phylogenetic examples

Contents

Comparative example from the main text

1

Reanalysis of Balasubramaniam et al data set

5

This document contains the code to replicate the two analyses of macaque hierarchy steepness (one analysis in the main text, one in the appendix).

```
library(EloSteepness.data)
library(rstan)
library(cmdstanr)
library(ape)
```

Comparative example from the main text

```
# path to raw model code
mod1 <- system.file("extdata/phylogenetic_examples/comparative_steepness_model_1.stan",</pre>
                     package = "EloSteepness.data")
file.exists(mod1)
#> [1] TRUE
# load tree, matrices and key table
data("Primates301_nex")
data("phylo_matrices")
data("phylo_key")
# prune tree to available species
species <- unique(phylo_key$species_fixed)</pre>
tree <- keep.tip(Primates301_nex, species)</pre>
# plot(tree)
length(tree$tip.label) == length(species) # sanity check
#> [1] TRUE
# sort matrices and key so they have the same order
interaction matrices <- phylo matrices[sort(names(phylo matrices))]</pre>
key <- phylo_key[order(phylo_key$datasetname), ]</pre>
all(names(interaction_matrices) == key$datasetname) # sanity check
#> [1] TRUE
# summary data for each group
specdata <- data.frame(datasetname = key$datasetname,</pre>
                        spec = gsub(" ", "_", key$species_fixed),
                        groupsize = NA,
                        n_interactions = NA,
                        dyads = NA,
```

```
prunks = NA)

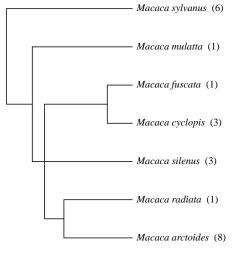
for (m in seq_len(nrow(specdata))) {
  mat <- interaction_matrices[[specdata$datasetname[m]]]
  specdata$groupsize[m] <- ncol(mat)
  specdata$n_interactions[m] <- sum(mat)
  specdata$dyads[m] <- ncol(mat) * (ncol(mat) - 1) / 2
  specdata$prunks[m] <- EloRating::prunk(mat)[1]
}</pre>
```

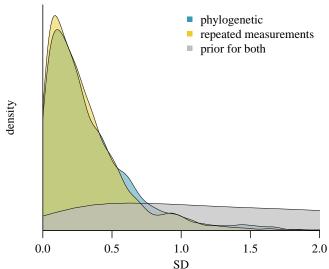
For the sake of this vignette we reduce the number of species and also the number of interactions in the networks. We do this just because it speeds up the sampling from the model. With the full data set, this would take several hours (about six on my fairly old laptop). With the reduced data set this goes faster (but still takes a few minutes).

```
# ignore these lines if you want to run the full data set
specdata <- specdata[specdata$groupsize < 6, ]</pre>
interaction_matrices <- interaction_matrices[specdata$datasetname]</pre>
tree <- keep.tip(tree, specdata$spec)</pre>
interaction_matrices <- lapply(interaction_matrices, function(x) ceiling(x / 5))
# prepare the data set as list for sending it to Stan
xdata <- list()</pre>
xdata$n_datasets <- length(interaction_matrices)</pre>
xdata$n_ids_per_dataset <- as.integer(unlist(lapply(interaction_matrices, ncol)))</pre>
xdata$n_total_ids <- sum(xdata$n_ids_per_dataset)</pre>
xdata$n_interactions_per_dataset <- as.integer(unlist(lapply(interaction_matrices, sum)))</pre>
xdata$n_total_interactions <- sum(xdata$n_interactions_per_dataset)</pre>
xdata$interaction index <- as.integer(seq len(xdata$n total interactions))</pre>
xdata$individual_index <- as.integer(seq_len(xdata$n_total_ids))</pre>
# generate sequences
set.seed(1)
s <- lapply(interaction_matrices, randomized_sequence_from_matrix)</pre>
# global winner and loser indices
xdata$winner <- do.call("rbind", lapply(s, function(x)x$winnermat))</pre>
xdata$loser <- do.call("rbind", lapply(s, function(x)x$losermat))</pre>
xdata$index_dataset_interactions_start <- cumsum(xdata$n_interactions_per_dataset) -</pre>
                                               xdata$n_interactions_per_dataset + 1
xdata$index_individuals_start <- cumsum(xdata$n_ids_per_dataset) -</pre>
                                      xdata$n ids per dataset + 1
xdata$y <- rep(1, xdata$n_total_interactions)</pre>
# phylogeny-related
n_spec <- length(tree$tip.label)</pre>
tree_mat <- vcv.phylo(tree, corr = TRUE)</pre>
species_index <- integer(nrow(specdata))</pre>
for (i in seq_len(nrow(specdata))) {
  species_index[i] <- which(colnames(tree_mat) == specdata$spec[i])</pre>
# for phylogeny part
```

```
xdata$N_phyl <- n_spec # N_1</pre>
xdata$N_coeff_phyl <- 1 # number of parameters: 1 = SD for phyl 'intercepts'
xdata$spec_index <- species_index</pre>
xdata$chol_mat <- t(chol(tree_mat)) # cholesky factor of phylogenetic correlation matrix
xdata$phyl_predictor <- rep(1, nrow(specdata)) # group-level predictor values
# repeated measures
xdata$N_repeated <- n_spec # the same as for the phyl part</pre>
\# number of parameters: 1 = SD for repeated measurements (species intercepts)
xdata$N_coeff_repeated <- 1</pre>
xdata$index_rep_measures <- species_index # the same as for the phyl part
# group-level predictor values, the same as for the phyl part
xdata$rep measures predictor <- rep(1, nrow(specdata))</pre>
# compile stan model
xmod1 <- cmdstan_model(stan_file = mod1, compile = TRUE)</pre>
#> Compiling Stan program...
# sampling
res1 <- xmod1$sample(data = xdata,</pre>
                   refresh = 100,
                   parallel_chains = 4,
                   iter warmup = 1500,
                   iter_sampling = 2500,
                   seed = 1,
                   adapt_delta = 0.9)
stanfit <- rstan::read_stan_csv(res1$output_files())</pre>
xsummary <- summary(stanfit)$summary</pre>
round(summary(stanfit, pars = c("sd_repeated_measures", "sd_phyl"))$summary, 2)
                         mean se_mean sd 2.5% 25% 50% 75% 97.5% n_eff Rhat
#> sd_phyl[1]
                         # optional saving of results
# save(xmod1, xsummary, res1, stanfit, xdata, specdata, tree, file = "comp_analysis_1.RData")
par(mfrow = c(1, 2), family = "serif")
par(family = "serif", mgp = c(1.8, 0.7, 0), las = 1, mar = c(2, 1, 1, 1))
plot.phylo(tree, cex = 0.8, show.tip.label = FALSE, x.lim = c(0, 19))
tips <- tree$tip.label</pre>
for (i in seq_along(tips)) {
 x <- sum(specdata$spec == tips[i])</pre>
 y <- gsub("_", " ", tips[i])
 tiplabels(text = bquote(italic(.(y))~.(paste0(" (", x, ")"))),
           tip = i, bg = NULL, adj = 0, frame = "none", offset = 0.3, cex = 0.8)
}
par(mar = c(3, 2.5, 1.5, 1))
x <- as.numeric(rstan::extract(stanfit, pars = c("sd_phyl"))$sd_phyl)
y <- as.numeric(rstan::extract(stanfit, pars = c("sd_repeated_measures"))$sd_repeated_measures)</pre>
xvals <- brms::rstudent_t(10000, 3, 0, 2.5)</pre>
```

```
while (any(xvals < 0)) {</pre>
  xvals[xvals < 0] <- brms::rstudent_t(sum(xvals < 0), 3, 0, 2.5)</pre>
p3 <- density(xvals, adjust = 1)
p1 <- density(x, adjust = 1)
p2 <- density(y, adjust = 1)</pre>
yr < -c(0, max(p1$y, p2$y) * 1.05)
xr \leftarrow c(0, max(p1$x, p2$x) * 1.05)
xr[2] < -2
plot(0, 0, type = "n", xlim = xr, ylim = yr, xaxs = "i", yaxs = "i", las = 1,
     axes = FALSE, xlab = "SD", ylab = "density")
axis(1)
polygon(c(p1$x, rev(p1$x)), c(rep(0, length(p1$y)), rev(p1$y)),
        border = NA, col = adjustcolor("#3B99B1", 0.5))
polygon(c(p2$x, rev(p2$x)), c(rep(0, length(p2$y)), rev(p2$y)),
        border = NA, col = adjustcolor("#EACB2B", 0.5))
polygon(c(p3$x, rev(p3$x)), c(rep(0, length(p3$y)), rev(p3$y)),
        border = NA, col = adjustcolor("grey", 0.7))
points(p1x, p1y, type = "1", wd = 0.5)
points(p2$x, p2$y, type = "1", lwd = 0.5)
points(p3$x, p3$y, type = "1", lwd = 0.5)
box(bty = "l")
legend("topright", ncol = 1, col = c("#3B99B1", "#EACB2B", "grey"),
       legend = c("phylogenetic", "repeated measurements", "prior for both"),
       pch = 15, bty = "n")
```





Reanalysis of Balasubramaniam et al data set

```
# path to raw model code
mod2 <- system.file("extdata/phylogenetic_examples/comparative_steepness_model_2.stan",</pre>
                     package = "EloSteepness.data")
file.exists(mod2)
#> [1] TRUE
# tree data
data("Primates301_nex")
# data from Balasubramaniam et al 2012 (table 2)
sp <- c("Macaca_assamensis", "Macaca_fascicularis", "Macaca_fascicularis",</pre>
        "Macaca_fuscata", "Macaca_fuscata", "Macaca_mulatta", "Macaca_mulatta",
        "Macaca_nigra", "Macaca_radiata", "Macaca_sylvanus", "Macaca_thibetana",
        "Macaca_thibetana", "Macaca_tonkeana", "Macaca_tonkeana")
st < c(0.65, 0.94, 0.79, 0.56, 0.92, 0.65, 0.78,
        0.49, 0.60, 0.45, 0.87, 0.80, 0.22, 0.20)
specdata <- data.frame(spec = sp,</pre>
                        steepness = st)
# prune tree to 9 macaque species
species <- unique(specdata$spec)</pre>
tree <- keep.tip(Primates301_nex, species)</pre>
# plot(tree)
length(tree$tip.label) == length(species) # sanity check
#> [1] TRUE
# prepare data set to be handed over to stan
xdata <- list()</pre>
xdata$N <- nrow(specdata)</pre>
xdata$the_steepness <- specdata$steepness</pre>
# phylogeny-related
tree_mat <- vcv.phylo(tree, corr = TRUE) # covariance matrix</pre>
n_spec <- length(tree$tip.label)</pre>
# index for species
species index <- integer(nrow(specdata))</pre>
for (i in seq_len(nrow(specdata))) {
  species_index[i] <- which(colnames(tree_mat) == specdata$spec[i])</pre>
}
# for phylogeny part
xdata$N_phyl <- n_spec # number of levels</pre>
xdata$N_coeff_phyl <- 1 # number of pars: SD for phyl 'intercepts'
xdata$spec_index <- species_index</pre>
xdata$chol_mat <- t(chol(tree_mat)) # cholesky factor of phylogenetic correlation matrix
xdata$phyl_predictor <- rep(1, nrow(specdata)) # group-level predictor values
# for repeated measures part
xdata$N_repeated <- n_spec # the same as for the phyl part (number of levels)
# number of pars: SD for repeated measurements (species intercepts)
```

```
xdata$N_coeff_repeated <- 1</pre>
xdata$index_rep_measures <- species_index # the same as for the phyl part
# group-level predictor values, the same as for the phyl part
xdata$rep_measures_predictor <- rep(1, nrow(specdata))</pre>
# compile stan model
xmod2 <- cmdstan model(stan file = mod2, compile = TRUE)</pre>
#> Compiling Stan program...
# sampling
res2 <- xmod2$sample(data = xdata,
                    refresh = 0,
                    parallel_chains = 4,
                    seed = 123,
                    adapt_delta = 0.9)
# convert back to rstan (for easier handling of output)
stanfit <- rstan::read_stan_csv(res2$output_files())</pre>
xsummary <- summary(stanfit)$summary</pre>
round(summary(stanfit,
     pars = c("sd_phyl", "sd_repeated_measures", "phi", "Intercept"))$summary, 2)
#>
                           mean se_mean sd 2.5% 25% 50%
                                                             75% 97.5%
                                                                          n_eff Rhat
#> sd_phyl[1]
                           0.79 0.01 0.50 0.05 0.41 0.74 1.09 1.94 1139.80 1.00
#> sd_repeated_measures[1] 0.68 0.01 0.47 0.03 0.32 0.62 0.95 1.72 1266.04 1.01
                          11.01 0.12 5.65 3.19 6.90 9.91 14.11 24.41 2158.18 1.00
#> phi
#> Intercept
                           0.55 0.01 0.56 -0.59 0.22 0.55 0.88 1.66 2114.03 1.00
# optional saving of results
# save(xmod, xsummary, res, stanfit, xdata, specdata, tree, file = "comp_analysis_2.RData")
# plot in the manuscript
sd_phyl <- sprintf("%.2f", xsummary["sd_phyl[1]", 1])</pre>
sd_rep <- sprintf("%.2f", xsummary["sd_repeated_measures[1]", 1])</pre>
par(mfrow = c(1, 2), family = "serif")
par(family = "serif", mgp = c(1.8, 0.7, 0), las = 1, mar = c(2, 1, 1, 1))
# plot tree with number of groups
ape::plot.phylo(tree, cex = 0.8, show.tip.label = FALSE, x.lim = c(0, 19))
tips <- tree$tip.label</pre>
for (i in seq_along(tips)) {
 x <- sum(specdata$spec == tips[i])</pre>
 v <- gsub("_", " ", tips[i])</pre>
 ape::tiplabels(text = bquote(italic(.(y))~.(paste0(" (", x, ")"))),
                tip = i, bg = NULL, adj = 0,
                frame = "none", offset = 0.3, cex = 0.8)
par(mar = c(3, 2.5, 1.5, 1))
# extract posterior samples for SDs and corresponding priors
est_phyl <- as.numeric(rstan::extract(stanfit, pars = c("sd_phyl"))$sd_phyl)</pre>
est_rep <- as.numeric(rstan::extract(stanfit, pars = c("sd_repeated_measures"))$sd_repeated_measures)</pre>
```

```
pr_phyl <- as.numeric(rstan::extract(stanfit, pars = c("prior_sd_phyl"))$prior_sd_phyl)</pre>
pr_rep <- as.numeric(rstan::extract(stanfit, pars = c("prior_sd_repeated_measures"))$prior_sd_repeated_n</pre>
# and densities of posteriors
est_phyl <- density(est_phyl, adjust = 1)</pre>
est_rep <- density(est_rep, adjust = 1)</pre>
# and for priors
pr_phyl <- density(pr_phyl[pr_phyl < 10], adjust = 1) # truncated for smoother display</pre>
pr_rep <- density(pr_rep[pr_rep < 10], adjust = 1)</pre>
# data ranges for plot
yr \leftarrow c(0, max(est_phyl$y, est_rep$y) * 1.05)
xr \leftarrow c(0, max(est_phyl$x, est_rep$x) * 1.05)
plot(0, 0, type = "n", xlim = xr, ylim = yr, xaxs = "i", yaxs = "i",
     las = 1, axes = FALSE, xlab = "SD", ylab = "density")
axis(1)
polygon(c(est_phyl$x, rev(est_phyl$x)), c(rep(0, length(est_phyl$y)), rev(est_phyl$y)),
        border = NA, col = adjustcolor("#3B99B1", 0.5))
polygon(c(est_rep$x, rev(est_rep$x)), c(rep(0, length(est_rep$y)), rev(est_rep$y)),
        border = NA, col = adjustcolor("#EACB2B", 0.5))
polygon(c(pr_phyl$x, rev(pr_phyl$x)), c(rep(0, length(pr_phyl$y)), rev(pr_phyl$y)),
        border = NA, col = adjustcolor("grey", 0.4))
polygon(c(pr_rep$x, rev(pr_rep$x)), c(rep(0, length(pr_rep$y)), rev(pr_rep$y)),
        border = NA, col = adjustcolor("grey", 0.4))
points(est_phyl$x, est_phyl$y, type = "l", lwd = 0.5)
points(est_rep$x, est_rep$y, type = "1", lwd = 0.5)
points(pr_phyl$x, pr_phyl$y, type = "1", lwd = 0.5)
points(pr_rep$x, pr_rep$y, type = "1", lwd = 0.5)
box(bty = "1")
legend("topright", ncol = 1, col = c("#3B99B1", "#EACB2B", "grey"),
       legend = c("phylogenetic", "repeated measurements", "priors"),
       pch = 15, bty = "n")
```

