

Quetzal: a library for integrating distributional, demographic and coalescent models

Computational Medicine and Bioinformatics
Tools & Technology Seminar Series

Arnaud Becheler

February 4th, 2021

University of Michigan

Introduction

Environmental demogenetics

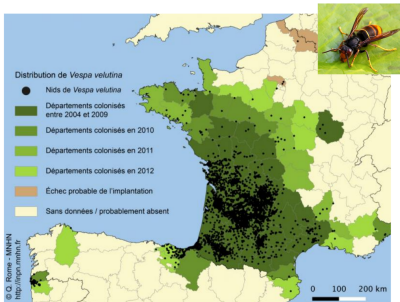
Environmental demogenetics aims to analyze geographic patterns of genetic diversity to inform the influence of **environment** on **demography**.

1. what data ?
2. how to formalize these processes (what model)?
3. how to extract information from data (what inference method)?
4. how to implement the method (what tools)?

Example of a spatial genetic dataset

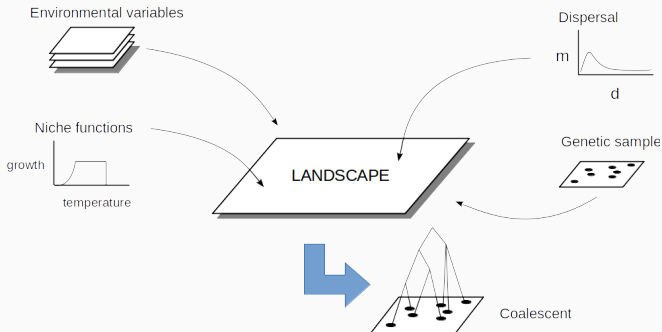
Vespa velutina nigritorax (yellow-legged hornet) invasion in Europe.

- first encountered in South-West France in 2004
- fast expansion
- economical/ecological impact (honey bee predator)
- 84 females genotyped for 22 SSR loci in 2008



lat	lon	List2003	VMA8		
43,50	2,21	163	163	262	264
43,50	2,21	163	169	262	264
43,56	1,46	163	169	260	264
43,97	3,68	177	177	262	264
44,14	4,17	177	177	264	267
44,22	0,55	177	177	264	264
44,22	0,55	163	177	262	264
44,22	0,55	169	177	262	264
44,29	0,69	177	177	264	264
44,29	0,69	163	163	246	264
44,37	3,41	165	169	0	0
44,57	0,23	163	169	264	264
44,57	0,23	163	177	249	264
44,57	0,23	163	163	249	264
44,57	0,23	153	163	249	264
44,57	0,23	169	177	249	264
44,57	0,19	163	163	264	264
44,57	0,19	163	169	264	264
44,57	0,19	163	169	249	264
44,57	0,19	163	177	249	264
44,67	2,16	169	177	264	264

Model summary



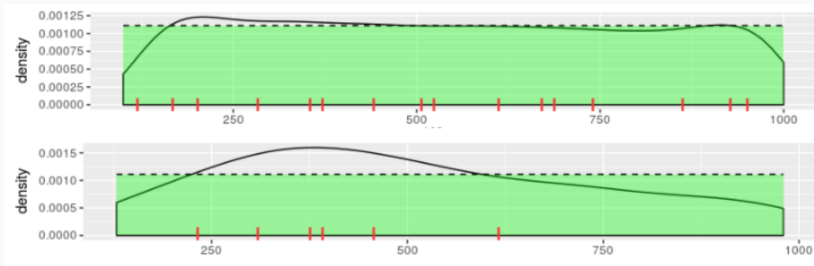
The main challenges are:

- chose submodels (**model selection**)
- estimate parameters from genetic data (**statistical inference**)

Inferential method: Approximate Bayesian Computation

Simulation/reject algorithm:

- sample parameters in a prior distribution: $\theta' \sim p(\theta)$
- simulate data from the model: $y' \sim p(y|\theta')$
- weigh (θ', y') as a function of the distance $\|y' - y_{obs}\|$



What tools for spatially explicit simulations of gene flow within landscapes?

Simulation tools, see Yannic et al. (2020)

Program	Simulator	Level	Lg.	Reference
Splatche3	Backward	Population	C++	Currat et al. (2019)
PhyloGeoSim	Backward	Population	Java	Dellicour (2013)
IBDsim	Backward	Population	C	Leblois et al. (2009)
SLIM3	Forward	Individual	C++	Haller and Messer (2019)

There is an abundance of simulation **programs**, but no **library**:

C++ Library

egglib-cpp is the underlying C++ library of EggLib. It was designed with the aim of improving performance at the expense of safety and intuitive design. Therefore it might be difficult to use directly. The complete contents are listed below:

`egglib._eggwrapper`

C++ library--direct use is strongly discouraged!

C Library

The low-level code for `msprime` is written in C, and is structured as a standalone library. This code is all contained in the `lib` directory. Although the code is structured as a library, it is not intended to be used outside of the `msprime` project! The interfaces at the C level change considerably over time, and are deliberately undocumented.


The no-library-available game

1. If your biological model fits the assumptions of an existing simulation program x , then use x and win.
2. If your biological model presents important deviations from x , then use x anyway and win.
3. If your biological model presents critical deviations from x , then a new program y is required. Since developing y is too time consuming, go back to 2 or forfeit.

The library-available game

1. If your biological model fits the assumptions of an existing simulation program x , then use x and win.
2. If not, use the library to assemble simulation components as you see fit into a program and win.

MOLECULAR ECOLOGY RESOURCES

RESOURCE ARTICLE |  [Free Access](#) |

The Quetzal Coalescence template library: A C++ programmers resource for integrating distributional, demographic and coalescent models

Arnaud Becheler , Camille Coron, Stéphane Dupas

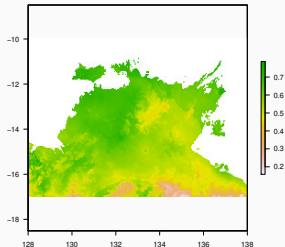
Programming choices for QuetzalCoalTL:

- C++ for a good compromise between design and performances
- Give as much information as possible to the compiler to make the code faster/safer
- Large use of generic programming (templates, metaprogramming)
- Focus on modularity and user experience

Using Quetzal for an Australian lizard: demonstration

Heterogeneous landscape

- Landscape is discretized in n demes (grid cells).
- Let be $L = 1$ environmental variables (suitability derived from a niche model).



quetzal code:

```
string path = "suitability.tiff";  
using landscape_t = DiscreteLandscape<string, int>;  
landscape_t land( {"suitability", path}, {0});  
// auto s = land["suitability"]; // -> callable s(x,t)  
// For temporal heterogeneity:  
// land({"s1", path1}, {"s2", path2}, {0, -1000});
```

Demography initialization

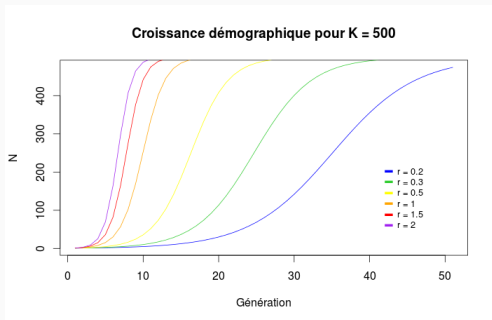
- The ancestral population is a Wright-Fisher of N_A haploid individuals
- Very ancient demography assumed non-spatial.
- At time t_0 , N_0 individuals are introduced in deme x_0 .
- The following history is then spatially explicit.

quetzal code:

```
using core_t = SpatiallyExplicit<coord_t, time_t,
                                demographic_policy,
                                coalescence_policy >;
core_t core(x_0, t_0, N_0);
core.ancestral_Wright_Fisher_N(N_0);
```

Demographic growth

- r : growth rate, constant
- k : carrying capacity, function of suitability



Quetzal code:

```
literal_factory <coord_t, time_t> lit;  
auto r = lit( 2.0 ); // -> callable r(x,t)  
// auto r = lit( options["r"].as<double>() );
```

Demographic growth

Carrying capacity:

- scaled by the suitability value on continental cells
- null in ocean cells "most of the time", but draft dispersal possible

Quetzal code:

```
auto s = land["suitability"]; // -> callable s(x,t)

auto K = [&gen, s] // -> callable K(x,t)
(coord_type const& x, time_type)
{
    // if ocean cell:
        return 0 with proba 0.9, or 1
    // if continental cell:
        return 100*s(x,t)
};
```

Demographic growth

Carrying capacity:

- scaled by the suitability value on continental cells
- null in ocean cells "most of the time", but draft dispersal possible

Quetzal code:

```
...  
  if( s(x,0) <= 0.001){ //ocean cell  
    std::bernoulli_distribution dist(0.1);  
    return dist(gen) ? 1 : 0;  
  }else{ // continental cell  
    return 100*s(x,0);  
  }  
...
```


Demographic growth

Number of children: $\tilde{N}_x^t \sim \text{Poisson}(g(x, t))$

Logistic growth: $g : \begin{cases} \mathbb{X} \times \mathbb{N} & \mapsto \mathbb{R}^+ \\ x, t & \mapsto \frac{N(x, t) \cdot (1+r)}{1 + \frac{r \cdot N(x, t)}{K(x, t)}} \end{cases}$

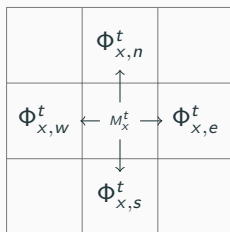
Quetzal code:

```
auto g = N*(lit(1)+r)/(lit(1)+((r*N)/K));
auto children = [g](auto& gen, auto x, auto t){
    std::poisson_distribution<unsigned int> dist(g(x, t));
    return dist(gen);
};
```

Dispersal

- number of effective emigrants going out of deme x : $M_x^t = e \times \tilde{N}_x^t$
- set of x neighbours (North, South, East, West): \mathbb{V}_x
- number of individuals going from x to $y \in \mathbb{V}_x$ at time t : $\Phi_{x,y}^t$
- sampling emigrants destination in a multinomial law defines $\Phi_{x,y}^t$:

$$(\Phi_{x,y}^t)_{y \in \mathbb{V}_x} \sim \mathcal{M}(\tilde{N}_x^t, (p_{xy})_y).$$



Quetzal code:

```
using demographic_policy = strategy::mass_based;  
auto neighbors = make_neighboring_cells_functor(land);15
```

Friction

The term $(p_{xy})_y$ denotes the parameters of the multinomial law, giving for an emigrant in x its probability to go to $y \in \mathbb{V}_x$:

$$\begin{aligned} p & : \mathbb{X} \times \mathbb{V}_x \mapsto [0, 1] \\ (x, y) & \mapsto \frac{1}{h(y) \cdot \sum_{i \in \mathbb{V}_x} \frac{1}{h(i)}} . \end{aligned}$$

where h is a function of the suitability.

Quetzal code:

```
auto h = [s](auto x){
    if(s(x,0) <= 0.5) {return 0.99;} // ocean or deserts
    else return 1.0 - s(x, 0);
};
kernel = make_light_neighboring_migration(
emigrant_rate, h, neighbors
);
```

Demographic process

Flow of migrants Φ :
 $(\Phi_{x,y}^t)_{y \in \mathbb{V}_x} \sim \mathcal{M}(\tilde{N}_x^t, (p_{xy})_y) .$

Population size N :

$$N_j^{t+1} = \sum_{i \in \mathbb{X}} \Phi_{i,j}^t$$

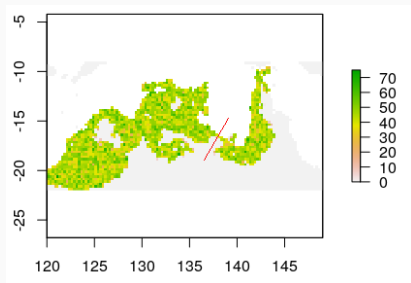


Figure 1: Click.

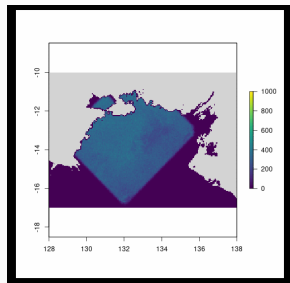


Figure 2: Click.

Qetzal code:

```
core.expand_demography(2021, children, kernel, gen)); 17
```

Genetic process

Let be a set S de n gene copies sampled at time t_s . Coalescent trees are then simulated backward in time, from t_s to t_0 .

Knowing that a child node c is in deme $j \in \mathbb{X}$, the probability for its parents p to be in $i \in \mathbb{X}$ is a function of the migration flow Φ :

$$P(p \in i \mid e \in j) = \frac{\Phi_{i,j}^t}{\sum_k \Phi_{k,j}^t}$$

Knowing that the parent nodes p_1 (p_2) of the nodes c_1 (c_2) are in i at time t , the probability for the children to coalesce in the same parent is:

$$P(p_1 = p_2 \mid p_1 \in i, p_2 \in i) = 1/N_i^t$$

Quetzal code:

```
using coal_policy=distance_to_parent_leaf_name <...>;
core.coalesce_to_mrca <>(
sample, 2021, get_position, get_name, gen);
```

Conclusion

- Very flexible resource for demographic processes
- Easy to couple to coalescence simulators
- Next big step is to couple it to Tskit library (Kelleher et al., 2014) for efficient generation of correlated trees.
- Open source on github

Thank you for your attention !

DECRYPT

Simulation pipeline

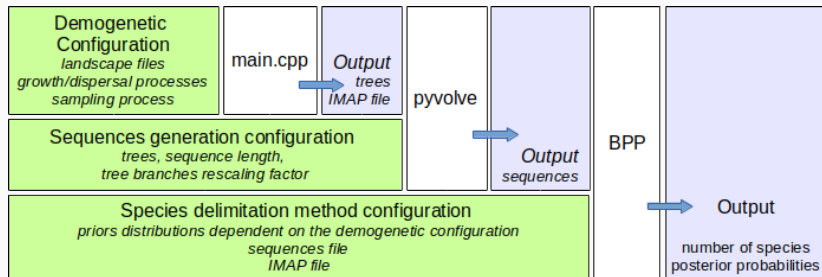


Figure 3: Pipeline to simulate coalescence trees conditioned on a complex spatially explicit demographic history and sampling schemes using Quetzal (Becheler et al., 2019), then simulating sequences along the trees using Pyvolve (Spielman and Wilke, 2015) and delimiting species using BPP (Flouri et al., 2018)

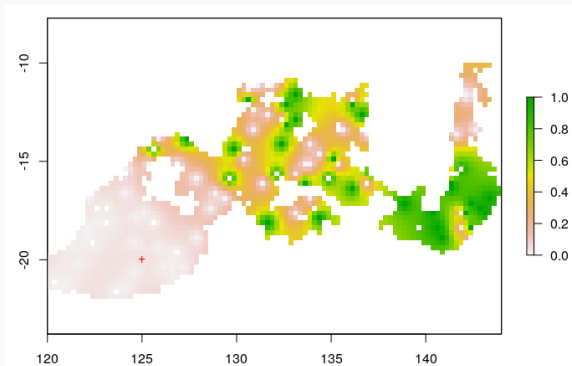


Figure 4: Spatial interpolation of p_x the probability to detect 2 species in a population expanding in an heterogeneous landscape under the MSC when the sequences sample is constructed at time t_s by two 2D gaussian sampling processes centered on (i) the population origin x_0 (red cross), and (ii) on a random coordinate x (with $N(x, t_s) > 30$ to avoid inconsistent sampling in very low density areas).

References

- Becheler, A., Coron, C., and Dupas, S. (2019). The quetzal coalescence template library: a c++ programmers resource for integrating distributional, demographic and coalescent models. *Molecular ecology resources*.
- Curat, M., Arenas, M., Quilodràn, C. S., Excoffier, L., and Ray, N. (2019). Splat3: simulation of serial genetic data under spatially explicit evolutionary scenarios including long-distance dispersal. *Bioinformatics*, 35(21):4480–4483.
- Dellicour, S. (2013). Phylogeosim 1.0: a simulator of dna sequences evolution under a demographic and geographic model of coalescence. In *5th CÉCI Scientific Meeting, Location: Namur, Belgium*.

- Flouri, T., Jiao, X., Rannala, B., and Yang, Z. (2018). Species Tree Inference with BPP Using Genomic Sequences and the Multispecies Coalescent. *Molecular Biology and Evolution*, 35(10):2585–2593.
- Haller, B. C. and Messer, P. W. (2019). Slim 3: forward genetic simulations beyond the wright–fisher model. *Molecular biology and evolution*, 36(3):632–637.
- Kelleher, J., Etheridge, A., and Barton, N. (2014). Coalescent simulation in continuous space: Algorithms for large neighbourhood size. *Theoretical Population Biology*, 95:13–23.
- Leblois, R., Estoup, A., and Rousset, F. (2009). IBDSim: a computer program to simulate genotypic data under isolation by distance. *Molecular Ecology Resources*, 9(1):107–109.

Spielman, S. J. and Wilke, C. O. (2015). Pyvolve: a flexible python module for simulating sequences along phylogenies. *PLoS one*, 10(9):e0139047.

Yannic, G., Hagen, O., Leugger, F., Karger, D. N., and Pellissier, L. (2020). Harnessing paleo-environmental modeling and genetic data to predict intraspecific genetic structure. *Evolutionary Applications*, 13(6):1526–1542.