

A wavelet-based method to remove spatial autocorrelation in the analysis of species distributional data

Gudrun Carl, Carsten F. Dormann and Ingolf Kühn

Carl, G., Dormann, C. F. and Kühn, I. 2008. A wavelet-based method to remove spatial autocorrelation in the analysis of species distributional data. – *Web Ecol.* 8: 22–29.

Species distributional data based on lattice data often display spatial autocorrelation. In such cases, the assumption of independently and identically distributed errors can be violated in standard regression models. Based on a recently published review on methods to account for spatial autocorrelation, we describe here a new statistical approach which relies on the theory of wavelets. It provides a powerful tool for removing spatial autocorrelation without any prior knowledge of the underlying correlation structure. Our wavelet-revised model (WRM) is applied to artificial datasets of species' distributions, for both presence/absence (binary response) and species abundance data (Poisson or normally distributed response). Making use of these published data enables us to compare WRM to other recently tested models and to recommend it as an attractive option for effective and computationally efficient autocorrelation removal.

G. Carl and I. Kühn (ingolf.kuehn@ufz.de), UFZ – Helmholtz Centre for Environmental Research, Dept Community Ecology (BZF), Theodor-Lieser-Strasse 4, DE-06120 Halle, Germany. GC and IK also at: Virtual Institute Macroecology, Theodor-Lieser-Strasse 4, DE-06120 Halle, Germany. – C. F. Dormann, UFZ – Helmholtz Centre for Environmental Research, Dept Computational Landscape Ecology (CLE), Permoser Str. 15, DE-04318 Leipzig, Germany.

Species distributional models based on lattice data often display spatial autocorrelation. Spatial autocorrelation means that observations from nearby locations are often more similar than would be expected on a random basis (Legendre and Legendre 1998). Spatial autocorrelation can arise in both species distributions and environmental variables. Note that statistical analyses of such data need not be seen as problematic in principle. However, a chosen method is inconsistent with its application, if and only if (1) these autocorrelated variables lead to autocorrelated errors and (2) independently and identically distributed (i.i.d.) errors are assumed in the used statistical model. In that case results of the method are not reliable (Kühn 2007).

There are several reasons for autocorrelated errors in linear regressions (Kissling and Carl 2008). (1) Response variables, e.g. species distributions are spatially structured due to endogenous properties such as, e.g. dispersal, speciation, and extinction. Because structure is only inherent in the response variable, it can not be explained by explanatory environmental variables. Therefore, it leaves its mark on the regression errors. (2) Response variables are spatially structured due to exogenous properties such as specific environmental variables, e.g. wind or other climatic constraints. In the model, however, the very same variables are either not included or improperly used due to neglected non-linear transformations. Here the spatial structure, in turn, affects the errors.

Accepted 4 April 2008

Copyright © EEF

ISSN 1399-1183

Note that i.i.d. errors are assumed in so-called standard regression models such as in ordinary least squares and generalized linear models. However, Dormann et al. (2007) presented an overview of different modelling approaches that are available to account for spatial autocorrelation in the analysis of lattice data. In this paper, we describe and test a new method which is based on wavelet transforms. We focus on a short description of the so-called Wavelet-revised model (WRM), and on a presentation of the WRM results for the data of Dormann et al. (2007) in exactly the same way as specified in that paper. An application of the new method on those artificial species distribution datasets enables us to further test the potential of the wavelet method and to compare it to more established methods.

Wavelets are 'small' or 'local' waves, i.e. specific functions useful for the transformation of time series or images. An analysis of such data based upon a wavelet transform can help to pick out features of interest (Percival and Walden 2000). In particular, wavelet transforms are a promising tool to remove spatial autocorrelation (Carl and Kühn 2008). The key idea is that data preparation can be carried out by means of, for example, Haar wavelets. This procedure basically averages data within squared subareas and subtracts this average from the data to remove autocorrelation.

Method

Simulated datasets

The datasets used here are exactly the same artificial datasets as in the paper published by Dormann et al. (2007). In that paper the authors offer lattice data containing 1108 grid cells. Two explanatory variables are introduced. The first one 'rain' is a significant predictor of the response, whereas the second one 'djungle' does not have any explanatory power (noise variable). The response itself can be imagined as species distribution. Three types of response distributions are considered: normal, binomial and Poisson. The models for the expected values of responses are:

- (1) $E(y_i) = 80 - 0.015 \times \text{rain}_i$ for normal distribution;
- (2) $E(y_i) = g^{-1}(3 - 0.003 \times \text{rain}_i)$ for binomial distribution;
- (3) $E(y_i) = g^{-1}(3 - 0.001 \times \text{rain}_i)$ for Poisson distribution,

where g is the logit-link and log-link function for binomial and Poisson distribution, respectively. The responses basically consist of their expected values and associated errors. In the strict sense, they are generated as normally, binomial or Poisson distributed values of $E(y) + \text{sd}(y) \times \epsilon$, where sd is the standard deviation and ϵ is the error. Spatial autocorrelation is mainly implemented on the errors. It is

simulated by use of an exponential function which results in strongly correlated errors for neighbouring cells but a steep decline of autocorrelation for increasing distance.

Although the second predictor 'djungle' is not used in simulation, it is entered as predictor into all of the following statistical models. This is done to assess models regarding their significance tests.

Generalized linear models: GLM

In order to estimate regression coefficients for the above mentioned data, Generalized linear models (GLM) are commonly used. GLM is the standard method in case response variables have distributions other than the normal distribution. The GLM estimator for regression coefficients is:

$$b^{(m)} = (X'WX)^{-1}X'Wz,$$

where X is the design matrix and W is a weights matrix. The vector z

$$z = Xb^{(m-1)} + W^{-1/2}V^{-1/2}(y - \mu)$$

is an adjusted response variable depending on the real response y , its expected value μ , and its variance matrix V (Dobson 2002, Myers et al. 2002). These equations have to be solved iteratively because, in general, z and W depend on b . Therefore, it is named an iterative weighted least squares procedure.

If the responses are normally distributed, then W is the identity matrix and z is equivalent to y . Moreover, the iterative algorithm leads to a non-iterative model and GLM is simplified to the straightforward Ordinary least squares (OLS) method where

$$b = (X'X)^{-1}X'y$$

Unfortunately, standard methods such as OLS or GLM may yield wrong results when data are sampled in a spatial context. Due to the fact that data sampled at adjacent locations are more likely to be similar than distant ones, the basic assumption of i.i.d. errors can be violated. In spatial statistics it is, therefore, desirable to revise data by removing spatial autocorrelations. We are going to show that we can achieve this goal by using the concept of wavelet decomposition.

Autocorrelation removal via wavelets

First of all, it is necessary that the following preliminary decisions are made: (1) there are numerous types of wavelets in general use (Daubechies 1992, Hubbard 1998). We decide to utilize so-called Haar wavelets which have a com-

compact and square-edged shape and are thus useful in the detection of edges and gradients. (2) We deal with datasets as they appear in statistical samples, especially in linear regressions. To this end we apply the wavelet transform in a specific form, the so-called discrete wavelet transform, which is a calculation for a finite set of discrete data (Bruce and Gao 1996). (3) We have to take into account the spatial, i.e. two-dimensional structure of lattice data. Therefore, we have to apply a two-dimensional approach of wavelet theory (Bruce and Gao 1996).

Because Haar wavelets are the simplest wavelets, it is possible to illustrate how autocorrelation removal works under the conditions outlined above. Haar transforms can be viewed as a series of averaging and differencing operations.

Here is an example of this procedure. Let us demonstrate the Haar wavelet analysis of the following data matrix F :

$$F = \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix}$$

In the first step the matrix is simplified by calculating the average value of the resulting non-overlapping 2×2 blocks of cells. Therefore, the so-called smooth matrix S_1 is:

$$S_1 = \begin{pmatrix} q & q & r & r \\ q & q & r & r \\ s & s & t & t \\ s & s & t & t \end{pmatrix}$$

where q =average (a, b, e, f), r =average (c, d, g, h), etc.

The so-called detail matrix D_1 gives the difference to the original matrix F :

$$D_1 = \begin{pmatrix} a-q & b-q & c-r & d-r \\ e-q & f-q & g-r & h-r \\ i-s & j-s & k-t & l-t \\ m-s & n-s & o-t & p-t \end{pmatrix}$$

In the second step we proceed analogously, now averaging 4×4 blocks. Therefore, the smooth matrix S_2 on level 2

$$S_2 = \begin{pmatrix} u & u & u & u \\ u & u & u & u \\ u & u & u & u \\ u & u & u & u \end{pmatrix}$$

is associated with the detail matrix D_2 on the same level

$$D_2 = \begin{pmatrix} a-u & b-u & c-u & d-u \\ e-u & f-u & g-u & h-u \\ i-u & j-u & k-u & l-u \\ m-u & n-u & o-u & p-u \end{pmatrix}$$

More formally this procedure can be explained as follows: a two-dimensional discrete Haar wavelet analysis can be performed, provided that data are given as a $2^n \times 2^n$ matrix. On level J we divide this original matrix F in $2^J \times 2^J$ submatrices and construct a smooth $2^n \times 2^n$ matrix S_J by assigning the mean value within each submatrix to each element of the submatrix. Then the detail matrix D_J is the difference between original matrix and smooth matrix $D_J = F - S_J$.

Therefore, the original data can be regarded as a sum of detail and smooth components on a certain level, i.e. at a certain resolution (Bruce and Gao 1996). In general, this means that in the smooth component the smoothness as data feature is captured. Smoothness implies that data values of locations close to each other are more similar than those further apart. Thus it represents autocorrelation at a certain scale. Detail components, however, are data adjusted for autocorrelation.

Description of the spatial statistical method: WRM

Our new method is named Wavelet-revised model (WRM), because it provides the revision of data outlined above (Carl and Kühn 2008). The WRM approach for calculating regression parameters embeds this data revision into the framework of GLM. To this end additional steps have to be incorporated into the GLM iteration loops.

Note that the wavelet approach can be carried out by means of so-called multiresolution analysis. Here the resolution level J should be small (i.e. fine resolution) in order to extract short-range autocorrelation. Moreover, the two-dimensional analysis should be applied to both the responses and the individual predictors. They are given as vectors or columns of the design matrix within the GLM iteration loop. Therefore, we have to convert these vectors into matrices which reflect their spatial sampling structure. After autocorrelation removal we go back to vectors to proceed as normal.

The approach can be summarized as follows:

Step 1. Create matrices for all columns of the matrix of weighted predictors and for the vector of the adjusted dependent variables according to their spatial structure.

Step 2. Perform two-dimensional multiresolution analysis on each of these matrices.

Step 3. Add up all detail components D , excluding the smooth components S .

Step 4. Transform matrices into vectors.

Step 5. Use these vectors updated within each loop in an iterative weighted least squares procedure.

Software

Our computations are based on a software package in the computer language R (R Development Core Team 2005).

Table 1. Model quality: Spatial autocorrelation in the model residuals (given as global Moran's I) and mean estimates for the coefficients 'rain' and 'djungle' (± 1 SE across the 10 simulations). True coefficient values are given in the first row for each distribution in italics. ***, * and ^{ns} refer to median significance levels of $p < 0.001$, $0.01 < p < 0.05$ and $p > 0.1$, respectively, across the 10 realisations. GLM: Generalized linear model, WRM: Wavelet-revised model.

		Moran's I	'rain'	Coefficients	'djungle'
Normal			<i>-0.015</i>		<i>0.0</i>
	GLM	0.016 \pm 0.026	-0.0143 \pm 0.0010***		0.0220 \pm 0.0508 ^{ns}
	WRM	-0.001 \pm 0.001	-0.0138 \pm 0.0021***		0.0250 \pm 0.0285 ^{ns}
Binomial			<i>-0.003</i>		<i>0.0</i>
	GLM	0.006 \pm 0.011	-0.0022 \pm 0.0003***		0.0052 \pm 0.0130 ^{ns}
	WRM	-0.002 \pm 0.000	-0.0025 \pm 0.0011*		0.0034 \pm 0.0154 ^{ns}
Poisson			<i>-0.001</i>		<i>0.0</i>
	GLM	0.018 \pm 0.024	-0.0010 \pm 0.0000***		0.0006 \pm 0.0018 ^{ns}
	WRM	-0.001 \pm 0.001	-0.0010 \pm 0.0002***		0.0010 \pm 0.0021 ^{ns}

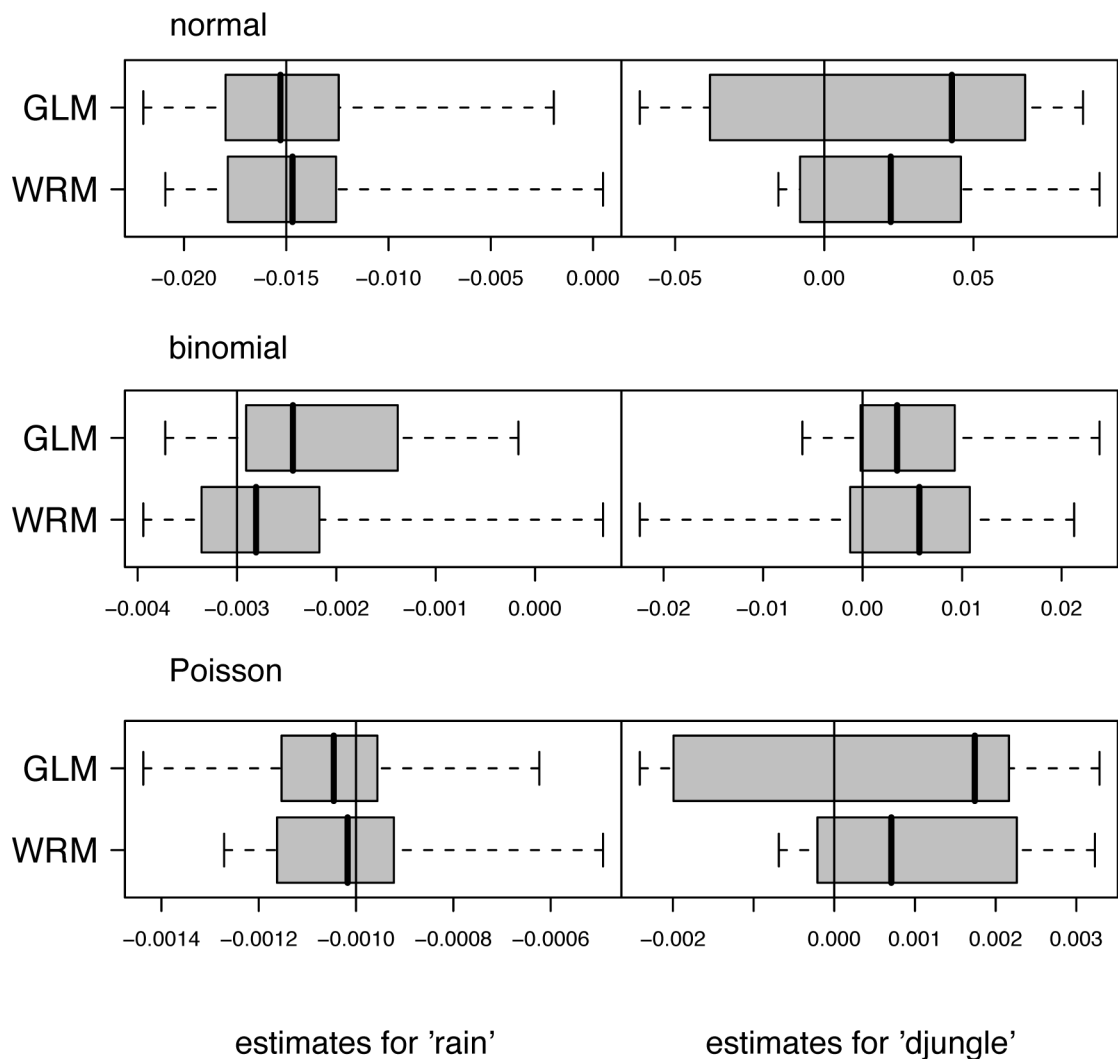


Fig. 1. Comparison of non-spatial (GLM) and spatial (WRM) modelling approaches for data with normally, binomially and Poisson-distributed responses. Box and whiskers refer to 25/75%, 0/100%. Vertical lines indicate true values of parameter.

The tools for calculating wavelet transforms are available in package *waveslim* (Whitcher 2005). We used the function *mra.2d* for multiresolution analysis. Our R codes are available as supplementary material.

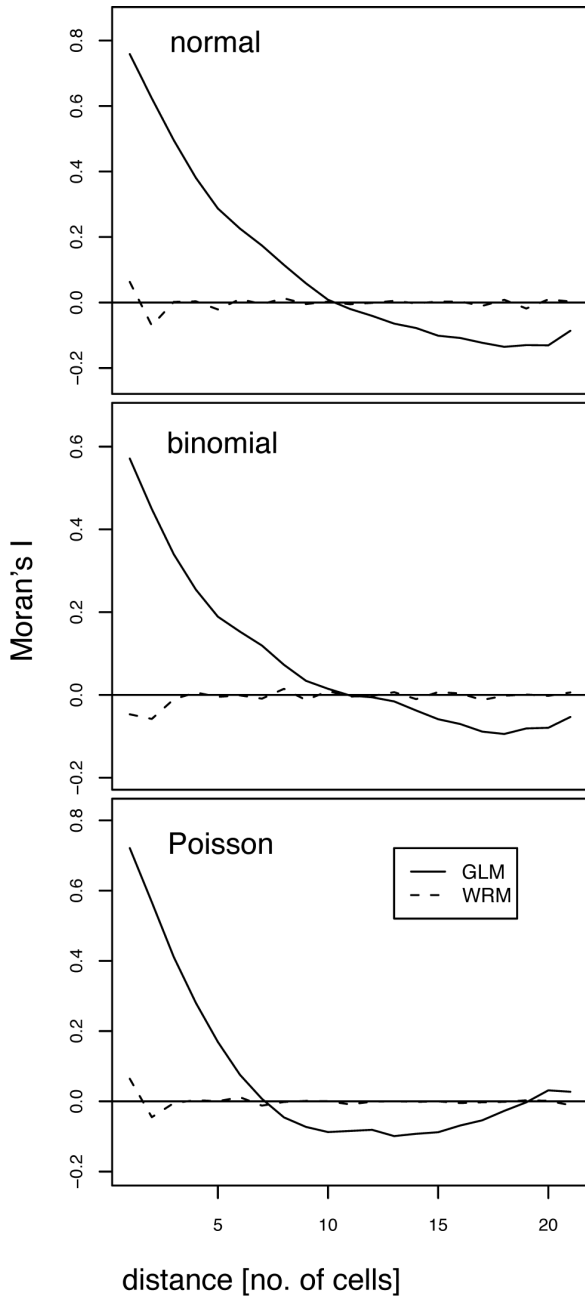


Fig. 2. Correlograms of one realisation for each of the three different distributions (normal, binomial, Poisson) and the methods compared.

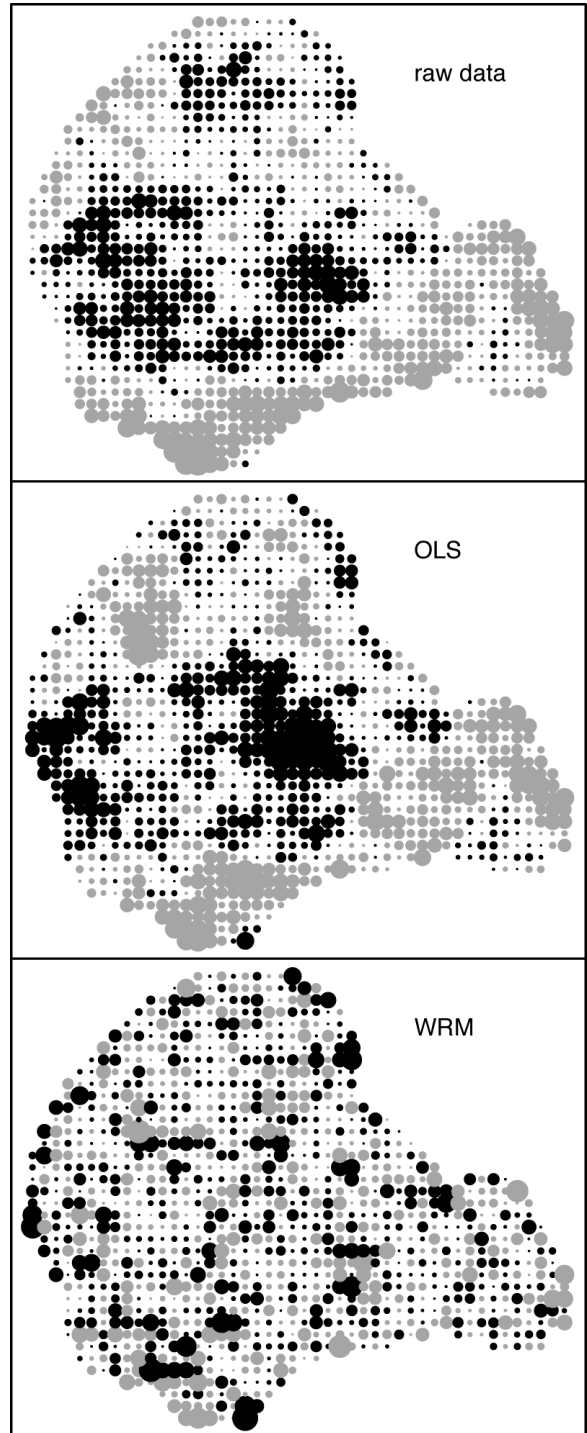


Fig. 3. Raw distributional data and residual maps for the different methods for one realisation of the data with normally distributed responses. On the residual maps, pixel sizes indicate sizes of residuals. The two colours black and grey represent the positive and negative signs of residuals, and areas of equal colour indicate autocorrelation. For WRM residuals the areas of equal colour are essentially reduced compared to OLS residuals.

Tricks and tips

The function `mra.2d` offers various wavelets. We used Haar wavelets. Because of the restriction to finite sets in discrete wavelet transforms it is necessary to give boundary treatment rules. Type `periodic` is implemented for boundary conditions in the software for the two-dimensional discrete wavelet transform. This causes a restriction on the

sample size. The number of rows and columns must be dyadic (i.e. 2^n , where n is an integer). In general, one wishes to analyze samples of arbitrary size though. For this reason data are padded with zeros until a quadratic matrix of required size is reached (Bruce and Gao 1996).

For all subsequent analyses, our WRM method should be compared to the methods given in Dormann et al. (2007). Therefore, we present the results in the same way

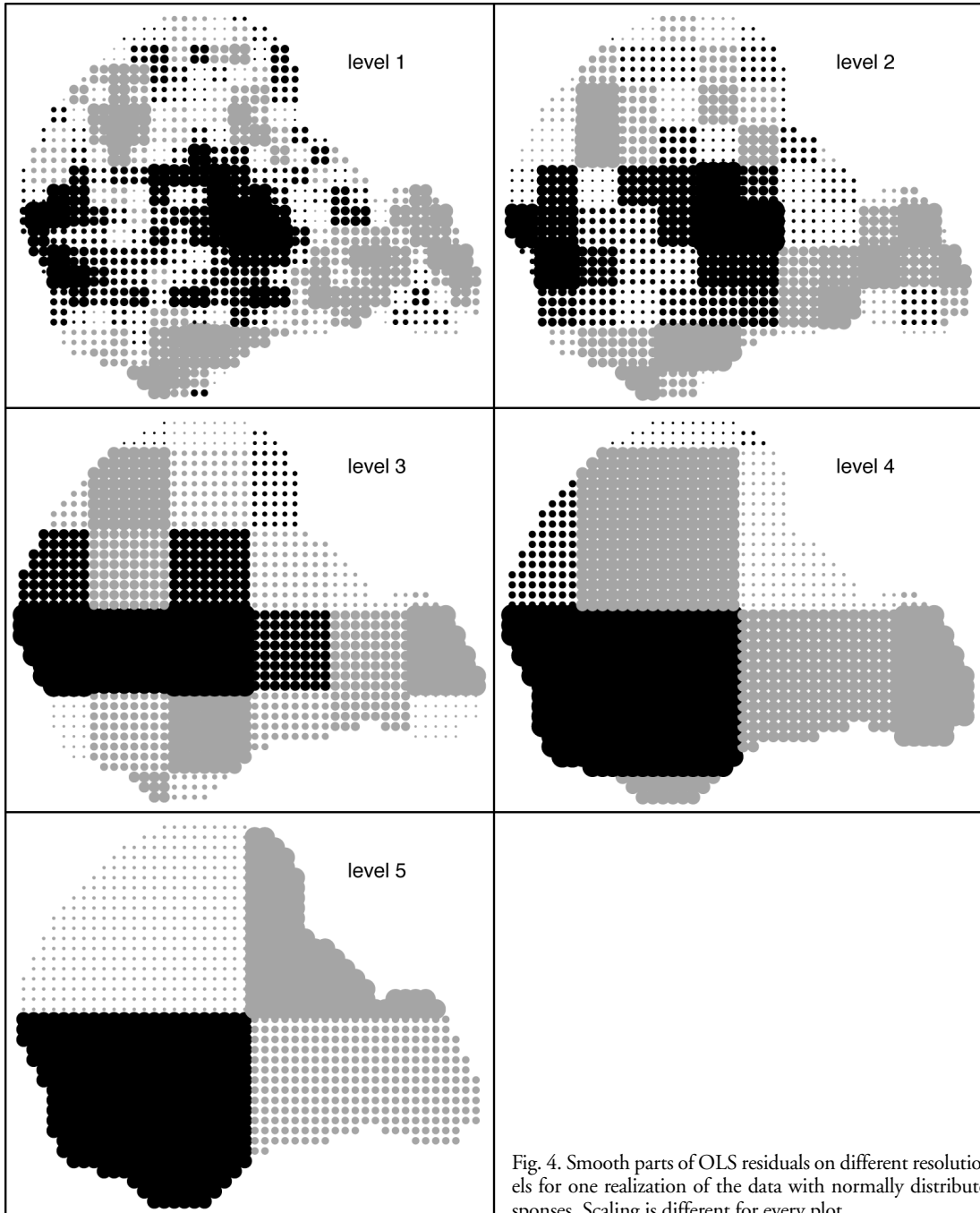


Fig. 4. Smooth parts of OLS residuals on different resolution levels for one realization of the data with normally distributed responses. Scaling is different for every plot.

as it was done in that paper. Table 1 and Fig. 1–3 given here correspond to Table 2 and Fig. 1–3 in Dormann et al. (2007). For the purpose of comparison, the results of the non-spatial models OLS/GLM are given here as well.

Results

WRM models yield approximately the same values as expected and as calculated by GLM for means and variances of the estimates ‘rain’ and ‘djungle’ (Fig. 1). However, only 10 data realizations were evaluated in each case of data with normally, binomially and Poisson-distributed responses. Thus conclusions regarding model performance should be avoided here.

Our WRM model and the non-spatial model GLM differ obviously in the spatial signature of their residuals. Figure 2 shows that residual autocorrelation measured by Moran’s I is considerably reduced compared to GLM. Corresponding values for Global Moran’s I are given in Table 1. WRM always performs better than GLM. Moreover, we present maps to determine whether there are any spatial patterns of regression residuals (Fig. 3). The maps show raw data y , OLS residuals and WRM residuals for a dataset with normally distributed responses. They demonstrate that WRM removes the clusters of positive or negative OLS residuals. WRM was carried out on level 1. Figure 4 shows patterns for smooth parts of OLS residuals at different resolutions. These smooth components need to be removed for WRM residuals. Level 1 captures the smooth parts best describing the spatial autocorrelation structure in this dataset. The decorrelated WRM residuals (Fig. 3) can be imagined as the difference of OLS residuals (Fig. 3) and their smooth parts at the finest resolution, i.e. on level 1 (Fig. 4, top left).

Discussion

Dormann et al. (2007) presented an overview of different modelling approaches for multiple regressions displaying spatial autocorrelation. The authors state that the most flexible methods applicable to non-normal distributions are spatial GLMM, GEE and SEVM. Our new WRM method represents an alternative approach to obtain an approximately equal performance.

Carl and Kühn (2008) investigated the type I error control of WRM in a more reliable analysis of 500 data realizations per distribution. The results showed a better error calibration curve for WRM than for GLM. Furthermore, in comparing the above-mentioned models one has to consider the following facts:

(1) GLMM and GEE are useful to correct autocorrelation effects when the correlation structure is known as in our simulated datasets. WRM, however, provides a powerful tool for removing autocorrelation without any prior

knowledge of the underlying correlation structure in lattice data. However, WRM requires comparably large sample sizes whereas GEE may reach their limits with large sample sizes (Carl and Kühn 2007, Dormann et al. 2007). The methods may therefore be regarded as complementary.

(2) SEVM (and Bayesian) methods are very time-consuming, whereas WRM is a computationally very fast and efficient procedure.

(3) Almost all methods assume spatial stationarity, i.e. spatial autocorrelation to be constant across the region, whereas WRM is a method that allows for spatial non-stationarity. Moreover, WRM is able to detect anisotropic autocorrelation. This is true for at least such autocorrelation which acts differently in vertical, horizontal, and diagonal direction.

Summarizing, this paper presented a wavelet-based method for regressions influenced by spatial autocorrelation. WRM is a good alternative to the methods described in Dormann et al. (2007), especially for large data-sets.

Acknowledgements – The authors acknowledge funding by the European Union within the FP 6 Integrated Project ‘ALARM’ (GOCE-CT-2003-506675). Moreover, they acknowledge support from the ‘Virtual Institute for Macroecology’, funded by the Helmholtz Association (VH-VI-153 Macroecology, Kühn et al. 2008). This contribution is based on the international workshop ‘Analysing spatial distribution data: principles, applications and software’ (GZ 4850/191/05) supported by the German Science Foundation (DFG).

References

- Bruce, A. and Gao, H. Y. 1996. Applied wavelet analysis with S-plus. – Springer.
- Carl, G. and Kühn, I. 2007. Analyzing spatial autocorrelation in species distributions using Gaussian and logit models. – *Ecol. Modell.* 207: 159–170.
- Carl, G. and Kühn, I. 2008. Analyzing spatial ecological data using linear regression and wavelet analysis. – *Stochastic Environ. Res. Risk Assess.* 22: 315–324.
- Daubechies, I. 1992. Ten lectures on wavelets. CSBM-NSF Series Application Mathematics. Vol. 61. – SIAM Publication.
- Dobson, A. J. 2002. An introduction to generalized linear models. – Chapman and Hall.
- Dormann, C. F. et al. 2007. Methods to account for spatial autocorrelation in the analysis of species distributional data: a review. – *Ecography* 30: 609–628.
- Hubbard, B. B. 1998. The world according to wavelets. – A. K. Peters.
- Kissling, W. D. and Carl, G. 2008. Spatial autocorrelation and the selection of simultaneous autoregressive models. – *Global Ecol. Biogeogr.* 17: 59–71.
- Kühn, I. 2007. Incorporating spatial autocorrelation may invert observed patterns. – *Div. Distr.* 13: 66–69.
- Kühn, I. et al. 2008. Macroecology meets global change research. – *Global Ecol. Biogeogr.* 17: 3–4.
- Legendre, P. and Legendre, L. 1998. Numerical ecology. – Elsevier.

- Myers, R. H. et al. 2002. Generalized linear models. – Wiley.
- Percival, D. B. and Walden, A. T. 2000. Wavelet methods for time series analysis. – Cambridge Univ. Press.
- R Development Core Team. 2005. R: a language and environment for statistical computing. – R Found. Stat. Comp.
- Whitcher, B. 2005. Waveslim: basic wavelet routines for one-, two- and three-dimensional signal processing. – R package, ver. 1.5.


```

WRMcode.txt
#####
# WRMcode.txt #
#####

#
# Carl & Dormann & Kühn: A wavelet-based method to remove spatial autocorrelation
# in the analysis of species distributional data
# Electronic supplementary material
# Functions used to do the analyses
# Codes are written for R (www.r-project.org)
# using package waveslim
# Gudrun Carl, 2007
#
# Autocorrelation correcting code using wavelets
# to calculate regression models
# for responses: "gaussian", "binomial"(binary) or "poisson"
# for spatial (2-dimensional) autocorrelation
# in macroecological data (regular gridded datasets,
# grid cells are assumed to be square)
#
#####

#####
WRM<-function(formula, family, data, coord, level=0,
wavelet="haar", plot=FALSE, graph=FALSE){
#####
# Arguments:
# formula with specified notation according to names in data frame
# family "gaussian", "binomial"(binary) or "poisson"
# data data frame
# coord corresponding coordinates which have to be integer
# level 0 for GLM
# 1 for best autocorrelation removal
# higher integers possible (limit depends on sample size)
# wavelet "haar" is recommended
# value: WRM returns a list containing the following elements
# b estimate of regression parameters
# s.e. standard errors
# z z values (or corresponding values for statistics)
# p probabilities
# fitted fitted values
# resid Pearson residuals
# if plot or graph is true:
# w.ac autocorrelation of level parts for wavelet.residuals
# ac.glm autocorrelation of glm.residuals
# ac autocorrelation of wavelet.residuals
#####
require(waveslim)
n<-dim(data)[1]
l<-dim(data)[2]
x<-coord[,1]
y<-coord[,2]
if(length(x)!=n) stop("error in dimension")
logi c1<-identical(as.numeric(x), round(x, 0))
logi c2<-identical(as.numeric(y), round(y, 0))
if(!logi c1 | !logi c2) stop("coordinates not integer")

X<-model.matrix(formula, data)
nvar<-dim(X)[2]

```

```

if(is.vector(model.frame(formula, data)[[1]])){
  yold<-model.frame(formula, data)[[1]]
  ntr<-1
}
if(family=="binomial" & is.matrix(model.frame(formula, data)[[1]])){
  yold<-model.frame(formula, data)[[1]][, 1]
  ntr<-model.frame(formula, data)[[1]][, 1]+model.frame(formula, data)[[1]][, 2]
}

n.level<-level
length.s<-3*n.level+1
s<-rep(1, length.s)
s[length.s]<-0
if(level==0){s<-c(1, 1, 1, 1) ; n.level<-1}
beta<-matrix(NA, 4, nvar)
resi<-matrix(NA, 4, n)
ac<-matrix(NA, 4, 10)
acp<-matrix(NA, 4, 10)
se<-matrix(NA, 4, nvar)

pdim<-max(max(y)-min(y), max(x)-min(x))*3/2
power<-0
while(2^power<pdim) power<-power+1
xmargi n0<-as.integer((2^power-(max(x)-min(x)))/2)-min(x)+1
ymargi n0<-as.integer((2^power-(max(y)-min(y)))/2)-min(y)+1

if(power<n.level) stop("level is too high")

i4<-1
while(i4<5){ #.....
  if(i4==1){xmargi n<-xmargi n0 ; ymargi n<-ymargi n0}
  if(i4==2){xmargi n<-xmargi n0+1 ; ymargi n<-ymargi n0}
  if(i4==3){xmargi n<-xmargi n0+1 ; ymargi n<-ymargi n0+1}
  if(i4==4){xmargi n<-xmargi n0 ; ymargi n<-ymargi n0+1}

# GLM for comparison
m0<-glm(formula, family, data)
res0<-resid(m0, type="pearson")
beta0<-m0$coeff

#-----

betaw<-rep(0, nvar)
lin<-X%*betaw
if(family=="gaussian") pi<-lin
if(family=="binomial") pi<-exp(lin)/(1+exp(lin))
if(family=="poisson") pi<-exp(lin)

#if(family=="gaussian") pi<-rep(0, n)
#if(family=="binomial") pi<-rep(.5, n)
#if(family=="poisson") pi<-rep(1, n)

it<-0

repeat{
  it<-it+1
  if(family=="gaussian") var<-rep(1, n)
  if(family=="binomial") var<-ntr*pi*(1-pi)
  if(family=="poisson") var<-pi
  sigma<-as.vector(sqrt(var))
  W12<-diag(sigma)
  Am12<-diag(1/sigma)
  Xnew<-W12%*X

```

```

ynew<-W12%*%X%*%betaw+Am12%*(yold-ntr*pi)

F<-matrix(0, 2^power, 2^power)
T<-array(0, c(2^power, 2^power, nvar))
for(ii in 1:n){
kx<-x[ii]+xmargi n
ky<-y[ii]+ymargi n
F[ky, kx]<-ynew[ii]
for (i3 in 1:nvar)
T[ky, kx, i3]<-Xnew[ii, i3]
} # ii loop

p<-2^power*2^power
tt<-matrix(0, p, nvar)
if(is.na(max(abs(F)))) {mdwt$coeff<-rep(NA, nvar); break}
if(is.infinite(max(abs(F)))) {mdwt$coeff<-rep(NA, nvar); break}
FT<-mra.2d(F, wavel et, n. l evel , method="dwt")
FTS <- FT[[1]]
for(is in 2:length(s))
if(s[is]==1) FTS <- FTS + FT[[is]]
ft<-as.vector(FTS)
for (i3 in 1:nvar){
TT<-mra.2d(T[, , i3], wavel et, n. l evel , method="dwt")
TTS <- TT[[1]]
for(is in 2:length(s))
if(s[is]==1) TTS <- TTS + TT[[is]]
tt[, i3]<-as.vector(TTS)
}

xnam<-paste("tt[, ", 1:nvar, "]", sep="")
formul a.dwt<-as.formul a(paste("ft-", paste(xnam, col l apse="+"), "-1"))
mdwt<-lm(formul a.dwt)
if(max(abs(mdwt$coeff))>1e+10 ) {mdwt$coeff<-rep(NA, nvar); break}

lin<-X%*%mdwt$coeff
if(fami l y=="gaussi an") pi <-lin
if(fami l y=="bi nomi al") pi <-exp(lin)/(1+exp(lin))
if(fami l y=="poi sson") {pi <-exp(lin)
if(mi n(pi )<1e-10 | max(pi )>1e+10) {mdwt$coeff<-rep(NA, nvar); break}
}

if (i denti cal (round(betaw, 5), round(mdwt$coeff, 5)) ) {i4<-4 ; break}
if (i4==4 & it > 200) stop("too many iterations")
if (it > 200) break
betaw<-mdwt$coeff
} # next step of iteration

if(!is.na(mdwt$coeff[1])){
Resmdwt<-matrix(resi d(mdwt), 2^power, 2^power)
resmdwt<-rep(0, n)
for(i in 1:n) resmdwt[i ]<-Resmdwt[y[i ]+ymargi n, x[i ]+xmargi n]
if(plot | graph) {
acw<-acpart(x, y, resmdwt, resmdwt)
acpw<-acpart(x, y, resmdwt, res0)
}
if(!plot & !graph) {acw<-NA; acpw<-NA}
var.b<-sol ve(t(tt)%*%tt)
if(fami l y=="gaussi an"){
if(l evel ==0) df<-n-nvar
if(l evel !=0) df<-n-round(n/4^n. l evel )-nvar
si gma2<-sum(resmdwt^2)/df
var. b<-si gma2*var. b
}

```

```

}
s.e. <-rep(NA, nvar)
for(i in 1:nvar){
s.e. [i] <-sqrt(var. b[i, i])
}
}

if(is.na(mdwt$coeff[1])) {acw<-NA; acpw<-NA; resmdwt<-NA; s.e. <-NA}
beta[i 4, 1:nvar] <-mdwt$coeff[1:nvar]
resi [i 4, 1:n ] <-resmdwt[1:n]
ac[i 4, 1: 10] <-acw[1: 10]
acp[i 4, 1: 10] <-acpw[1: 10]
se[i 4, 1:nvar] <-s.e. [1:nvar]

#-----
i 4 <-i 4+1
} # i 4 loop #.....

glm.beta <-beta0
wavel et. beta <-apply(beta, 2, mean, na.rm=TRUE)
if(plot | graph) ac0 <-acpart(x, y, res0, res0)
if(!plot & !graph) ac0 <-NA
acw <-apply(ac, 2, mean, na.rm=TRUE)
acpw <-apply(acp, 2, mean, na.rm=TRUE)
resw <-apply(resi, 2, mean, na.rm=TRUE)
s.e. <-apply(se, 2, mean, na.rm=TRUE)
lin <-X%*%wavel et. beta
if(family=="gaussian") pi <-lin
if(family=="binomial") pi <-exp(lin)/(1+exp(lin))
if(family=="poisson") pi <-exp(lin)

# test statistics

z. val ue <-rep(NA, nvar)
pr <-rep(NA, nvar)
if(!is.na(wavel et. beta[1])) {
z. val ue <-wavel et. beta/s.e.
for(i in 1:nvar){
if(family=="gaussian"){
if(z. val ue[i] <=0) pr[i] <-2*pt(z. val ue[i], df)
if(z. val ue[i] >0) pr[i] <-2*(1-pt(z. val ue[i], df))
}
if(family=="binomial" | family=="poisson"){
if(z. val ue[i] <=0) pr[i] <-2*pnorm(z. val ue[i])
if(z. val ue[i] >0) pr[i] <-2*(1-pnorm(z. val ue[i]))
}
}
}

beta <-cbind(glm.beta, wavel et. beta, s.e., z. val ue, pr)
beta <-beta[, 2:5]
if(family=="gaussian")
col names(beta) <- c("Estimate", "Std. Err", "t val ue", "Pr(>|t|)")
if(family=="binomial" | family=="poisson")
col names(beta) <- c("Estimate", "Std. Err", "z val ue", "Pr(>|z|)")
cat(" ---", "\n", "Coefficients:", "\n")
printCoefmat(beta)

if(plot){
cat(" ---", "\n")
cat("Autocorrelation for glm. residual s", "\n")
print(ac0)

```

```

WRMcode.txt
cat("Autocorrelation for wavel et. residual s", "\n")
print(acw)
}

if(graph){
y1<-min(min(ac0), min(acw))-.1
y2<-max(max(ac0), max(acw))+.1
plot(ac0, type="b", ylim=c(y1, y2),
      ylab="Autocorrelation of residual s", xlab="Lag distance",
      main=paste("Autocorrelation for level = ", level))
points(acw, pch=2, type="b")
v<-1:2
leg<-c("glm. residual s", "wavel et. residual s")
legend(6, y2-.1, leg, pch=v)
}

coef<-as.vector(wavel et. beta)
names(coef)<-dimnames(X)[[2]]

call<-match.call()
fit<-list(call=call, b=coef, s.e.=s.e., z=z.value, p=pr, fitted=pi, resid=resw,
          w.ac=acpw, ac.glm=ac0, ac=acw)
class(fit)<-"wrm"
fit
}

```

```
#####
```

```
#####
# Function to calculate the parts of autocorrelation for residuals
acpart<-function(x, y, fpart, ftotal){
#####
if(!(length(x)==length(y) & length(x)==length(fpart)
     & length(x)==length(ftotal))) stop("error in acpart")
fpart<-fpart-mean(fpart)
ftotal<-ftotal-mean(ftotal)
mi<-max(x)-min(x)+1
mk<-max(y)-min(y)+1
n<-max(mi, mk)
n2<-n*n
P<-matrix(0, n, n)
T<-matrix(0, n, n)
mask<-matrix(0, n, n)
for(i in 1:length(x)){
kx<-x[i]-min(x)+1
ky<-y[i]-min(y)+1
P[ky, kx]<-fpart[i]
T[ky, kx]<-ftotal[i]
mask[ky, kx]<-1}
filter1<-matrix(0, n, n)
filter1[1, 1]<-1
leng<-length(ftotal)
ne<-convolve(convolve(T, filter1), T)[1, 1]/leng
n3<-3*n
P0<-matrix(0, n3, n3)
P1<-matrix(0, n3, n3)
n1<-n+1
nn<-n+n
P0[n1:nn, n1:nn]<-P[1:n, 1:n]
P1[1:n, 1:n]<-P[1:n, 1:n]

```

```
maske0<-matrix(0, n3, n3)
maske1<-matrix(0, n3, n3)
maske0[n1: nn, n1: nn]<-mask[1: n, 1: n]
maske1[1: n, 1: n]<-mask[1: n, 1: n]
nx<-rep(1: n3, n3)
ny<-as.numeric(gl(n3, n3))
```

```
gr<-1
gr1<-2
h<-n*n3+n+1
corr<-rep(0, 10)
kk<-0
while(kk<10){
  kk<-kk+1
  filter<-matrix(0, n3, n3)
  for(i in 1: (n3*n3)) {
    d<-sqrt((nx[h]-nx[i])^2+(ny[h]-ny[i])^2)
    if(d>=gr & d<gr1) filter[ny[i], nx[i]]<-1}
  sum<-convolve(convolve(maske0, filter), maske1)[1, 1]
  za<-convolve(convolve(P0, filter), P1)[1, 1]/sum
  corr[kk]<-za/ne
  gr<-gr+1
  gr1<-gr1+1
}
corr<-as.vector(corr)
}
```

```
#####
#####
```