Extended Bayesian Skyline Plot using highly variable RADseq loci

Emiliano Trucchi and Robin Cristofari

Since its description by Heled and Drummond (2008), the Extended Bayesian Skyline Plot (EBSP) model has been widely used to infer population history from multilocus genetic data. At the same time, RAD-sequencing (Baird *et al.* 2008) has been developed as a popular method to produce haplotype-based population-level datasets in non-model species, thus providing robust material for demographic inference, even if made of short genomic fragments. Using RADseq data in an EBSP analysis has been first proposed in Trucchi *et al* (2014) and proved to be quite effective in inferring recent demographic history based on only 8 king penguin individuals. The analysis was then improved by Robin Cristofari and Alexei Drummond. In Cristofari *et al* (submitted – *to be updated soon*), the latest developments of this approach are implemented showing how the resolution in the king penguin demography increases when including 50 individuals. In that paper, EBSP performance is shown in comparison with other methodologies (e.g. PSMC, Stairway Plot).

RAD loci are often processed without a reference genome, and lack positional or annotation information. Selection of appropriate loci is therefore a key point if we want to obtain an unbiased representation of the neutrally-evolving part of the genome. Selection of more informative loci has been commonly used in population genetics analyses and in demographic inferences from genetic data (*i.e.* using the mitochondrial fast evolving Control Region rather than a slower Cytochrome *b*), and we consider that, in most of the cases, it is unlikely to compromise the analysis.

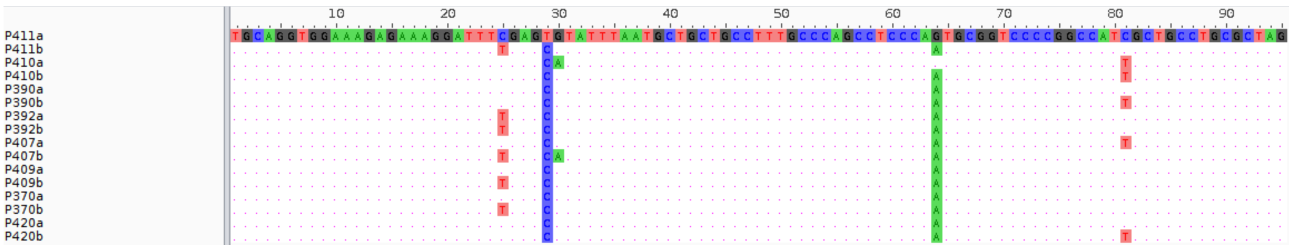
If your dataset has a particularly low coverage (e.g. because of a high duplication rate), it is possible that several individuals that appear as homozygous at a given locus are instead under-sequenced heterozygotes. This is likely to have a large impact on the coalescent reconstructions. In that case, we recommend to randomly sample only one haplotype per individual: to minimize the bias of undiscovered heterozygote samples and to guarantee that haplotype frequencies are unbiased. Indeed, we are only interested in sampling independent gene trees from the whole population (that is assumed panmictic), and not from particular individuals.

1 - Prepare your input files

Select RAD loci with at least 4 SNPs across the individuals included in the population you want to analyse. In our experience, loci with 3 SNPs or less contain information about too few coalescence events to be useful. Depending on the genetic diversity in your biological system, you can find loci with a rather high SNP proportion: however, if you are really careful with your controls (see above), you will see that finding loci with more than 9-10 genuine SNPs is quite unlikely.

In order to proceed, you will need these loci in nexus format, one per locus. You can generate these files in many ways depending on how you processed your RADseq dataset. If you have no reference genome, you probably have your loci in a catalog built by Stacks (Catchen *et al*. 2013). In that case, the easiest way to proceed is to export your full catalog using the *populations* program, in fasta format. Be aware that Stacks only outputs one haplotype for homozygotes, and two for heterozygotes: you will need to either randomly pick one haplotype for the heterozygotes, or to duplicate the haplotype for the homozygotes. If you are only using a subset of individuals, you will need to calculate the true number of SNPs in your haplotype subset. This whole process can be done not too painfully using the RAD_Haplotypes R script (https://github.com/rcristofari/RAD-Scripts/blob/master/RAD_Haplotypes.R).

Each RADseq locus has to be coded as a separated nexus or fasta file with each individual represented by its two alleles (no correction applied) or one allele (down-sampled) at that locus. We used only the forward reads (95 base pairs in our data) because the SNPs present there are all physically phased. Extension to the paired-end longer contigs is also possible but requires a bit of scripting in order to retain the information about physical phasing (contact Emiliano Trucchi if interested in developing this).

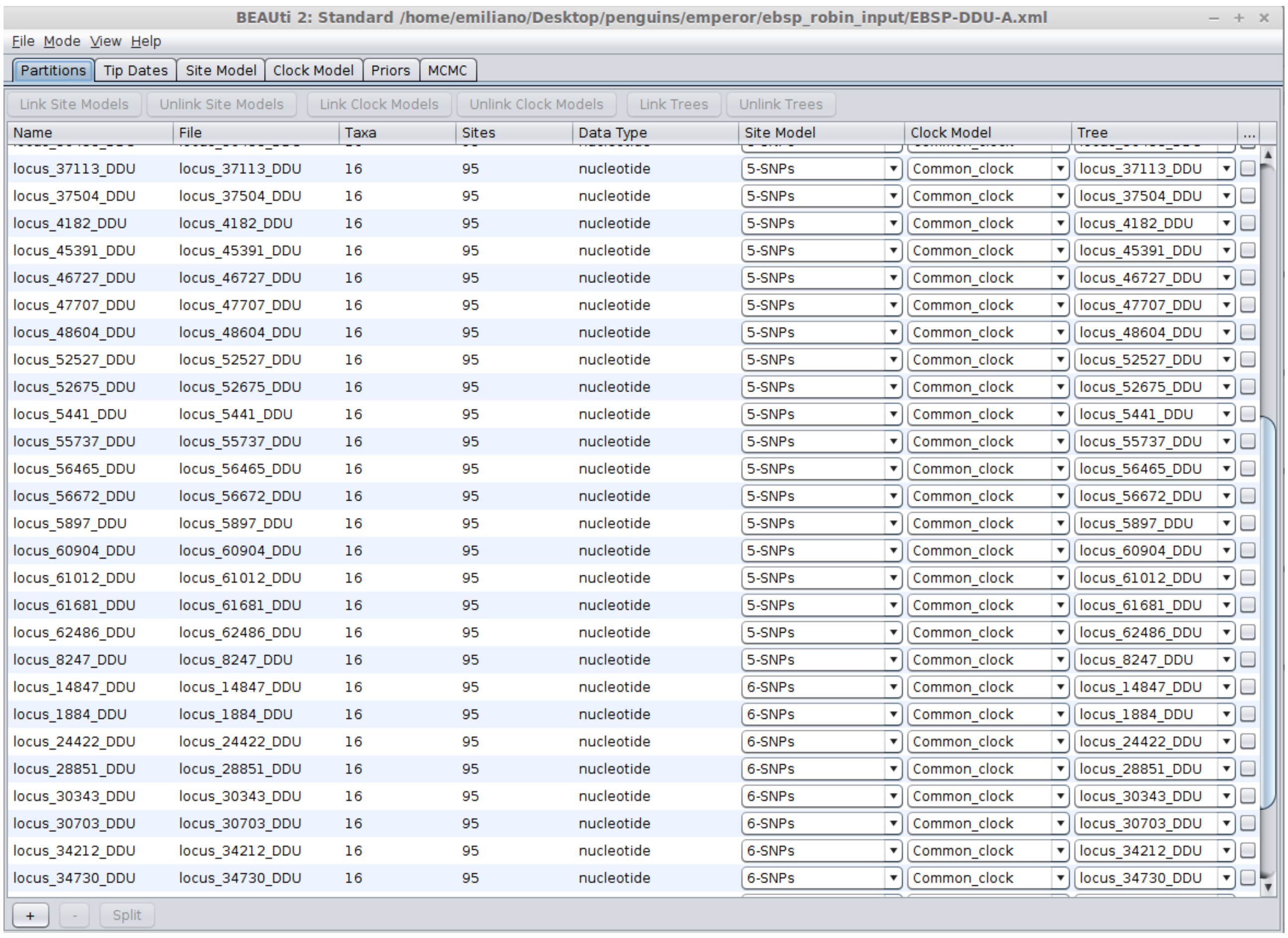


95 bp long RAD locus with 5 SNPs sequenced in 8 individuals; two alleles per individual.

2 – Open a standard *beauti* template for BEAST2 and upload your loci.

We recommend starting with a random selection of 50 loci and increasing the number if run length is not prohibitive. Run at least 3 replicates using a different random selection of loci to check for convergence in the inferred demography.

The number of steps required in order to achieve convergence in the Beast2 MCMC havily depends upon the number of parameters used. We therefore try to reduce the parameter space as much as possible by linking all likely redundant parameters. We will only define one sites model for each locus class (i.e. one model for loci with 4 SNPs, one for loci with 5 SNPs, etc). Therefore, in the partition panel **link the site model** for all of the loci with the same number of SNPs. You can give your site models meaningful names like “5-SNPs”, “6-SNPs”, etc. Then **link the clock model** for all of the loci (we assume that our RADome is evolving neutrally, and therefore with no rate heterogeneity between loci) and **unlink the trees** for all of the loci (since each tree is an independent dram from the population's coalescent history).



BEAUti 2: Standard /home/emiliano/Desktop/penguins/emperor/ebsp_robin_input/EBSP-DDU-A.xml

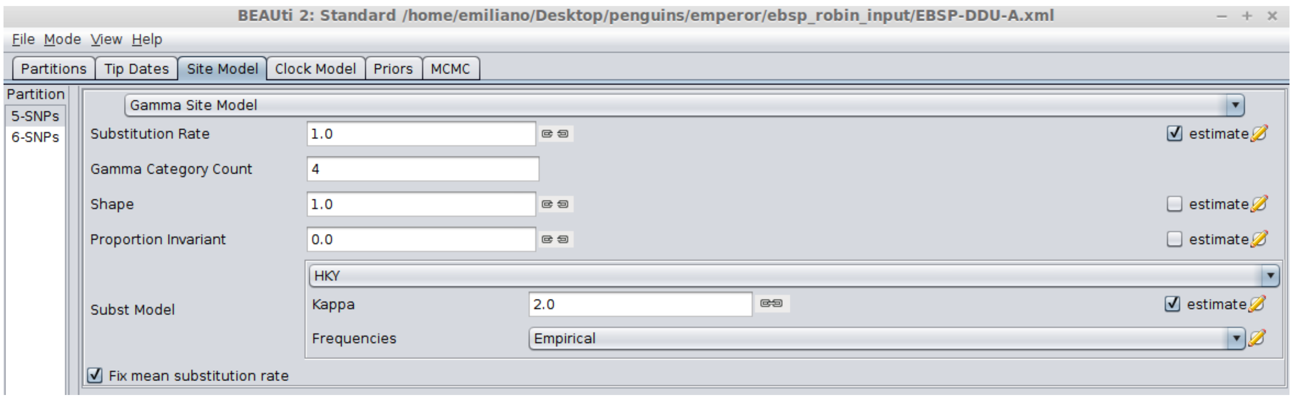
Name	File	Taxa	Sites	Data Type	Site Model	Clock Model	Tree
locus_37113_DDU	locus_37113_DDU	16	95	nucleotide	5-SNPs	Common_clock	locus_37113_DDU
locus_37504_DDU	locus_37504_DDU	16	95	nucleotide	5-SNPs	Common_clock	locus_37504_DDU
locus_4182_DDU	locus_4182_DDU	16	95	nucleotide	5-SNPs	Common_clock	locus_4182_DDU
locus_45391_DDU	locus_45391_DDU	16	95	nucleotide	5-SNPs	Common_clock	locus_45391_DDU
locus_46727_DDU	locus_46727_DDU	16	95	nucleotide	5-SNPs	Common_clock	locus_46727_DDU
locus_47707_DDU	locus_47707_DDU	16	95	nucleotide	5-SNPs	Common_clock	locus_47707_DDU
locus_48604_DDU	locus_48604_DDU	16	95	nucleotide	5-SNPs	Common_clock	locus_48604_DDU
locus_52527_DDU	locus_52527_DDU	16	95	nucleotide	5-SNPs	Common_clock	locus_52527_DDU
locus_52675_DDU	locus_52675_DDU	16	95	nucleotide	5-SNPs	Common_clock	locus_52675_DDU
locus_5441_DDU	locus_5441_DDU	16	95	nucleotide	5-SNPs	Common_clock	locus_5441_DDU
locus_55737_DDU	locus_55737_DDU	16	95	nucleotide	5-SNPs	Common_clock	locus_55737_DDU
locus_56465_DDU	locus_56465_DDU	16	95	nucleotide	5-SNPs	Common_clock	locus_56465_DDU
locus_56672_DDU	locus_56672_DDU	16	95	nucleotide	5-SNPs	Common_clock	locus_56672_DDU
locus_5897_DDU	locus_5897_DDU	16	95	nucleotide	5-SNPs	Common_clock	locus_5897_DDU
locus_60904_DDU	locus_60904_DDU	16	95	nucleotide	5-SNPs	Common_clock	locus_60904_DDU
locus_61012_DDU	locus_61012_DDU	16	95	nucleotide	5-SNPs	Common_clock	locus_61012_DDU
locus_61681_DDU	locus_61681_DDU	16	95	nucleotide	5-SNPs	Common_clock	locus_61681_DDU
locus_62486_DDU	locus_62486_DDU	16	95	nucleotide	5-SNPs	Common_clock	locus_62486_DDU
locus_8247_DDU	locus_8247_DDU	16	95	nucleotide	5-SNPs	Common_clock	locus_8247_DDU
locus_14847_DDU	locus_14847_DDU	16	95	nucleotide	6-SNPs	Common_clock	locus_14847_DDU
locus_1884_DDU	locus_1884_DDU	16	95	nucleotide	6-SNPs	Common_clock	locus_1884_DDU
locus_24422_DDU	locus_24422_DDU	16	95	nucleotide	6-SNPs	Common_clock	locus_24422_DDU
locus_28851_DDU	locus_28851_DDU	16	95	nucleotide	6-SNPs	Common_clock	locus_28851_DDU
locus_30343_DDU	locus_30343_DDU	16	95	nucleotide	6-SNPs	Common_clock	locus_30343_DDU
locus_30703_DDU	locus_30703_DDU	16	95	nucleotide	6-SNPs	Common_clock	locus_30703_DDU
locus_34212_DDU	locus_34212_DDU	16	95	nucleotide	6-SNPs	Common_clock	locus_34212_DDU
locus_34730_DDU	locus_34730_DDU	16	95	nucleotide	6-SNPs	Common_clock	locus_34730_DDU

3 – Site models

Go to the site model panel in *beauti*. On the left, in the partition column, you can see the site models you set in the Partition panel. In our example we have two partitions: 5-SNPs and 6-SNPs. Clicking on the names, you can set the parameters for each of them. Click on Mode on the bar menu and make sure that “Automatic set clock rate” and “Allow parameter linking” are flagged while “Automatic set fix mean substitution rate flag” is not flagged.

Since we assume that all our loci are mostly drawn from a neutrally evolving part of the genome, we expect the underlying substitution model to be the same: only the inferred substitution rate should diverge between classes, as a result of the different numbers of observed SNPs. All other parameters (e.g. the kappa ratio in an HKY model) can be linked across sites models. Therefore set all of the site models you have using the same parameters. In our case, we decided for a simple HKY with 4 Gamma categories and base frequencies “Empirical” (which is fine for 95bp loci with no strong base frequency bias, and saves 4 parameters per sites model!). Check the boxes to estimate the substitution rate and kappa. Check the box to Fix mean substitution rate and then link kappa across the partitions (click on the link symbol on the side and select the other partition than the one you are editing).

If you include a locus with a known substitution rate and wish to use it to calibrate your reconstruction, on the other hand, you will need to provide a separate sites model for that locus. In order to reflect the uncertainty on the subsitution rate for that locus, the rate will be estimated too, but with a prior reflecting your known confidence interval. In that case, uncheck the "Fix mean substitution rate" box, and give the point estimate as a starting value here (keep the "estimate" box checked for the substitution rate).



BEAUti 2: Standard /home/emiliano/Desktop/penguins/emperor/ebsp_robin_input/EBSP-DDU-A.xml

File Mode View Help

Partitions | Tip Dates | Site Model | Clock Model | Priors | MCMC

Partition: 5-SNPs, 6-SNPs

Gamma Site Model

Substitution Rate 1.0 [✓ estimate]

Gamma Category Count 4

Shape 1.0 [ ] estimate

Proportion Invariant 0.0 [ ] estimate

Subst Model: HKY

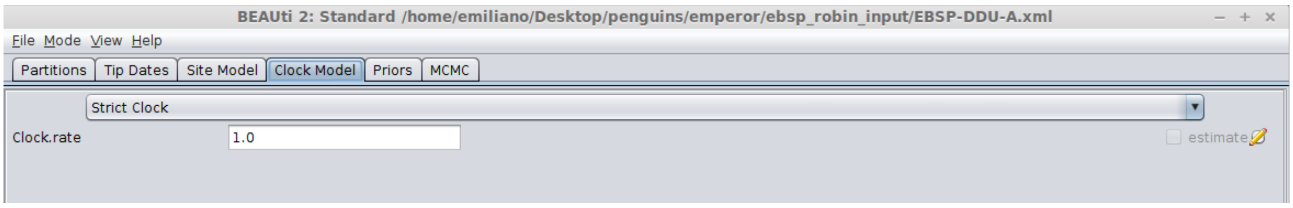
Kappa 2.0 [✓ estimate]

Frequencies Empirical

[✓] Fix mean substitution rate

4 – Clock model

Select a strict clock and leave the Clock.rate to 1.0. As you can see, the estimate check box is not active given the settings in the Site Model panel



BEAUti 2: Standard /home/emiliano/Desktop/penguins/emperor/ebsp_robin_input/EBSP-DDU-A.xml

File Mode View Help

Partitions | Tip Dates | Site Model | Clock Model | Priors | MCMC

Strict Clock

Clock.rate 1.0 [ ] estimate

5 – Priors

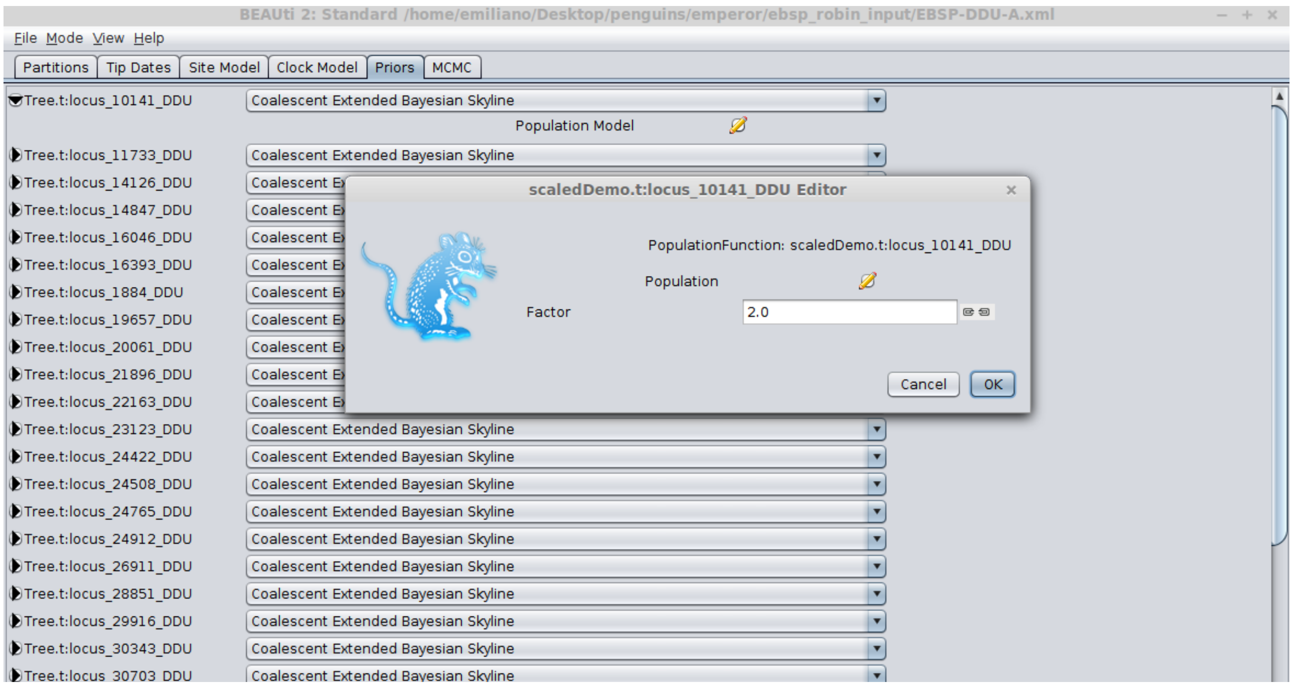
Select the Coalescent Extended Bayesian Skyline model for each locus (we unlinked all of the trees in the Partitions panel). Click on the black triangles on the left and the on the edit button of the Population model to set the Factor to 2 for diploid loci. If this is too tedious, ploidy can also be changed by editing the xml file: look for a line like:

<parameter id="RealParameter.0" lower="0.0" name="factor" upper="0.0">1.0</parameter>

and change *`>1.0<`* to *`>2.0<`*) for each locus.

If your ML model (from JModelTest or any other program) for a particular locus requires it, you may need to change the prior distribution for some parameters. For example, under our HKY model, the kappa prior needs to be a broader log-normal distribution than the default, frequently allowing values up to 60-70. Leave the rest to the default settings.

If you are including a calibration locus with a known substitution rate, you will want to set a very informative prior on that parameter. Generally, you would want a normal distribution centered around the mean value, and reflecting its 95%CI. However it may happen that such a distribution will have a negative lower tail - and a negative substitution rate is definitely not something you want. In that case, you may want to specify a log-normal prior distribution instead - the skew in sampling probability should not prevent the parameter from converging to the correct mean value.


BEAUti 2: Standard /home/emiliano/Desktop/penguins/emperor/ebsp_robin_input/EBSP-DDU-A.xml

File Mode View Help

Partitions | Tip Dates | Site Model | Clock Model | Priors | MCMC

Tree.t:locus_10141_DDU — Coalescent Extended Bayesian Skyline
Population Model
Tree.t:locus_11733_DDU — Coalescent Extended Bayesian Skyline
Tree.t:locus_14126_DDU — Coalescent E
Tree.t:locus_14847_DDU — Coalescent E
Tree.t:locus_16046_DDU — Coalescent E
Tree.t:locus_16393_DDU — Coalescent E
Tree.t:locus_1884_DDU — Coalescent E
Tree.t:locus_19657_DDU — Coalescent E
Tree.t:locus_20061_DDU — Coalescent E
Tree.t:locus_21896_DDU — Coalescent E
Tree.t:locus_22163_DDU — Coalescent E
Tree.t:locus_23123_DDU — Coalescent Extended Bayesian Skyline
Tree.t:locus_24422_DDU — Coalescent Extended Bayesian Skyline
Tree.t:locus_24508_DDU — Coalescent Extended Bayesian Skyline
Tree.t:locus_24765_DDU — Coalescent Extended Bayesian Skyline
Tree.t:locus_24912_DDU — Coalescent Extended Bayesian Skyline
Tree.t:locus_26911_DDU — Coalescent Extended Bayesian Skyline
Tree.t:locus_28851_DDU — Coalescent Extended Bayesian Skyline
Tree.t:locus_29916_DDU — Coalescent Extended Bayesian Skyline
Tree.t:locus_30343_DDU — Coalescent Extended Bayesian Skyline
Tree.t:locus_30703_DDU — Coalescent Extended Bayesian Skyline

scaledDemo.t:locus_10141_DDU Editor

PopulationFunction: scaledDemo.t:locus_10141_DDU
Population
Factor 2.0

Cancel | OK

6 – Operators

Click on View on the bar menu and select the **Operator panel**. You are now able to change the operators. Operator weights will determine how often each parameter will be updated along the MCMC chain, relatively to the others: a high weight means that an operator is renewed often, and will generally be more densely sampled along the chain.

All parameters relative to individual trees can be left as they are, as gene trees with only a handful of SNPs are quite easy to estimate. On the other hand, parameters relative to the full EBSP model will need to be seriously increased if you want to avoid running your MCMC forever. When using 50 loci, we recommend to multiply all common parameters' operators by 100 or more - except sites models' kappa that can be multiplied by 10 only.

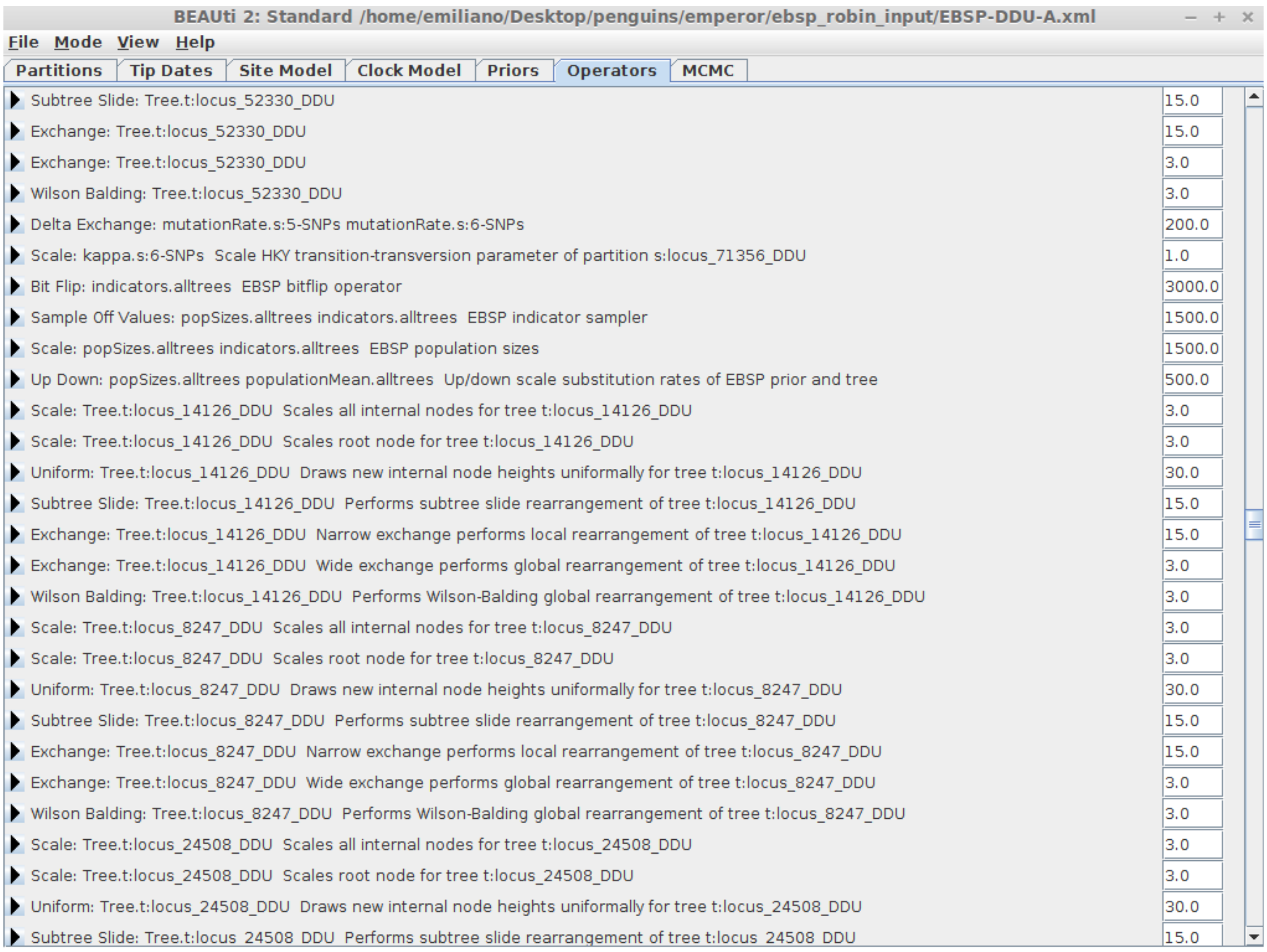
So, in our example, with 50 loci:

- Bit Flip: indicators.alltrees EBSP bitflip operator - **Weight = 3000**

- Sample Off Values: popSizes.alltrees indicators.alltrees EBSP indicator sampler – **Weight = 1500**

- Scale: popSizes.alltrees indicators.alltrees EBSP population sizes – **Weight = 1500**

- Up Down: popSizes.alltrees populationMean.alltrees Up/down scale substitution rates of EBSP prior and tree – **Weight = 500**


BEAUti 2: Standard /home/emiliano/Desktop/penguins/emperor/ebsp_robin_input/EBSP-DDU-A.xml
File Mode View Help
Partitions Tip Dates Site Model Clock Model Priors Operators MCMC

Subtree Slide: Tree.t:locus_52330_DDU	15.0
Exchange: Tree.t:locus_52330_DDU	15.0
Exchange: Tree.t:locus_52330_DDU	3.0
Wilson Balding: Tree.t:locus_52330_DDU	3.0
Delta Exchange: mutationRate.s:5-SNPs mutationRate.s:6-SNPs	200.0
Scale: kappa.s:6-SNPs Scale HKY transition-transversion parameter of partition s:locus_71356_DDU	1.0
Bit Flip: indicators.alltrees EBSP bitflip operator	3000.0
Sample Off Values: popSizes.alltrees indicators.alltrees EBSP indicator sampler	1500.0
Scale: popSizes.alltrees indicators.alltrees EBSP population sizes	1500.0
Up Down: popSizes.alltrees populationMean.alltrees Up/down scale substitution rates of EBSP prior and tree	500.0
Scale: Tree.t:locus_14126_DDU Scales all internal nodes for tree t:locus_14126_DDU	3.0
Scale: Tree.t:locus_14126_DDU Scales root node for tree t:locus_14126_DDU	3.0
Uniform: Tree.t:locus_14126_DDU Draws new internal node heights uniformally for tree t:locus_14126_DDU	30.0
Subtree Slide: Tree.t:locus_14126_DDU Performs subtree slide rearrangement of tree t:locus_14126_DDU	15.0
Exchange: Tree.t:locus_14126_DDU Narrow exchange performs local rearrangement of tree t:locus_14126_DDU	15.0
Exchange: Tree.t:locus_14126_DDU Wide exchange performs global rearrangement of tree t:locus_14126_DDU	3.0
Wilson Balding: Tree.t:locus_14126_DDU Performs Wilson-Balding global rearrangement of tree t:locus_14126_DDU	3.0
Scale: Tree.t:locus_8247_DDU Scales all internal nodes for tree t:locus_8247_DDU	3.0
Scale: Tree.t:locus_8247_DDU Scales root node for tree t:locus_8247_DDU	3.0
Uniform: Tree.t:locus_8247_DDU Draws new internal node heights uniformally for tree t:locus_8247_DDU	30.0
Subtree Slide: Tree.t:locus_8247_DDU Performs subtree slide rearrangement of tree t:locus_8247_DDU	15.0
Exchange: Tree.t:locus_8247_DDU Narrow exchange performs local rearrangement of tree t:locus_8247_DDU	15.0
Exchange: Tree.t:locus_8247_DDU Wide exchange performs global rearrangement of tree t:locus_8247_DDU	3.0
Wilson Balding: Tree.t:locus_8247_DDU Performs Wilson-Balding global rearrangement of tree t:locus_8247_DDU	3.0
Scale: Tree.t:locus_24508_DDU Scales all internal nodes for tree t:locus_24508_DDU	3.0
Scale: Tree.t:locus_24508_DDU Scales root node for tree t:locus_24508_DDU	3.0
Uniform: Tree.t:locus_24508_DDU Draws new internal node heights uniformally for tree t:locus_24508_DDU	30.0
Subtree Slide: Tree.t:locus_24508_DDU Performs subtree slide rearrangement of tree t:locus_24508_DDU	15.0

7 – MCMC

Set a chain length of 500,000,000. It should run quite fast so that you can easily increase it. Log the trace every 10,000, the screen every 50,000 and the EBSP logger every 5000.

Save the xml.

You can edit the xml file and remove all of the loggers of the trees. It will save a lot of space on your hard drive!

```
<logger id="treelog.t:king_red_snp4_loc40039" fileName="$(tree).trees"
logEvery="1000" mode="tree">
        <log id="TreeWithMetaDataLogger.t:king_red_snp4_loc40039"
spec="beast.evolution.tree.TreeWithMetaDataLogger"
tree="@Tree.t:king_red_snp4_loc40039"/>
    </logger>
```

You are now ready to run your Beast2 analysis. Do use several cores if you can (4 to 8 is fine), and use Beagle libraries if available (see the general Beast2 documentation for details). However, if running on a cluster, not try to use accelerated (GPU) partitions, as the overhead of dispatching computations will largely ruin the benefit of GPUs for these relatively simple likelihood calculations.

8 – Get EBSP analysis done.

Have a look at your run in Tracer. How does it look like? How much would you give of burn-in in percentage? Are your ESS ok? Then execute this and follow the instructions

```
java -cp path-to-dir/beast.jar beast.app.tools.EBSPAnalyser
```
where “path-to-dir” points to the directory where beast.jar is located.

You need to specify input and output file, burn-in percentage and if you want a linear, or a stepwise reconstruction (you may want to try both).

9 – Draw your plot.

Choose your favourite software, import the output file you just generated and draw your demographic reconstruction.

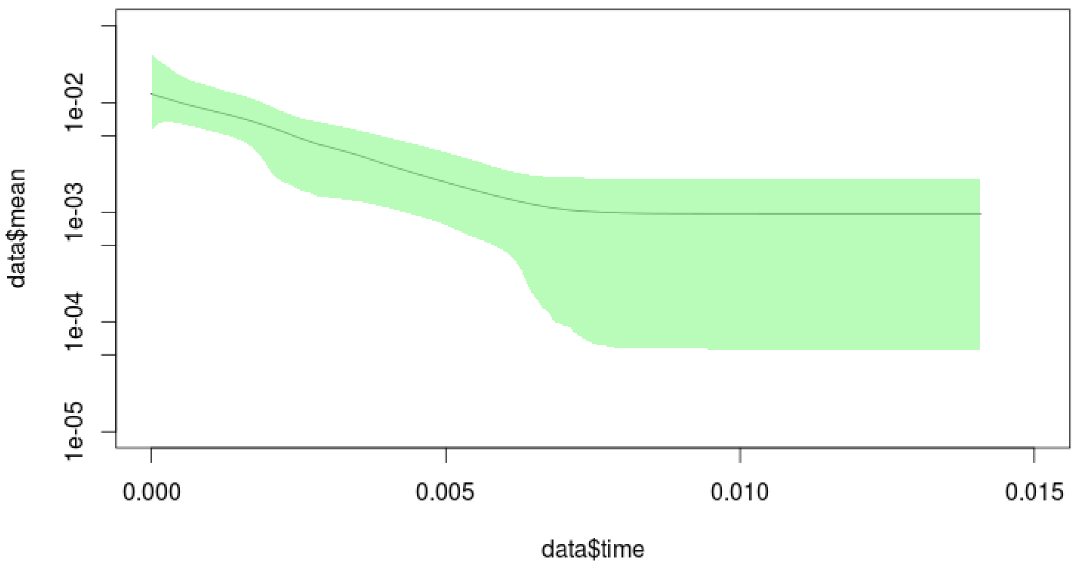
Examples with R

Remove the extra tab at the end of each line after the header in the output file of EBSPAnalyser!

```
read.table('EBSPAnalyser.out', header = T, sep="\t")->data
plot(data$time, data$mean, log="y", type="l", col=1, xlim=c(0,0.015),
ylim=c(0.00001,0.05))
polygon(c(data$time, rev(data$time)), c(data$X95HPD.upper,
rev(data$X95HPD.lower)), col = "#98FB98ac", border = NA)
```

Or, using the ggplot2 package:

```
require(ggplot2)
read.table('EBSPAnalyser.out', header=T)->data
ggplot(data, aes(x=time)) +
theme_bw() + scale_x_log10() +  scale_y_log10() +
geom_line(data=ebsp.r, aes(y=median), linetype='dashed') +
geom_ribbon(data=ebsp.r, aes(ymin= X95HPD.lower, ymax= X95HPD.upper,
 linetype='dashed', size=.2, alpha=.25)
```



EBSP using 49 loci with 4 and 5 SNPs sequenced in 8 individuals of king penguin, 2 alleles per individual.

10 - Calibrating your demographic reconstruction

Given the bias we generated when selecting highly variable RAD loci, it is not possible to directly apply an average genome-wide substitution rate to the EBSP analysis. However, there are ways to get a good guess. An ideal case would be to add loci/genes for which you have a robust estimate of the substitution rate. In the king penguin paper (Trucchi *et al* 2014), we added the mitochondrial Hyper Variable Region to our analysis. We first confirmed that we obtained the same pattern in the demographic reconstruction with or without the HVR, and then used the run with the HVR to get the final time-calibrated estimates (see above for the appropriate parametrisation in that case).

Citations

For BEAST2:
Bouckaert R, Heled J, Kuhnert D, Vaughan T, Wu C-H, Xie D, Suchard MA, Rambaut A, and Drummond AJ. BEAST 2: a software platform for Bayesian evolutionary analysis. *PLoS Computation Biology*, **10(4),** e1003537, Apr 2014. doi: 10.1371/journal.pcbi.1003537. URL http://dx.doi.org/10.1371/journal.pcbi.1003537.

For EBSP:
Heled J, Drummond AJ (2008) Bayesian inference of population size history from multiple loci. *BMC Evolutionary Biology*, **8**, 289.

For EBSP using RAD loci:
Trucchi E *et al.* (2014) King penguin demography since the last glaciation inferred from genome-wide data. *Proceedings of the Royal Society B: Biological Sciences*, **281**, 20140528.

References

Baird NA, Etter PD, Atwood TS *et al.* (2008) Rapid SNP Discovery and Genetic Mapping Using Sequenced RAD Markers. *PLoS ONE*, **3**, e3376.

Heled J, Drummond AJ (2008) Bayesian inference of population size history from multiple loci. *BMC Evolutionary Biology*, **8**, 289.

Trucchi E, Gratton P, Whittington JD *et al.* (2014) King penguin demography since the last glaciation inferred from genome-wide data. *Proceedings of the Royal Society B: Biological Sciences*, **281**, 20140528.