TwisstNTern 2: Ternary Analysis of Topology Weights from Tree Sequences

Hila Lifchitz* [1], Sanketh Vedula [2,3], Arka Pal [1], Daria Shipilina [1], and Sean Stankowski* [4]

[1]Institute of Science and Technology Austria (ISTA), Klosterneuburg, Austria
[2]Department of Electrical and Computer Engineering, Princeton University, USA
[3]Schmidt Center, Broad Institute of MIT and Harvard, USA
[4]Department of Ecology and Evolution, University of Sussex, Falmer, Brighton, UK

*Corresponding authors: Hila.Lifchitz@ist.ac.at; S.Stankowski@sussex.ac.uk

Abstract

Recent advances in genealogical inference now allow the reconstruction of genome-wide sequences of trees for large sets of samples, providing detailed records of how evolutionary relationships vary along the genome. Tree sequences encode a vast amount of information, but new approaches are needed to extract relevant patterns and make inferences. *TwisstNtern* is a program for visualizing and analyzing topology weights from four-population tree sequences. *TwisstNtern* takes a wide range of tree sequence formats as input, conducts topology weighting using the *Twisst* algorithm, and projects the topology weights in a ternary plot. This enables intuitive visualization of the joint distribution of weights and formal tests for asymmetrical genealogical discordance caused by processes such as introgression. The package also includes tools for simulation, significance testing, and comparison between empirical and simulated datasets.

1 Introduction

Advances in genealogical inference have transformed the way we study evolutionary processes from genomic data. New methods now make it possible to reconstruct the genome-wide tree sequence, which is a 'complete' set of genealogies describing how a set of genomes has been shaped by coalescence and recombination events throughout history ([Kel+19], [DNS24], [Spe+19]). Tree sequences are extraordinarily rich in information, and provide access to the branching patterns and coalescent times that underlie genetic variation ([RTK20]). However, the scale and complexity of these data present new challenges and new tools are needed to efficiently translate the vast topological variation into interpretable patterns ([Shi+23]).

One powerful framework for doing so is topology weighting ([MV17]). Topology weighting quantifies the proportion of genealogical support for each of the possible relationships among a set of taxa by sampling and classifying subtrees that contain only one sample of each group ([MV17]). In a tree with four populations, only three unrooted subtree can be observed, and their relative frequencies can be visualized and analyzed to detect processes such as incomplete lineage sorting (ILS) and introgression. This logic underlies many classical tests, including the ABBA–BABA test, which measures asymmetry in site patterns to detect gene flow ([Gre+06]). Topology weighting generalizes this idea from individual sites to entire genealogies, allowing evolutionary processes to be inferred directly from tree sequences rather than from sequence alignments or allele frequencies.

Here, we present TwisstNtern, a program for visualizing and analyzing topology weights from four-population tree sequences. TwisstNtern maps each local genealogy to a point within a ternary plot, enabling the joint distribution of weights to be visualized for an entire tree sequence. By exploring how evolutionary parameters – such as divergence times, effective population sizes, and migration rates – affect the shape and symmetry of this distribution, the program provides an intuitive and quantitative framework for visualizing tree-space and detecting signals of introgression, divergence, and ILS in tree-sequence data.

2 Logic and basic operations of TwisstNtern

We first used TwisstNtern to study the genetic basis of the evolution of reproductive mode in Littorina snails ([Sta+24]), and subsequently to study the evolution of flower color in Antirrhinum majus ([Pal+]). This version, which can be installed using pip, includes several new features. First, it takes a wide range of tree sequence formats as input, automatically performs the weighting using the *Twisst* algorithm, and projects the topology weights onto a ternary plot. This enables intuitive visualization of the joint distribution of weights and formal tests for asymmetrical genealogical discordance caused by processes such as introgression. The package

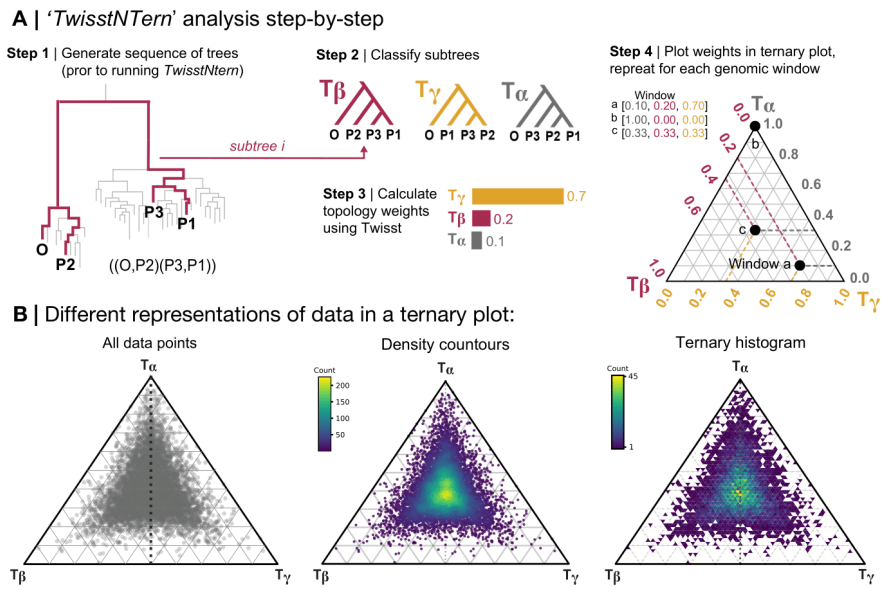

Figure 1: (A) Overview of TwisstNtern analysis, including the inference of a tree sequence (done prior to running TwisstNtern), topology weighting in *Twisst* ([MV17]), and plotting the weights in the ternary plot. (B) Different representations of the ternary distribution, including the full distribution of all data points, a contoured heatmap and a ternary histogram

also includes tools for simulation, significance testing, and comparison between empirical and simulated datasets. Here we outline the basic operations performed by TwisstNtern, as well as some of the logic behind the inference scheme. In doing so, we assume that the reader understands what topology weights are and how they are calculated. Readers are advised to consult [MV17] for a detailed description of the *Twisst* algorithm before continuing with this manuscript.

2.1 Plotting the joint distribution of topology weights in a ternary plot

In a tree with four taxa, a two-dimensional simplex– more commonly known as a ternary plot– provides a natural and intuitive framework for analyzing the distribution of these weights. This is because it is possible to graphically represent each tree as a single point in an equilateral triangle (Figure 1). The three corners of the ternary plot, defined by coordinates $[1, 0, 0], [0, 1, 0]$ and $[0, 0, 1]$, correspond to trees where 100% of the sampled subtrees perfectly match one of the three alternative topologies, indicating that each of the four groups is monophyletic (Figure 1). In contrast, the very center of the ternary plot–the point whose coordinates are $[0.33, 0.33, 0, 33]$– corresponds to a genomic window in which all three topologies were sampled with equal frequency. Any other location in the ternary plot indicates a bias towards one of the topologies (Figure 1). By representing every tree in the sequence as a single point in the ternary plot, we can view the joint distribution of topology weights for a tree sequence with a length of L trees (Figure 1). This may ultimately become difficult to visualize for long tree sequences, so it is convenient to show the distribution of topology weights as a density function or divide the ternary space into small bins that can be colored according to the local density of trees (Figure 1).

2.2 Factors that influence the ternary distribution of weights

Ternary analysis exploits the fact that different evolutionary processes and factors impact the ternary distribution of topology weights for many loci. The effects we describe are predicted by coalescent theory ([Wak16]), but we also illustrate them by simulating gene histories under

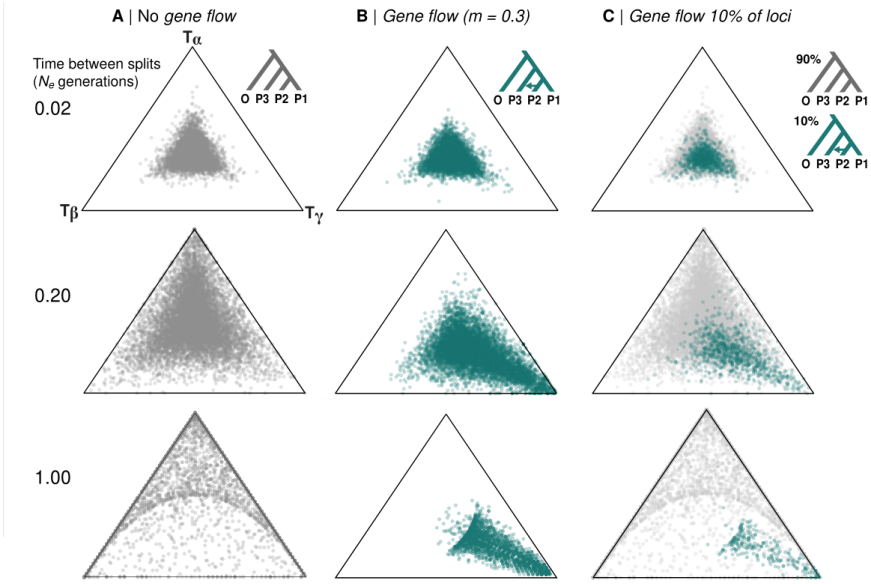

Figure 2: **Simulations show how the degree of lineage sorting and gene flow affect the ternary distribution of weights.** (A) A greater opportunity for lineage sorting biases the distribution toward the topology that matches the demographic history of the populations (i.e., T_α). Note that the distribution is always symmetrical between the left and right half-triangles. (B) Gene flow between non-sister lineages creates bias toward one of the discordant topologies, breaking the left-right symmetry. (C) A mix of both processes mimics a scenario where a small fraction of the genome experiences introgression (here 10%)

a four-population model using the coalescent simulator MSprime ([Bau+22] (Figure 2). Note that other factors aside from the ones we describe, such as selection, may affect the distribution of weights. A more detailed simulation study can be found in the supplementary information of [Sta+24] In all of our simulations, the four populations O, P_3, P_2, P_1 have the relationship $(O\,(P_3\,(P_2, P_1)))$. These populations arise from three population splits going backwards in time: at time t_1, the ancestral population P_{12} splits to form the descendant populations P_1 and P_2; at time t_2, P_{12} and P_3 diverge; and at time t_3, the common ancestor of all descendant populations splits, giving rise to the lineages O and P_{123}. For each scenario, we simulated $10,000$ independent coalescent trees representing independent loci, performed topology weighting, and projected the weights in a ternary plot. We modified this base model by varying the parameters to illustrate their individual and combined effects. We first varied the split times scaled relative to the effective population size (i.e. time in N_e generations) in a four-population model without migration. As a general expectation, any bias in the distribution of weights should always be towards the subtree topology that matches the branching history of the four populations (i.e. T_α, oriented at the top of the triangle). This is because population structure makes it more likely that gene copies sampled from the same population will coalesce with each other, rather than with a gene copy sampled from one of the other three populations. However, the probability of coalescence is determined by two parameters: the effective population size, N_e, which determines the rate of coalescence (i.e., the probability that two gene copies coalesce in the previous generation is $\frac{1}{2N_e}$ in a diploid population), and the time since the population split in generations, t, which determines how that rate accumulates into a meaningful probability that coalescence will occur.

As expected, time has a strong influence on the distribution of topology weights (Fig. **??**). When t is small (i.e., $t \ll 2N_e$ generations), coalescent events occur largely in deeper parts of the tree, making concordant and discordant topologies equally likely. This results in similar weights for

all three subtrees (~ 0.33 each), and the mass of the distribution is in the center of the ternary plot. As t increases, more lineages will coalesce before reaching deeper ancestral populations, causing the weight of T_α to exceed those of T_β and T_γ. This shifts the distribution of loci to the apex of the triangle that corresponds to T_α. As t becomes very large (i.e., $t \gg 4N_e$ generations), all loci will inevitably show perfectly concordant trees, with weights for T_β and T_γ dropping to zero and all points falling precisely at the apex.

When $t < 4N_e$ generations, a substantial fraction of coalescent events will be discordant, resulting in non-zero weights for T_β and T_γ. In the standard four-population model without migration, these two discordant topologies occur with equal probability, so the distribution of loci is symmetrically weighted toward both corners. This symmetry reflects the fact that, for any given locus, there is an equal chance that it will resemble either of the discordant topologies. As a result, the ternary plot is vaguely mirrored across the median axis that runs from the base to the apex (Fig. 2).

Migration between non-sister populations breaks the symmetry of the ternary distribution by making one of the discordant topologies more common than the other. For example, gene flow from population P_2 into P_3 increases the probability that gene copies from these populations will coalesce, leading to an excess of loci supporting the discordant topology where P_2 and P_3 are sister to one another. This results in a skewed distribution, with a higher density of points on one side of the ternary plot (Fig. 2). The degree of asymmetry depends on both the proportion of the genome affected by introgression and the rate of migration m (i.e., the number of migrants per generation). However, when $t \ll 2N_e$, coalescence is largely random and the signal of gene flow is masked, making migration difficult to detect.

2.3 Partitioning the ternary space

When analyzing the distribution of weights to quantify the patterns outlined above, it is useful to subdivide the ternary plot into smaller areas. For a genome-wide test for asymmetrical discordance, we divide the ternary plot into two equal halves by bisecting it with a median line from the apex that corresponds to the topology T_α. In practice, counting the number of genomic regions that fall on each side, n_{Left} and n_{Right}, can be done by finding the number of cases where the weights of one of the two alternative topologies (i.e. T_β and T_γ) are greater than the other: points where $T_\beta > T_\gamma$ are on the left side, and points where $T_\gamma > T_\beta$ correspond to points on the right side (Fig. 3).

For other applications, we divide the ternary plot into a fine grid of smaller triangular bins to analyze local regions of the distribution (Fig. 3). This is done by selecting a grid size α (e.g. 0.25, 0.1, or 0.05). Using this α, we define sub-triangles by choosing ternary coordinates (x_1, x_2, x_3) from the sequence $\{0, \alpha, 2\alpha, \ldots, 1-\alpha\}$. Each coordinate set defines a region as a half-open interval: $(x_1, x_1+\alpha] \times (x_2, x_2+\alpha] \times (x_3, x_3+\alpha]$ corresponding to a subtriangle in our space. Only coordinate combinations within the simplex are valid (i.e., those whose coordinates sum to 1). At the triangle's edges, intervals starting at zero are closed to ensure full coverage. Once the grid is defined, we systematically scan each valid sub-triangle, excluding those that cross the central horizontal line (to maintain left/right consistency); while coarse grids may leave larger gaps along the center line, finer grids minimize these and provide more precise local analysis. We give more advice on choosing an appropriate grid granularity based on the characteristics of a given dataset further below.

2.4 Quantifying symmetry with D_{LR} and significance testing

The results of the simulations outlined above show that divergence under an idealized four-population model without migration produces a symmetrical distribution of topology weights.

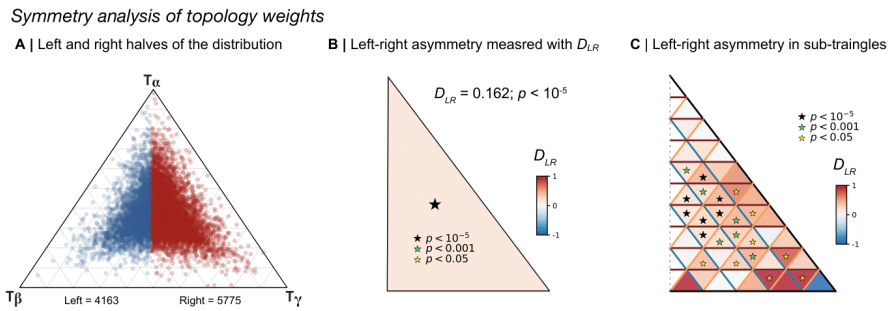

Figure 3: **Symmetry analysis at different scales.** (A) A genome-wide test for asymmetry is conducted by tesfting for an inequality of trees between the left and right half-triangles, quantified by the statistic D_{LR}. (B) The significance of the DLR estimate can be determined using a G-test. (C) Left-right asymmetry can be evaluated for each sub-triangle in a grid.

This occurs because, under incomplete lineage sorting (ILS), there is an equal chance that any given gene tree will resemble either of the alternative topologies.

Deviations from a simple four-population model (including gene flow and selection) can lead to a bias in the probability towards one alternative topology, leading to an asymmetrical distribution of topology weights. This asymmetry can be quantified using the statistic D_{LR} (Fig. 3). D_{LR}, which is similar to Patterson's statistic D [Gre+06], can range from -1 to $+1$ and gives an indication of the strength of the bias towards an alternative topology compared to the expectation of equality ($D_{LR} = 0$). A genome-wide estimate of D_{LR} can be calculated directly from the topology weights as:

$$D_{LR} = \frac{n_{\text{right}} - 0.5n_T}{0.5n_T} \tag{1}$$

where n_{right} is the number of trees on the right side of the plot and n_T (T for total) is the sum of the windows on the left and right sides (see Section 2.3 for how we obtain these counts). Under ILS, the expected number of windows in each half of the plot is simply half of the total number of windows (i.e., $0.5n_T$).

The deviation of D_{LR} from equality can be assessed using a G-test, although this approach assumes independence between trees. In practice, this assumption is rarely met in tree sequences, as adjacent trees tend to be correlated in structure owing to the gradual breakdown of genealogical relationships by recombination. A simple way to mitigate this non-independence is to thin the tree sequence by retaining every ith tree. We provide some practical advice on choosing an appropriate down-sampling interval in Section 3.2.

In addition to a genome-wide estimate, symmetry can also be quantified for any arbitrary area of the distribution, provided that analogous sections (mirrored across the main median) from both sides of the ternary plot are compared. This allows the user to examine whether the asymmetry varies as a function of the overall value of the topology weights. This is achieved using the lattice of sub-triangles and calculating D_{LR} based on the counts of trees in the analogous left- and right-sided sub-triangles. This makes it possible to understand how the degree of symmetry varies as a function of the strength of the discordance.

3 Running TwisstNtern

3.1 Tree file format

TwisstNtern accepts several input formats, including TreeSequence files from *tskit* (including `.trees` and `.ts`) [Kel+19], as well as Newick (`.newick`, `.tree`) and Nexus format (`.nexus`) for trees built in arbitrary windows. Output from *Relate* [Spe+19] and *Singer* [DNS24] can be used, but must first be converted using a formatting script that we provide on GitHub. *TwisstNtern* also accepts a CSV input file (`.csv`) containing pre-computed topology weights from previous *Twisst* analyzes. All file types may also be supplied in gzip-compressed form (`.gz`) and are automatically decompressed during loading.

3.2 Choosing values for key parameters

It is important to consider the resolution of the analysis, which refers to the granularity of the lattice of sub-triangles that divides the ternary plot. This can be adjusted with the `--granularity` flag. We provide three preset resolutions that should be sufficient for most users: coarse ($\alpha = 0.25$), fine ($\alpha = 0.1$), or superfine ($\alpha = 0.05$). A custom grid size can be used by specifying a value of α with $1/\alpha$ being an even integer.

The most meaningful granularity will depend on the details of the dataset. Where the genome has been sectioned into blocks of arbitrary size (e.g., non-overlapping 5 kb windows) or SNP number (e.g., non-overlapping 50 SNP windows), the fine option is a good place to start. Tools like *Relate* [Ras+14], *tsinfer* [Kel+19], and *Singer* [DNS24] can produce very large tree sequences containing hundreds of thousands or even millions of trees, so the superfine option may be more suitable. Studies with reduced representation datasets may have fewer loci than those based on whole genome sequencing, so coarse may be a better option. Similarly, in cases where the distribution of weights may be restricted to a fairly small section of the ternary plot, superfine may be more appropriate. In summary, the fine granularity is a good place to start, but use common sense to choose a granularity that gives the best balance between resolution (i.e., good subdivision of the occupied part of the ternary plot) and power (i.e., reasonable counts of points within the sub-triangles) for your dataset.

If the user is interested in reporting a p-value for the significance of the D_{LR} statistic, then it is a good idea to subsample the tree sequence for this purpose as adjacent trees tend to be correlated in structure owing to the gradual breakdown of genealogical relationships by recombination. This can be achieved using the `--downsample ``N+i''` flag, which retains only every Nth tree starting from the ith tree. The appropriate scale for down-sampling can be roughly estimated from the decay of linkage disequilibrium and the average treespan.

3.3 Other optionality

Many additional options can be used to control the visualization, axis order, and verbosity of the output. The `--axis` flag defines which topology corresponds to each vertex of the ternary plot when using a `.csv` file containing pre-computed topology weights. To set the topological axis when running a tree file, you can specify which axis corresponds to which topology using the `--topology_mapping` flag. For example, over a tree with populations *O*, *p1*, *p2*, and *p3*, one can determine `topology_mapping='T1="(0,(p3,(p1,p2)))"; T2="(0,(p1,(p2,p3)))"; T3="(0,(p2,(p1,p3)))";`. Enabling `--verbose` provides detailed runtime information and logs all parameters used in the analysis, facilitating reproducibility. The `--colormap` flag customizes the color scheme of both the ternary heatmap and the radcount plots, which display the point density. For convenience, the flag `-o` specifies the name of the output directory to be created, where all results will be stored. See GitHub for a full description of the optional functions.

3.4 Output

Each run produces a structured output directory containing both tabular and graphical results. If no output directory is specified, *TwisstNtern* automatically generates a timestamped folder (e.g., `Results_2025-07-03_14-30-25/`) to ensure organized and reproducible analyses. The main outputs include:

i. `<run_id>_topology_weights.csv`: A file that contains the raw topology weights for trees in the sequence computed using *Twisst*;

ii. `<run_id>_triangle_analysis[granularity].csv`: Contains a table listing all D_{LR} values, their corresponding G-test scores and counts for both the overall data set and each individual sub-triangle;

iii. `<run_id>_fundamental_asymmetry.pdf`: A figure that illustrates the strength of the genome-wide asymmetry between the two main sub-triangles;

iv. `<run_id>_analysis_granularity<value>.pdf`: A figure displaying the main ternary plot, which illustrates the distribution of topology weights across the dataset. `<run_id>_heatmap.pdf` and `<run_id>_radcount.pdf`: Two heatmap visualizations of the same dataset of interest.

4 Other new features

4.1 Exploring with TwisstNtern simulate

We have found it helpful to use simulations to help us build an intuition for how different processes shape the ternary distribution of weights. *TwisstNtern* includes the module *TwisstNtern-simulate*, a command-line tool for simulating and analyzing genealogical data under four-population demographic models using *msprime* [Bau+22]. The simulation settings and the values of the demographic parameters are managed through a YAML configuration file or by specifying the parameters directly on the command line.

TwisstNtern-simulate supports two simulation modes: chromosome mode and locus mode. In chromosome mode, *msprime* is used to simulate a continuous stretch of sequence along a single chromosome with recombination, generating a series of correlated genealogies that reflect linkage. In locus mode, *msprime* simulates many independent, non-recombining loci under a shared demographic model, producing unlinked genealogies. The choice between these modes depends on the research question, with chromosome mode capturing how genealogies change along real genomes, and replicate mode providing statistical replication across independent regions. Once tree sequences are simulated, the software automatically computes topology weights, generates ternary plots, and conducts tests of asymmetry. It also provides options for downsampling the tree sequence (as described in Section 3.2), and color customization of the graphs.

4.2 Comparing ternary distributions

It is also possible to compare two different ternary distributions, which is useful for quantifying how well simulated distributions fit empirical datasets (Fig. 4). This is accomplished by treating our plots as histograms, using the counts of trees across the lattice of sub-triangles, which enables direct comparison of analogous regions in two distinct ternary distributions. To obtain an overall measure of the two different distributions, we first find the residuals for each sub-triangle by computing the difference in the number of trees observed in that defined area between the two distributions being compared (Fig. 4). A measure of the distance, L_2, is then obtained for each sub-triangle as the square of the residuals (Fig. 4). The total distance between two distributions can be quantified as the sum of the squared residuals.

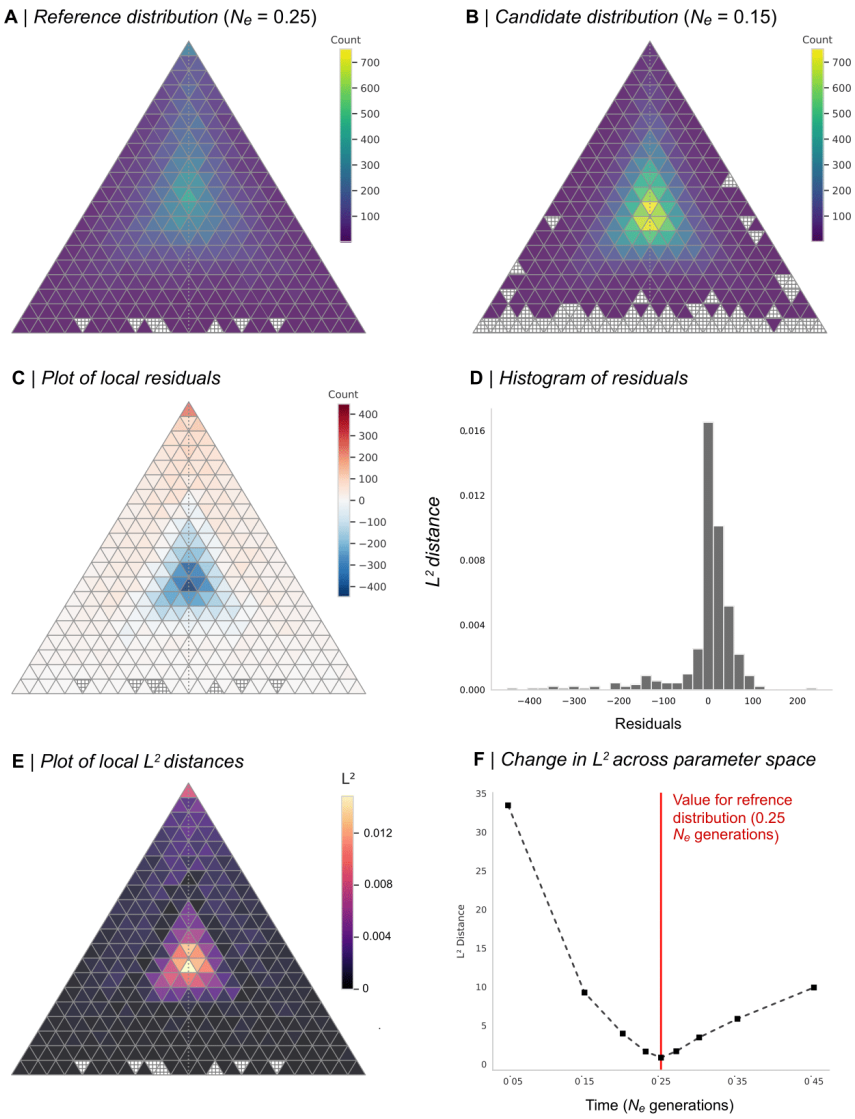

Figure 4: **Comparing ternary distributions using the L_2 distance.** We compare several simulated datasets that differ only by the amount of time between population splits (time scaled in N_e generations). (A) Ternary distribution for our reference dataset, where $T = 0.25N_e$ generations. (B) Distribution for one of nine candidate datasets, in this case where $T = 0.15N_e$. (C) Map and histogram (D) of the residuals of the counts of trees in each sub-triangle, determined by subtracting the two distributions. (E) Distribution of L_2 distances, calculated as the square of the residuals. (F) The overall L_2 distance, calculated as the sum of the squared residuals, between each of the candidate simulations and the reference simulation. The red line indicates the value of T used in the reference simulation. Note that the lowest L_2 distance was found between the pair of simulations where $T = 0.25N_e$.

We demonstrate how this approach might be useful for inferring demographic parameters associated with four-population histories. We begin by using *TwisstNtern-simulate* to generate a reference dataset consisting of $10,000$ loci, each evolving under a scenario where three demographic splits are evenly separated by $0.25N_e$ generations. This data set serves as our "real" data, with its parameters defining the "true model". Next, we simulate nine alternate datasets following similar demographic histories, but varying only the intervals between the splits, from $0.05N_e$ generations to $0.45N_e$ generations. For each alternate scenario, we compute

the L_2 distance from the reference dataset (see Fig. 4). The scenario with the lowest L_2 distance also features splits in every $0.25N_e$ generations, demonstrating that the method identifies the true simulation parameter used to create the reference data set.

We provide a standalone Python script for comparing two distributions called `TwisstNtern_compare.py`, and are currently working on a more comprehensive demographic inference scheme (both for identifying the most likely model and for parameter estimation) based on large grid searches and Approximate Bayesian Computation.

5 Accessing Program

TwisstNtern is available on GitHub at https://github.com/HilaLifchitz/twisstntern_v2. Detailed instructions for installing and running it are available via the README. If you have any suggestions for improving *TwisstNtern* or discover any issues, please contact the corresponding authors.

Acknowledgments

We thank Roger K. Butlin and Nicholas H. Barton for feedback during development.

References

[Bau+22] Franz Baumdicker et al. "Efficient ancestry and mutation simulation with msprime 1.0". eng. In: *Genetics* 220.3 (Mar. 2022), iyab229. ISSN: 1943-2631. DOI: 10.1093/genetics/iyab229.

[DNS24] Yun Deng, Rasmus Nielsen, and Yun S. Song. *Robust and Accurate Bayesian Inference of Genome-Wide Genealogies for Large Samples*. en. Pages: 2024.03.16.585351 Section: New Results. Mar. 2024. DOI: 10.1101/2024.03.16.585351. URL: https://www.biorxiv.org/content/10.1101/2024.03.16.585351v1 (visited on 11/06/2025).

[Gre+06] Richard E. Green et al. "Analysis of one million base pairs of Neanderthal DNA". eng. In: *Nature* 444.7117 (Nov. 2006), pp. 330–336. ISSN: 1476-4687. DOI: 10.1038/nature05336.

[Kel+19] Jerome Kelleher et al. "Inferring whole-genome histories in large population datasets". eng. In: *Nature Genetics* 51.9 (Sept. 2019), pp. 1330–1338. ISSN: 1546-1718. DOI: 10.1038/s41588-019-0483-y.

[MV17] Simon H Martin and Steven M Van Belleghem. "Exploring Evolutionary Relationships Across the Genome Using Topology Weighting". In: *Genetics* 206.1 (May 2017), pp. 429–438. ISSN: 1943-2631. DOI: 10.1534/genetics.116.194720. URL: https://doi.org/10.1534/genetics.116.194720 (visited on 06/26/2025).

[Pal+] Arka Pal et al. "Genealogical Analysis of Replicate Flower Colour Hybrid Zones in Antirrhinum". en. In: *Molecular Ecology* (), e70067. ISSN: 1365-294X. DOI: 10.1111/mec.70067. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/mec.70067 (visited on 11/06/2025).

[Ras+14] Matthew D. Rasmussen et al. "Genome-Wide Inference of Ancestral Recombination Graphs". en. In: *PLOS Genetics* 10.5 (May 2014). Publisher: Public Library of Science, e1004342. ISSN: 1553-7404. DOI: 10.1371/journal.pgen.1004342. URL: https://journals.plos.org/plosgenetics/article?id=10.1371/journal.pgen.1004342 (visited on 06/04/2024).

[RTK20] Peter Ralph, Kevin Thornton, and Jerome Kelleher. "Efficiently Summarizing Relationships in Large Samples: A General Duality Between Statistics of Genealogies and Genomes". In: *Genetics* 215.3 (July 2020), pp. 779–797. ISSN: 0016-6731. DOI: 10.1534/genetics.120.303253. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC7337078/ (visited on 11/06/2025).

[Shi+23] Daria Shipilina et al. "On the origin and structure of haplotype blocks". eng. In: *Molecular Ecology* 32.6 (Mar. 2023), pp. 1441–1457. ISSN: 1365-294X. DOI: 10.1111/mec.16793.

[Spe+19] Leo Speidel et al. "A method for genome-wide genealogy estimation for thousands of samples". In: *Nature genetics* 51.9 (Sept. 2019), pp. 1321–1329. ISSN: 1061-4036. DOI: 10.1038/s41588-019-0484-x. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC7610517/ (visited on 11/06/2025).

[Sta+24] Sean Stankowski et al. "The genetic basis of a recent transition to live-bearing in marine snails". eng. In: *Science (New York, N.Y.)* 383.6678 (Jan. 2024), pp. 114–119. ISSN: 1095-9203. DOI: 10.1126/science.adi2982.

[Wak16] J. Wakely. *Coalescent Theory: An Introduction*. Macmillan Learning, 2016. ISBN: 978-0-9747077-5-4. URL: https://books.google.at/books?id=x30RAgAACAAJ.