statistical methods for population genetics (SSMPG 2017)

Detecting and measuring selection from genome-wide SNP data

Renaud Vitalis

Centre de Biologie pour la Gestion des Populations
INRA, Montpellier, France

Collaborators: Mark A Beaumont, Kevin J Dawson and Mathieu Gautier

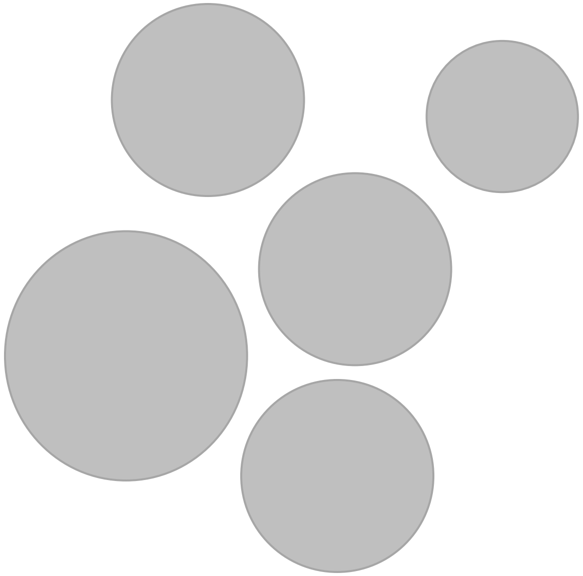
An old problem with new data

- Cavalli-Sforza (1966): “We have dedicated some effort to determining the variance that would be expected [...] as a consequence of drift, in order to compare it with the observed variation”

- Lewontin and Krakauer (1973): “While natural selection will operate differently for each locus and each allele at a locus, the effect of breeding structure is uniform over all loci and all alleles”

- **SelEstim: inferring the parameters of a full model that accounts for drift, migration, and selection…**

A simple population model



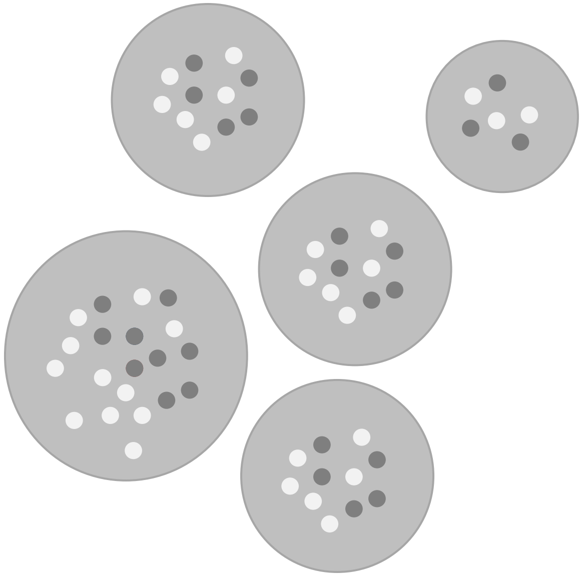
- Consider an island model of population structure, where:
- $M_i = 4N_i m_i$ is the migration parameter
- π_j is the frequency at the *j*th locus in the total population (migrant pool)

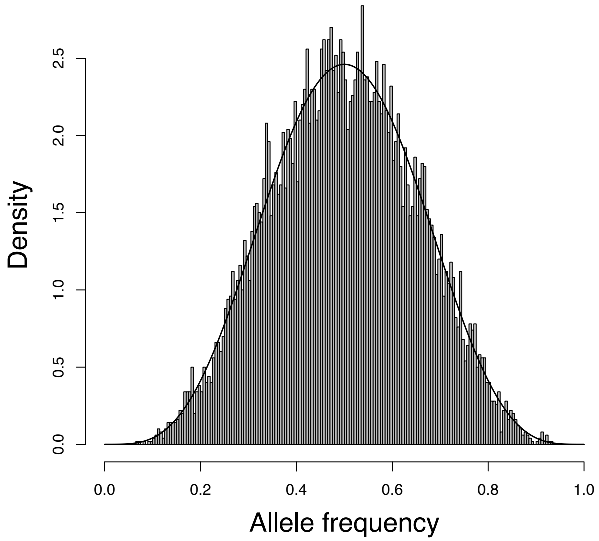
The data

A G A G T T C T G T C G A G A
A G G G T T A T G G C G A G A
C G T T G C G G G A A T C C C
C C T T A C G G A A A T C T C
T C T T T G A C G A C T C C T
T C T T A G A G G A C T C C T
A G A G C T G G T C T A G A T
A G A A C T G G T A T A G G T
C C T A G G C G T T A C A A
C C T T G G C G T G A C A C
A A G C T T G G C C G A A C G
A G G C T T A G C C G A A C G
C C A G T A C A T G A A C G A
C C G G T A C A T G T A C G A
YGG-01-0305

- Single Nucleotide Polymorphisms (SNPs) genotyped in different populations
- The data consist in allele counts for each locus in each population
- The likelihood of a sample of genes is binomial

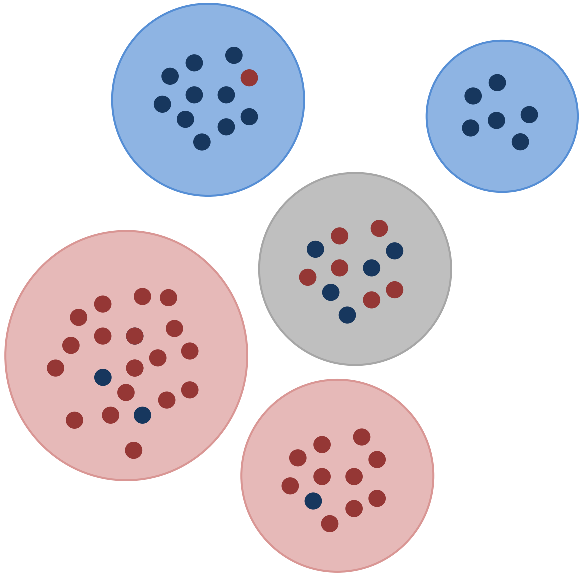
Neutral polymorphisms





- Diffusion theory gives the distribution of allele frequencies, as a function of M_i and π_j

Locally adapted genes

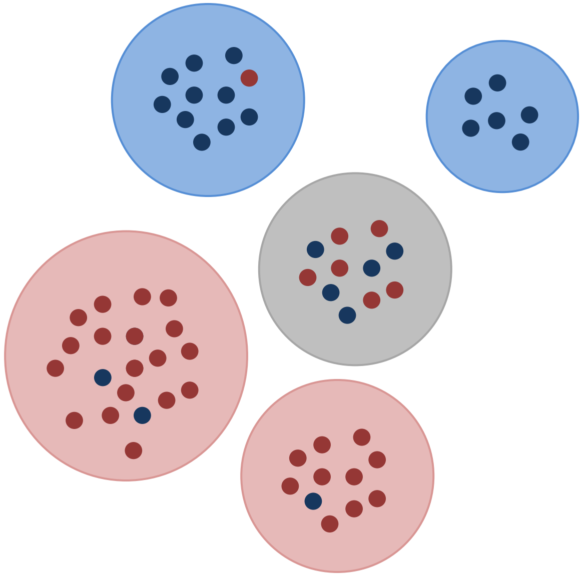


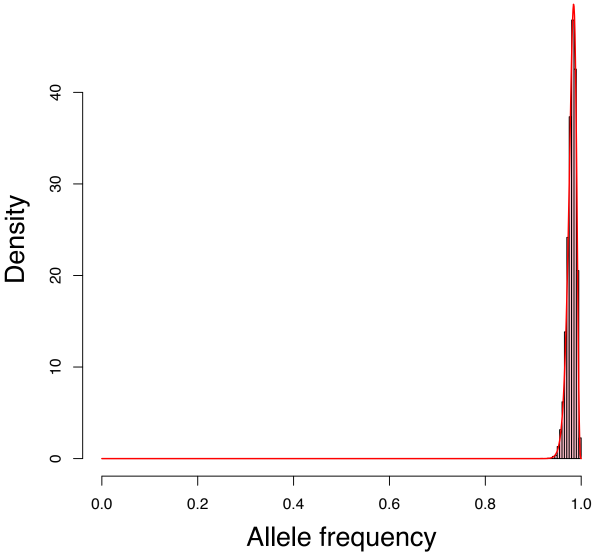
- In population *i*, at locus *j*, genotypes AA, Aa and aa have relative fitness:

AA	Aa	aa
$1 + s_{ij}$	$1 + s_{ij} / 2$	1

- $\sigma_{ij} = 2\ N_i s_{ij}$ is the selection parameter
- κ_{ij} indicates which of the 2 alleles is A

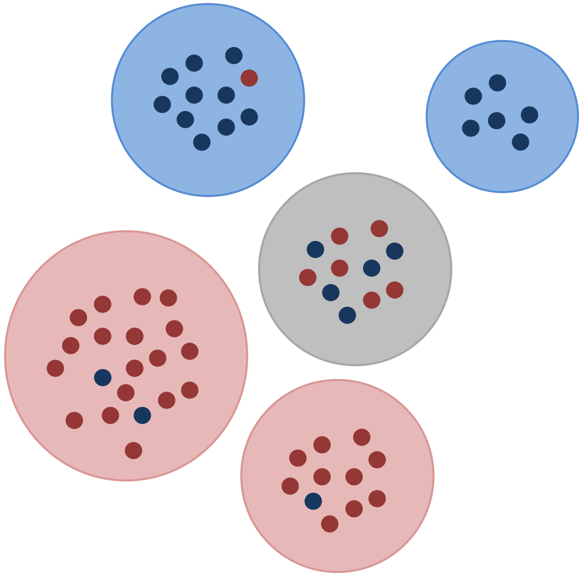
Locally adapted genes





- Diffusion theory gives the distribution of allele frequencies, as a function of M_i, π_j, σ_{ij} and κ_{ij}

A model-based approach

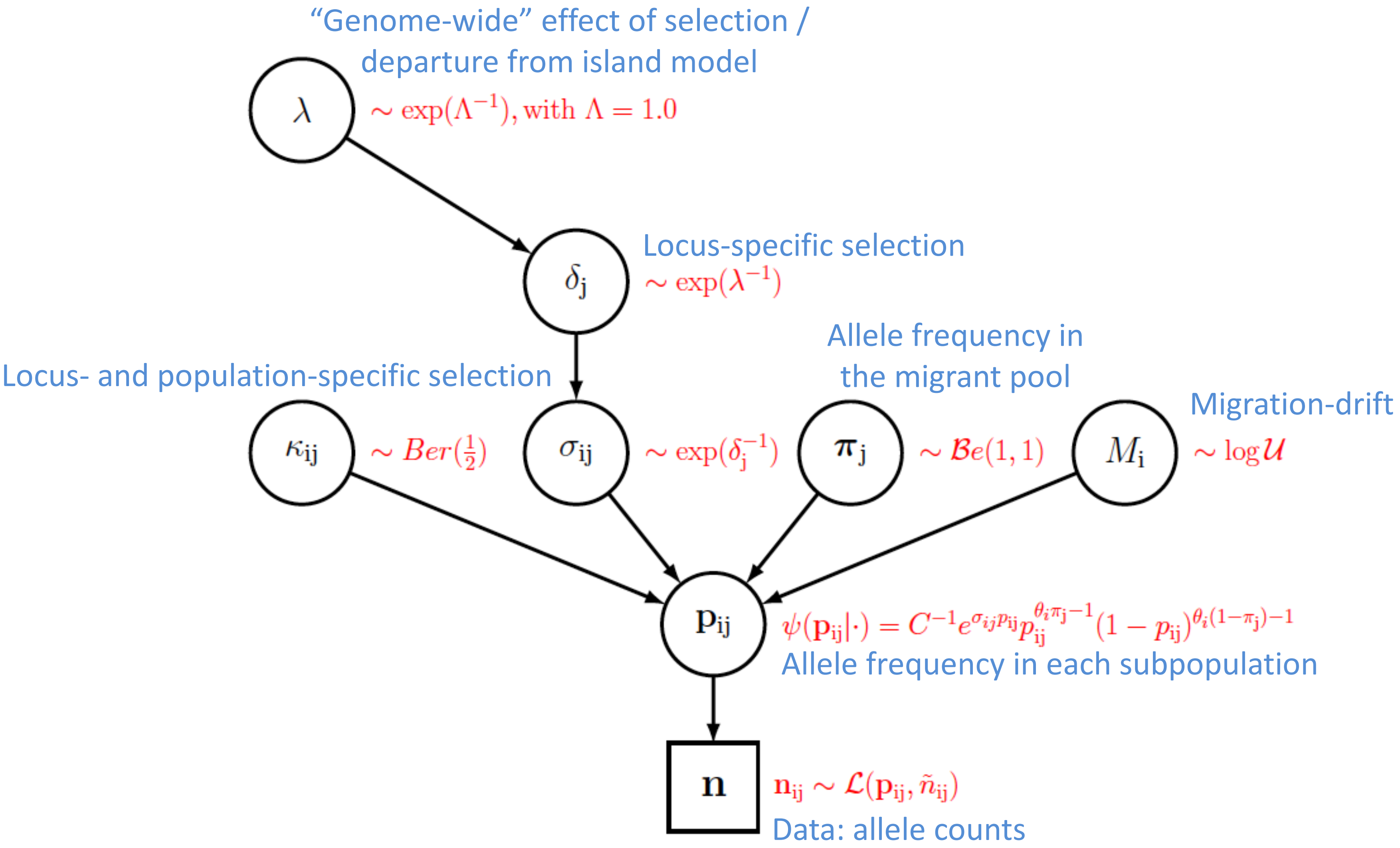


- **We assume that *all* markers are targeted by selection, to some extent**

- We infer the model parameters from the data (allele counts) using MCMC

- We provide a decision criterion to discriminate neutral markers from presumably selected loci

A hierarchical Bayesian model



“Genome-wide” effect of selection / departure from island model

$\sim \exp(\Lambda^{-1}), \text{with } \Lambda = 1.0$

Locus-specific selection

$\sim \exp(\lambda^{-1})$

Allele frequency in the migrant pool

Locus- and population-specific selection

Migration-drift

$\sim Ber(\frac{1}{2})$

$\sim \exp(\delta_{\mathrm{j}}^{-1})$

$\sim \mathcal{B}e(1,1)$

$\sim \log \mathcal{U}$

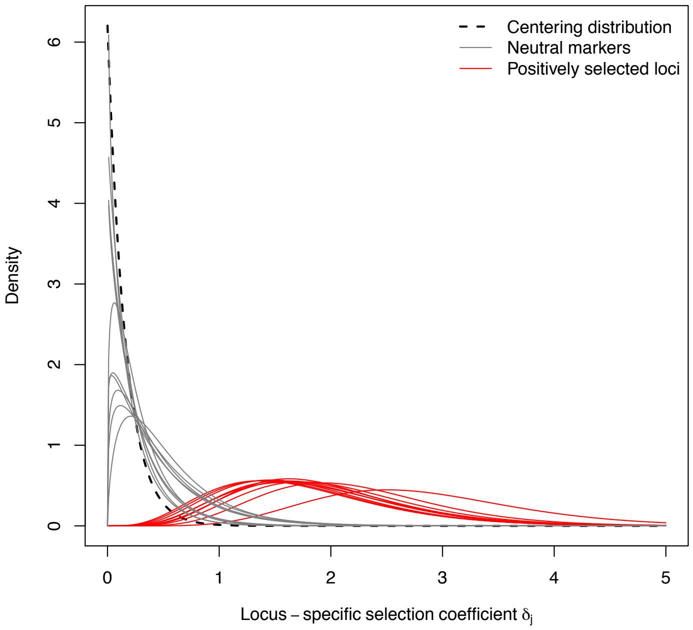
$\psi(\mathbf{p}_{\mathrm{ij}}|\cdot) = C^{-1} e^{\sigma_{ij} p_{\mathrm{ij}}} p_{\mathrm{ij}}^{\theta_i \pi_{\mathrm{j}} - 1} (1 - p_{\mathrm{ij}})^{\theta_i (1 - \pi_{\mathrm{j}}) - 1}$

Allele frequency in each subpopulation

$\mathbf{n}_{\mathrm{ij}} \sim \mathcal{L}(\mathbf{p}_{\mathrm{ij}}, \tilde{n}_{\mathrm{ij}})$

Data: allele counts

Decision criterion

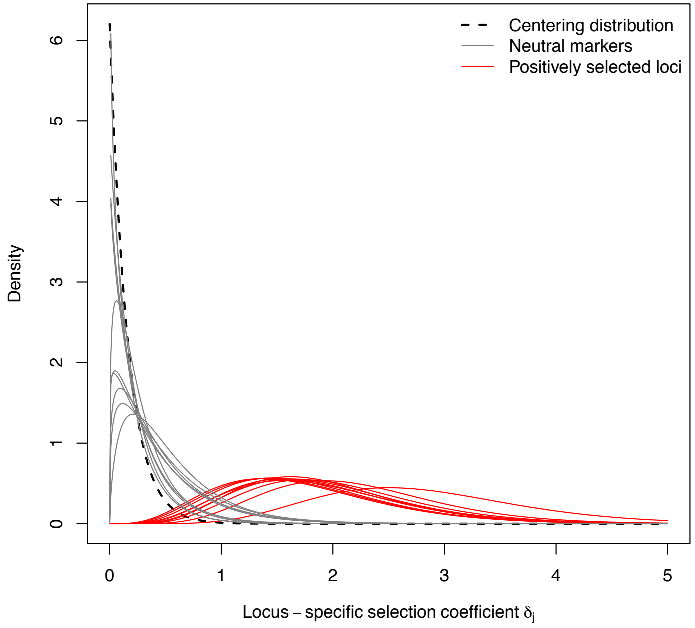


- We expect that the posterior distribution of δ_{ij} for a selected locus departs from zero

- We compare the posterior distribution of δ_{ij} to a “centering distribution” that integrates over the overall departure from neutrality

- We use the Kullback-Leibler divergence (KLD) as a distance between these distributions

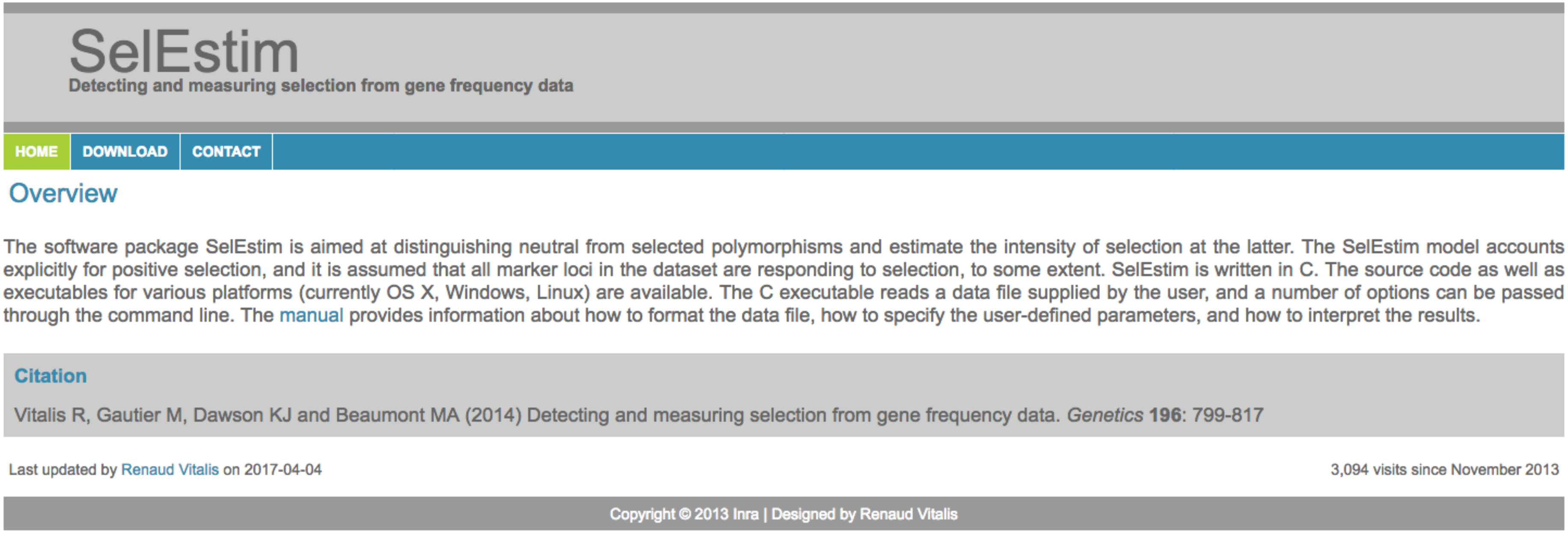
Calibration of the KLD



- We calibrate the KLD by generating pseudo observed data (pod), drawn from the posterior distribution of the model parameters

- The pod is analysed, and the quantiles of the KLD distribution so obtained are then used as threshold values

A software package



Overview

The software package SelEstim is aimed at distinguishing neutral from selected polymorphisms and estimate the intensity of selection at the latter. The SelEstim model accounts explicitly for positive selection, and it is assumed that all marker loci in the dataset are responding to selection, to some extent. SelEstim is written in C. The source code as well as executables for various platforms (currently OS X, Windows, Linux) are available. The C executable reads a data file supplied by the user, and a number of options can be passed through the command line. The manual provides information about how to format the data file, how to specify the user-defined parameters, and how to interpret the results.

Citation

Vitalis R, Gautier M, Dawson KJ and Beaumont MA (2014) Detecting and measuring selection from gene frequency data. *Genetics* **196**: 799-817

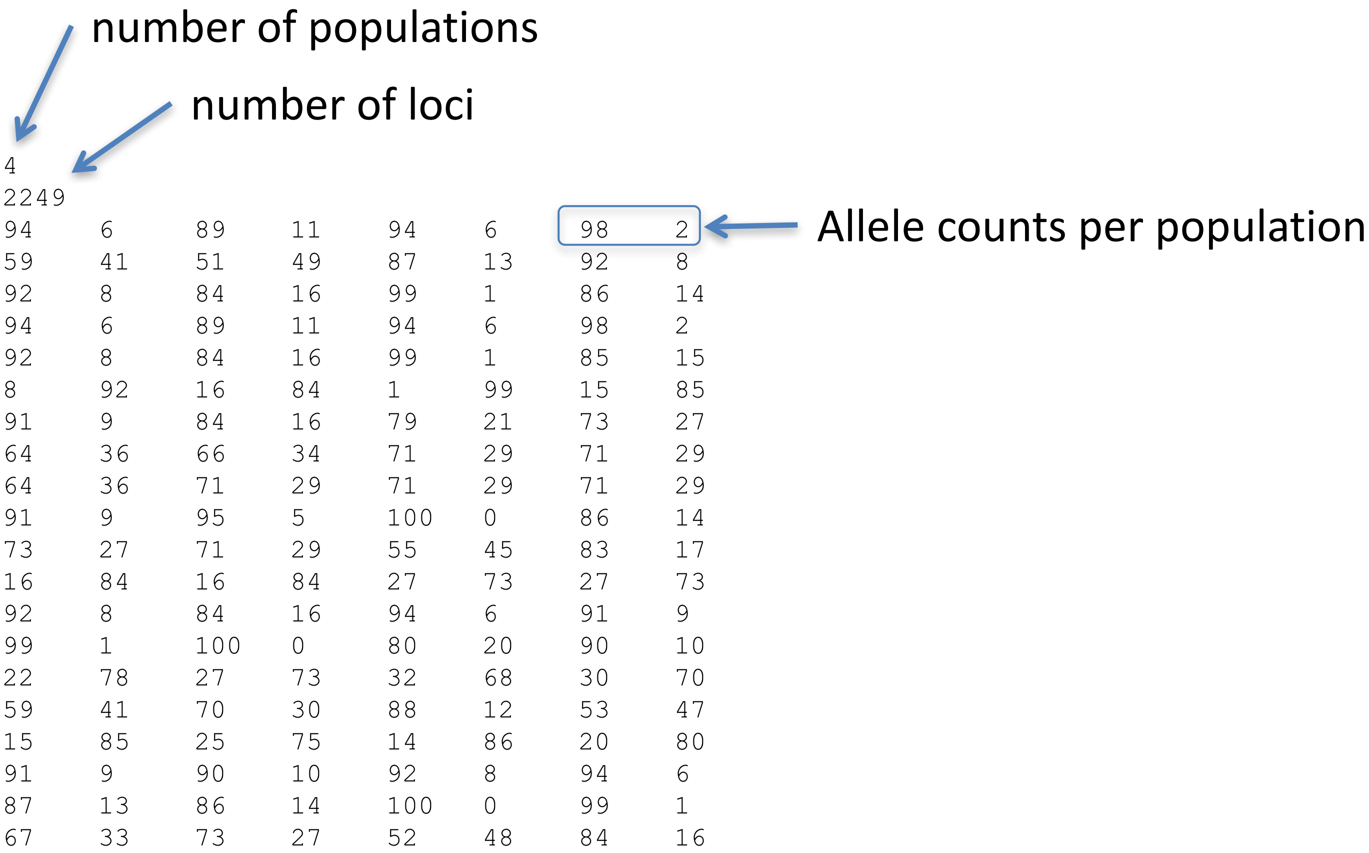
Last updated by Renaud Vitalis on 2017-04-04

3,094 visits since November 2013

Copyright © 2013 Inra | Designed by Renaud Vitalis

A command-line, parallelized (OpenMP), interface:
http://www1.montpellier.inra.fr/CBGP/software/selestim/index.html

Using SELESTIM: input file



number of populations

number of loci

Allele counts per population

```
4
2249
94    6     89    11    94    6     98    2
59    41    51    49    87    13    92    8
92    8     84    16    99    1     86    14
94    6     89    11    94    6     98    2
92    8     84    16    99    1     85    15
8     92    16    84    1     99    15    85
91    9     84    16    79    21    73    27
64    36    66    34    71    29    71    29
64    36    71    29    71    29    71    29
91    9     95    5     100   0     86    14
73    27    71    29    55    45    83    17
16    84    16    84    27    73    27    73
92    8     84    16    94    6     91    9
99    1     100   0     80    20    90    10
22    78    27    73    32    68    30    70
59    41    70    30    88    12    53    47
15    85    25    75    14    86    20    80
91    9     90    10    92    8     94    6
87    13    86    14    100   0     99    1
67    33    73    27    52    48    84    16
```

Using SelEstim

Using the command line:

```
./selestim -help
usage: ./src/selestim [ options ]
valid options are :
-help               print this message
-version            print version
-file               name of the input file (default: data.dat)
-outputs            directory where the outputs will be produced (default: current directory)
-seed               initial seed for the random number generator (default: computed from current time)
-threads            number of threads to be used (default: number of cpu available)
-length             run length of the Markov chain (default: 100000)
-thin               thinning interval size (default: 40)
-burnin             length of the burn-in period (default: 50000)
-npilot             number of pilot runs (default: 25)
-lpilot             length of each pilot run (default: 500)
-pool               option to analyse data from pooled DNA samples (default: unset)
-fixed_beta         option to fix the shape parameters of the beta prior distribution of pi (default: unset)
-beta_a             shape parameter of the beta prior distribution of pi (default: 0.70)
-beta_b             shape parameter of the beta prior distribution of pi (default: 0.70)
-fixed_lambda       option to fix the value of lambda (default: unset)
-lambda_prior       prior distribution of lambda, which can only be inverse gamma ('invgam', by default) or an exponential ('exp')
-invgam_shape       shape parameter of the inverse gamma prior distribution of lambda (default: 3.00)
-invgam_rate        rate parameter of the inverse gamma prior distribution of lambda (default: 2.00)
-captl_lambda       rate parameter of the exponential prior distribution of lambda (default: 1.00)
-min_M              lower bound for the log-uniform prior on M (default: 0.001)
-max_M              upper bound for the log-uniform prior on M (default: 10000)
-max_sig            upper bound for the exponential prior on sigma (default: 700)
-dlt_cnt            half window width from which updates of allele counts are randomly drawn (default: 5)
-dlt_p              half window width from which updates of p are randomly drawn (default: 0.25)
-dlt_M              standard deviation of the lognormal distribution from which updates of M are drawn (default: 0.10)
-dlt_pi             half window width from which updates of pi are randomly drawn (default: 0.25)
-dlt_sig            standard deviation of the lognormal distribution from which updates of sigma are drawn (default: 2.50)
-dlt_del            standard deviation of the lognormal distribution from which updates of delta are drawn (default: 0.80)
-dlt_lam            standard deviation of the lognormal distribution from which updates of lambda are drawn (default: 0.05)
-dlt_beta_mu        half window width from which updates of the beta mu parameters are drawn (default: 0.03)
-dlt_beta_nu        standard deviation of the lognormal distribution from which updates of the beta nu parameters are drawn (default: 1.00)
-calibration        option to generate pseudo-observed data and calibrate the Kullback-Leibler divergence
-calibration_only   option to generate pseudo-observed data and calibrate the Kullback-Leibler divergence from previous analyses
-pod_nbr_loci       option to specify the number of loci to be simulated for calibration (if different from the dataset)
-verbose            option to print the traces of all parameters (generates big output files!)
```

Pilot runs are used to adjust the parameters of the proposal functions, in order to get acceptance rates between 0.25 and 0.40. The burnin corresponds to the preliminary part of the chain before it reaches stationarity.

```
./src/selestim -file data/data.dat -burnin 5000 -npilot 15 -lpilot 500 -length 25000 -thin 25 -outputs run-example/
------------------------------------------------------------------------------------
Wed Sep  6 15:18:21 2017
------------------------------------------------------------------------------------


./src/selestim -file data/data.dat -burnin 5000 -npilot 15 -lpilot 500 -length 25000 -thin 25 -outputs run-example/


This analysis was performed using selestim (version 1.1.7)


Checking file `data/data.dat'... OK
The data consist in 2249 SNPs and 4 sampled populations


------------------------------------------------------------------------------------
Mean sample size (min, max) per sampled population:
------------------------------------------------------------------------------------
Population no.  1: 100.00 (100,100)
Population no.  2: 100.00 (100,100)
Population no.  3: 100.00 (100,100)
Population no.  4: 100.00 (100,100)
------------------------------------------------------------------------------------
Overall          : 100.00 (100,100)
------------------------------------------------------------------------------------


------------------------------------------------------------------------------------
Overall genetic differentiation (F_ST)                                = 0.0621
------------------------------------------------------------------------------------


------------------------------------------------------------------------------------
Prior distribution of lambda is inverse gamma (lambda_prior)
    with shape parameter (invgam_shape)                               = 3.000000
    and rate parameter (invgam_rate)                                  = 2.000000


Number of threads used (threads)                                      = 8
Random number generator's seed (seed)                                 = 1504703901


Length of the burn-in period (burnin)                                 = 5000
Run length of the Markov chain (length)                               = 25000
Thinning interval (thin)                                              = 25
Number of MCMC samples (length / thin)                                = 1000
Number of pilot studies (npilot)                                      = 15
Length of each pilot study (lpilot)                                   = 500


Lower bound of the interval for M (min_M)                             = 0.001000
Upper bound of the interval for M (max_M)                             = 10000.00
Upper bound of the interval for sigma (max_sig)                       = 700.00
Initial half window width for updates of allele counts (dlt_cnt)      = 5
Initial half window width for updates of p (dlt_p)                    = 0.250000
Initial SD of the lognormal for updates of M (dlt_M)                  = 0.100000
Initial half window width for updates of pi (dlt_pi)                  = 0.250000
Initial SD of the lognormal for updates of sigma (dlt_sig)            = 2.500000
Initial SD of the lognormal for updates of delta (dlt_del)            = 0.800000
Initial half window width for updates of mu (dlt_beta_mu)             = 0.025000
Initial SD of the lognormal for updates of nu (dlt_beta_nu)           = 1.000000
------------------------------------------------------------------------------------
```

```
--------------------------------------------------------------------------------
Pilot run # 1:
--------------------------------------------------------------------------------


Allele frequencies p_ij's
average value = 0.6997 [0.0010,1.0000]
average updating parameter = 0.2328 [0.2000,0.2500]
average acceptance rate = 0.2573 [0.0100,0.3880]
3087 parameters have been scaled, out of 8996


Population parameters M_i's
average value = 16.9116 [11.7429,23.1240]
average updating parameter = 0.1000 [0.1000,0.1000]
average acceptance rate = 0.3185 [0.3160,0.3240]
0 parameters have been scaled, out of 4


Shape parameter (a) of the prior distribution of migrant allele frequencies pi_j's
current value = 2.2349
updating parameter = 0.0250
average acceptance rate = 0.2840
0 parameters have been scaled, out of 1


Shape parameter (b) of the prior distribution of migrant allele frequencies pi_j's
current value = 1.0079
updating parameter = 0.8000
average acceptance rate = 0.0200
1 parameters have been scaled, out of 1


Migrant allele frequencies pi_j's
average value = 0.7001 [0.0528,0.9910]
average updating parameter = 0.2485 [0.2000,0.3125]
average acceptance rate = 0.3192 [0.1220,0.4260]
112 parameters have been scaled, out of 2249


Genome-wide coefficient of selection lambda
current value = 1.3414


Locus-specific selection coefficient delta_j's
average value = 1.3405 [0.0002,9.7590]
average updating parameter = 0.9999 [0.8000,1.0000]
average acceptance rate = 0.5490 [0.4000,0.6380]
2248 parameters have been scaled, out of 2249


Locus- population-specific selection coefficient sigma_ij's
average value = 1.3305 [0.0000,31.0552]
average updating parameter = 3.1180 [2.0000,3.1250]
average acceptance rate = 0.4592 [0.1540,0.5500]
8904 parameters have been scaled, out of 8996


--------------------------------------------------------------------------------
Pilot run # 2:
--------------------------------------------------------------------------------

[…]
```

```
[…]

------------------------------------------------------------------------------------
Wed Sep  6 15:21:10 2017
------------------------
Computing time elapsed since beginning = 39 secs.
Estimated time until the MCMC stops    = 2 mins. 32 secs.
------------------------------------------------------------------------------------


------------------------------------------------------------------------------------
Running the MCMC
------------------------------------------------------------------------------------


 starting [..........]
 10% done [..........]
 20% done [..........]
 30% done [..........]
 40% done [..........]
 50% done [..........]
 60% done [..........]
 70% done [..........]
 80% done [..........]
 90% done [..........]
100% done !


------------------------------------------------------------------------------------
Computation of the effective sample size (ESS)


log posterior density                           = 12.950260
parameters M                                    = (48.879781,109.951214,55.613403,173.586716)
shape parameter (alpha) of the parameter pi     = 1000.000000
shape parameter (beta) of the parameter pi      = 625.047719
(hyper-)parameter lambda                        = 13.964527


ESS is a measure of how well a Markov chain is mixing. ESS represents the number
of effectively independent draws from the posterior distribution that the Markov
chain is equivalent to [ESS must be compared to the chain length = 1000].


Warning! Low ESS (due to strong autocorrelation) indicates poor mixing of the
Markov chain. The ESS of the (hyper-)parameter lambda is typically lower than that
of the other parameters. You are strongly recommended to inspect the trace of the
lambda parameter in the 'trace_lambda.out' file. The trace shall show relatively
good mixing (low autocorrelation, AND no decreasing trend). Otherwise, you may want
to increase the length of the burn-in period and/or the total length of the Markov
chain.
------------------------------------------------------------------------------------


------------------------------------------------------------------------------------
Wed Sep  6 15:23:42 2017
------------------------
Total computing time elapsed           = 3 mins. 11 secs.
------------------------------------------------------------------------------------


The program has successfully terminated.
```

Example of outputs

`summary_delta.out`

```
   locus        mean         std         KLD
       1    1.191570    1.180116    0.000469
       2    1.419017    1.453338    0.012270
       3    1.182796    1.104979    0.006164
       4    1.234939    1.167513    0.003878
       5    1.212765    1.198372    0.000217
       6    1.193120    1.168858    0.000844
       7    1.155191    1.111058    0.003505
       8    1.020180    0.995173    0.016372
       9    1.078560    1.050439    0.008527
      10    1.152572    1.148356    0.001782
      11    1.165585    1.190205    0.001759
      12    1.148280    1.240980    0.010388
      13    1.143361    1.136574    0.002306
      14    1.390084    1.320479    0.011686
      15    1.139665    1.102747    0.003822
      16    1.242628    1.244602    0.000120
      17    1.083716    1.088243    0.007131
      18    1.171430    1.094571    0.006502
      19    1.217950    1.193716    0.000512
      20    1.269960    1.202416    0.004332
      21    1.262542    1.218703    0.002030
      22    1.244927    1.246819    0.000150
      23    1.424222    1.420541    0.012102
      24    1.249770    1.302800    0.002559
      25    1.301608    1.292840    0.001990

[…]
```

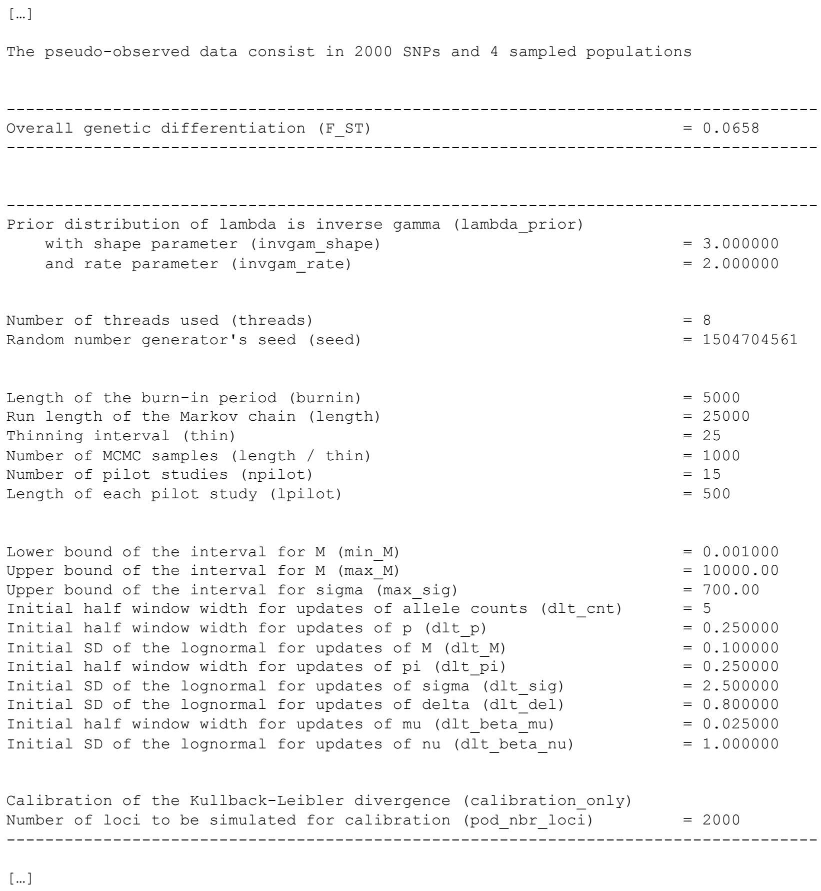
- Use R scripts to analyse the outputs (e.g., the CODA package to test for convergence) and plot graphs (some ad-hoc functions in `R/SelEstim.R`)

KLD calibration

```
./src/selestim -file data/data.dat -burnin 5000 -npilot 15 -lpilot 500 -length 25000 -thin 25
-outputs run-example/ -calibration_only -pod_nbr_loci 2000
-------------------------------------------------------------------------------------
Wed Sep  6 15:29:21 2017
-------------------------------------------------------------------------------------


./src/selestim -file data/data.dat -burnin 5000 -npilot 15 -lpilot 500 -length 25000 -thin 25
-outputs run-example/ -calibration_only -pod_nbr_loci 2000


This analysis was performed using selestim (version 1.1.7)


-------------------------------------------------------------------------------------
Calibration of the Kullback-Leibler divergence using pseudo-observed data
-------------------------------------------------------------------------------------


Generating file `run-example-2/calibration/pod_data.dat'...


 starting [..........]
 10% done [..........]
 20% done [..........]
 30% done [..........]
 40% done [..........]
 50% done [..........]
 60% done [..........]
 70% done [..........]
 80% done [..........]
 90% done [..........]
100% done !

[…]
```

- The first step of the calibration requires generating pseudo-observed data (pod)

KLD calibration



```
[…]

The pseudo-observed data consist in 2000 SNPs and 4 sampled populations


-------------------------------------------------------------------------------------
Overall genetic differentiation (F_ST)                                = 0.0658
-------------------------------------------------------------------------------------


-------------------------------------------------------------------------------------
Prior distribution of lambda is inverse gamma (lambda_prior)
    with shape parameter (invgam_shape)                               = 3.000000
    and rate parameter (invgam_rate)                                  = 2.000000


Number of threads used (threads)                                      = 8
Random number generator's seed (seed)                                 = 1504704561


Length of the burn-in period (burnin)                                 = 5000
Run length of the Markov chain (length)                               = 25000
Thinning interval (thin)                                              = 25
Number of MCMC samples (length / thin)                                = 1000
Number of pilot studies (npilot)                                      = 15
Length of each pilot study (lpilot)                                   = 500


Lower bound of the interval for M (min_M)                             = 0.001000
Upper bound of the interval for M (max_M)                             = 10000.00
Upper bound of the interval for sigma (max_sig)                       = 700.00
Initial half window width for updates of allele counts (dlt_cnt)      = 5
Initial half window width for updates of p (dlt_p)                    = 0.250000
Initial SD of the lognormal for updates of M (dlt_M)                  = 0.100000
Initial half window width for updates of pi (dlt_pi)                  = 0.250000
Initial SD of the lognormal for updates of sigma (dlt_sig)            = 2.500000
Initial SD of the lognormal for updates of delta (dlt_del)            = 0.800000
Initial half window width for updates of mu (dlt_beta_mu)             = 0.025000
Initial SD of the lognormal for updates of nu (dlt_beta_nu)           = 1.000000


Calibration of the Kullback-Leibler divergence (calibration_only)
Number of loci to be simulated for calibration (pod_nbr_loci)         = 2000
-------------------------------------------------------------------------------------

[…]
```

- The second step of the calibration involves the full analysis of that pod

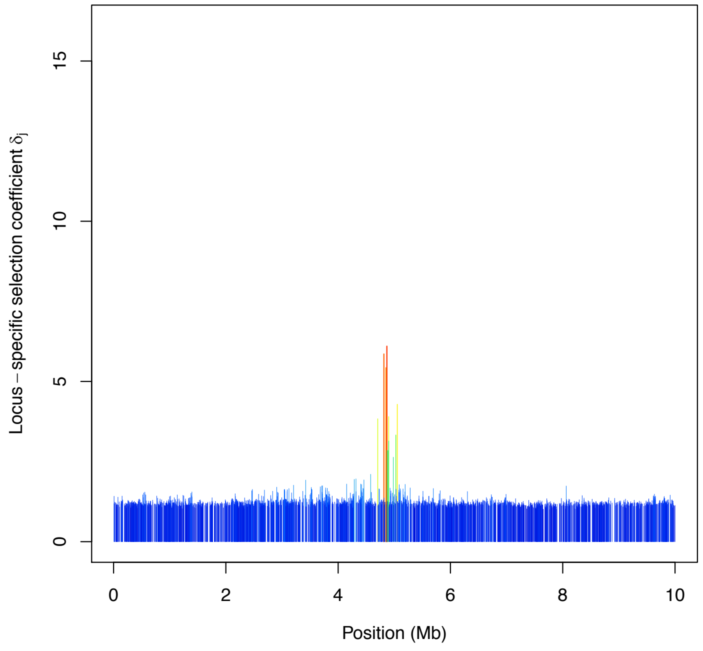
KLD calibration

```
KLD_quantiles.out
```

```
quantile          KLD
  50.00%     0.004500
  90.00%     0.017195
  95.00%     0.023965
  98.00%     0.034586
  99.00%     0.040743
  99.50%     0.056360
  99.90%     0.127105
  99.95%     0.133458
  99.99%     0.234392
```

- The quantiles provide threshold values that can be used as a decision criterion to discriminate between neutral markers and selected loci

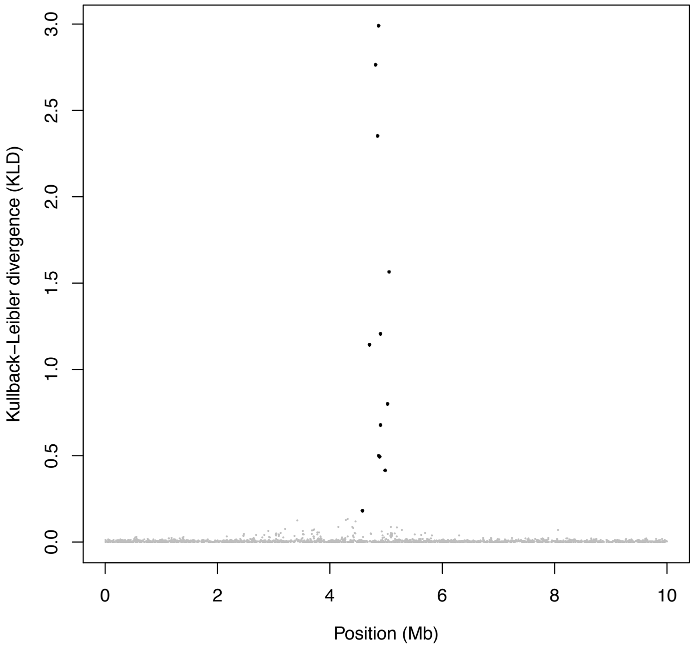
A worked example



```
> source('R/SelEstim.R')

> plot.delta(file = "run-
  example/summary_delta.out",map =
  "data/data.map"
```

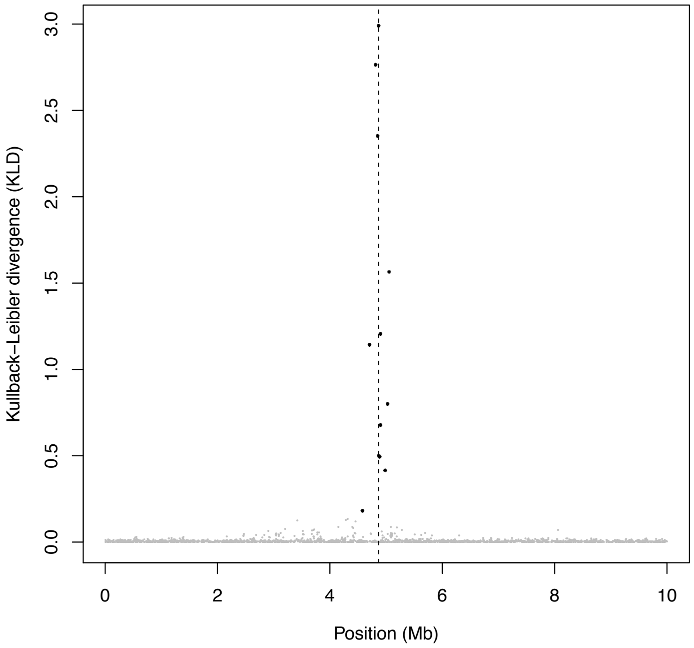
- The data consist in a simulation performed with simuPOP, with 4 populations made of 1,000 diploids diverging for 100 generations. A single mutation (at position 4,867,859 bp) is selected for in population 1

Plotting outputs with R



```
> plot.kld(file = "run-example/summary_delta.out",map = "data/data.map",calibration_file = "run-example/calibration/summary_delta.out",limit = 0.001)
```

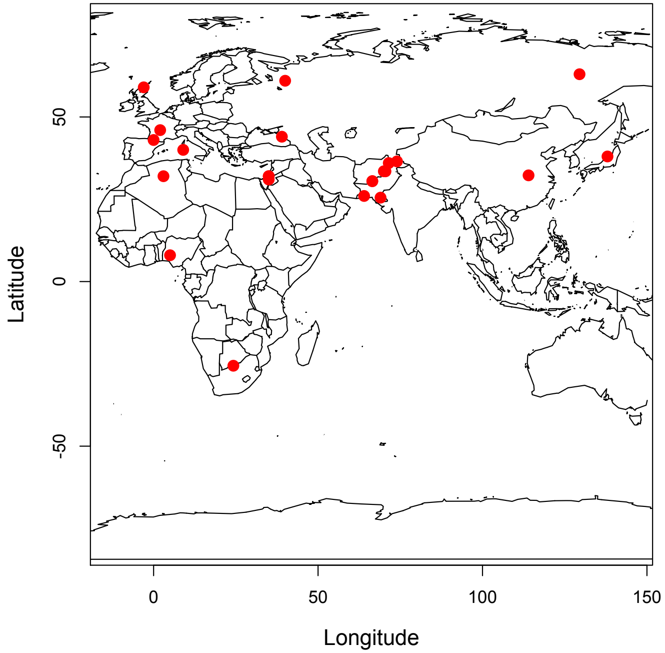
Plotting outputs with R



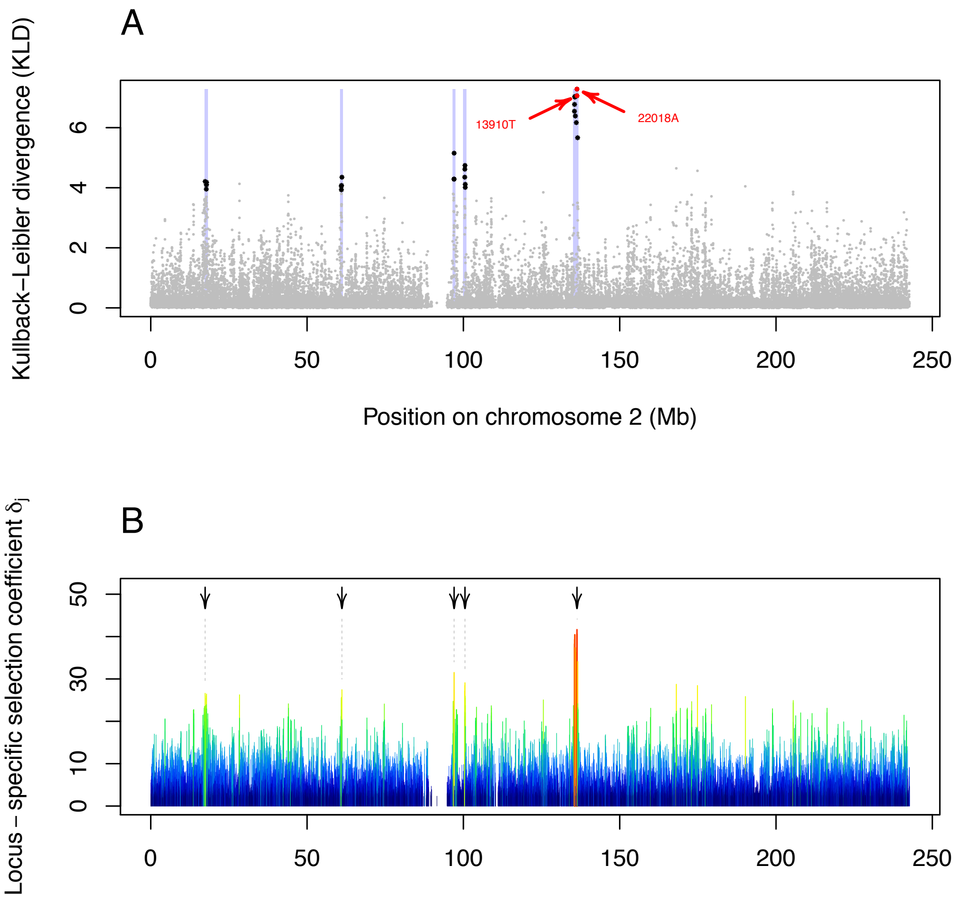
```
> rslt <- read.table("run-
   example/summary_delta.out",header =
   TRUE)
> top.snp <- which(rslt$KLD ==
   max(rslt$KLD))
> top.snp
[1] 1124

> abline(v = 4.867859,lty = 2)
```

Application on human data (CEPH)

We have applied the method on a subset of the Stanford HGDP-CEPH SNP genotyping data from chromosome 2 (52,631 SNPs)



Application on human data (CEPH)

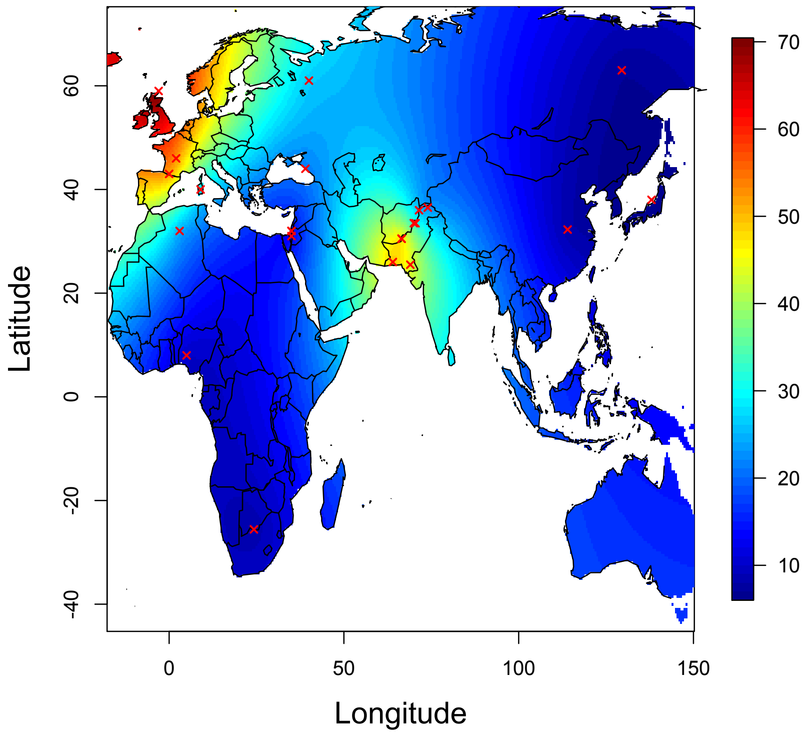


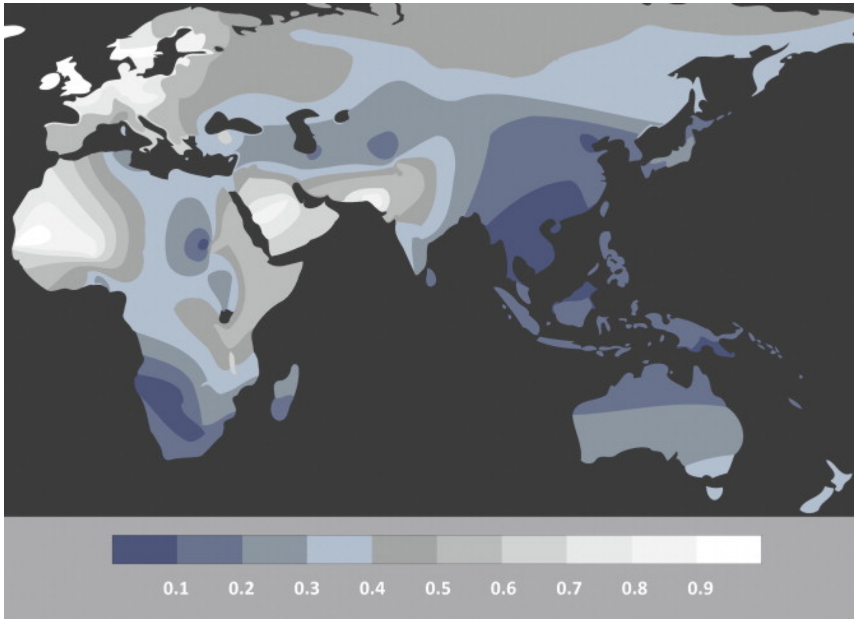
1Mb windows that contain at least 3 SNPs with KLD > 3.912

Strong signature of selection in the vicinity of the lactase gene *LCT*, in particular at 2 SNPs reported to be very tightly associated with lactase persistence (13919T and 22018A; see Bersaglieri *et al*. 2004).

Application on human data (CEPH)

Coefficient de sélection σ_{ij} at 13910





Distribution of lactase persistence phenotype (Itan *et al*. 2010)

The selection coefficient at 13910T (left) is stronger in milk-drinking populations. It correlates with lactase persistence in Europe and the Indus valley, not in Africa or the Near and Middle East: convergent evolution.

Take home messages

- Bayesian methods: check for convergence and mixing properties! (see the R package CODA)

- This family of approaches does not take linkage disequilibrium (LD) into account (yet)

- Be aware of the underlying population models and assumptions (equilibrium island model, etc.)