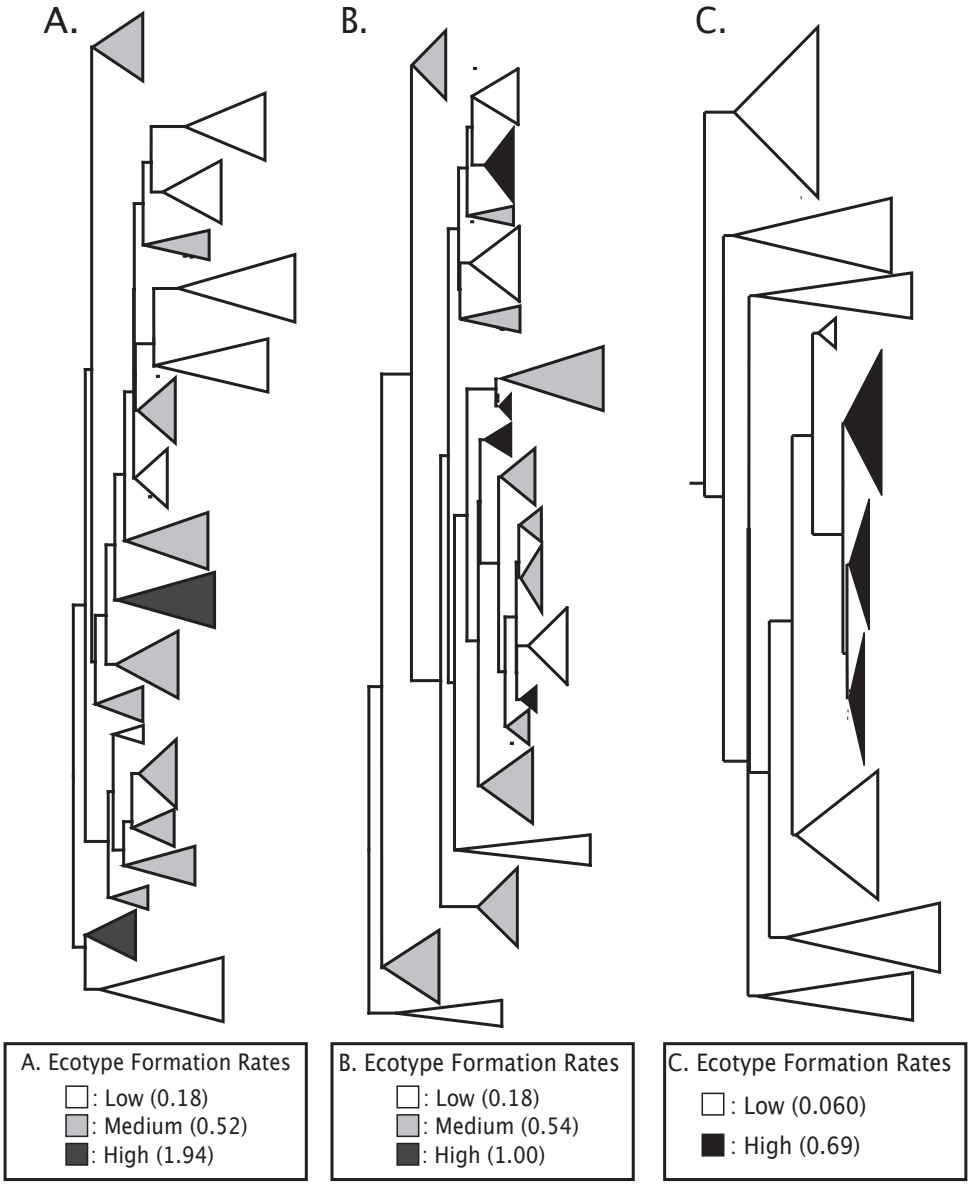


Figure S6: QuickES Validation results for *Bacillus simplex* dataset. This figure shows a maximum likelihood phylogeny of 116 sequences of *B. simplex*. The tree is based on a concatenation of three protein-coding genes (*gapA, rpoB,* and *uvrA)*. Ecotypes for these sequences have previously been demarcated by the full ES algorithm. Three trials were run in each QuickES replicate. The best of three trials was shown for each replicate.

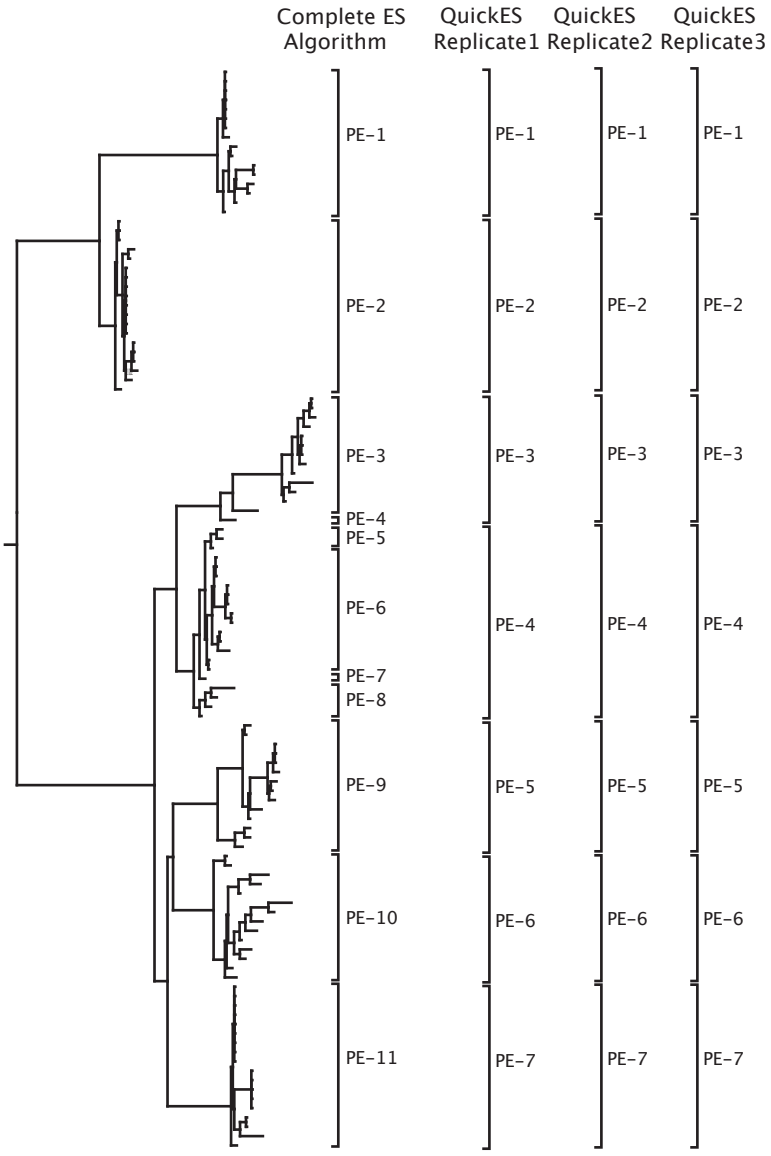
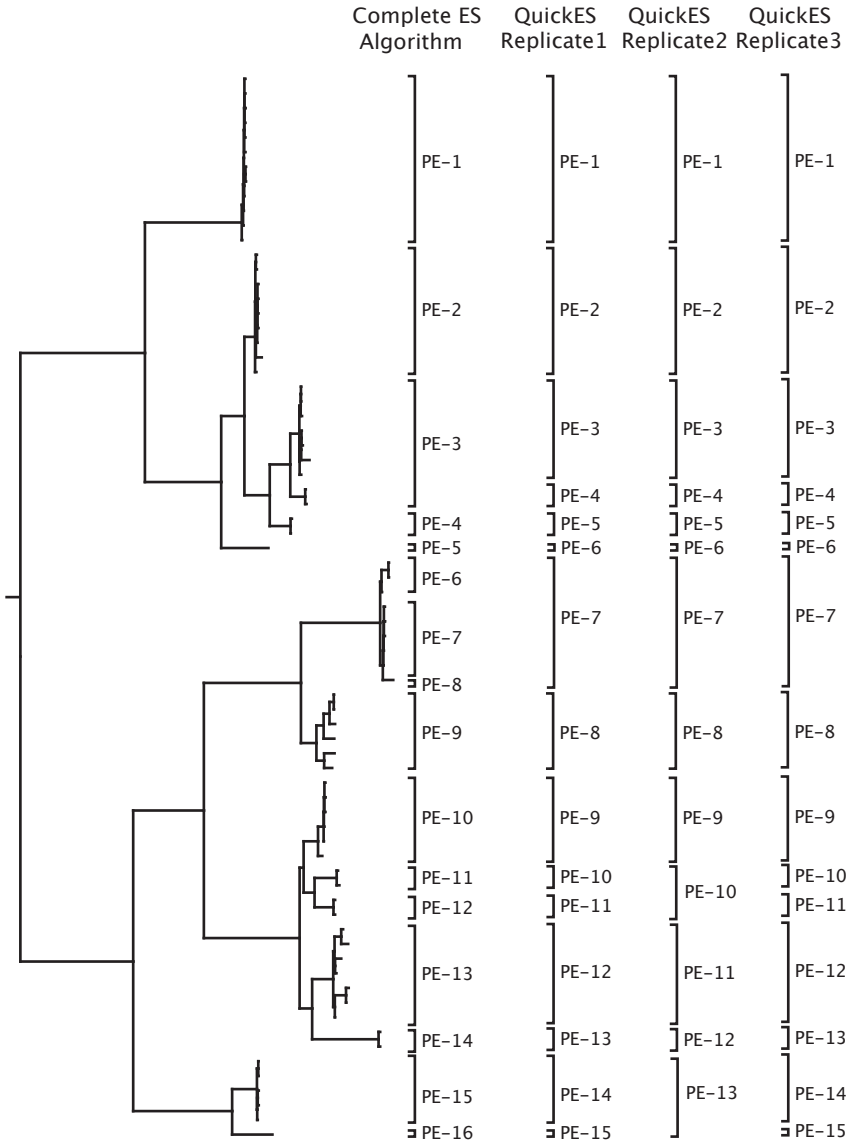


Figure S7: QuickES Validation results for *Bacillus subtilis-licheniformis* dataset. This figure is identical in format to Figure S6, except that it depicts a tree based on a concatenation of the *gapA, gyrA, and rpoB* genes from samples of *B. subtilis* and *B. licheniformis*.



Supplementary Tables

Table S1: Paraphyly among OTUs of marine *Vibrio* dataset. This table displays the number of monophyletic OTUs in the genus *Vibrio* at five different identity thresholds (85%, 90%, 95%, 97% and 99%). As in Tables 1 and 2, only OTUs containing at least two sequences were considered.

OTU Identity Threshold	Number of OTUs	% Monophyletic
85%	15	80.00%
90%	27	92.59%
95%	36	88.89%
97%	49	81.63%
99%	114	74.56%

Supplementary Methods

Comparison of periodic selection and ecotype formation rates within genera.

Genera were subdivided into clades as described in the main text. Ten replicate estimates of the periodic selection and ecotype formation rates were generated for each clade using the original ES algorithm. A Tukey-Kramer test was then used to determine clades that had rates significantly lower or higher than the rates in other clades. Clades were then grouped into categories such that the clades in each category had rates significantly different from the clades in other categories, but not significantly different from the clades within the same category. Typically, one category contained the majority of the clades in a genus, and this category was designated as "medium", with categories with significantly higher rates designated as "high" and significantly lower rates designated as "low". These categorizations are relative to other clades in the same genus, and should not be compared between genera. These rate categories are displayed in Figures S4 and S5.

Running QuickES

QuickES, version 1.0

A new wrapper for the Ecotype Simulation algorithm (Koeppel et. al., PNAS, 2008).
QuickES sacrifices some of the precision of the complete algorithm, in exchange for dramatically increased speed, and therefore the number of sequences capable of being analyzed.

Author, Alexander F. Koeppel, Department of Biology, University of Virginia, afk2s@virginia.edu.

INSTALLATION
==========

The program consists of a python wrapper and pre-complied fortran binaries. No special installation is required (beyond what is necessary for standard ES).
Just extract the contents of the archive to your hard drive.

REQUIREMENTS
==========

Linux/UNIX Operating system.
Python 2.7.1 or later.
You must also have the python modules Numpy, SciPy, and BioPython installed.
You will also need the fortran binaries for the original version of ecotype simulation, and to meet all the requirements for running ecotype simulation.
The binaries can be downloaded from http://sourceforge.net/projects/eco-sim/?source=directory along
with a README file that details the requirements for running standard ES.

USAGE (OPTIONAL)--Tree_Splitter_OTU.py
==========

If your dataset contains more than 1000 sequences you will want to split up the data before proceeding with QuickES. Tree-Splitter divides up a phylogeny based on the OTU identity of the sequences within it.

Inputs:
1) A phylogenetic tree in newick format.
2) An OTU file:

This file must have each OTU on a single line. The name of the OTU should appear first, followed by a comma-separated list of the sequence ids in each OTU.

e.g.

OTU-1,seq1,seq2,seq3
OTU-2,seq4,seq5,seq6,seq7
OTU-3,seq8
etc.

3) A fasta file containing the sequences.

NOTE: The sequence ids in all three files must match exactly!

Arguments:
The program requires five arguments:
1) The absolute path the tree file
2) The absolute path to the OTU file
3) An integer from 1-100, describing the stringency with which you want clades to be subdivided by OTU.

i.e. A value of 100 here will insist that all members of a clade belong to the same OTU.

A value of 90 in this argument, means that the script will save any clade for which 90% of the sequences

in the clade belong to the same OTU.

Given the paraphyletic nature of most OTUs we recommend a value of 90, but different values may work better

for different datasets. The idea is to get a large set of clades that are "key" clades, (i.e. between 25 and 200 sequences).
4) The absolute path to the fasta file.
5) The absolute path to a working directory, in which you want the output to appear.

Example Usage: Tree_Splitter_OTU.py /path/to/tree/file.tree /path/to/otu.file 90 /path/to/sequences.fasta /path/to/working-directory

Tree_Splitter_OTU.py Output:
The output of the program should be two directories, which are created by the program within the user-supplied working directory.

1) The directory "All_Clades" will contain all of the clades generated by the program, in both newick tree format, and in the
form of fasta files containing the sequences. These files will be used by QuickES to demarcate ecotypes over the
entire tree.
2) The directory "Key_Clades" will contain fasta files of the sequences in each clade containing between 25 and 200 sequences.

USAGE (MAIN)--QuickES.py
==========

Requirements: To run this script, you will need a working directory that contains the following fortran binaries:

binningdanny.amd64
correctpcr.amd64
demarcationsCI.amd64
divergencematrix.amd64
fredMethod.amd64
readsynec.amd64
removegaps.amd64

These can be downloaded from: http://sourceforge.net/projects/eco-sim/?source=directory

The working directory should also contain two subdirectories:

1) All_Clades: This directory should contain fasta files and corresponding newick format tree files for all clades for which ecotypes will
be demarcated by QuickES. The names of the sequence ids in the files must match. The filenames can be anything you like, but
the tree and fasta filenames must match, and end in .fasta and .tree respectively (e.g. Clade-1.fasta and Clade-1.tree). If your dataset
contains fewer than 1000 sequences you can just put the (correctly named) fasta and tree files for the whole set in this directory. Otherwise,
run the Tree-Splitter script (see above) and this directory will be created automatically (you just need to copy or move it into your working directory)

2) Key_Clades: This directory contains only fasta files for those subclades of your dataset containing between 25 and 200 sequences.
These clades will be used by QuickES to compute the parameter solutions using the Brute Force search. These solutions will then be used to run demarcations

on all of the clades in All_Clades. As with Key_Clades, if you have fewer than 1000 sequences, just put the entire fasta file into this directory.
Otherwise, Tree-Splitter will create the directory for you, just move or copy it into your working directory.

Inputs:
1) A Variables file containing the values for several variables needed by ES (Variables_File.txt).

Format:

```
VARS  pcrerror          8.33E-6
VARS  pcr random seed   20440587
VARS  BF omega range    0.001,10.0
VARS  BF sigma range    0.001,10.0
VARS  BF xnumincs  8,8,8,0
VARS  BF nrep           200
VARS  BF random seed    20325215
VARS  Demarcation Crit  1.5x
BINS  13
BINS  0.65
BINS  0.7
BINS  0.75
BINS  0.8
BINS  0.85
BINS  0.9
BINS  0.95
BINS  0.96
BINS  0.97
BINS  0.98
BINS  0.99
BINS  0.995
BINS  1.0
```

It is recommended that you use the default variables file provided with the software and leave this file alone,

however, in case you need to modify these values to fit with your particular dataset, the variables are as follows

1) The pcr error rate. ES automatically "corrects" a few nucleotide substitutions to account for pcr error.

If you have a good estimate fore the pcr error rate for your data you can substitute it here. Otherwise use the default

value.

2) Random seed for pcr error correction. This line is from an older version of the code and is no longer read directly. Nonetheless do not delete this line.

3) BF Omega range. Range of ecotype formation rates per nucleotide substitution tried by QuickES in the Brute Force search. Do not change these values unless

you have a very good reason to expect that your sequences have a much lower or much higher rate of ecotype formation than is usual.

4) BF Sigma range. Range of periodic selection rates per nucleotide substitution tried by QuickES in the Brute Force search. Do not change these values unless

you have a very good reason to expect that your sequences have a much lower or much higher rate of periodic selection than is usual.

5) BF xnumincs. The number of increments into which QuickES divides the omega and sigma ranges. Increasing these numbers can improve the precision of the Brute Force search, but at a high cost in speed. In order, separated by commas, the increments for omega,sigma,npop,drift.

6) BF nrep. The number of replicates used by QuickES per quartet of parameter values to estimate the likelihood. Increasing this number will yield more precise likelihood estimate, but at the cost of speed.

7) BF random seed. As with PCR random seed, this value is now generated directly by QuickES and is no longer read directly.

8) Demarcation Crit. This line is no longer read, but is instead now a user-supplied argument to QuickES (see below).

9) The number of bin levels.

10-onward) The bin levels to be used by QuickES during binning. It is recommended that these be left alone. However, if you know that all the sequences in your dataset are within a certain sequence identity you can get a slight improvement in performance by eliminating the redundant bin levels (i.e. if all are within 90% identity, you could eliminate the lines reading 0.85 and below), however, if you do this, you MUST then change the number in line 9 to correspond to the total number of bin levels.

2) Tree and fasta files for all clades you wish to analyze, set up in the All_Clades and Key_Clades directories (see above).

Arguments:
The program requires three arguments:
1) The absolute path to the Variables_File.txt file.
2) The stringency criteria for demarcation. Acceptable values are 5x,2x,1.5x,1.25x,1.1x,or 1.05x. The lower the value the more precise a clade sequence diversity match is required between the model and actual data to be considered a "success". We recommend 1.5x as a good balance between precision and speed.

NOTE: Depending on your data, and on your Brute Force search settings, it may not be possible to run all datasets at all values. If you select a value for which no useable parameter solutions are found in the Brute Force search, QuickES will automatically move to the next least stringent criteria (e.g. if you select 1.25x and no useable solutions are found, it will move automatically to 1.5x, then to 2x, and so on). A warning will echo on the screen if this occurs.
3) The absolute path to the working directory.

Output:
QuickES gives two outputs.
1) Parameter_Solutions.txt: This file contains the parameter solutions found by brute force for each "key" clade run. It also shows the average values for each parameter, which are then used for demarcation/
2) *Ecotypes.list. Each clade in the All_Clades file will now have a corresponding ecotype list, showing the sequences belonging to each ecotype. Each line of the file will contain one ecotype.

The name of the ecotype appears first, followed by a comma-separated list of the sequences belonging to that ecotype. For example:

Clade1_PE-3,seq7,seq8
Clade1_PE-1,seq1,seq2,seq3,seq4
Clade1_PE-2,seq5,seq6
etc.

NOTE: We recommend that QuickES.py be run a minimum of three times, and the solution with the highest likelihood value (found in Parameter_Solutions.txt) be used as the final result.