Partitioning

James Fleming - j.f.fleming@nhm.uio.no

@JamesfvFleming

NHM Uio

What is Partitioning?

- Partitioning is the assigning of different substitution matrices to different portions of the same dataset.
- It increases the computational complexity, but can yield better results.
- Particularly for multiple-gene ‘concatenated’ datasets
  - Useful for taxon studies
- Rather than just giving each gene a unique model, we can sometimes bracket genes into same-model partitions!

Why might we need it?

- In a large multiple gene dataset, giving each gene their own model might end up being computationally exhausting.
- At the same time, we know different genes evolve at different rates
  - One gene might be heavily under selection, another might be incredibly conserved.
  - **Over long scales of time, even, the same gene might require different substitution matrices, and there are ways to create partitions over a pre-defined tree to account for this, but this won't be covered here.**
- **Partitioning balances Computational Efficiency and Biological Reality**
- Partitions can be set manually if you know which models model the genes in your dataset well – otherwise…

Partition Finder

- Partition Finder is a software that treats Partitioning as an optimization problem, using the same logic as maximum likelihood tree finding!

Number of possible partition schemes

$$B_n = \sum_{k=0}^{n} \left\{ {n \atop k} \right\}$$

n = 12: $4.2*10^{6}$
n = 60: $9.8*10^{59}$

B_n is the number of possible partitioning schemes given *n* user-defined data blocks

S_n is the number of *n* user-defined data blocks of possible nonempty subsets

k is the number of subsets in the best scheme from the previous round of searches

Number of possible, non-empty partitions (e.g., subset)

$$S_n = 2^n - 1$$

n = 12: 4,095
n = 60: $1.2*10^{18}$

Lanfear et al. (2012)

Aren’t those numbers a bit large?

- You’re quite right!
- The intent isn’t to perform an exhaustive search of all of them in most cases, but a heuristic one.

Number of possible partition schemes

n = 60: $9.8*10^{59}$

$$P_{n\_\text{greedy}} = 1 + \sum_{k=2}^{n} \binom{k}{2} = 1 + n(n^2 - 1)/6$$

n = 60: 35,991

Number of possible, non-empty partitions (e.g., subset)

n = 60: $1.2*10^{18}$

$$S_{n\_\text{greedy}} = n^2 - n + 1$$

n = 60: 3,541

Performing an Exhaustive Search

- 1. Estimate a phylogenetic tree of sequences
- 2. Select the best-fit substitution model for each possible subset
- 3. Calculate the log likelihood of each partitioning scheme by summing the log likelihoods of the subsets that make up that scheme
- 4. Select a partitioning scheme using information- theoretic metrics such BIC, AIC or AICc.
- Fast up to 12 datablocks. Too slow thereafter.

Performing a Heuristic Search

- 1. Calculate the score of the partitioning scheme with n subsets
- 2. Calculate the score of all partitioning schemes with n-1 subsets (i.e. merge two subsets)
- 3. Select the scheme with the best score
- 4. If new scheme is better iterate procedure

- The Hill-climbing procedure is substantially faster, but doesn‘t guarantee the best scheme.

Running a Partition Finding Search

- Partition Finding is implemented in IQTree, which we'll be using, as part of its model selection package
- You can provide a partition file you have already made using the –p command to run a tree analysis
- Before running a model search on your partitions, you will need to prepare some information about the genes you have, which is called a partition file.
- We'll be focusing on writing a new partition file together today.

What does a partition file look like?

- Partition files follow two commonly used formats, RaxML and Nexus.
- One is more involved than the other! IQTree accepts both.
  - A RaxML formatted file looks like this:
    - <Data Type>, <Partition name> = <Partition location>
    - DNA, part1 = 1-100
      DNA, part2 = 101-384

What does a partition file look like?

- A Nexus formatted file looks like this:

  -
    ```
    #Nexus
    begin sets;
    charset <Partition name> = <Partition location>;
    charpartition mine = <model>:<Partition name>;
    end;
    ```

  -
    ```
    #nexus
    begin sets;
    charset part1 = 1-100;
    charset part2 = 101-384;
    charpartition mine = HKY+G:part1, GTR+I+G:part2;
    end;
    ```
  - To have both Protein and DNA data in a Nexus partition, include a file pointer to separate DNA and Protein files in the Partition Location section

  -
    ```
    charset part1 = dna.phy: 1-100;
    charset part2 = protein.phy: 101-384;
    ```

IQTree Partition Finding Options

- The standard IQTree partition finder starts with the supplied gene partition information, and then merges gene partitions together until the model fit doesn’t increase any further. This option is ModelFinder Partition + Merge
  - -m MFP+MERGE
- It can take a long time, and this next option, which merges partitions but doesn’t consider every model in IQTree and only changes invariable site and gamma variation, you can use the TestMerge option, but that’s only best for preliminary work
  - -m TESTMERGE

IQTree Partition Finding Options: Complex Options

- IQTree also implements relaxed hierarchical clustering in its partition finding analysis.
- This is far more thorough than the other options, but a lot more computationally intensive.
- –rcluster is used as an option along with a percentage number, such as
  - -rcluster 10
- This number is the maximum size of similar “datablocks” allowed by the algorithm, meaning that it will only put up to 10% of your predefined blocks into any one model partition.

Summary

- Partitioning data is a way to ensure that the evolutionary pressures acting on different genes can be modelled.
- Partitions apply different models to blocks of your dataset
- IQTree can implement a partition finding system where it will assign models to these blocks based on its own model finding technologies