Learning to perform computational experiments in phylogenetics, and then write them up

Tandy Warnow

1 Overview

The goal of my lab is to develop new algorithms for phylogeny estimation, and related bioinformatics steps, that have exceptional accuracy and are fast enough to analyze large datasets. Doing experimental studies, both on simulated and biological data, is a key part of this research program! Experiments are fun and they lead to real insights, and these insights will enable you to design new methods that are better than previous ones and perhaps even prove some mathematical properties about methods or models to explain what you see. So you should use experiments to explore your ideas, to understand which methods perform well and under what circumstances, and use what you think to propel you to future discoveries.

The rest of this document is help train you so that you can be a careful and rigorous researcher, and then to write about your research. Towards these goals,

- don't try to write your own code for basic steps in analyses that are already available in published vetted codes (that includes computing error rates, computing various statistics on datasets, computing distances, etc.)
- always document everything you do, in sufficient detail to reproduce the experiment and interpret it
- always use the best available methods for each analysis (which means you have to learn what those are)
- don't draw conclusions too quickly (because differences between methods depend on the model condition and all sorts of details), but do hypothesize about trends from data
- ask for help if you get stuck, and don't change the plan or protocol without checking with me (because it might be easy to fix, or you might just be confused about something)
- do try to write well; see Section 3 for tips, but expect this to take awhile to learn, and don't get discouraged

Section 2 is about learning to do experiments and Section 3 is about writing; these are the biggest parts of this paper. Section 4 has some additional comments.

2 Learning to do experiments

2.1 Level 1: Execute a plan given to you with a fair bit of detail

In this first assignment, you are given a plan that is provided in fair detail, and you need to execute it. Even so, it's not trivial.

Canonical example: Take all the replicates from the 1000L1 dataset, and compute neighbor joining trees (NJ) from PAUP*, using p-distances, logdet distances, and at least one other distance in PAUP*. Use the Potentially Inferrible Model Tree (PIMT, in rose.tt) as the reference tree. Report average FN tree error rates using Erin's code and put the results in a table with a clearly defined table caption. Discuss what you observe (e.g., which method performed better, and is this what you expected to see?). Make sure everything is reproducible. Write this all up in latex.

What you need to learn: To do this, you'll need to learn many things:

- where to get the 1000L1 datasets
- how to run NJ with those distances within PAUP*
- how to find Erin's code and then use it to compute FN tree error rates
- how to write a paper in latex, and in particular to make a table

All these tasks are fairly straightforward, but they aren't trivial (and you need to spend some time to be able to do them).

This was just an example of a basic assignment, and many others exist. Try to do exactly what you were asked to do, and don't make substitutions. For example, don't use some other software than PAUP* to compute NJ trees, and don't use some other software than Erin's code to compute tree error rates. Don't modify any code you are given (e.g., don't modify Erin's scripts to compute errors). If you have problems, make sure to ask for help, and ask for help early. Sometimes the assignment, as stated, has a problem, and it will need to be changed; but often there's no problem, and it's just a question of understanding how to use the code.

2.2 Level 2: Execute a simple task, given at a higher-level

The previous assignment was very straightforward, but now imagine the assignment is not quite as specifically defined.

Canonical example: Take all the replicates from the 1000L1 dataset, and compute ML (maximum likelihood) trees and MP (maximum parsimony) trees using some software packages (your choice). Report average FN tree error rates using Erin's code and put the results in a table with a clearly defined table caption. Discuss what you observe (e.g., which method performed better, and is this what you expected to see?). Make sure everything is reproducible. Write this all up in latex.

What you need to learn: For this assignment, you'll need to learn many things:

- where to get the 1000L1 datasets
- what ML code and what MP code to use, and how
- how to use Erin's code to compute FN tree error rates
- how to make a table in latex

The new things here are what ML code and what MP code to use, and how to run the codes to get good results. You need to learn what codes work well (i.e., give good results for their optimization problems, and are not too expensive to use), and this will take some effort. You probably need to ask people, or read papers (recent ones) and get a sense for this. There are many methods for all both problems, so you need to be careful.

For ML, there are many well known codes, including Phylip, PAUP*, RAxML, IQTree, PhyML, GARLI, and FastTree. Which one will you use and why? (What are the pros and cons?) Once you pick the software package, then you need to figure out the exact commands to get good results, and you need to report how you run it. For example, if you picked FastTree, it can be compiled for single or double precision; which one did you do? For ML, what model of sequence evolution will you pick, what search strategy, what stopping criterion? Why do you make these decisions? What do you think will happen if you make different decisions?

The same thing is true for MP, except that you don't need to specify a sequence evolution model, and you will need to specify how you select a tree from the set of possibly many different optimal trees. Make intelligent choices, and specify everything.

2.3 Level 3: Designing an experiment to answer a question within a limited framework

The third level is where it gets more interesting: designing your own experiment to answer a question. Several projects in this category could be published, but of course you need to do your work well (both in choices of methods and how you write.) Here are some examples of experiments of this type.

- Figure out the best way to compute a tree (using standard methods, but you can vary how you run the analysis) for the 1000M1 datasets.
- Figure out the best way to compute a tree (using standard methods, but you can vary how you run the analysis) on the 1000M5 datasets
- Figure out how changing the rate of evolution (slow rates to high rates) impacts tree estimation (using the best methods you found), using 1000M1 and 1000M5 as the two model conditions.
- Sometimes using logdet distances produces better results for NJ than using p-distances, and sometimes it's the reverse. Look at the 1000M1 and 1000M5 model conditions, and report what you see. Do the same thing for BME in FastME.
- Sometimes using ML under the Jukes-Cantor model produces better results for than using ML under the GTR model, and sometimes it's the reverse. Look at the 1000M1 and 1000M5 model conditions, and report what you see.
- Sometimes FastTree is the same as RAxML and sometimes it's worse. See if you can find a model condition in the SATé-1 1000-taxon model conditions where FastTree is worse. What are the characteristics of those datasets?
- How does alignment error impact ML and distance-based tree estimation? Use the 1000M1 and 1000M5 model conditions to explore this.
- Can you find biological datasets for which alignment estimation changes the ML tree in significant ways? (Here you have to define what it means to be "significant" differences, but the use of branch support, perhaps computed using bootstrapping, can help.) What are the properties of those datasets?
- Can you find biological datasets for which RAxML and FastTree produce significantly different trees? (Here you have to define what it means to be "significant" differences, but the use of branch support, perhaps computed using bootstrapping, can help.) What are the properties of those datasets?
- Can you find biological datasets for which FastTree and NJ produce significantly different trees? (Here you have to define what it means to be "significant" differences, but the use of branch support, perhaps computed using bootstrapping, can help.) What are the properties of those datasets?
- Can you find simulation conditions for which some pair of "good" tree estimation protocols produce significantly different trees? (Here you have to define what it means to be "significant" differences, but the use of branch support, perhaps computed using bootstrapping, can help.) What are the properties of those datasets? By a tree estimation "protocol" I mean a

two-phase method that first aligns the sequences and then computes a tree on the alignment, and by "good" I mean a protocol that seems to provide good accuracy in general.

- Try to simulate sequences under a complex sequence evolution model (e.g., one with heterotachy) and then estimate trees under this model using all the usual suspects, plus maximum parsimony. What do you see?

In these analyses, you have a lot more room to explore methods and how you run methods. The datasets are specified and you can only use existing methods, but you nevertheless have a lot you need to figure out. And you need to design the experiment carefully, or the results won't be that useful.

Here are some things to think about.

- Select data to analyze that are suitable: This is surprisingly important and often overlooked. Don't analyze data that no careful reader will think is worth studying. That means: make sure the data are either already in some prior publication and were considered worth studying, or make sure the data have properties that make them worth studying. In particular, if you are using biological data that hasn't yet been analyzed, be very cautious! More likely you will be using previous data or generating new simulated data. Data that are too easy (all good methods have nearly perfect accuracy) or too difficult (all methods have poor accuracy) aren't that relevant, and don't help distinguish between methods. It will take some effort to generate data that are biologically relevant and neither too easy nor too difficult, so that you can distinguish between methods.

- Select methods and codes. Always use the best current methods to analyze your data. This means, in general, the best alignment methods, the best tree estimation methods, etc. This is important even if you are just learning to explore phylogenetic estimation issues.

- Don't get excited by small distances that aren't important

- Don't jump to conclusions too early (but do hypothesize and think about how to test your hypotheses)

2.4 Level 4: Modify an existing method and test it

When you get to this point, you are ready to do something more creative. Examples here include taking an existing method and testing it in a new context, to modifying the method in some relatively straightforward way and testing it on same context it was designed for. Here is one such study.

- For SVDquartets analyses, the species tree is computed by combining quartet trees, using some existing quartet amalgamation method. You could change the quartet amalgamation method and see what the result is, in terms of accuracy and speed. You could also consider weighting the

quartet trees using the SVDquartets criterion score, somehow, and then use weighted quartet amalgamation methods.

- Modify how the set X of constraint trees is computed in ASTRAL and FastRFS, and see how it impacts accuracy and speed
- Change the distance method used in ASTRID from FastME to something else (other than NJ), and see how it impacts accuracy
- Change the distance matrix computed in ASTRID from being the internode distance matrix to something else, and then run FastME
- Use the divide-and-conquer strategy in TreeMerge to compute gene trees from sequences that evolve under the GTR model

2.5 Level 5: Designing a method from scratch, and testing it.

This level is of course the best and most fun... but also harder to do. Here are some very high level problems to try to work on.

- Develop a new quartet amalgamation method and test it (e.g., replace the quartet amalgamation method used in ASTRAL or SVDquartets with your method)
- Develop a supertree method that can be used with trees that have branch lengths

If you are introducing a new method, you need to give an accessible explanation for the new method that allows the reader to potentially implement it themselves, and interpret the results. Most methods are too complex to be implementable from a high-level description, and a detailed description of such a method will be too difficult to read. So aim to give a fairly good high-level description followed by a pointer to your code (open source) where a more detailed description is also available. Learning how to do this takes time, but make the effort.

You need to test your method well. So do more than just design the method – test it, rigorously, on carefully selected simulated and real data (and not just the data where your method does well!), and compare your method to the best alternative methods. And of course you should prove theoretical guarantees about the method (e.g., statistical consistency under a model, running time, etc.). The point is, don't write a paper that shows off your method's best properties: design the study to really understand the method, even if your study reveals some weakness in your method, as that will be an opportunity for you to improve the method (and it's better that you find this out than someone else does, and they publish the embarassing results!).

3 Writing

To write well, you need to read a lot, and learn the style and content expected for where you want to submit your paper. Read many papers, and read them in their entirety (including supplementary materials). See if they are reproducible. Look at figures and captions. Look at how they cite papers. Learn about the research community so that what you are working on will interest them. And of course, be rigorous, be reproducible, and be clear. The rest of the advice here aims to expand on these points, and to provide some concrete examples. (Please also see `http://tandy.cs.illinois.edu/ten-rules-writing-papers.pdf` for more about writing papers.)

Make your work reproducible. In the previous sections, I talked about the importance of designing a good experiment, following the protocol, and documenting exactly what you did so that it is reproducible and interpretable. This is always important, and so I'm going to discuss it again.

Generally follow the standards in published papers, giving the same level of detail as you find in their methods sections. This will mean: version number and command for each method, where you got the software from, and what the command means. This material would go into a supplementary materials section in your write-up. You also need to either use previously studied data (and say where the data are from) or generate your own data and then make those data available.

Use figures and tables well. Learn how to show results in figures and/or tables. Both should always have informative captions that provide enough information for the reader to interpret the results. For figures with subfigures, the y-axes in subfigures should always be common range. All y-axes should nearly always start at 0.0. Always make sure to label you axes meaningfully.

Don't show results replicate-by-replicate; always show results for the entire set of replicates, whether you report just the mean or you report other statistics (e.g., a box plot) for the distribution. Don't show more digits than are meaningful. For example, 12.3% RF error is fine, or you can write it as 0.123, but writing 12.3512 (or using more digits) is unnecessary.

Make sure to explain the units! For example, in running time analyses, say if the units are seconds, minutes, or hours, etc.. For Robinson-Foulds (RF) tree error, make sure to say if it's a proportion (maximum 1.0), a percentage (maximum 100), or the number of edges in the symmetric difference. Also make sure to define your terms, even for terms that you think are well established, since not all people may use the term identically. (This is true, for example, for RF error rates.) Also, it is better to report error as proportions or percentages, rather than number of edges, so that results can be interpreted without checking for the number of leaves in the tree.

Write up your observations. For each figure or table, write a paragraph (or more) about what the results show. Think about what you see. Say if it's surprising, or if it's what you expected.

Think about what you observe in your experiments, and discuss! You need to think about what your data show, and not just report the results. This means many things, including thinking about whether a difference in performance is worth commenting on. (For example, if method A has tree error rate 12.35% and method B has tree error rate 12.36%, no one will care.) But it also means trying to make connections between results you see and the model conditions. Maybe a method is good when the datasets are small but otherwise it's poor? Or maybe it's good when the maximum p-distance is low and otherwise it's poor? Or maybe it's good on true alignments and poor on estimated alignments? Maybe the model condition you are considering is not biologically realistic? In other words, think about what you are seeing and compare trends in different conditions, and then write up what you are seeing.

In order to gain insight into the interplay of data and methods, you need to explore the data. You can do this in many ways, including computing and reporting basic statistics about datasets. Examples in our work include maximum and mean pairwise p-distances, the gap length distribution for sequence alignments that have indels, the percentage of the true alignment that is gapped, etc. If you are working with species tree estimation, then measurements of gene tree heterogeneity (however you want to report this) are helpful. It's also a good idea to visualize the model trees (i.e., make figures).

Also, always write what you are observing from your analyses; don't just show a figure with results, but also write a paragraph about each figure and summarize the trends in the figure.

Take some care in writing. Good writing takes practice, and has many levels. At the low level, this has to do with spelling, grammar, punctuation, paragraph structure. But at the higher level, it includes describing your ideas well and using technical language correctly. For example, make sure you understand the meaning of the terms you use, including the meaning of "consistency". Communication in writing is extremely important; what you write has to successfully communicate what you did and what you mean to communicate.

Make sure to include a bibliography and cite your references correctly. Don't omit to cite the literature, and that means reading enough to know what to cite.

Assuming you are learning to use latex, make sure the latex compiles–fix the compile errors. Check for spelling mistakes. Print the paper and read it! (And check the bibliography, make sure the figure is readable on a printed page.)

Don't assume that I am your only reader: think of your document as going to a journal and being assigned to a reviewer who will need to understand what you have done.

Structuring your document The actual structure of your paper needs to follow the instructions for the conference or journal, but in essence should have the following elements (usually in this order):

- Introduction: What you are trying to do, and why
- High-level description of the study (in terms of data, methods, measurements)
- Results: what you observed (figures, tables, and text).
- Discussion: high level of what you learned, and how it compares to other papers or what you thought going into the study.
- Appendix: Details that are sufficient to reproduce the analysis, including where the data are.

If you read one of my papers, you'll see that roughly this is the structure, with the details almost always in a supplementary materials document, or in an appendix.

4 Closing comments

You also need to learn enough about phylogeny estimation that you understand what you are reading (i.e., what is a model, what is maximum parsimony, what does it mean to say that something is statistically consistent, what does it mean to say that something is identifiable). Learning enough about phylogeny estimation will also help you avoid making mistakes and correct mistakes when you make them.

Also, learning enough about phylogeny estimation will help you know what the "standard" approaches are, and you'll make sure to use those standard approaches when you do an analysis. Learning enough about phylogeny estimation will help you choose how to do your analyses. For example, when using neighbor joining, how you calculate distances matters (p-distances vs. corrected distances, and if corrected then what formula was used) impacts the theoretical properties of the method, and will influence your choices. You should be careful about how you run your methods, since default settings may or may not give the best results (this is true for the multiple sequence alignment method MAFFT, for example).

On the other hand, deviating from the "standard" approach can lead to many insights. For example, NJ is statistically consistent under the GTR when you use certain distances but not when you use p-distances. Even so, you could run it both ways and see what you get. Similarly, you could run FastTree with double precision or single precision, and see what you get. Or you could run FastTree or RAxML under the Jukes-Cantor model on sequence datasets that were generated under some other model (not Jukes-Cantor) and see what you get. All of these experiments can lead to insights that could surprise you.