IgPhyML

Version 0.7 201609.22
GNU General Public License V3

Kenneth B. Hoehn
Contact: kenneth.hoehn@oriel.ox.ac.uk
&
Gerton Lunter
Oliver G. Pybus

January 8, 2017

Based on codonPhyML
Marcelo Serrano Zanetti (www.marceloserranozanetti.org), Stefan Zoller,
Manuel Gil, Louis Du Plessis and Maria Anisimova
and PhyML
Stphane Guindon guindon@stat.auckland.ac.nz

Contents

1 Download and Installation

IgPhyML is available for download at GitHub: https://github.com/kbhoehn/IgPhyML

In Linux, installation is performed exactly as in codonPhyML, with the exception that the folders and executables created will be called igphyml instead of codonphyml. It is strongly recommended that you do installation with OpenMP and BLAS/LAPACK using ./make_phyml_blas_omp. In Ubuntu Linux, you will likely need to install the package libatlas-base-dev in order to get the appropriate dependencies (apt-get install libatlas-base-dev). This will usually speed up analysis time considerably on multicore machines. Once compiled, add the src directory to your PATH variable and things should work from there. If BLAS/LAPACK isn't available you can still take advantage of multicore machines using ./make_phyml_omp. FYI: You'll need to re-compile the program if you move the installation directory after compilation.

Installation on Mac OS X is trickier, but possible. The primary issue is gaining OpenMP support, and installing some GNU command line tools. The best way is to just install the latest version of llvm available through homebrew, as well as autoconf and automake. To do these youll need to:

1. Install homebrew (http://brew.sh/index.html). If it's already installed be sure it's at the latest version (brew update). You may need to install Xcode as well.

2. Install autoconf, automake, and llvm:
   brew install autoconf
   brew install automake
   brew install llvm

3. Specify the llvm version of clang in Makefile.am and src/Makefile.am by adding the line CC=<path to llvm clang> to the beginning of both files. You will also need to add MACOMP=<path to omp.h> and MACLLVM=<path to llvm lib> to src/Makefile.am. For instance, if you've install llvm 3.9.1 via homebrew, you will likely need to add the line:
   CC=/usr/local/Cellar/llvm/3.9.1/bin/clang
   to Makefile.am.
   and the lines
   CC=/usr/local/Cellar/llvm/3.9.1/bin/clang
   MACOMP=/usr/local/Cellar/llvm/3.9.1/lib/clang/3.9.1/include/omp.h
   MACLLVM=/usr/local/Cellar/llvm/3.9.1/lib
   to src/Makefile.am.
   Your specific path may look different, but you can check locations of these files and folders by looking around in /usr/local/Cellar/llvm/. The directory structure should be similar.3

4. Run ./make_blas_phyml_omp, or other versions, as desired, and add the src folder to your PATH variable.

2 Program Usage

The basic operation, which will estimate ω, kappa, $h^{WR\underline{C}/\underline{G}YW}$, branch lengths, equilibrium frequencies under a fixed tree topology with a symmetric WR$\underline{C}$/$\underline{G}$YW motif model, is:

```
igphyml -i <input.phy> -m HLP16 --root <root_id> -u <tree> -o lr
```

The input file must be in Phylip format (see `seqret` in EMBOSS if you want to easily convert your sequences to Phylip format). Because the model is non-reversible it is important to specify the correct root sequence id using `--root`. **Note that this "root" sequence is an actual, direct ancestor of the entire lineage, not an extant outgroup sequence.** For antibody sequences this sequence is the rearranged, un-mutated germline ancestor.

Tree topology should be estimated first using the M0/GY94 (Goldman and Yang 1994; Yang and Bielawski 2000) model, and then fit the HLP16 model to the data set after fixing the topology.

```
igphyml -i <input.phy> -m GY -w M0 -t e --run_id gy94
```

```
igphyml -i <input.phy> -m HLP16 --root <root_id> -o lr -u <input.phy>_igphyml_tree.txt_gy94
```

To do hypothesis testing, you can also fix h using the `--hotness` option. For instance, to constrain h to zero:

```
igphyml -i <input.phy> -m HLP16 --root <root_id> -o lr -u <input.phy>_igphyml_tree.txt_gy94
--hotness 0
```

While constraining $h = 0$ is useful for hypothesis testing the GY94 model, any value of $h > -1$ may be specified. This may be useful for applications such as creating profile likelihood curves.

Important for hypothesis testing: DO NOT compare likelihood values between models fitted with `-m HLP16` and `-m GY`, or for that matter any other GY94 implementation, for hypothesis testing. These models are not nested because HLP16 uses a given ancestor sequence whereas `-m GY` (and most other GY94 implementations) uses codon frequencies at the root. Only use `-m HLP16 --hotness 0` for hypothesis testing under the GY94/M0 equivalent substitution model.

3 Specifying motif models

While the default motif model is symmetric WR$\underline{C}$/$\underline{G}$YW motifs, `IgPhyML` may specify a much more diverse set of motif models using the `--motifs` and `--hotness` options. Currently `IgPhyML` supports models with WR$\underline{C}$, $\underline{G}$YW, W$\underline{A}$, $\underline{T}$W, SY$\underline{C}$, and $\underline{G}$RS motifs. Motifs are specified by their name, mutable position (underlined character), and index in the array of h values specified using the `--hotness` option, using the form:

```
<motif>_<mutable site>:<index of h>
```

The default symmetric WR$\underline{C}$/$\underline{G}$YW model is equivalent to adding the options:

```
--motifs WRC_2:0,GYW_0:0 --hotness e
```

The asymmetric WR$\underline{C}$/$\underline{G}$YW model is specified by adding an additional h parameter to `--hotness` and specifying that new parameter is for $\underline{G}$YW by using `GYW_0:1` in `--motifs`:

```
--motifs WRC_2:0,GYW_0:1 --hotness e,e
```

Each h parameter can be fixed as well. To set $h^{\underline{G}YW}$ to 0, for instance:

```
--motifs WRC_2:0,GYW_0:1 --hotness e,0
```

which is equivalent to:

```
--motifs WRC_2:0 --hotness e
```

More complex models using W$\underline{A}$, $\underline{T}$W, SY$\underline{C}$, and $\underline{G}$RS motifs may be specified using similar rules. For instance, the "Free coldspots and hotspots" model, in which each motif and its reverse complement have separate h values, can be specified by:

```
--motifs WRC_2:0,GYW_0:1,WA_1:2,TW_0:3,SYC_2:4,GRS_0:5 --hotness e,e,e,e,e,e
```

A model in which trimer motifs are symmetric but dimers are not can be specified by:

```
--motifs WRC_2:0,GYW_0:0,WA_1:1,TW_0:2,SYC_2:3,GRS_0:3 --hotness e,e,e,e
```

4 Partitioning ω

BCRs are divided into known complementary determining (CDR) and framework (FWR) regions, which generally experience different types of selection. Because of this, it may makes sense to estimate a separate value of ω (non-synonymous/synonymous substitution ratio) for CDRs and FWRs. These partitions can be specified using a plain text file and the `--partfile` command line option.

For instance, to estimate two ω's – one for CDRs and one for FWRs for the V segment of CH103 – the text file `part.20.txt` in the `examples` subfolder shows:

```
2 81
FWR:0..10,21..35,45..80
CDR:11..20,36..44
```

The first line shows the number of partitions (2) and the number of sites (81). The next line shows the sites that will be used to estimate the FWR ω. Note that the indexing for these sites begins at 0 and includes the final site number specified. So `0..10` specifies the 11 sites from 0 up to and including 10. Non-consecutive sites are separated by commas. This file is then specified using:

```
--partfile <partition file>
```

In this case:

```
--partfile part.20.txt
```

Note that while it is possible to specify more partitions (such as treating FWRs 1, 2, and 3 separately), it generally isnt recommended as this will leave only a small number of sites to estimate each value of ω. Also, partitioned ω is currently only available under the HLP16 model, so it cannot currently be specified for the GY94 topology search.

5 Other options

Number of threads By default, the OpenMP version will use all available threads. You can alternatively set the maximum number of threads by using:

```
--threads <number of threads to use>
```

Although `IgPhyML` generally preserves the capabilities and command line options of `codonPhyML`, modifying options other than those specified here is not supported and is not recommended.

6 Output

Like in `codonPhyML`, the MLE value of each parameter can be found in the file ending in `<input.phy>_igphyml_stats.txt`. Information about the h parameter is in a tabular format:

```
. Hotspot model h_index optimized?  h_value:
Motif:  WRC_2 0 1 2.91097240
Motif:  GYW_0 0 1 2.91097240
```

Here, the motifs (WR$\underline{\text{C}}$ and $\underline{\text{G}}$YW) use the same h value (`h_index`), and h is optimized (`optimized?`). The number after the underscore in the motif name indicates the position in the motif that experiences increased mutability. The MLE of $h^{WR\underline{C}/\underline{G}YW}$ here is 2.91. If h is set using `--hotness`, the `optimized?` column will be 0, and the `h_value` will be the value specified.

The maximum likelihood tree is printed in `<input.phy>_igphyml_tree.txt`. Note that this tree is printed as an unrooted tree, and, if visualized, should be viewed by re-rooting the tree by the germline sequence and shrinking that branch length to zero.

7 Examples

The file CH103.20.phy is in the examples subfolder, and is 20 randomly sampled sequences from the CH103 broadly neutralizing antibody lineage (Liao et al. 2013), plus the V segment of the germline sequence (V4-59).

Fit GY94 to get tree topology:

```
igphyml -i CH103.20.phy -m GY -w M0 -t e --run_id gy94
```

Then use fixed topology from GY94 fit:

```
igphyml -i CH103.20.phy -m HLP16 --root V4-59 -o lr -u CH103.20.phy_igphyml_tree.txt_gy94
--run_id hlp16
```

Fit HLP16 with $h = 0$ for hypothesis testing:

```
igphyml -i CH103.20.phy -m HLP16 --root V4-59 -o lr -u CH103.20.phy_igphyml_tree.txt_gy94
--hotness 0 --run_id hlp16_0
```

Fit HLP16 under the asymmetric WR$\underline{\text{C}}$/$\underline{\text{G}}$YW motif model:

```
igphyml -i CH103.20.phy -m HLP16 --root V4-59 -o lr -u CH103.20.phy_igphyml_tree.txt_gy94
--motifs WRC_2:0,GYW_0:1 --hotness e,e --run_id hlp16_asym
```

See **Specifying motif models** for example on how to run other motif models.

Fit HLP16 under the asymmetric WR$\underline{\text{C}}$/$\underline{\text{G}}$YW motif model with ω partitioned between CDRs and FWRs:

```
igphyml -i CH103.20.phy -m HLP16 --root V4-59 -o lr -u CH103.20.phy_igphyml_tree.txt_gy94
--motifs WRC_2:0,GYW_0:1 --hotness e,e --partfile part.20.txt --run_id hlp16_asym_part
```

8 References

Goldman N, Yang Z. 1994. A codon-based model of nucleotide substitution for protein-coding DNA sequences. Mol. Biol. Evol. 11:725736.

Liao H-X, Lynch R, Zhou T, Gao F, Alam SM, Boyd SD, Fire AZ, Roskin KM, Schramm CA, Zhang Z, et al. 2013. Co-evolution of a broadly neutralizing HIV-1 antibody and founder virus. Nature 496:469476.

Yang Z, Bielawski JP. 2000. Statistical methods for detecting molecular adaptation. Trends Ecol. Evol. 15:496503.