

1. Abstract

We present a new open source software package that contains a number of useful tools, datastructures and algorithms for inferring and visualising evolutionary patterns associated with reticulate evolution. The current version is focused on tools that either create or use split networks, such as SuperQ [2], FlatNJ [3], NetME [4] and several NeighborNet variants [5,6]. All of them have the advantage of producing split systems that can be represented in two dimensions, which makes them easier to assess and interpret. Furthermore we provide a tool for visualising the produced trees and networks. The tools are designed to have consistent graphical and command line interfaces and are efficient, reliable and portable to all common desktop platforms and high performance computing environments.

2. Users and Aims

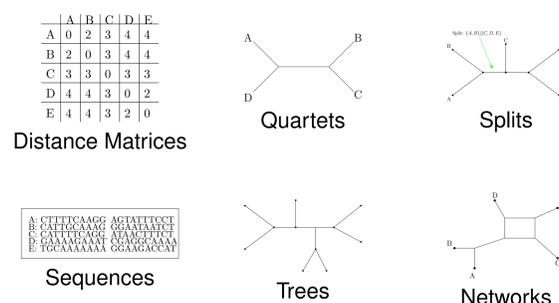
Our aim is to make phylogenetic tools accessible and useful to three groups.

- Biologists
 - Graphical interfaces
 - Viewing option
- Bioinformaticians
 - Tools usable in high performance computing environment
 - Command line interfaces
 - Support for common file formats
- Developers
 - Easy-to-integrate library
 - Clear Application Programming Interface (API)
 - Open source
 - Easy to reuse and extend codebase and add new tools

3. Core Library

We store interfaces and implementations of commonly used phylogenetic data structures, algorithms and resources in a separate library. This library allows us to reduce redundancy and inconsistency between tools by using shared code. It also makes these resources easy to reuse in both other future open source projects, and future SPECTRE tools.

- Data structures - We provide implementations for concepts such as: Splits, Trees, Networks, Distances, Quartets, Quadruples, Alignments, ect.

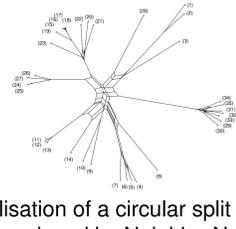
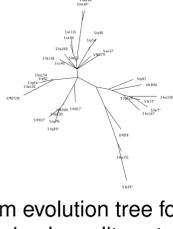
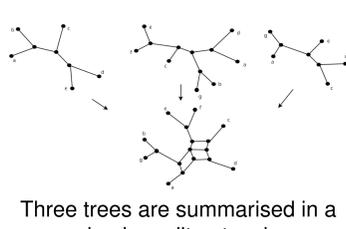
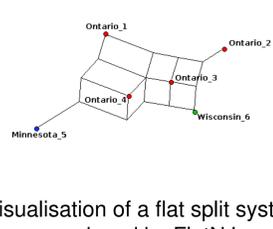


- Input and Output - Functionality that helps loading and saving of datastructures to commonly used phylogenetic file formats.



Some data structures can be stored in different file formats. Also some file formats can manage multiple data structures (e.g. Nexus). The SPECTRE IO subsystem helps to simplify the process of loading and saving data structures to and from any supported and appropriate file formats.

4. Tools

	NeighborNet [6] and variants [5]	NetME [4]	SuperQ [2]	FlatNJ [3]
Input	Distance matrix	Circular split system	Set of partial trees	Sequences or trees
Output	Circular split system	Tree	Circular split system	Flat split system
Features	<ul style="list-style-type: none"> • Gives snapshot of data • Runtime $O(n^3)$ • Implementations of several variations of original method 	<ul style="list-style-type: none"> • Uses circular split systems as search space for minimum evolution tree • Runtime of $O(n^4)$ • Compares favourably to another leading method (especially on ambiguous data) 	<ul style="list-style-type: none"> • Novel approach to incorporating information from input • Input is not restricted to trees • Scaling method • Similar results as current methods 	<ul style="list-style-type: none"> • Allows 2 D representation but inner vertices can be labelled with taxon name • Runtime $O(n^4)$
	 <p>Visualisation of a circular split system produced by NeighborNet.</p>	 <p>A minimum evolution tree found within a circular split system.</p>	 <p>Three trees are summarised in a circular split network.</p>	 <p>Visualisation of a flat split system produced by FlatNJ.</p>

Optimization Problems

SPECTRE supports several external solvers for linear and quadratic programming problems including GLPK, JOptimizer and Gurobi. Users can select which solver they wish to use at runtime, assuming it is installed and configured for their system.

Viewer

We provide a basic viewing option for visualising circular and flat split systems produced by SPECTRE tools. The viewer can be called directly on the output of the tools or separately on an input file.

Other

- Other useful tools performing smaller tasks like:
- QMaker - Breaks down trees into a set of quartets
 - GenQS - Prepares input for FlatNJ
 - Random distance matrix generator
 - Scaler - Scales trees within a set of trees

5. Making SPECTRE

Testing

We automate much of our testing regime through the use of both unit and integration tests that get executed via the maven buildcycle. This improves resistance against the introduction of bugs and therefore reliability of the software. We also provide a suite of test files that we use to manually test the software before final releases. These sample files may also be run by the users to test and familiarise themselves with the software.

Dependencies

We will provide a pre-compiled version of the software that requires only the Java runtime environment (V1.7+) to be installed on the users machine. However, some tools in SPECTRE solve optimisation problems, and these problems can be solved faster using external software, some of which must be installed separately. SPECTRE provides a mechanism for hooking into several open source and commercial solvers.

Source control

SPECTRE's source code will be made publicly available on github when the first release is ready. SPECTRE is a community project and we will encourage external contributions via the fork and pull request mechanism.

Documentation

Software manual will be hosted online via ReadTheDocs and is automatically updated when new releases are tagged on github.

References

- [1] S. Grünwald, K. Forslund, A. Dress, V. Moulton (2006) QNet: An Agglomerative Method for the Construction of Phylogenetic Networks from Weighted Quartets, *Molecular Biology and Evolution*, 24(2), 532-538.
- [2] S. Grünwald, A. Spillner, S. Bastkowski, A. Bögershausen, V. Moulton (2013) SuperQ: Computing Supernetworks from Quartets. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 10(1):151-60.
- [3] M. Balvocute, A. Spillner, V. Moulton (2014) FlatNJ: A novel network-based approach to visualize evolutionary and biogeographical relationships. *Systematic Biology*.
- [4] S. Bastkowski, A. Spillner, V. Moulton (2014) Fishing for minimum evolution trees with Neighbor-Nets. *Information Processing Letters*, 114(12): 1318.
- [5] D. Levy, L. Pachter (2010) The neighbor-net algorithm. *Advances in Applied Mathematics*, 47: 240- 258.
- [6] D. Bryant, V. Moulton (2004) Neighbor-Net: An Agglomerative Method for the Construction of Phylogenetic Networks. *Molecular Biology and Evolution*, 21 (2): 255-265.

¹The Genome Analysis Center

²Greifswald University, Institute for Mathematics and Computing Sciences

³University of East Anglia, School of Computing Sciences