BaTS – Bayesian Tip-association Significance testing

Joe Parker, Viral Evolution Group, Department of Zoology, University of Oxford.

Version 1.0 Documentation

Introduction

Welcome to the documentation for this version of `BaTS`, version 0.9. This is the first publically-available version to be released. I hope you find `BaTS` accessible and of use to you in your research.

`BaTS` was essentially concieved in response to a specific problem encountered in my own studies and although a number of changes have been incorporated in this (beta) release that allow a wider range of problems to be addressed, it remains a fairly inflexible tool, both in terms of technical requirements and logical problems that can be solved. Of course, over time, other researchers may yet find `BaTS` useful in situations as-yet unthought of by - us.

I would gratefully like to acknowledge Oli Pybus and Andrew Rambaut, my supervisors, whose direction and input brought `BaTS` to this point, and Aris Katzourakis, Rob Belshaw, Philippe Lemey, Simon Ho and Beth Shapiro for adivce and help.

(For the specific Bayesian approach to quantifying phylogeny-trait associations, as well as an exploration of the three statistics and discussion of their merits, users are encouraged to read (Parker et al*, 2007) which should also be used as the preferred reference when citing* `BaTS`*.)*

License & Disclaimer

License

`BaTS` is supplied under the GNU Lesser General Public License, Version 3. This is an open-source software liscence, and others are authorised and encouraged to examine and modify code if they see fit, *as long as* the contribution of previous workers is recognised. For more details see http://www.gnu.org/licenses/lgpl.html

Disclaimer

No guarantee of the **functionality** of this software, or of the **accuracy of results** obtained using it is made, expressed or implied. The programmers, authors and editors of this documentation and the institutions they represent **will not be held responsible** for any errors of analysis, damage to software or hardware, or other losses incurred as a result of using this programme.

What is BaTS?

This software aims to provide a method by which the degree to which phenotypic traits seen in a phylogeny are associated with ancestry are correlated. In other words, where a set of character states are seen on the tip of a phylogenetic tree, is any given taxon more likely to share a character state with a sister taxon than we would expect due to chance?

This problem has been posed in a variety of contexts over the last three decades, particularly molecular epidemiology and phylogeography. A number of approaches have been developed over the years, of which the method of Slatkin & Maddison (1989) is perhaps the best known.

`BaTS` uses two established statistics (the Association Index, AI, and Fitch parsimony score, PS) as well as a third statistic (maximum exclusive single-state clade size, MC) introduced by us in the `BaTS` citation, where the merits of each of these are discussed. What sets `BaTS` aside from previous methods, however, is that we incorporate uncertainty arising from phylogenetic error into the analysis through a Bayesian framework. While other many other methods obtain a null distribution for significance testing through tip character randomization, they rely on a single tree upon which phylogeny-trait association is measured for any observed or expected set of tip characters.

To improve on this basic approach we use posterior sets of trees (PSTs), obtained through earlier Bayesian MCMC analysis of the data, that integrate over all likely phylogenies and incorporate the phylogenetic uncertainty arising from the data. Although a Bayesian MCMC analysis is therefore a precondition to using `BaTS`, we do not feel that this is likely to deter potential users since these analyses are increasingly common.

What can it do?

BaTS can estimate phylogeny-trait associations for *n* different states of a single character at a time using the AI, PS and MC statistics, and provide 95% confidence intervals and significance estimation for these. BaTS is also capable of performing batch analyses of large numbers of datasets.

System requirements

Java

BaTS is written and compiled for Java 1.5.0, ("J2SE 5.0"). You will need a computer capable of running this version of Java or higher. For most platforms it is sufficient to download the required version of Java directly from java.sun.com Mac OS X users should note however that on versions 10.4.5 and lower the procedure for upgrading to Java 1.5.0 (from 1.4.2, the default on these systems) differs. They should consult http://www.apple.com/downloads/macosx/apple/macosx_updates/index_abc.html for further instructions. If unsure, typing '`java -version`' from a Terminal session will tell you which version of Java is currently used by the operating system.

Hardware

We have not identified any specific minimum hardware requirements; these in any case scale with the number of taxa in the tree, number of possible states observed and number of null sets used to form the null distribution. Generally speaking, at least 256 Mb of system memory (physical RAM, not virtual memory or swap file cache) should be available for each separate instance of BaTS running. Note that in some cases this may not be sufficient and users will need to increase the amount of memory available to the Java Virtual Machine (JVM) using the `–Xms` command; for more information type '`man java`' from the command-line or see
http://edocs.bea.com/wls/docs70/perform/JVMTuning.html

Installing BaTS

Given the correct JVM (1.5.0 or higher) is available, installation of `BaTS` is simple. The complete `BaTS` package is hosted at evolve.zoo.ox.ac.uk/software for download as an archived jarfile. Simply download the jarfile to some memorable location on your hard drive.

Using BaTS: input file requirements

Preconditions

Because `BaTS` uses the PST from a Bayesian MCMC analysis to integrate over all credible trees to account for error in the phylogenetic signal, users must first use an appropriate package to produce a PST. This is a single treefile containing many trees from the posterior set, weighted by the MCMC sampler so that the most likely topologies are sampled more often. MrBayes (Huelsenbeck & Ronquist, (2001); http://mrbayes.csit.fsu.edu/) and BEAST (Drummond & Rambaut, (2001); Drummond *et al*, (2002); http://beast.bio.ed.ac.uk/) are ideal packages available for this task, and produce these treefiles automatically.

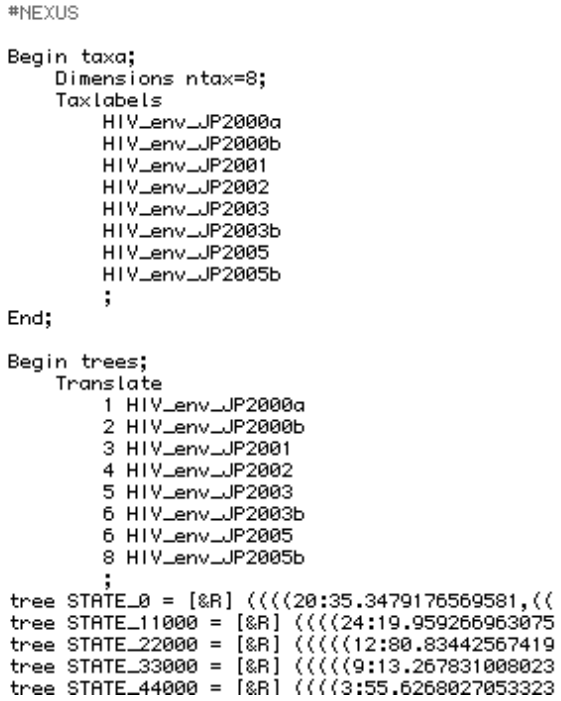
Burnin period

Before they begin to efficiently sample from the posterior likelihood distribution of interest, MCMC samplers typically require an initial period of optimisation during which they arrive in the vicinity of highest-likelihood and tune weighting parameters, etc. During this initial 'burnin' period the posterior likelihood fluctuates wildly; once the lieklihood becomes more stable the MCMC chain can be said to be sampling efficiently. It is common to discard the initial burnin, but currently no burnin is automatically discarded from treefiles in `BaTS`. Users **must** therefore decide on an appropriate burnin period using external data analysis software such as `Tracer` (evolve.zoo.ox.ac.uk/software) and remove these trees from the start of their treefile accordingly.

Format

Input files for `BaTS` follow the popular `NEXUS` file format, with a small modification: Instead of the normal '`taxa`' and '`translate`' blocks, a single '`states`' block is present. The formatting for this is shown below. It is currently necessary to format these by hand; we are working on a graphical interface to parse these input files, please check for updates at evolve.zoo.ox.ac.uk/software

Typical NEXUS format:

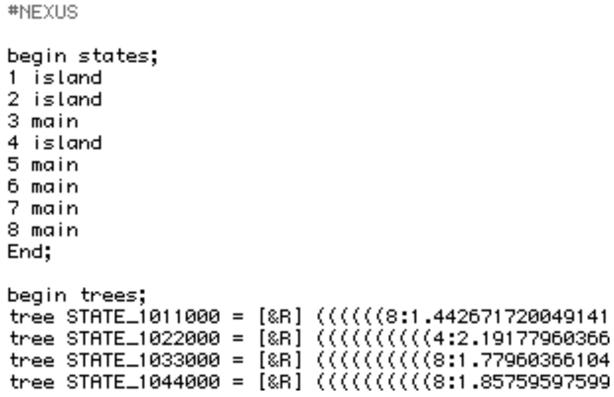


```
#NEXUS

Begin taxa;
    Dimensions ntax=8;
    Taxlabels
        HIV_env_JP2000a
        HIV_env_JP2000b
        HIV_env_JP2001
        HIV_env_JP2002
        HIV_env_JP2003
        HIV_env_JP2003b
        HIV_env_JP2005
        HIV_env_JP2005b
        ;
End;

Begin trees;
    Translate
        1 HIV_env_JP2000a
        2 HIV_env_JP2000b
        3 HIV_env_JP2001
        4 HIV_env_JP2002
        5 HIV_env_JP2003
        6 HIV_env_JP2003b
        6 HIV_env_JP2005
        8 HIV_env_JP2005b
        ;
tree STATE_0 = [&R] ((((20:35.3479176569581,((
tree STATE_11000 = [&R] ((((24:19.959266963075
tree STATE_22000 = [&R] (((((12:80.83442567419
tree STATE_33000 = [&R] (((((9:13.267831008023
tree STATE_44000 = [&R] ((((3:55.6268027053323
```

BaTS format for the same data:



```
#NEXUS

begin states;
1 island
2 island
3 main
4 island
5 main
6 main
7 main
8 main
End;

begin trees;
tree STATE_1011000 = [&R] ((((((8:1.442671720049141
tree STATE_1022000 = [&R] ((((((((((4:2.19177960366
tree STATE_1033000 = [&R] ((((((((((8:1.77960366104
tree STATE_1044000 = [&R] ((((((((((8:1.85759597599
```

The formatting is relatively simple. The key difference in the BaTS-formatted example is that the numbered taxon labels found in the trees no longer correspond, through the ‘translate’ block, to individual taxon names. Instead, through the ‘begin states’ block, each taxon is assigned a character state. Here for instance two states are present, ‘island’ and ‘main’. The state coding used, number of states, and their biological nature are all irrelevant to BaTS; furthermore, all states are weighted to be equally different to each other in state reconstruction.

Note that the ‘begin states’ and ‘begin trees’ statements are case-sensitive and that no whitespace characters (spaces or tabs) appear at the start of any line within the ‘begin states’ block.

Using BaTS: running an analysis

Versions

Two modes of operation are available: the `single BaTS` estimates significance values for a single dataset and also provides 95% CIs as well as *p*-values; while the `batch BaTS` batch-analyses datafiles a directory at a time and only provides a summary set of *p*-values. This version is useful for analysing a very large number of datasets, for example those derived by simulation.

It is also planned to include a `DetailedSingleBaTS` in an imminent future release; this will report the entire posterior observed distribution and composite posterior expected distribution. Users requiring this functionality before then should contact the author directly.

Usage: Single BaTS

To use `BaTS` to analyse a single treefile in detail from the command-line, type:

```
java -jar BaTS_beta.jar single <treefile_name> <reps> <states>
```

where
`<treefile_name>` is the name and full location of the treefile to be analysed,
`<reps>` is the number (an integer > 1, typically 100 at least) of state randomizations to perform to yield a null distribution, and
`<states>` is the number of different states seen.

Usage: Batch BaTS

To use `BaTS` to analyse summary statistics of a whole directory of files from the command-line, type:

```
java -jar BaTS_beta.jar batch <dataset_dir> <reps> <max_states>
```

where
`<dataset_dir>` is the name and full location of the directory of treefiles to be analysed (`BaTS`) will attempt to analyse all files with the extension `.trees` in that directory,
`<reps>` is the number (an integer > 1, typically 100 at least) of state randomizations to perform to yeild a null distribution, and
`<max_states>` is the maximum number of different states seen in any one treefile.

Important: The order in which MC sizes are reported will be the same as the order in which they are introduced in the treefiles. This means that if treefiles in a batch analysis have states introduced in a different order, or different numbers of states, great care must be taken to match the input treefile with the output and compare the order in which states are introduced. Unfortunately this must be done by hand at present, until a future release.

Examples:

Use these examples to check you have installed `BaTS` correctly. From the command-line, navigate to the directory where the `BaTS` folder is located.
We are going to analyse a single treefile modelled on viral data sampled from different compartments of the central nervous system. The 'trait' in question is tissue type; we want to know whether viruses from the same tissue tend to be more closely related. Use a text editor to look at the '`example.trees`' file in the top directory. You will see that in the 'states' block there are seven separate tissues: the frontal, occipital and temporal lobes, the meninges and spinal cord, and also some samples from the seminal vesicles and lymph nodes (The treefile has been truncated to just 30 trees to speed up analysis and troubleshooting).

Let's run a `single` analysis: We'll only use 10 replicates to construct the null distribution as this is a test, and there are seven states present. So from the command-line, type:

```
java -jar BaTS_beta.jar single example.trees 10 7
```

The program should run and execute with a table of results and a `'Done'` message. Look at the output – there is a row for each statistic analysed with column headings for the observed and expected (null) estimates of the statistic. The column headings are the mean and confidence interval information for the observed and expected sets of trees respectively, followed by the 'significance' *P* of the result.
The row titles are the AI and PS statistics for the whole tree, followed by a separate MC size statistic for each character trait value. These are listed in the order in which they are introduced in the treefile, so in this case:

This trait…	…labelled in the treefile as this…	…appears in the screen output as this…	…and has this observed mean estimate:
Frontal lobe	frontalLobe	`MC (state 0)`	`12.63333321`
Occipital lobe	occipitalLobe	`MC (state 1)`	`19`
Meninges	meninges	`MC (state 2)`	`12.66666698`
Lymph nodes	lymphNodes	`MC (state 3)`	`8.566666603`
Temporal lobe	temporalLobe	`MC (state 4)`	`11`
Seminal vesicles	seminalVesicles	`MC (state 5)`	`3.433333397`
Spinal cord	spinalCord	`MC (state 6)`	`5.066666603`

Suppose we had a large number of files that we wanted to analyse at once – they might be data from a large number of patients, or the output from a simulation experiment. In this case we might not be as interested in the mean and confidence interval information from individual data sets, but want to quickly compare significance values across the whole data set. In this case a `batch` analysis might me more useful. Look at the `/batch_data` directory inside the `BaTS` folder. This contains a set of dummy simulation data of a binary ('black' / 'white') character trait. We can analyse all the trees in this folder at once from the command line with the following command (notice the path to the directory is given as './directory/') :

```
java -jar BaTS_beta.jar batch ./batch_data/ 10 2
```

The program will run and execute with a 'done' message. The results are also written to a summary log file as well as the screen, in case a very large number of files are being analysed. The output in a batch analysis is transposed compared to that of a single analysis: there is a set of three columns for the observed and expected means and the significance of each statistic, while each file occupies a single row of the table (the first row is blank).

Note that the order in which the files are analysed and appear in the results table is the same as the order in which they are listed by the filesystem – this order is reported in the screen output for reference. For instance, although an intuitive (numerical) ordering of the files we just analysed might be:

```
simulation1.trees
simulation2.trees
simulation10.trees
simulation11.trees
simulation100.trees
```

the Mac OS/Darwin filesystem will in fact list, analyse and report them in strict alphanumerical ordering, e.g.:

```
simulation1.trees
simulation10.trees
simulation100.trees
simulation11.trees
simulation2.trees
```

Using BaTS: interpreting analyses

The test statistics

BaTS currently includes implementation of three test statistics used to quantify the strength of phylogeny-trait association. The character states of internal nodes must be known for each statistic, and this is achieved through a Fitch (1971b) parsimony reconstruction given all tip states are known. The statistics are the parsimony score ('PS') statistic of Slatkin & Maddison (1989), association index ('AI') of Wang *et al* (2001) and a new measure introduced by Parker *et al* (2007, in press), the maximum monophyletic clade ('MC') size. For a full discussion of the statistics, see Parker *et al* (2007, in press.)

Release note: *An imminent future release will also include the Phylogenetic Diversity ('PD') statistic of Hudson et al (1992), the Nearest Taxa and Net Relatedness indices ('NTI' & 'NRI') of Webb et al (2000; 2002) and the Unique Fraction ('UniFrac') index of Lozupone et al (2005.) These indices also include branch length information as well as tree topology, hence weighting related clades by the strength of their relatedness.*

The null hypothesis

The null hypothesis under test is one of random phylogeny-trait association; that is, that

> *"No single tip bearing a given character trait is any more likely to share that trait with adjoining taxa than we would expect due to chance"*

As implemented in BaTS, each statistic is scored on the PST and a null distribution generated against which the true posterior statistic distribution is compared. The p-value reported is the proportion of trees from the null distribution equal to, or more extreme than, the median posterior estimate of the statistic from the PST. By convention therefore, we reject the null hypothesis at the desired level of significance α where $p \leq \alpha$, e.g. $p \leq 0.05$ for a significance level of 0.05. We leave it to the user to decide what level of significance is appropriate. Computationally, the p-value is stored, manipulated and printed to output as a Java `float`, a 32-bit floating-point number of variable precision. Because of the way Java handles and rounds floating-point numbers, it is possible that, for instance, a number reported as:

```
0.0500
```

might actually represent the number:

```
0.0500953…
```

in the analysis. For this reason we advise that users accept decimal numbers to at least 3 decimal places; preferably more.

Output

The `BaTS` output varies depending on whether a Single or Batch analysis has been carried out.

Single `BaTS` analyses output a table of information with rows corresponding to posterior estimates of observed and expected values for the PS, AI and MC metrics respectively. Regardless of the number of character states (trait values) only one row of information is presented for the PS and AI metric, but there will be as many MC metric rows as trait values. The MC metrics appear in the order in which they occur in the input .trees file. Columns in the table correspond to the mean, median, and upper and lower 95% HPD intervals of the observed values of the metric, followed by those of the expected (null set), then a p-value. This is the proportion of trees in the observed set equal to, or more extreme than, the median value of the expected (null) set.

Batch `BaTS` analyses are not conducted any differently to single analyses. However they are intended to support large-scale analysis, such as simulation or automated sequence analysis, where the investigator is more interested in the behaviour of a set of metadata than individual datasets. To this end, in order reduce computation time where possible and simplify output data, only p-values are recorded. The output takes the form of a table with one row per dataset where columns give the p-values for PS, AI and MC*{0..n}* statistics respectively. If you intend to carry out a large scale `Batch BaTS` analysis it is recommended that you first carry out a number of trial `Single BaTS` analyses of a random selection of the input datasets to check parsing of the input files is correct and that posterior observed and expected values are approximately in line with those predicted by other methods or *a priori* expectation.

Note that the expected (null) distributions generated are a function of the PST and terminal taxon trait values in combination; even if the same taxa are used the null distributions `BaTS` generates are not valid for PSTs generated under a different tree model, nor for different tip labels. If you reanalyse your data under a different model, or change your tip labelling scheme, you *must* re-analyse the data.

Using BaTS: caveats and warnings

Computational constraints:

Users should be aware that although `BaTS` has been rigorously tested on our development machines, this is the first public release of the package.
As such, the real-world performance of the core packages (which in any case are highly dependent on hardware architecture, and software architecture to a degree) is unknown at this point; in fact, we would appreciate any feedback concerning performance.

As a guide, a `GeneralizedSingleBaTS` of 32 taxa with a binary (2-value) trait on a PST of 10,000 trees with expected distributions assembled from 100 replicates per tree typically takes approximately 5 minutes on a 1.25 GHz Apple PowerPC machine under Java 1.5.0 / Mac OS 10.4.10.

As usual, physical factors increasing compute time performance will include:

- Slower system CPUs
- Slower system bus speeds
- Slower system RAM access (in particular, because `BaTS` uses a lot of memory at present heavy reliance on virtual memory coupled with a slow hard drive is likely to adversely affect performance

Problem parameters that will slow the analysis include:

- Number of replicates to construct expected posterior distribution. We have not specified a default value, since we prefer that users take responsibility for this key parameter; however we have not noticed significantly better power or Type I performance when the number of replicates increases above 100 and typically use this value. Increasing the number of replicate sets dramatically increases both compute time and memory use. We're not sure by how much (depends on platform) but it is expected to be a linear increase at best (e.g. 1000 replicates will take at least 10 times as long, and use 10 times as much memory as 100 replicates)
- Number of taxa. Increasing numbers of taxa will increase compute time and memory usage.
- Increasing sizes of PST. Longer tree sets linearly increase compute time, though memory usage increases by a lesser amount since most of the memory allocated to each tree is re-used for subsequent ones. We have often found it useful to downsample PSTs obtained from long MCMC chains (e.g. to 1000 trees from 90,000 states following a 10,000 state discard as burnin from an original 100,000 state chain); this should be done at regular (not random) intervals.

- Large numbers of different trait values may slow performance (e.g. an analysis with 20 rather than 2 trait values may suffer) because more tree traversals are required. We have not collected substantial data on this effect though, and would particularly welcome feedback.

Biological constraints

We have developed this method to analyse multi-state data on a single character. While there is no computational reason why large numbers of states cannot be analysed, there seems little point in, say, analysing the phylogeny-trait association of a character with 20 discrete states on a 25-taxon tree; any association seen may be as due to sampling error as genuine data signal. Users must use their judgement as to whether `BaTS` is an appropriate way to analyse their data, but are encouraged to contact the author for help.

The MCMC requirements given at the start of this documentation must be obeyed, since the analysis depends on an accurately estimated PST. In particular, users must have confidence in the MCMC, which should have reached stationarity and have no limits, priors or transformations on tree topology or branch lengths, except where there is a good *a priori* reason to apply them.

Lastly, the method uses unweighted parsimony reconstruction. This assumes that transitions between states are all equally likely, totally reversible, and independent of branch length. If your data does not obey these criteria then unfortunately a `BaTS` analysis should not be performed (though future versions of `BaTS` using the PD, NTI/NRI and UniFrac indices will include branch length information.)

FAQ

Note: this is just a preliminary list of FAQs; for any persistent problems, or if you have any other comments, please contact the author.

Why don't my analyses run?
Try running the sample datasets included in this package. If the sample files don't run either you may well have the wrong version of java installed; check and install the correct version. If they run but your data does not, it is likely you have parsed the input files incorrectly. Refer to the 'Input File Requirements' section of this documentation. If problems persist, contact the author.

Why do I get an error message saying I've run out of memory?
Typically this will manifest itself with a message such as
"`Exception in thread 'main': java.lang.OutOfMemoryError: java heap space`"
This error arises then the JVM doesn't have enough system memory ('RAM') available to hold all the data it needs to. Unfortunately invreasing your computer's virtual memory allocation will not solve this problem. You can try increasing the default amount of RAM allocated to the JVM with the '`-Xms`' command (see the 'System Requirements > Hardware' section of this documentation for details.)

Why do I get an error message saying `(ArrayList casting error)`*?*
This is because you are using an earlier version of Java (below 1.5.0) that does not support the way this software handles `java.util.ArrayList`s. Check you have the correct version of Java installed.

Can I run BaTS on a bootstrapped set of trees?
No. This package and the discussion of statistics evaluated with it (Parker *et al*, 2007) is designed to be used in a Bayesian MCMC context. You can evaluate any set of trees you like as long as they are parsed properly, be they a bootstrapped set or some deliberately chosen set: but if you do, all assumptions about the power and behaviour of the statistics are invalid.

Which package should I use to produce a PST first?
Both `MrBayes` and `BEAST` produce acceptable output PSTs. We are unaware of any others.

Contact

The author of this documentation and software is Joe Parker. You can contact him at:

Joe Parker
Viral Evolution Group
Department of Zoology, University of Oxford
South Parks Road
OX1 3PS
United Kingdom

Or by email: joe.parker@zoo.ox.ac.uk

References

Drummond, A.J. & Rambaut, A. (2003) BEAST v1.0, Available at http://evolve.zoo.ox.ac.uk/beast/

Drummond, A.J., Nicholls, G.K., Rodrigo, A.G. & Solomon, W. (2002). Estimating mutation parameters, population history and genealogy simultaneously from temporally spaced sequence data. *Genetics* **161**: 1307-1320.

Fitch, W.M. (1971b). Toward defining the course of evolution: Minimal change for a specific tree topology. *Syst. Zool.* **20**: 406-416.

Hudson, R.R., Boos, D.D. & Kaplan, N.L. (1992). A statistical test for detecting geographic subdivision. *Molecular Biology and Evolution* **9**(1):138-151.

Huelsenbeck, J.P., & Ronquist., F. (2001). MRBAYES: Bayesian inference of phylogeny. *Bioinformatics* **17**:754-755.

Lozupone, C. & Knight, R. (2005) UniFrac: A new method for comparing microbial communities. *App. & Environ. Microbiol.* **71**(12):8228-8235.

Parker *et al* (2007). Correlating viral phenotypes with phylogeny: Accounting for phylogenetic uncertainty. *Infect. Genet. Evol.* In press.

Slatkin, M., & Maddison, W.P. (1989). A cladistic measure of gene flow measured from the phylogenies of alleles. *Genetics* **123**(3):603-613.

Wang, T.H., Donaldson, Y.K., Brettle, R.P., Bell, J.E. & Simmonds, P. (2001). Identification of shared populations of Human immunodeficiency Virus Type 1 infecting microglia and tissue macrophages outside the central nervous system. *J. Virol.* **75** (23): 11686-11699.

Webb, C.O. (2000) Exploring the phylogenetic structure of ecological communities: an example for rain forest trees. *Am. Nat.* **156**(2):145-155

Webb, C.O., Ackerly, D.D, McPeek, M.A. & Donoghue, M.J. (2002) Phylogenies and community ecology. Annu*. Rev. Ecol. Syst.* **33**:475-505