Lathyrus ms2: Selection on reaction norms - multivariate modeling for phenotypic selection on plasticity 1 (Houslay & Wilson 2017)

Contents

Random regression mixed model (RRMM) using MCMCglmm 1

Bivariate model with random regression on one trait 5

Code based on Houslay & Wilson 2017 (finally not used! - Code on Arnold et al. 2019 Phil. Trans. R. Soc. B. seems better).

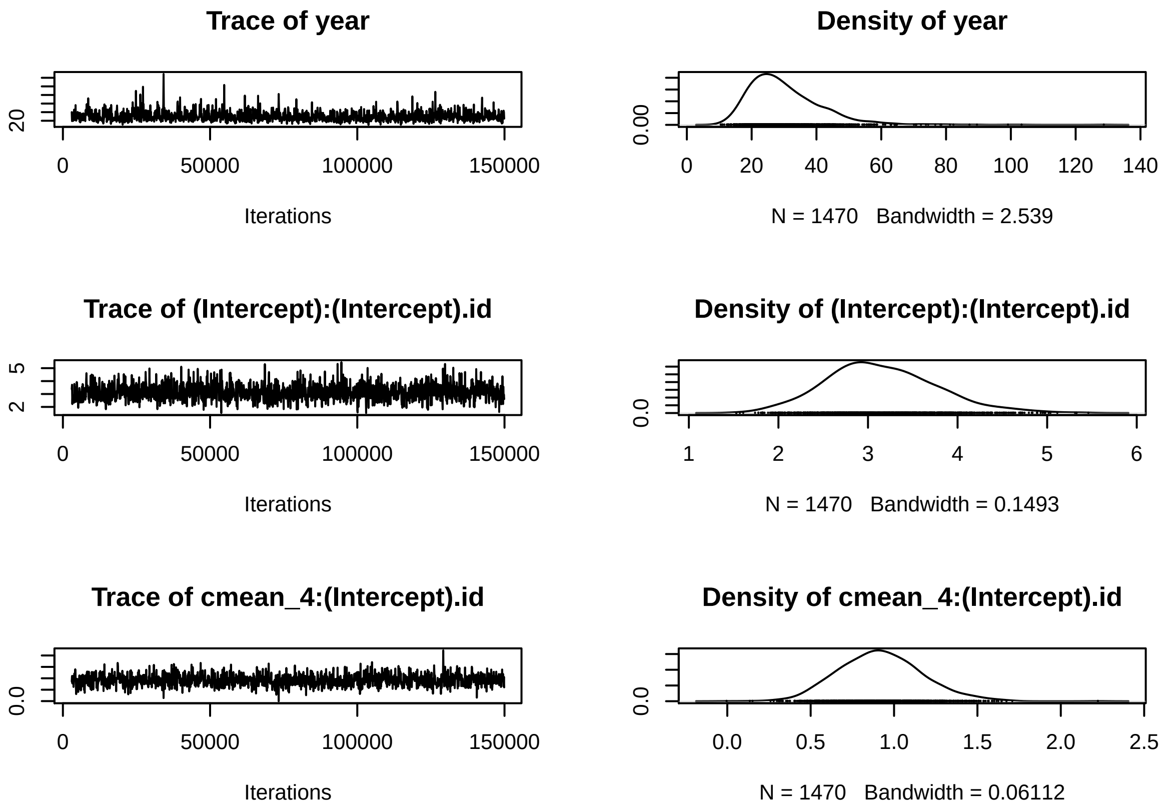
Random regression mixed model (RRMM) using MCMCglmm

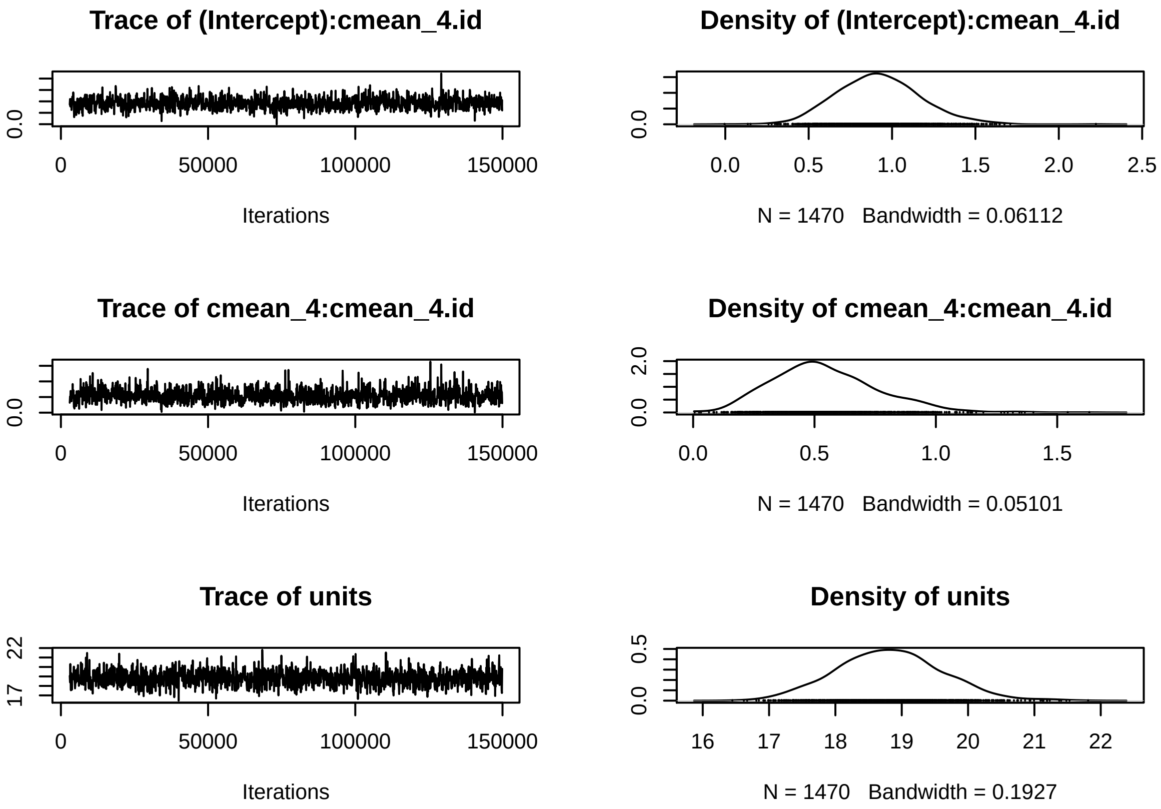
Fit the model:

```
# Parameter-expanded prior should be uninformative
# for variances and covariance
prior_RR <- list(R = list(V = 1, nu = 0.002),
                 G = list(G1 = list(V = 1, nu = 1,
                                    alpha.mu = 0,
                                    alpha.V= 25^2),
                          G2 = list(V = diag(2), nu = 2,
                                    alpha.mu = rep(0, 2),
                                    alpha.V= diag(25^2, 2, 2))))
# I tweaked this prior from the one below, it works but not 100% sure what I'm doing
# READ ON PRIOR SPECIFICATIOM!

# prior_RR <- list(R = list(V = 1, nu = 0.002),
#                  G = list(G1 = list(V = diag(2), nu = 2,
#                                     alpha.mu = rep(0, 2),
#                                     alpha.V= diag(25^2, 2, 2))))

mcmc_A_RR <- MCMCglmm(FFD ~ cmean_4,
                      # fixed effect of (mean-centred) mean daily temperature April
                      #random =~ year + us(1 + cmean_4):id,
                      random=~ year + us(1 + cmean_4):id,
                      # random effects: year (to take account of the repeated measures
                      rcov =~ units,         # no specific structure for the residuals
                      family = "gaussian",
                      prior =  prior_RR,     # parameter-expanded prior
                      #nitt=750000,burnin=50000,thin=350, #provided
                      #nitt=500000,burnin=10000,thin=10,  #ms1
                      nitt=150000,burnin=3000,thin=100,     #trial
                      #nitt=10000,burnin=1000,thin=50,    # smaller numbers for fast run
                      verbose = FALSE,
                      data = as.data.frame(data_4yrs),
                      pr = TRUE, # Saves posterior distribution of
                      # individual random effects (analagous to BLUP)
                      saveX = TRUE, # Save fixed effect design matrix
                      saveZ = TRUE) # Save random effect design matrix
```

Check model diagnostics using plots of the MCMC samples. Look at the trace/density plots of the posterior distributions for the (co)variances

```
plot(mcmc_A_RR$VCV)
```





For any real analysis various other checks and tests (e.g. of autocorrelation, robustness to different priors, and good model convergence using the geweke.diag and gelman.diag diagnostic functions) should be used before accepting final results.

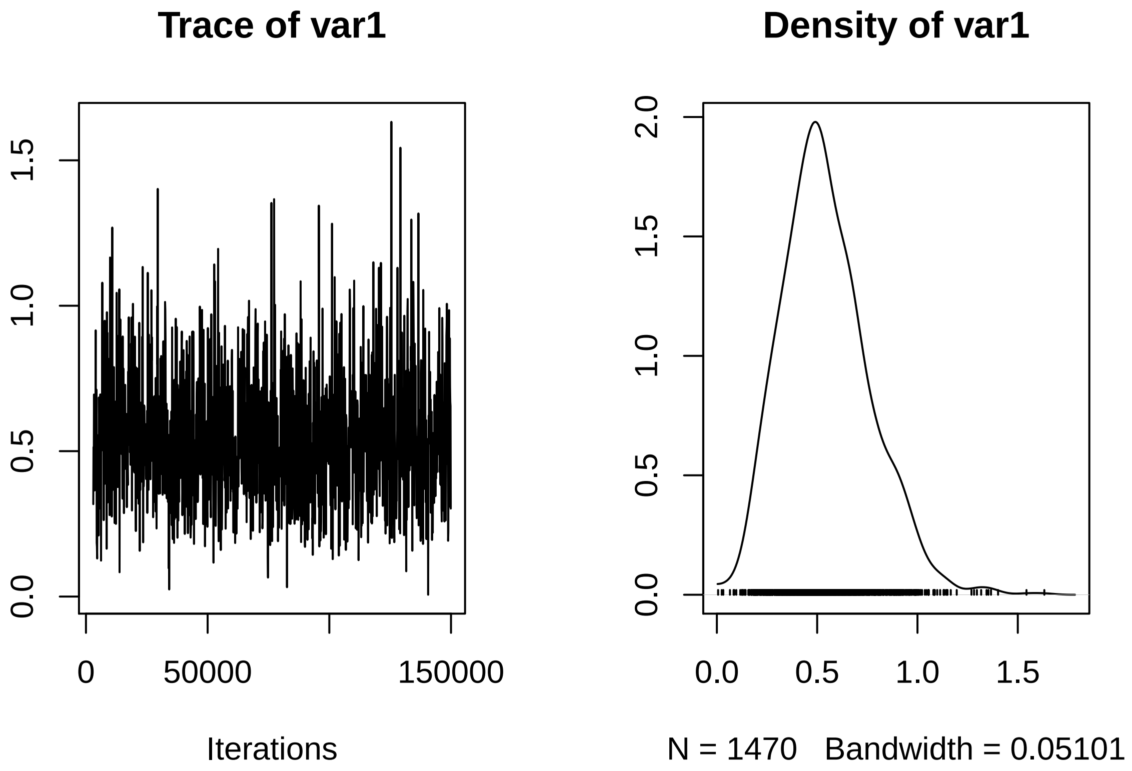
Check for autocorrelation between successive stored iterations (suggested to be less than 0.1):

```
diag(autocorr(mcmc_A_RR$VCV)[2, , ])
```

```
##                       year (Intercept):(Intercept).id
##                0.079490369               -0.002795354
##     cmean_4:(Intercept).id     (Intercept):cmean_4.id
##                0.032962212                0.032962212
##         cmean_4:cmean_4.id                      units
##                0.074464844                0.016087224
```

Given that a random regression allows IxE, we want to investigate whether there is support for the hypothesis that individuals vary in the slope of FFD against temperature. Look at the trace and density plots for the posterior distribution of the among-individual variation in slopes term:

```
plot(mcmc_A_RR$VCV[,"cmean_4:cmean_4.id"])
```



```
mean(mcmc_A_RR$VCV[,"cmean_4:cmean_4.id"])
```

```
## [1] 0.5443935
```

```
HPDinterval(mcmc_A_RR$VCV[,"cmean_4:cmean_4.id"])
```

```
##          lower     upper
## var1 0.1708802 0.9715408
## attr(,"Probability")
## [1] 0.9496599
```

The posterior distribution for slope variance looks good, and the credible intervals show that the lower bound is not close to zero

```
summary(mcmc_A_RR)
```

```
##
##  Iterations = 3001:149901
##  Thinning interval  = 100
##  Sample size  = 1470
##
##  DIC: 8565.367
##
##  G-structure:  ~year
##
##      post.mean l-95% CI u-95% CI eff.samp
## year     30.36    13.48    51.95     1253
##
```

```
##                  ~us(1 + cmean_4):id
##
##                               post.mean l-95% CI u-95% CI eff.samp
## (Intercept):(Intercept).id    3.1510   1.9689   4.3872     1470
## cmean_4:(Intercept).id        0.9203   0.4683   1.4363     1470
## (Intercept):cmean_4.id        0.9203   0.4683   1.4363     1470
## cmean_4:cmean_4.id            0.5444   0.1709   0.9715     1265
##
##  R-structure:  ~units
##
##       post.mean l-95% CI u-95% CI eff.samp
## units     18.82    17.23    20.32     1470
##
##  Location effects: FFD ~ cmean_4
##
##             post.mean l-95% CI u-95% CI eff.samp  pMCMC
## (Intercept)   58.1014  55.6868  60.3216     1326 <7e-04 ***
## cmean_4       -2.4133  -4.2378  -0.7569     1470 0.0122 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(Intercept):(Intercept).id Among-individual variation in intercepts cmean_4:(Intercept).id Covariances (Intercept):cmean_4.id Covariances cmean_4:cmean_4.id Among-individual variation in slopes

There is among-individual variation in plasticity (IxE) — i.e., individuals change their phenology at different rates in response to temperature.

For interpretation of the intercept-slope covariance, it is often easier to convert this to a correlation. Here we can use the formula for a correlation with the posterior distributions of our (co)variance components, giving us a distribution of correlation values that we can use to calculate estimates and 95% credible intervals:

```
mcmc_cor_RR <- mcmc_A_RR$VCV[,"cmean_4:(Intercept).id"]/
(sqrt(mcmc_A_RR$VCV[,"(Intercept):(Intercept).id"])*
sqrt(mcmc_A_RR$VCV[,"cmean_4:cmean_4.id"]))
posterior.mode(mcmc_cor_RR)
```

```
##      var1
## 0.8520873
```

```
HPDinterval(mcmc_cor_RR)
```

```
##          lower    upper
## var1 0.4536982 0.985781
## attr(,"Probability")
## [1] 0.9496599
```

We find a strong positive correlation between among-individual variance in intercepts and slopes, at the intercept (x = 0).

Bivariate model with random regression on one trait

Having established that there is among-individual variation in plasticity, we want to now test whether there is an association between variation in FFD (intercept and/or slope) and (relative) fitness. We use mean fitness over all years (relativized).

Add mean fitness over all years (relativized) to data (only one value of fitness for each individual, added at the first year)

```r
data_3yrs<-data_3yrs%>%
  right_join(data_3yrs_total[c(1,5)],by="id")%>%
  group_by(id)%>%
  mutate(mean_fitness_rel=ifelse(year==min(year),mean_fitness_rel,NA))%>%
  ungroup()
data_4yrs<-data_4yrs%>%
  right_join(data_4yrs_total[c(1,7)],by="id")%>%
  group_by(id)%>%
  mutate(mean_fitness_rel=ifelse(year==min(year),mean_fitness_rel,NA))%>%
  ungroup()
data_5yrs<-data_5yrs%>%
  right_join(data_5yrs_total[c(1,5)],by="id")%>%
  group_by(id)%>%
  mutate(mean_fitness_rel=ifelse(year==min(year),mean_fitness_rel,NA))%>%
  ungroup()
```

We have two response variables, but only one of them (FFD) has repeated observations at the individual level and is a function of the x variable (temperature). We can set up the covariance matrix such that we estimate random intercepts of both traits (FFD and fitness), and random slopes for trait 1 only (FFD).

In addition, as fitness is measured only once then we need to constrain the residual/'within-individual' variance such that it is effectively zero (i.e., meaning that all the variance in fitness will be in the among-individual level). We can do this in the R section of the prior, using the fix keyword. Note that the value of 0.0001 is adequate here – a smaller value, such as 1e-08, can cause problems with the chain mixing. For the among-individual section, G, we use an uninformative parameter-expanded prior for a 3x3 covariance matrix.

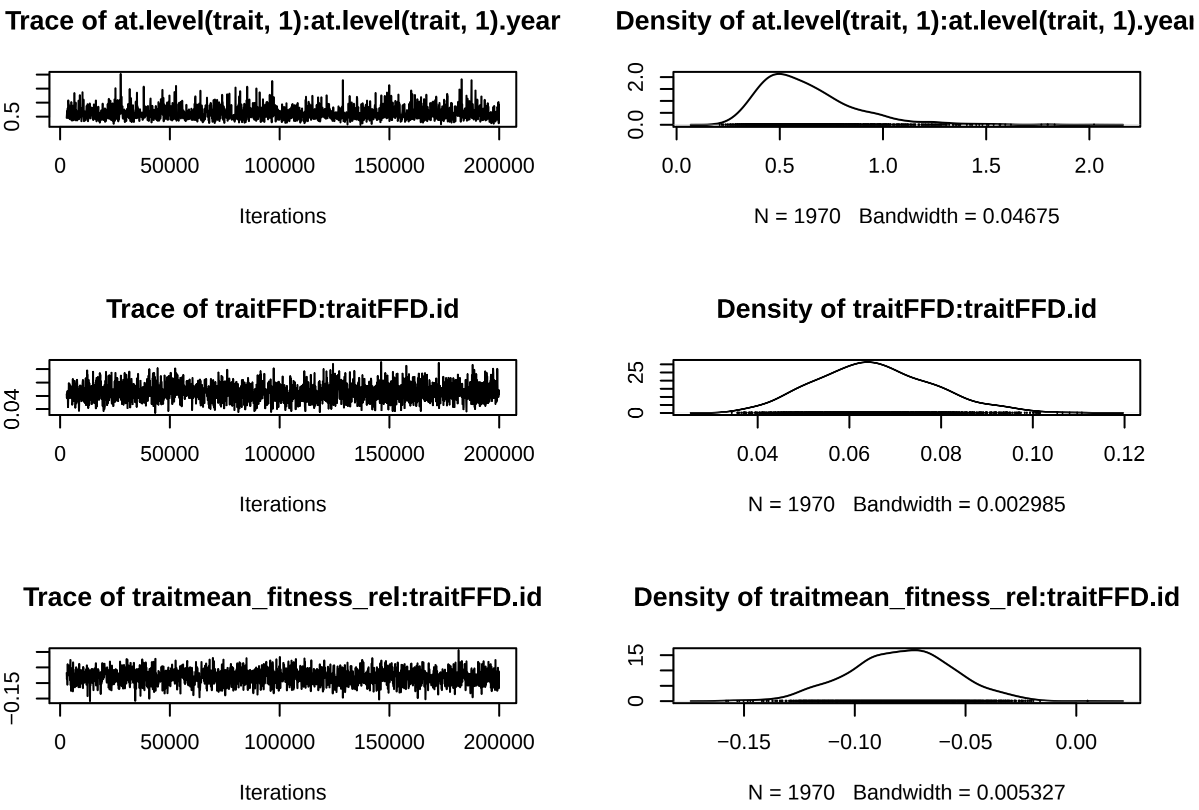
Within the model specification itself, we standardise FFD by scaling it (to make the multivariate model easier to fit), and we have already standardised fitness as relative fitness earlier on. The at.level keyword specifies fixed effects as relating to only one of our response variables — here, we have the fixed effect of temperature for FFD (but not for fitness). We set up our random effects in a similar way to the univariate random regression, but here we have 3 variance terms – random intercepts for both response traits, and random slopes for only the first response trait, FFD (we use the at.level keyword here just as we did in the fixed effects). We then fit residual variances for each of our response variables, but do not model the covariance between them (and remember that we have set residual variance in fitness to be essentially zero).

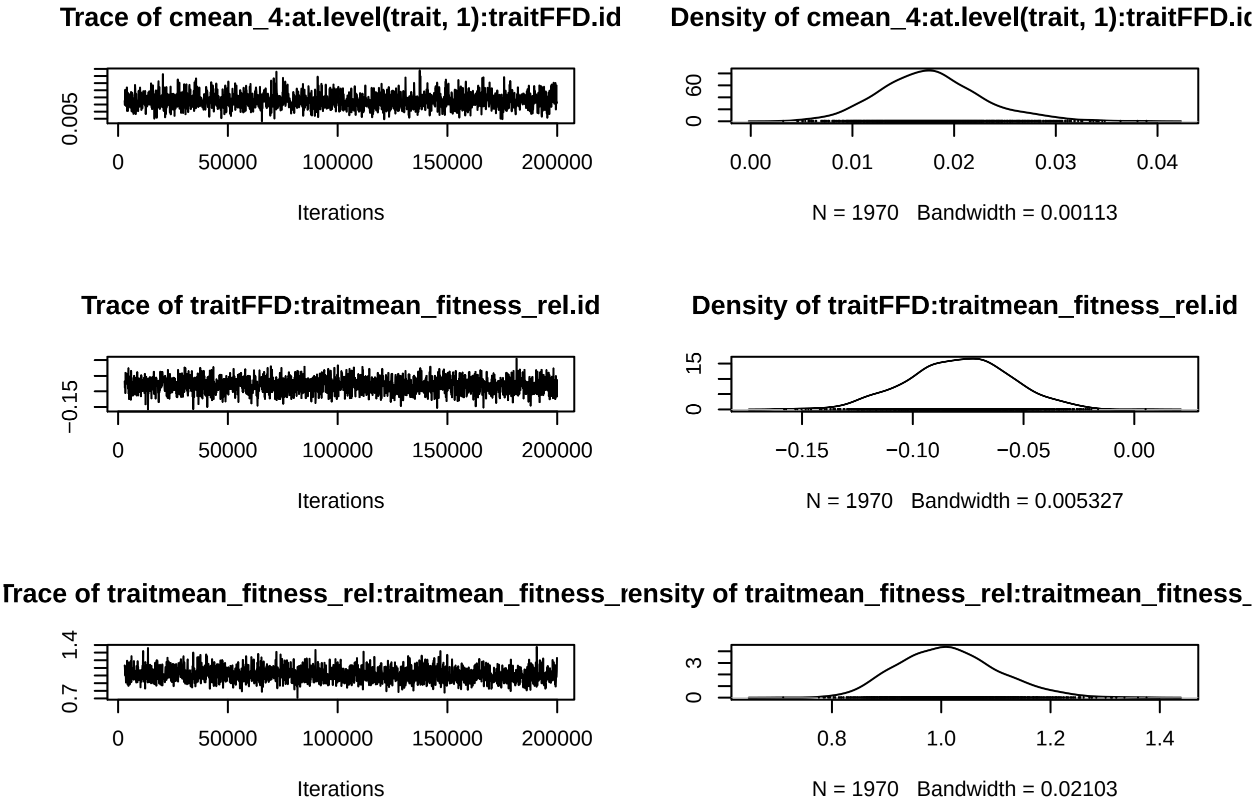
Fit the model:

```r
prior_biv_RR_px <- list(R = list(V = diag(c(1,0.0001),2,2), nu = 0.002, fix = 2),
                        G = list(G1 = list(V = 1, nu = 1,
                                    alpha.mu = 0,
                                    alpha.V= 25^2),
                                 G2 = list(V = matrix(c(1,0,0,
                                                        0,1,0,
                                                        0,0,1),3,3,
                                                      byrow = TRUE),
                                           nu = 3,
                                           alpha.mu = rep(0,3),
                                           alpha.V = diag(25^2,3,3))))

mcmc_biv_RR <- MCMCglmm(cbind(scale(FFD),mean_fitness_rel) ~ trait-1 +
                          at.level(trait,1):cmean_4,
                        random =~ us(at.level(trait,1)):year +
                          us(trait + cmean_4:at.level(trait,1)):id,
                        rcov =~ idh(trait):units,
                        family = c("gaussian","gaussian"),
                        prior = prior_biv_RR_px,
                        #nitt=950000,burnin=50000,thin=450, #provided
                        nitt=200000,burnin=3000,thin=100,     #trial
                        #nitt=10000,burnin=1000,thin=50, # smaller numbers for fast run
                        verbose = FALSE,
                        data = as.data.frame(data_4yrs),
                        pr = TRUE,saveX = TRUE, saveZ = TRUE)
```

Check model diagnostics using plots of the MCMC samples. Look at the trace/density plots of the posterior distributions for the (co)variances:

```
plot(mcmc_biv_RR$VCV)
```





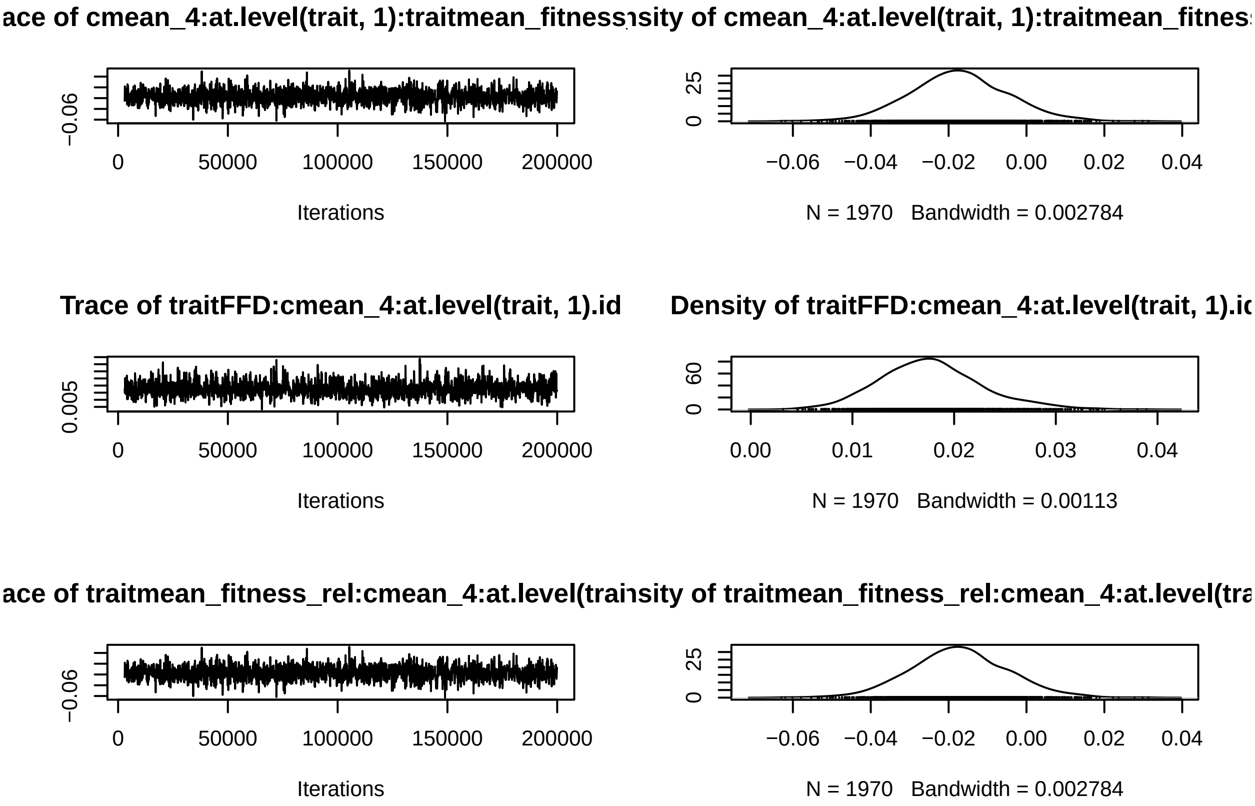
Trace of cmean_4:at.level(trait, 1):traitFFD.id

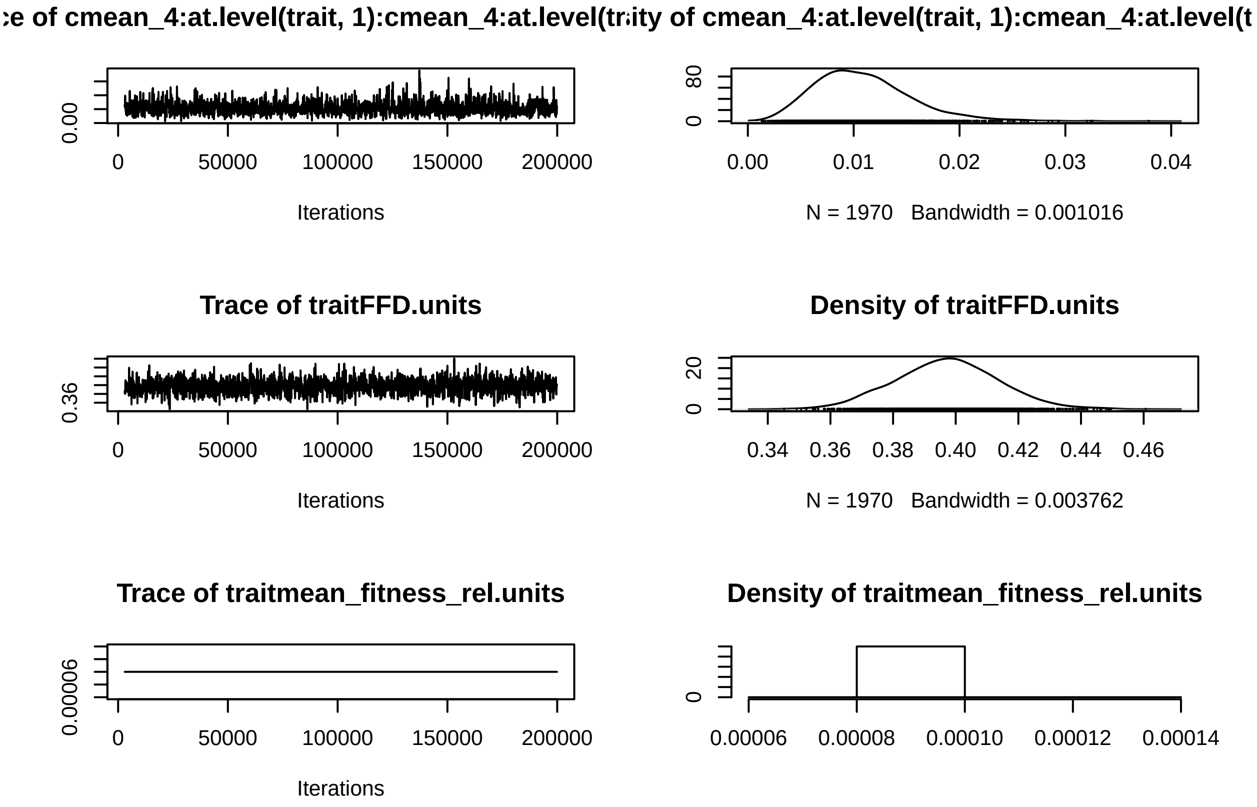
Density of cmean_4:at.level(trait, 1):traitFFD.ic

Trace of traitFFD:traitmean_fitness_rel.id

Density of traitFFD:traitmean_fitness_rel.id

Trace of traitmean_fitness_rel:traitmean_fitness_rensity of traitmean_fitness_rel:traitmean_fitness_


ace of cmean_4:at.level(trait, 1):traitmean_fitness
nsity of cmean_4:at.level(trait, 1):traitmean_fitnes
−0.06
0
50000
100000
150000
200000
Iterations
0
25
−0.06
−0.04
−0.02
0.00
0.02
0.04
N = 1970 Bandwidth = 0.002784
Trace of traitFFD:cmean_4:at.level(trait, 1).id
Density of traitFFD:cmean_4:at.level(trait, 1).ic
0.005
0
50000
100000
150000
200000
Iterations
0
60
0.00
0.01
0.02
0.03
0.04
N = 1970 Bandwidth = 0.00113
ace of traitmean_fitness_rel:cmean_4:at.level(trai
nsity of traitmean_fitness_rel:cmean_4:at.level(tra
−0.06
0
50000
100000
150000
200000
Iterations
0
25
−0.06
−0.04
−0.02
0.00
0.02
0.04
N = 1970 Bandwidth = 0.002784



Check for autocorrelation between successive stored iterations (suggested to be less than 0.1):

```
diag(autocorr(mcmc_biv_RR$VCV)[2, , ])
```

```
##                at.level(trait, 1):at.level(trait, 1).year
##                                                0.05755928
##                                     traitFFD:traitFFD.id
##                                                0.02154807
##                           traitmean_fitness_rel:traitFFD.id
##                                                0.02258196
##                      cmean_4:at.level(trait, 1):traitFFD.id
##                                                0.02863592
##                           traitFFD:traitmean_fitness_rel.id
##                                                0.02258196
##              traitmean_fitness_rel:traitmean_fitness_rel.id
##                                                0.02620744
##        cmean_4:at.level(trait, 1):traitmean_fitness_rel.id
##                                                0.01200387
##                      traitFFD:cmean_4:at.level(trait, 1).id
##                                                0.02863592
##        traitmean_fitness_rel:cmean_4:at.level(trait, 1).id
##                                                0.01200387
## cmean_4:at.level(trait, 1):cmean_4:at.level(trait, 1).id
##                                                0.02491460
##                                           traitFFD.units
##                                               -0.02016795
##                              traitmean_fitness_rel.units
##                                                      NaN
```

Look at the among-individual variance components:

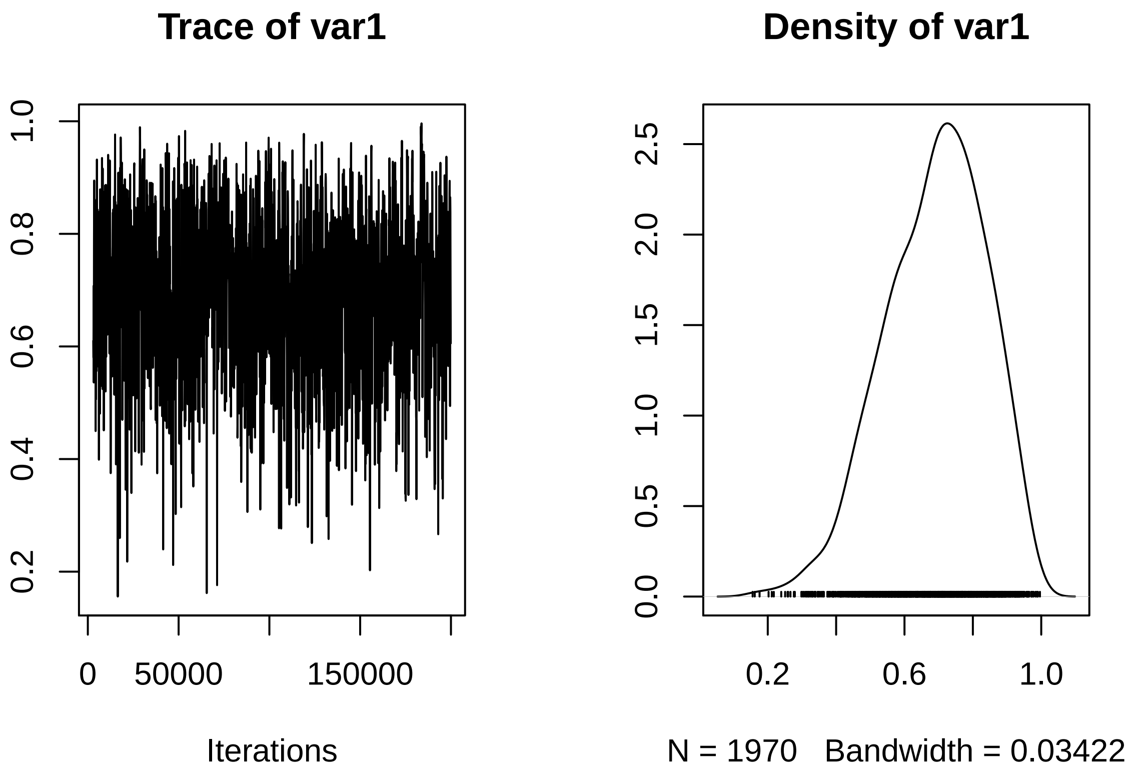
```
summary(mcmc_biv_RR)$Gcovariances
```

```
##                                                            post.mean
## at.level(trait, 1):at.level(trait, 1).year                0.62785595
## traitFFD:traitFFD.id                                      0.06547990
## traitmean_fitness_rel:traitFFD.id                        -0.07912112
## cmean_4:at.level(trait, 1):traitFFD.id                    0.01773217
## traitFFD:traitmean_fitness_rel.id                        -0.07912112
## traitmean_fitness_rel:traitmean_fitness_rel.id            1.01372299
## cmean_4:at.level(trait, 1):traitmean_fitness_rel.id      -0.01728961
## traitFFD:cmean_4:at.level(trait, 1).id                    0.01773217
## traitmean_fitness_rel:cmean_4:at.level(trait, 1).id      -0.01728961
## cmean_4:at.level(trait, 1):cmean_4:at.level(trait, 1).id  0.01087864
##                                                              l-95% CI
## at.level(trait, 1):at.level(trait, 1).year                0.277061327
## traitFFD:traitFFD.id                                      0.041517607
## traitmean_fitness_rel:traitFFD.id                        -0.125940612
## cmean_4:at.level(trait, 1):traitFFD.id                    0.008128723
## traitFFD:traitmean_fitness_rel.id                        -0.125940612
## traitmean_fitness_rel:traitmean_fitness_rel.id            0.852833004
## cmean_4:at.level(trait, 1):traitmean_fitness_rel.id      -0.042782550
## traitFFD:cmean_4:at.level(trait, 1).id                    0.008128723
## traitmean_fitness_rel:cmean_4:at.level(trait, 1).id      -0.042782550
## cmean_4:at.level(trait, 1):cmean_4:at.level(trait, 1).id  0.003209807
##                                                              u-95% CI
## at.level(trait, 1):at.level(trait, 1).year                1.080181692
## traitFFD:traitFFD.id                                      0.092105346
## traitmean_fitness_rel:traitFFD.id                        -0.035966258
## cmean_4:at.level(trait, 1):traitFFD.id                    0.028406326
## traitFFD:traitmean_fitness_rel.id                        -0.035966258
## traitmean_fitness_rel:traitmean_fitness_rel.id            1.204192880
## cmean_4:at.level(trait, 1):traitmean_fitness_rel.id       0.006997828
## traitFFD:cmean_4:at.level(trait, 1).id                    0.028406326
## traitmean_fitness_rel:cmean_4:at.level(trait, 1).id       0.006997828
## cmean_4:at.level(trait, 1):cmean_4:at.level(trait, 1).id  0.020180774
##                                                          eff.samp
## at.level(trait, 1):at.level(trait, 1).year               1389.686
## traitFFD:traitFFD.id                                     1970.000
## traitmean_fitness_rel:traitFFD.id                        1970.000
## cmean_4:at.level(trait, 1):traitFFD.id                   1359.468
## traitFFD:traitmean_fitness_rel.id                        1970.000
## traitmean_fitness_rel:traitmean_fitness_rel.id           1335.398
## cmean_4:at.level(trait, 1):traitmean_fitness_rel.id      1970.000
## traitFFD:cmean_4:at.level(trait, 1).id                   1359.468
## traitmean_fitness_rel:cmean_4:at.level(trait, 1).id      1970.000
## cmean_4:at.level(trait, 1):cmean_4:at.level(trait, 1).id 1970.000
```

traitFFD Variance in intercepts for FFD cmean_4:at.level(trait, 1) Variance in temperature-related slopes for FFD traitmean_fitness_rel Variance in intercepts for relative fitness

Ensure that the among-individual correlation between intercepts and slopes for FFD is (approximately) the same as we estimated in our earlier univariate random regression model. We should find the covariance denoted cmean_4:at.level(trait, 1):traitFFD.id, and then calculate the correlation by standardising this by the product of the square root of both variances:

```
mcmc_cor_RR_intslope <- mcmc_biv_RR$VCV[,"cmean_4:at.level(trait, 1):traitFFD.id"]/
  (sqrt(mcmc_biv_RR$VCV[,"traitFFD:traitFFD.id"])*
     sqrt(mcmc_biv_RR$VCV[,"cmean_4:at.level(trait, 1):cmean_4:at.level(trait, 1).id"]))
plot(mcmc_cor_RR_intslope)
```



```
posterior.mode(mcmc_cor_RR_intslope)
```

```
##      var1
## 0.7111733
```

```
HPDinterval(mcmc_cor_RR_intslope)
```

```
##          lower     upper
## var1 0.4130161 0.9485523
## attr(,"Probability")
## [1] 0.9502538
```

Determining the among-individual correlation between FFD and fitness: The covariance trait-mean_fitness_rel:traitFFD.id is the among-individual covariance between fitness and variation in intercepts for FFD — here we convert it to a correlation, and we find it is negative Note that, unlike variances, covariances (and correlations) can take on positive and negative values, so we can use the 95% CIs to think about 'significance'.

```
mcmc_cor_RR_intfit <- mcmc_biv_RR$VCV[,"traitmean_fitness_rel:traitFFD.id"]/
  (sqrt(mcmc_biv_RR$VCV[,"traitmean_fitness_rel:traitmean_fitness_rel.id"])*
     sqrt(mcmc_biv_RR$VCV[,"traitFFD:traitFFD.id"]))
```

```
posterior.mode(mcmc_cor_RR_intfit)
```

```
##       var1
## -0.2914098
```

```
HPDinterval(mcmc_cor_RR_intfit)
```

```
##           lower      upper
## var1 -0.4660182 -0.1436879
## attr(,"Probability")
## [1] 0.9502538
```

We also find a negative among-individual correlation between fitness and variation in slopes for FFD (converting the covariance, cmean_4:at.level(trait, 1):traitmean_fitness_rel.id, to a correlation). BUT in this case, it is not significant because the CIs encompass zero!!!

```
mcmc_cor_RR_slopefit <-
  mcmc_biv_RR$VCV[,"cmean_4:at.level(trait, 1):traitmean_fitness_rel.id"]/
  (sqrt(mcmc_biv_RR$VCV[,"traitmean_fitness_rel:traitmean_fitness_rel.id"])*
     sqrt(mcmc_biv_RR$VCV[,"cmean_4:at.level(trait, 1):cmean_4:at.level(trait, 1).id"]))
posterior.mode(mcmc_cor_RR_slopefit)
```

```
##       var1
## -0.1696286
```

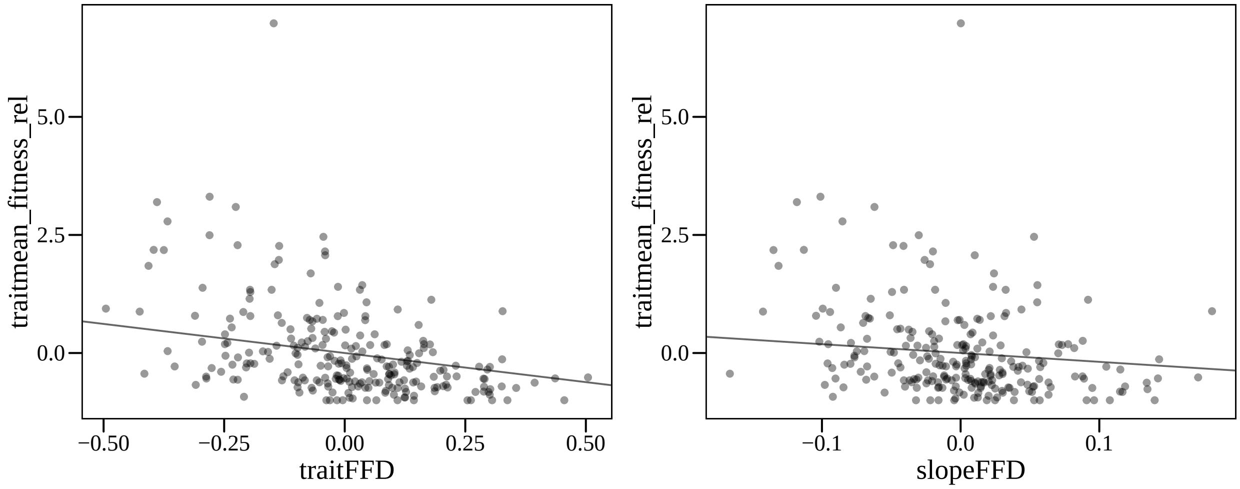
```
HPDinterval(mcmc_cor_RR_slopefit)
```

```
##           lower      upper
## var1 -0.4026574 0.08237151
## attr(,"Probability")
## [1] 0.9502538
```

Again, recall that variation in intercepts is among-individual variance for FFD at x=0 in a random regression model, so the correlation between fitness and intercept is interpretable as the fitness-FFD correlation in this 'environment'. It should not be too surprising that the intercept:fitness and slope:fitness correlations are so similar, because we have seen from our earlier univariate random regression models that at x=0 the intercept:slope correlations are very high. Those with higher intercepts at x=0 also tended to have higher slopes at that point. We can also visualise this result by extracting the BLUPs for each individual and plotting them. This is a reasonable use of BLUP because are not running statistical analyses on them, we are using the BLUPs (ie the model predictions) only as a visual aid for the interpretation of a statistical model we have fitted. Remember that each of these points has a fair amount of uncertainty around it! We also use the (co)variance estimates from our model to calculate the regression slope directly from the bivariate random regression model:

```
# Get coefficients
df_biv_rr_coefs <- data_frame(Trait = attr(colMeans(mcmc_biv_RR$Sol), "names"),
                              Value = colMeans(mcmc_biv_RR$Sol)) %>%
  separate(Trait, c("tmp","Trait","Type","id"), sep = "\\.", fill = "left") %>%
  filter(Type == "id") %>%
  filter(Trait %in% c("traitFFD","traitmean_fitness_rel","level(trait, 1)")) %>%
  mutate(Trait = ifelse(Trait == "level(trait, 1)", "slopeFFD", Trait)) %>%
  select(id, Trait, Value) %>%
  spread(Trait, Value)

# Calculate regression lines from the model fit (co)variances
ai_fit_slope <- mean(mcmc_biv_RR$VCV[,"traitFFD:traitmean_fitness_rel.id"]/
                       mcmc_biv_RR$VCV[,"traitFFD:traitFFD.id"])
```

```
as_fit_slope <- mean(mcmc_biv_RR$VCV[,"cmean_4:at.level(trait, 1):traitmean_fitness_rel.id"]/
mcmc_biv_RR$VCV[,"cmean_4:at.level(trait, 1):cmean_4:at.level(trait, 1).id"])

# Create plots of fitness values against BLUPs of:
# (i) FFD intercepts,
# (ii) FFD slopes
gg_ai_fit <- ggplot(df_biv_rr_coefs,aes(x = traitFFD,y = traitmean_fitness_rel)) +
  geom_abline(intercept = 0, slope = ai_fit_slope,colour = "grey40") +
  geom_point(alpha = 0.4) +  my_theme()
gg_as_fit <- ggplot(df_biv_rr_coefs,aes(x = slopeFFD,y = traitmean_fitness_rel)) +
  geom_abline(intercept = 0,slope = as_fit_slope,colour = "grey40") +
  geom_point(alpha = 0.4) + my_theme()
grid.arrange(gg_ai_fit,gg_as_fit,ncol = 2)
```



Those with higher average FFD (at the intercept) tend to have lower fitness, but also those that have higher slope values have lower fitness (BUT this is supposed to be non-significant according to the among-individual correlation between fitness and variation in slopes for FFD, where the CIs encompassed zero - see above). These are often characterised as (for example) correlations between fitness and average FFD, and fitness and plasticity. However, interpreting (and generalising) this kind of result is actually more difficult than it seems. For instance, could we infer that plasticity is under selection? Maybe, but not necessarily. For example, we know that slopes and intercepts are strongly positively correlated (at x=0), so we would at least need to try and separate out the “direct” effects of each “trait” (i.e. intercept and slope) on fitness. This could be done, following the conceptual approach of classical selection analysis (e.g. Lande and Arnold 1983) by calculating the partial regressions (interpretable as selection gradients) on intercept and slope on fitness.

We also know that significant among-individual variation in slopes really means that the among-individual variation in FFD changes as a function of our x variable (here, temperature): in other words, individual-by-environment interactions are occurring.