slendr: a framework for spatio-temporal population genomic simulations on geographic landscapes

Martin Petr[1], Benjamin C. Haller[2], Peter L. Ralph[3], Fernando Racimo[1]

We present a new R package, *slendr* (slendr.net), designed for declarative, visually-focused encoding of complex spatio-temporal population models on real and abstract geographic landscapes. *slendr* uses a tailor-made SLiM script (messerlab.org/slim) as a simulation back end bundled with the package, and saves spatially-annotated tree sequences as its output. Furthermore, *slendr* also provides a new way to simulate data from traditional, random-mating demographic models using an alternative back end implemented with *msprime*. With its R-idiomatic interface to tree sequence analysis library *tskit* (tskit.dev), *slendr* opens up the possibility of efficient, reproducible, large-scale, population genetic simulations and analyses entirely using the tools of the R ecosystem. We demonstrate the usage of the R package on several complete examples.

website: www.slendr.net

preprint: www.biorxiv.org/content/10.1101/2022.03.20.485041v1

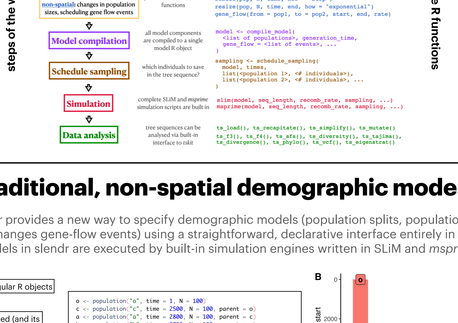
interactive examples from this poster: www.github.com/bodkan/probgen2022

mp@bodkan.net

@dr_bodkan

Overview of a typical *slendr* workflow

Model definition, simulation, and data analysis steps can be part of a single reproducible R script, without having to write code in multiple languages or convert data between file formats.



Traditional, non-spatial demographic models

slendr provides a new way to specify demographic models (population splits, population size changes gene-flow events) using a straightforward, declarative interface entirely in R. Models in slendr are executed by built-in simulation engines written in SLiM and *msprime*.

populations are regular R objects

model is compiled (and its consistency verified) after all events are specified

defined models can be visualized

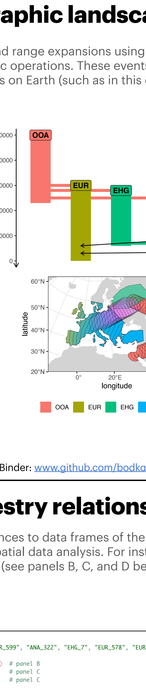
coalescent simulations are executed by slendr's built-in *msprime* script

load a simulated tree sequence and overlay mutations on it

get a table with names, times, nodes of individuals in the tree sequence

compute population divergences and f4-ratio of "b" ancestry in "x1" and "x2"

```
o <- population("o", time = 1, N = 100)
c <- population("c", time = 2500, N = 100, parent = o)
a <- population("a", time = 2800, N = 100, parent = o)
b <- population("b", time = 3700, N = 100, parent = a)
x1 <- population("x1", time = 4000, N = 15000, parent = c)
x2 <- population("x2", time = 4300, N = 15000, parent = x1)

gf <- gene_flow(from = b, to = x1, start = 5400, end = 5800, 0.1)

model <- compile_model(
  populations = list(o, a, b, c, x1, x2), gene_flow = gf,
  generation_time = 1, sim_length = 6000
)

plot_model(model, sizes = FALSE, proportions = TRUE) # panel B

msprime(model, sequence_length = 100e6, recombination_rate = 1e-8)

ts <- ts_load(model) %>% ts_mutate(mutation_rate = 1e-8)

samples <- ts_samples(ts) %>% group_by(pop) %>% sample_n(100)

# panel B
divergence <- ts_divergence(ts, split(samples$name, samples$pop))

# panel B
f4ratio <- ts_f4ratio(
  ts, X = filter(samples, pop %in% c("x1", "x2"))$name,
  A = "a_1", B = "b_1", C = "c_1", O = "o_1"
)
```



run ex1.R in your browser on Binder: www.github.com/bodkan/probgen2022

Spatial model on an abstract spatial landscape

If a simulation world map is defined (in this example, an abstract, featureless map), the model can be simulated with a built-in SLiM back end script.

define a circular world map

spatial models use the same interface as non-spatial models

SLiM spatial parameters: mating, competition, and dispersal distance

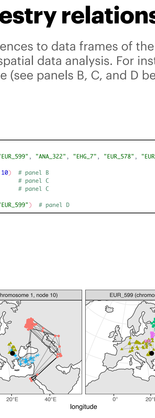
run simulation in SLiM

load tree sequence and add mutations to it

compute heterozygosity in a subset of individuals from each population

extract data frame with nodes, individuals, and their times and locations

```
map <- world(xrange = c(-10, 10), yrange = c(0, 10), crs = 3035)

p1 <- population("pop1", time = 1, N = 2000, map = map, competition = 8)
p2 <- population("pop2", time = 1, N = 2000, map = map, competition = 9)
p3 <- population("pop3", time = 1, N = 2000, map = map, competition = 6)
p4 <- population("pop4", time = 1, N = 2000, map = map, competition = 5)
p5 <- population("pop5", time = 1, N = 2000, map = map, competition = 4)
p6 <- population("pop6", time = 1, N = 2000, map = map, competition = 3)
p7 <- population("pop7", time = 1, N = 2000, map = map, competition = 2)
p8 <- population("pop8", time = 1, N = 2000, map = map, competition = 1)

model <- compile_model(
  populations = list(p1, p2, p3, p4, p5, p6, p7, p8),
  generation_time = 1, sim_length = 5000, resolution = 0.1,
  mating = 0.1, dispersal = 0.05
)

slim(model, sequence_length = 100e6, recombination_rate = 1e-8)

ts <- ts_load(model) %>% ts_simplify() %>% ts_mutate(mutation_rate = 1e-7)

# panel B
heterozygosity <- ts_samples(ts) %>%
  group_by(pop) %>%
  sample_n(100) %>%
  mutate(pi = ts_diversity(ts, name)$diversity)

locations <- ts_data(ts) %>% filter(time == max(time)) # panel C
```



run ex2.R in your browser on Binder: www.github.com/bodkan/probgen2022

Spatial model on realistic geographic landscape

slendr allows scheduling of large-scale population migrations and range expansions using a set of dedicated functions, without the need for handling spatial geometric operations. These events can occur on abstract landscapes but can be also defined on realistic regions on Earth (such as in this example).

download geographic spatial features of West Eurasia

define a circular population, schedule a movement along a given trajectory

define a polygon range

schedule range expansion into a given territory

gene-flow events are collected in a list (*slendr* enforces internal consistency of times of populations' existences)

specify when (and how many) individuals from some populations should be recorded in the tree sequence

"compressed" visualization of scheduled spatial events

simulate the model in SLiM

```
map <- world(xrange = c(-15, 60), yrange = c(20, 65), crs = 3035)

ooa <- population(
  "OOA", time = 52000, N = 500, remove = 23000,
  map = map, center = c_33, 30, radius = 400e3
) %>%
  move(trajectory = ..., start = 50000, end = 48000)

ehg <- population(
  "EHG", time = 23000, N = 1000, parent = ooa, remove = 6000,
  map = map, polygon = ...
)

eur <- population(
  "EUR", time = 36000, N = 2000, parent = ooa,
  map = map, polygon = ...
) %>%
  resize(N = 10000, time = 5000, end = 0, how = "exponential")

ana <- population(
  "ANA", time = 25000, N = 4000, parent = ooa, remove = 3000,
  map = map, polygon = ...
) %>%
  expand_range(by = 3e6, start = 10000, end = 7000, polygon = ...)

yam <- population(
  "YAM", time = 7000, N = 600, parent = ehg, remove = 2500,
  map = m, polygon = ...
) %>%
  move(trajectory = ..., start = 5000, end = 3000)

gf = list(
  gene_flow(ana, to = yam, rate = 0.5, start = 6500, end = 5000),
  gene_flow(ana, to = eur, rate = 0.6, start = 8000, end = 6000),
  gene_flow(yam, to = eur, rate = 0.7, start = 3500, end = 3000)
)

model <- compile_model(
  populations = list(ooa, ehg, eur, ana, yam), gene_flow = gf,
  generation_time = 30, resolution = 10e3,
  competition = 10e3, mating = 100e3, dispersal = 70e3,
)

schedule <- schedule_sampling(
  model, times = seq(0, 50000, by = 1000),
  list(ehg, 20), list(ana, 20), list(yam, 20), list(eur, 20)
)

plot_model(model, sizes = FALSE) # panel B
plot_map(model)                  # panel C

slim(model, burnin = 200000, sampling = schedule,
     sequence_length = 200000, recombination_rate = 1e-8)
```


run ex3-4.R in your browser on Binder: www.github.com/bodkan/probgen2022

Analysis of spatio-temporal ancestry relationships

slendr automatically translates spatial information in tree sequences to data frames of the class *sf*, allowing analysis using the vast array of R packages for geospatial data analysis. For instance, support for this data type is built into the *ggplot2* R package (see panels B, C, and D below).

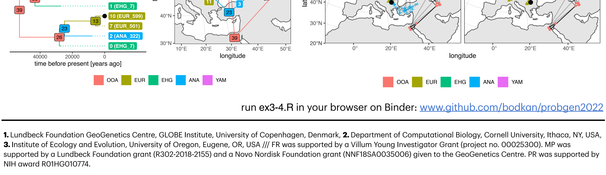
load tree sequence produced by the previous model (and make a smaller version of it through simplification)

convert the 10[th] tree of the *tskit* tree sequence into the phylogenetic format of the *ape* R package

extract tables of spatial locations of each node and edge in the tree

collect the locations (and times) of all ancestral nodes of a given individual across the entire tree sequence

```
ts <- ts_load(model)
ts_small <- ts_simplify(ts, c("EUR_599", "ANA_322", "EHG_7", "EUR_578", "EUR_581", "YAM_30"))

tree <- ts_phylo(ts_small, i = 10)  # panel B
nodes <- ts_data(tree)              # panel C
branches <- ts_branches(tree)       # panel C

ancestors <- ts_ancestors(ts, "EUR_599")  # panel D
```



run ex3-4.R in your browser on Binder: www.github.com/bodkan/probgen2022

1. Lundbeck Foundation GeoGenetics Centre, GLOBE Institute, University of Copenhagen, Denmark, **2.** Department of Computational Biology, Cornell University, Ithaca, NY, USA, **3.** Institute of Ecology and Evolution, University of Oregon, Eugene, OR, USA /// FR was supported by a Villum Young Investigator Grant (project no. 00025300). MP was supported by a Lundbeck Foundation grant (R302-2018-2155) and a Novo Nordisk Foundation grant (NNF18SA0035006) given to the GeoGenetics Centre. PR was supported by NIH award R01HG01074.