jsdgam :: CHEATSHEET

A joint species distribution model (JSDM) describes many species at many sites at once. Each species responds to the measured environment. A few latent factors capture the co-occurrence the environment leaves unexplained. `jsdgam()` fits JSDMs with `Stan`, taking the smooths, random effects and families of any `mvgam` model.

Modelling with `jsdgam()`

Usage: `jsdgam(formula, factor_formula, data, family, n_lv, ...)`

`formula`: the observation model. `y ~ env * series` gives each species its own response to `env`

`factor_formula`: covariates of the latent factors. The default `~ -1` fits unconstrained factors. A term with `lv_axis()`, as in `~ -1 + env:lv_axis()`, gives each factor its own effect: a constrained ordination

`data`: a long `data.frame`, one row per site and species. `unit` names the site column and `species` the species factor

`n_lv`: the number of latent factors, 2 by default

`family`: one family shared by every species, `binomial()` by default. Options include (among others):

- Presence-absence: `binomial()`, `bernoulli()`
- Counts: `poisson()`, `negbinomial()`, `beta_nb()`
- Imperfect detection: `occ()`, `nmix()` (N-mixture article)
- Compositions: `diri()`, `multi()`, `categ()`
- Continuous: `mvn()`, `mvt()`

One `bf()` per species gives each species its own family

Structured priors on the loadings. Options include:

- `traits`: a species-by-trait table. Species with similar traits load alike
- `phylo`: an `ape::phylo` tree or a distance matrix. Close relatives load alike
- `loadings_prior = 'mgp'`: shrinks later factors to zero, so `n_lv` becomes a ceiling

`trait_slopes = ~ body_size` makes species responses to the environment depend on their traits, the fourth-corner model. Every term of `formula` takes part, smooths and factors among them. The traits are columns of `data`

Prior to modelling, it is useful to:

- Convert a site-by-species matrix with `pivot_species_matrix()`, or detection histories from `unmarked` or `spOccupancy` with `pivot_detection_array()`
- Check the data with `mvgam_data()`
- Inspect default priors with `get_prior()` on a model built with `run_model = FALSE`

`sim_jsdgam()` simulates data to practise on:

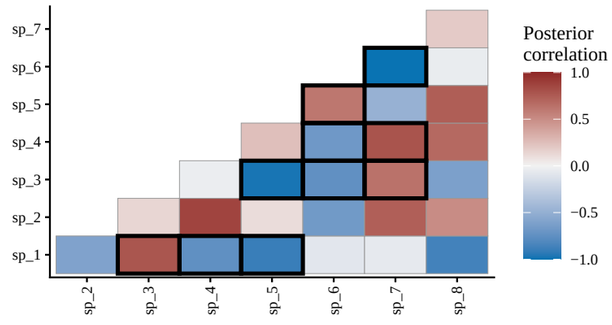
```
simdat <- sim_jsdgam(n_species = 8, n_sites = 60)
traits <- data.frame(series = paste0("sp_", 1:8),
                     body_size = rnorm(8))
mod <- jsdgam(y ~ env * series, n_lv = 2,
              data = simdat$data_train,
              family = negbinomial(), traits = traits)
```

Associations and ordination

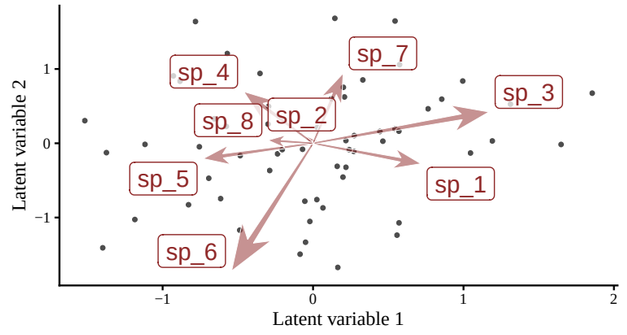
Which species co-occur?

`summary(mod)`: posterior summaries, including the loadings and the length scales of a trait prior. Aim for every Rhat below 1.01

`residual_cor(mod)`: correlations among species once the environment is accounted for. `plot()` draws the heatmap. `summary()` lists the pairs



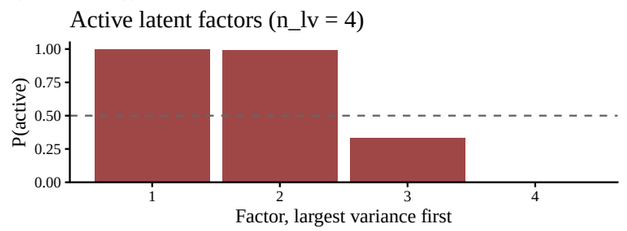
`ordinate(mod, rotation = 'varimax')`: a biplot of site scores and species loadings. A varimax rotation loads each species on as few factors as it can



`shared_variation(mod)`: the covariance the factors imply, with the variance each species shares on the diagonal. `compare_loadings(mod1, mod2)` aligns the loadings of two fits

How many factors?

`active_factors(mod)` gives the posterior probability that each factor is active under `loadings_prior = 'mgp'`, here fitted with `n_lv = 4`

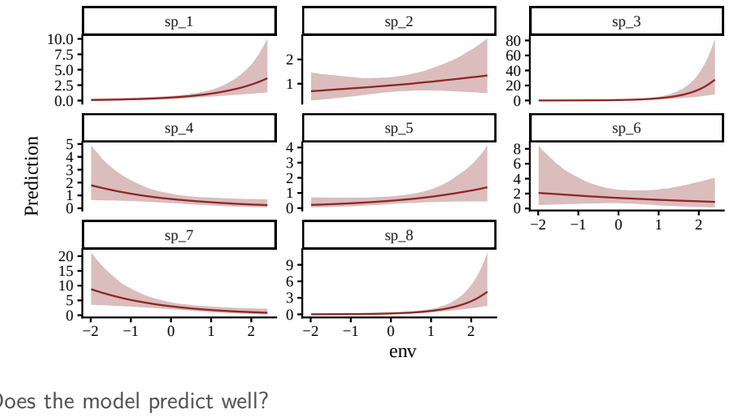


Effects and prediction

How does each species respond?

`species_effects(mod, variables = 'env')`: each species' slope on `env`. Its `plot()` outlines a slope whose sign has posterior probability above 0.95. A `trait_slopes` fit adds the trait effects

`conditional_effects(mod, effects = 'env:series')` plots the expected count of each species along `env`



Does the model predict well?

`pp_check(mod)`: simulated against observed responses, one panel per species with `type = 'dens_overlay_grouped'` and `group = 'series'`

`kfold(mod, group = 'site')` holds out whole sites in turn and scores their expected log predictive density (higher is better). Each site's factor scores are estimated from a handful of observations. `loo(mod)` then often warns of high Pareto *k*

A constrained ordination moves each factor with `env`. Each species then responds through its loadings. Compare it with free per-species slopes:

```
mod_con <- update(mod, formula. = y ~ series,
                  factor_formula = ~ -1 + env:lv_axis())
```

```
print(loo_compare(mod, mod_con), p_worse = FALSE)
```

```
##    model elpd_diff se_diff
##      mod       0.0     0.0
##  mod_con     -18.1     4.9
```

`posterior_epred(mod, newdata = new_sites)` predicts every species at unsurveyed sites from their covariates

How do I report the model?

`how_to_cite(mod)` drafts a methods paragraph with its references. `methods_md(mod)` writes the model's equations and priors

The JSDM article fits ant counts with traits, constrained ordination and factor shrinkage. The mvgam cheatsheet covers time-series models

CC BY-SA 4.0 ○ Nicholas J. Clark ○ Learn more at the JSDM article ○ package version 2.0.0 ○ updated: 2026-11-26