Advanced Molecular Evolution using *HyPhy*

Sergei L. K. Pond, Art F. Y. Poon, and Spencer V. Muse

1 Introduction

1.1 What is *HyPhy*?

We're going to take three cracks at answering this question, so just bear with us.

***HyPhy* is a software package.** Its name stands for '**Hy**pothesis testing using **Phy**logenies'. So, the primary function of *HyPhy* is to analyze genetic (DNA or RNA) sequences in a phylogenetic framework. A phylogeny is a hypothesis — it is a tree-shaped (hierarchical) model of how sequences are descended from common ancestors. Because of these evolutionary relationships, sequences are not independent random outcomes of molecular evolution, like rolling dice. Moreover, we are usually more interested in the evolutionary processes that led to the observed sequences than in the end products of evolution themselves. We will go so far as to declare that if you want to test *any* hypothesis that involves genetic sequence data, then you are obliged to account for the phylogeny! Because of this, phylogenies are a core component of *HyPhy*.

***HyPhy* is scientific software.** It was initially developed by Sergei Pond to clean up and extend some programs that were written by Spencer Muse. Because *HyPhy* was designed to be highly customizable, it became useful to simply keep extending *HyPhy* as new research projects arose. *HyPhy* became an open-source project under the General Public License (GPL), allowing anyone to download, use and modify the source code. It is also distributed for free as executable binaries that are compiled for Mac OS X, Windows, and Linux. At the time of writing this book, *HyPhy* has over 6,000 registered users and has been cited in over 500 peer-reviewed scientific publications.

***HyPhy* is as simple or as complex as you need it to be.** It has a rich graphical user interface that can display colourful tables, charts, sequence alignments and trees. You can design an analysis using a point-and-click interface, or choose from a wide selection of popular methods in a Standard Analyses menu. On the other hand, *HyPhy* also has its own scripting language for implementing as complex an analysis as you can imagine. (We'll elaborate on scripting languages in a bit.) *HyPhy* can also be compiled as a shared library that can be called from another programming environment such as Python, Ruby, or R, so that it can act as a component of a bioinformatic pipeline. It can be run in a parallel computing environment where an analysis is broken down into tasks that can be distributed among tens or hundreds of processors and run simultaneously.

This is a lot to absorb in one go, so let's try a goofy analogy. *HyPhy* is sort of like a nice restaurant:

- There are piles of fruit, vegetables and meats in the pantry that nearly all diners will never see. This is the source code.
- There are lots of shiny pots, pans, knives and ovens that turn this raw material into meals. Again, most of this stuff is hidden from the dining room – perhaps by jaunty red vinyl-covered swinging doors. This is the batch language.
- We have a set menu of meals. These meals are on the menu because they're popular and they're tasty. These are the template batch files.
- We have a kids' menu with a selection of those tasty meals accompanied by nice pictures and crayons. This is the GUI.
- We even do deliveries so you never have to drive to the restaurant. This is the *Datamonkey* webserver.

After closing time, we kick back and mess around with the recipes. We tinker. We experiment. Some of it doesn't turn out so well, but that one dish last night? Man.

We want to invite you into the kitchen. We're gonna to teach you how to cook.

1.2 Is this book for me?

This book was written for you. You are a graduate student or post-doc who once mentioned that you wrote a program once and, as a result, you've been asked by your lab principal investigator[1] to run analysis $\mathcal{X}$ on some data. $\mathcal{X}$ is similar to $\mathcal{Y}$, a standard analysis that has been used hundreds of times by other laboratories and is available in dozens of software packages and web applications.

But, owing to the uniqueness of the model system being studied in the lab, or the research interests of the PI, there is a significant difference that means that you can't run $\mathcal{X}$ using software designed to run $\mathcal{Y}$. Or any software that you can find. Just as you've resigned yourself to writing your own #$!&@ program to implement $\mathcal{X}$, one of your colleagues mentions recently seeing a similar analysis in a paper. After downloading and scanning over the paper, you discover two things: (1) the analysis is definitely not $\mathcal{X}$, but (2) they didn't have to write their own software - they just used something called *HyPhy*.

HyPhy enables you to do just about anything in the domain of phylogenetics. This is both a blessing and a curse — to borrow a popular phrase from C programming, it "gives you enough rope to hang yourself with". While there are thousands of biologists that use the standard analyses distributed with *HyPhy* or accessed as web applications on our public computing cluster (*Datamonkey*), there are only a small (but growing!) number of biologists that have mastered the *HyPhy* batch language to the degree that they can write or modify scripts for their own research.

This book is *not* for you if you're not even remotely interested in modifying a template batch file. You can get what you need to get done through *Datamonkey* or one of the items in the standard analysis menu. We've tried our best to make the standard analyses in *HyPhy* accessible and self-explanatory. We even have several book chapters and a rudimentary software manual that do a pretty good job of telling you what you need to know to get things done.

[1] We know that you're not the principal investigator, because no PI would have the time to be reading this right now.

1.3 How do I get *HyPhy*?

HyPhy is free and open-sourced under the General Public License (GPL). Just direct your web browser to our homepage: `http://www.hyphy.org` and hit the big shiny 'download' button. You can either download the source code and compile it yourself, or download one of our pre-compiled binaries for Mac OS X, Linux, or Windows.

2 The *HyPhy* Batch Language

A batch language is a collection of function calls to one or more applications. Since we often want to run a sequence of function calls many times, or run different sequences depending on the outcomes of previous calls, batch languages typically include syntax for iteration (loops) and conditioning.

The *HyPhy* batch language (HBL) is an application-specific language for modeling molecular evolution. Most programming languages share a common set of objects such as integers and arrays. Because *HyPhy* is a language for modeling molecular evolution, it also includes objects that represent multiple sequence alignments (DataSet), phylogenetic trees (Tree), and substitution rate models (Model). These domain-specific objects are so important and elaborate that we will dedicate a chapter for each of them. For the time being, we're just going to get acquainted with the conventions and syntax of the HBL.

2.1 Basics: Input and Output

The HBL is modeled after C. Specifically, statements are delimited with a semi-colon, and blocks of code are enclosed in curly braces.

2.2 Types

2.3 Operators

2.4 Expressions

2.5 Control Flow

2.6 Functions

2.7 Keywords