

Posterior Analysis of Coalescent Trees

Trevor Bedford

Department of Ecology and Evolutionary Biology and Howard Hughes Medical
Institute, University of Michigan, Ann Arbor, MI 48109, USA.

Email: bedfordt@umich.edu

Website: www.trevorbedford.com

June 29, 2010

Contents

1	Overview	2
2	Installation	4
3	Tree import	4
4	Parameter input	5
5	Tree drawing	5
6	Tree manipulation	6
7	Summary statistics	9
8	Skyline statistics	10
9	Tip statistics	11
10	Combined analyses	11

1 Overview

The program PACT computes a variety of statistics from a sample of genealogical trees. It is meant to extend the functionality of already existing coalescent inference programs such as Migrate [1,2], BEAST [3], IM [4,5] and LAMARC [6]. PACT reads genealogies in NEWICK format and performs various operations on these genealogies. PACT is designed to work with both structured genealogies and also with genealogies assembled from temporally spaced sequence data.

The functionality of PACT is highly modular, relying on combinations of tree manipulation operations and summary statistics to produce useful results. For example, the operation `tmrca` returns the TMRCA of the entire tree, but when combined with the `prune to label`, it returns the TMRCA of samples with a specified label. In another example, diversity at a specified point in time may be calculated with a combination of `time slice` and `diversity`.

Operations available in PACT.

General

`burnin` removes an initial set of trees from the analysis

Tree manipulation

`push times back` adjust dates associated with tips
`reduce tips` prune tips from tree at random
`renew trunk` adjust the trunk of the genealogy by tracing backward from recent samples
`prune to trunk` reduces tree to just the trunk of genealogy
`prune to label` reduces tree to just the ancestral lineages of a subset of labeled tips
`collapse labels` resets the entire tree to a single label
`trim ends` reduces tree to the branches between time x and time y
`section tree` break tree up into temporal sections
`time slice` reduces tree to all ancestors of lineages that exist at a particular time

Tree output

`print rule tree` prints out the highest posterior tree in rule-list format

Tree statistics

Each of these may be used with `summary` or with `skyline`

`tmrca` returns the time to the most recent common ancestor of all tips
`length` returns length of tree
`proportions` returns proportion of total tree length of each label
`coal rates` returns rates of coalescence, separate rates for each label
`mig rates` returns rates of migration, separate rates for each label pair
`diversity` returns the average time to coalescence for pairs of lineages
`fst` returns the relative diversity between labels compared to within labels
`tajima d` returns Tajima's D

Tip statistics

`time to trunk` returns time to coalescence with trunk for each tip in the tree

2 Installation

PACT is a small program and installation should be simple. On UNIX systems (Mac OS X and Linux), the source code can be compiled by navigating to the directory `source` and running the command `make`. The program `make` is not installed on a Mac by default. You need to run an optional install of Apple Developer Tools. The binary `pact` will be created within the `source` directory. This binary can then be moved anywhere it is convenient. Placing a copy in `usr/bin/` will allow command-line access from within any directory (assuming the `PATH` variable is set correctly).

I've supplied executables for both Mac Intel and Windows. You shouldn't need to compile unless you're running on a PowerPC Mac or on Linux. Because I don't have ready access to a Windows machine, I compiled the Windows version using [MinGW](#). Let me know if you have any difficulties with running it.

3 Tree import

PACT requires a text file titled `in.trees` to be within the working directory. This file is expected to be a line-by-line list of NEWICK trees, like so:

```
(((((0ZFCOHT:0.0000060778,(((0BUKMLJ:0.0008690960,0JHXRX:0.0002.....
((((((((1NRVIII:0.0002334233,0RZJHGM:0.0073234023 [&M10:0.00174595.....
((((0CGSUVH:0.0013774643,(((0BUKMLJ:0.0006668040,0JHXRX:0.0000.....
((((((((0ITGYFC:0.0001063175,(((0QUYTLG:0.0012714366,0YRCXIQ:0.....
((((((((1XDDTKE:0.0019800936,1GPPRVZ:0.0003301111):0.0009091307,1JFZ.....
```

Node names can be any length, but must be entirely comprised of 0-9, A-Z, a-z, -, . and underscore characters. For compatibility with Migrate's structured coalescent functionality, the first characters must indicate the tip's deme, starting at 0.

Rooted trees are required. Migration events are assumed to follow the Migrate style of [`&M10 :0.0017`] indicating a migration event (forward in time) from 1 to 0 at a distance of 0.0017 from the end of the branch in question.

Tree output from BEAST can be used as is. The initial tip listing will be ignored automatically. PACT supports the labeling of nodes via the `ancestralTreeLikelihood` function of BEAST. In the tree file, nodes are immediately followed by [`&state="XXXX"`]

For both Migrate and BEAST, PACT reads in each tree and tags each with its likelihood. The `treefile` produced by Migrate or the `XXXX.trees` file produced by BEAST can should be renamed to `in.trees` for use with PACT.

Genealogies from LAMARC may also be used as is. Unfortunately, to the best of my knowledge LAMARC can only output the highest probability tree, rather than the full sample of trees, even though it outputs distributions for numeric parameters.

Currently, the tree likelihood is only used to determine which tree to print with `print rule tree`. All statistics treat each tree equally.

4 Parameter input

PACT also requires a file titled `in.param` that lists the operations to perform on the trees. A complete listing of options can be seen in the file `parameters.txt`. A `#` comments out anything that follows. Each command gets its own line. The ordering of commands in the input file does not matter. Operations are described in the following sections.

5 Tree drawing

Rather than write extensive plotting functions in C, I've relied on Mathematica to render my graphics output. The Mathematica notebook `analysis.nb` contains routines to render trees and other output. The operation `print rule tree` prints out the highest likelihood tree to the file `out.rules` in rule-list format. If trees are not tagged with likelihoods, then `print rule tree` prints out the final tree in `in.trees`. The rule-list output looks like so:

```
526 518 521 2256 502 473 464 2196 469 712 2508 711 2506 706 704 705 .....
3372 3370 3367 3365 3354 3353 3344 3320 3314 3312 3311 3300 3299 3298 ....
2284->3372 2282->2284 2271->2282 526->2271 2263->2271 518->2263 .....
3372->1 2284->6 2282->6 2271->6 526->6 2263->6 518->2 2261->7 .....
3372->{2003.77,33.433} 2284->{2004.08,11.2188} 2282->{2004.56,11.2188} ...
526->"5EU502063" 518->"1EU502381" 521->"3EU501873" 502->"3EU502004" .....
```

Each node of the tree has been assigned an arbitrary number. The first line is a listing of all the tips in the tree. The second line is a listing of all nodes comprising the trunk of the tree. The third line is a mapping of child nodes to parent nodes. The fourth line is mapping of nodes to their labels (demes). The fifth line is a mapping of nodes to xy -coordinates. The sixth line is a mapping of tips to the names that were supplied in `in.trees`.

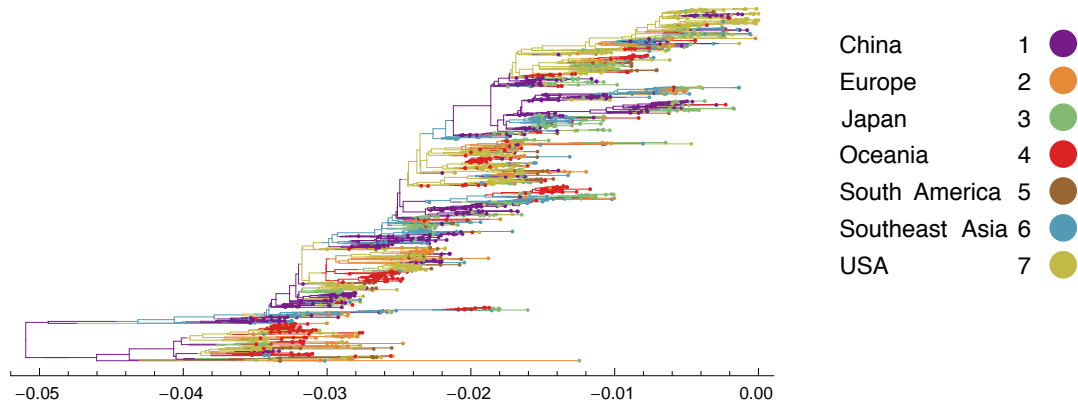
If you prefer some other program to Mathematica, it should be fairly straight-forward to still use the file `out.rules` for tree drawing.

Because coalescent genealogies are rooted, x -axis locations are simply determined by a node's date. Tree layout on the y -axis is accomplished in three stages. First, sister nodes are sorted so that the first sister node always has fewer descendant nodes than the second sister node. This causes more successful lineages to be sorted higher on the y -axis than less successful lineages. In a temporally spaced genealogy, this usually places the trunk of the genealogy higher on the y -axis than all other lineages. Second, a preorder traversal is performed on the tree, and each time a tip is encountered, its y coordinate is incremented. Finally, a postorder traversal is performed. A node with one daughter node takes the daughter node's y coordinate as its own. A node with two daughter nodes takes the average of the daughter's y coordinates.

Here is an example tree output from running `analysis.nb` on `out.rules`. The data are sequences of the HA gene of influenza A. Worldwide samples were taken from 2002 to 2008. Migrate was run on these sequences to estimate the migration rates between

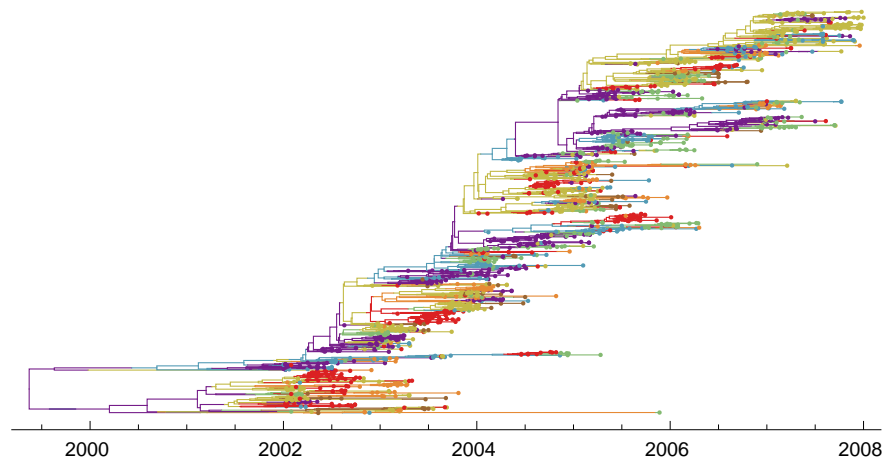
different regions of the world and the likely genealogies connecting the sequences. If you're interested, you can find the full analysis in Bedford *et al.* 2010 [7]. I'll use this tree in examples throughout the rest of the manual.

`print rule tree`. This genealogy comes directly from Migrate. Migrate reports trees scaled in terms of substitutions per site.

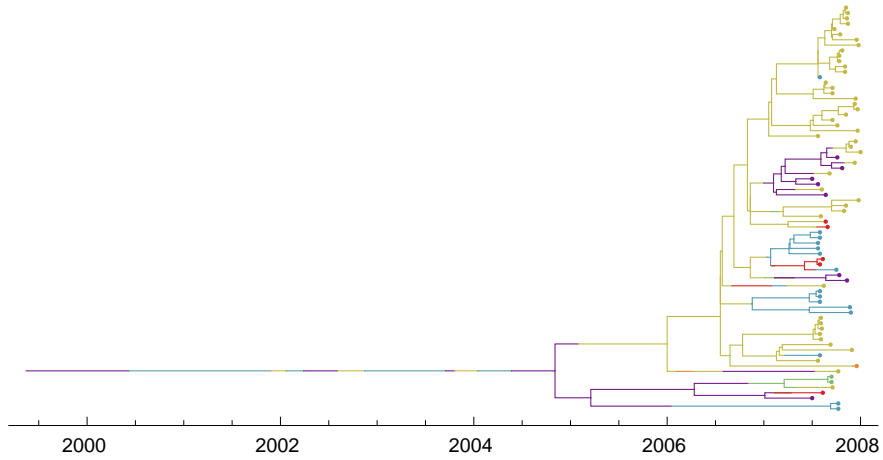


6 Tree manipulation

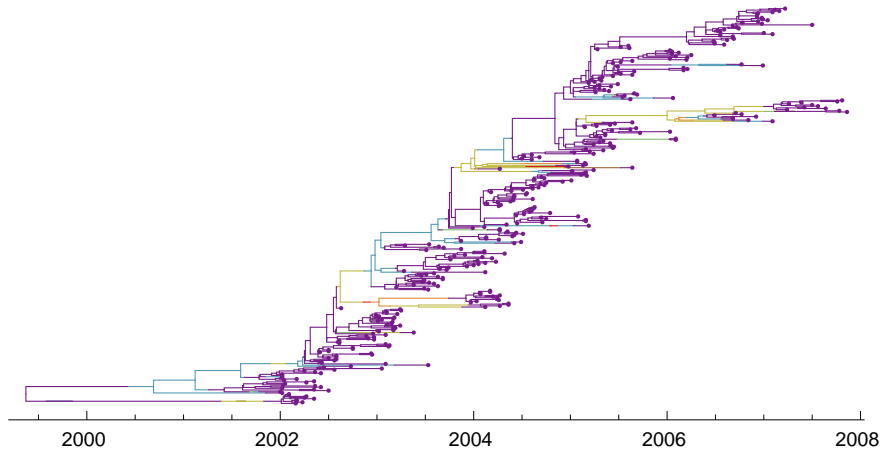
`push times back 2002 2008`. The function `push times back` rescales the branch lengths of the tree. Here, 2002 is the date of the oldest sample and 2008 is the date of the youngest sample. Calling `push times back 10` adjusts dates so that the most recent sample is dated 10 rather than 0, but does not modify branch lengths. `push times back 2002 2008` has been called in all the following examples.



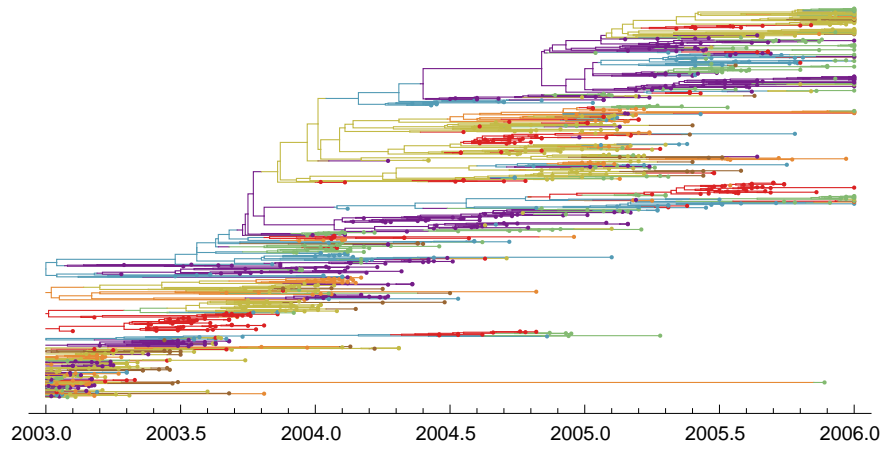
`renew trunk 0.5`, `prune to trunk`. The function `renew to trunk` redefines the genealogy trunk by working backward from recent samples. Here, samples from $2008 - 0.5 = 2007.5$ to 2008 are considered when working backwards. If `renew trunk` is not called, the trunk defaults to the ancestors of samples present in the final 1/100 of the genealogy. The function `prune to trunk` removes all non-trunk lineages. The root of the tree is not adjusted.



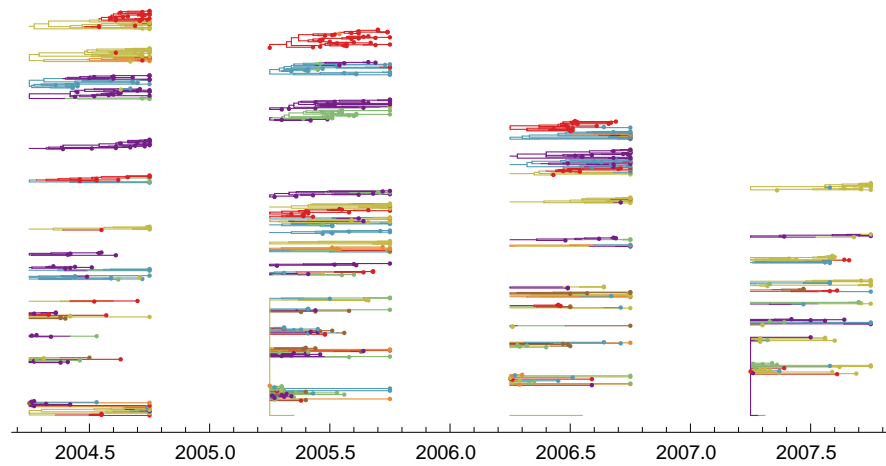
`prune to label 1`. The function `prune to label` works backward from only those tips with a particular label, keeping only their ancestors. Here, only samples with label 1 (China) are considered. The root of the tree is the MRCA of these lineages.



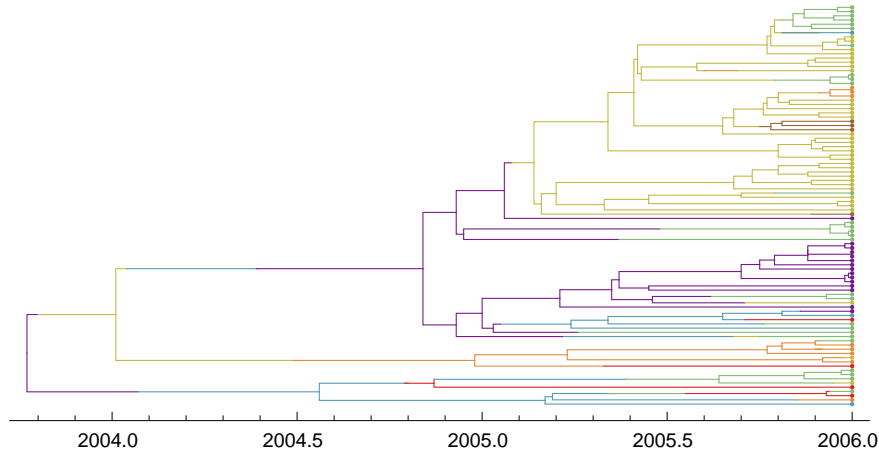
`trim ends 2003 2006`. The function `trim ends` prunes the tree to only those lineages existing between two dates. Here, only lineages extant between 2003 and 2006 are kept.



`section tree 2004.25 0.5 1`. The function `section tree` cuts the tree up into multiple evenly-spaced windows. Here, starting from 2004.25, windows of 0.5 years are taken, moving forward 1 year to get to the start of the next window.



`time slice 2006`. The function `time slice` takes all lineages that exist at a single point in time and traces their ancestry. Here, lineages and their ancestors that exist precisely at 2006 are kept. The root of the tree is the MRCA of these lineages. With temporally spaced samples, this function is a necessary prerequisite to get sensible results from `diversity`, `fst` and `tajima d`.



7 Summary statistics

Summary statistics print to the file `out.stats`. This file is a tab-delimited text file.

statistic	lower	mean	upper
<code>tmrca</code>	8.14525	8.62911	9.18639

The mean is the numeric mean of all of the sampled trees. The 95% credible interval is determined by the empirical distribution function of the sample of trees. The tree value immediately under the 2.5 percentile is averaged with the value immediately above the 2.5 percentile to get the lower end of the credible interval. A similar procedure is used for the 97.5 percentile to get the upper end of the interval.

Throughout the following I assume that the genealogy has been scaled in terms of years. Everything works perfectly scaled otherwise.

summary `tmrca`. Gives the span of time from the most recent tip of the tree to the root of the tree. The MRCA of the flu genealogy exists 8.63 years before the most recent sample, or at $2008 - 8.63 = 1999.37$.

summary `length`. Gives the total length of the tree, summing over each branch. The total length of the flu tree is 442.85 years.

summary `proportions`. Gives the proportion of each label relative to the total length of the tree. In the flu tree, branches residing in deme 1 (China) account for 21.6% of the total tree length.

summary `coal rates`. Gives the rate of coalescence within each label, measured in terms of coalescent events between each pair of lineages per year. In the flu tree, deme 1 (China) has 0.58 coalescent events per pair of lineages per year. The rate

of coalescence is inversely proportional to the effective population size. The effective population size of deme 1 (China) would be $1/0.58 = 1.72$ years per coalescent event \times 91.25 generations per year = 157.33 generations = 157.33 individuals (Wright-Fisher model) or 314.66 individuals (Moran model).

summary mig rates. Gives the rate at which labels change over the course of the genealogy. Each pair of labels is considered. Rates are measured as migration events per lineage per year. The observed rate of transition from deme 1 (China) to deme 6 (Southeast Asia) in the flu genealogy is 0.34 migration events per lineage per year.

summary diversity. Compares each pair of tips in the tree and reports the average TMRCA across each pair of tips. This does not have a ready interpretation for a temporally sampled genealogy. With tip-dated trees, `time slice` should be called as well to get the diversity at a specific point in time. Running `time slice 2006, summary diversity` on the flu tree gives 2.55 years of evolution separating pairs of lineages in the population.

summary fst. Compares diversity between pairs of tips with different labels π_b to diversity between pairs of tips with the same label π_w . F_{ST} is calculated as $(\pi_b - \pi_w)/\pi_b$. As with diversity, `time slice` should be run with temporally sampled trees. For the flu genealogy, running `time slice 2006, summary fst` gives 0.24, indicating moderate subdivision.

summary tajima d. Compares diversity to the total length of the tree. Tajima's D is calculated as $D = \pi - S/a_1$, where π is diversity, S is total tree length and a_1 is a normalization factor. Positive D indicates longer internal branches compared to the standard coalescent expectation and negative D indicates longer tip branches compared to the standard expectation. Running `time slice 2006, summary tajima d` on the flu genealogy gives -2.10 , suggesting either population growth or selection.

8 Skyline statistics

Skylines were first introduced to summarize changes in the effective population size over the course of a coalescent genealogy [8]. However, the same logic works for any other summary statistic. To calculate a skyline statistic, I break the genealogy up into multiple temporal sections and calculate the statistic for each of them. Temporal sampling isn't necessary to estimate skyline statistics, although without it, there is significantly less power to estimate summary statistics in the more distant past.

For some statistics, such as diversity, I take a `time slice` at the center of each window of time. In these cases, time points are not independent of one another.

To run any skyline statistics, the command `skyline settings` must first be run. `skyline settings 2002 2007 1.0` sets up windows of 1 year starting at 2002 and ending at 2007. Skyline statistics are output to the file `out.skylines`. This is a tab-delimited text file.

statistic	time	lower	mean	upper
tmrca	2002.5	0.500	0.652	0.797
tmrca	2003.5	1.105	1.199	1.303
tmrca	2004.5	1.995	2.070	2.134
tmrca	2005.5	2.805	3.034	3.136
tmrca	2006.5	1.063	1.250	1.394

`skyline tmrca`. Automatically runs `time slice` at the center point of each time window. Runs `summary tmrca` on each time slice.

`skyline length`. Runs `time slice` at the center point of each time window. Runs `summary length` on each time slice.

`skyline proportions`. Starts by running `trim ends` to separate time windows. Runs `summary proportions` on each time window.

`skyline coal rates`. Runs `trim ends` to separate time windows, and then runs `summary coal rates` on each time window.

`skyline mig rates`. Automatically runs `trim ends` to separate time windows. Runs `summary mig rates` on the resulting time windows.

`skyline diversity`. Starts by running `time slice` at the center of each time window. Runs `summary diversity` on each time slice.

`skyline fst`. Runs `time slice` at the center of each time window, then runs `summary fst` on the resulting slices.

`skyline tajima d`. Runs `time slice` on each time window. Runs `summary tajima d` on each time slice.

9 Tip statistics

These are statistics that apply to individual samples, rather than to the entire tree. Tip statistics are output to the tab-delimited file `out.tips`.

statistic	name	label	time	lower	mean	upper
time_to_trunk	6CY039880	7	2002.05	0.987	1.320	1.848
time_to_trunk	6CY003560	7	2002.47	0.660	0.790	0.966
time_to_trunk	6CY006475	7	2002.01	0.942	1.275	1.803
time_to_trunk	6CY006587	7	2002.1	1.038	1.371	1.899

`tips time to trunk`. Gives the time to takes for each tip to coalesce with the trunk of the genealogy. In a genealogy sampled from a single moment in time, everything will be considered trunk, and so this will be equivalent to the time to coalesce with any other branch.

10 Combined analyses

Much of the utility of PACT comes from the ability to combine tree manipulation operations with statistical calculations. For instance, this is the only way to calculate

diversity in a temporally sampled genealogy. Here are some more ideas for combined analyses:

prune to trunk, skyline proportions. This gives the posterior probability that the trunk bears a particular label at any given point in time.

time slice, prune to label, diversity. This gives the diversity between only those lineages bearing a particular label.

Copyright

This document and all source code of PACT is Copyright 2009-2010 Trevor Bedford.

Acknowledgments

I thank Peter Beerli, Mercedes Pascual, Denise Kühnert and Mike Boyle for helpful advice and feedback on this project.

References

- [1] Beerli, P. & Felsenstein, J. Maximum likelihood estimation of a migration matrix and effective population sizes in n subpopulations by using a coalescent approach. *Proc Natl Acad Sci U S A* **98**, 4563–4568 (2001).
- [2] Beerli, P. Comparison of bayesian and maximum-likelihood inference of population genetic parameters. *Bioinformatics* **22**, 341–345 (2006).
- [3] Drummond, A. J. & Rambaut, A. BEAST: Bayesian evolutionary analysis by sampling trees. *BMC Evol Biol* **7**, 214 (2007).
- [4] Nielsen, R. & Wakeley, J. Distinguishing migration from isolation: a Markov chain Monte Carlo approach. *Genetics* **158**, 885–896 (2001).
- [5] Hey, J. & Nielsen, R. Multilocus methods for estimating population sizes, migration rates and divergence time, with applications to the divergence of *Drosophila pseudoobscura* and *D. persimilis*. *Genetics* **167**, 747–760 (2004).
- [6] Kuhner, M. K. LAMARC 2.0: maximum likelihood and Bayesian estimation of population parameters. *Bioinformatics* **22**, 768–770 (2006).
- [7] Bedford, T., Cobey, S., Beerli, P. & Pascual, M. Global migration dynamics underlie evolution and persistence in human influenza A (H3N2). *PLoS Pathogens* **6**, e1000918 (2010).
- [8] Drummond, A. J., Rambaut, A., Shapiro, B. & Pybus, O. G. Bayesian coalescent inference of past population dynamics from molecular sequences. *Mol Biol Evol* **22**, 1185–1192 (2005).