swsc: A sitewise UCE partitioner

Ryan A. Hagenson

Omaha’s Henry Doorly Zoo and Aquarium

Summary

Ultraconserved elements (UCEs) are regions of the genome that retain partial identity across a vast number of species. This identity retention makes UCEs especially useful for inferring otherwise intractable phylogenies. UCE partitioning acts to split a UCE into three parts: variable left flank, conserved core, and variable right flank. The heightened variation found in the flanks allows for phylogenetic inferences [Baca, Alexander, Gustafson, & Short, 2017; Blaimer, Lloyd, Guillory, & Brady, 2016; Crawford et al., 2012; Faircloth et al., 2012; Faircloth, Sorenson, Santini, & Alfaro, 2013; Harrington et al., 2016; McCormack et al., 2012; Moyle et al., 2016; Smith, Harvey, Faircloth, Glenn, & Brumfield, 2014].

Based on a method originally described by [Tagliacollo & Lanfear, 2018] as Sliding-Window Site Characteristics (SWSC), swsc partitions UCEs based on chosen sitewise metrics such as Shannon’s entropy or GC percentage. Input is either a modified Nexus file or standard FASTA+CSV files, containing the concatenated UCE sequences of individuals under analysis along with the range of each UCE in the concatenation (see example-data/ in code repository for example formats). Output is a single CSV containing the UCE partitions. Optionally, a configuration file for PartitionFinder2 can be produced.

The original method by [Tagliacollo & Lanfear, 2018] used a sequential, brute-force approach, considering all potential core windows from the provided minimum window size up to 1/3 of each UCE’s length. This is inefficient when small minimum window sizes combined with either many UCEs or large UCEs. The method herein uses a candidate windows plus extension procedure.

Overview of swsc’s candidate window plus extension procedure:

1. Generate candidate windows of size --minWin across the UCE (both from the start forward and the end backwards)
   - Example: a UCE of length 120 and --minWin of 50 has forward windows of 1 – 50 and 51 – 100, as well as backwards windows of 81 – 120 and 31 – 80.
2. Find the best C candidates
   - Fitness is determined by minimum sum of square errors of sitewise metrics, minimum variance of left flank, core, right flank lengths, and user preference for --largeCore or not, in that order.
   - Minimum sum of square error finds the best core windows, minimum variance acts to select more centered cores, while user preference allows flexibility in desired results
3. Extend the best C candidates by 1/2 of --minWin in both directions; as well, consider the single maximum window made by the lowest starting position and highest stopping position among best C candidates (not extended further)
4. Find the best window within the extended candidate windows set using the same criteria as before

All UCEs are processed concurrently using the goroutines afforded by the Go programming language – further speed up is bounded by Amdahl’s law [Amdahl, 1967] as only IO is done sequentially.

Statement of Need

UCEs have the ability to resolve otherwise intractable phylogeny questions, but as these questions confront resolving more distant relationships or require longer UCEs to capture enough variation in the flanks to be useful, a more efficient method is required. Using a candidate window plus extension procedure cuts the combinatorial search from definite:

$$\sum_{i=0}^{N} \frac{n_i(n_i+1)}{2}$$

where \( N \) is the number of UCEs, \( n_i = (UCE_i - m) + 1 \), and \( UCE_i \) is the length of the \( i \)-th UCE

down to an upward bound of:

$$2 \sum_{i=1}^{N} \left\lfloor \frac{UCE_i}{m} \right\rfloor + C \frac{m(m+1)}{2}$$

where \( N \) is the number of UCEs, \( UCE_i \) is the length of the \( i \)-th UCE, \( m \) is the minimum window size, and \( C \) is the number of best candidates to consider.

In the initial project swsc was built for this change equated to roughly a \( 10^4 \) order of reduction in the search space.

Acknowledgements

I acknowledge Cynthia L. Frasier, Timothy M. Sefczek, and Melissa T. R. Hawkins for their input on decisions made during development.

References


