Accommodating spatial or temporal autocorrelation

Pekka Korhonen

Residual covariance matrix

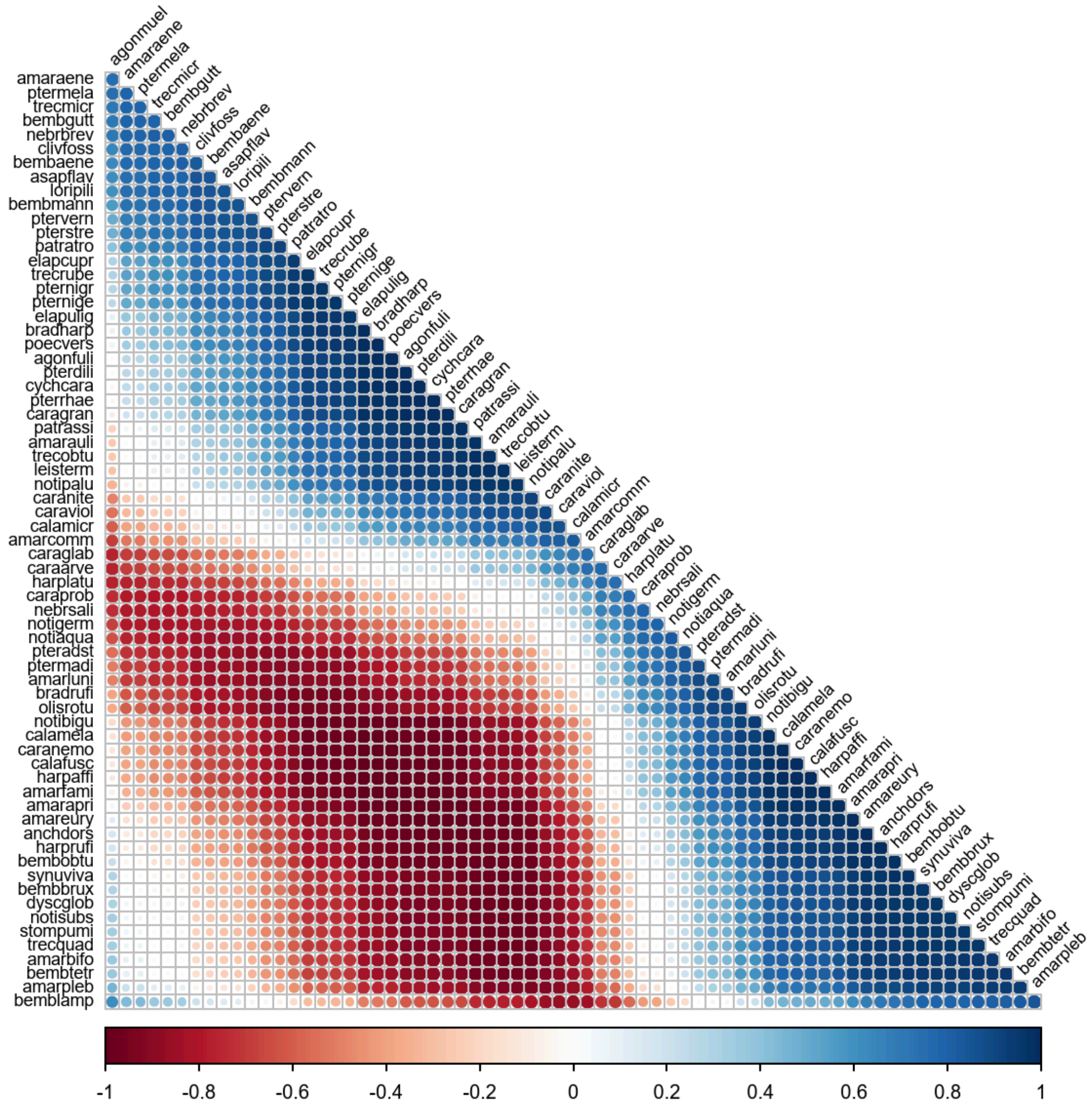
A really important part of working with JSDMs, including GLMMs and GLLVMs alike, is the residual covariance matrix

$$\Sigma \approx \Gamma\Gamma^{\top},$$

which can be used to examine correlations, i.e., strength and direction of co-occurence patterns among the species studied.

As a reminder, in `gllvm`:

```r
library(corrplot)
Sigma <- getResidualCor(fit)
corrplot(Sigma, type="lower", diag=FALSE, order="AOE", tl.srt=45)
```



Output of `corrplot()`

Other types of correlation

What if, instead of remaining static these co-occurence patterns are in fact:

- subject to changes in time, e.g., seasons or some long-term trend
- stronger/weaker/different altogether depending on the geographical region
- affected by some nested/hierarchical aspect of the study design, e.g., transects belonging to the same observational site

Spatial/temporal structures in JSDMs

- let us take into account e.g., yearly sampling variation across all species abundances
- allow us to examine patterns in species composition in space and/or time, instead of a fixed setting
- allow us to study the underlying ecological processes in a more complete sense
- offer us the capacities to make precitions and infer long-term trends for how communities evolve in time and space

Where do these effects fit in the GLLVM framework?

Currently in `gllvm`, two main options exist:

- As row/community-level effect(s), e.g., $\alpha_i \implies \alpha_{\text{year}(i)}$
  - This serves for explaining sampling variations in the overall species abundances
- As correlated latent variables, e.g., $u_i \implies u_{\text{loc}(i)}$
  - Allows species' correlations to vary spatially or in time
  - Env. gradients similar for nearby observations

Reminder: baseline GLLVM

Let the response $y_{ij}, i = 1, \ldots, n,\ j = 1, \ldots, m$ be distributed according to some distribution $F(\mu_{ij}, \phi_j)$, with mean μ_{ij} (and dispersion ϕ_j). Then, let

$$g(\mu_{ij}) = \eta_{ij} = \alpha_i + \beta_{0j} + x_i^\top \beta_j + u_i^\top \gamma_j,$$

where

- $g(\cdot)$ is a link function
- α_i are the row effects
- u_i, γ_j are the LV scores and loadings, respectively

Changing assumptions

Under the basic model, we assume that the row effects and LV scores are independent draws from standard Gaussians, i.e., $\alpha \sim \mathcal{N}(0, \sigma^2 I_n)$ and $u_i \overset{\text{iid}}{\sim} \mathcal{N}(0, I_d)$. For correlated effects, instead:

- $\alpha \sim \mathcal{N}(0, \Sigma_\alpha)$
- $u_{.q} \sim \mathcal{N}(0, \Sigma_q)$, for $q = 1, \ldots, d$.

Here, $u_{.q}$ denotes a vector of all scores for the qth LV, indicative of the need to change the perspective from rows to columns.

Structures for Σ_α and Σ_q

Depending on the application, a plethora of valid covariance structures exist. In `gllvm`, arguments `row.eff` and `lvCor` have the following options:

- `corAR1` for AR(1)
- `corExp` for the exponential correlation function
- `corMatern` for the Matérn correlation function
- `corSym` for compound symmetry

Additionally, the option `corWithin` can be used to specify whether correlations are assumed between, or within groups.

Compound symmetry

The compound symmetry assumes equal variances among the variables, as well as equal covariances between any two variables, i.e.,

$$\Sigma = \begin{bmatrix} \sigma^2 & \sigma_{cs} & \sigma_{cs} & \cdots & \sigma_{cs} \\ \sigma_{cs} & \sigma^2 & \sigma_{cs} & \cdots & \sigma_{cs} \\ \sigma_{cs} & \sigma_{cs} & \sigma^2 & \cdots & \sigma_{cs} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sigma_{cs} & \sigma_{cs} & \sigma_{cs} & \cdots & \sigma^2 \end{bmatrix}$$

Perhaps useful within some nested sampling designs

Order one autoregressive

Autoregressive means dependence on previous values; in case of AR(1) specifically, dependence on the one immediate previous value, indicating the following form:

$$\Sigma = \sigma^2 \begin{bmatrix} 1 & \rho & \rho^2 & \cdots & \rho^{T-1} \\ \rho & 1 & \rho & \cdots & \rho^{T-2} \\ \rho^2 & \rho & 1 & \cdots & \rho^{T-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho^{T-1} & \rho^{T-2} & \rho^{T-3} & \cdots & 1 \end{bmatrix}, \rho \in (-1, 1)$$

Fitting a GLLVM with AR(1) LVs

We will use corAR1 to accommodate temporal dependencies in a subset of the kelpforest data of Reed and Miller (2023):

```r
fit_AR1lv2 <- gllvm(y = Y01, Xenv, family="binomial", num.lv=2,
                    formula = ~logKELP_FRONDSsc + PERCENT_ROCKYsc,
                    studyDesign = Xenv[,c("SITE", "TRANSECT", "YEAR")],
                    row.eff = ~ (1|SITE/TRANSECT), lvCor = ~corAR1(1|YEAR))
# afterwards, check the estimates for the parameters of the AR(1)-process
fit_AR1lv2$params$rho.lv
```

```
  rho.lv1   rho.lv2
0.9133615 0.9350670
```

The above value indicate a very strong correlation in the responses between consequent years.

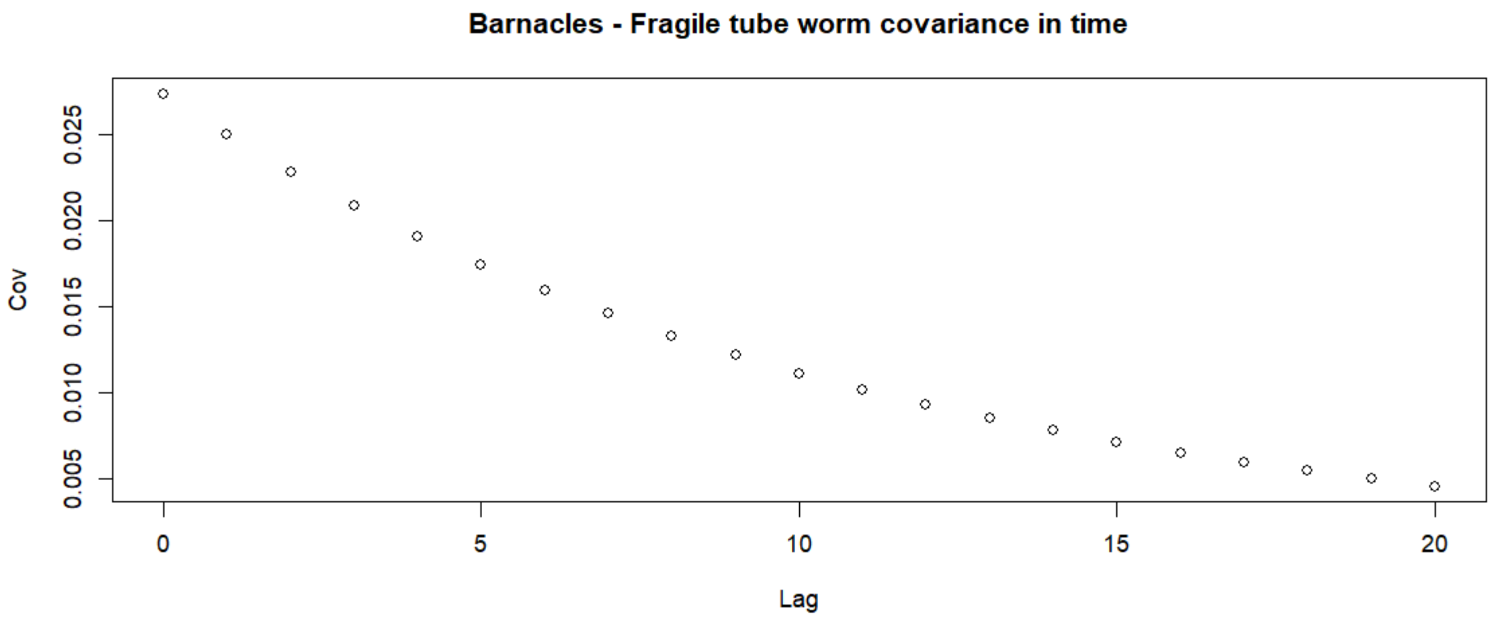
AR(1) residual covariance

Classically, the residual covariance between species i and j is:

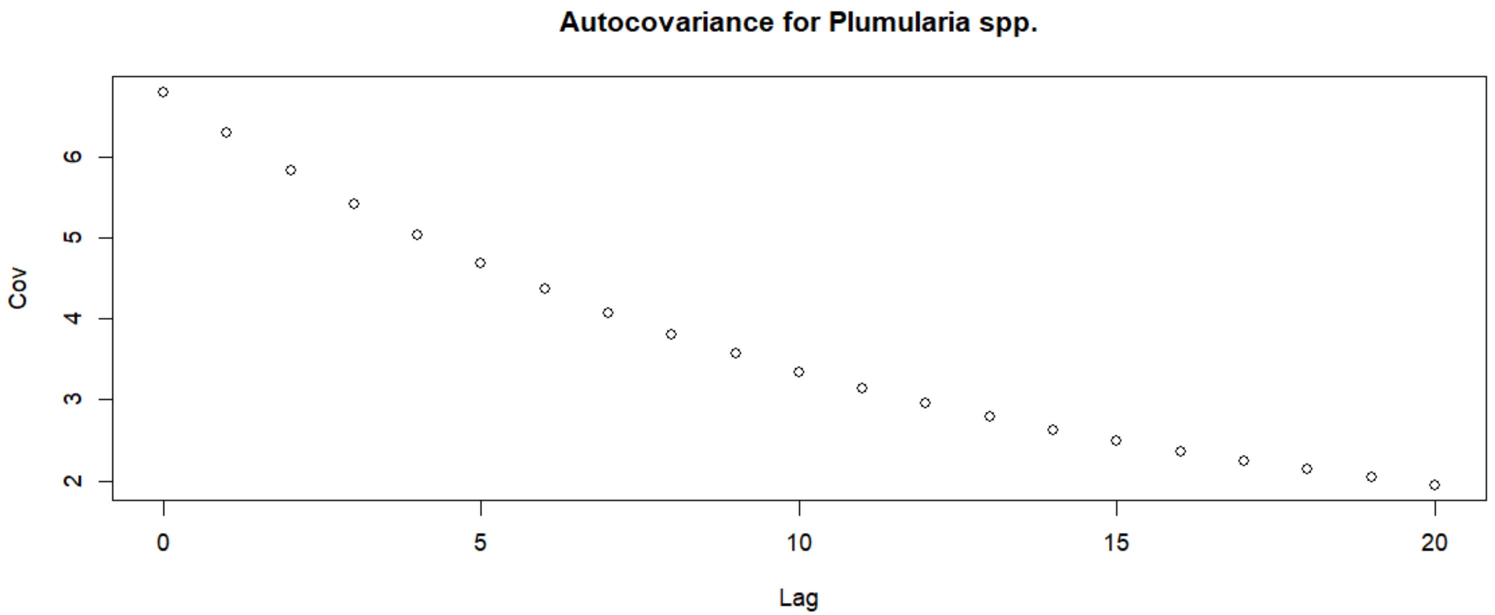
$$\text{Cov}[i, j] = {\gamma_i}^\top \gamma_j = \sum_{q=1}^{d} \gamma_{iq}\gamma_{jq}$$

Now, the AR(1) LVs allow us to consider both between and within species covariances at different timepoints/lags

$$\text{Cov}[i(t), j(t')] = \sum_{q=1}^{d} \gamma_{iq}\gamma_{jq}\text{Cov}[u_{.q}(t), u_{.q}(t')]$$

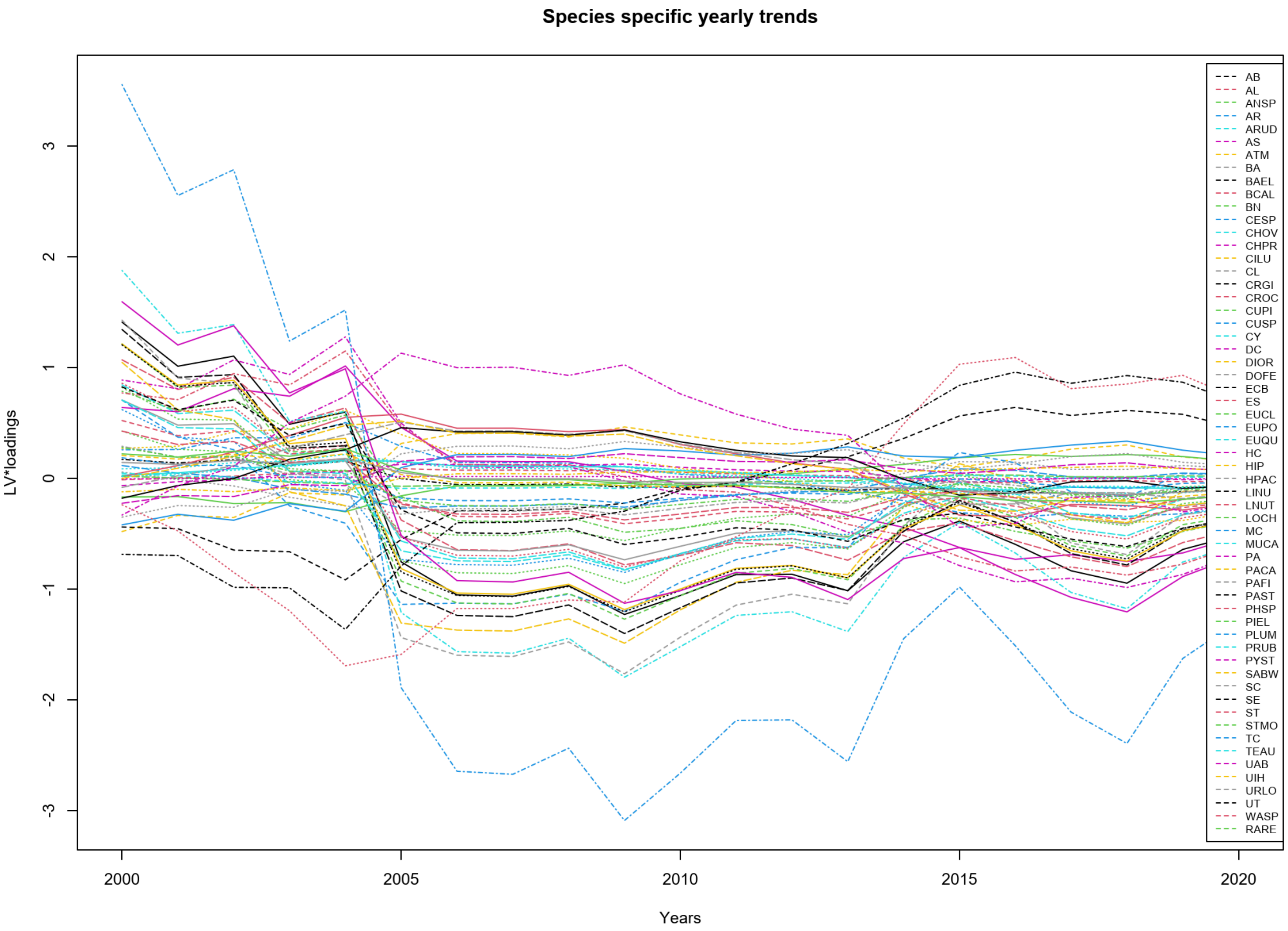


Barnacles - Fragile tube worm covariance in time
Cov
0.005
0.010
0.015
0.020
0.025
Lag
0
5
10
15
20



Autocovariance for Plumularia spp.
Cov
2
3
4
5
6
Lag
0
5
10
15
20

Yearly trends in $\gamma_j^\top u_{\mathrm{year}(i)}$?



Correlation functions

Often it makes sense to define the variance-covariance matrix via some parameterized covariance function or *kernel*:

$$\Sigma[s, s'] = K(s, s'),$$

Two conditions that need to be met in order to ensure that the resulting Σ is a proper covariance matrix:

- $K(s, s') = K(s', s)$ for any pair s, s'.
- $\sum_{i=1}^{S} \sum_{j=1}^{S} c_i c_j K(i, j) \geq 0$ for all $c_1, \ldots, c_S \in \mathbb{R}$.

Exponential covariance kernel

When, instead of the locations, the value of $K(s, s')$ depends only on distance $d = \operatorname{dist}(s, s')$, it is called *stationary*.

A prime example of a stationary covariance function is given by the exponential covariance function:

$$K(d) = \sigma^2 \exp\left(-\frac{d}{\rho}\right),$$

where $\rho > 0$ is a parameter controlling the spatial range, i.e., the rate at which the covariances diminish as distance d increases.

Fitting a model with spatial corExp

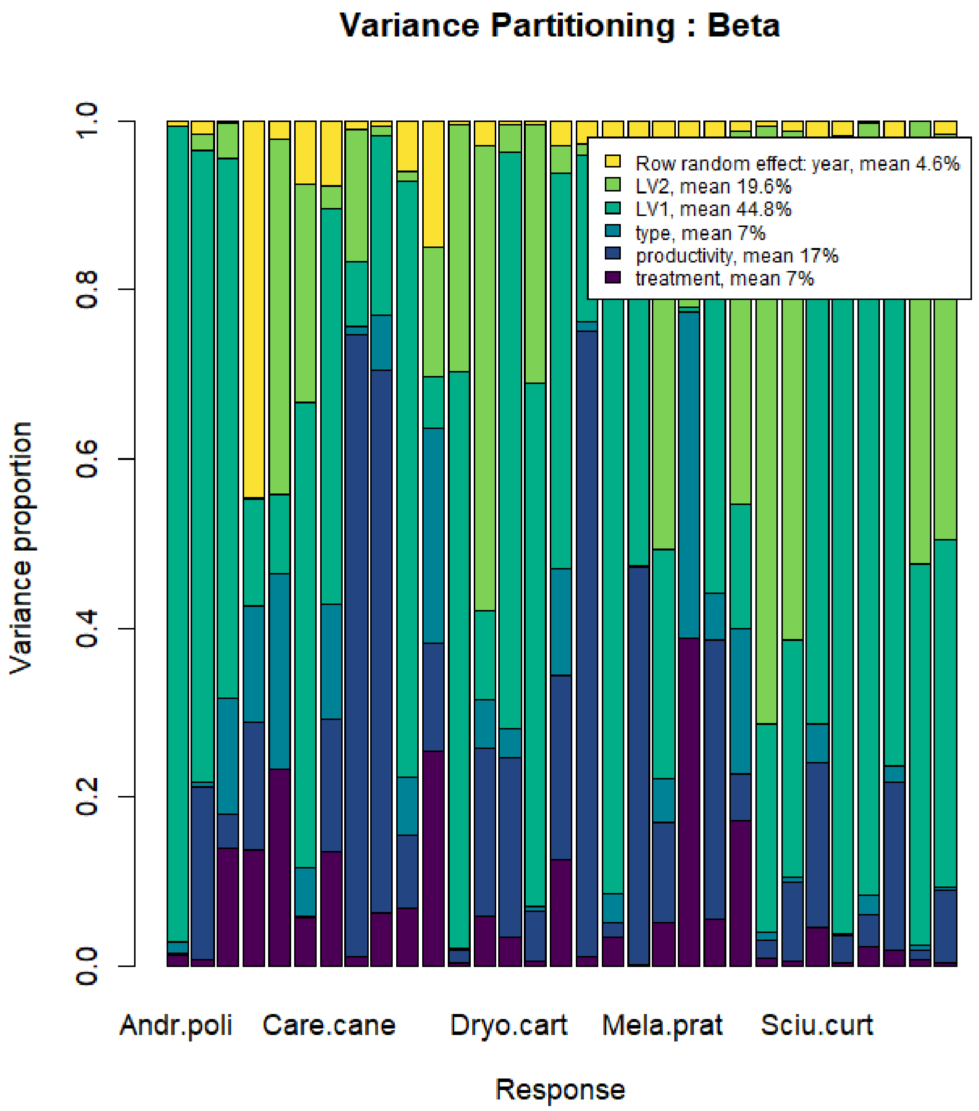
Application goes similar to the case of corAR1, but now we also have to supply the coordinates using distLV:

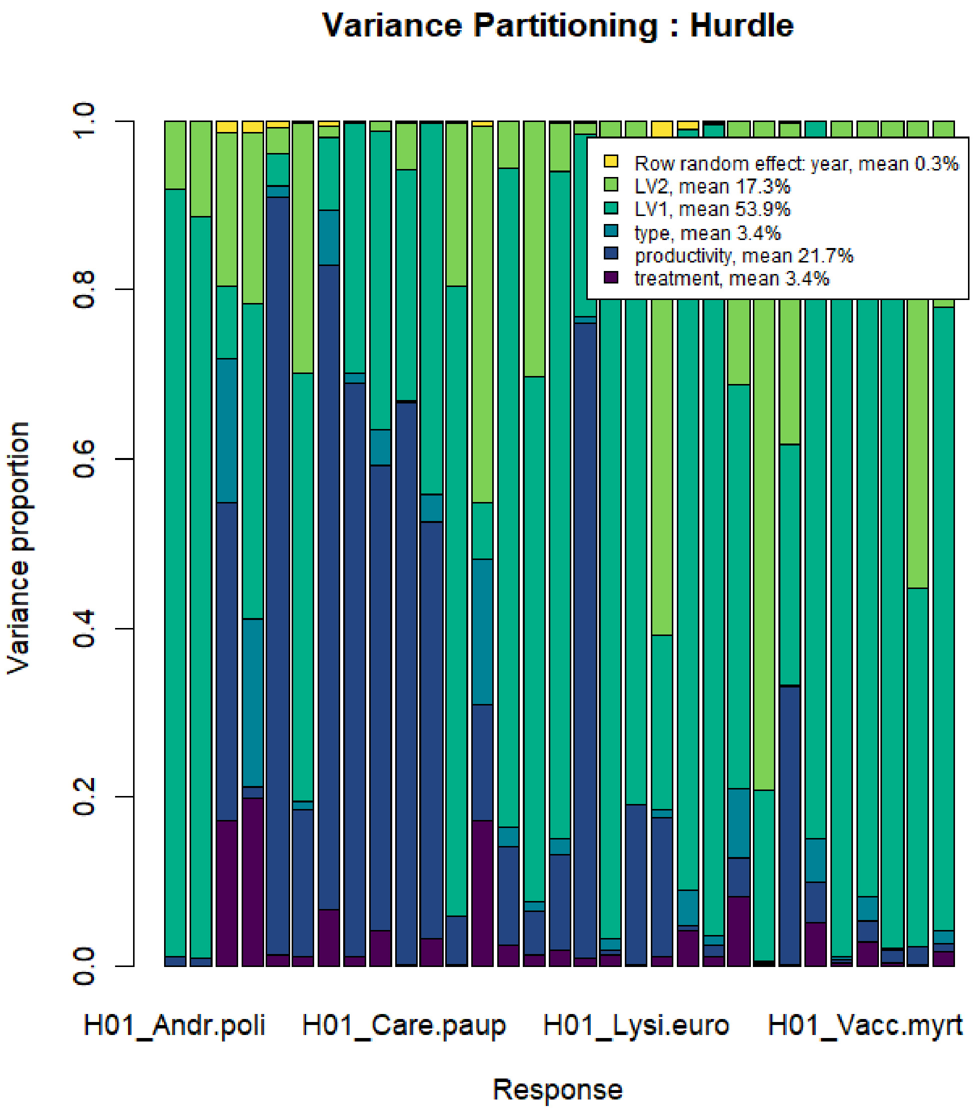
```r
fit_cExpLV2 <- gllvm(Yvasc, Xenv, formula=~ treatment + productivity + type,
                     num.lv = 2, studyDesign = Xenv[,c("site", "year")],
                     lvCor=~corExp(1|site), family="betaH", distLV = sitexy,
                     Lambda.struc="UNN", NN=10, row.eff =~corAR1(1|year))
```

For a model with lots of effects on different levels, it may be useful to compute the variance partitioning, to see the explanatory power of each component:

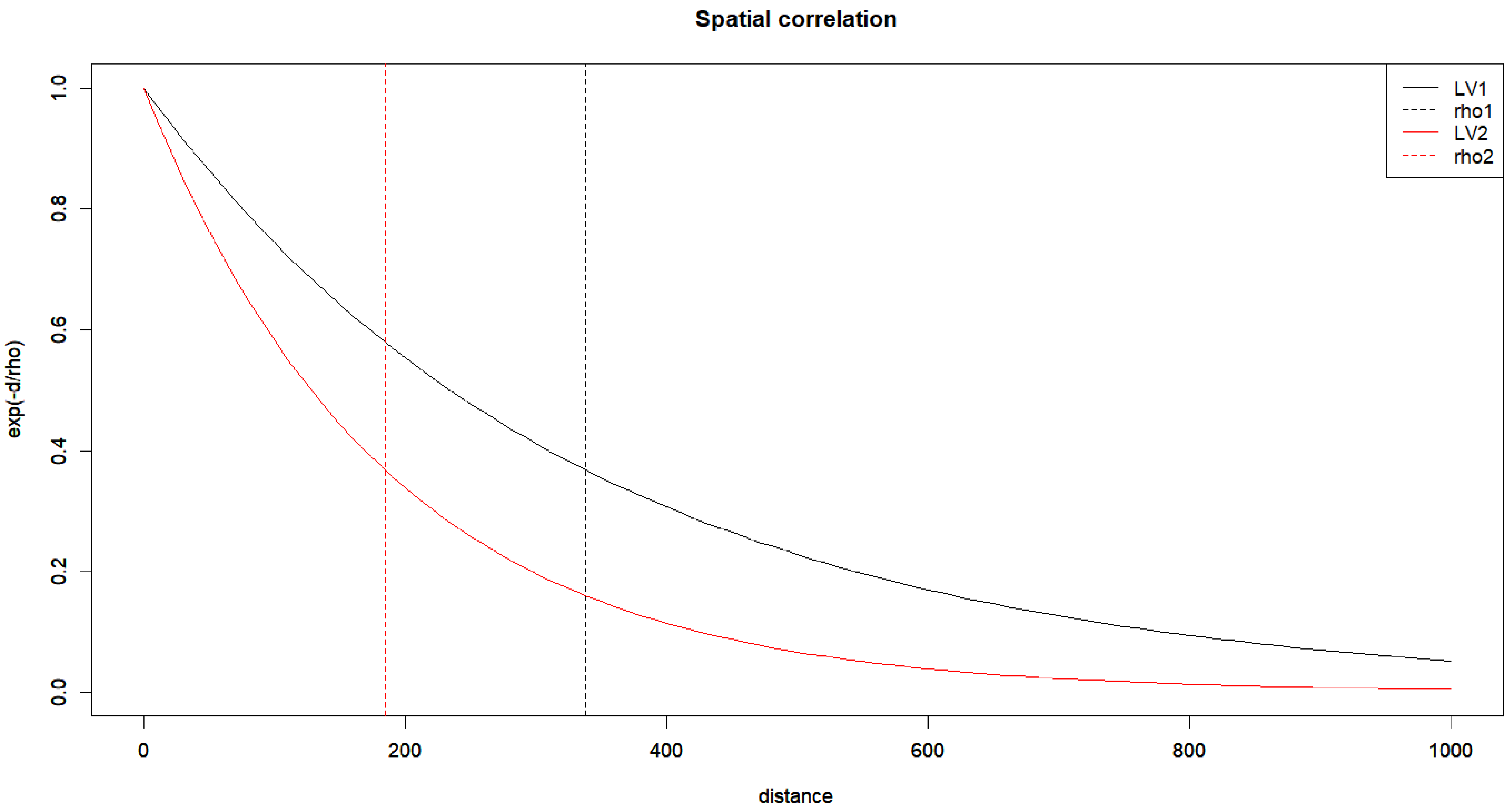
```r
VP <- varPartitioning(fit_cExpLV2)
plotVP(VP, args.legend=list(cex=0.7), col=hcl.colors(6, "viridis"))
```

Variance partitioning

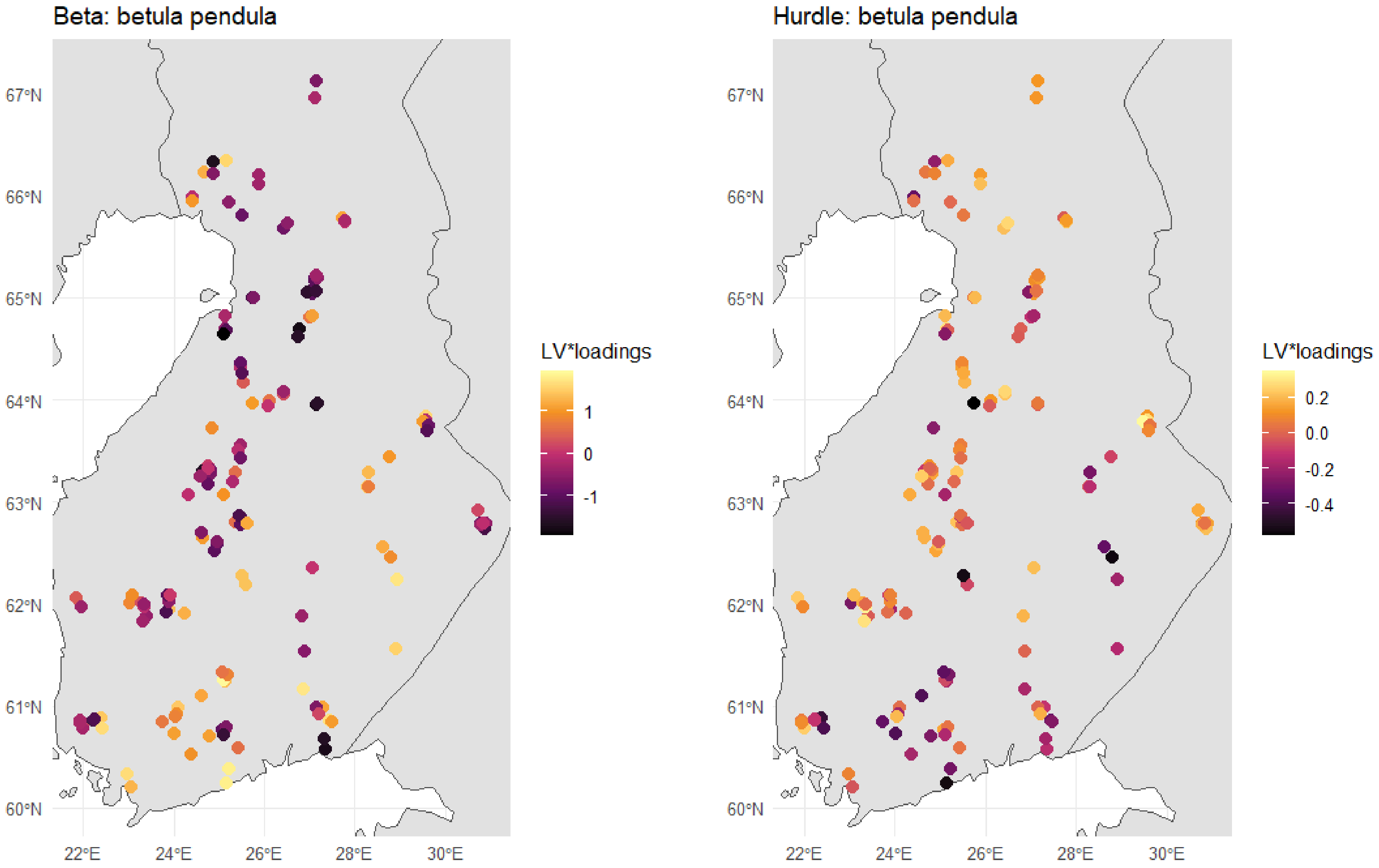




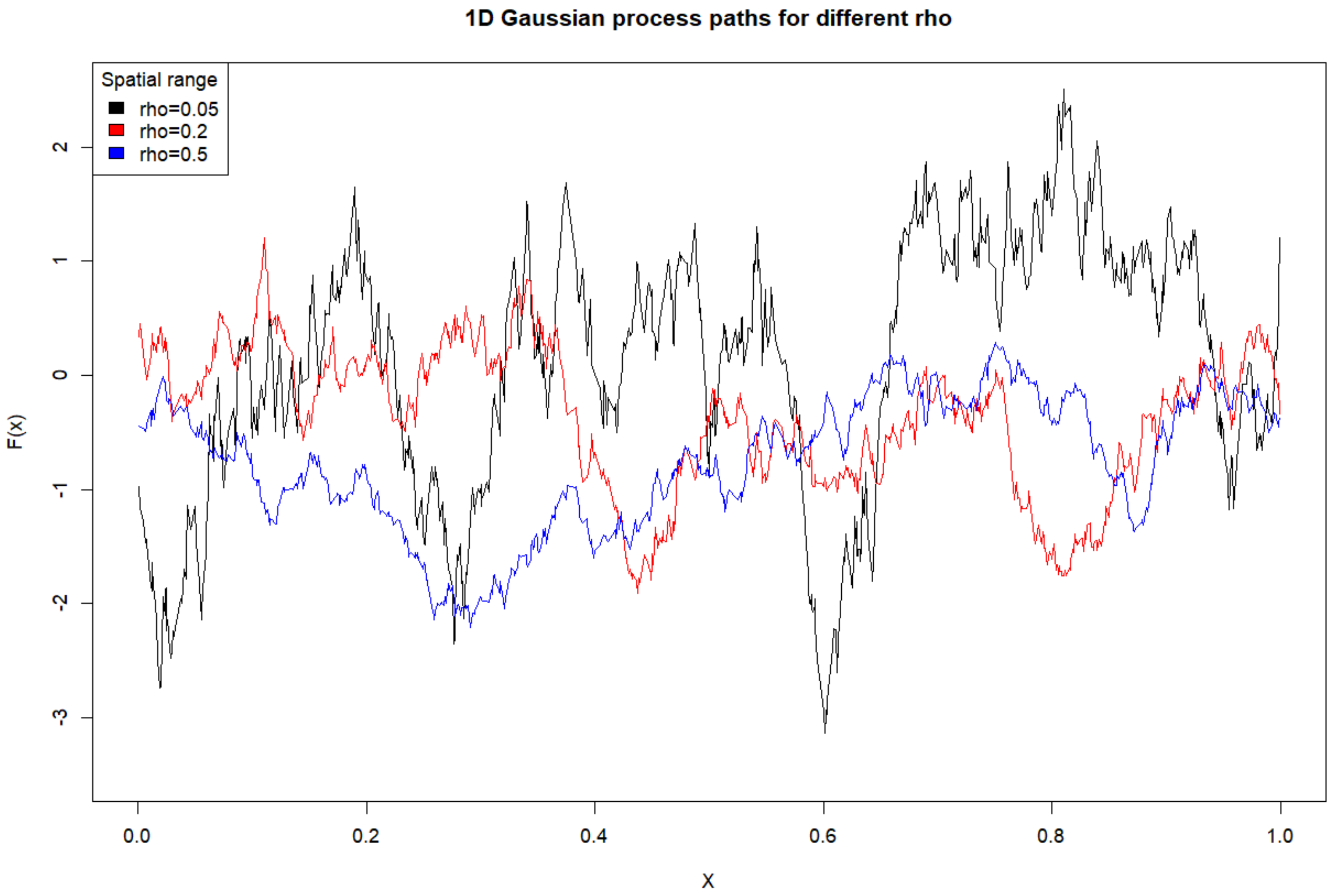
Similar to earlier, we may want to visualize the strength of the implied correlation, now w.r.t. distance:



We can now also examine spatial trends per species:



corExp paths are not “smooth”



Bonus: squared exponential kernel

With a slight modification of `corExp`, we can define the famous —and as smooth as it gets—squared exponential kernel:

$$K(d) = \sigma^2 \exp\left(-\frac{d^2}{2\rho^2}\right)$$

- The degree of smoothness needed might vary greatly in different applications and scenarios
- As an example, consider stratiform vs. orographic rainfall[1]

The Matérn family

strikes a balance between the previous two kernel functions:

$$K(d) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu} d}{\rho} \right)^{\nu} \mathcal{K}_\nu \left(\frac{\sqrt{2\nu} d}{\rho} \right),$$

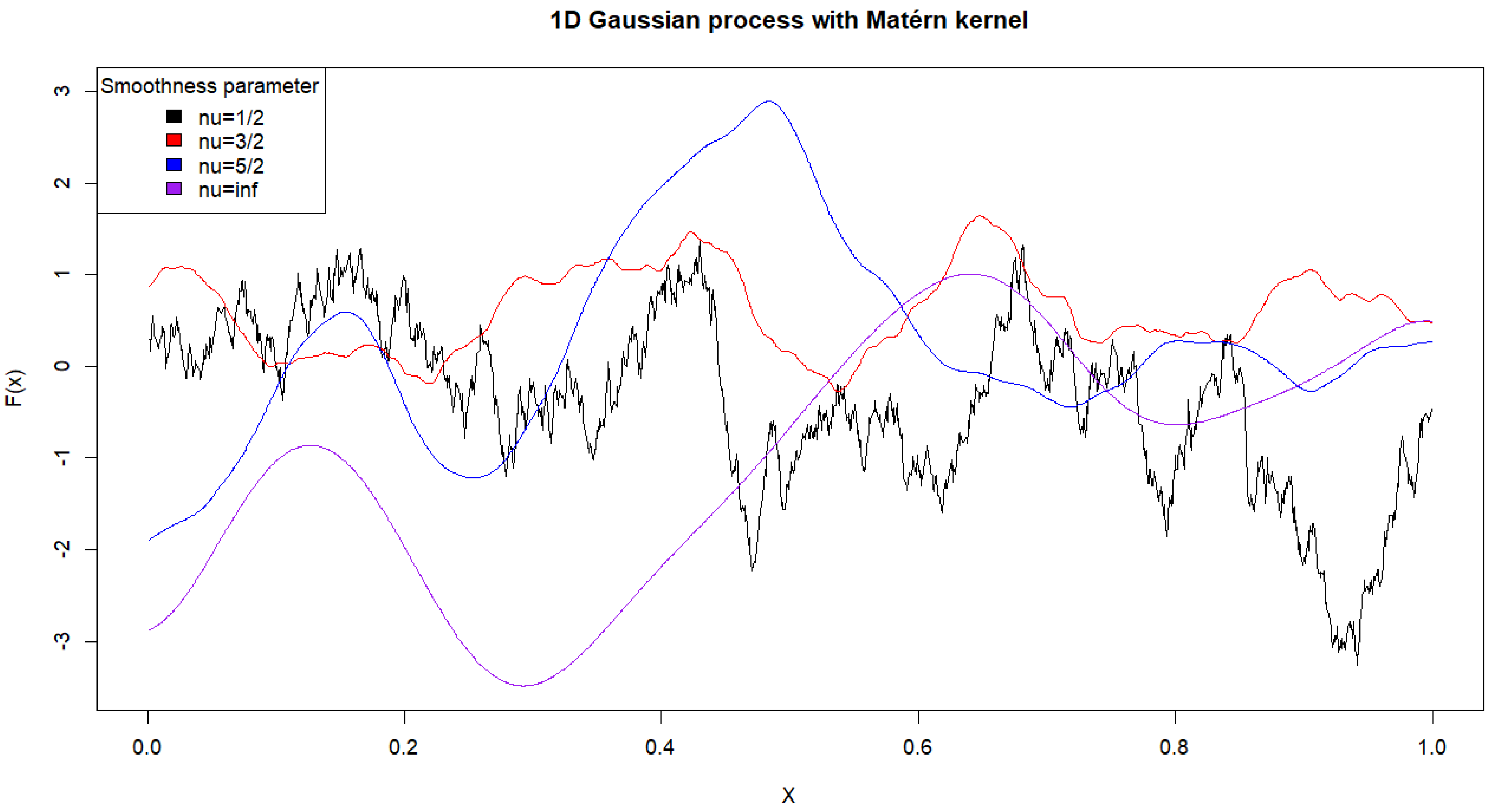
where $\Gamma(\cdot)$ is the gamma function, $\mathcal{K}_\nu(\cdot)$ is the modified Bessel function, and $\nu > 0$ controls the smoothness.

- If (squared) `corExp` has smoothness of (∞) 0, Matérn has that of $\lceil \nu \rceil - 1$; e.g., $\nu = 5/2$ implies degree of 2.
- ν is notoriously difficult to estimate, and thus often fixed

Some facts regarding Matérn kernels

- For half-integer ν, $K(d)$ simplifies into a product of the exponential kernel and a polynomial, e.g.:
  - $\nu = 1/2 : K(d) = \sigma^2 \exp(-d/\rho)$
  - $\nu = 3/2 : K(d) = \sigma^2 (1 + \sqrt{3} d/\rho) \exp(-\sqrt{3} d/\rho)$
- Dubbed the “most natural” covariance function for 2D
  - Similar to how `corExp` is for 1D (the Markov property)
- For fixed ν and upto 3D, only the *microergodic* parameter $\sigma^2/\rho^{2\nu}$ can be consistently estimated from data

Matérn paths



Estimation of dynamic GLLVMs

On top of obviously having to deal with the various newly added parameters like σ^2, ρ, ν, etc., how much does the model estimation actually change?

Sadly, quite a bit. Regarding variational approximations:

- The variational log-likelihood now involves terms with Σ_q, or rather, their inverses, which are costly to compute
- The number of variational parameters also increases, to accommodate the covariances between LVs or rows
    - increases both computation time and memory requirements

Sparse approximations

are required to facilitate efficient estimation of dynamic GLLVMs, ideally for both Σ^{-1} and A, i.e., we seek some $\hat{\Sigma}^{-1}$, $\hat{A}$ such that most of the elements in the matrices are 0.

- Popular approaches for Σ^{-1} get discussed in the practicals
- For $A = LL^\top$, a couple of alternatives exist in `gllvm`:
  - block-diagonal nearest neighbour sparse Cholesky factorization (`Lambda.struc="bdNN"`)
  - Kronecker product of unstructured with NN sparse Cholesky factorization (`Lambda.struc="UNN"`)

Nearest neighbour approximation

Let $A = LL^\top$ be a $n \times n$ variational (spatial) covariance matrix, with lower diagonal matrix L as its Cholesky factor.

Assume that the locations $i = 1, \ldots, n$ have been arranged according to a particular ordering (can be random). For each i, determine then its *neighbours* $N(i)$ by choosing k sites among $l = 1, \ldots, i-1$, which are closest to site i, geographically.

Then, a sparsity pattern can be imposed by requiring $\hat{L}[l, i]$ to be zero, whenever $l \notin N(i)$, i.e., l is not a neighbour of i.

The quality of the approximation depends on both the number of neighbours k and the ordering of the sites.

Work in progress for `gllvm`

- NNGP for spatial models
- Gaussian process with inducing points

State of spatial JSDM packages in general

- HMSC
- Community-level basis functions
- sdmTMB

References

De Oliveira, Victor, and Zifei Han. 2022. "On Information about Covariance Parameters in Gaussian Matérn Random Fields." *Journal of Agricultural, Biological and Environmental Statistics* 27 (4): 690–712.

Reed, Daniel C., and Robert J. Miller. 2023. "SBC LTER: Reef: Kelp Forest Community Dynamics: Cover of Sessile Organisms, Uniform Point Contact. LTER Network Member Node." https://doi.org/10.6073/pasta/0af1a5b0d9dde5b4e5915c0012ccf99c.