

# Posterior comparison of regression coefficients

September 25, 2014

The idea here is pretty simple: The credible intervals associated with each coefficient in the regressions identifies associations for which we have good evidence, i.e., those for which the credible interval doesn't overlap zero. But we can't compare the regression coefficients for *Protea* and *Pelargonium* simply by comparing which coefficients are detectably different from zero, except in the simple case when coefficients don't overlap zero and have opposite signes.

It is, however, easy to construct statistics to compare them using posterior comparisons [1]. We already have a posterior sample for each set of regression coefficients, so we simply select one coefficient from the posterior in the *Protea* model, one from the posterior in the *Pelargonium* model, take the difference, repeat a large number of times, calculate the credible interval of the difference, and determine whether it overlaps zero. If it does, we don't have evidence that the coefficients are different. If it doesn't, we do.

```
load("results-2014-03-06-09-01-02.Rsave")
protea <- fit$BUGSoutput$sims.list
load("results-2014-03-08-13-25-09.Rsave")
pellie <- fit$BUGSoutput$sims.list
rm(fit)
```

`protea` and `pellie` are now simple objects that contain the posterior sample for each coefficient in the model. We're only interested in comparing the `beta.*` coefficients. Since the two chains were run separately from one another and are of the same size we can construct our posterior sample by simply running through each sample from the posterior sequentially. First, we set up the posterior comparison function. That's really easy (so long as

we're using symmetric credible intervals) because all we have to do is determine the lower and upper bounds, take the difference of the input vectors and return the quantiles.

```
compare <- function(x, y, prob = 0.95) {
  lo <- (1 - prob)/2
  hi <- 1 - lo
  diff <- x - y
  quantile(diff, probs = c(lo, hi))
}
```

Now we simply iterate through each of the **beta.\*** and report the results. To make the coefficients that are detectably different from one another a bit easier to spot, we add a '\*' after the credible interval.

```
## get the indices for beta coefficients in the
## objects they'll be the same for protea and
## pellie
idx <- grep("beta", names(pellie))
for (i in idx) {
  for (j in 1:4) {
    cred <- compare(protea[[i]][, j], pellie[[i]][,
      j])
    cat(names(protea)[[i]], "[", j, "]", " (",
      cred[1], ",", cred[2], ")", sep = "")
    ## we know that the credible interval doesn't
    ## overlap zero if either the lower bound is
    ## positive or the upper bound is negative
    if ((cred[1] > 0) || (cred[2] < 0)) {
      cat("*", sep = "")
    }
    cat("\n")
  }
}

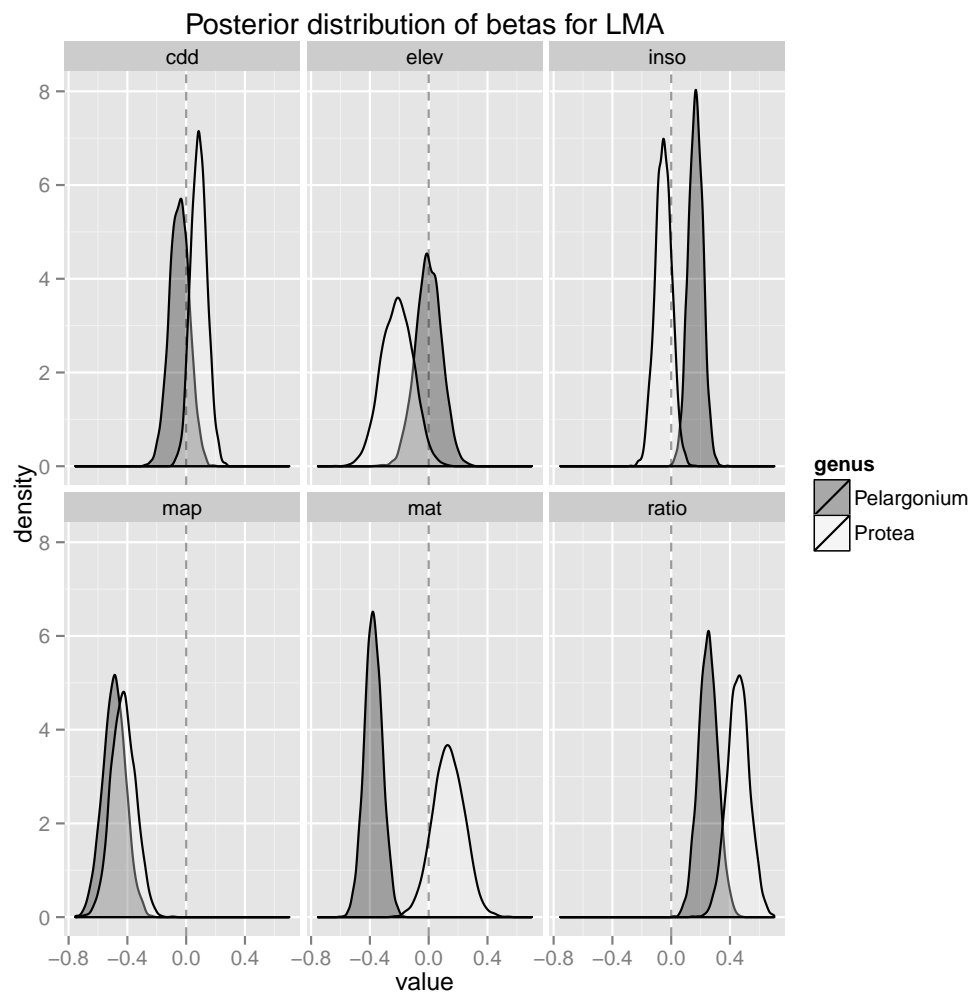
## beta.cdd[1] (-0.03624,0.3088)
## beta.cdd[2] (-0.785,-0.3749)*
## beta.cdd[3] (-0.04717,0.182)
```

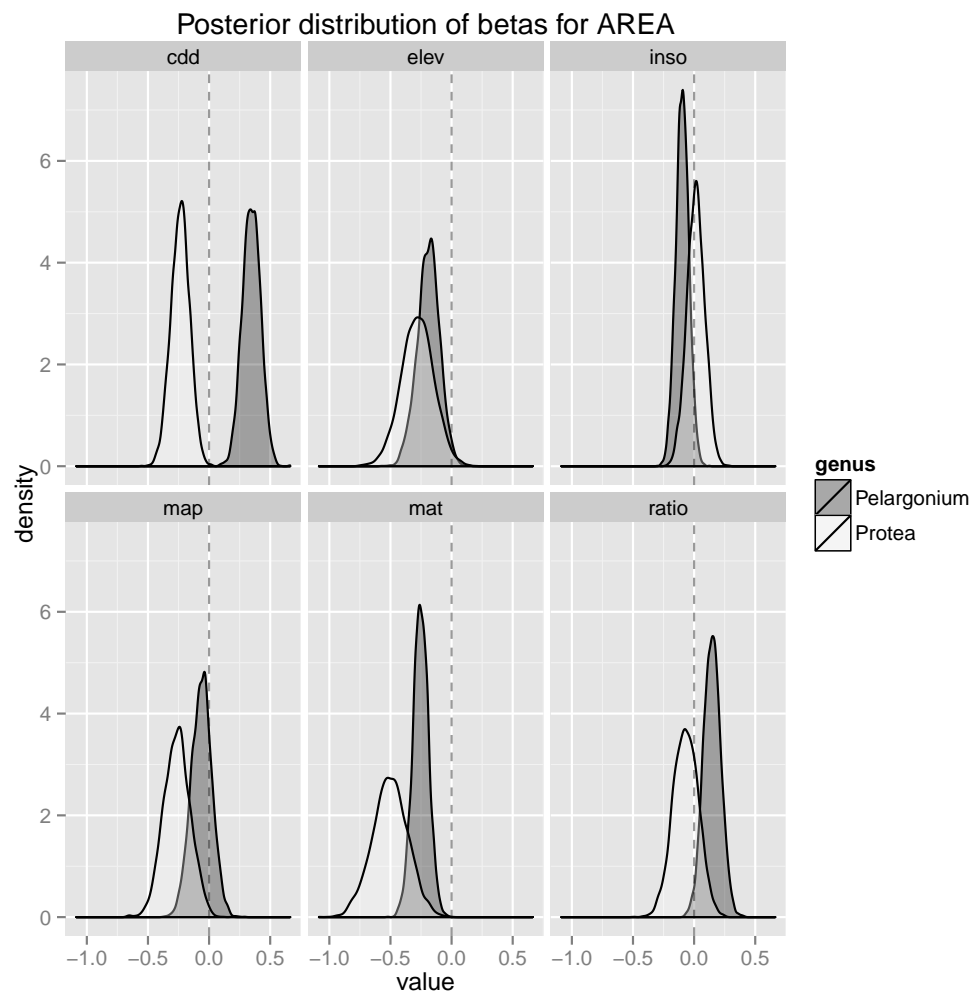
```
## beta.cdd[4] (-0.0724,0.2932)
## beta.elev[1] (-0.4811,0.06748)
## beta.elev[2] (-0.4061,0.2245)
## beta.elev[3] (-0.1862,0.2201)
## beta.elev[4] (-0.1839,0.3939)
## beta.inso[1] (-0.3677,-0.07219)*
## beta.inso[2] (-0.07133,0.2912)
## beta.inso[3] (-0.1027,0.1064)
## beta.inso[4] (-0.02798,0.2842)
## beta.map[1] (-0.1653,0.2921)
## beta.map[2] (-0.4748,0.07021)
## beta.map[3] (0.1035,0.424)*
## beta.map[4] (-0.271,0.1902)
## beta.mat[1] (0.2681,0.7574)*
## beta.mat[2] (-0.5638,0.05674)
## beta.mat[3] (-0.182,0.1915)
## beta.mat[4] (-0.2947,0.2045)
## beta.ratio[1] (0.01782,0.4088)*
## beta.ratio[2] (-0.4621,0.02279)
## beta.ratio[3] (-0.3998,-0.1067)*
## beta.ratio[4] (-0.2784,0.1458)
```

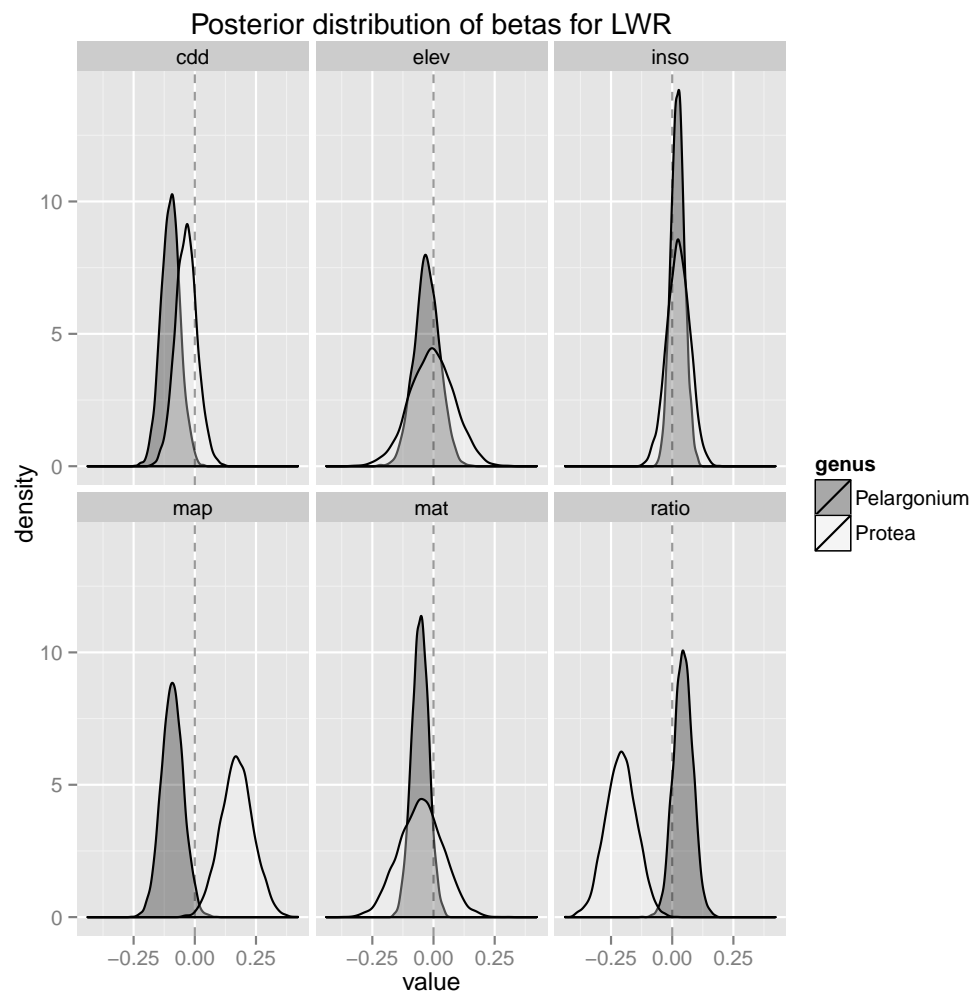
It's probably helpful to visualize the differences by plotting the posteriors.

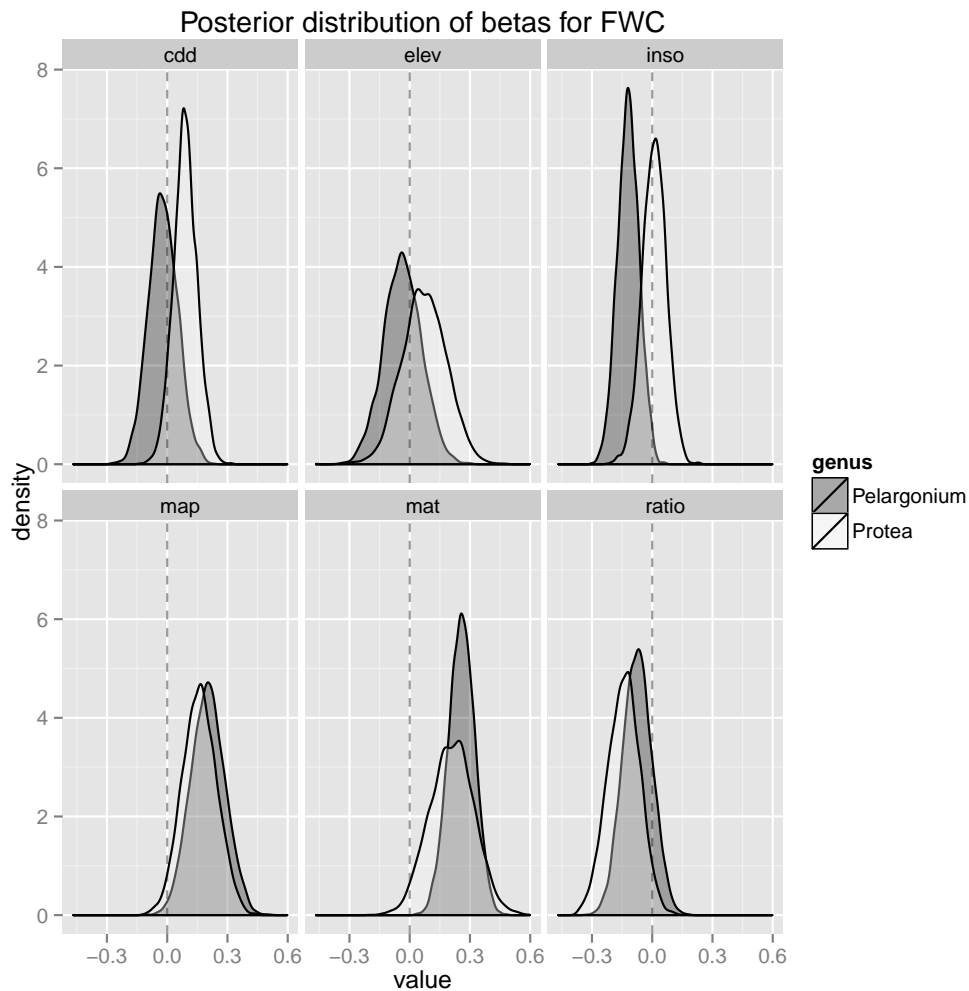
```
source("posterior-comparisons.R")

## Loading required package: ggplot2
```









Here's another approach: Instead of looking at the difference between regression coefficients, we ask "Do these coefficients have the same sign?" There are three possible answers: "(1) Yes. Both are negative. (2) No. One is positive and the other is negative. (3) Yes. Both are positive."

```
compare.sign <- function(x, y) {
  stopifnot(length(x) == length(y))
  negative <- sum(as.numeric((x < 0) & (y <
    0)))
  positive <- sum(as.numeric((x > 0) & (y >
```

```

    0)))
  conflict <- sum(as.numeric((x < 0) & (y >
    0))) + sum(as.numeric((x > 0) & (y < 0)))
  stopifnot(length(x) == (negative + positive +
    conflict))
  negative <- negative/length(x)
  positive <- positive/length(x)
  conflict <- conflict/length(x)
  list(negative = negative, positive = positive,
    conflict = conflict)
}

```

Now we iterate through each of the `beta.*` as before, store the results in a table (`results`), and report the results.

```

## get the indices for beta coefficients in the
## objects they'll be the same for protea and
## pellie
idx <- grep("beta", names(pellie))
coefficient <- character(0)
negative <- numeric(0)
conflict <- numeric(0)
positive <- numeric(0)
for (i in idx) {
  for (j in 1:4) {
    results <- compare.sign(protea[[i]][,
      j], pellie[[i]][, j])
    coefficient <- c(coefficient, paste(names(protea)[[i]],
      j, sep = ""))
    negative <- c(negative, results$negative)
    conflict <- c(conflict, results$conflict)
    positive <- c(positive, results$positive)
  }
}
results <- data.frame(coefficient, negative, conflict,
  positive)

```



```

require(xtable)

## Loading required package: xtable
## Warning: package 'xtable' was built under R version 3.1.1

results.table <- xtable(results, digits = 3)
print(results.table, file = "table-temp.tex",
      include.rownames = FALSE)

```

coefficient	negative	conflict	positive
beta.cdd1	0.043	0.715	0.242
beta.cdd2	0.000	0.999	0.001
beta.cdd3	0.787	0.212	0.001
beta.cdd4	0.040	0.598	0.362
beta.elev1	0.495	0.493	0.012
beta.elev2	0.966	0.034	0.000
beta.elev3	0.376	0.481	0.142
beta.elev4	0.164	0.575	0.261
beta.inso1	0.001	0.825	0.175
beta.inso2	0.406	0.572	0.022
beta.inso3	0.067	0.369	0.565
beta.inso4	0.424	0.570	0.005
beta.map1	1.000	0.000	0.000
beta.map2	0.769	0.230	0.001
beta.map3	0.004	0.973	0.024
beta.map4	0.000	0.035	0.965
beta.mat1	0.102	0.898	0.000
beta.mat2	1.000	0.000	0.000
beta.mat3	0.677	0.306	0.017
beta.mat4	0.000	0.026	0.974
beta.ratio1	0.000	0.000	1.000
beta.ratio2	0.014	0.733	0.253
beta.ratio3	0.131	0.869	0.000
beta.ratio4	0.809	0.187	0.004

## References

- [1] Kent E. Holsinger and Lisa E. Wallace. Bayesian approaches for the analysis of population structure: an example from *Platanthera leucophaea* (orchidaceae). *Molecular Ecology*, 13:887–894, 2004.