Pythia

Predicting the Difficulty of Phylogenetic Analyses

Julia Haag
Phylogenetic Analysis

Sequence Data

Phylogenetic Tree
Phylogenetic Analysis

• Fast, but less accurate methods:
  • Maximum Parsimony
  • Neighbor Joining
  • ...

• Slow, but more accurate methods:
  • Maximum Likelihood (e.g. RAxML-NG)
  • Bayesian Methods (e.g. MrBayes)
  • ...

Based on "Compiling" https://xkcd.com/303/
What does difficult mean?

MSA → Tree Inference → Post-Processing → Statistical Tests → Bootstrapping → ...
What does difficult mean?

MSA → Tree Inference → Post-Processing → Statistical Tests → Bootstrapping → …
What does difficult mean?

Difficulty = ruggedness of the tree space

- Few highly similar tree topologies
- Single likelihood peak

- Highly distinct topologies, statistically indistinguishable
- Multiple likelihood peaks
Pythia

The oracle of difficulty
Pythia

- Pythia = Boosted Tree Regressor
- Supervised regression task:
  - predict difficulty from 0.0 (easy) to 1.0 (difficult)
  - ground-truth difficulty as target for training based on 100 ML tree inferences
- Trained on ~12.5k empirical MSAs
  - Mean absolute percentage error 1.7%
Prediction Features

• 10 features:
  • 5 MSA attributes:
    • sites-over-taxa, patterns-over-taxa, patterns-over-sites % gaps, % invariant sites
  • 3 MSA information metrics:
    • Shannon entropy, Bollback multinomial test statistic, Entropy-like pattern metric
  • 2 Parsimony-tree-based features:
    • Infer 100 parsimony trees → average RF-Distance, % unique topologies
Prediction Features: Runtime

![Runtime vs MSA size graph]

Runtime relative to single RAxML-NG tree inference.

MSA size
(\# Taxa x \# Sites)
How to use Pythia

- 3 options:
  - **Command Line Interface**, Python module: [https://github.com/tschuelia/PyPythia](https://github.com/tschuelia/PyPythia)
  - C library: [https://github.com/tschuelia/CPythia](https://github.com/tschuelia/CPythia)

- Phylip or FASTA format
- DNA, Protein, or morphological data
How to use Pythia: example MSA

```
pythia -h
pythia -m examples/example.phy -r path/to/raxml-ng -v -b -shap
```

- Single likelihood peak $\rightarrow$ easy (difficulty = 0.16)
- Runtime:
  - Pythia: $\sim$10 seconds
  - 1 tree inference: $\sim$16 minutes
Shapley Values: example.phy

155.488 = num_patterns/num_taxa

0.176 = avg_rfdist_parsimony

1 = proportion_unique_topos_parsimony

-254912.541 = bollback

44744.306 = pattern_entropy

233.192 = num_sites/num_taxa

0.667 = num_patterns/num_sites

0.493 = entropy

0.327 = proportion_gaps

0.341 = proportion_invariant
How to use Pythia: example MSA

```
pythia -h
pythia -m examples/example.phy -r path/to/raxml-ng -v -b -shap
```

- Single likelihood peak $\rightarrow$ easy (difficulty = 0.16)
- Runtime:
  - Pythia: $\sim$10 seconds
  - 1 tree inference: $\sim$16 minutes
Example: Covid Data

"Phylogenetic Analysis of SARS-CoV-2 Data Is Difficult" (https://doi.org/10.1093/molbev/msaa314)

The predicted difficulty for MSA examples/covid.fasta is: 0.82.

FEATURES:
num_taxa: 4869
num_sites: 28361
num_sites/num_taxa: 5.82
avg_rfdist_parsimony: 0.79
proportion_unique_topos_parsimony: 1.0

Feature computation runtime: 737.182 seconds

~12min ≪ 12 hours
Use and Misuse of Pythia

- Prior to tree inferences
- Choose inference + post-processing setup
- Adjust MSA
- Adaptive Search Heuristic

Difficulty equals number of tree inferences
Summary

- Pythia = difficulty predictor
- Difficulty = ruggedness of the tree space
- Prediction prior to time-intensive tree inference
- Accurate and fast
  - faster than a single ML tree inference
- Paper: https://doi.org/10.1093/molbev/msac254
- Pythia on Github: https://github.com/tschuelia/PyPythia