SNPs2CF

https://github.com/melisaolave/SNPs2CF

An R function to compute Concordance Factors from SNP datasets

Melisa Olave
University of Konstanz, Germany
Department of Biology

Version 1.3
Last updated on January 31st 2020

About SNPs2CF()

This is an R function that performs the quartet-level concordance factor (CF) calculations from single nucleotide polymorphism (SNP) matrices. The output generated can later be loaded into the PhyloNetworks julia package (Solis-Lemus et al. 2017) to estimate phylogenetic networks. Information about PhyloNetworks can be found in its author's website https://github.com/crsl4/PhyloNetworks.jl

Although PhyloNetworks works taking upstream reconstructed gene trees, the advantage of SNPs2CF() is that it is now possible to use SNP data to reconstruct phylogenetic networks using PhyloNetworks.

Citation

Olave M. & Meyer A (2020). Implementing Large Genomic SNP Datasets in Phylogenetic Network Reconstructions: A Case Study of Particularly Rapid Radiations of Cichlid Fish. Systematic Biology (in press).

Background

Concordance Factors (CF) from SNP data

Let's first review how the CF calculations work when using gene trees. CF are calculated as the proportion of gene trees supporting the three possible splits in a given quartet (Figure 1). For example, assuming a sampling of 10 gene trees, and that there are five gene trees recovering the "topology 1", describing the association of AB_CD, plus a total of three gene trees displaying an alternative split "topology 2" = AC_BD, and two following the "topology 3" = AD_BC; then, the CF for $CF_{AB\_CD} = 0.50$; $CF_{AC\_BD} = 0.30$; and $CF_{AD\_BC} = 0.20$.

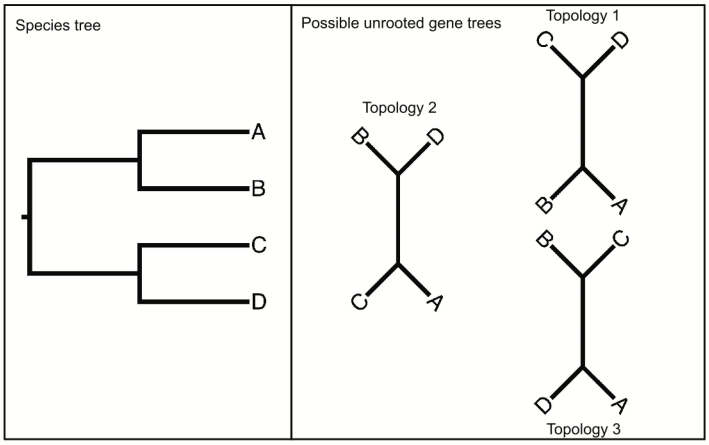

Figure 1: Example of a species quartet (left) and the three possible unrooted gene trees for a given species quartet (right). The concordance factors (CF) can be calculated as the proportion of observed gene trees supporting each topology. Thus, assuming a sampling of 10 gene trees, and that there are five gene trees recovering the "topology 1", describing the association of AB_CD, a total of three gene trees displaying the alternative "topology 2" AC_BD, and two following the "topology 3" AD_BC; then, the CF for $CF_{AB\_CD} = 0.50$; $CF_{AC\_BD} = 0.30$; and $CF_{AD\_BC} = 0.20$.

Following the same logical description of above, the CF can be calculated from SNP data. Each species quartet is considered, and from a sample of biallelic SNPs the proportion of sites

supporting each of the three possible alternatives is calculated. For example, consider the matrix below containing 10 SNPs:

Locus →	1	2	3	4	5	6	7	8	9	10
Sp A	1	1	0	1	0	1	1	0	0	1
Sp B	1	1	0	1	0	0	0	1	1	0
Sp C	0	0	1	0	1	1	1	0	1	0
Sp D	0	0	1	0	1	0	0	1	0	1

At this example, the number of times that the split AB_CD is shown, is equal to five (locus 1 to 5). Then, the split AC_BD appears three times (locus 6 to 8); and AD_BC appears twice (locus 8 and 10). Then, each occurrence is divided by the total number of SNPs in the matrix (=10 in this case), then: CF_{AB_CD} = 0.50; CF_{AC_BD} = 0.30; and CF_{AD_BC} = 0.20.

Some notes about the algorithm

The SNPs2CF() function will decompose the full matrix into all the different species quartet combinations and compute the CF using only biallelic SNPs and ignoring singletons, ambiguities and missing data. Indels are ignored by default, but it is possible to take them as a fifth state (setting indels.as.fifth.state = TRUE). Note that the number of SNPs that satisfy these conditions are commonly fewer than the full matrix and that, also, the number of SNPs used normally differs among quartets. However, a full matrix can be provided as input, and the function will later take advantage of the SNPs that satisfies the conditions in each quartet (i.e. some SNPs might be ignored for a given quartet, but can be useful for others).

Requirements

1. R program: https://www.r-project.org
2. R packages: foreach and doMC.
   To install, open R and type:

```
> install.packages("foreach", repos="http://R-Forge.R-project.org");

> install.packages("doMC", repos="http://R-Forge.R-project.org");
```

Input data

1. **SNP matrix (mandatory)**. This is an SNP matrix either including digits {0,1,2,3…9} or bases {A,G,T,C}. Each SNP is assumed to be independent (no-linked). Thus, each SNP has to be extracted from DNA segments separated long enough to reduce the linkage probability. Note that concatenating SNPs within the same locus is not appropriate, since it violates the independence assumption.
   Check the example matrix in example/5taxa-30K_SNPs.phy

2. **Imap (optional)**. A map file with the individual-species association is only required if there are multiple individuals per species. The format consists in two tab-separated columns. Names in columns are not important, but they have to be named, and the first column have to be the individuals (or alleles) and the second column are the species.
   The most important thing to know about the Imap file is that ONLY the taxa listed here will be considered for CF calculations. Thus, it is important to carefully include all the

taxa. However, this is useful if you want to manually subsample the individuals in the SNP matrix, since it is possible to simply delete them from the Imap file.
Check the example in examples/Imap.txt

Usage

```
SNPs2CF(wd=getwd(),seqMatrix, ImapName=NULL,
rm.outgroup=FALSE, outgroupSp="outgroup",
indels.as.fifth.state=FALSE, bootstrap=TRUE,
boots.rep=100, outputName="SNPs2CF.csv",
n.quartets="all", between.sp.only=FALSE,
starting.sp.quartet=1, max.SNPs=NULL,
max.quartets=100000, save.progress=TRUE, cores=1);
```

wd	working directory path. This is the path where the SNP matrix and Imap (optional) will be found. The current directory is taken by default.
seqMatrix	a character string defining the name of the SNP matrix. An example input file is provided in examples/5taxa-30K_SNPs.phy
ImapName	a character string defining the name of the Imap file with individual – species associations. An example input file is provided in examples/Imap.txt
rm.outgroup	a logical value whether to remove a taxon from the data. This was originally designed for outgroup removal, but any taxon listed here can be deleted. If FALSE (default), then no taxon is removed. If TRUE, then provide the name of the taxon using outgroupSp.
outgroupSp	a character string with the name of the taxon to be removed (only one allowed). If more than one taxon need to be removed, this can be done by manually removing them from the Imap.
indels.as.fifth.state	a logical value whether to take indels as fifth state (then, they are informative for CF calculations) or to ignore them. When FALSE (default) they are ignored, when TRUE they are considered as fifth state.
bootstrap	a logical value whether to calculate a credibility interval of CF from pseudoreplicates. It is strongly recommended to use the default value = TRUE. This information is used by PhyloNetworks for bootstrap calculations on the inferred network.
boots.rep	an integer with the number of replicates. By default 100 replicates are performed. It is ignored if boostrap = FALSE.
outputName	a character string defining the output name. By default "SNPs2CF.csv".

n.quartets	number of quartets to subsample *within* a species quartet. By default, all quartets are sampled. For details, see section "About subsampling quartets" below.
between.sp.only	a logical value whether to explore species-quartet level only (default = FALSE). This means that, when TRUE, all the sampled quartets will only involve different species (and not more than one individual of the same species). See details in the section "Sampling more than one individual per species within a quartet (between.sp.only = FALSE)".
starting.sp.quartet	an integer specifying the starting quartet number for the calculations. This element is useful when having a partial analysis that was not done and users want to continue from a more advance quartet. See also save.progress.
max.SNPs	an integer to limit the maximum number of SNPs desired to inform a species quartet. This is only useful for some specific analyses but, normally, in empirical datasets you want to set this as NULL (default) and use all the possible SNPs to inform a species quartet.
max.quartets	an integer to limit the number of species quartets. By default = 100000.
save.progress	a logical value whether to save the progress while the loop is running. If TRUE, then a temporal folder is created and one table is saved per species quartet. The tables can later be combined using the function combine.CF.table() (see tutorial).
cores	number of cores to parallelize calculations. Each species quartet can be analyzed in a different core. By default only 1 core is used.

About subsampling quartets

The SNPs2CF() functions allows to subsample the number of quartets, something that can be useful when having multiple individuals within species. Shortly, all possible quartet combinations among species are always considered, but the number of individuals can be randomly subsampled by providing an integer in n.quartets. Thus, if n.quartets = 2, a total of two individuals will be randomly subsampled for each species in a quartet. The number of quartets increases as: species quartet * n.quartets

Here, I explain in more details. First, let's make clear two definitions:

- **species quartet**: we refer here as "species quartet" to a group of four species. For example, in a matrix with species: sp1, sp2… sp5, and only one individual per species, then there are five total combinations of species quartets:

sp1, sp2, sp3, sp4
sp1, sp2, sp3, sp5
sp1, sp2, sp4, sp5
sp1, sp3, sp4, sp5
sp2, sp3, sp4, sp5

- **individual quartet**: When there are multiple individuals within a species, we refer here as "individual quartet" to a group of four different individuals (either from the same or different species). Let's assume for now between.sp.only = TRUE, and consider the example of a matrix with the same species than before (sp1 – sp5), but also two individuals per species labeled as sp1.A, sp1.B … sp5.A, sp5.B. Here, the total number of quartets increases importantly. In this example of two individuals per species, **only for the first species quartet** (sp1, sp2, sp3, sp4) there are a total of 16 possible individual quartets:

sp1.A, sp2.A, sp3.A, sp4.A
sp1.A, sp2.A, sp3.A, sp4.B
sp1.A, sp2.A, sp3.B, sp4.B
sp1.A, sp2.B, sp3.B, sp4.B
sp1.B, sp2.B, sp3.B, sp4.B
.
.
.
sp1.B, sp2.B, sp3.B, sp4.A

Thus, having two individuals per species, rises the number of the total quartet combinations from five to 80

5 species quartets * 16 individual quartets = 80

Then, the SNPs2CF() function allows to subsample the number of **individual quartets**. This means, that **always** all the possible species quartet combinations are considered, while the number of individuals sampled within species can be reduced to a given number. Thus, if there are multiple individuals per species and, for example n.quartets = 2, then each of the five species quartets is sampled twice (instead of 16). Then, the total number of quartets considered is

5 species quartet * 2 individual quartets = 10

reducing the total quartet number from 80 to only 10.

Sampling more than one individual per species within a quartet (between.sp.only = FALSE)

The number of total quartets becomes even higher when allowing more than one individual per species to be sampled in the same individual quartet (by setting between.sp.only = FALSE). In our previous example of two individuals (A, B) per five species (sp1, sp2 … sp5), when between.sp.only = FALSE the following are also included:

sp1.A, sp1.B, sp2.A, sp2.B
sp1.A, sp1.B, sp3.A, sp3.A
sp2.A, sp2.B, sp3.A, sp3.A
.
.
.
sp4.A, sp4.B, sp5.A, sp5.B

Up to three individuals of the same species are sampled ("*up to three*" because a quartet including four individual of the same species is not informative, then it is not computed). This means that the number of quartets in our example of two individuals per each of the five species rises now to 2,100 in total.
Note: between.sp.only = FALSE is only allowed when n.quartets = "all".

Output

The output is a CF table and should look like:

t1	t2	t3	t4	CF12_34	CF13_24	CF14_23	genes
sp1	sp2	sp3	sp4	0.927	0.029	0.044	2000
sp1	sp2	sp3	sp5	0.528	0.4065	0.0655	2000
sp1	sp2	sp4	sp5	0.5235	0.4135	0.063	2000
sp1	sp3	sp4	sp5	0.062	0.0705	0.8675	2000
sp2	sp3	sp4	sp5	0.032	0.03	0.938	2000

Each quartet is listed in one row. The three CF columns (CF12_34, CF13_24 and CF14_23) gives the proportion of SNPs that support each split. If bootstrap = TRUE then there will also be two extra columns per each CF for lower and upper limit of the credibility interval. For example, for the case of the first CF12_34, the lower limit is named CF12_34_lo and the upper limit CF12_34_hi.
The output table is in the same format than the one required by PhyloNetworks, and can be loaded into julia (see https://julialang.org):

```
julia> using PhyloNetworks
julia> CF = readTableCF("SNPs2CF.csv")
```

Then, it is possible to continue with the previously available PhyloNetworks pipeline. For further steps and phylogenetic network reconstruction, continue with the tutorials in http://crsl4.github.io/PhyloNetworks.jl/latest/man/snaq_plot/#Network-Estimation-1

Frequently Asked Questions (FAQ)

When to reconstruct a phylogenetic network instead of a strict coalescent-based phylogenetic tree?

A phylogenetic network is more appropriate than a phylogenetic tree when approaching organisms that hybridize (either in the present or the past). Traditional species tree methods assume a vertical transfer of genetic material from ancestral lineage to new lineages and they are not appropriate in the many cases of genes transferred horizontally in nature. Species tree

method working under the multispecies coalescent accommodate for the gene tree discordance due to ancestral polymorphism (i.e. incomplete lineage sorting [ILS], or deep coalescences) and completely ignore gene flow and horizontal gene transfer as a possible source of gene tree discordance. In consequence, it has been shown that phylogenetic tree inferences are highly bias when gene flow is present (Leaché et al. 2013; Solis-Lemus et al. 2016; Wen and Nakhleh 2017; Long and Kubatko 2018). In faces of these problems, the recently developed phylogenetic network methods include models that account for ILS and gene flow under the multispecies network coalescent (MSNC; Wen et al. 2016).

Why using SNPs in phylogenomics?

I recommend to check the nice review by Leaché and Oaks (2017) describing the advantages of SNP data applied to phylogenomics. Here, I mention the most important advantages of using SNPs in phylogenetic network reconstruction (compared to gene trees):

1. **Flexibility**. Reconstructing gene trees requires a suitable amount of mutations present in each locus. This is a problem many times when having short DNA sequences, such as the ones produced by genotyping by sequencing or RADseq technologies (normally ~150 bp). Additionally, there are many cases in which long sequences, such as the ones produced by target sequencing (e.g. ultra-conserved elements [UCE] or hybrid enrichment), the number of mutations are very low in recent speciation scenarios. Then, gene trees cannot be (or are poorly) reconstructed.
   On the other hand, SNPs can be extracted from sequences produced by any genomic sequencing technology, completely removing this limitation.

2. **Time consumption.** Using the SNPs2CF() function takes a fraction of time compared to the time required for gene tree reconstruction plus CF calculations in PhyloNetworks. Specifically, in our simulation study (five taxa), inferring 2,000 gene trees plus the CF calculations took around 2.5 hs in average, while CF calculations for 2,000 SNPs required as little as 20 – 30 seconds.

3. **No recombination issues**. Phylogenetic tree/network programs assume no within locus recombination. This is an assumption that cannot always be satisfied and, while it is usually desirable to have longer DNA sequences (to capture more mutations), the probability of within locus recombination increases as well.
   This is not an issue anymore when having SNPs.

References

Leaché, A. D., Harris, R. B., Rannala, B., & Yang, Z. (2013). The influence of gene flow on species tree estimation: a simulation study. *Systematic Biology*, *63*(1), 17-30.

Leaché, A. D., & Oaks, J. R. (2017). The utility of single nucleotide polymorphism (SNP) data in phylogenetics. *Annual Review of Ecology, Evolution, and Systematics*, *48*, 69-84.

Long, C., & Kubatko, L. (2018). The effect of gene flow on coalescent-based species-tree inference. *generations*, *1*, 2N.

Solís-Lemus, C., Yang, M., & Ané, C. (2016). Inconsistency of species tree methods under gene flow. *Systematic biology*, *65*(5), 843-851.

Solís-Lemus, C., Bastide, P., & Ané, C. (2017). PhyloNetworks: a package for phylogenetic networks. *Molecular biology and evolution*, *34*(12), 3292-3298.

Wen, D., Yu, Y., & Nakhleh, L. (2016). Bayesian inference of reticulate phylogenies under the multispecies network coalescent. *PLoS genetics*, *12*(5), e1006006.

Wen, D., & Nakhleh, L. (2017). Coestimating reticulate phylogenies and gene trees from multilocus sequence data. *Systematic biology*, *67*(3), 439-457.