Methods in Ecology and Evolution

British Ecological Society

Methods in Ecology and Evolution 2013 doi: 10.1111/2041-210X.12118

APPLICATION

SIDIER: substitution and indel distances to infer evolutionary relationships

Antonio Jesús Muñoz-Pajares*

Centro de Investigação em Biodiversidade e Recursos Genéticos, CIBIO, Campus Agrário de Vairão, Rua Padre Armando Quintas, 4485-661 Vairão, Portugal

Summary

1. The amount of evolutionary information in phylogeographic studies is usually limited due to the low divergence between closely related organisms. Consequently, obtaining the maximum evolutionary information contained in sequence alignments would be of great importance to infer phylogeographic relationships.
2. SIDIER is a software package that allows inferring evolutionary relationships from gapped alignments using information contained in both substitutions and insertions/deletions (indels).
3. SIDIER estimates the number of indel events that occurred during sequence evolution to obtain a distance matrix. This indel distance matrix may be combined with the substitution distance matrix calculated separately from the same data set.
4. Using this software, the inferred evolutionary events can be represented by means of percolation networks.
5. SIDIER is written in the open source R language and is freely available through the Comprehensive R Archive Network (http://cran.r-project.org/web/packages/sidier/index.html).

Key-words: evolutionary biology, molecular evolution, phylogenetics, phylogeography

Introduction

Although insertions and deletions (indels) may provide valuable information for the reconstruction of sequence evolution (Simmons, Ochoterena & Carr 2001; Blair & Murphy 2011), phylogenetic and phylogeographic inferences are usually based on nucleotide substitutions only as they dismiss gapped positions prior to the analysis (Talavera & Castresana 2007). Because combining indel and substitution information would surely improve the accuracy of evolutionary inference (Vogt 2002; Müller 2006) , an increasing number of studies have explored this possibility using different theoretical frameworks; such as maximum parsimony, maximum likelihood and Bayesian inference (e.g. Baum, Sytsma & Hoch 1994; Rivas & Eddy 2008). However, this question has been scarcely explored using distance-based methods (Ogden & Rosenberg 2007) despite the fact that these methods allow for an intuitive combination of the data from different sources and do not require lengthy computation times (Tamura, Nei & Kumar 2004). Such distance methods are particularly powerful to accurately reconstruct both short-branch trees and large evolutionary trees, regardless of their branch lengths (Roch 2010).

The main objective of the SIDIER package is to disentangle the evolutionary relationships between gapped sequences and between the populations they represent. It combines the information from both indels and substitutions in a distance-based framework. The complete process is represented in Fig. 1 and described below, together with the main features of the software implementation and a detailed example. The Supporting Information and the package manual provide details on specific functions and further examples.

*Correspondence author. E-mail: ajesusmp@cibio.up.pt

Description and implementation

DISTANCE ESTIMATION

Given an alignment, SIDIER estimates the number of indel events between each pair of sequences according to the Barriel method (as described by Simmons, Müller & Norton 2007), the Simple Indel Coding method (SIC; Simmons & Ochoterena 2000), the Modified Complex Indel Coding method (MCIC; Müller 2006) and following the fifth-state rationale. For the first two methods, SIDIER also provides an indel coding matrix that contains information on the presence or absence of gaps and their position in the aligned sequence.

Briefly, the SIC method codifies all gaps as different characters while leaving overlapping gaps as missing data (see Supporting Information). The same rationale is followed by the Barriel method, which also codes singleton gaps as fifth state not taken into account for distance estimation. Whereas these methods define the gap positions in the whole alignment, the MCIC method performs pairwise comparisons to define the indel events. MCIC ignores those gap positions that are shared by the two sequences being compared, which yields a biologi-

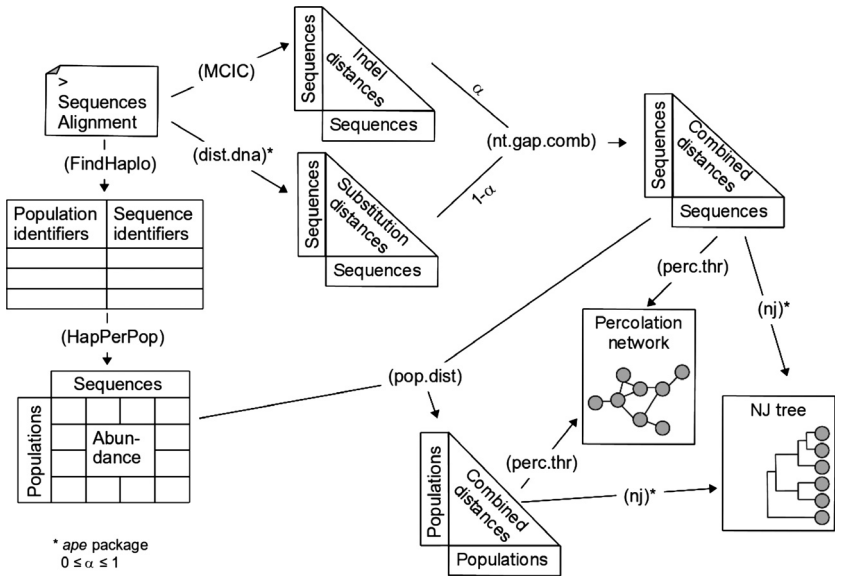


Fig. 1. Depiction of work-flow to obtain percolation networks by combining the information provided by substitutions and indels. Functions used for each step are represented in brackets. The procedure uses an alignment in fasta format to estimate distances between sequence based on indels and substitutions (the latter using the ape package), which are combined giving different weights to each matrix (α and 1-α, respectively). The resulting combined distance matrix can be represented and used (together with the abundance of sequences per population) to estimate a population distance matrix, which may also be represented. SIDIER represents distance matrices as percolation networks but see other packages for further representations.

cally realistic distance matrix (Müller 2006). For these three methods, a string of adjacent gap positions is treated as a single evolutionary event. Contrastingly, the fifth-state rationale treats all gapped sites as independent events (see Supporting Information).

If the studied sequences represent different geographical locations, SIDIER can also estimate the relationships among populations. A pairwise population distance matrix is built, where each element is calculated as the arithmetic mean of the distances between all the sequences sampled in both populations:

$$dist(i,j) = \frac{\sum_{k=1}^{m}\sum_{l=1}^{n} dist(H_{ki,}H_{lj})}{m * n} \qquad \text{eqn1}$$

where $dist(i,j)$ represents the distance between populations i and j; m and n are the number of sequences in populations i and j, respectively; and $dist(H_{ki},H_{lj})$ is the distance between the *k-th* sequence found in population i and the *l-th* sequence found in population j. To build the population distance matrix, SIDIER estimates and uses as input the abundance of each sequence in each population.

PERCOLATION NETWORKS

The relationships between well-differentiated taxa can certainly be properly represented by a tree topology. However, the evolution of closely related organisms (e.g. those showing recent origin or recurrent hybridization events) might be better represented using network approaches (Morrison 2005; Mardulyn 2012). To represent a distance matrix as a network, one has to define which distance values should be depicted as links between the nodes and which values should not (that is, a connection threshold must be defined). For that, SIDIER determines the percolation threshold as described by Rozenfeld *et al.* (2008) and connects only the populations that show genetic distances lower than the established threshold value.

To determine the percolation threshold, SIDIER calculates different networks assuming the connection thresholds provided by the user. To build each network, SIDIER computes a new distance matrix by setting to zero all the distances over the defined connection threshold, and the resulting matrix is handled as an adjacency matrix by *igraph* (Csardi & Nepusz 2006) and *network* (Butts 2008). For each network, SIDIER estimates the average size of clusters excluding the largest one (<*s*>; Rozenfeld *et al.* 2008) as follows:

$$< S > = \frac{1}{N}\sum\nolimits_{S<S\max} S^2 n_S \qquad \text{eqn 2}$$

where N is the number of nodes not included in the largest cluster, and n_s is the number of nodes containing s nodes. The percolation threshold is calculated as an increase in $< s >$ value as connection threshold decreases (Rozenfeld *et al.* 2008). By default, SIDIER calculates 101 networks using 101 different connection thresholds, ranging from 100% of the maximum distance found in the matrix (where all the nodes are connected) to 0% of this maximum distance (all nodes isolated). For each network, SIDIER can find modules (defined as subsets of nodes that conform densely connected subgraphs) by means of random walks as implemented in the *igraph* package.

In contrast with the traditional approaches that provide information on genealogy (e.g. median-joining or parsimony network), percolation networks group nodes into clusters

depending on their similarity. Consequently, the former may be more suitable for representing sequence relationships, whereas the latter may be better for identifying population relationships (because populations usually show, for instance, gene flow or recent common origins). However, haplotype networks are of crucial interest in biogeography and phylogeography (e.g. Chan, Brown & Yoder 2011), where several of the main assumptions of tree methods may be violated (Gurushidze, Fritsch & Blattner 2010). For instance, markers showing rapid evolution (as required in phylogeography) may usually show the coexistence of ancestral haplotypes and their descendant alleles (e.g. Zhao *et al*. 2013). It is also usual to find that ancient haplotypes have led to different observed alleles (e.g. Zhao *et al*. 2013), thus requiring multifurcating relationships to represent sequence evolution. Additionally, some markers may show complex mechanisms of evolution such as recombination (e.g. Ansell *et al*. 2007), which may also be represented using networks instead of tree topologies. Consequently, percolation networks may also constitute a useful tool for understanding the evolution of sequences with complex genealogies as well as of large data sets or closely related samples.

Using the aforementioned methodology, SIDIER allows visualization and comparison of the results estimated from indel and substitution distances of the same sequence data set (substitution distances can be estimated using, for instance, the *dist.dna* function in the R package *ape*). To allow visualization of a single network, the weighted combination of both indel and substitution matrices is also possible through the use of the *nt.gap.comb* function (Fig. 1 Combination of the two types of matrices must be carried out only if the information provided by each of these matrices is congruent (see Campbel, Legendre & Lapointe 2011 and CADM tests in *ape*). A separate analysis of the data sets followed by a comparison of the results is recommended for incongruent matrices, in order to determine the causes of the lack of congruency.

Even when the information provided by indels and substitutions is congruent, it may be difficult to determine the appropriate weight for combining both mutation types. Using equal weights for both sources may be the most conservative option if no further information is available. However, another interesting approach would be to consider the ratio of informative sites per mutation type as weight (i.e. if the substitutions/indel ratio is two, then the former distance must count twice than the latter; see the 'alpha = info' option within the function 'nt.gap.comb').

MISSING DATA AND AMBIGUOUS ALIGNMENTS

Although getting a confident alignment is of crucial importance for obtaining information from indels, this is not the goal of SIDIER, which assumes that the input alignment is correct. However, it is possible to incorporate the uncertainty of ambiguously aligned regions by estimating different distance matrices using each of the equally likely alignments. The resulting matrices can be combined and weighted, using the function 'distance.comb', to obtain one distance matrix that subsumes the information contained in the different alignments. This function is also helpful to combine distance matrices from multiple loci.

All symbols, except '-' (which represents a gap), are considered as a nucleotide position (regardless the symbol is 'A', 'T', 'C', 'G', 'Y', 'R', '?', …). To infer as correctly as possible the homology of the inferred gaps, it is desirable to include only gaps that are flanked by known sequences. The analysis of alignments containing gapped extremes is not possible unless using the 'addExtremes = TRUE' option.

COMPARING THE ACCURACY OF THE PROPOSED METHODS

SIDIER includes a function for simulating sequence evolution according to the provided values of substitution, indel and insertion/deletion rates. This function is helpful for testing the best approach to reconstructing the evolutionary process (e.g. distance method, combination weights,…) based on the evolutionary history of the sequences (e.g. topology, branch lengths, …).

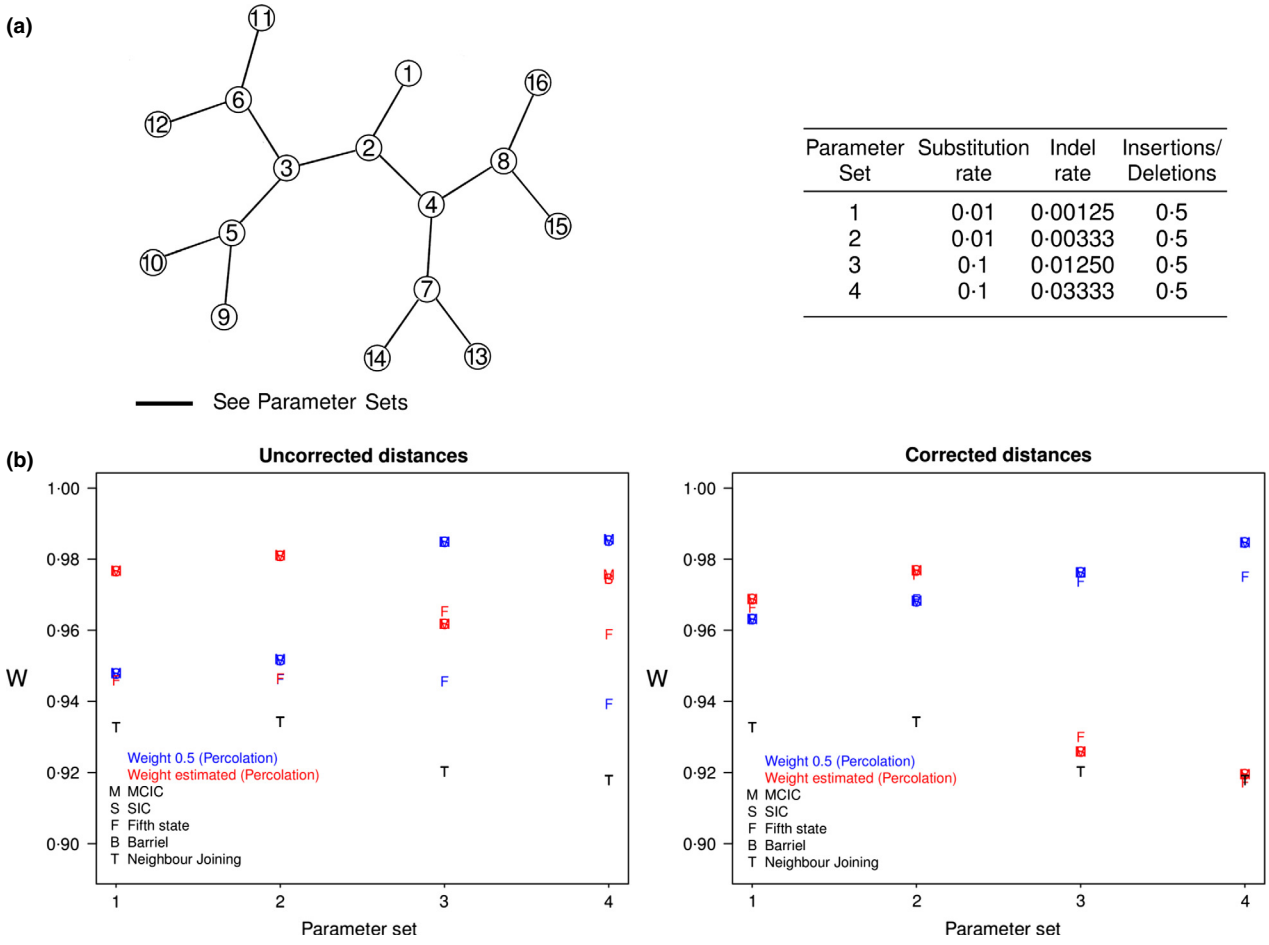
The simple simulation represented in Fig. 2 shows that the methods implemented in SIDIER accurately reconstruct the original topology. In fact, in all cases, the yielded Kendall's coefficient of concordance (W; see Campbell, Legendre & Lapointe 2011) is higher than the one obtained using the information provided by substitutions only (K2P method and NJ tree as implemented in *ape*; Paradis, Claude & Strimmer 2004). These results also suggest that the best method for reconstructing the original topology depends on the parameter set. Further studies are required to evaluate the performance of each method under different evolutionary scenarios.

Complete example

As an example, this section guides the user step-by-step to carry out the computation described above. A single batch file is also available in the Supporting Information. Before going through these steps, one has to take into account that the complementary use of functions implemented in SIDIER and other existing packages such as *ape*, *adegenet* and *pegas* may provide a more complete understanding of the evolutionary processes that shaped the studied sequences and populations.

Step 1: Defining the input alignment. This example analyses the file 'align1.fas' available from the Supporting Information. Download the file to your computer and select its folder as the R working directory using setwd('/folder/in/your/computer'). Create a variable that contains the alignment file name using *inputAll <-"align1.fas"*. To facilitate population distance estimation, it is recommended for the input file to include population information and to use equal-length names (e.g. Pop01Id001, Pop74Id133,…).

Step 2: Selecting unique sequences. Repeated sequences (e.g. haplotypes shared by several individuals) increase computation time but do not provide additional information (because they produce duplicated rows and columns in the final distance matrix). To reduce computation time, do



Parameter Set	Substitution rate	Indel rate	Insertions/ Deletions
1	0·01	0·00125	0·5
2	0·01	0·00333	0·5
3	0·1	0·01250	0·5
4	0·1	0·03333	0·5

Fig. 2. Simulation summary. Simulations consisted in sixteen sequences of 1000 bp length each. Four parameter sets for substitution and indel rates were tested under the same topology of equal branch lengths and using indel events involving 30 bp each. (a) Simulated topology and the four different parameter sets defining branch lengths. (b) Congruence mean values (measured as Kendall's coefficient of concordance, W, using CADM tests) for each parameter set using different methods implemented in SIDIER and the NJ over substitution distances. Indels and substitutions were combined using the same weight each (blue) and the proportion of informative mutations (red). Each point was estimated using 100 simulated replicates. Standard errors were smaller than symbol sizes.

estimate and write in a new file the unique sequences (haplotypes) contained in the input alignment, using *GetHaplo(inputFile = inputAll, outname = "align_unique1.fas")*. Sequence names in the new haplotype file are identical to those in the original input file. To automatically define a new name for each haplotype (recommended for population network estimation), use *GetHaplo(inputFile = inputAll, outname = "align_unique2.fas", seqsNames = "Inf.Hap")*

Step 3: Defining the new input alignment. To create a variable containing the file name of the haplotypes alignment, use *inputUnique <- "align_unique2.fas"*.

Step 4: Indel distances. To estimate (and save in a file) the indel-based distance matrix between unique sequences type, use *distGap <- MCIC(input = inputUnique, saveFile = TRUE)*.

Step 5: Substitution distances. The substitution-based distance matrix between unique sequences is obtained using the ape package (Paradis, Claude & Strimmer 2004) as follows:

Step 5·1: Read the alignment using *align <- read.dna(file = inputUnique,format = "fasta")*.

Step 5·2: Estimate p-distances using *dist.nt <- dist.dna(align,model = "raw",pairwise.deletion = TRUE)*.

Step 5·3: Create a matrix class object from the estimated distance class object using *DISTnt <- as.matrix(dist.nt)*.

Step 5·4: Write the distance matrix in a file using: *write.table(file = "align_unique2.fas SubstitutionMatrix.txt",DISTnt)*.

Step 6: Combining distances. To combine and save indel and substitution distance matrices, giving the same weight to each matrix and using the corrected method, use *CombinedDistance <- nt.gap.comb(DISTgap = distGap, alpha = 0·5, method = "Corrected", saveFile = TRUE, DISTnuc = DISTnt)*.

Step 7: Haplotype percolation network. SIDIER allows representing the combined distance matrix as a percolation network that shows sequence modules. For that, use *perc.thr(dis = as.data.frame(CombinedDistance), modules = TRUE)* (Fig. 3A and see Supporting Information for further percolation network options). Alternatively,

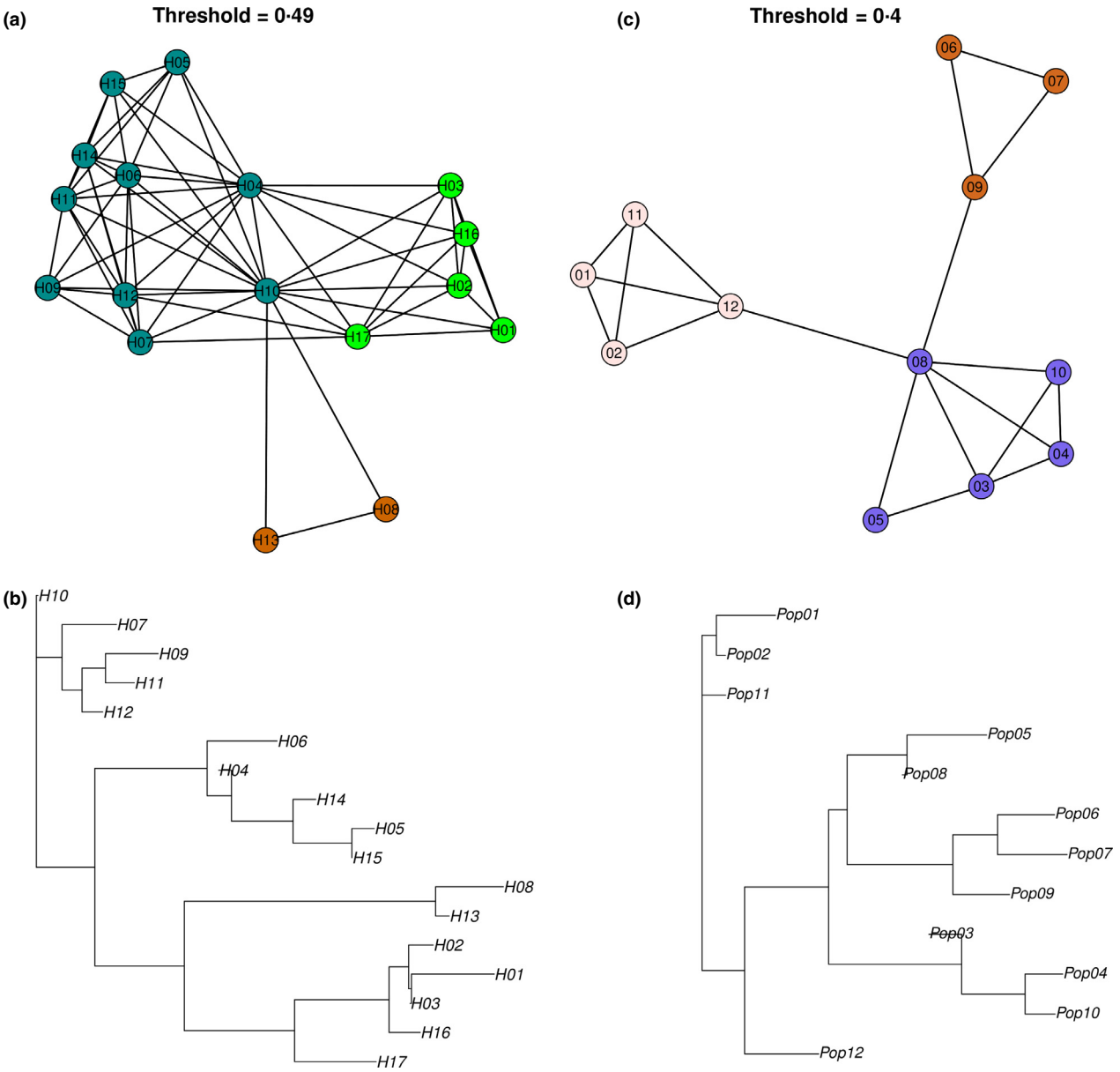


Fig. 3. Percolation networks and NJ trees resulting from the example file analysis. (a, b) Haplotype relationships. (c, d) Population relationships. Trees were estimated using the bionj function in ape.

sequence distances may be represented using genealogical approaches, such as NJ (Fig. 3B). Additionally, SIDIER can estimate and represent population relationships. For that, the haplotypic composition for each individual as well as the abundance of each haplotype per population must be defined. It is possible to include this information automatically as follows:

Step 8: Haplotype names. Define the haplotype names in the original input alignment using *HaplosAll < -FindHaplo (input = inputAll,saveFile = TRUE, outname = paste("FindHaplo", inputAll,sep = "_"))*.

Step 9: Checking row names: To estimate the number of haplotypes per population, provide the position of the initial and final characters indicating population names. For that, check the individual names defined by *HaplosAll[1]*. Note that, for this example, all individuals show population names spanning from character 1 to 4. It may also be convenient to check the distance matrix row names using *row.names(CombinedDistance)* to identify the position of the initial and final characters that indicate the haplotypes names (ranging from characters 1 to 3 in this example).

Step 10: Haplotypes per population. The population names values from Step 9 are included in the *HapPepPop* function. To estimate the number of haplotypes per population, use *HaplosPop < -HapPerPop(saveFile = TRUE, input = HaplosAll, NameIniPopulations = 1, NameEndPopulations = 4)*.

Step 11: Population distance. To estimate the population distances, we include in *pop.dist* function, the haplotype distance matrix (Step 6), the number of haplotypes per population (Step 10) and the haplotype and population positions in string names (Step 9). For that, use *PopDist < -pop.*

dist(distances = CombinedDistance, Haplos = HaplosPop[[1]], logfile = TRUE, saveFile = TRUE, NameIniHaplotypes = 1, NameEndHaplotypes = 3, NameIniPopulations = 1, NameEndPopulations = 4). Note that the haplotype names are read from the distance matrix and that the population names correspond to sequence names in the original alignment.

Step 12: Population percolation network. Finally, the resulting population distance can be represented as a percolation network using *perc.thr(dis = as.data.frame(PopDist), modules = TRUE, label = substr (row.names(as.data.frame (PopDist)),4,5))* (Fig. 3C).

Acknowledgements

I thank M. Bakkali, E. Paradis, M.P. Simmons and three anonymous reviewers for improving this manuscript. I am also grateful to M. Abdelaziz, F. Perfectti and J.M. Gómez for comments during methods development and the Spanish Ministry of Education and Science [FPU:AP2006-00685] for funding.

References

Ansell, S.W., Schneider, H., Pedersen, N., Grundmann, M., Russell, S.J. & Vogel, J.C. (2007) Recombination diversifies chloroplast trnF pseudogenes in *Arabidopsis lyrata*. *Journal of Evolutionary Biology*, **20**, 2400–2411.

Baum, D.A., Sytsma, K.J. & Hoch, P.C. (1994) A phylogenetic analysis of *Epilobium* (Onagraceae) based on nuclear ribosomal DNA sequences. *Systematic Botany*, **19**, 363–388.

Blair, C. & Murphy, R.W. (2011) Recent trends in molecular phylogenetic analysis: where to next? *Journal of Heredity*, **102**, 130–138.

Butts, C.T. (2008) network: A Package for Managing Relational Data in R. *Journal of Statistical Software*, **24**, 1–36.

Campbell, V., Legendre, P. & Lapointe, F.-J. (2011) The performance of the Congruence Among Distance Matrices (CADM) test in phylogenetic analysis. *BMC Evolutionary Biology*, **11**, 64.

Chan, L.M., Brown, J.L. & Yoder, A.D. (2011) Integrating statistical genetic and geospatial methods brings new power to phylogeography. *Molecular phylogenetics and evolution*, **59**, 523–537.

Csardi, G. & Nepusz, T. (2006) The igraph software package for complex network research. *InterJournal Complex Systems*, 1695.

Gurushidze, M., Fritsch, R.M. & Blattner, F.R. (2010) Species-level phylogeny of *Allium* subgenus *Melanocrommyum*: Incomplete lineage sorting, hybridization and trnF gene duplication. *Taxon*, **59**, 829–840.

Mardulyn, P. (2012) Trees and/or networks to display intraspecific DNA sequence variation? *Molecular Ecology*, **21**, 3385–3390.

Morrison, D.A. (2005) Networks in phylogenetic analysis: new tools for population biology. *International Journal for Parasitology*, **35**, 567–582.

Müller, K. (2006) Incorporating information from length-mutational events into phylogenetic analysis. *Molecular Phylogenetics and Evolution*, **38**, 667–676.

Ogden, T.H. & Rosenberg, M.S. (2007) How should gaps be treated in parsimony? A comparison of approaches using simulation. *Molecular Phylogenetics and Evolution*, **42**, 817–826.

Paradis, E., Claude, J. & Strimmer, K. (2004) APE: analyses of phylogenetics and evolution in R language. *Bioinformatics*, **20**, 289–290.

Rivas, E. & Eddy, S. (2008) Probabilistic phylogenetic inference with insertions and deletions. *PLoS Computational Biology*, **4**, e1000172.

Roch, S. (2010) Toward extracting all phylogenetic information from matrices of evolutionary distances. *Science*, **327**, 1376–1379.

Rozenfeld, A.F., Arnaud-Haond, S., Hernández-García, E., Eguíluz, V.M., Serrão, E.A. & Duarte, C.M. (2008) Network analysis identifies weak and strong links in a metapopulation system. *Proceedings of the National Academy of Sciences*, **105**, 18824–18829.

Simmons, M.P., Müller, K. & Norton, A.P. (2007) The relative performance of indel-coding methods in simulations. *Molecular Phylogenetics and Evolution*, **44**, 724–740.

Simmons, M., Ochoterena, H. & Carr, T. (2001) Incorporation, relative homoplasy, and effect of gap characters in sequence-based phylogenetic analyses. *Systematic Biology*, **50**, 454–462.

Simmons, M.P. & Ochoterena, H. (2000) Gaps as Characters in Sequence-Based Phylogenetic Analyses. *Systematic Biology*, **49**, 369–381.

Talavera, G. & Castresana, J. (2007) Improvement of phylogenies after removing divergent and ambiguously aligned blocks from protein sequence alignments. *Systematic Biology*, **56**, 564–577.

Tamura, K., Nei, M. & Kumar, S. (2004) Prospects for inferring very large phylogenies by using the neighbor-joining method. *Proceedings of the National Academy of Sciences of the United States of America*, **101**, 11030–11035.

Vogt, L. (2002) Weighting indels as phylogenetic markers of 18S rDNA sequences in Diptera and Strepsiptera. *Organisms Diversity & Evolution*, **2**, 335–349.

Zhao, Y., Qi, Z., Ma, W., Dai, Q., Li, P., Cameron, K.M., Lee, J., Xiang, Q.-Y. & Fu, C., (2013) Comparative phylogeography of the *Smilax hispida* group (Smilacaceae) in eastern Asia and North America – Implications for allopatric speciation, causes of diversity disparity, and origins of temperate elements in Mexico. *Molecular Phylogenetics and Evolution*, **68**, 300–311.

Received 1 July 2013; accepted 11 September 2013
Handling Editor: Emmanuel Paradis

Supporting Information

Additional Supporting Information may be found in the online version of this article.

Fig. S1. Example of an indel distance matrix estimation using the *MCIC* function.

Fig. S2. Schematic view of the MCIC pairwise indel distance estimation between sequences C and E represented in Figure S1.

Fig. S3. Example of an indel distance matrix estimation using the *SIC* function.

Fig. S4. Example of an indel distance matrix estimation using the *BARRIEL* function.

Fig. S5. Example of an indel distance matrix estimation using the *FIFTH* function.

Data S1. Sequence alignment analysed using SIDIER as described in the main text.

Data S2. Sequence alignment used as input for the batchfile provided in Data S4.

Data S3. Commented script to perform a complete sequence analysis using SIDIER.

Data S4. Batchfile to perform a customized analysis using SIDIER in only 5 steps (see instructions in the first lines of the file).

