LFMM_for_LMK.Rmd

Tanya Lama

8/12/2020

Latent factor mixed modeling (LFMM) is used to detect specific loci with strong environmental associations suggestive of natural selection for local adaptation. Population genetic variation is influenced by demography, mating patterns, and natural selection, each of which is shaped by the environment. Theoretically, demographic changes and gene flow (“neutral” processes) affect genetic variation throughout the genome, whereas natural selection affects a relatively small number of genes. To disentangle which parts of the genome are under natural selection by the environment, we can use genome-wide data sets to identify SNPs/alleles that are strongly associated with environmental variables after accounting for the background association of genetic variation with the environment due to demographic patterns. These “outliers” are candidate loci that could be under the influence of natural selection for local adaptation along the environmental gradient. We will use latent factor mixed models (LFMM), which is a method that can handle individual-based data (rather than “population”-based) and is considered to be a powerful method for detecting loci under selection.

I discussed this with Brenna Forester last week. She suggested I could combine LFMM with GDM or another multivariate ordination method like redundancy analysis (RDA) (e.g. Forester et al. 2018). Brenna says that LFMM is more conservative than BayEnv2 or BayeScan, but GDM is even *more* conservative because it does a better job of partitioning variance between netural processes and actual selection. Here we’ll present what was done for LFMM

Refer to Frichot et al. (2013) for details of LFMM

#Climate data We’ll use WorldClim data which I have downloaded to the LFMM/climate_data in GeoTIFF (.tif) format. Now we will take the raw data and prepare it for our LFMM and RDA analyses in R

Note that instead of using climate variables directly, we ALSO ran LFMM using use principal components of the climate variables as uncorrelated, synthetic climate variables. We found that PCA decomposed variables into major components, PC1 loaded most of the temperature variables and PC2 loaded most of the precip variables and they accounted for 77% and 17% of the variation in our data respectively. I can’t remember where I had run that PC, but I will track down the code and add it below. When dealing with correlated climate variables (as is typically the case), this strategy may be preferrable, but in the code below we will simply use the climate variables themselves. I discussed this with Brenna and she didn’t favor either method. RDA and GDM actually use the climate variables directly rather than PC’s, so as long as you are addressing multicollinearity and mutiple-testing, you are OK.

#Using PCA to decompose environmental variables into synthetic variables for LFMM

```
#insert here
```

#Assessing population strcuture LFMM accounts for background associations of genetic variation with environmental variation using latent factors to model unobserved variation. A key step is determining the number of latent factors to include, as this affects power. It’s advised to start with the number of clusters (K) as the initial value. We have estimated K=3 using PCA and sNMF in the LEA package in R (this is presented in other scripts and we’re confident on the K=3 finding).

#Latent factor mixed modeling (LFMM)

We’ll use temperature seasonality (bio4 = tseas), minimum temperature of coldest month (bio6 = tmin), precipitation seasonality (bio15 = pseas), and precipitation of the coldest quarter (bio19 = Pdry)

We’re ready to run LFMM! We can run it simultaneously for all the climate variables contained in our climate data table (default) or one at a time. Run the following basic command:

Note that the number of iterations here is lower than we’ve run just to try and get RMarkdown to publish our code. We should run 10k iterations with a burning of at least 500. Note K=3 in agreement with the results of our clustering analyses. Also note that we are using an LD-pruned SNP set from LEA.

```
project = NULL
project = lfmm("LD_pruned_snp.lfmm", "lynx_clim.env", K = 3, repetitions = 3, CPU = 16,
 iterations = 10, burnin = 5, project = "new")
```

```
## The project is saved into :
##  LD_pruned_snp_lynx_clim.lfmmProject
##
## To load the project, use:
##  project = load.lfmmProject("LD_pruned_snp_lynx_clim.lfmmProject")
##
## To remove the project, use:
##  remove.lfmmProject("LD_pruned_snp_lynx_clim.lfmmProject")
##
## [1] "********************************"
## [1] "* K = 3  repetition 1  d = 1   *"
## [1] "********************************"
## Summary of the options:
##
##         -n (number of individuals)      61
##         -L (number of loci)             53265
##         -K (number of latent factors)   3
##         -o (output file)                LD_pruned_snp_lynx_clim.lfmm/K3/run1/LD_prune
d_snp_r1
##         -i (number of iterations)       10
##         -b (burnin)                     5
##         -s (seed random init)           1734834813
##         -p (number of processes (CPU))  16
##         -x (genotype file)              LD_pruned_snp.lfmm
##         -v (variable file)              lynx_clim.env
##         -D (number of covariables)      5
##         -d (the dth covariable)         1
##
## Read variable file:
##      lynx_clim.env       OK.
##
## Read genotype file:
##      LD_pruned_snp.lfmm      OK.
##
## <<<<
##   Analyse for variable 1
##
##      Start of the Gibbs Sampler algorithm.
##
##  [                                                                           ]
##  [===========================================================================]
##
##      End of the Gibbs Sampler algorithm.
##
##  ED:3249306.119   DIC: 3249279.211
##
##  The statistics for the run are registered in:
##          LD_pruned_snp_lynx_clim.lfmm/K3/run1/LD_pruned_snp_r1_s1.3.dic.
##
##  The zscores for variable 1 are registered in:
##          LD_pruned_snp_lynx_clim.lfmm/K3/run1/LD_pruned_snp_r1_s1.3.zscore.
##  The columns are: zscores, -log10(p-values), p-values.
##
```

```
##  -------------------------
##  The execution for variable 1 worked without error.
## >>>>
##
## The project is saved into :
##  LD_pruned_snp_lynx_clim.lfmmProject
##
## To load the project, use:
##  project = load.lfmmProject("LD_pruned_snp_lynx_clim.lfmmProject")
##
## To remove the project, use:
##  remove.lfmmProject("LD_pruned_snp_lynx_clim.lfmmProject")
##
## [1] "********************************"
## [1] "* K = 3  repetition 1  d = 2   *"
## [1] "********************************"
## Summary of the options:
##
##         -n (number of individuals)      61
##         -L (number of loci)             53265
##         -K (number of latent factors)   3
##         -o (output file)                LD_pruned_snp_lynx_clim.lfmm/K3/run2/LD_prune
d_snp_r2
##         -i (number of iterations)       10
##         -b (burnin)                     5
##         -s (seed random init)           1734834813
##         -p (number of processes (CPU))  16
##         -x (genotype file)              LD_pruned_snp.lfmm
##         -v (variable file)              lynx_clim.env
##         -D (number of covariables)      5
##         -d (the dth covariable)         2
##
## Read variable file:
##      lynx_clim.env       OK.
##
## Read genotype file:
##      LD_pruned_snp.lfmm      OK.
##
## <<<<
##   Analyse for variable 2
##
##      Start of the Gibbs Sampler algorithm.
##
##  [                                                                           ]
##  [===========================================================================]
##
##      End of the Gibbs Sampler algorithm.
##
##  ED:3249527.362   DIC: 3249470.636
##
##  The statistics for the run are registered in:
##          LD_pruned_snp_lynx_clim.lfmm/K3/run2/LD_pruned_snp_r2_s2.3.dic.
##
##  The zscores for variable 2 are registered in:
```

```
##          LD_pruned_snp_lynx_clim.lfmm/K3/run2/LD_pruned_snp_r2_s2.3.zscore.
##  The columns are: zscores, -log10(p-values), p-values.
## 
##  -------------------------
##  The execution for variable 2 worked without error.
## >>>>
## 
## The project is saved into :
##  LD_pruned_snp_lynx_clim.lfmmProject 
## 
## To load the project, use:
##  project = load.lfmmProject("LD_pruned_snp_lynx_clim.lfmmProject")
## 
## To remove the project, use:
##  remove.lfmmProject("LD_pruned_snp_lynx_clim.lfmmProject")
## 
## [1] "********************************"
## [1] "* K = 3  repetition 1  d = 3   *"
## [1] "********************************"
## Summary of the options:
## 
##         -n (number of individuals)      61
##         -L (number of loci)             53265
##         -K (number of latent factors)   3
##         -o (output file)                LD_pruned_snp_lynx_clim.lfmm/K3/run3/LD_prune
d_snp_r3
##         -i (number of iterations)       10
##         -b (burnin)                     5
##         -s (seed random init)           13830345240942705277
##         -p (number of processes (CPU))  16
##         -x (genotype file)              LD_pruned_snp.lfmm
##         -v (variable file)              lynx_clim.env
##         -D (number of covariables)      5
##         -d (the dth covariable)         3
## 
## Read variable file:
##      lynx_clim.env       OK.
## 
## Read genotype file:
##      LD_pruned_snp.lfmm      OK.
## 
## <<<<
##   Analyse for variable 3
## 
##      Start of the Gibbs Sampler algorithm.
## 
##  [                                                                           ]
##  [===========================================================================]
## 
##      End of the Gibbs Sampler algorithm.
## 
##  ED:3249597.479   DIC: 3249521.361 
## 
##  The statistics for the run are registered in:
```

```
##          LD_pruned_snp_lynx_clim.lfmm/K3/run3/LD_pruned_snp_r3_s3.3.dic.
##
##  The zscores for variable 3 are registered in:
##          LD_pruned_snp_lynx_clim.lfmm/K3/run3/LD_pruned_snp_r3_s3.3.zscore.
##  The columns are: zscores, -log10(p-values), p-values.
##
##  -------------------------
##  The execution for variable 3 worked without error.
## >>>>
##
## The project is saved into :
##  LD_pruned_snp_lynx_clim.lfmmProject
##
## To load the project, use:
##  project = load.lfmmProject("LD_pruned_snp_lynx_clim.lfmmProject")
##
## To remove the project, use:
##  remove.lfmmProject("LD_pruned_snp_lynx_clim.lfmmProject")
##
## [1] "********************************"
## [1] "* K = 3  repetition 1  d = 4   *"
## [1] "********************************"
## Summary of the options:
##
##         -n (number of individuals)      61
##         -L (number of loci)             53265
##         -K (number of latent factors)   3
##         -o (output file)                LD_pruned_snp_lynx_clim.lfmm/K3/run4/LD_prune
d_snp_r4
##         -i (number of iterations)       10
##         -b (burnin)                     5
##         -s (seed random init)           1734834813
##         -p (number of processes (CPU))  16
##         -x (genotype file)              LD_pruned_snp.lfmm
##         -v (variable file)              lynx_clim.env
##         -D (number of covariables)      5
##         -d (the dth covariable)         4
##
## Read variable file:
##      lynx_clim.env       OK.
##
## Read genotype file:
##      LD_pruned_snp.lfmm      OK.
##
## <<<<
##   Analyse for variable 4
##
##      Start of the Gibbs Sampler algorithm.
##
##  [                                                                            ]
##  [============================================================================]
##
##      End of the Gibbs Sampler algorithm.
##
```

```
##  ED:3249613.871   DIC: 3249534.513
##
##  The statistics for the run are registered in:
##          LD_pruned_snp_lynx_clim.lfmm/K3/run4/LD_pruned_snp_r4_s4.3.dic.
##
##  The zscores for variable 4 are registered in:
##          LD_pruned_snp_lynx_clim.lfmm/K3/run4/LD_pruned_snp_r4_s4.3.zscore.
##  The columns are: zscores, -log10(p-values), p-values.
##
##  -------------------------
##  The execution for variable 4 worked without error.
## >>>>
##
## The project is saved into :
##  LD_pruned_snp_lynx_clim.lfmmProject
##
## To load the project, use:
##  project = load.lfmmProject("LD_pruned_snp_lynx_clim.lfmmProject")
##
## To remove the project, use:
##  remove.lfmmProject("LD_pruned_snp_lynx_clim.lfmmProject")
##
## [1] "********************************"
## [1] "* K = 3  repetition 1  d = 5   *"
## [1] "********************************"
## Summary of the options:
##
##         -n (number of individuals)      61
##         -L (number of loci)             53265
##         -K (number of latent factors)   3
##         -o (output file)                LD_pruned_snp_lynx_clim.lfmm/K3/run5/LD_prune
d_snp_r5
##         -i (number of iterations)       10
##         -b (burnin)                     5
##         -s (seed random init)           1734834813
##         -p (number of processes (CPU))  16
##         -x (genotype file)              LD_pruned_snp.lfmm
##         -v (variable file)              lynx_clim.env
##         -D (number of covariables)      5
##         -d (the dth covariable)         5
##
## Read variable file:
##      lynx_clim.env       OK.
##
## Read genotype file:
##      LD_pruned_snp.lfmm      OK.
##
## <<<<
##   Analyse for variable 5
##
##      Start of the Gibbs Sampler algorithm.
##
##  [                                                                           ]
##  [===========================================================================]
```

```
##
##      End of the Gibbs Sampler algorithm.
##
##  ED:3249515.008   DIC: 3249459.638
##
##  The statistics for the run are registered in:
##          LD_pruned_snp_lynx_clim.lfmm/K3/run5/LD_pruned_snp_r5_s5.3.dic.
##
##  The zscores for variable 5 are registered in:
##          LD_pruned_snp_lynx_clim.lfmm/K3/run5/LD_pruned_snp_r5_s5.3.zscore.
##  The columns are: zscores, -log10(p-values), p-values.
##
##  -------------------------
##  The execution for variable 5 worked without error.
## >>>>
##
## The project is saved into :
##  LD_pruned_snp_lynx_clim.lfmmProject
##
## To load the project, use:
##  project = load.lfmmProject("LD_pruned_snp_lynx_clim.lfmmProject")
##
## To remove the project, use:
##  remove.lfmmProject("LD_pruned_snp_lynx_clim.lfmmProject")
##
## [1] "********************************"
## [1] "* K = 3  repetition 2  d = 1   *"
## [1] "********************************"
## Summary of the options:
##
##         -n (number of individuals)      61
##         -L (number of loci)             53265
##         -K (number of latent factors)   3
##         -o (output file)                LD_pruned_snp_lynx_clim.lfmm/K3/run6/LD_prune
d_snp_r6
##         -i (number of iterations)       10
##         -b (burnin)                     5
##         -s (seed random init)           2109302058
##         -p (number of processes (CPU))  16
##         -x (genotype file)              LD_pruned_snp.lfmm
##         -v (variable file)              lynx_clim.env
##         -D (number of covariables)      5
##         -d (the dth covariable)         1
##
## Read variable file:
##      lynx_clim.env       OK.
##
## Read genotype file:
##      LD_pruned_snp.lfmm      OK.
##
## <<<<
##   Analyse for variable 1
##
##      Start of the Gibbs Sampler algorithm.
```

```
##
##  [                                                                          ]
##  [==========================================================================]
##
##      End of the Gibbs Sampler algorithm.
##
##  ED:3249556.713   DIC:  3249492.5
##
##  The statistics for the run are registered in:
##          LD_pruned_snp_lynx_clim.lfmm/K3/run6/LD_pruned_snp_r6_s1.3.dic.
##
##  The zscores for variable 1 are registered in:
##          LD_pruned_snp_lynx_clim.lfmm/K3/run6/LD_pruned_snp_r6_s1.3.zscore.
##  The columns are: zscores, -log10(p-values), p-values.
##
##  -------------------------
##  The execution for variable 1 worked without error.
## >>>>
##
## The project is saved into :
##  LD_pruned_snp_lynx_clim.lfmmProject
##
## To load the project, use:
##  project = load.lfmmProject("LD_pruned_snp_lynx_clim.lfmmProject")
##
## To remove the project, use:
##  remove.lfmmProject("LD_pruned_snp_lynx_clim.lfmmProject")
##
## [1] "********************************"
## [1] "* K = 3  repetition 2  d = 2   *"
## [1] "********************************"
## Summary of the options:
##
##         -n (number of individuals)      61
##         -L (number of loci)             53265
##         -K (number of latent factors)   3
##         -o (output file)                LD_pruned_snp_lynx_clim.lfmm/K3/run7/LD_prune
d_snp_r7
##         -i (number of iterations)       10
##         -b (burnin)                     5
##         -s (seed random init)           734710387801941290
##         -p (number of processes (CPU))  16
##         -x (genotype file)              LD_pruned_snp.lfmm
##         -v (variable file)              lynx_clim.env
##         -D (number of covariables)      5
##         -d (the dth covariable)         2
##
## Read variable file:
##      lynx_clim.env       OK.
##
## Read genotype file:
##      LD_pruned_snp.lfmm      OK.
##
## <<<<
##   Analyse for variable 2
##
##      Start of the Gibbs Sampler algorithm.
##
##  [                                                                           ]
##  [===========================================================================]
##
##      End of the Gibbs Sampler algorithm.
##
##  ED:3249517.065   DIC: 3249454.075 
##
##  The statistics for the run are registered in:
##          LD_pruned_snp_lynx_clim.lfmm/K3/run7/LD_pruned_snp_r7_s2.3.dic.
##
##  The zscores for variable 2 are registered in:
##          LD_pruned_snp_lynx_clim.lfmm/K3/run7/LD_pruned_snp_r7_s2.3.zscore.
##  The columns are: zscores, -log10(p-values), p-values.
##
##  -------------------------
##  The execution for variable 2 worked without error.
## >>>>
##
## The project is saved into :
##  LD_pruned_snp_lynx_clim.lfmmProject 
##
## To load the project, use:
##  project = load.lfmmProject("LD_pruned_snp_lynx_clim.lfmmProject")
##
## To remove the project, use:
##  remove.lfmmProject("LD_pruned_snp_lynx_clim.lfmmProject")
##
## [1] "********************************"
## [1] "* K = 3  repetition 2  d = 3   *"
## [1] "********************************"
## Summary of the options:
##
##         -n (number of individuals)      61
##         -L (number of loci)             53265
##         -K (number of latent factors)   3
##         -o (output file)                LD_pruned_snp_lynx_clim.lfmm/K3/run8/LD_prune
d_snp_r8
##         -i (number of iterations)       10
##         -b (burnin)                     5
##         -s (seed random init)           2109302058
##         -p (number of processes (CPU))  16
##         -x (genotype file)              LD_pruned_snp.lfmm
##         -v (variable file)              lynx_clim.env
##         -D (number of covariables)      5
##         -d (the dth covariable)         3
##
## Read variable file:
##      lynx_clim.env       OK.
##
## Read genotype file:
```

```
##      LD_pruned_snp.lfmm      OK.
##
## <<<<
##   Analyse for variable 3
##
##      Start of the Gibbs Sampler algorithm.
##
##  [                                                                           ]
##  [===========================================================================]
##
##      End of the Gibbs Sampler algorithm.
##
##  ED:3249566.127   DIC: 3249505.064
##
##  The statistics for the run are registered in:
##          LD_pruned_snp_lynx_clim.lfmm/K3/run8/LD_pruned_snp_r8_s3.3.dic.
##
##  The zscores for variable 3 are registered in:
##          LD_pruned_snp_lynx_clim.lfmm/K3/run8/LD_pruned_snp_r8_s3.3.zscore.
##  The columns are: zscores, -log10(p-values), p-values.
##
##  -------------------------
##  The execution for variable 3 worked without error.
## >>>>
##
## The project is saved into :
##  LD_pruned_snp_lynx_clim.lfmmProject
##
## To load the project, use:
##  project = load.lfmmProject("LD_pruned_snp_lynx_clim.lfmmProject")
##
## To remove the project, use:
##  remove.lfmmProject("LD_pruned_snp_lynx_clim.lfmmProject")
##
## [1] "********************************"
## [1] "* K = 3  repetition 2  d = 4   *"
## [1] "********************************"
## Summary of the options:
##
##         -n (number of individuals)      61
##         -L (number of loci)             53265
##         -K (number of latent factors)   3
##         -o (output file)                LD_pruned_snp_lynx_clim.lfmm/K3/run9/LD_prune
d_snp_r9
##         -i (number of iterations)       10
##         -b (burnin)                     5
##         -s (seed random init)           4615916790956516650
##         -p (number of processes (CPU))  16
##         -x (genotype file)              LD_pruned_snp.lfmm
##         -v (variable file)              lynx_clim.env
##         -D (number of covariables)      5
##         -d (the dth covariable)         4
##
## Read variable file:
```

```
##      lynx_clim.env       OK.
##
## Read genotype file:
##      LD_pruned_snp.lfmm      OK.
##
## <<<<
##   Analyse for variable 4
##
##      Start of the Gibbs Sampler algorithm.
##
##  [                                                                           ]
##  [===========================================================================]
##
##      End of the Gibbs Sampler algorithm.
##
##  ED:3249564.264   DIC: 3249499.386
##
##  The statistics for the run are registered in:
##          LD_pruned_snp_lynx_clim.lfmm/K3/run9/LD_pruned_snp_r9_s4.3.dic.
##
##  The zscores for variable 4 are registered in:
##          LD_pruned_snp_lynx_clim.lfmm/K3/run9/LD_pruned_snp_r9_s4.3.zscore.
##  The columns are: zscores, -log10(p-values), p-values.
##
##  -------------------------
##  The execution for variable 4 worked without error.
## >>>>
##
## The project is saved into :
##  LD_pruned_snp_lynx_clim.lfmmProject
##
## To load the project, use:
##  project = load.lfmmProject("LD_pruned_snp_lynx_clim.lfmmProject")
##
## To remove the project, use:
##  remove.lfmmProject("LD_pruned_snp_lynx_clim.lfmmProject")
##
## [1] "********************************"
## [1] "* K = 3  repetition 2  d = 5   *"
## [1] "********************************"
## Summary of the options:
##
##         -n (number of individuals)      61
##         -L (number of loci)             53265
##         -K (number of latent factors)   3
##         -o (output file)                LD_pruned_snp_lynx_clim.lfmm/K3/run10/LD_prun
ed_snp_r10
##         -i (number of iterations)       10
##         -b (burnin)                     5
##         -s (seed random init)           2109302058
##         -p (number of processes (CPU))  16
##         -x (genotype file)              LD_pruned_snp.lfmm
##         -v (variable file)              lynx_clim.env
##         -D (number of covariables)      5
##          -d (the dth covariable)         5
##
## Read variable file:
##      lynx_clim.env       OK.
##
## Read genotype file:
##      LD_pruned_snp.lfmm      OK.
##
## <<<<
##   Analyse for variable 5
##
##      Start of the Gibbs Sampler algorithm.
##
##  [                                                                           ]
##  [===========================================================================]
##
##      End of the Gibbs Sampler algorithm.
##
##  ED:3249439.093   DIC: 3249376.595
##
##  The statistics for the run are registered in:
##          LD_pruned_snp_lynx_clim.lfmm/K3/run10/LD_pruned_snp_r10_s5.3.dic.
##
##  The zscores for variable 5 are registered in:
##          LD_pruned_snp_lynx_clim.lfmm/K3/run10/LD_pruned_snp_r10_s5.3.zscore.
##  The columns are: zscores, -log10(p-values), p-values.
##
##  -------------------------
##  The execution for variable 5 worked without error.
## >>>>
##
## The project is saved into :
##  LD_pruned_snp_lynx_clim.lfmmProject
##
## To load the project, use:
##  project = load.lfmmProject("LD_pruned_snp_lynx_clim.lfmmProject")
##
## To remove the project, use:
##  remove.lfmmProject("LD_pruned_snp_lynx_clim.lfmmProject")
##
## [1] "********************************"
## [1] "* K = 3  repetition 3  d = 1   *"
## [1] "********************************"
## Summary of the options:
##
##          -n (number of individuals)      61
##          -L (number of loci)             53265
##          -K (number of latent factors)   3
##          -o (output file)                LD_pruned_snp_lynx_clim.lfmm/K3/run11/LD_prun
ed_snp_r11
##          -i (number of iterations)       10
##          -b (burnin)                     5
##          -s (seed random init)           240139778
##          -p (number of processes (CPU))  16
##         -x (genotype file)               LD_pruned_snp.lfmm
##         -v (variable file)               lynx_clim.env
##         -D (number of covariables)       5
##         -d (the dth covariable)          1
## 
## Read variable file:
##      lynx_clim.env       OK.
## 
## Read genotype file:
##      LD_pruned_snp.lfmm      OK.
## 
## <<<<
##   Analyse for variable 1
## 
##      Start of the Gibbs Sampler algorithm.
## 
##  [                                                                           ]
##  [===========================================================================]
## 
##      End of the Gibbs Sampler algorithm.
## 
##  ED:3249535.327   DIC: 3249483.479 
## 
##  The statistics for the run are registered in:
##          LD_pruned_snp_lynx_clim.lfmm/K3/run11/LD_pruned_snp_r11_s1.3.dic.
## 
##  The zscores for variable 1 are registered in:
##          LD_pruned_snp_lynx_clim.lfmm/K3/run11/LD_pruned_snp_r11_s1.3.zscore.
##  The columns are: zscores, -log10(p-values), p-values.
## 
##  -------------------------
##  The execution for variable 1 worked without error.
## >>>>
## 
## The project is saved into :
##  LD_pruned_snp_lynx_clim.lfmmProject 
## 
## To load the project, use:
##  project = load.lfmmProject("LD_pruned_snp_lynx_clim.lfmmProject")
## 
## To remove the project, use:
##  remove.lfmmProject("LD_pruned_snp_lynx_clim.lfmmProject")
## 
## [1] "********************************"
## [1] "* K = 3  repetition 3  d = 2   *"
## [1] "********************************"
## Summary of the options:
## 
##         -n (number of individuals)      61
##         -L (number of loci)             53265
##         -K (number of latent factors)   3
##         -o (output file)                LD_pruned_snp_lynx_clim.lfmm/K3/run12/LD_prun
ed_snp_r12
##         -i (number of iterations)       10
##         -b (burnin)                     5
##         -s (seed random init)           240139778
##         -p (number of processes (CPU))  16
##         -x (genotype file)              LD_pruned_snp.lfmm
##         -v (variable file)              lynx_clim.env
##         -D (number of covariables)      5
##         -d (the dth covariable)         2
##
## Read variable file:
##      lynx_clim.env       OK.
##
## Read genotype file:
##      LD_pruned_snp.lfmm      OK.
##
## <<<<
##   Analyse for variable 2
##
##      Start of the Gibbs Sampler algorithm.
##
##  [                                                                           ]
##  [===========================================================================]
##
##      End of the Gibbs Sampler algorithm.
##
##  ED:3249674.556   DIC: 3249573.553
##
##  The statistics for the run are registered in:
##          LD_pruned_snp_lynx_clim.lfmm/K3/run12/LD_pruned_snp_r12_s2.3.dic.
##
##  The zscores for variable 2 are registered in:
##          LD_pruned_snp_lynx_clim.lfmm/K3/run12/LD_pruned_snp_r12_s2.3.zscore.
##  The columns are: zscores, -log10(p-values), p-values.
##
##  -------------------------
##  The execution for variable 2 worked without error.
## >>>>
##
## The project is saved into :
##  LD_pruned_snp_lynx_clim.lfmmProject
##
## To load the project, use:
##  project = load.lfmmProject("LD_pruned_snp_lynx_clim.lfmmProject")
##
## To remove the project, use:
##  remove.lfmmProject("LD_pruned_snp_lynx_clim.lfmmProject")
##
## [1] "********************************"
## [1] "* K = 3  repetition 3  d = 3   *"
## [1] "********************************"
## Summary of the options:
##
##         -n (number of individuals)      61
##         -L (number of loci)             53265
##         -K (number of latent factors)   3
##         -o (output file)                LD_pruned_snp_lynx_clim.lfmm/K3/run13/LD_prun
ed_snp_r13
##         -i (number of iterations)       10
##         -b (burnin)                     5
##         -s (seed random init)           240139778
##         -p (number of processes (CPU))  16
##         -x (genotype file)              LD_pruned_snp.lfmm
##         -v (variable file)              lynx_clim.env
##         -D (number of covariables)      5
##         -d (the dth covariable)         3
## 
## Read variable file:
##      lynx_clim.env       OK.
## 
## Read genotype file:
##      LD_pruned_snp.lfmm      OK.
## 
## <<<<
##   Analyse for variable 3
## 
##      Start of the Gibbs Sampler algorithm.
## 
##  [                                                                            ]
##  [============================================================================]
## 
##      End of the Gibbs Sampler algorithm.
## 
##  ED:3249570.873   DIC: 3249499.063 
## 
##  The statistics for the run are registered in:
##          LD_pruned_snp_lynx_clim.lfmm/K3/run13/LD_pruned_snp_r13_s3.3.dic.
## 
##  The zscores for variable 3 are registered in:
##          LD_pruned_snp_lynx_clim.lfmm/K3/run13/LD_pruned_snp_r13_s3.3.zscore.
##  The columns are: zscores, -log10(p-values), p-values.
## 
##  -------------------------
##  The execution for variable 3 worked without error.
## >>>>
## 
## The project is saved into :
##  LD_pruned_snp_lynx_clim.lfmmProject 
## 
## To load the project, use:
##  project = load.lfmmProject("LD_pruned_snp_lynx_clim.lfmmProject")
## 
## To remove the project, use:
##  remove.lfmmProject("LD_pruned_snp_lynx_clim.lfmmProject")
## 
## [1] "********************************"
## [1] "* K = 3  repetition 3  d = 4   *"
## [1] "********************************"
## Summary of the options:
## 
```

```
##         -n (number of individuals)      61
##         -L (number of loci)             53265
##         -K (number of latent factors)   3
##         -o (output file)                LD_pruned_snp_lynx_clim.lfmm/K3/run14/LD_pruned_snp_r14
##         -i (number of iterations)       10
##         -b (burnin)                     5
##         -s (seed random init)           4535107074
##         -p (number of processes (CPU))  16
##         -x (genotype file)              LD_pruned_snp.lfmm
##         -v (variable file)              lynx_clim.env
##         -D (number of covariables)      5
##         -d (the dth covariable)         4
##
## Read variable file:
##      lynx_clim.env       OK.
##
## Read genotype file:
##      LD_pruned_snp.lfmm      OK.
##
## <<<<
##   Analyse for variable 4
##
##      Start of the Gibbs Sampler algorithm.
##
##  [                                                                           ]
##  [===========================================================================]
##
##      End of the Gibbs Sampler algorithm.
##
##  ED:3249521.387   DIC: 3249455.413
##
##  The statistics for the run are registered in:
##          LD_pruned_snp_lynx_clim.lfmm/K3/run14/LD_pruned_snp_r14_s4.3.dic.
##
##  The zscores for variable 4 are registered in:
##          LD_pruned_snp_lynx_clim.lfmm/K3/run14/LD_pruned_snp_r14_s4.3.zscore.
##  The columns are: zscores, -log10(p-values), p-values.
##
##  -------------------------
##  The execution for variable 4 worked without error.
## >>>>
##
## The project is saved into :
##  LD_pruned_snp_lynx_clim.lfmmProject
##
## To load the project, use:
##  project = load.lfmmProject("LD_pruned_snp_lynx_clim.lfmmProject")
##
## To remove the project, use:
##  remove.lfmmProject("LD_pruned_snp_lynx_clim.lfmmProject")
##
## [1] "********************************"
## [1] "* K = 3  repetition 3  d = 5   *"
## [1] "********************************"
## Summary of the options:
##
##         -n (number of individuals)      61
##         -L (number of loci)             53265
##         -K (number of latent factors)   3
##         -o (output file)                LD_pruned_snp_lynx_clim.lfmm/K3/run15/LD_prun
ed_snp_r15
##         -i (number of iterations)       10
##         -b (burnin)                     5
##         -s (seed random init)           240139778
##         -p (number of processes (CPU))  16
##         -x (genotype file)              LD_pruned_snp.lfmm
##         -v (variable file)              lynx_clim.env
##         -D (number of covariables)      5
##         -d (the dth covariable)         5
##
## Read variable file:
##      lynx_clim.env       OK.
##
## Read genotype file:
##      LD_pruned_snp.lfmm      OK.
##
## <<<<
##   Analyse for variable 5
##
##      Start of the Gibbs Sampler algorithm.
##
##  [                                                                            ]
##  [============================================================================]
##
##      End of the Gibbs Sampler algorithm.
##
##  ED:3249703.896   DIC: 3249611.614
##
##  The statistics for the run are registered in:
##          LD_pruned_snp_lynx_clim.lfmm/K3/run15/LD_pruned_snp_r15_s5.3.dic.
##
##  The zscores for variable 5 are registered in:
##          LD_pruned_snp_lynx_clim.lfmm/K3/run15/LD_pruned_snp_r15_s5.3.zscore.
##  The columns are: zscores, -log10(p-values), p-values.
##
##  -------------------------
##  The execution for variable 5 worked without error.
## >>>>
##
## The project is saved into :
##  LD_pruned_snp_lynx_clim.lfmmProject
##
## To load the project, use:
##  project = load.lfmmProject("LD_pruned_snp_lynx_clim.lfmmProject")
##
## To remove the project, use:
##  remove.lfmmProject("LD_pruned_snp_lynx_clim.lfmmProject")
## LFMM uses a very naive imputation method which has low power when
##              genotypes are missing: See impute() for a better imputation method.
```

This specifies the file name with the SNP input data in 012 format, the K number of clusters inferred (above) and the number of reps for the model to run, n processors to use, n iterations, and burn-in. Each can be adjusted, but you want to run 5-10 reps and increase the n iterations and burnin 10-fold (at least).

When LFMM is done running, we combine the data from >3 repetitions and compute new calibrated P-values. To do that, first extract the z-scores for all reps for a given climate variable, then take the median. This can be done using the LEA function z.scores and the base function apply to take the median at each locus. Here is an example for association tests with Pdry.

```
z.pdry = z.scores(project, K = 3, d = 1)
z.pdry <- apply(z.pdry, 1, median)
```

Next, we need to calculate λ (the “genomic inflation factor”), which is commonly used for calibration of P-values. However, it is often considered too conservative, so some suggest using a value lower than λ for the calibration. λ is calculated from the median of the median z-scores (from above) and a χ2 distribution for each set of associations:

```
lambda.pdry = median(z.pdry^2)/qchisq(0.5, df = 1)
lambda.pdry
```

```
## [1] 0.6392084
```

The calibrated or “adjusted” P-values are then calculated as follows:

```
p.pdry.adj = pchisq(z.pdry^2/lambda.pdry, df = 1, lower = FALSE)
```

Now, repeat this correction procedure with the other three climate variables. Note that the value of d changes below to get the results for the second climate variable, which is pseas. Be sure to change d accordingly to retrieve results for each climate variable.

```
z.pseas = z.scores(project, K = 3, d = 2)
z.pseas <- apply(z.pseas, 1, median)
lambda.pseas = median(z.pseas^2)/qchisq(0.5, df = 1)
p.pseas.adj = pchisq(z.pseas^2/lambda.pseas, df = 1, lower = FALSE)

z.tmin = z.scores(project, K = 3, d = 3)
z.tmin <- apply(z.tmin, 1, median)
lambda.tmin = median(z.tmin^2)/qchisq(0.5, df = 1)
p.tmin.adj = pchisq(z.tmin^2/lambda.tmin, df = 1, lower = FALSE)

z.tseas = z.scores(project, K = 3, d = 4)
z.tseas <- apply(z.tseas, 1, median)
lambda.tseas = median(z.tseas^2)/qchisq(0.5, df = 1)
p.tseas.adj = pchisq(z.tseas^2/lambda.tseas, df = 1, lower = FALSE)

z.bio6 = z.scores(project, K = 3, d = 5)
z.bio6 <- apply(z.bio6, 1, median)
lambda.bio6 = median(z.bio6^2)/qchisq(0.5, df = 1)
p.bio6.adj = pchisq(z.bio6^2/lambda.bio6, df = 1, lower = FALSE)
```

To confirm that the model is behaving well with the K we chose and the adjustments to P, we need to inspect histograms of the P-values. The "best" K and proper calibration value will lead to histograms of P-values that are flat, except perhaps with an elevated frequency of very low P-values, representing the outliers. We can make all the histograms in a multi-paneled plot very simply with the base hist function. #hmmm

How do these look?: Flat, with a slight elevation at the lowest pvalues. Two of our temperature variables have taken on a slightly different shape. Refer back to the LEA/LFMM manual for guidance if you're unsure. If they suggest an overly conservative (right skew) or overly liberal (left skew) model/calibration (neither, really), then we could repeat the analysis with a lower or higher value of K, respectively, or if there is slight right skew we might consider substituting a value lower than λ to manually calibrate. Keep in mind that we are working with a very small sample size in this analysis, so the patterns may not be typical. These plots looked ok, and we tried this with K=2 and K=1. A lower K performed better, but I need to think more about that before I choose a different n K, so we're going to stick with our chosen K value of 3 for now and proceed. Also, the distribution for our two variables in question was neither left nor right skewed, but pretty normal in terms of distribution…unsure what to do with that. Perhaps this is more incentive to use the principal component approach, rather than raw variables for LFMM.

Once we are convinced that the model is behaving well, we can move on. But, there is one final adjustment we need to make. We need to correct for multiple testing. We performed thousands of statistical tests (one per locus per climate variable), so many tests will appear significant by chance. The most common method of multiple testing correction is the false discovery rate (FDR) method of Benjamini and Hochberg (instead of Bonferroni correction, for example). In this process, we will adjust the P-values to Q-values. This correction can be easily implemented with the library qvalue.

```
q.pdry<-qvalue(p.pdry.adj)$qvalues
sum(q.pdry<0.01)
```

```
## [1] 1027
```

```
q.pseas<-qvalue(p.pseas.adj)$qvalues
sum(q.pseas<0.01)
```

```
## [1] 1423
```

```
q.tmin<-qvalue(p.tmin.adj)$qvalues
sum(q.tmin<0.01)
```

```
## [1] 2
```

```
q.tseas<-qvalue(p.tseas.adj)$qvalues
sum(q.tseas<0.01)
```

```
## [1] 3343
```

```
q.bio6<-qvalue(p.bio6.adj)$qvalues
sum(q.bio6<0.01)
```

```
## [1] 1585
```

#We found that roughly 9% of SNPs were significantly (<0.01) associated with at least one climate variable.

For perspective, how many candidate outliers do we get, if we remove the qvalue multiple testing correction? How does the number of significant tests based on Q-values (e.g., sum(q.pdry<0.05)) compare to the P-values (e.g., sum(p.pdry.adj<0.05))?

```
sum(p.pdry.adj<0.01)
```

```
## [1] 3270
```

```
sum(p.pseas.adj<0.01)
```

```
## [1] 3820
```

```
sum(p.tmin.adj<0.01)
```

```
## [1] 121
```

```
sum(p.tseas.adj<0.01)
```

```
## [1] 4733
```

```
sum(p.bio6.adj<0.01)
```

```
## [1] 3596
```

```
par(mfrow = c(5,1))
```

#Basically: a lot You can visualize them here using a manhattan plot, but no need really:

A common way to visually summarize large numbers of association tests is using Manhattan plots. All we need to do is plot -log10(Q) for each of the sets of association tests.

```
#pdf("LFMM_Manhattan.pdf")
par(mfrow = c(3,2))
plot(-log10(q.pdry), pch = 19, col = "blue", cex = .7, xlab = '', main="significant SNP
 x env association significance <0.05 & < 0.01", ylim=(c(0,5))) + abline(h=2, col="red")
```

```
## integer(0)
```

```
plot(-log10(q.pseas), pch = 19, col = "blue", cex = .7, xlab = '', ylim=(c(0,5))) + abli
ne(h=2, col="red")
```

```
## integer(0)
```

```
plot(-log10(q.tmin), pch = 19, col = "blue", cex = .7, xlab = '', ylim=(c(0,5))) + ablin
e(h=2, col="red")
```

```
## integer(0)
```

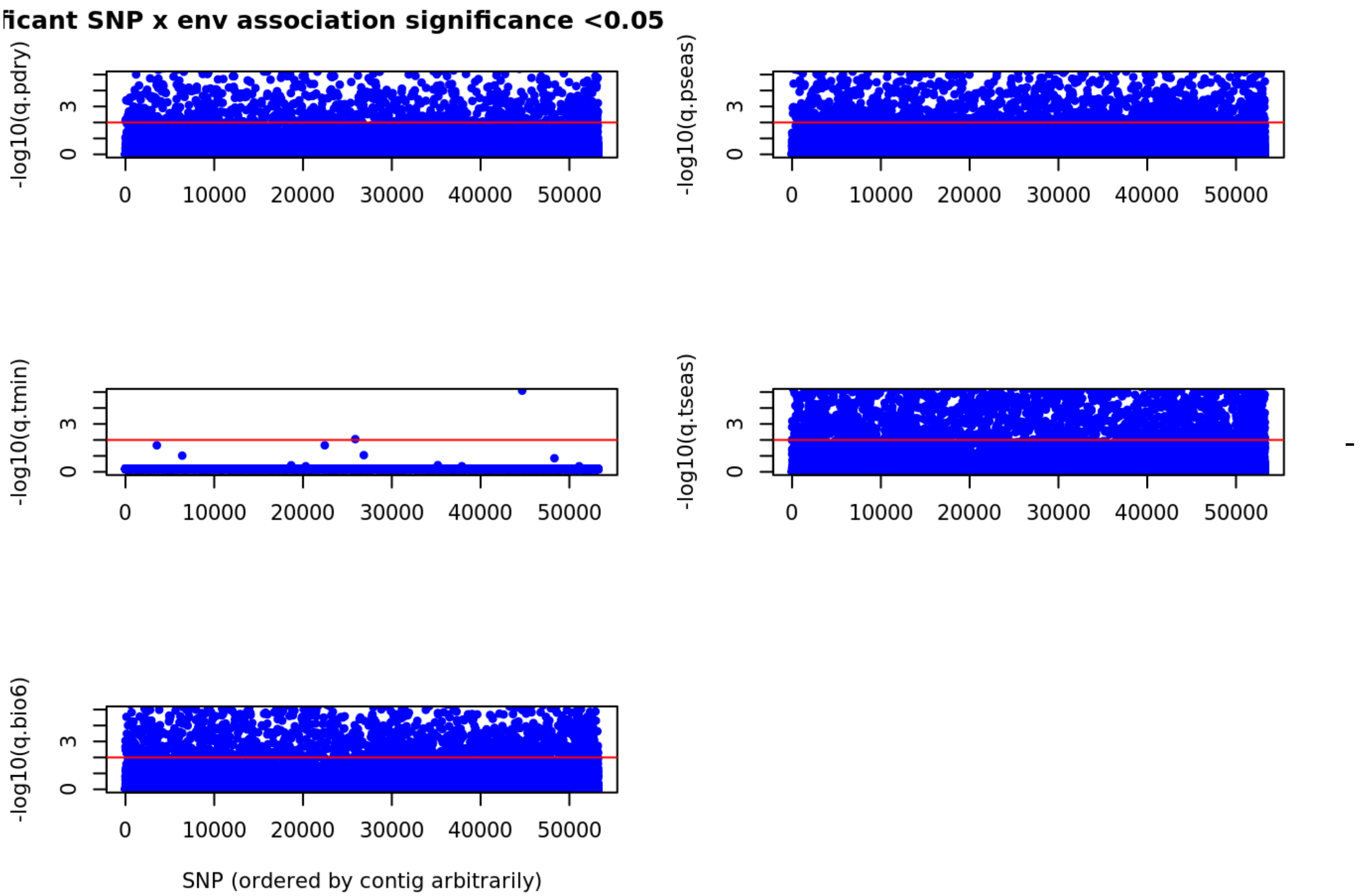
```
plot(-log10(q.tseas), pch = 19, col = "blue", cex = .7, xlab = '', ylim=(c(0,5))) + abli
ne(h=2, col="red")
```

```
## integer(0)
```

```
plot(-log10(q.bio6), pch = 19, col = "blue", cex = .7, xlab = "SNP (ordered by contig ar
bitrarily)", ylim=(c(0,5))) + abline(h=2, col="red")
```

```
## integer(0)
```

```
#dev.off()
```



log10(0.01) = 2, so you can see that there are many extremely high values greater than 2, representing significant associations. How many of the significant associations (Q < 0.01) are *also of large effect*? To answer this question, you can look at the set of significant SNPs that also have very high or very low z-scores.

We can pull loci that meet both criteria like this:

```
sum(q.pdry<0.05 & abs(z.pdry)>1.3)
```

```
## [1] 1934
```

```
#We can also look for SNPs that have significant relationships with multiple climate variables

sum(q.pdry<0.05 & abs(z.pdry)>2 & q.pseas<0.05 & abs(z.pseas)>2) #5711!
```

```
## [1] 692
```

```
#Explore the results further in this way. Finally, we might want to combine all the z and Q-values into a single table and then save for future use.

lfmm.results <- cbind(z.pdry, q.pdry, z.pseas, q.pseas, z.tmin, q.tmin, z.tseas, q.tseas, z.bio6, q.bio6)
head(lfmm.results)  #Note that the SNP locus numbers and positions are absent.
```

```
##          z.pdry    q.pdry   z.pseas   q.pseas    z.tmin    q.tmin   z.tseas
## [1,]  0.191480 0.9989960 -0.270755 0.9998151  3.584120 0.6469303 -0.404854
## [2,] -0.555883 0.9971023  0.411794 0.9998151  3.156310 0.6469303 -0.292373
## [3,]  1.595670 0.4167720 -0.490297 0.9968564  0.796914 0.6565998 -3.957970
## [4,]  2.292520 0.0888455 -0.546902 0.9789811 -0.329124 0.6777454 -1.978580
## [5,] -0.314083 0.9989960  1.541840 0.2810306 -0.133194 0.6882365  0.797439
## [6,] -0.420321 0.9989960 -0.121131 0.9998151  1.078650 0.6490034 -0.313624
##           q.tseas     z.bio6       q.bio6
## [1,] 9.966210e-01  0.0175934 0.9996095662
## [2,] 9.999446e-01  0.1419850 0.9996095662
## [3,] 5.128475e-10  3.0966300 0.0022013882
## [4,] 9.698538e-03  3.2838900 0.0008964075
## [5,] 7.450052e-01 -0.9581010 0.8502637778
## [6,] 9.999446e-01 -0.4983850 0.9996095662
```

Now that we've identified a bunch of loci that meet our criteria for significant association and large effect, we can save those positions as a matrix for annotation with SNPEff and SNPSift

#save the positions and write to matrix for annotation

```
locusposition<- lfmm.results[,1:2]
head(locusposition)
```

```
##          z.pdry    q.pdry
## [1,]  0.191480 0.9989960
## [2,] -0.555883 0.9971023
## [3,]  1.595670 0.4167720
## [4,]  2.292520 0.0888455
## [5,] -0.314083 0.9989960
## [6,] -0.420321 0.9989960
```

```
#write.matrix(locusposition, "/project/uma_lisa_komoroske/Tanya/scripts/LFMM/outlier_LD_
pruned_snps/LD_pruned_snps_lfmm_results")
```

#And now we do the whole thing over again with the same SNPset and projected bioclimatic variables from 2061-2080 We'll then compare significant associations that differ between present and future.

#Future Climate Data SSP2 represents a "middle of the road" scenario historical patterns of development are continued throughout the 21st century. We'll prepare bioclimatic variables from SSP2 as above, and run LFMM in the same way. https://www.worldclim.org/data/bioclim.html (https://www.worldclim.org/data/bioclim.html)

#See the HackMD for candidate loci annotation and results

###################Scratch Pad – ignore but please don't delete this

#Extract portions of the lfmm results table e.g.: