To perform demographic inference using simulations, you can follow these general steps:

1. Define the demographic model: Start by defining the demographic model that you want to infer from your data. This includes specifying population size changes, migration rates, admixture events, and any other relevant parameters that describe the demographic history of the populations you're studying.

2. Simulate data: Use a simulation tool like msprime or similar software to generate simulated genetic data based on the demographic model you defined. Simulations generate genetic data that mimic the expected patterns under the specified model.

3. Generate summary statistics: Calculate summary statistics from the simulated data that capture important features of the genetic variation. These statistics could include measures like nucleotide diversity, Tajima's D, Fst values, or site frequency spectrum (SFS) patterns.

4. Choose an inference method: Select a suitable inference method to estimate the demographic parameters from the summary statistics. Popular methods include approximate Bayesian computation (ABC), composite-likelihood approaches, machine learning algorithms, or coalescent-based methods that directly compare the simulated and observed summary statistics.

5. Perform parameter estimation: Apply the chosen inference method to estimate the demographic parameters from the observed summary statistics. This typically involves comparing the observed summary statistics to those generated from the simulated data under different parameter values. The inference method will find the parameter values that best explain the observed data.

6. Validate and assess the fit: Once you have estimated the parameters, assess the goodness-of-fit by comparing the observed summary statistics to those generated from the inferred parameter values. Statistical tests, model selection criteria, or visual comparisons can be used to assess the fit between the inferred model and the observed data.

7. Refine the model: If the fit is not satisfactory, you may need to refine your demographic model by modifying the parameters or exploring alternative scenarios. Iterate through steps 2-6 until you obtain a satisfactory fit between the model and the data.

8. Interpret and analyze the results: Finally, interpret the estimated demographic parameters in the context of your study question. Consider the implications of the inferred demographic history and how it relates to the population or species under investigation.

It's important to note that demographic inference is a complex process, and the choice of simulation and inference methods depends on the specific goals of your study and the characteristics of your data. Consulting relevant literature, seeking guidance from experts, or considering established methodologies can be helpful in implementing these steps effectively.

QUESTIONS TO STEPHANE:

How to set:

effective population size?

mutation rates? 3e-7?

recombination rate? 3e-7?

Since our model is a classical coalescent, we don’t need to specify the effective population size. Instead we can specify the product of the population size and of the mutation rate (θ) but we even skip that since we condition simulations on the observed number of polymorphic sites. We also skip the recombination rates since we assume that there is no physical linkage between the SNPs we are using (that’s why we take them far apart, with a 200kb spacer).

With other software or other models you will need to specify them. In my view it is better to estimate them because if you assume a value that is actually wrong you’ll make the analysis unreliable.

Which statistics should we use?

One-way, or multi-way? Multi-way by populations?

The random forest method is described as tolerant to correlation between statistics, so we are good to take as many as possible and especially as complementary as possible to capture as many features of the patterns of polymorphism as possible.

How do specify population size changes, migration rates, admixture events, and any other relevant parameters that describe the demographic history?

That’s all in the documentation of the simulation software you are using.

How can I specify such parameters that define the demographic models?

Parameters are estimated from the data. But the more parameters, the more the simulations we need to capture all combinations of possible values.

1. Should we add maf effect to compute observed stats, as well?

It is already the case since you only included sites satisfying the MAF criterion in the VCF dataset.

1. To implement the new model, I can keep all merge evenst as in the model4, right?

model4: coal.params.add\_event('merge', T=0, src=0, dst=3)

coal.params.add\_event('merge', T=0, src=1, dst=3)

coal.params.add\_event('merge', T=0, src=3, dst=4)

coal.params.add\_event('merge', T=0, src=2, dst=4)

Technically you can but you don’t need and you should not. We use the populations #3 and #4 as ancestral populations because ancestral populations were not identified as one of the three continents.

So the sequence of events above was:

- merge NAM to ancestral population 1  
- merge EUR to ancestral population 1  
- merge ancestral population 1 to ancestral population 2  
- merge BR to ancestral population 2  
  
Now the ancestral populations are identified. In this case (model 7) it is Europe in both cases. The merge of the population to itself has no meaning and is just an artifact of the fact that previously you modelled the merging of BR and EUR by merging them both to a third population. Now you just merge BR to EUR, and then NAM to EUR and you don’t need the second merge, like this:

- merge NAM to EUR  
- merge BR to EUR

Model7:

coal.params.add\_event('merge', T=0, src=2, dst=1)

coal.params.add\_event('merge', T=0, src=0, dst=1)

But now, for instance for model 7 (Europe as ancestral lineage), for event0, Europe will the src and dst? Same for event 2?

Model7: coal.params.add\_event('merge', T=0, src=1, dst=1)

coal.params.add\_event('merge', T=0, src=2, dst=1)

coal.params.add\_event('merge', T=0, src=1 dst=1)

coal.params.add\_event('merge', T=0, src=0, dst=1)

A diagram of a diagram

Description automatically generated

In this graph there are actually no event0 and event2: the yellow line is a single population that actually doesn’t change over the course of the simulation. It would be clearer if you drew it as a straight line, while the other two separate from it

EUR─────────────> EUR sample

│

│*event4*

│

└><─NAM────────> NAM sample

│

│*event1*

│

└><─BR──> BR sample

INPLEMENTACION MORE COMPLEX MODELS:

-Ancestral lineage’s definition: older lineage, the source lineage which split in two or more lineages. In this context, the ancestral lineage encompasses the individuals that descended from the common ancestor.

I understand it this way: ancestral population = which one of the current populations who has a continuous history without event of colonization (which we can detect in principle through the signatures of a bottleneck) and from which the other populations have emerged.

**Model 7, 8, 9 and 10: to detect the ancestral lineage using models including bottlenecks:**

This representation can be a bit easier to understand if you make the ancestral population continue in the past. Then you read it from past to present to follow the events of colonization

A screenshot of a computer

Description automatically generated

<= Questions!!!!

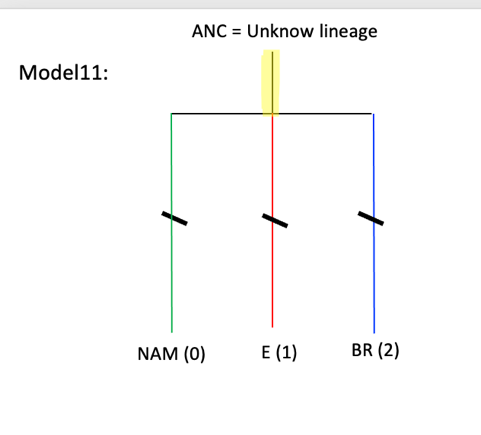
Models 8 and 9 are not correct because NAM is not external. For NAM to be external, BR and E should be branched together. So the first merge should be BR to E or E to BR.  
  
If the colonization route is NAM→BR→EUR, looking backward you have first a merge or EUR to BR and then a merge of BR to NAM.

* For the model 7, 8, 9 and 10 we are considering each lineage as ancestral. Keeping my previous definition of ancestral lineage, could have a scenario which we did not sample the ancestral lineage, i.e., as we don’t have samples from all countries, so we could have more lineages and one of them could be ancestral).

-Does it make sense?

Yes it does.

-How should the implementation of this model?Something like that?

****

Yes. The crucial point is that all three populations had a bottleneck. But this is a special case where the three populations emerged at the same time. There are actually four scenarios keeping NAM as ancestral but I expect some will be impossible to discriminate.

Case 1 Case 3

ANC -> NAM ANC -> NAM

-> BR -> BR -> EUR

-> EUR

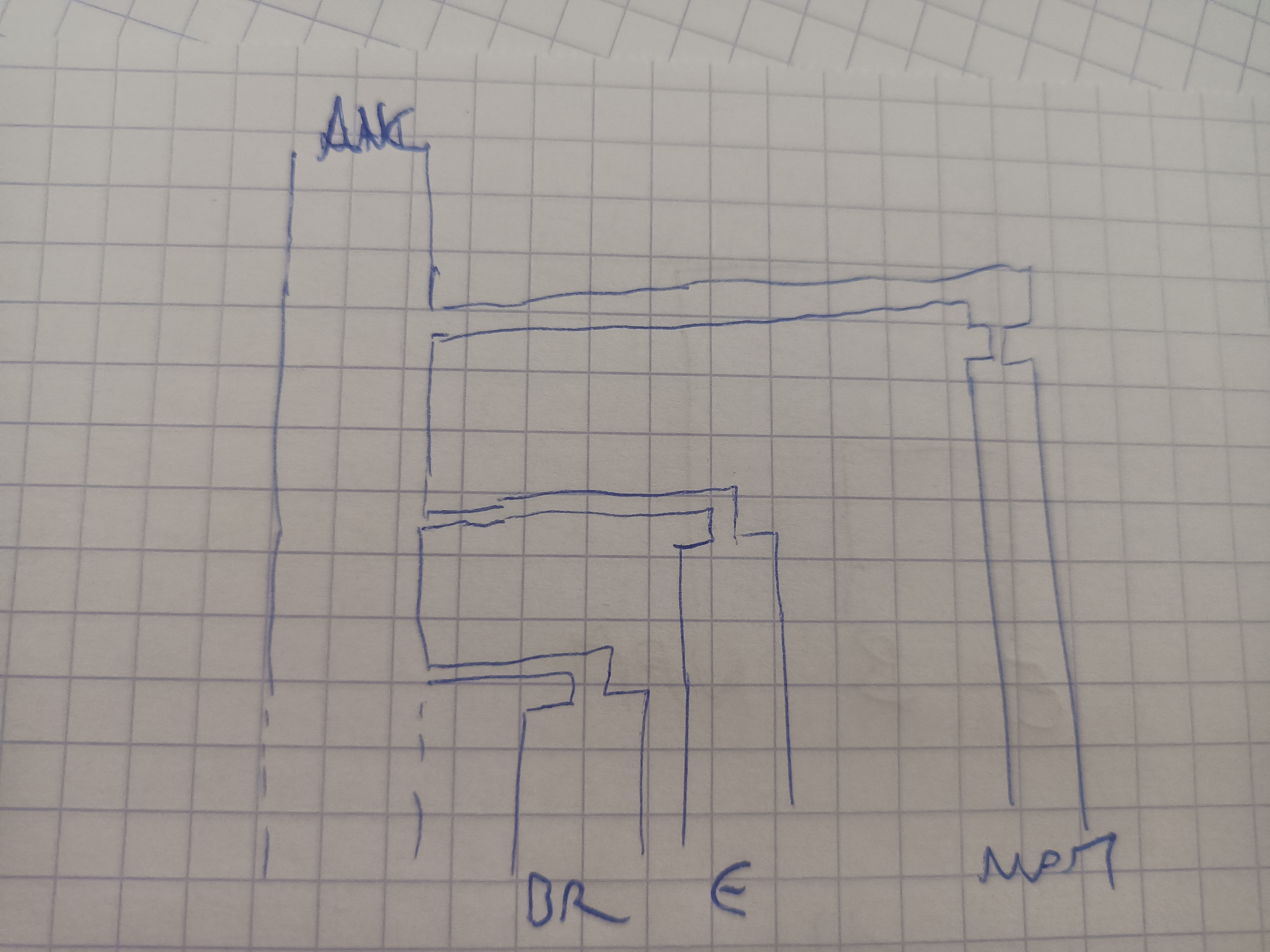
Case 2 Case 4

ANC -> NAM ANC -> NAM

-> EUR -> EUR -> BR

-> BR

Case 2 represented in the picture below



* In egglib tutorial there are two ways to implement bottleneck in the model:

1) Get independent population size before and after the bottleneck (2 times and 2 population sizes)

2) Using the bottleneck event, which has the advantage of saving the parameter size (N). Instead of requiring both duration and strength parameters, it requires only a strength parameter (S). This assumes that the bottleneck is short enough that its duration can be neglected. Instead, a random number of coalescence events are instantaneously performed at the time of the bottleneck.

-By what we talked we’ll use this method, right? In this case, only parameters T and S are implemented?

Yes it is the one.

-Both parameters will be simulated, or fixed? how should be their range values? Should we use “loguniform\_draw” function?

The parameters should be drawn from a prior and estimated. For the time, it makes sense to place the bottleneck immediately before each merge event. The bottleneck is a side effect of the colonization event. So you don’t need to draw a different date. For S I am using now a log-normal distribution with parameters math.log(0.1), 1.

-On our models, we have two bottleneck events. Does both occur at same time? Or in different times? I suppose that simulate them at the same is easier.

If the bottlenecks correspond to two colonization events occurring at different time, they should be modelled accordingly.

-How should I refer to the parameter T to differentiate it from split time? you told something about add a “small values” for it to avoid overlap with split size, for instance use +1e6?

How effectively do that?

You just need to set the bottleneck time to T and the merge event time to T+1e6.