*Journal Title Here,* 2022, pp. 1–3

**doi: DOI HERE**
Advance Access Publication Date: Day Month Year
Paper

PAPER

# Article Title

First Author,[1,*] Second Author,[2] Third Author,[3] Fourth Author[3] and Fifth Author[4]

[1]Department, Organization, Street, Postcode, State, Country, [2]Department, Organization, Street, Postcode, State, Country, [3]Department, Organization, Street, Postcode, State, Country and [4]Department, Organization, Street, Postcode, State, Country
[*]Corresponding author. email-id.com
FOR PUBLISHER ONLY Received on Date Month Year; revised on Date Month Year; accepted on Date Month Year

## Abstract

Abstracts must be able to stand alone and so cannot contain citations to the paper's references, equations, etc. An abstract must consist of a single paragraph and be concise. Because of online formatting, abstracts must appear as plain as possible.

**Key words:** keyword1, Keyword2, Keyword3, Keyword4

## Introduction

Phylogenetic networks is a acyclic directed graph that generalizes conventional bifurcating phylogenetic trees by allowing two branches to merge at a single node and createing a reticulation. Phylogenetic networks can depict complex biological scenarios that the trees cannot, such as hybrid speciation, introgression, allopolyploid speciation and so on. A hand of computational methods that estimate phylogenetic networkrs from genomic data has been proposed, however, the wide use of networks in practice is hindered by their lack of scalability. Scalability in general refers to the ability of a system to process a growing amount of work in a decreasing or stable amount of tim (Bondi, 2000). Even worse, phylogenetic network estimation belongs to the class of NP-Complete or also NP-Hard problems (non-deterministic polynomial-time). Some attempts to ameliorate this issue has been made previously but the computational burdern is too heavy for dataset size that are typically applied to the tree analyses.

One way to improve computational efficiency in the network inference is to pre-process the genomic sequence data into a set of gene trees in prior to the network inference, with which many existing methods implements (e.g., PhyloNet, SNaQ, etc.). Computational cost can be further ameliorated by using composite likelihood (or pseudolikelihood) framework, which decomposed the large network in a set of smaller problems (e.g., trees or networks with three or four tips), excecute likelihood computation on each of them, and combine them together to approximate the likelihood of the large network. This approach has been useful in several tree (e.g., MP-EST) and network (e.g., SNaQ, PhyNEST) inference methods and shown to be much faster than computing full likelihood of a topology without compromising accuracy.

Nevertheless, network inference is still a computationally demanding. It becomes prohibitive for more than 25 taxa using SNaQ, one of the fastest method available, and in practice, it is not uncommon that a network analysis only contains a handful of species (e.g., XXX, YYY, ZZZ) which narrows the scope of biological investigation. Like many existing bioinformatic tools, it is time to make use of different parallelization strategies for network estimation in order to reduce the execution time. In this study, we present SNaQ 2, an network inference method that enhances the computational speed through multithreading and making probabilistic decisions during the inference process. We first present SNaQ and the key improvemetns made in SNaQ2. Then, we present the result of our benchmarks that compares the accuracy and efficiency between SNaQ and SNaQ2; followed by appication of SNaQ2 on empirical datasets. Based on our study, we show that SNaQ2 estimates phylogenetic networks order of magnitude faster than the original SNaQ without compromising accuracy.

## Methods

### Original SNaQ

Phylogenetic networks in SNaQ are estimated from multi-locus data using pseudolikelihood. In brief, pseudolikelihood of a network is computed using the likelihood formulas of the quarnets, or four-taxon subnetworks. Since these quarnets are not independent with each other hence the likelihood calculated this approach as *pseudo*-likelihood.

SNaQ utilizes the concordance factor (CF) of a quarnet that refers to the proportion of genes whose true tree displays that quartet. A split divides a quartet into two, where each side of a split contains non-overlapping two taxa, and there are $\binom{4}{2}$ possible splits. For example, consider a taxon set $X \in \{a, b, c, d\}$, the three splits are $q_1 = ab|cd$, $q_2 = ac|bd$, and $q_3 = ad|bc$. Talk about G, X, CFs

$$L = \prod_{s \in S} (CF_{q_1})^{X_{q_1}} (CF_{q_2})^{X_{q_2}} (CF_{q_3})^{X_{q_3}} \quad (1)$$

© The Author 2022. Published by Oxford University Press. All rights reserved. For permissions, please e-mail: journals.permissions@oup.com

Now we have a way to quantify the fit of data onto a phylogenetic networks, the network that maximizes pseudolikelihood is searched using hill climbing algorithm. SNaQ uses one of six 'moves' to traverse the network space and jump between different dimensions. The five 'moves' are (1) nearest-neighbor interchange, (2) addition (or deletion) of a reticulation, (3) change direction of the reticulation edge, and move (4) the target or (5) the origin of an existing hybridization edge. One of five moves are randomly selected at each iteration.

## Improvements in SNaQ2

*Parallelization of the pseudolikelihood calculation*

Original SNaQ make use of parallelization mechanisms by allowing each run on different processors (or cores) using Julia package Distributed. SNaQ2 further improves the computational speed by multithreading pseudolikelihood calculation. In particular, extraction of quartet topologies from a network, calculation of expected CFs of the extracted quartet, calculation of delta CF (i.e., $\sum |CF_{obs_i} - CF_{exp_i}|$ where $i$ corresponds to one of three possible quartet topologies are parallelized. This setting allows to allocate all runs independently on seaprate high-performance computing nodess, with each node fully utilized to parallelize the pseudolikelihood calcuation for the run it is responsible for.

*Sampling subset of quartets for pseudolikelihood*

In the orginal SNaQ, all extracted quartets were used to compute pseudolikelihood of a network. While this computation is generally efficient, it may lead to the bottleneck as the number of taxa increases since there are $nchoose4$ quartets in a network. Several studies that also utilizes quartet has shown that subsampling some quartets lead to accurate estimation of a phylogeny (e.g., SVD Quartets). In SNaQ2, we added a new argument `propQuartets` in the network inference function `snaq!`. The argument `propQuartets` specifies the proportion of randomly sampled quartets for the pseudolikelihood calculation (i.e., nonnegative float equal to or smaller than 1).

*Proposals using quartet weighting*

During the network search, one of fives 'moves' mentioned above makes a modification in the original topology $N_0$ and propose the new topology $N_1$. All moves, except the move that changes direction of the selected reticulation edge, involves random selection of a tree edge in $N_0$ that will be modified. For example, to move the origin on an existing reticulation edge, a reticulation edge whose head is at a randomly selected reticulation node $u$ is selected at prorbability of $\gamma$, followed by a *random* selection of a tree edge that will have a new reticulation node in the middle. This stochasticity can result in increased time requires to find the global optimum during the searching process.

We make an improvement in heuristics by selecting the edge via weighted random sampling where the weight is equal to $\Delta CF = \sum_{i=1}^{3} |X_{q_i} - CF_{q_i}|$ calculated for every quartet extracted from a network. It can be done for a valid edges to choose from, although not implemented in the method. The rational here is... The new variable that is added to snaq! is probQR (varied from 0.0=full random to 1.0=full weighted).

## Evaluation using simulated and empirical data

*Simulation*

We evaluate the performance of SNaQ2 using simulation. A set of species networks wher each network has $n = \{10, 20, 30\}$ tips with 1 or 3 reticulations when $n = 10$ and 1, 3, or 5 otherwise was generated. For each $n$, we first generated a species tree under a Yule process using R package phytools, then we sequentially added reticulation onto the topology at arbitrary position. We checked each network is level-1 considering that snaq! can only infer networks in this class, both manually and using R package SiPhyNetworks. For each species network, we set each branch length $= \{0.5, 1.0, 2.0\}$ colescent unit to represent high, medium, and low amount of incomplete lineage sorting.

Using Julia package PhyloCoalSimulations, we generate a set of $g \in \{300, 1000, 3000\}$ gene trees for each species network. For each gene tree, we generated multiple sequence alignment that is $10^3$ bp long, setting the scale branch parameter=0.03 and base frequency of nucleotides as A=0.3, C=0.2, G=0.2, and T=0.3 under the HKY model. The generated sequence alignment was used to estimate a gene tree using IQ-TREE 1.6.12 with the best substituion model being identified withtin the sotftware with default parameters. Gene tree estimation error was measured using python package FastMulRFS.

The set of estimated gene trees were subsequently used as an input file for network estimation using SNaQ1 and SNaQ2. For SNaQ1, a randomly selected estimated gene tree was used as the starting topology, a table of CFs computed from the set of estimated gene trees were used as input, and the true number of reticulations were provided. For SNaQ2, all parameter setting was identical to SNaQ1 with additional parameters probQuartets$\in \{1.0, 0.9, 0.7\}$ and proqQR$\in \{0, 0.5, 1.0\}$. We recorded runtime for each network analysis. We evaluated the accuracy of the estimated network using hardwired cluster dissimilarity metric in Julia package PhyloNetworks. More specifically, we compared between the estimated network with the true network as well as the major trees of the estimated network and the true network.

All computations were done using condor at University of Wisconsin-Madison. We ran the analyses in different computing power setting the number of processors $\in \{4, 8, 16\}$ to compare the efficiency in various conditions. One hundred replicates were made.

*Empirical data*

Empirical data: 1. the fish data in snaq1; 2. find something else

# Results and discussion

## Simulation

GTEE

## Empirical data

# Conclusion

# Competing interests

No competing interest is declared.

# Author contributions statement

NK and SK designed and consucted analysis, prepared the manuscript. TC implemented improvements into SNaQ.

## Acknowledgments

WID Server?