Supplementary File S1 for:

OptM: estimating the optimal number of migration edges on population trees using *Treemix*

Robert R. Fitak (Robert.Fitak@ucf.edu)
Department of Biology, Genomics and Bioinformatics Cluster, University of Central Florida

Methods

Mathematical derivation of Δm

The parameter $L(m)$ corresponds with the composite likelihood of the model fit using *TREEMIX* for m migration edges. $L(m)$ is thus equivalent to $L(\widehat{W}|W)$ from equation 28 in Pickrell and Pritchard (2012):

$$L(\widehat{W}|W) = \prod_{i=1}^{m}\prod_{j=1}^{m} N(\widehat{W_{ij}}|G, \widehat{\sigma}_{ij}^{2}) \tag{1}$$

First, the mean $L(m)$ is estimated across N iterations for successive values of m where $m \geq 0$:

$$L(m) = \frac{1}{N}\sum_{i=1}^{N} L(m_i) \tag{2}$$

Then, following the methodology outlined by Evanno et al (2005), the mean difference between successive composite likelihoods, or the first order rate of change $L'(m)$, is calculated as:

$$L\prime(m) = L(m) - L(m-1) \tag{3}$$

Next, the second order rate of change in $L(m)$ with respect to m is calculated:

$$L\prime\prime(m) = |L\prime(m+1) - L\prime(m)| \tag{4}$$

or by simple algebra:

$$\begin{gathered} L\prime\prime(m) = |L(m+1) - L(m) - L(m) + L(m-1)| \\ L\prime\prime(m) = |L(m+1) - 2 * L(m) + L(m-1)| \end{gathered} \tag{5}$$

In the last step, Δm is calculated by normalizing $L\prime\prime(m)$ by the standard deviation in $L(m)$, $\sigma_{L(m)}$:

$$\Delta m = \frac{L\prime\prime(m)}{\sigma_{L(m)}} \tag{6}$$

Simulated datasets

The simulated datasets were generated using ARGON v0.1 (Palamara 2016). ARGON simulates the discrete time Wright Fisher process backwards in time quickly for whole-genome sized datasets for arbitrary, user-defined demographic histories. The population graph without any migration events was identical to that in Pickrell and Pritchard (2012) which was based upon recreating patterns of diversity and linkage disequilibrium consistent with that of SNP genotype data for many modern human datasets (DeGiorgio et al. 2009) (**Fig. S1**). The parameters across simulations included:

- a serial bottleneck model composed of 20 populations
- each population had a current effective population size of 10000
- each serial bottleneck event originated from 250 individuals ever 100 generations
- a mutation rate of 10^{-8} substitutions site^{-1} generation^{-1}
- a recombination rate of 10^{-8} recombinations site^{-1} generation^{-1}
- recombinations must occur at least every 100 bp apart
- a sequence (chromosome) length of 250 megabases
- 60 chromosomes (30 diploid individuals) sampled from each population
- All populations shared a common ancestor 2000 generations in the past

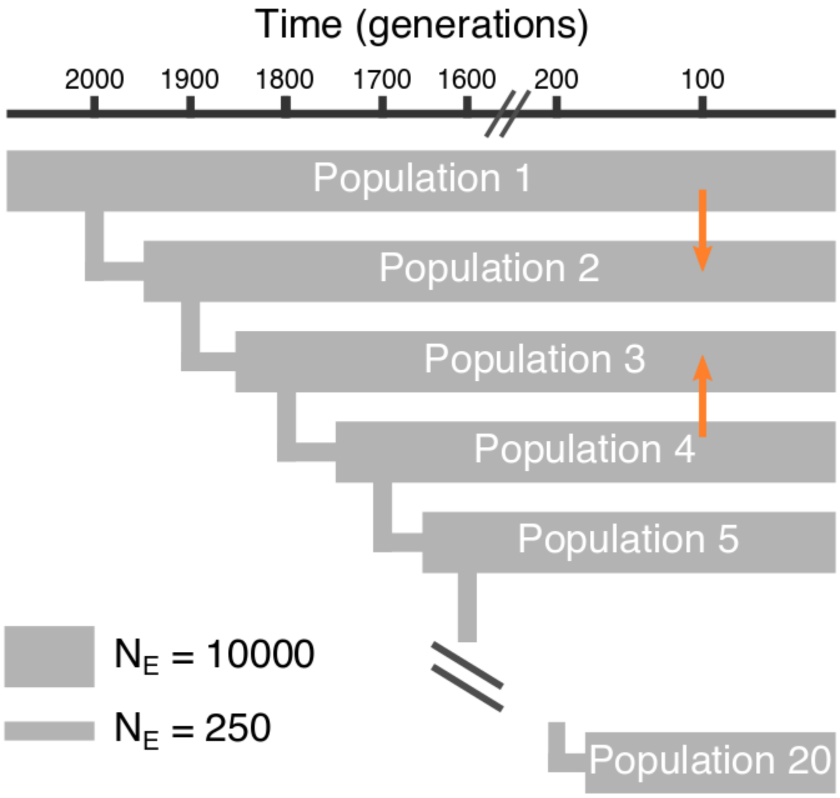


Fig. S1 Diagram of the serial population bottleneck model used for the simulated datasets. The model is as described according to Pickrell and Pritchard (2012) and (DeGiorgio et al. 2009). Two hypothetical migration edges, from population 1 into 2, and 4 into 3 are shown (orange arrows)

Either 1, 3, 5, or 8 migrations events we included in each simulation. The source and recipient population for each migration event were selected at random without replacement, and the recipient population received 30% of its genetic ancestry from the source population. All simulation events occurred 100 generations in the past. The 8 randomly selected migration edges were as follows:

1. 13 ---> 5
2. 4 ---> 12
3. 7 ---> 16
4. 6 ---> 3
5. 9 ---> 17
6. 15 ---> 1
7. 14 ---> 8
8. 19 ---> 10

An example run of ARGON v0.1 for $m = 8$. The configuration file `M8_Argon.txt` is available as a Supplementary data file.

```
# Set seed
SEED=$RANDOM
echo "$SEED"

# Run ARGON v0.1 for 8 migration edges
java -jar ARGON.0.1.jar \
   -N M8_Argon.txt \
   -pop 20 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 \
   -size 250 \
   -rec 1E-8 \
   -mut 1E-8 \
   -out M8 \
   -gz \
   -len 100 \
   -seed $SEED
```

The output from ARGON v0.1 is a compressed VCF file (e.g. above: `M8.vcf.gz`). The VCF files were converted to the PED/MAP format utilized by PLINK v1.07 (Purcell et al. 2007) with VCFTOOLS v0.1.13 (Danecek et al. 2011). This step also removed any SNP loci with a minimum allele frequency (MAF) < 0.05.

```
# Convert VCF to PED/MAP and remove maf<0.05
zcat M8.vcf.gz | \
   vcf-sort | \
   vcftools \
      --gzvcf - \
      --maf 0.05 \
      --plink \
      --out M8
```

Next, the FID and IID columns in the PLINK-formatted file were corrected so the FID column is the population (1-20), and the IID column is the individual number (1-30). A cluster file, utilized by PLINK to calculate stratified allele frequencies, was also created. The cluster file is a 3 column list of FID, IID, and the cluster (a.k.a. population) to which the sample belongs. The cluster here is the same as the FID.

```
# Change FID and IID encoding
cut -f2- M8.ped | \
   tr "_" "\t" > tmp
mv tmp M8.ped

# Make cluster file
for pop in {1..20}
   do
   for i in {1..30}
      do
      echo "$pop $i $pop" >> pops.cluster
   done
done
```

Once the PLINK PED file and the cluster file were prepared, PLINK was used to make allele counts within each cluster (population). The resulting stratified allele counts file was compressed and converted to a TREEMIX input file using the python script `plink2treemix.py`. The script plink2treemix.py is distributed along with TREEMIX v1.13 and can be downloaded from the link provided.

```
# Make a stratified allele frequency (counts) file
plink \
   --file M8 \
   --noweb \
   --freq \
   --within pops.cluster \
   --out M8

# Compress the file
gzip M8.frq.strat

# Convert to TREEMIX input file
plink2treemix.py M8.frq.strat.gz M8.treemix.gz
```

The last step was to run TREEMIX. TREEMIX was run for 10 replicates each for $m = [1..10]$. To avoid converging on the same composite likelihood for each replicate, the number of SNPs per window (`-k`) was varied across runs from 100-1000 in 50 SNP increments. A global set of rearrangements (`-global`) was also included.

```
# Run 100 runs of treemix
   # m = number of migration edges
   # i = number of replicates for each value of m
   # k = number of SNPs per window
   # s = random seed

for m in {1..10}
   do
   for i in {1..10}
      do

      # Generate random seed
      s=$RANDOM
      echo "Random seed = ${s}"

      # Generate random k between 100 and 1000 in 50 SNP increments
      k=$(seq 100 50 1000 | shuf -n 1)

      treemix \
         -i M8.treemix.gz \
         -o M8.${i}.${m} \
         -global \
         -m ${m} \
         -k ${k} \
         -seed ${s}
      done
```

```
done
```

Once completed, the TREEMIX output files were analyzed with OptM v0.1.5 using default parameters and plotted for *m* = 1, 3, 5, 8 simulated models (**Fig. S2**). All analyses were performed in R v3.4.3.

```
# Install and Load OptM v0.3
install.packages("OptM")
library(OptM)

# Load treemix output files
   # Example shown below: all treemix output for m=8 are inside a folder called "M8"
data = optM("M8")

# Plot results
plot_optM(data)
```

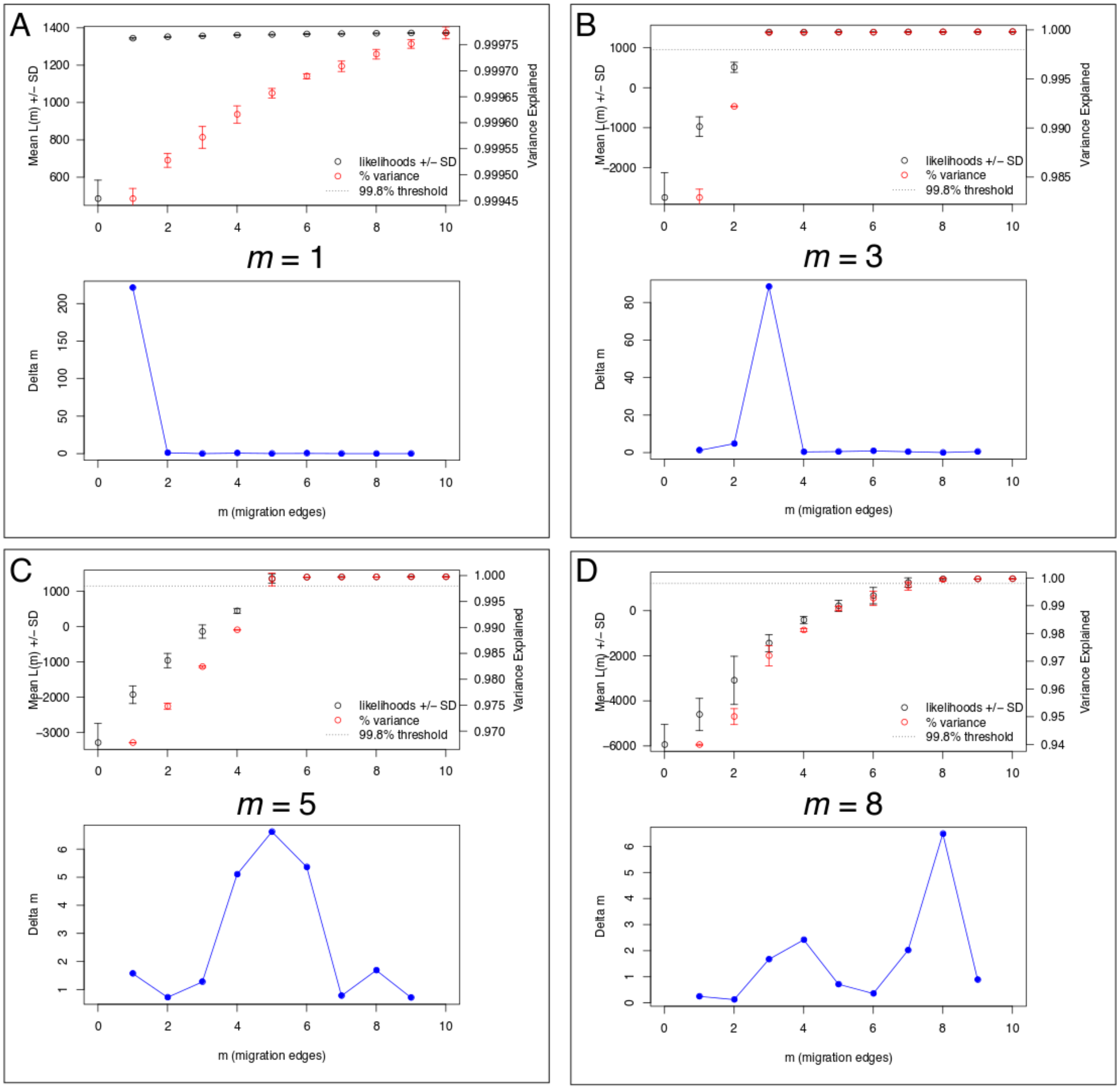


Fig. S2 The output from OptM for 1 (A), 3 (B), 5 (C), and 8 (D) simulated migration edges. The upper panels in A-D show the mean composite likelihoods (black circles, left axis) and percent variance (red circles, right axis) explained by the model with m edges. The horizontal dashed line shows the 99.8% variation threshold recommended by Pickrell and Pritchard (2012) as the cutoff for adding migration edges. In each case, OptM recovers the correct, or optimal, number of migration edges as indicated by the peak in Δm in the lower panels of A-D (blue lines). Notice that for $m = 1$, even the null model ($m = 0$) exceeds the 99.8% threshold

---

Empirical domestic dog and wolf dataset

As an empirical example, OptM was applied to a dataset composed of 532 domestic dogs from 48 breeds and 15 wolves genotyped on the CanineHD BeadChip (Lequarré et al. 2011; Vaysse et al. 2011). A total of 15 breeds were removed because the number of genotyped individuals was less than eight - and thus susceptible to high variance in estimates of allele frequencies. The SNPs were filtered to include only autosomal loci with a minimum allele frequency ≥ 0.05 and a genotyping rate ≥ 0.9 using PLINK. Individuals with a genotyping rate ≤ 0.9 were omitted from the analysis. Links to the dataset are shown in the code below.

```
# Download Lupa dataset, PLINK PED/MAP format
curl \
    -o lupa.map \
    http://dogs.genouest.org/SWEEP.dir/HDselection_updated_trees.map
curl \
    -o lupa.ped \
    http://dogs.genouest.org/SWEEP.dir/HDselection_updated_trees.ped

# Get sex chromosome SNPs (X = 39, Y = 40)
grep \
    -E "^39|^40" lupa.map | \
     cut -f2 > XY.exclude

# Make a list of breeds to remove: n<8
echo "ASh
Chi
CKC
CWD
DAL
ECS
ESS
FcR
Hus
LMu
Mop
Sam
Sar
Scn
Ter" > breeds-too-few.txt

# Grab the FID and IID for these breeds form the PED file


grep \
    -F \
    -f breeds-too-few.txt lupa.ped | \
    cut -d" " -f1-2 > IDs-remove.txt

# Clean data in plink
plink \
    --noweb \
    --nonfounders \
    --dog \
    --file lupa \
    --remove IDs-remove.txt \
    --exclude XY.exclude \
    --maf 0.05 \
    --geno 0.1 \
    --mind 0.1 \
    --make-bed \
    --out lupa.clean
```

Here is the output from PLINK:

```
@----------------------------------------------------------@
|        PLINK!       |     v1.07      |   10/Aug/2009     |
|----------------------------------------------------------|
|  (C) 2009 Shaun Purcell, GNU General Public License, v2  |
|----------------------------------------------------------|
|  For documentation, citation & bug-report instructions:  |
|        http://pngu.mgh.harvard.edu/purcell/plink/        |
@----------------------------------------------------------@

Skipping web check... [ --noweb ]
Writing this text to log file [ lupa.clean.log ]
Analysis started: Sat Feb 23 14:06:27 2019

Options in effect:
  --noweb
  --nonfounders
  --dog
  --file lupa
  --remove IDs-remove.txt
  --exclude XY.exclude
  --maf 0.05
  --geno 0.1
  --mind 0.1
  --make-bed
  --out lupa.clean

** For gPLINK compatibility, do not use '.' in --out **
174810 (of 174810) markers to be included from [ lupa.map ]
Warning, found 547 individuals with ambiguous sex codes
Writing list of these individuals to [ lupa.clean.nosex ]
547 individuals read from [ lupa.ped ]
0 individuals with nonmissing phenotypes
Assuming a disease phenotype (1=unaff, 2=aff, 0=miss)
Missing phenotype value is also -9
```

```
0 cases, 0 controls and 547 missing
0 males, 0 females, and 547 of unspecified sex
Reading list of SNPs to exclude [ XY.exclude ] ... 5744 read
Reading individuals to remove [ IDs-remove.txt ] ... 36 read
36 individuals removed with --remove option
Before frequency and genotyping pruning, there are 169066 SNPs
511 founders and 0 non-founders found
Writing list of removed individuals to [ lupa.clean.irem ]
3 of 511 individuals removed for low genotyping ( MIND > 0.1 )
Total genotyping rate in remaining individuals is 0.983518
4151 SNPs failed missingness test ( GENO > 0.1 )
28349 SNPs failed frequency test ( MAF < 0.05 )
After frequency and genotyping pruning, there are 138306 SNPs
After filtering, 0 cases, 0 controls and 508 missing
After filtering, 0 males, 0 females, and 508 of unspecified sex
Writing pedigree information to [ lupa.clean.fam ]
Writing map (extended format) information to [ lupa.clean.bim ]
Writing genotype bitfile to [ lupa.clean.bed ]
Using (default) SNP-major mode

Analysis finished: Sat Feb 23 14:07:08 2019
```

Next, a cluster file which lists each individual and the cluster (population) assignment was generated for use with PLINK. Then, the TREEMIX input file was created as described above for the simulated datasets.

```
# Clean FAM file
cut \
   -d" " \
   -f1 lupa.clean.fam | \
   sed "s/_.*//g" | \
   paste \
      -d" " \
      - \
      <(cut -d" " -f2- lupa.clean.fam) > new.fam
mv new.fam lupa.clean.fam

# Prepare cluster file
cut -d" " -f1-2 lupa.clean.fam | \
   paste -d" " - \
   <(cut -d" " -f1 lupa.clean.fam) > within.txt

# Prepare the allele counts file
plink \
   --noweb \
   --dog \
   --nonfounders \
   --bfile lupa.clean \
   --freq \
   --within within.txt \
   --out Lupa.treemix

# Compress the stratified allele counts file
gzip Lupa.treemix.frq.strat

# Download the script plink2treemix.py
wget https://bitbucket.org/nygcresearch/treemix/downloads/plink2treemix.py
chmod 770 plink2treemix.py

# Convert to a TREEMIX input file
./plink2treemix.py Lupa.treemix.frq.strat.gz Lupa.treemix.gz
```

Finally, TREEMIX v1.13 was run on the dog dataset for $m = 1 - 40$ with 10 iterations for each value of m. A window (`-k`) of 500 SNPs was used, along with a global rearrangement (`-global`). The optimal number of migration edges was estimated using OptM. Although a `for` loop is shown below, it is recommended to submit these as separate jobs to a compute cluster to reduce the time taken for analysis.

```
# Run treemix 10 times for m from 1-40 (400 runs)
   # m = number of migration edges
   # i = number of replicates for each value of m

for m in {1..40}
   do
   for i in {1..10}
      do

      # Generate random seed
      s=$RANDOM
      echo "Random seed = ${s}"

      # Run treemix
      treemix \
         -i Lupa.treemix.gz \
         -o Lupa.${i}.${m} \
         -global \
         -m ${m} \
         -k 500 \
         -seed ${s}
      done
done
```

```
# in R v3.4.3
# Load OptM
library(OptM)

# Run OptM (all files are in a folder called "LUPA")
Lupa.out = optM("LUPA")

# Plot Results
plot_optM(Lupa.out)

# Load plotting functions from Treemix
source("/Path/to/treemix-1.13/src/plotting_funcs.R")

# Plot tree with migration edges for one iteration of m=5 (Fig. 3 in main text)
plot_tree("Lupa.1.5")
```

Fitting threshold models

OptM is also integrated with the SiZer v0.1-5 package (Sonderegger et al. 2009) for fitting various ecological threshold models (see **Fig. S3 and Table S1** below for models M1, M3, M5, and M8 simulated above). These models, such as a 'piecewise linear' (PL) and 'bent cable' (BC) models, can be used to estimate the threshold, or change point, of the response as a function of an independent variable. On one side of a change point, small increases in the independent variable produce negligible effects on the response, whereas on the other side of the change point, small increases in the independent variable can produce substantial effects on the response. In the context of estimating the optimal value of m, the goal is to identify the change point where an increase in m no longer produces a worthwhile increase in $L(m)$. Both the PL and BC models are parametric models that fit two linear relationships connected by either an abrupt or a quadratic bend, respectively. OptM also fits a simple exponential model as well as a non-linear least squares exponential model primarily for visualization purposes. All parametric models are compared with the Akaike information criterion (AIC; Akaike 1973). Finally, the non-parametric “significant zero crossings” method (SiZer; Sonderegger et al. 2009), which predicts change points by examining the first and second derivatives of non-parametric smoothing functions, is also available for comparison purposes if of interest to the user but is not shown below. Please refer to the OptM manual.

```
# in R v3.4.3
# Load OptM
library(OptM)

# Load treemix output files, from simulated datasets
   # Example shown below: all treemix output for m=8 are inside a folder called "M8"
data = optM("M8", method = "linear")

# Print the model comparison (AIC) output
data$out

# Plot results
plot_optM(data, method = "linear")
```

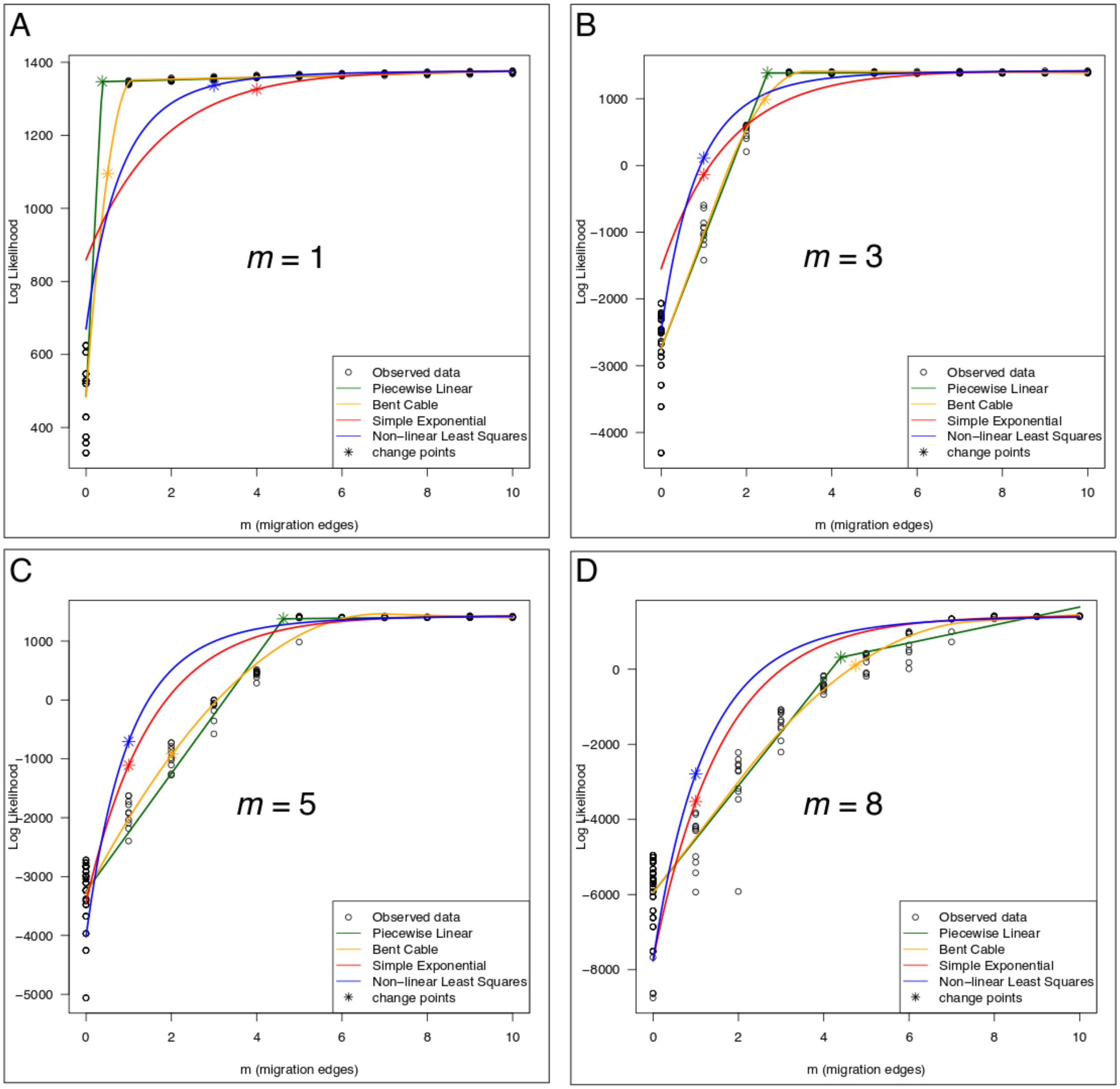


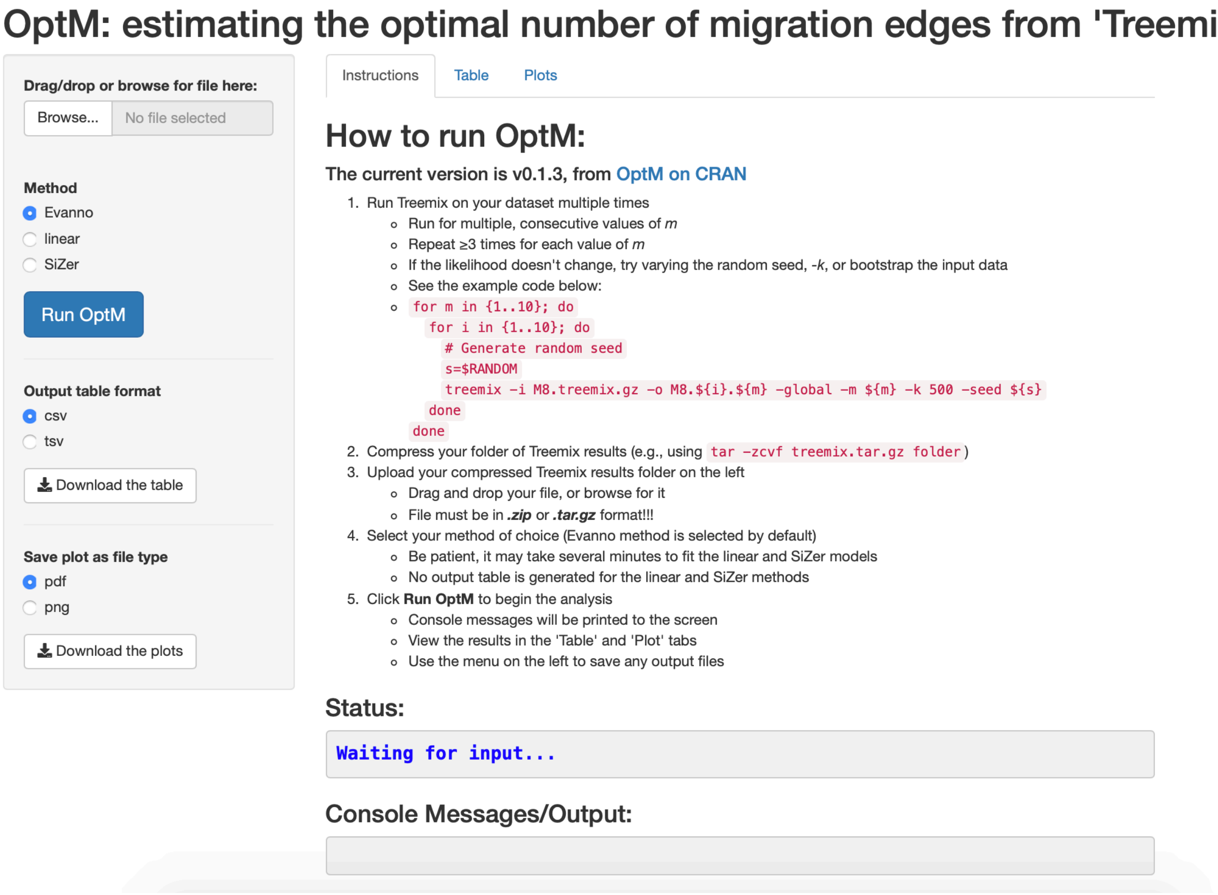
Fig. S3 The output from various ecological threshold models fit using OptM for 1 (A), 3 (B), 5 (C), and 8 (D) simulated migration edges. Each panel shows the observed composite likelihoods (black circles) for each TREEMIX run, and the four models fit (colored lines, see legend). The change points predicted by each model are shown as colored stars. Change points for the simple exponential and non-linear least squares models should not be considered accurate. For a complete description of the threshold models shown, see Sonderegger et al. 2009

Model	df	AIC	ΔAIC	Change Point
M1				
Non-linear Least Squares	3	403.6	0	3.0
Simple Exponential	3	545.9	142.3	4.0
Piecewise Linear	5	2278.0	1874.4	0.38
Bent Cable	6	2279.9	1876.3	0.51
M3				
Non-linear Least Squares	3	464.9	0	1.0
Simple Exponential	3	575.7	110.8	1.0
Piecewise Linear	5	3016.4	2551.4	2.5
Bent Cable	6	3018.0	2553.0	2.4
M5				
Simple Exponential	3	455.9	0	1.0
Non-linear Least Squares	3	456.0	0.060	1.0
Bent Cable	6	2980.5	2524.6	2.0
Piecewise Linear	5	2989.8	2533.9	4.6
M8				
Simple Exponential	3	418.0	0	1.0
Non-linear Least Squares	3	503.7	85.7	1.0
Bent Cable	6	3204.8	2786.8	4.7
Piecewise Linear	5	3206.9	2788.9	4.4

Table S1 The output from various ecological threshold models fit using OptM for 1 (M1), 3 (M3), 5 (M5), and 8 (M8) simulated migration edges. Models are ordered by the ΔAIC. df = degrees of freedom, AIC = Akaike information criterion. Change points for the simple exponential and non-linear least squares models should not be considered accurate. For a complete description of the threshold models shown, see Sonderegger et al. 2009

OptM Web Application

To make OptM as easy as possible to implement, especially for those without essential programming skills in R, we have also generated a web interface to *OptM* (**Fig. S4**; https://rfitak.shinyapps.io/OptM/). The web application allows the user to quickly load a zipped folder of TREEMIX v1.13 results, including by simple drag-and-drop, select a few options if necessary using check boxes, then run the analysis. The output from OptM will be quickly generated and viewed in separate *Table* and *Plots* tabs. The results can be downloaded in multiple formats with the simple click of a mouse.



OptM: estimating the optimal number of migration edges from 'Treemix'

Drag/drop or browse for file here:

Browse... No file selected

Method

● Evanno
○ linear
○ SiZer

Run OptM

Output table format

● csv
○ tsv

Download the table

Save plot as file type

● pdf
○ png

Download the plots

Instructions | Table | Plots

How to run OptM:

The current version is v0.1.3, from OptM on CRAN

1. Run Treemix on your dataset multiple times
   - Run for multiple, consecutive values of *m*
   - Repeat ≥3 times for each value of *m*
   - If the likelihood doesn't change, try varying the random seed, -*k*, or bootstrap the input data
   - See the example code below:

   -
     ```
     for m in {1..10}; do
       for i in {1..10}; do
         # Generate random seed
         s=$RANDOM
         treemix -i M8.treemix.gz -o M8.${i}.${m} -global -m ${m} -k 500 -seed ${s}
       done
     done
     ```
2. Compress your folder of Treemix results (e.g., using `tar -zcvf treemix.tar.gz folder`)
3. Upload your compressed Treemix results folder on the left
   - Drag and drop your file, or browse for it
   - File must be in ***.zip*** or ***.tar.gz*** format!!!
4. Select your method of choice (Evanno method is selected by default)
   - Be patient, it may take several minutes to fit the linear and SiZer models
   - No output table is generated for the linear and SiZer methods
5. Click **Run OptM** to begin the analysis
   - Console messages will be printed to the screen
   - View the results in the 'Table' and 'Plot' tabs
   - Use the menu on the left to save any output files

Status:

```
Waiting for input...
```

Console Messages/Output:

Fig. S4 The web implementation of *OptM* built using R shiny.

References

1. Akaike,H. (1973) Information theory and an extension of the maximum likelihood principle. In: Second International Symposium on Information Theory. (Petrov BN, Csaki F, eds), pp. 267-281. Akadémiai Kiadó, Budapest.
2. Danecek,P. et al. (2011) The variant call format and VCFtools. *Bioinformatics*, **27**, 2156–2158.
3. DeGiorgio,M., et al. (2009) Explaining worldwide patterns of human genetic variation using a coalescent-based serial founder model of migration outward from Africa. *Proc. Natl. Acad. Sci. USA*, **106**, 16057.
4. Evanno,G. et al. (2005) Detecting the number of clusters of individuals using the software structure: a

simulation study. *Mol. Ecol.*, **14**, 2611-2620.

5. Lee,R. (1897) A history and description of the modern dogs of Great Britain and Ireland. Sporting division vol 1. H. Cox, London.
6. Lee,R. (1903) A history and description of the modern dogs of Great Britain and Ireland. The terriers vol 3. H. Cox, London.
7. Lequarré,A. et al. (2011) LUPA: A European initiative taking advantage of the canine genome architecture for unravelling complex disorders in both human and dogs. *Vet. J.*, **189**, 155-159.
8. Palamara,P. (2016) ARGON: fast, whole-genome simulation of the discrete time Wright-fisher process. *Bioinformatics*, **32**, 3032-3034.
9. Parker,H., et al. (2017) Genomic analyses reveal the influence of geographic origin, migration, and hybridization on modern dog breed development. *Cell Rep.*, **19**, 697-708.
10. Pickrell,J. and Pritchard J. (2012) Inference of population splits and mixtures from genome-wide allele frequency data. *PLoS Genet.*, **8**, e1002967.
11. Pilot,M. et al. (2015) On the origin of mongrels: evolutionary history of free-breeding dogs in Eurasia. *Proc. Biol. Sci.*, **282**, 20152189
12. Sonderegger,D. et al. (2009) Using SiZer to detect thresholds in ecological data. *Front. Ecol. Environ.*, **7**, 190-195.
13. Vaysse,A. et al. (2011) Identification of genomic regions associated with phenotypic variation between dog breeds using selection mapping. *PLoS Genet.*, **7**, e1002316.
14. Wang,G. et al. (2016) Out of southern East Asia: the natural history of domestic dogs across the world. *Cell Res.*, **26**, 21-33