Supplementary material: Bayesian inference of natural selection from spatiotemporal phenotypic data

Olivier David   Gaëlle van Frank   Isabelle Goldringer   Pierre Rivière

Michel Turbet Delof

A   Genetic evolution

We show that the genetic composition of population (i, j) evolves according to the same equations as those of the Kalman filter. Equations (3), (4) and (5) can then be derived using the recursions known for the Kalman filter. In this appendix, $\mathcal{N}(x, \mu, \sigma^2)$ denotes the density of the normal distribution $\mathcal{N}(\mu, \sigma^2)$ taken at x.

A.1   Hidden Markov model

First we define a hidden Markov model (Fig. A.1). The state process $(\mathbb{G}_{ijt})_{t \in \mathbb{N}}$ of this model is defined by

$$\mathbb{G}_{ij0} = G_{ij0},$$
$$\forall t \in \mathbb{N}, \ \mathbb{G}_{ij(t+1)} \,|\, \mathbb{G}_{ijt} \sim \mathcal{N}(\mathbb{G}_{ijt}, \tau_M^{-2}).$$

The observation process $(\mathbb{X}_{ijt})_{t \in \mathbb{N}}$ of the model is defined by

$$\forall t \in \mathbb{N}, \ \mathbb{Z}_{ijt} \,|\, \mathbb{G}_{ijt} \sim \mathcal{N}(\mathbb{G}_{ijt} + A_{it} + B_{ijt}, \tau_{Cijt}^{-2}),$$
$$\mathbb{X}_{ijt} \,|\, \mathbb{Z}_{ijt} \sim \mathcal{N}(\mathbb{Z}_{ijt}, \tau_{Fi}^{-2}),$$

so that

$$\mathbb{X}_{ijt} \,|\, \mathbb{G}_{ijt} \sim \mathcal{N}(\mathbb{G}_{ijt} + A_{it} + B_{ijt}, \tau_{Sijt}^{-2}).$$

The variables $\mathbb{G}_{ij0}$, $(\mathbb{G}_{ij(t+1)} - \mathbb{G}_{ijt})_{t \in \mathbb{N}}$ and $(\mathbb{X}_{ijt} - \mathbb{G}_{ijt} - A_{it} - B_{ijt})_{t \in \mathbb{N}}$ are assumed to be independent.

A.2   Kalman filter

The Kalman filter aims at inferring the latent variables $(\mathbb{G}_{ijt})_{t \in \mathbb{N}}$ given observations for the variables $(\mathbb{X}_{ijt})_{t \in \mathbb{N}}$ (Durbin and Koopman, 2001, Chap. 4; Del Moral and Vergé, 2014, Chap. 12). It is
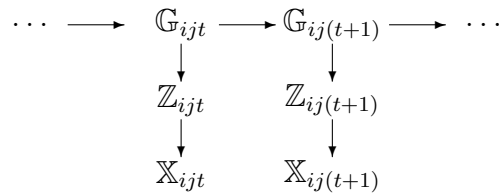

Figure A.1: Graph of a hidden Markov model with latent variables $(\mathbb{G}_{ijt})_{t \in \mathbb{N}}$ and observed variables $(\mathbb{X}_{ijt})_{t \in \mathbb{N}}$.

based on the densities $\forall t \in \mathbb{N}$, $p(\mathbb{G}_{ijt} \mid \mathbb{X}_{ij0:t})$ and $p(\mathbb{G}_{ij(t+1)} \mid \mathbb{X}_{ij0:t})$, where $\mathbb{X}_{ij0:t} = (\mathbb{X}_{ij0}, \ldots, \mathbb{X}_{ijt})$, called respectively the optimal filters and the one-step predictors.

We show by induction that the selection step amounts to an optimal filter:

$$\forall t \in \mathbb{N}, \; p(G'_{ijt} = g_{ijt}) = p(\mathbb{G}_{ijt} = g_{ijt} \mid \mathbb{X}_{ij0:t} = \mu_{Fi0:t})$$

and that the mutation step amounts to a one-step predictor:

$$\forall t \in \mathbb{N}, \; p(G_{ijt} = g_{ijt}) = p(\mathbb{G}_{ijt} = g_{ijt} \mid \mathbb{X}_{ij0:(t-1)} = \mu_{Fi0:(t-1)}), \tag{A.1}$$

with the convention that for $t = 0$, $p(\mathbb{G}_{ijt} = g_{ijt} \mid \mathbb{X}_{ij0:(t-1)} = \mu_{Fi0:(t-1)}) = p(\mathbb{G}_{ij0} = g_{ij0})$. Equation (A.1) is satisfied year 0. For $t \in \mathbb{N}$, we assume that

$$p(G_{ijt} = g_{ijt}) = p(\mathbb{G}_{ijt} = g_{ijt} \mid \mathbb{X}_{ij0:(t-1)} = \mu_{Fi0:(t-1)})$$

and show that

$$
\begin{aligned}
p(G'_{ijt} = g_{ijt}) &= p(\mathbb{G}_{ijt} = g_{ijt} \mid \mathbb{X}_{ij0:t} = \mu_{Fi0:t}), \\
p(G_{ij(t+1)} = g_{ij(t+1)}) &= p(\mathbb{G}_{ij(t+1)} = g_{ij(t+1)} \mid \mathbb{X}_{ij0:t} = \mu_{Fi0:t}).
\end{aligned}
$$

A.3   Selection

From (2), $p(G'_{ijt} = g_{ijt})$ writes

$$p(G'_{ijt} = g_{ijt}) = \frac{w_{it}(g_{ijt}) \, p(G_{ijt} = g_{ijt})}{\int_{\mathbb{R}} w_{it}(g_{ijt}) \, p(G_{ijt} = g_{ijt}) \, dg_{ijt}}.$$

We have

$$
\begin{aligned}
w_{it}(g_{ijt}) &= \int_{\mathbb{R}} F_{it}(z_{ijt}) \, p(Z_{ijt} = z_{ijt} \mid G_{ijt} = g_{ijt}) \, dz_{ijt} \\
&\propto \int_{\mathbb{R}} \mathcal{N}(z_{ijt}, \mu_{Fit}, \tau_{Fi}^{-2}) \, \mathcal{N}(z_{ijt}, g_{ijt} + A_{it} + B_{ijt}, \sigma_{Cijt}^2) \, dz_{ijt} \\
&\propto \int_{\mathbb{R}} \mathcal{N}(\mu_{Fit}, z_{ijt}, \tau_{Fi}^{-2}) \, \mathcal{N}(z_{ijt}, g_{ijt} + A_{it} + B_{ijt}, \sigma_{Cijt}^2) \, dz_{ijt} \\
&\propto \int_{\mathbb{R}} p(\mathbb{X}_{ijt} = \mu_{Fit} \mid \mathbb{Z}_{ijt} = z_{ijt}) \, p(\mathbb{Z}_{ijt} = z_{ijt} \mid \mathbb{G}_{ijt} = g_{ijt}) \, dz_{ijt} \\
&\propto \int_{\mathbb{R}} p(\mathbb{X}_{ijt} = \mu_{Fit} \mid \mathbb{Z}_{ijt} = z_{ijt}, \mathbb{G}_{ijt} = g_{ijt}) \, p(\mathbb{Z}_{ijt} = z_{ijt} \mid \mathbb{G}_{ijt} = g_{ijt}) \, dz_{ijt} \\
&\propto p(\mathbb{X}_{ijt} = \mu_{Fit} \mid \mathbb{G}_{ijt} = g_{ijt}).
\end{aligned}
$$

Thus we obtain

$$
\begin{aligned}
p(G'_{ijt} = g_{ijt}) &\propto p(\mathbb{X}_{ijt} = \mu_{Fit} \mid \mathbb{G}_{ijt} = g_{ijt}) \, p(\mathbb{G}_{ijt} = g_{ijt} \mid \mathbb{X}_{ij0:(t-1)} = \mu_{Fi0:(t-1)}) \\
&\propto p(\mathbb{X}_{ijt} = \mu_{Fit} \mid \mathbb{G}_{ijt} = g_{ijt}, \mathbb{X}_{ij0:(t-1)} = \mu_{Fi0:(t-1)}) \times \\
&\quad\; p(\mathbb{G}_{ijt} = g_{ijt} \mid \mathbb{X}_{ij0:(t-1)} = \mu_{Fi0:(t-1)}) \\
&\propto p(\mathbb{G}_{ijt} = g_{ijt} \mid \mathbb{X}_{ij0:t} = \mu_{Fi0:t}).
\end{aligned}
$$

Since the integrals of $p(G'_{ijt} = g_{ijt})$ and $p(\mathbb{G}_{ijt} = g_{ijt} \mid \mathbb{X}_{ij0:t} = \mu_{Fi0:t})$ with respect to g_{ijt} are both equal to 1, we obtain

$$p(G'_{ijt} = g_{ijt}) = p(\mathbb{G}_{ijt} = g_{ijt} \mid \mathbb{X}_{ij0:t} = \mu_{Fi0:t}).$$

A.4 Mutation

The distribution of genetic effects among adults writes

$$
\begin{aligned}
p(G_{ij(t+1)} = g_{ij(t+1)}) &= \int_{\mathbb{R}} p(G_{ij(t+1)} = g_{ij(t+1)} \mid G'_{ijt} = g_{ijt})\, p(G'_{ijt} = g_{ijt})\, dg_{ijt} \\
&= \int_{\mathbb{R}} p(\mathbb{G}_{ij(t+1)} = g_{ij(t+1)} \mid \mathbb{G}_{ijt} = g_{ijt})\, p(\mathbb{G}_{ijt} = g_{ijt} \mid \mathbb{X}_{ij0:t} = \mu_{Fi0:t})\, dg_{ijt} \\
&= \int_{\mathbb{R}} p(\mathbb{G}_{ij(t+1)} = g_{ij(t+1)} \mid \mathbb{G}_{ijt} = g_{ijt}, \mathbb{X}_{ij0:t} = \mu_{Fi0:t}) \times \\
&\qquad p(\mathbb{G}_{ijt} = g_{ijt} \mid \mathbb{X}_{ij0:t} = \mu_{Fi0:t})\, dg_{ijt} \\
&= p(\mathbb{G}_{ij(t+1)} = g_{ij(t+1)} \mid \mathbb{X}_{ij0:t} = \mu_{Fi0:t}).
\end{aligned}
$$

Thus for all $t \in \mathbb{N}$,

$$
\begin{aligned}
p(G'_{ijt} = g_{ijt}) &= p(\mathbb{G}_{ijt} = g_{ijt} \mid \mathbb{X}_{ij0:t} = \mu_{Fi0:t}), \\
p(G_{ijt} = g_{ijt}) &= p(\mathbb{G}_{ijt} = g_{ijt} \mid \mathbb{X}_{ij0:(t-1)} = \mu_{Fi0:(t-1)}).
\end{aligned}
$$

B MCMC

MCMC methods were implemented using the `R` package `rjags` (Plummer, 2016; R Core Team, 2018). Two MCMC chains were run independently for 50 000 iterations, after a burn-in of 1000 iterations, with a thinning interval of 5, so that 10 000 samples were available for inference for each chain. Convergence of MCMC chains was studied using the Gelman-Rubin test (Gelman and Rubin, 1992) of the `R` package `coda` (Plummer et al., 2006): for all the analyses carried out in this article and all the parameters monitored, the univariate and multivariate potential scale reduction factors were smaller than or equal to 1.1. Highest Posterior Density intervals were constructed using the `R` package `coda`.

C `R` session

This appendix shows how to implement the developed method in `R` (R Core Team, 2018). In this session, data are first simulated and are then analysed using `rjags` (Plummer, 2016).

```
> # License
> #
> # This R script is free; you can redistribute it and/or
> # modify it under the terms of the GNU General Public License
> # Version 3 as published by the Free Software Foundation.
> #
> # It is distributed in the hope that it will be
> # useful, but WITHOUT ANY WARRANTY; without even the implied warranty
> # of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
> # General Public License for more details.
>
> rm(list=ls())
>
> # Libraries
>
> library(rjags)
Loading required package: coda
Linked to JAGS 4.2.0
```

```
Loaded modules: basemod,bugs
>
> ###########################################################
> # Functions
>
> #>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
> data_simul <- function(n_loc,n_pop,n_year,n_sample,
+                         prior_mean,delta,rho,sigma_A,
+         sigma_B,sigma_eps,gamma_C,tau_F,
+         theta,sigma_M,loc=loc){
+
+ tau2_F <- tau_F*tau_F
+ tau2_M <- 1/sigma_M^2
+
+ B <-        array(NA,dim=c(n_pop,n_year))
+ A <-        array(NA,dim=c(n_loc,n_year))
+ eps <-      array(NA,dim=c(n_loc,n_year))
+ mu_G <-     array(NA,dim=c(n_pop,n_year))
+ tau2_G <-   array(NA,dim=c(n_pop,n_year))
+ tau2_S <-   array(NA,dim=c(n_pop,n_year))
+ sigma_C <-   array(NA,dim=c(n_pop,n_year))
+ tau2_Z <-   array(NA,dim=c(n_pop,n_year))
+ unscaled_ln_sigma_C <-  array(NA,dim=c(n_pop,n_year))
+ Z <- NULL
+
+ mu_G[,1] <- rep(rep(c(prior_mean-delta,prior_mean+delta),
+                   each=n_pop/(2*n_loc)),times=n_loc)
+ tau2_G[,1] <- 1/rep(rho,times=n_pop)^2
+
+ for (i in 1:n_loc) {
+ A[i,] <- rnorm(n_year,mean=0,sd=sigma_A)
+ eps[i,] <- rnorm(n_year,mean=0,sd=sigma_eps)
+ }
+
+ for (j in 1:n_pop) {
+ unscaled_ln_sigma_C[j,] <- rnorm(n_year,mean=0,sd=1)
+ sigma_C[j,] <- rho*exp(gamma_C*unscaled_ln_sigma_C[j,])
+ }
+
+ tau2_C <- 1/sigma_C^2
+
+ for (j in 1:n_pop) {
+ B[j,] <- rnorm(n_year,mean=0,sd=sigma_B)
+
+ tau2_Z[j,1] <- tau2_G[j,1]*tau2_C[j,1]/(tau2_G[j,1]+tau2_C[j,1])
+
+ Z_temp <- rnorm(n_sample,mean=mu_G[j,1]+A[loc[j],1]+B[j,1],
+                           sd=1/sqrt(tau2_Z[j,1]))
+ Z <-c(Z,Z_temp)
+
+ for (t in 2:n_year) {
+ tau2_S[j,t-1] <- tau2_F*tau2_C[j,t-1]/(tau2_F+tau2_C[j,t-1])
+
+ mu_G[j,t] <- (tau2_G[j,t-1]*mu_G[j,t-1]+
```

```
+  tau2_S[j,t-1]*(theta[loc[j]]+eps[loc[j],t-1]-A[loc[j],t-1]-B[j,t-1]))/
+  (tau2_G[j,t-1]+tau2_S[j,t-1])
+
+ tau2_G[j,t] <- tau2_M*(tau2_G[j,t-1]+tau2_S[j,t-1])/
+  (tau2_M+tau2_G[j,t-1]+tau2_S[j,t-1])
+
+ tau2_Z[j,t] <- tau2_G[j,t]*tau2_C[j,t]/(tau2_G[j,t]+tau2_C[j,t])
+
+ Z_temp <- rnorm(n_sample,mean=mu_G[j,t]+A[loc[j],t]+B[j,t],
+                 sd=1/sqrt(tau2_Z[j,t]))
+ Z <-c(Z,Z_temp)
+ }
+ }
+
+ return(Z)
+ }
> #>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>
> ###########################################################
> # Parameters
>
> # Parameters for data simulation
> # There are 10 locations, 5 years and 10 populations per location.
> # Selection is present in each location with a strength of 1.
> n_year <- 5
> n_pop <- 100
> n_loc <- 10
> loc <- rep(1:n_loc,each=n_pop/n_loc)
> n_sample <- 20
> prior_mean <- 2
> prior_sd <- 0.8
> tau_F <- 1
> sigma_B <- 0.2
> sigma_A <- 0.2
> rho <- 0.3
> gamma_C <- 0.1
> sigma_eps <- 0.1
> sigma_M <- 0.1
> delta <- 0.3
> Delta <- 0.3
> theta <- rep(c(prior_mean-Delta,prior_mean+Delta),each=n_loc/2)
>
> # Parameters for bayesian inference
> # Mainly parameters of prior distributions
> nb_iterations <- 50000
> prior_location_theta <- prior_mean
> prior_scale_theta <- prior_sd*0.75
> prior_scale_omega <- 0.2
> prior_scale_sigma_M <- prior_sd/4
> prior_scale_sigma_A <- prior_sd/sqrt(2)
> prior_scale_sigma_B <- prior_sd/sqrt(2)
> prior_scale_rho <- prior_sd/sqrt(2)
> prior_scale_sigma_eps <- prior_sd/5
> prior_location_mu_G <- prior_mean
```

```
> prior_scale_mu_G <- prior_sd
> prior_scale_sigma_G <- prior_sd/10
> prior_scale_gamma_C <- 0.3
>
> ############################################################
> # Data simulation
>
> set.seed(1234)
>
> year <- rep(rep(1:n_year,each=n_sample),times=n_pop)
> pop <- rep(1:n_pop,each=n_sample*n_year)
>
> Z <- data_simul(
+         n_loc=n_loc,
+         n_pop=n_pop,
+         n_year=n_year,
+         n_sample=n_sample,
+         prior_mean=prior_mean,
+         delta=delta,
+         rho=rho,
+         sigma_A=sigma_A,
+         sigma_B=sigma_B,
+         sigma_eps=sigma_eps,
+         gamma_C=gamma_C,
+         tau_F=tau_F,
+         theta=theta,
+         sigma_M=sigma_M,
+         loc=loc)
>
> print(summary(Z))
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.09057 1.57255 1.96739 1.96710 2.34941 4.36854
>
> ############################################################
> # Bayesian inference
>
> # Data
> d <- list(Z=Z,n_obs=length(Z),
+           year=year,n_year=n_year,
+           pop=pop,n_pop=n_pop,
+           loc=loc,n_loc=n_loc,
+     prior_location_theta=prior_location_theta,
+           prior_scale_theta=prior_scale_theta,
+           prior_scale_omega=prior_scale_omega,
+           prior_scale_sigma_M=prior_scale_sigma_M,
+           prior_scale_sigma_A=prior_scale_sigma_A,
+           prior_scale_sigma_B=prior_scale_sigma_B,
+           prior_scale_sigma_eps=prior_scale_sigma_eps,
+           prior_location_mu_G=prior_location_mu_G,
+           prior_scale_mu_G=prior_scale_mu_G,
+           prior_scale_sigma_G=prior_scale_sigma_G,
+     prior_scale_rho = prior_scale_rho,
+     prior_scale_gamma_C = prior_scale_gamma_C,
+           prior_sd=prior_sd)
```

```
>
> # Initial values
> init1 <- list(".RNG.name"="base::Mersenne-Twister",
+ ".RNG.seed"=1234)
>
> init2 <- list(".RNG.name"="base::Wichmann-Hill",
+ ".RNG.seed"=5678)
>
> init <- list(init1,init2)
>
> # Model
> modelString ="
+ model {
+ #
+ # Likelihood
+ #
+
+ for (i in 1:n_obs) {
+ Z[i] ~ dnorm(mu_Z[pop[i],year[i]],tau2_Z[pop[i],year[i]])
+ }
+
+ for (j in 1:n_pop) {
+ for (t in 2:n_year) {
+ tau2_S[j,t-1] <- tau2_F[loc[j]]*tau2_C[j,t-1]/(tau2_F[loc[j]]+tau2_C[j,t-1])
+
+ mu_G[j,t] <- (tau2_G[j,t-1]*mu_G[j,t-1]+
+ tau2_S[j,t-1]*(theta[loc[j]]+eps[loc[j],t-1]-A[loc[j],t-1]-B[j,t-1]))/
+ (tau2_G[j,t-1]+tau2_S[j,t-1])
+
+ tau2_G[j,t] <- tau2_M*(tau2_G[j,t-1]+tau2_S[j,t-1])/
+ (tau2_M+tau2_G[j,t-1]+tau2_S[j,t-1])
+ sigma_G[j,t] <- pow(tau2_G[j,t],-0.5)
+ }}
+
+ for (j in 1:n_pop) {
+ for (t in 1:n_year) {
+ mu_Z[j,t] <- mu_G[j,t]+A[loc[j],t]+B[j,t]
+ tau2_Z[j,t] <- tau2_G[j,t]*tau2_C[j,t]/(tau2_G[j,t]+tau2_C[j,t])
+ sigma_Z[j,t] <- pow(tau2_Z[j,t],-0.5)
+ }}
+
+ #
+ # Priors
+ #
+
+ for (i in 1:n_loc) {
+ theta[i] ~ dnorm(prior_location_theta,pow(prior_scale_theta,-2))
+ tau2_F[i] <- pow(tau_F[i],2)
+ tau_F[i] ~ dnorm(0,pow(omega/prior_sd,-2)) T(0,)
+ }
+
+ omega ~ dnorm(0,pow(prior_scale_omega,-2)) T(0,)
+
+ tau2_M <- pow(sigma_M,-2)
```

```
+ sigma_M ~ dnorm(0,pow(prior_scale_sigma_M,-2)) T(0,)
+
+ for (i in 1:n_loc) {
+ for (t in 1:n_year) {
+ A[i,t] ~ dnorm(0,tau2_A)
+ eps[i,t] ~ dnorm(0,tau2_eps)
+ }}
+ tau2_A <- pow(sigma_A,-2)
+ sigma_A ~ dnorm(0,pow(prior_scale_sigma_A,-2)) T(0,)
+ tau2_eps <- pow(sigma_eps,-2)
+ sigma_eps ~ dnorm(0,pow(prior_scale_sigma_eps,-2)) T(0,)
+
+ for (j in 1:n_pop) {
+ for (t in 1:n_year) {
+ B[j,t] ~ dnorm(0,tau2_B)
+ }}
+ tau2_B <- pow(sigma_B,-2)
+ sigma_B ~ dnorm(0,pow(prior_scale_sigma_B,-2)) T(0,)
+
+ for (j in 1:n_pop) {
+ for (t in 1:n_year) {
+ tau2_C[j,t] <- pow(sigma_C[j,t],-2)
+ sigma_C[j,t] <- rho[loc[j]]*exp(gamma_C*unscaled_ln_sigma_C[j,t])
+ unscaled_ln_sigma_C[j,t] ~ dnorm(0,1)
+ }}
+
+ for (i in 1:n_loc) {
+ rho[i] ~ dnorm(0,pow(prior_scale_rho,-2)) T(0,)
+ }
+
+ gamma_C ~ dnorm(0,pow(prior_scale_gamma_C,-2)) T(0,)
+
+ for (j in 1:n_pop) {
+ mu_G[j,1] ~ dnorm(prior_location_mu_G,pow(prior_scale_mu_G,-2))
+ tau2_G[j,1] <- pow(sigma_G[j,1],-2)
+ sigma_G[j,1] ~ dnorm(rho[loc[j]],pow(prior_scale_sigma_G,-2)) T(0,)
+ }
+
+ }
+ "
> writeLines(modelString, con="model.txt")
>
> model <- jags.model(file="model.txt",
+                data=d,inits=init,n.chains=2)
Compiling model graph
   Resolving undeclared variables
   Allocating nodes
Graph information:
   Observed stochastic nodes: 10000
   Unobserved stochastic nodes: 1336
   Total graph size: 47260

Initializing model
```

```
>
> # Burn-in
> update(model,1000)
>
> # MCMC
> rslt <- coda.samples(model,
+                 c("omega","tau_F","sigma_eps","theta","sigma_M"),
+                 n.iter=nb_iterations,thin = 5)
>
>
> # MCMC convergence
> gelman.diag(rslt, confidence = 0.95, transform=FALSE, multivariate=TRUE)
Potential scale reduction factors:

          Point est. Upper C.I.
omega          1.00       1.01
sigma_M        1.00       1.00
sigma_eps      1.00       1.02
tau_F[1]       1.00       1.00
tau_F[2]       1.00       1.00
tau_F[3]       1.00       1.00
tau_F[4]       1.00       1.00
tau_F[5]       1.01       1.03
tau_F[6]       1.00       1.00
tau_F[7]       1.00       1.00
tau_F[8]       1.00       1.01
tau_F[9]       1.00       1.00
tau_F[10]      1.00       1.01
theta[1]       1.00       1.00
theta[2]       1.00       1.01
theta[3]       1.00       1.02
theta[4]       1.00       1.00
theta[5]       1.00       1.00
theta[6]       1.00       1.00
theta[7]       1.00       1.00
theta[8]       1.02       1.09
theta[9]       1.01       1.07
theta[10]      1.00       1.00


Multivariate psrf


1.03
>
> # Results
> summary_post <- summary(rslt)
> summary_post$quantiles
                 2.5%        25%        50%        75%       97.5%
omega     0.660885637  0.7780296 0.84609725 0.9165928  1.0666966
sigma_M   0.107104427  0.1231470 0.13058775 0.1378170  0.1513753
sigma_eps 0.002824992  0.0385993 0.09197706 0.1641287  0.3228490
tau_F[1]  0.633840662  1.1660364 1.43315419 1.7161913  2.3812936
tau_F[2]  0.689579702  1.1254569 1.36339251 1.6304208  2.2912915
tau_F[3]  0.891256441  1.3907339 1.66965172 1.9891076  2.7799521
tau_F[4]  1.156722739  1.6113232 1.87926551 2.1952476  2.9323318
```

Trait	1		2	
T	4	6	4	6
σ_A	0.2	0.4	30	100
σ_B	0.2	0.3	30	50
γ_C	0.1	0.2	0.1	0.2
δ	0.3	0.6	100	200
τ_{Fi}	0	1	0	0.003
ρ_i	0.3		50	
σ_ε	0.1		20	
σ_M	0.1		20	
μ_{G0}	2		1100	
σ_{Gij0}	0.3		50	
Δ	0.3		100	

Table D.1: Parameter values used in the simulation study.

```
tau_F[5]   1.055984766 1.5266916 1.80897160 2.1255554 2.8739377
tau_F[6]   0.989370636 1.4182525 1.66434805 1.9431110 2.6182511
tau_F[7]   0.397942630 1.0016914 1.25986761 1.5268167 2.1462668
tau_F[8]   1.224467521 1.6432737 1.92038880 2.2349449 3.0081915
tau_F[9]   0.841599003 1.3055560 1.58003674 1.8868735 2.6343188
tau_F[10]  1.026657183 1.5150439 1.78582278 2.0922071 2.8012392
theta[1]   1.128864110 1.5798028 1.78897576 1.9952936 2.4444526
theta[2]   1.525662464 1.9432239 2.15129755 2.3780776 2.8894340
theta[3]   1.022284554 1.4239799 1.61558728 1.8002632 2.1809490
theta[4]   0.912795981 1.2895075 1.47325010 1.6534730 2.0222713
theta[5]   0.990114389 1.4072594 1.58853900 1.7599639 2.0917839
theta[6]   1.556301903 1.9199542 2.09655683 2.2778145 2.6866019
theta[7]   1.168978775 1.7370586 1.97841410 2.2110184 2.7248737
theta[8]   2.129372241 2.4608864 2.63436452 2.8129043 3.2224737
theta[9]   1.828794522 2.2166851 2.40677659 2.6118409 3.0865847
theta[10]  1.672702054 2.0518046 2.22743498 2.4067346 2.7857430
```

D Simulation study

D.1 Data

Data were simulated for two quantitative traits using the parameter values given in Table D.1.

D.2 Analysis

The simulated data were analysed using the model of Sections 2 and 3. Prior distributions were the same as in Section E except that $\gamma_\omega = 0.2$.

D.3 Results

Posterior medians and credible intervals of selection strengths and mean optimal phenotypes for trait 1 (respectively trait 2) and replicate 1 are shown in Fig. D.1 and D.2 (respectively Fig. D.3 and D.4).
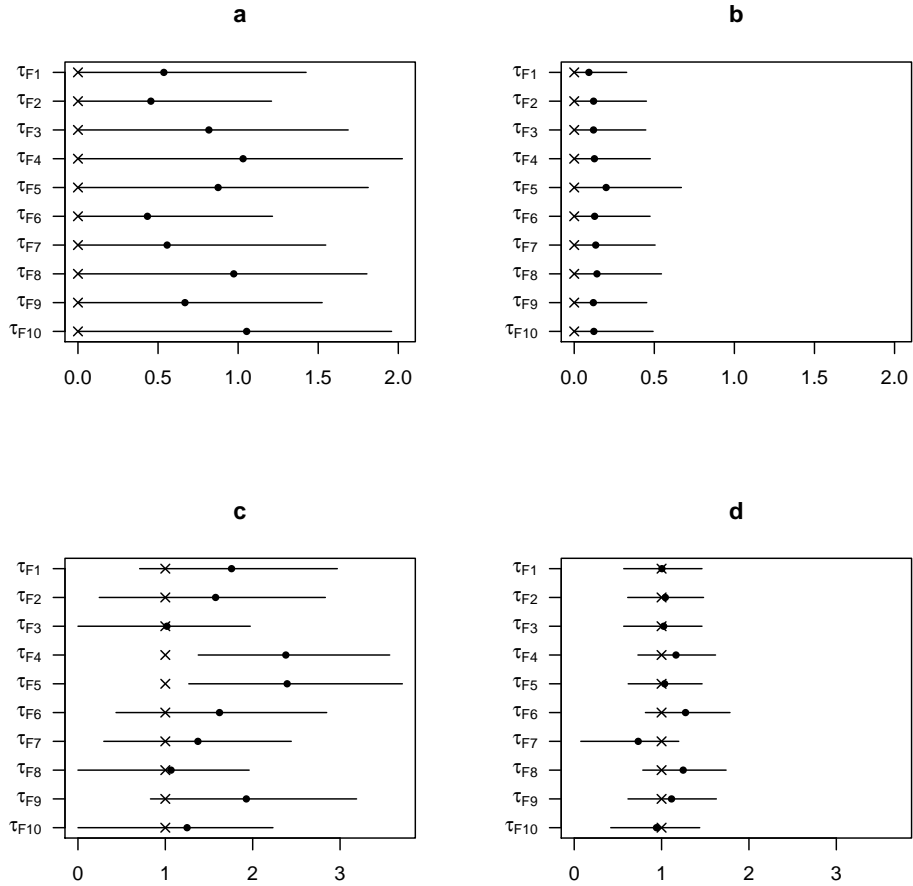
Figure D.1: Analysis of simulated data: estimates ($\bullet$), true values ($\times$) and 95% HPD credible intervals (solid lines) of selection strengths for trait 1 and replicate 1. a: Weakly informative set-up without selection; b: Informative set-up without selection; c: Weakly informative set-up with selection; d: Informative set-up with selection. In these analyses, $\gamma_\omega = 0.2$.
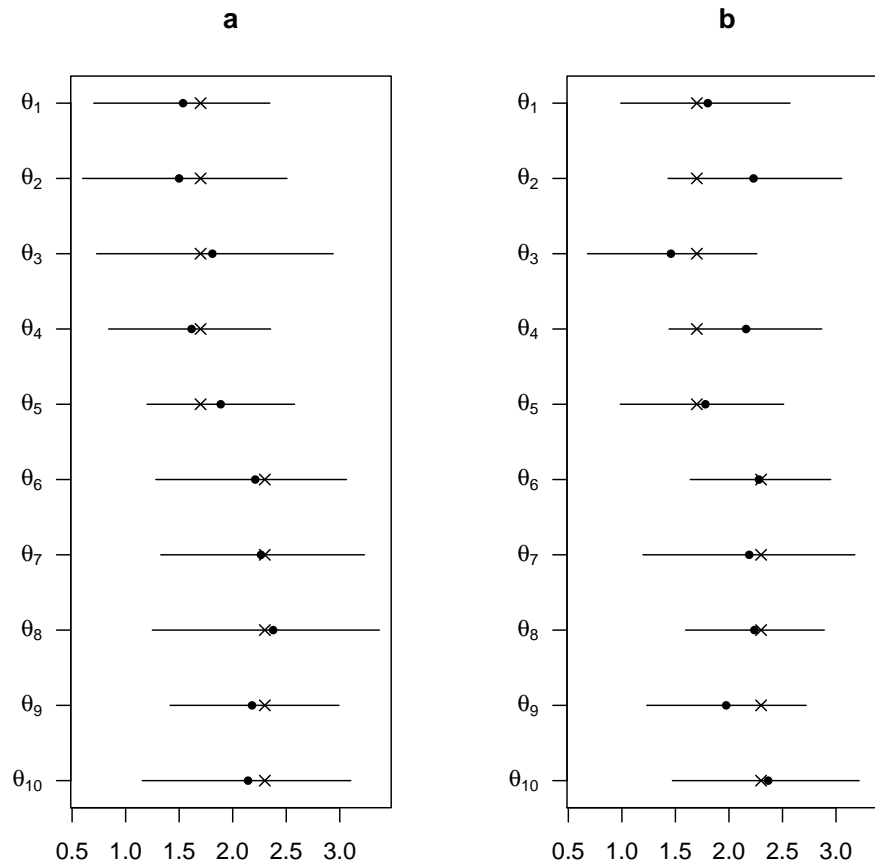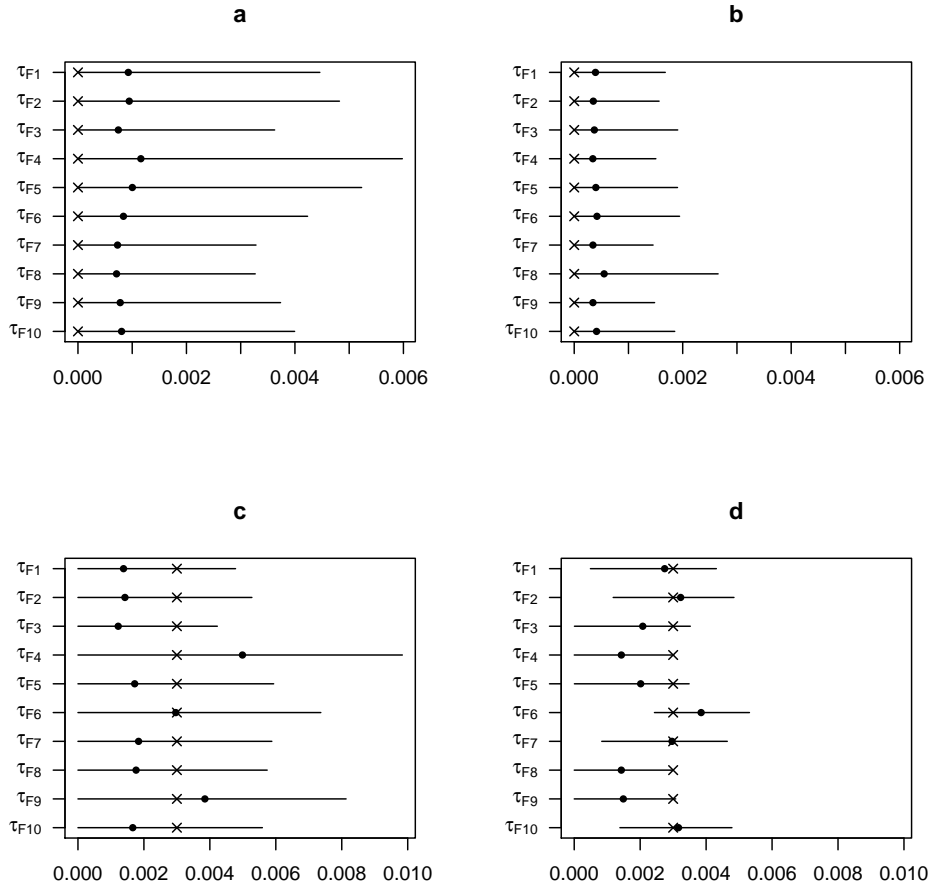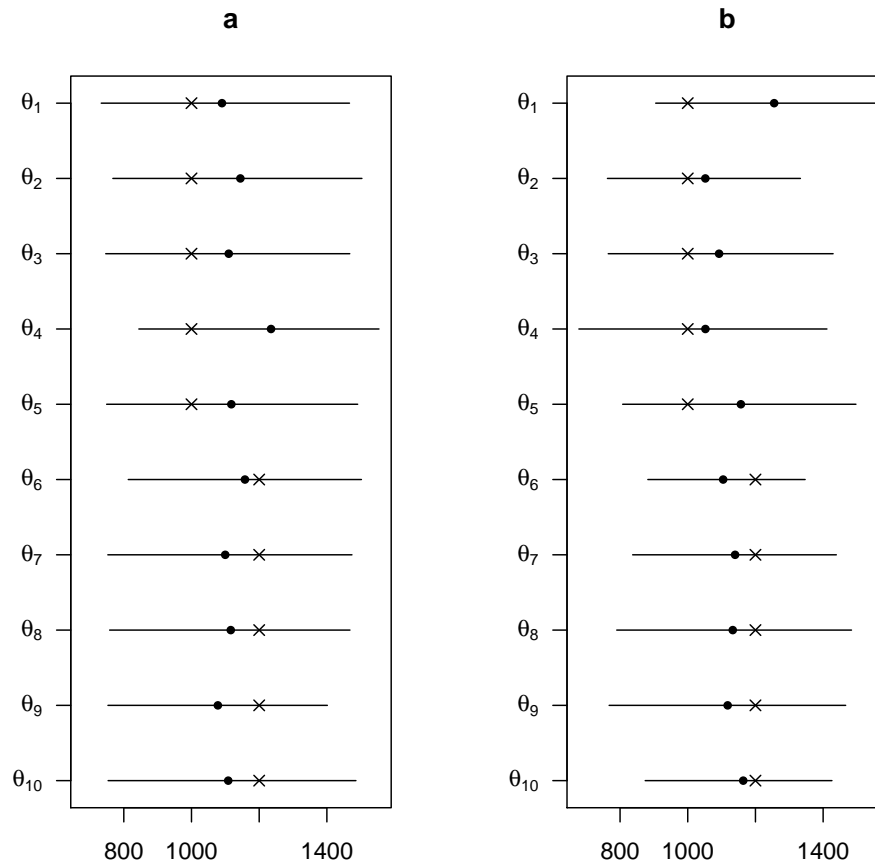
Figure D.2: Analysis of simulated data: estimates (•), true values (×) and 95% HPD credible intervals (solid lines) of mean optimal phenotypes for trait 1 and replicate 1. a: Weakly informative set-up with selection; b: Informative set-up with selection. In these analyses, $\gamma_\omega = 0.2$.

Figure D.3: Analysis of simulated data: estimates ($\bullet$), true values ($\times$) and 95% HPD credible intervals (solid lines) of selection strengths for trait 2 and replicate 1. a: Weakly informative set-up without selection; b: Informative set-up without selection; c: Weakly informative set-up with selection; d: Informative set-up with selection. In these analyses, $\gamma_\omega = 0.2$.

Figure D.4: Analysis of simulated data: estimates ($\bullet$), true values ($\times$) and 95% HPD credible intervals (solid lines) of mean optimal phenotypes for trait 2 and replicate 1. a: Weakly informative set-up with selection; b: Informative set-up with selection. In these analyses, $\gamma_\omega = 0.2$.

| | Number of | | Largest phenotypic |
Farm	populations	years	difference
1	5	3	0.357
2	18	5.67	0.592
3	4	3	1.235
4	16	3.81	0.625
5	6	3	0.598
6	7	3.43	0.924
7	12	3	1.076
8	8	3	0.829
9	11	4.18	0.825
10	7	3	0.592
11	9	4	1.067
12	3	3	0.432

Table E.1: Spike weight data. The number of years corresponds to the mean number of consecutive years a population is sown in a farm. The largest phenotypic difference is the largest difference between population phenotypic means the first year (in g).

| | Number of | | Largest phenotypic |
Farm	populations	years	difference
1	12	4.75	392.4
2	18	4.72	333.2
3	6	3	284.7
4	16	3.81	249.3
5	12	4.75	195.3
6	11	4.64	206.9
7	11	4.55	344.2

Table E.2: Plant height data. The number of years corresponds to the mean number of consecutive years a population is sown in a farm. The largest phenotypic difference is the largest difference between population phenotypic means the first year (in mm).

E   Analysis of wheat data

E.1   Data

Tab. E.1 and E.2 describe the data collected in each farm for spike weight and plant height. The sample means of spike weight and plant height were equal to 2.086 g and 1149.596 mm respectively. The sample standard deviations of spike weight and plant height were equal to 0.842 g and 247.898 mm respectively.

E.2   Prior distribution

Spike weight and plant height data were analysed using the following prior distributions:

$$
\begin{aligned}
\theta_i &\sim \mathcal{N}(\tilde{\mu}_Z, (0.75\,\tilde{\sigma}_Z)^2),\ \tau_{Fi} \sim \mathcal{N}^+(0, (\omega/\tilde{\sigma}_Z)^2),\ \sigma_\varepsilon \sim \mathcal{N}^+(0, (\tilde{\sigma}_Z/5)^2), \\
\omega &\sim \mathcal{N}^+(0, \gamma_\omega^2),\ \sigma_M \sim \mathcal{N}^+(0, (\tilde{\sigma}_Z/4)^2), \\
\sigma_A &\sim \mathcal{N}^+(0, \tilde{\sigma}_Z^2/2),\ \sigma_B \sim \mathcal{N}^+(0, \tilde{\sigma}_Z^2/2), \\
\rho_i &\sim \mathcal{N}^+(0, \tilde{\sigma}_Z^2/2),\ \gamma_C \sim \mathcal{N}^+(0, 0.3^2), \\
\mu_{Gij0} &\sim \mathcal{N}(\tilde{\mu}_Z, \tilde{\sigma}_Z^2),\ \sigma_{Gij0} \sim \mathcal{N}^+(\rho_i, (\tilde{\sigma}_Z/10)^2),
\end{aligned}
$$

where $i = 1, \ldots, I$, $j = 1, \ldots, J_i$, $\tilde{\mu}_Z$ (respectively $\tilde{\sigma}_Z$) was a prior value for phenotypic mean (respectively overall phenotypic standard deviation, which took account of all possible sources of variation). These quantities were equal to $\tilde{\mu}_Z = 2$ (respectively $\tilde{\mu}_Z = 1100$) and $\tilde{\sigma}_Z = 0.8$ (respectively $\tilde{\sigma}_Z = 250$) for spike weight (respectively plant height). Parameter γ_ω took two possible values: $\gamma_\omega = 0.1$ or $\gamma_\omega = 0.4$.

References

Del Moral, P., Vergé, C., 2014. Modèles et méthodes stochastiques: Une introduction avec applications. volume 75. Springer Science & Business Media.

Durbin, J., Koopman, S.J., 2001. Time series analysis by state space methods. Oxford university press.

Gelman, A., Rubin, D.B., 1992. Inference from iterative simulation using multiple sequences. Statistical science 7, 457–472.

Plummer, M., 2016. rjags: Bayesian Graphical Models using MCMC. URL: `http://CRAN.R-project.org/package=rjags`. R package version 4-6.

Plummer, M., Best, N., Cowles, K., Vines, K., 2006. Coda: Convergence diagnosis and output analysis for mcmc. R News 6, 7–11. URL: `https://journal.r-project.org/archive/`.

R Core Team, 2018. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. Vienna, Austria. URL: `https://www.R-project.org/`.