Mapping the presence/absence of peat for the northern till plateau with the Generalized Linear Geostatistical Model

Bas Kempen*

April 5, 2012

Contents

*Alterra, part of Wageningen UR (bas.kempen@wur.nl)

Arguments

```
> arguments <- list(
   variables = list(
         primary = "ind", # a 0/1 indicator for the presence/absence of peat
         spatialResolutionIn = 25, #meter
         sub_area = 1,
         year = 2002
   ),
   model = list(
         data_col = 14,
         covar_start = 15,
         covar_end = 64,
         trendModelGLM = ind~veenstat+gtz3mw+reclam33+lgn32+relhoogte1000,
         trendModelGLGM = ~veenstat+gtz3mw+reclam33+lgn32+relhoogte1000,
         familyGLM = "binomial",
         covarList = c("veenstat.txt","gtz3mw.txt","lgn32.txt","reclam33.txt","relhoogte1000.txt")
 ),
   paths = list(
         GRIDDATA_DIR = "D:/PROJECTEN/Veenactualisatie/covariates/Deelgebied1",
         GISDATA_DIR = "D:/PROJECTEN/Veenactualisatie/spatial data/shapefiles",
         WORKING_DIR = "D:/PROJECTEN/Veenactualisatie/Workingdir"
   ),
   extent = list(
         spatial =  "Omtrek_deelgebied1",
         temporal = c(start = "2002-01-01", stop = "2010-12-31")
   ),
   maps = list(
         sa =  "Deelgebieden_dis",  #study areas, i.e. peat soils in the Netherlands
         sa_clip =  "Deelgebieden_clip_dis", # peat areas in a rectangle surrounding study area 1
         prov_nl = "Provinciegrenzen", # provincial boundaries
         prov_sa = "Provinciegrenzen_DG1", # provincial boundaries in study area 1
         rectangle = "Rectangle_DG1", # extent study area
         soilmap = "Deelgebieden", # map of peat soils in the Netherlands
         soilmap_sa = "Bodemkaart50_2006_veen_dis" # peat areas in study area 1
   ),
   tables = list(
         bis_data = "Boringen_gebied1_covariates.csv", # table with BIS data, including covariate values
         legend = "legend_sm50k_dg1.csv", # color legend for the peat map units in study area 1
         lookup_soil = "lookupTable_Soil.csv" # contains attributes that can be joined to the shape file
   ),
   mcmc = list(
      scalepar1 = 0.245,
      scalepar2 = 0.25,
      scalepar3 = 0.25,
      thinning = 50,
      burnin = 25000,
      iterations = 150000,
      tile_size = 2000
   ),
   output = list(
         filename = "result.asc",
         spatialResolution = 100 #m
   )
 )
```

1 Introduction

The aim is to create a map of the presence/absence of peat for the northern till plateau in the Netherlands with the generalized linear geostatistical model (Diggle and Ribeiro Jr., 2007). This map will be used for the spatial prediction of the thickness of the peat layer in a subsequent modelling step. The map has a spatial resolution of 100 m $\times$ 100 m. The period of interest is from 2002-01-01 to 2010-12-31

This document is created by means of 'literate programming' (Knuth, 1984), in particularly the `Sweave` implementation (Leisch, 2002). This means that documentation and source code are weaved to a single document. Literate programming improves communication between scientists and is a huge step forwards in reproducible research.

2 Methodology

To create a map, we need to predict target variable $y(x_0)$ (in our case the presence or absence of a peat layer in the soil profile) at unvisited locations x_0. A presence/absence variable is a categorical (binary) random variable with two outcomes: the target property is either present or absent at a site. The nature of such variable makes that it cannot be modelled with standard geostatistical methods since these methods assume that the random variable is continuous, Gaussian-distributed variables.

The generalized linear geostatistical model (GLGM) is central to the framework of model-based geostatistics (Diggle et al., 1998) and can be used to model and predict non-Gaussian distributed spatial data such as the presence/absence of peat in a soil profile. The GLGM has three components (Diggle and Ribeiro Jr., 2007). The first component is the *signal process* $S(\cdot)$, which is a real-valued Gaussian spatial process with $E[S(\mathbf{x})] = m$, $\text{var}[S(\mathbf{x})] = \sigma^2$, and correlation function $\rho(\mathbf{h})$. Here m is a spatial trend $\mathbf{d}(\mathbf{x}_i)^{\mathrm{T}}\beta$, where $\mathbf{d}(\mathbf{x}_i)$ is a vector of explanatory variables at spatial location $\mathbf{x}_i$ and β is a vector of trend coefficients. The second component is the *measurement process* $Y(\cdot)$. Realizations of this process are the observed data $y(\mathbf{x}_1), \ldots, y(\mathbf{x}_n)$, which are considered (indirect) measurements of the signal process. The $Y(\mathbf{x}_i)$ are assumed to follow a common distributional family (e.g. Bernoulli, Poisson, binomial or Gaussian), depending on the mechanism that generated the data, and are mutually independent conditional on the signal. The responses have conditional mean $E[Y(\mathbf{x}_i)|S(\cdot)]$. The third component is the *link function* $g(\cdot)$, which links the conditional mean $E[Y(\mathbf{x}_i)|S(\cdot)]$ to the linear predictor $S(\mathbf{x}_i)$. The GLGM is thus defined as:

$$g(E[Y(\mathbf{x}_i)|S(\cdot)]) = S(\mathbf{x}_i) = \mathbf{d}(\mathbf{x}_i)^{\mathrm{T}}\beta + U(\mathbf{x}_i) \;, \tag{1}$$

where $U(\mathbf{x}_i)$ is a second-order stationary, Gaussian distributed, spatial process with zero mean and variance σ^2.

A suitable candidate distribution to model the presence/absence of peat is the Bernoulli distribution. The conditional mean $E[Y(\mathbf{x}_i)|S(\cdot)]$ then represents the probability of peat being present at location x_i. A Bernoulli-distributed random variable Y can be modelled with a GLGM (Diggle et al., 2002; Ben-Ahmed et al., 2010; Kempen et al., 2012) with a logit link function. The GLGM of Eq. 1 can then be written as:

$$g(E[Y(\mathbf{x}_i)|S(\cdot)]) = logit(\pi_i) = log(\frac{\pi_i}{1-\pi_i}) = S(\mathbf{x}_i) = \mathbf{d}(\mathbf{x}_i)^{\mathrm{T}}\beta + U(\mathbf{x}_i) \;, \tag{2}$$

where π_i is the probability of the presence of peat at sampling location x_i.

The observations $y(\mathbf{x}_i)$ are obtained from the Dutch soil information system *BIS*. Each observation is checked for the presence of a peat layer in the soil profile. If peat is present at a sampling site, then the observation observation gets value 1 and 0 otherwise.

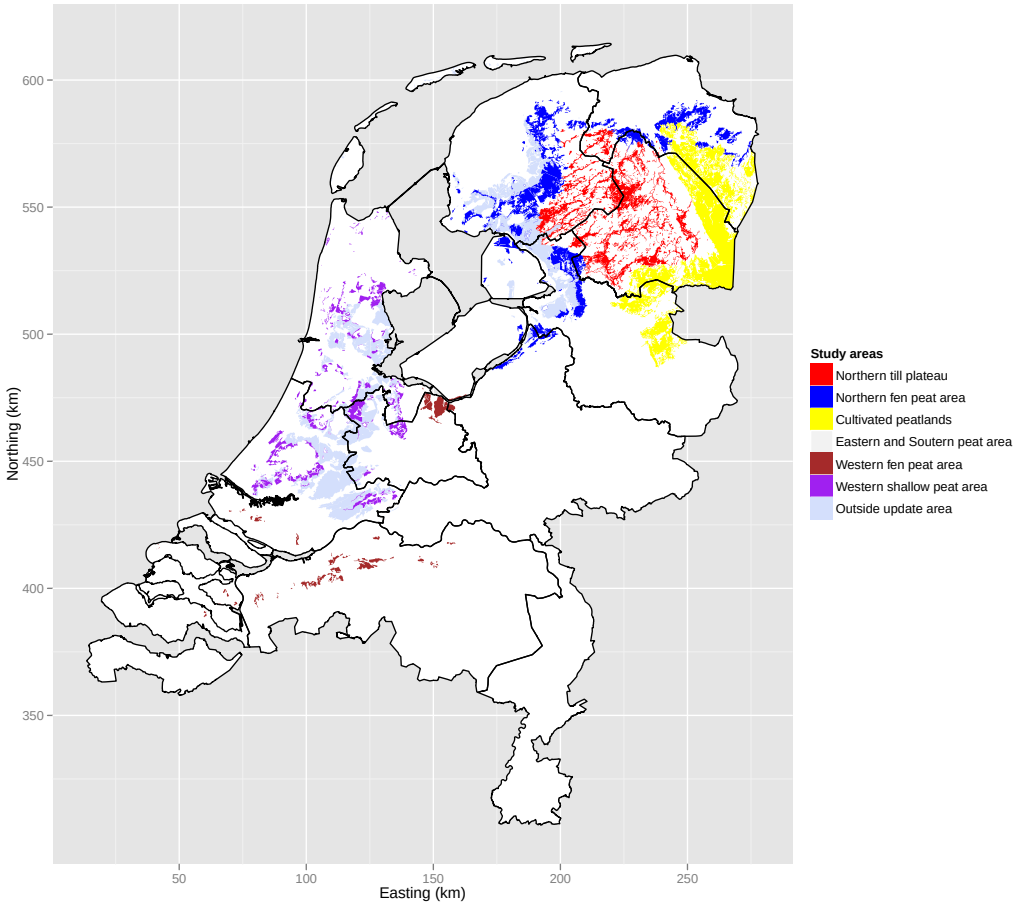
Estimation of the model parameters of the GLGM as well as spatial prediction with this model is complex. It involves repetitive use of Markov Chain Monte Carlo methods (Minasny et al., 2011) to obtain simulations of the unobserved signal process $S(\cdot)$ given the observations $y(\mathbf{x}_i)$ and Monte Carlo maximum likelihood estimation of the model parameters (Christensen, 2004). We refer to Kempen et al. (2012) for an exposition of these methods in the context of digital soil mapping with the GLGM.

3 Data preparation

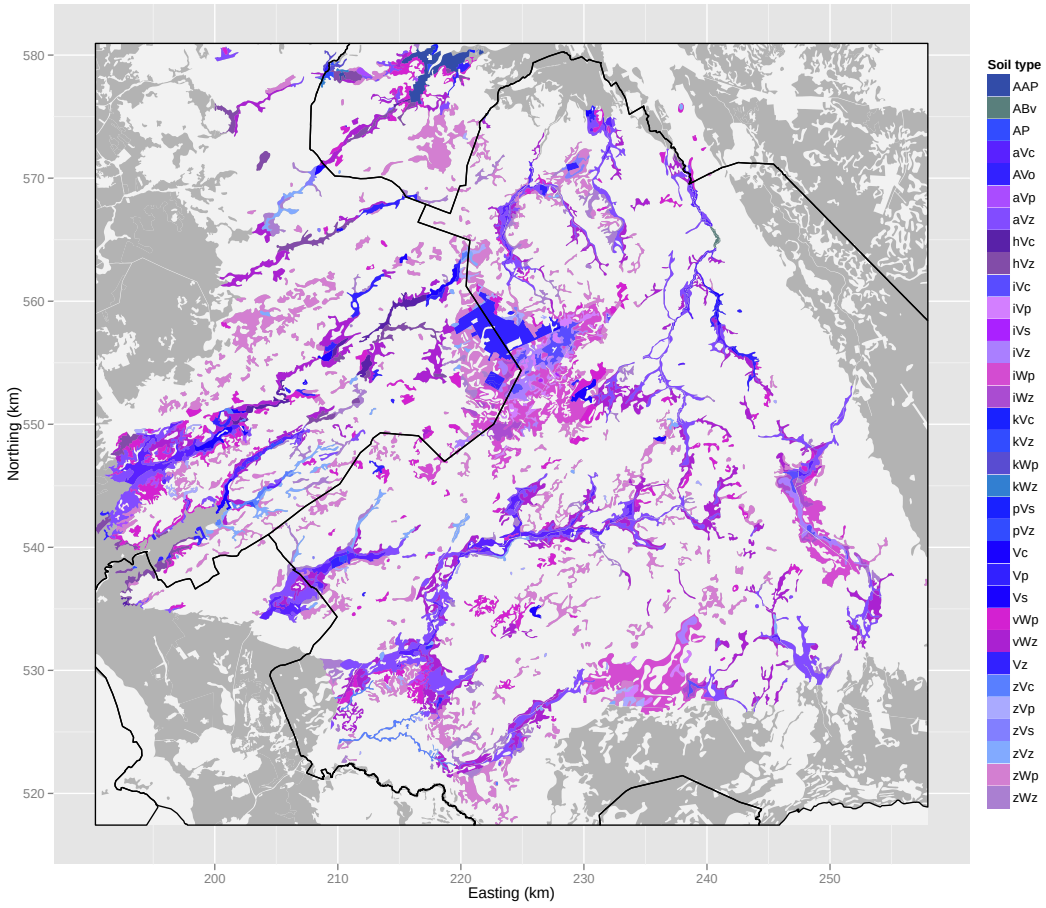
3.1 Study area

The figure below shows the extent of the peat soils in the Netherlands, according to the national soil map at scale 1:50 000. The area with peat soils has been divided into 6 sub-areas. Each area will be updated individually. The area with deep peat soils in the northern and western fen meadow landscape will not be updated. This study focusses on sub-area 1: the northern till plateau. The sizes of the six mapping areas are given in the following Table.

area	ha
Northern till plateau	67904
Northern fen peat area	83927
Cultivated peatlands	117420
Eastern and Soutern peat area	43246
Western fen peat area	14144
Western shallow peat area	38161
Outside update area	162253



The 67904 ha study area comprises the peat soils of the glacial till plateau in the northern part of the Netherlands. The till plateau is dissected by a system of brook valleys that are filled with fen peat. Remains of once vast highmoor bogs on the plateau are now reclaimed for agriculture. The figure below shows the 1:50 000 national soil map for the peat areas in study area and the table reports the areas of the individual soil types. Deep peat soils (peat layer $>$ 1.20 m thick) cover 7600 ha, shallow peat soils (peat layer 0.40–1.20 m thick) 19615 ha, and peaty soils (peat layer $<$ 0.40 m thick) 40687 ha.



3.2 Primary data

Field data on the presence/absence of peat were retrieved from the Dutch Soil Information System BIS (de Vries et al., 2008). In addition, field data collected during the 2002–2004 assessment of the status of peat soils in the Pleistocene part of the Netherlands were used (van Kekem et al., 2005). These data were digitized from field maps for this project. Data from both sources were combined into one database. A indicator variable was created, which takes value 1 if peat is present in the soil profile (here we used a minimum thickness of 5 cm in order to be classified as 'present'), and 0 otherwise. Soil profile descriptions were not available for the peat assessment data set. For these data points, presence/absence of peat was determined from the soil classification code (STPC).

```
> d <- read.csv(
      file = file.path(arguments$paths$WORKING_DIR, arguments$tables$bis_data),
      header = TRUE,
      as.is = TRUE
  )
```

soil	ha	description
AAP	754	Aangemaakte petgaten
ABv	39	Venige beekdalgronden
AP	127	Petgaten
aVc	2975	Madeveengronden op zeggeveen; rietzeggeveen of broekveen
AVo	1542	Veen in ontginning
aVp	265	Madeveengronden op zand met humuspodzol; beginnend ondieper dan 120 cm
aVz	10089	Madeveengronden op zand zonder humuspodzol; beginnend ondieper dan 120 cm
hVc	628	Koopveengronden op zeggeveen; rietzeggeveen of (mesotroof) broekveen
hVz	1914	Koopveengronden op zand; beginnend ondieper dan 120 cm
iVc	719	Veengronden met een veenkoloniaal dek op zeggeveen; rietzeggeveen of moerasbosveen
iVp	1446	Veengronden met een veenkoloniaal dek op zand met humuspodzol; beginnend ondieper dan 120 cm
iVs	18	Veengronden met een veenkoloniaal dek op veenmosveen
iVz	1431	Veengronden met een veenkoloniaal dek op zand zonder humuspodzol; beginnend ondieper dan 120 cm
iWp	6017	Moerige podzolgronden met een veenkoloniaal dek en een moerige tussenlaag
iWz	790	Moerige eerdgronden met een veenkoloniaal dek en een moerige tussenlaag op zand
kVc	23	Waardveengronden op zeggeveen; rietzeggeveen of (mesotroof) broekveen
kVz	49	Waardveengronden op zand; beginnend ondieper dan 120 cm
kWp	67	Moerige podzolgronden met een zavel- of een kleidek en een moerige tussenlaag
kWz	40	Moerige eerdgronden met een zavel- of kleidek en een moerige tussenlaag op zand
pVs	9	Weideveengronden op veenmosveen
pVz	18	Weideveengronden op zand; beginnend ondieper dan 120 cm
Vc	516	Vlierveengronden op zeggeveen; rietzeggeveen of (mesotroof) broekveen
Vp	257	Vlierveengronden op zand met humuspodzol; beginnend ondieper dan 120 cm
Vs	417	Vlierveengronden op veenmosveen
vWp	5710	Moerige podzolgronden met een moerige bovengrond
vWz	10154	Moerige eerdgronden met een moerige bovengrond op zand
Vz	1234	Vlierveengronden op zand zonder humuspodzol; beginnend ondieper dan 120 cm
zVc	505	Meerveengronden op zeggeveen. rietzeggeveen of broekveen
zVp	719	Meerveengronden op zand met humuspodzol; beginnend ondieper dan 120 cm
zVs	244	Meerveengronden op veenmosveen
zVz	1306	Meerveengronden op zand zonder humuspodzol; beginnend ondieper dan 120 cm
zWp	15152	Moerige podzolgronden met een humushoudend zanddek en een moerige tussenlaag
zWz	2714	Moerige eerdgronden met een zanddek en een moerige tussenlaag op zand

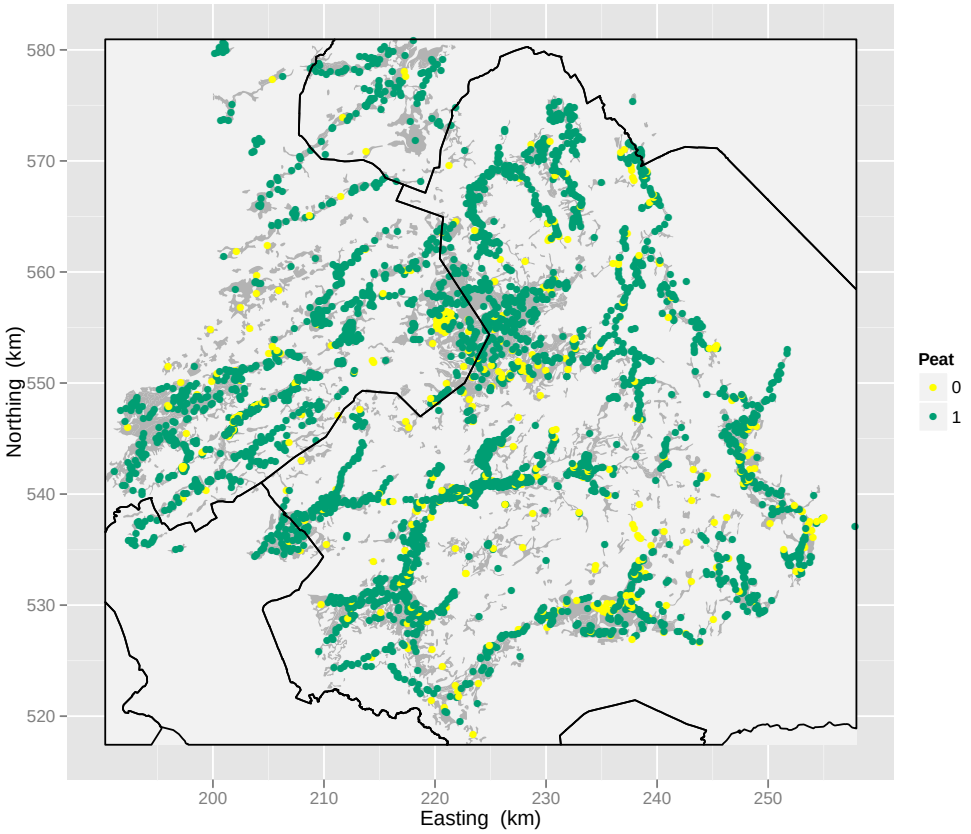
From this dataset, the observation sites located in sub-area 1 were selected. Next, point observations dating before 2002 were excluded, as well as observations collected during 1:10 000 soil surveys, and observations for which the presence of peat could not be determined because of incomplete soil profile data. Finally, point observations which lacked values for the explanatory variables were also excluded.

```
> # select data
> d <- subset(x = d,
              subset = (dg == arguments$variables$sub_area & jaar >= arguments$variables$year
                        & bron!="BPKDETAIL")
 )
> # exclude indicators with value NA (presence/absence of peat is unknown)
> d <- d[!is.na(d[,arguments$variables$primary]),]
> # remove data points with NA for explanatory variables
> d <- d[!is.na(d$lgn5),]
```

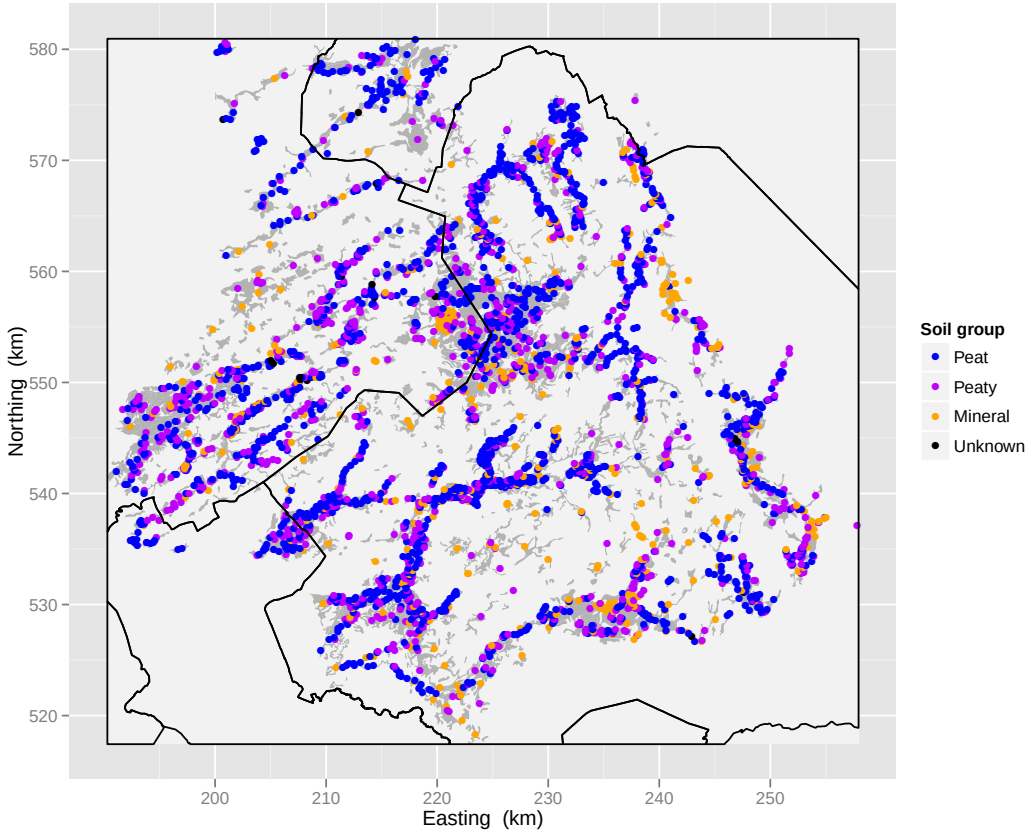
The result is a `data.frame` consisting of 3306 rows. A snippet of this `data.frame` is given below.

The figure below shows the locations of the point observations in sub-area 1 and the presence (1) or absence (0) of peat. A peat layer is present at 2851 locations and absent at 455 locations.

id	x	y	bron	jaar	stpc	begin	eind	dikte	bod50	hoofd	dg	cens	ind
553	222263	519530	BPK	2011	4k	60	85	25	zW	M	1	0	1
871	223812	521097	BPK	2011	2m	30	45	15	zW	W	1	0	1
1058	221400	521760	BPK	2007	z4d	15	20	5	zW	W	1	0	1
1243	220232	522503	BPK	2011	4h	50	110	60	zW	M	1	0	1
1320	218552	522835	BPK	2008	1t	35	80	45	zV	V	1	0	1
:	:	:	:	:	:	:	:	:	:	:	:	:	
2495	215453	580266	vc	2004	1h	0	90	90	aV	V	1	0	1
2496	201100	580478	vc	2004	2n	0	20	20	aV	W	1	1	1
2497	200919	580651	vc	2004	2n	0	20	20	aV	W	1	1	1
2498	218058	580864	vc	2004	1s	30	110	110	kV	V	1	1	1



The figure below shows the main observed soil group (peat, peaty or mineral) at the sampling sites.



The point data of the 2002-2004 peat soil assessment were not used for the calibration of the prediction model for mapping the presence/absence of peat (section 4). These data were, however, used for the application of the prediction model (section 5).

3.3 Explanatory variables

A total of 50 data layers with environmental explanatory variables were derived from maps of soil, groundwater, (historic) land cover, elevation, and geomorphology, and stored in a geodatabase. These data layers may assist spatial prediction of the presence/absence of the peat layer at unvisited locations. All data layers were in raster format with 25-m spatial resolution.

Seven of these variables were derived from the national 1:50 000 soil map, representing: peat thickness class (3), topsoil lithology (2), peat type, and peat status (van Kekem et al., 2005). Eight variables were derived from the MIPWA groundwater maps: mean highest water table (2), mean lowest water table (2), drainage condition, summer drainage condition, winter drainage condition, and seepage. A map representing oxidiation sensitivity was derived from peat type and groundwater table class maps according to (Finke et al., 1996). Nine maps representing land cover class were derived from the LGN3+, LGN4 and LGN5 land cover maps (Hazeu, 2005), each with a different number and combination of land cover classes. Six maps representing reclamation age were derived from historic land cover layers HGN1900 (Knol et al., 2004), Bosstatistiek1940

(Clement and Kooistra, 2003), HGN1960, HGN1980, HGN1990 (Knol et al., 2003) and LGN5. Ten maps respresenting relative elevation were derived from the 25-m digital elevation model[1] (DEM). Relative elevation captures local relief and is computed by subtracting the local mean elevation (determined within circles with 250 m, 500 m, 750 m, and 1000 m search radii). The relative elevation layer based on the 750-m search radius was reclassified into layers with two, three and four classes. In addition, the DEM was used, in combination with a layer with historic elevation —constructed by inverse distance weighted interpolation of a network of elevation measurements from the 1960s (1.2 ha^{-1})— to obtain a layer representing elevation change. Elevation change is informative because peat excavation and decomposition lower the surface (Hoogland et al., 2012). The layer was subsequently reclassified into layers with two, three and five classes. The elevation-change layer was also used to determine the mean elevation change for each of the delineations of the 1:50 000 soil map for the peat areas, yielding an additional four data layers. Finally, a map representing geomorphology (Koomen and Maas, 2004) was used to determine the main landform type for each of the soil map delineations.

4 Modelling

4.1 Model selection

A manual step-wise approach was used to select explanatory variables for the trend component of the GLGM. A univariate generalized linear model (GLM; i.e. a non-spatial GLGM) with a logit link function was fitted for each variable to assess the strength of the correlation with the target variable; the presence/absence indicator. Because explanatory variables with each variable group (soil, groundwater, land cover, reclamation age, relative elevation and elevation change) are strongly correlated, only one variable was selected from each group on basis of the Akaike Information Criterion (AIC) (Webster and McBratney, 1989) for further analysis. Next, the selected variables were sequentially added to a multivariate model in order of the strength of the univariate correlation. Again, the AIC was used to select the most parsimonious multivariate models.

A summary of the selected trend model is given below. Five explanatory variables were selected: peat status ('veenstat', 4 classes: (1) deformed peat soil, (2) non-deformed peat soil, (3) peaty soil, (4) no information on peat status); summer drainage condition ('gtz3mw', 3 classes: (1) good, (2) moderate, (3) poor); reclamation age ('reclam33', 3 classes: (1) $>$ 70 years, (2) 40–70 years, (3) $<$ 40 years); land cover ('lgn32', 3 classes: (1) grassland, (2) cropland, (3) natural vegetation); and relative elevation based on a 1 000 m search radius ('relhoogte1000'). All model coefficients except one, were significant at the $p = 0.05$-level.

```
> glmPeat <- glm(
      formula = arguments$model$trendModelGLM,
      family = arguments$model$familyGLM,
      data = d
 )
> summary(glmPeat)

Call:
glm(formula = arguments$model$trendModelGLM, family = arguments$model$familyGLM,
    data = d)
```

[1] www.ahn.nl

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4281  -0.8032   0.3399   0.7027   2.1076

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    1.480907   0.296339   4.997 5.81e-07 ***
veenstat2      0.764698   0.365279   2.093 0.036308 *
veenstat3     -0.726507   0.272700  -2.664 0.007719 **
veenstat4      2.621892   1.048371   2.501 0.012387 *
gtz3mw2       -1.052415   0.223060  -4.718 2.38e-06 ***
gtz3mw3       -1.627509   0.277716  -5.860 4.62e-09 ***
reclam332      0.554475   0.315663   1.757 0.078996 .
reclam333      2.093640   0.508777   4.115 3.87e-05 ***
lgn322        -0.717223   0.203285  -3.528 0.000418 ***
lgn323        -0.781394   0.383027  -2.040 0.041345 *
relhoogte1000 -0.007219   0.001968  -3.668 0.000245 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1053.78  on 814  degrees of freedom
Residual deviance:  755.16  on 804  degrees of freedom
AIC: 777.16

Number of Fisher Scoring iterations: 6
```

An analysis of deviance assesses whether an individual variable makes a significant contribution to the model. This test is also known as the likelihood ratio test (Hosmer and Lemeshow, 2000). The likelihood ratio statistic measures the reduction in deviance that results from including an explanatory variable (for a linear model the deviance is equal to the sum of squares). Under the null hypothesis that a variable does not make a significant contribution the likelihood ratio statistic will be chi-square distributed. The analysis of deviance shows that each variable makes a significant contribution to the model. The variable 'peat status' contributes most 57% to the reduction in deviance and is thus the strongest explanatory variable.

```
> anova(glmPeat, test="Chi")

Analysis of Deviance Table

Model: binomial, link: logit

Response: ind

Terms added sequentially (first to last)


             Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                           814    1053.78
veenstat      3  171.155       811     882.63 < 2.2e-16 ***
gtz3mw        2   74.069       809     808.56 < 2.2e-16 ***
reclam33      2   25.170       807     783.39 3.422e-06 ***
lgn32         2   13.876       805     769.51 0.0009701 ***
relhoogte1000 1   14.355       804     755.16 0.0001514 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

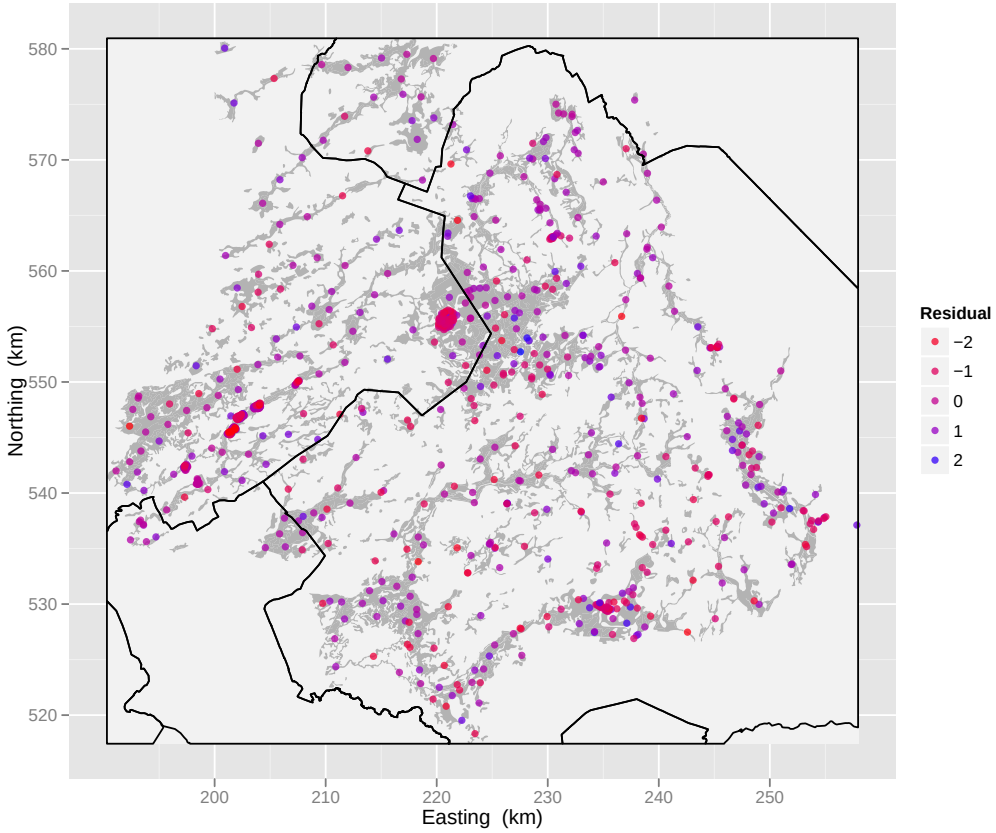
The fitted values (probability of peat being present at a data point) and the deviance residuals are added to the `data.frame` containing the data points.

```
> # fitted values
> d$fitted<-glmPeat$fitted.values
> # deviance residuals
> d$r <- resid(glmPeat, type = "deviance")
```

The estimated model coefficients are stored in a separate object and will be used later on.

```
> betas<-as.numeric(glmPeat$coefficients)
```

A plot of the deviance residuals at the data points is shown below.

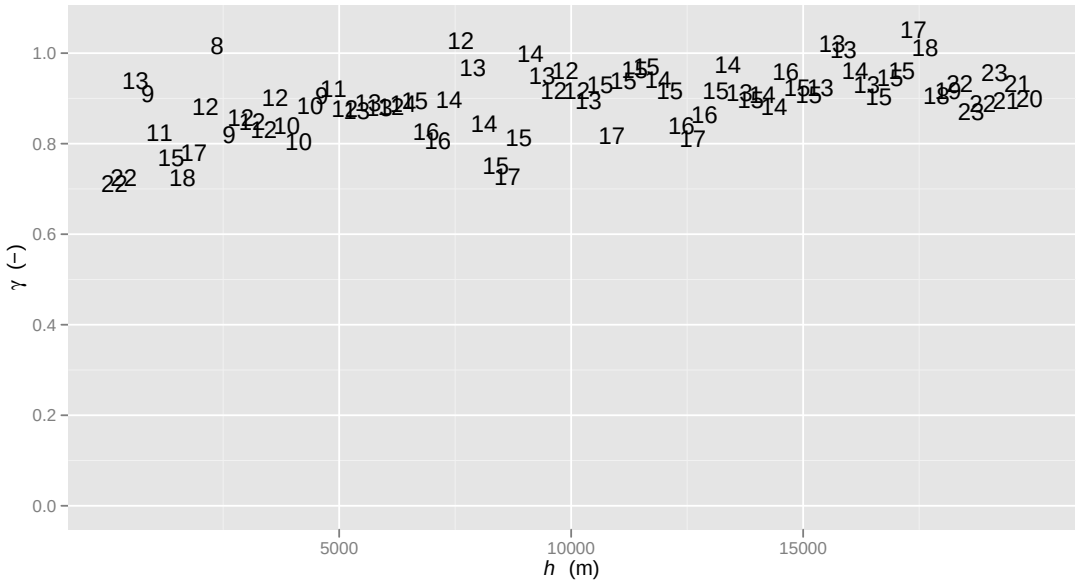


4.2 Variography

The sample semivariogram of the deviance residuals r will be estimated by means of the `gstat`-package (Pebesma, 2004).

```
> coordinates(d) <- ~x+y
> #remove duplicate locations
> zerodist(d, zero = 0.0, unique.ID = FALSE)
> d<-remove.duplicates(d, zero = 0.0, remove.second = TRUE)
> # fit sample semivariogram
> sampleSemivariogram <- variogram (r ~ 1, data = d, width = 250, cutoff = 20000)
```

The sample semivariogram is given in the figure below. The semivariances are given by numbers indicating the number of point pairs ($\times 0.01$) used to estimate the semivariances. The semivariances are somewhat increasing, also for large lag distances h, indicating the presence of a weak trend.



To obtain semivariances for lag distances not available in the data set, a model has to be fitted to the sample semivariogram. Models have to be selected with care since not all models guarantee a unique solution of the kriging system. See Goovaerts (1997) or the `gstat`-documentation for a list of permissible models. In this report, an exponential model has been used to fit the data.

```
> semivariogramModel <- vgm(
        psill = 0.3,
        model = "Exp",
        range = 5000,
        nugget = 0.6
 )
> semivariogramModel <- fit.variogram(sampleSemivariogram, model = semivariogramModel)
```

The semivariogram model has been fitted by means of weighted ordinary least squares, with weights proportional to the number of point pairs and inversely proportional to the squared lag distance (default setting). The estimated parameters of the exponential model are given in the table below.

model	psill	range	kappa	ang1	ang2	ang3	anis1	anis2
Nug	0.66	0.00	0.00	0.00	0.00	0.00	1.00	1.00
Exp	0.21	499.42	0.50	0.00	0.00	0.00	1.00	1.00

The estimated semivariogram parameters, the partial sill, nugget and range, are used as initial values for parameter estimation by residual maximum likelihood (REML) (Lark and Cullis, 2004). Estimation by REML has the advantage that full use is made of the available data, i.e. it is not necessary to group the data by lag distance in order to estimate the semivariogram model

such as with weighted ordinary least squares. Furthermore, it is a more appropriate method for parameter inference from spatial data that are obtained by a non-design-based sampling strategy. REML estimation was done by means of the `geoR`-package (Ribeiro Jr. and Diggle, 2001).

First, the `data.frame` containing the soil point observations must be converted to class `geodata`. This is done twice. Once to store the residuals, and once to store the target variable that indicates presence or absence of peat in the soil profile.

```
> dGDresid<-as.geodata(
                  obj = as.data.frame(d),
                  header = TRUE,
                  coords.col = 2:3,
                  data.col = arguments$model$covar_end+2,
                  data.names = NULL,
                  covar.col = arguments$model$covar_start:arguments$model$covar_end
 )
> dGDind<-as.geodata(
                  obj = as.data.frame(d),
                  header = TRUE,
                  coords.col = 2:3,
                  data.col = arguments$model$data_col,
                  data.names = NULL,
                  covar.col = arguments$model$covar_start:arguments$model$covar_end
 )
> dGDind$data<-as.numeric(dGDind$data)
```

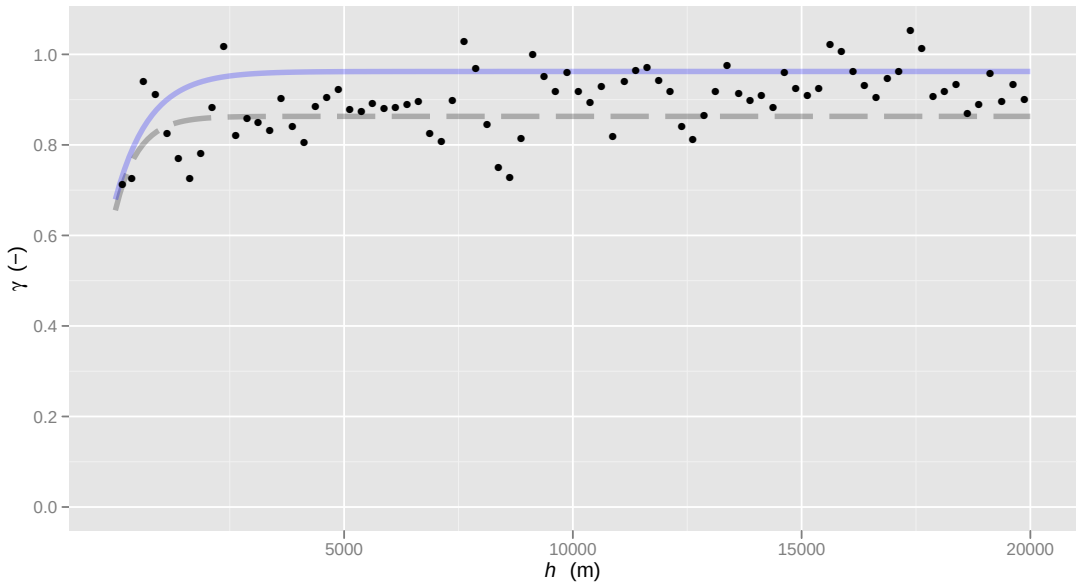
Next, the variogram parameters are estimated by REML with the `likfit` function.

```
> vgmREML <- likfit(
               geodata = dGDresid,
               trend="cte",
               cov.model="exponential",
               ini.cov.pars=c(semivariogramModel[2,2], semivariogramModel[2,3]),
               nugget=semivariogramModel[1,2],
               lik.method="REML"
 )
```

The REML estimates of the variogram parameters are given below. The parameter τ^2 (tausq) is the nugget, σ^2 (sigmasq) is the partial sill, and ϕ (phi) is the range.

```
> vgmREML

likfit: estimated model parameters:
      beta      tausq    sigmasq        phi
"  0.1054" "  0.6795" "  0.2827" "746.0299"
Practical Range with cor=0.05 for asymptotic range: 2234.906

likfit: maximised log-likelihood = -1106
```

The figure below shows the sample semivariogram (dots), the variogram model estimated by weighted ordinary least squares (dashed line) and the variogram model estimated by REML (solid line).



4.3 MCMC simulation

So far, we have selected a set of explanatory variables for the trend part of the prediction model ($\mathbf{d}(\mathbf{x}_i)$ in Eq. 1), and we obtained estimates of the model coefficients of these parameters with the GLM (β in Eq. 1), and estimates of the variance parameters of the spatial component of the prediction model ($U(\mathbf{x}_i)$ in Eq. 1). The latter were estimated by REML, using initial values as input that were estimated by weighted ordinary least squares from the sample semivariogram.

Now the estimated model coefficients and variance parameters are used to obtain values of the unobserved Gaussian spatial process $S(\cdot)$, from which our observations y_i (presence or absence of the peat layer) are realizations. For this purpose we use a technique called *Markov Chain Monte Carlo* (MCMC) simulation. MCMC is a general-purpose technique for simulating from complex probability distributions. It constructs a Markov chain, which has the desired distribution as its equilibrium distribution. Simulation from the chain after equilibrium has been attained, yields a sample from the target distribution, which is in our case the distribution of $S(\cdot)$ at an observation site i. Here MCMC is used to simulate samples (realizations) of S_i conditional on the observations y_i. MCMC simulation will be done with the `geoRglm`-package (Christensen and Ribeiro Jr., 2002).

The first step in MCMC simulation is to define the options for the MCMC algorithm. This is done with the `mcmc.control`-function. The *S.scale* parameter scales the proposal distribution and affects the fraction of proposals that are accepted. This fraction should be around 0.60, which is considered an optimal value for the simulation algorithm that is used (Christensen and Ribeiro Jr., 2002). The S.scale parameter is determined by trial-and-error. The parameter *n.iter* indicates the length of the chain, i.e. number of iterations (or simulations of $S(\cdot)$). The parameter *thin* indicates the sub-sampling rate. A chain is typically thinned to reduce autocorrelation between simulations since simulations should be mutually independent. Finally, the parameter *burn.in* is the length of the burn-in period. This is the number of iterations until

the chain approaches equilibrium. Burn-in samples are discarded.

Here the value 0.245 is taken for S.scale. The burn-in length is 25000 iterations and the chain length is 150000. We sample and store every 50^{th} interation, giving us 3000 simulations of $S(\cdot)$ at each sampling location x_i.

```
> mcmcSet <- mcmc.control(
       S.scale = arguments$mcmc$scalepar1,
       n.iter = arguments$mcmc$iterations,
       thin = arguments$mcmc$thinning,
       burn.in = arguments$mcmc$burnin
 )
```

The second step is the specification of the GLGM in form of a `list`. This specification includes the model family, the trend model, the theoretical semivariogram model, and initial values of the model parameters (the coefficients and variance parameters).

```
> glgm <- list(
       family=arguments$model$familyGLM,
       trend = trend.spatial(trend = arguments$model$trendModelGLGM, geodata = dGDind),
       cov.model="exponential",
       cov.pars=c(vgmREML$sigmasq, vgmREML$phi),
       nugget = vgmREML$tausq,
       beta=betas
 )
```

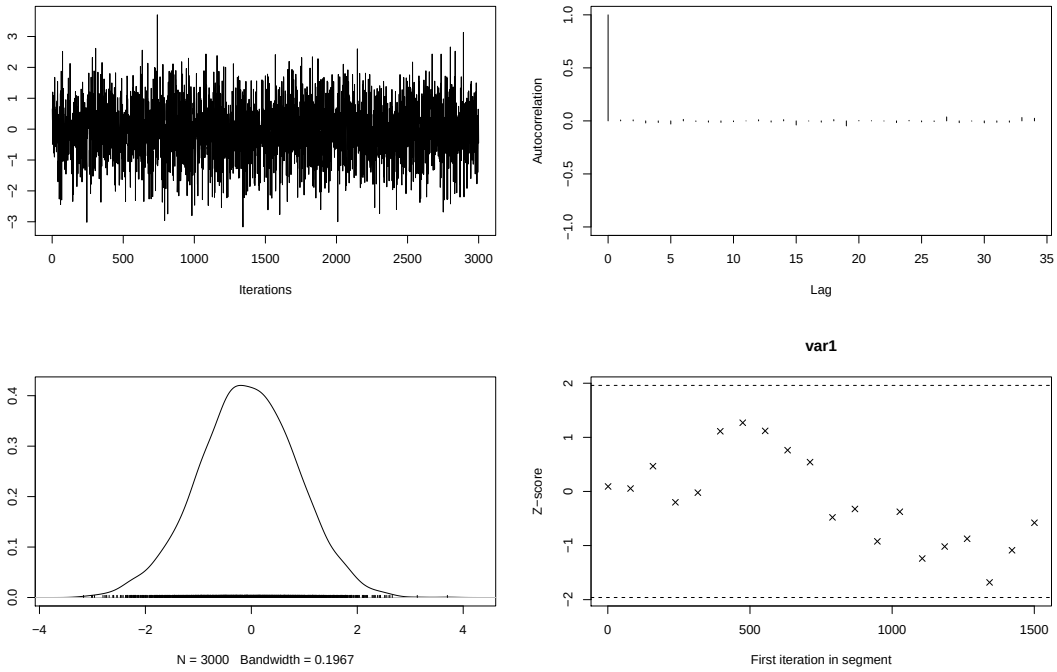
The third step is the simulation of the s_i at the sampling locations x_i. This is done with the `glsm.mcmc`-function.

```
> simFtilde <- glsm.mcmc(
       geodata = dGDind,
       units.m = "default",
       model = glgm,
       mcmc.input = mcmcSet,
       messages=TRUE
 )
```

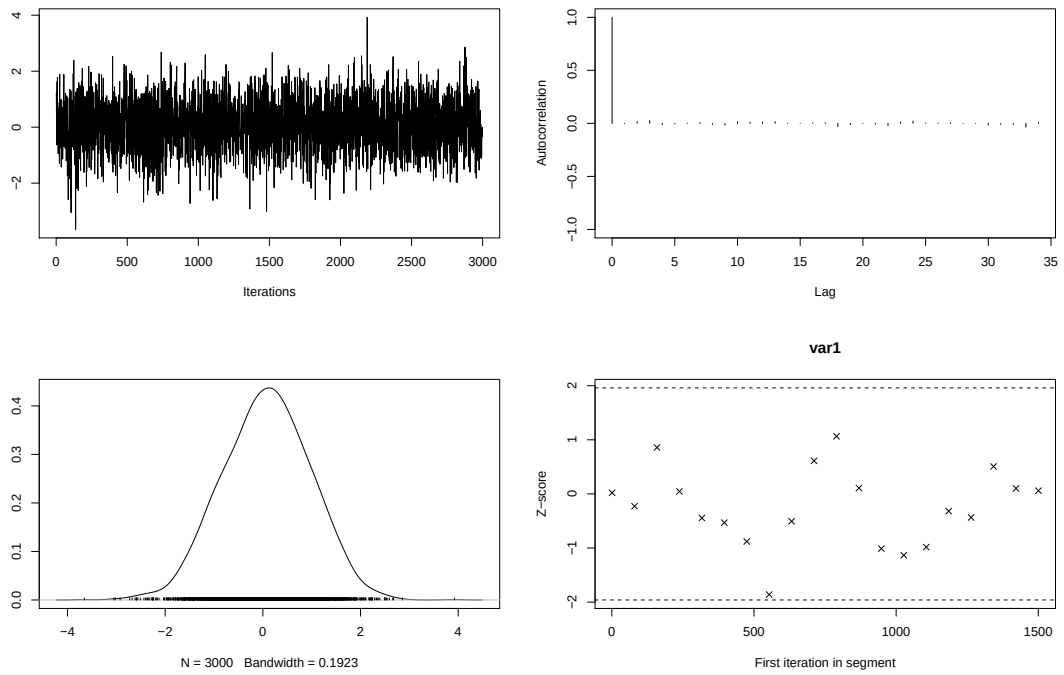
Like mentioned before, it is assumed that after a burn-in period the Markov chain approaches or converges to the equilibrium distribution. To make sure the chain has indeed reached its equilibrium, we assess the convergence of the chain using a *trace plot* and the Geweke's statistic (Geweke, 1992). Geweke's statistic compares the means of the simulated values for two non-overlapping parts of the Markov chain. This statistic is a Z-score and follows a standard normal distribution, $N(0,1)$, under the null hypothesis (there is no difference between the means for two parts of the Markov chain) is true. If a chain has converged, then the mean (and variance) of the simulated values from the first part of the chain will be equal to the mean from a later part of the chain. Large absolute values of the statistic indicate rejection of the null-hypothesis, and thus non-convergence.

In addition, we assess the mixing and autocorrelation of the chain. A chain is considered 'well mixing' if it fully explores the parameter space (in our case the space with all possible values for $S(\cdot)$), whereas a 'poorly mixing' chain remains in small regions of the parameter space. Mixing can also be judged from a trace plot. An autocorrelation plot informs us if the simulations are mutually independent. The convergence, mixing and autocorrelation properties are assessed with the `coda`-package. Below, plots are displayed for two randomly chosen sampling locations.

```
> chainConv1 <- create.mcmc.coda(
   x = simFtilde$simulations[round(runif(1, min = 1, max = nrow(simFtilde$simulations)),0),],
   mcmc.input = list(S.scale=arguments$mcmc$scalepar1,thin=1)
 )
> geweke.diag(chainConv1)

Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5

   var1
0.09195
```




```
> chainConv2 <- create.mcmc.coda(
   x = simFtilde$simulations[round(runif(1, min = 1, max = nrow(simFtilde$simulations)),0),],
   mcmc.input = list(S.scale=arguments$mcmc$scalepar1,thin=1)
   )
> geweke.diag(chainConv2)

Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5

   var1
0.02139
```



The top-left plots are the trace plots. These plots show the iterations versus the sampled values of s_i, after the burn-in period. The centre of the first chain is around the value -0.1 with small fluctuations, whereas the second chain is around the value 0.1. Both plots indicate that the chain could have reached the equilibrium distribution, i.e., the chain has converged. This is confirmed by Geweke's statistics. The Geweke's Z-scores are much smaller than ± 1.96; the 95% confidence interval for the standard normal distribution $N(0, 1)$. Furthermore, the Geweke's plots (bottom-right) show that the Z-scores nearly fall all within the 95% confidence interval, indicated by the two horizontal lines. These plots show what happens to the Z-scores when successively larger numbers of iterations are discarded from the beginning of the chain. The mixing of the chain seems quite good. The chain traverses the distribution quickly and visits regions in the parameter space with low density. The autocorrelation plots (top-right) show that the MCMC samples are mutually uncorrelated. This means that the thinning of the chains was sufficient. The density plots (bottom-left) show that the simulated values of s_i approximately follow a normal distribution.

4.4 Monte Carlo maximum likelihood estimation

Next, the simulated s_i are used to estimate the model parameters by maximum likelihood. If this is repeated a sufficiently large number of times (i.e., for each MCMC sample) the proper conditional distributions of the signal and its parameters are reconstructed. This method for parameter estimation is referred to as Monte Carlo (MC) maximum likelihood (Christensen, 2004).

```
> mcmlPrep <- prepare.likfit.glsm(simFtilde)
```

```
> mcmlEstimation <- likfit.glsm(
       mcmc.obj = mcmlPrep,
       trend = trend.spatial(trend = arguments$model$trendModelGLGM, geodata = dGDind),
       cov.model = "exponential",
       ini.phi = vgmREML$phi,
       nugget.rel = (vgmREML$tausq/vgmREML$sigmasq),
       fix.nugget.rel = FALSE,
       messages = FALSE
 )
> mcmlEstimation

likfit.glsm: estimated model parameters:
     beta0      beta1      beta2      beta3      beta4      beta5      beta6
"  1.6275" "  0.7294" " -0.9986" "  2.5287" " -1.3090" " -1.7981" "  0.4259"
     beta7      beta8      beta9     beta10    sigmasq        phi  tausq.rel
"  2.4827" " -0.4589" " -0.7444" " -0.0071" "  0.1897" "719.5982" "  3.2323"

 likfit.glsm : maximised log-likelihood = 14.24
```

Christensen (2004) recommends to repeat parameter estimation, where the new initial parameter values are chosen as the values maximising the We follow this recommendation here. We repeat MCMC simulation of $S(\cdot)$ using the MC maximum likelihood estimates of the model parameters and then use the new simulated values to repeat MC maximum likelihood estimation of the model parameters. This gives the following estimates of the model parameters.

```
likfit.glsm: estimated model parameters:
     beta0      beta1      beta2      beta3      beta4      beta5      beta6
"  1.8412" "  0.7953" " -0.9194" "  2.5234" " -1.4041" " -2.1088" "  0.8971"
     beta7      beta8      beta9     beta10    sigmasq        phi  tausq.rel
"  2.7517" " -0.5380" " -0.6921" " -0.0058" "  0.2762" "571.9009" "  1.9054"

 likfit.glsm : maximised log-likelihood = 13.05
```

Now that we have obtained the final estimates of the model parameters, we use these once more to simulate values of $S(\cdot)$ at the sampling sites using MCMC simulation. Now we do not only simulate at the sampling sites that were used to calibrate the prediction model, but also at the sites that were not used for model calibration (these are the sampling sites of the 2002-2004 peat soil assessment). Again, the point data set is read and processed as before with the only difference that the peat assessment points are not excluded, and a `geodata` object is created that is used for MCMC simulation.

```
> mcmcSet <- mcmc.control(
       S.scale = arguments$mcmc$scalepar3,
       thin = arguments$mcmc$thinning,
       n.iter = arguments$mcmc$iterations,
       burn.in = arguments$mcmc$burnin
 )
> set.seed(16850321)
> simF<-glsm.mcmc(
       geodata = dGDind,
       units.m = "default",
       model = mcmlEstimation2,
       mcmc.input = mcmcSet,
       messages = FALSE
 )
```

5 Mapping the presence/absence of peat

Our aim is to map the presence/absence of peat in the soil profile. This is done by mapping the peat occurrence probability. The mapped probabilities are used to separate the peat and no-peat areas using a probability cut-off value. This value is determined such as suggested by **?**.

5.1 Geostatistical interpolation

Spatial prediction of the probabilities is done via prediction of $S(\cdot)$ (Eq. 2). We have 3000 MCMC simulations of $S(\cdot)$ at each sampling site x_i. $S(\cdot)$ is predicted at prediction locations x_0 by kriging a single MCMC sample. Repeating this for each sample results in 3000 surfaces (maps) with predictions of $S(\cdot)$. Each of these surfaces is then back-transformed to the original scale to obtain the probability maps. Finally, the 3000 probability maps are averaged.

The signal process $S(\cdot)$ will be interpolated to the nodes of a fine prediction grid. The spatial resolution of this grid is 100 m $\times$ 100 m. The prediction grid will be constructed by overlaying the soil map of the peat areas of study area 1 with a grid of the requested resolution.

```
> p <- spsample(
     soilMap,
     cellsize = arguments$output$spatialResolution,
     type = "regular",
     offset = c(0.5, 0.5)
 )
```

To predict $S(\cdot)$ we need the values of the explanatory variables at the nodes of the prediction grid (i.e., the points at which $S(\cdot)$ is predicted). First, a `data.frame` is created that can store the explanatory variable values.

```
> expVarPred <- with(
     as(p, "data.frame"),
     data.frame(
         id = seq(1,nrow(as.data.frame(p)),1),
         x = x1,
         y = x2
     )
 )
```

Next, the data layers with the explanatory variables are read. Overlaying the prediction points with the explanatory data layers gives the values of the explanatory variables at the prediction points.

```
> for(i in 1:length(arguments$model$covarList)) {
    dum <- readGDAL(
       fname = file.path(arguments$paths$GRIDDATA_DIR, arguments$model$covarList[i])
    )
    o<-overlay(x=dum, y=p)
    names(o) <- substring(
       text=arguments$model$covarList[i],
       first=1, last=nchar(arguments$model$covarList[i])-4
    )
    expVarPred<-cbind(expVarPred,o@data)
 }
```

The object class of categorical explanatory variables is changed to *factor*, and prediction points that do not have a value for the explanatory variables are discarded.

```
> for(i in 4:ncol(expVarPred)){
 if (class(expVarPred[,i])=="integer") {expVarPred[,i]<-as.factor(expVarPred[,i])}
 }
> expVarPred <- expVarPred[!is.na(expVarPred[,4]),]
```

Spatial prediction with the GLGM is computationally intensive, and has to be done in steps to avoid memory problems. Therefore, the prediction area is split into tiles comprising 2000 prediction locations. Prediction is then for each tile individually. We must therefore first define the tile size and determine the number of tiles needed to cover the prediction grid `p`. Also, three objects have to be created in which the results of the predictions for the individual tiles can be compiled.

```
> # create dataframe with prediction locations
> predLocations <- data.frame(
     x=expVarPred$x,
     y=expVarPred$y
 )
> # define tileSize and determine the number of tiles
> tileSize <- arguments$mcmc$tile_size
> nTiles <- ceiling(nrow(predLocations)/tileSize)
> # create objects to compile predictions for the individual tiles
> pred <- NULL
> var <- NULL
> mc <- NULL
```

Once the tile size and the number of tiles are determined, kriging has to be applied to interpolate the signal $S(\cdot)$ to the nodes of the prediction grid `p`. This is done by means of the `glsm.krige` of the `geoRglm`-package.

```
> for(i in 1:nTiles){

   # select a tile
  if(tileSize*i<nrow(predLocations)){t<-predLocations[(tileSize*i-tileSize+1):(tileSize*i),]}else
     {t<-predLocations[(tileSize*i-tileSize+1):nrow(predLocations),]}

   if(tileSize*i<nrow(expVarPred)){z<-expVarPred[(tileSize*i-tileSize+1):(tileSize*i),]} else
     {z<-expVarPred[(tileSize*i-tileSize+1):nrow(expVarPred),]}

   # prepare geodata object with explanatory variables
   predGD <- as.geodata(
     obj = z,
     coords.col = 2:3,
     data.col = NULL,
     data.names = NULL,
     covar.col=4:ncol(z)
   )

   # kriging
   predGLGM <- glsm.krige(
     mcmc.output = simF,
     locations = t,
     trend.l = trend.spatial(trend=arguments$model$trendModelGLGM, geodata=predGD),
     output = output.glm.control(sim.predict = FALSE)
   )

   # compile predictions
   pred <- rbind(pred, data.frame(as.vector(predGLGM$predict)))
   var  <- rbind(var, data.frame(as.vector(predGLGM$krige.var)))
   mc   <- rbind(mc, data.frame(as.vector(predGLGM$mcmc.error)))
 }
```

A `data.frame` is created in which the predicted probabilities are stored. A snippet of this `data.frame` is given below.

```
> predictions <- data.frame(
      id = expVarPred$id,
      x  = expVarPred$x,
      y  = expVarPred$y,
      pred = pred[,1],
      var = var[,1],
      mcmcError = mc[,1]
  )
```

id	x	y	pred	var	mcmcError
1	222953.625000001	517479.3751	0.342999678351821	0.0388220128652348	0.00018640503252807
2	223053.625000001	517479.3751	0.352176465174268	0.0399755164824695	0.000203270691242147
3	222853.625000001	517579.3751	0.946448207656912	0.00155569078868817	5.1438805032336e-05
4	222953.625000001	517579.3751	0.958298178170104	0.000957221058476928	4.50461462048326e-05
5	223153.625000001	517579.3751	0.320859775619709	0.0357902750701179	0.000248244994185856
:	:	:	:	:	:
67967	216453.625000001	580879.3751	0.695730639310843	0.0335088603005955	6.76502439678985e-05
67968	216553.625000001	580879.3751	0.414714246952813	0.0465887032011928	7.87684014943818e-05
67970	217953.625000001	580879.3751	0.93324368693945	0.00237338215031552	2.60132030815978e-05
67971	218053.625000001	580879.3751	0.934230891719368	0.00230620298154478	2.39093679535488e-05

5.2 Selecting the probability cut-off value

The probability cut-off value that is used to separate the peat and no-peat areas is determined by validation, using the actual point observations on the presence/absence of peat. A prediction site where the probability exceeds the cut-off value is assigned a '1' (peat), whereas a prediction site where the probability is lower than the cut-off values is assigned a '0' (no peat). Here the approach suggested by **?** is followed. These authors define three validation measures that are determined from a contingency table, as shown below.

	Observed no peat	Observed peat
Predicted no peat	*N1*	*N2*
Predicted peat	*N3*	*N4*

In this table *N1* and *N4* represent the number of correct predictions, and *N2* and *N3* represent the incorrect predictions. **?** define a goodness score $S4$, which is the number of correctly predicted minus the number of incorrectly predicted over the total number of data points:

$$S = \frac{N1 + N4 - N2 - N3}{N1 + N2 + N3 + N4} \,. \tag{3}$$

S varies between -1 and 1; higher scores indicate more correct predictions. In addition, **?** define two bias measures B_1 and B_2:

$$B_1 = \frac{N1 + N2}{N1 + N3} \,, \tag{4}$$

$$B_2 = \frac{N3 + N4}{N2 + N4} \,, \tag{5}$$

B_1 is the number of locations where 'no peat' is predicted divided by the number of locations where 'no peat' is observed. B_2 is a similar measure, but then for the 'peat' category. When the bias equals 1, there is no bias. When the bias is smaller than 1, the peat or no peat areas are underrepresented, while values larger than 1 indicate overrepresentation.

To compute the validation measures, the predicted probabilities at the data points are extracted from the probability grid.

```
> # make data.frame with GLGM predictions spatial
> coordinates(predictions) <- ~x+y
> gridded(predictions) <- TRUE
> fullgrid(predictions) <- TRUE
> # overlay
> o <- as.data.frame(overlay(x=predictions, y=d))
> # combine data into one data.frame
> validation <- cbind(as.data.frame(d),o)
> # exclude observation points with NA
> validation <- validation[!is.na(validation$pred),]
```

A `function` is defined to compute the validation measures from the data for different cut-off values. After these values are evaluated, a probability cut-off value is selected.

```
> probThres <- function(t,validation){

      # define series of threshold values (0.10-0.90)
      threshold <- seq(
        from = 0.10,
        to = 0.90,
        by = 0.05
      )

      # create objects to store values of threshold measures
      a<-NULL
      s<-NULL
      b1<-NULL
      b2<-NULL

      # evaluate threshold measures for different thresholds
      for(i in 1:length(threshold)){

          # determine presence/absence of peat
          validation$peat <- ifelse(
            test = validation[,prob]<threshold[i],
            yes = 0,
            no = 1
          )

          # determine prediction error
          validation$error <- ifelse(
            test = (validation$ind==validation$peat),
            yes = 1,
            no = 0
          )

          # cross-table
```

```
        t <- table(validation$peat,validation$ind)

        # selection criteria
        a <- rbind(a,((t[1,1]+t[2,2])/sum(t)))               # accuracy
        s <- rbind(s,((t[1,1]+t[2,2]-t[1,2]-t[2,1])/sum(t))) # goodness
        b1 <- rbind(b1,((t[1,1]+t[1,2])/sum(t[1,1]+t[2,1]))) # bias for no peat
        b2 <- rbind(b2,((t[2,1]+t[2,2])/sum(t[1,2]+t[2,2]))) # bias for peat
    }

    # compile threshold measures in one data.frame
    x <- data.frame(
        cutOff = threshold,
        accuracy = round(a,3),
        goodness = round(s,3),
        bias1 = round(b1,2),
        bias2 = round(b2,2),
        biasdif = abs(round(b1,2)-round(b2,2))
    )

    # determine maximum goodness and minimum bias
    x$gInd<-ifelse(
        test = x$goodness==max(x$goodness),
        yes = 1,
        no = 0
    )

    x$bInd<-ifelse(
        test = x$biasdif==min(x$biasdif),
        yes = 1,
        no = 0
    )

    # determine probability threshold on basis of goodness and bias
    y<-x
    y$bInd<-ifelse(
          test = y$biasdif<0.15,
          yes = 1,
          no = 0
    )
    z<-y[y$bInd==1,]
    z$gInd<-ifelse(
          test = z$goodness==max(z$goodness),
          yes = 1,
          no = 0
    )
    z<-z[z$gInd==1,]

    return(list(
        x,
        x[x$gInd==1,][,2], # maximum accuracy
        x[x$gInd==1,][,1], # threshold for maximum accuracy
        x[x$bInd==1,][,4], # minimum for minimum bias1
        x[x$bInd==1,][,5], # minimum for minimum bias2
        x[x$bInd==1,][,1], # threshold for minimum accuracy
        z[,1], # threshold
        z[,2], # accuracy
        z[,4], # bias1
        z[,5]  # bias2
        )
    )
}
```

The cut-off `function` is applied to the 'validation' `data.frame` that stores the predicted probabilities at the data points.

```
> prob<-"pred"
> ptGLGM<-probThres(prob,validation)
```

The table below shows the validation measures for a set of cut-off probabilities.

cutOff	accuracy	goodness	bias1	bias2
0.10	0.67	0.33	0.02	1.51
0.15	0.69	0.38	0.09	1.47
0.20	0.71	0.43	0.20	1.41
0.25	0.72	0.45	0.27	1.38
0.30	0.73	0.47	0.37	1.33
0.35	0.77	0.54	0.61	1.20
0.40	0.77	0.54	0.74	1.13
0.45	0.78	0.56	0.88	1.06
0.50	0.77	0.54	0.97	1.01
0.55	0.77	0.53	1.05	0.97
0.60	0.77	0.54	1.15	0.92
0.65	0.76	0.53	1.25	0.87
0.70	0.77	0.53	1.36	0.81
0.75	0.74	0.48	1.54	0.72
0.80	0.70	0.39	1.72	0.63
0.85	0.66	0.31	1.87	0.55
0.90	0.60	0.20	2.08	0.44

The goodness value is largest for a cut-off value 0.45, whereas the bias is smallest for cut-off value 0.5. To select the 'optimal' cut-off value we decided that the absolute difference in bias measures B_1 and B_2 must be smaller than 0.15. Within this range, the cut-off value is selected for which the goodness is largest. In our case the selected cut-off value is 0.5.

The calibration accuracy of the predicted soil map is 0.779 for cut-off value 0.45. This means that the model correctly predicts the occurence of peat at 77.9 of the data points. For the selected cut-off, the calibration accuracy is 0.772. Note that map accuracy computed from independent data, i.e. data that are not used to calibrate the prediction model, typically is smaller than the calibration accuracy.

With the selected cut-off value, the peat category at the data points can be determined.

```
> validation$peat <- ifelse(
    test = validation$pred<ptGLGM[[7]],
    yes = 0,
    no = 1
 )
```

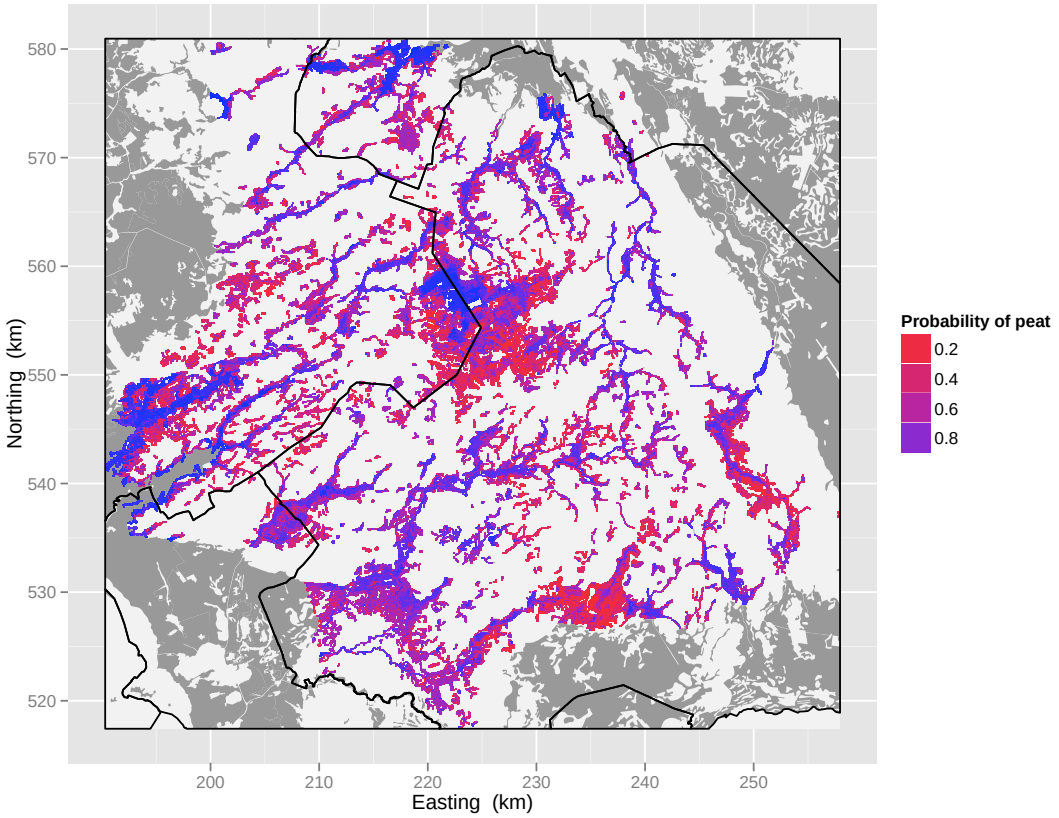
Observed no peat	Observed peat
169	83
90	417

The class representation, which is the proportion of the observations for which peat occurence is correctly predicted, is for 'no peat' 0.652509652509653, and for 'peat' 0.834.

5.3 Soil maps

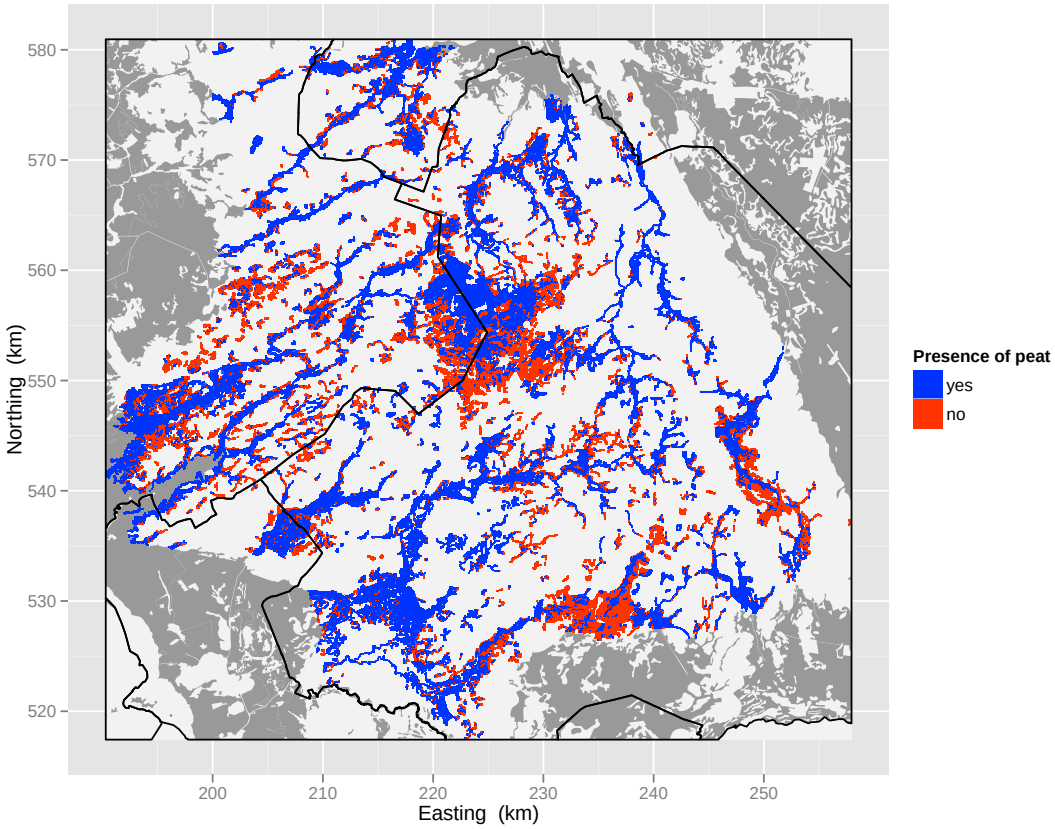
The map with the predicted probabilities of the occurence of peat is shown below. The map with the predicted probabilities of the occurence of peat is shown below. The dark grey areas

indicate the extent of the peat soils outside the study area.



With the selected cut-off value, the peat category at the prediction points can be determined. The resulting soil map is shown below.

```
> predictions$peat <- ifelse(
    test = predictions$pred<ptGLGM[[7]],
    yes = 0,
    no = 1
 )
```



6 Summary

6.1 Settings

document information		
title	:	Mapping the presence/absence of peat for the northern till plateau with the Generalized Linear Geostatistical Model
version	:	0.1-0
date	:	April 5, 2012
author	:	Bas Kempen
based on	:	Brus et al. (2010)
execution time	:	2:01:00
platform	:	Windows, 7 x64, build 7601, Service Pack 1, x86-64
target variable	:	Presence/Absence of peat
temporal extent	:	2002-01-01 to 2010-12-31

6.2 Session information

- R version 2.14.1 (2011-12-22), `x86_64-pc-mingw32`

- Base packages: base, datasets, graphics, grDevices, grid, methods, stats, utils
- Other packages: coda 0.14-6, foreign 0.8-48, geoR 1.7-2, geoRglm 0.9-2, ggcolpairs 0.2.4, ggplot2 0.9.0, gpclib 1.5-1, gstat 1.0-10, lattice 0.20-0, maptools 0.8-10, MASS 7.3-16, plyr 1.6, rgdal 0.7-1, rgeos 0.2-3, sp 0.9-91, spacetime 0.5-7, stringr 0.6, xtable 1.7-0, xts 0.8-2, zoo 1.7-7
- Loaded via a namespace (and not attached): colorspace 1.1-1, dichromat 1.2-4, digest 0.5.1, memoise 0.1, munsell 0.3, proto 0.3-9.2, RandomFields 2.0.54, RColorBrewer 1.0-5, reshape2 1.2.1, scales 0.2.0, splancs 2.01-31, tools 2.14.1

References

Ben-Ahmed, K., Bouratbine, A., and El-Aroui, M.-A. (2010). Generalized linear spatial models in epidemiology: A case study of zoonotic cutaneous leishmaniasis in Tunisia. *Journal of Applied Statistics*, 37(1):159–170.

Brus, D. J., Vasat, R., Heuvelink, G. B. M., Knotters, M., de Vries, F., and Walvoort, D. J. J. (2010). Towards a soil information system with quantified accuracy; a prototype for mapping continuous soil properties. Technical report, Statutory Research Tasks Unit for Nature and the Environment, WOt-werkdocument 197.

Christensen, O. F. (2004). Monte carlo maximum likelihood in model-based geostatistics. *Journal of Computational and Graphical Statistics*, 13(3):702–718.

Christensen, O. F. and Ribeiro Jr., P. (2002). `geoRglm`: A package for generalized linear spatial models. *R-News* 2:26-28. Available at: http://cran.r-project.org/doc/rnews.

Clement, J. and Kooistra, L. (2003). Eerste bosstatistiek digitaal; opbouw van een historisch basisbestand. Technical Report 744, Alterra.

de Vries, F., Mol, G., Hack-ten Broeke, M. J. D., Heuvelink, G. B. M., and Brouwer, F. (2008). Het Bodemkundig Informatie Systeem van Alterra. Technical Report 1709, Alterra.

Diggle, P., Moyeed, R., Rowlingson, B., and Thomson, M. (2002). Childhood malaria in the Gambia: A case-study in model-based geostatistics. *Journal of the Royal Statistical Society. Series C: Applied Statistics*, 51(4):493–506.

Diggle, P. J. and Ribeiro Jr., P. J. (2007). *Model-based Geostatistics*. Springer Series in Statistics. Springer, New York.

Diggle, P. J., Tawn, J. A., and Moyeed, R. A. (1998). Model-based geostatistics. *Journal of the Royal Statistical Society. Series C: Applied Statistics*, 47(3):299–325.

Finke, P. A., Groot Obbink, D. J., Rosing, H., and de Vries, F. (1996). Actualisatie Gt-kaarten 1 : 50 000 Drents deel kaartbladen 16 Oost (Steenwijk) en 17 West (Emmen). Technical Report 439, DLO-Staring Centrum.

Geweke, J. (1992). Evaluating the accuracy of sampling-based approaches to calculating posterior moments. In Bernado, J. M., Berger, J. O., Dawid, A. P., and Smith, A. F. M., editors, *Bayesian Statistics 4*. Clarendon Press, Oxford, UK.

Goovaerts, P. (1997). *Geostatistics for Natural Resources Evaluation*. Oxford University Press, New York.

Hazeu, G. W. (2005). Landelijk Grondgebruiksbestand Nederland (LGN5). Vervaardiging, nauwkeurigheid en gebruik. Technical Report 1213, Alterra.

Hoogland, T., van den Akker, J. J. H., and Brus, D. J. (2012). Modeling the subsidence of peat soils in the Dutch coastal area. *Geoderma*, 171-172:92–97.

Hosmer, D. and Lemeshow, S. (2000). *Applied Logistic Regression (2nd Edition)*. John Wiley & Sons, New York.

Kempen, B., Brus, D. J., and Heuvelink, G. B. M. (2012). Soil type mapping using the generalized linear geostatistical model: a case study in a Dutch cultivated peatland. *Submitted to Geoderma*.

Knol, W., Kramer, H., Dorland, G., and Gijsbertse, H. (2003). Historisch Grondgebruik Nederland: tijdreeksen grondgebruik Noord-Holland van 1950 tot 1980. Technical Report 751, Alterra.

Knol, W., Kramer, H., and Gijsbertse, H. (2004). Historisch Grondgebruik Nederland: een landelijke reconstructie van het grondgebruik rond 1900. Technical Report 573, Alterra.

Knuth, D. E. (1984). Literate programming. *The Computer Journal*, 27(2):97–111.

Koomen, A. and Maas, G. (2004). Geomorfologische Kaart Nederland (GKN); Achtergronddocument bij het landsdekkende digitale bestand. Technical Report 1039, Alterra.

Lark, R. M. and Cullis, B. R. (2004). Model-based analysis using REML for inference from systematically sampled data on soil. *European Journal of Soil Science*, 55(4):799–813.

Leisch, F. (2002). Sweave: Dynamic generation of statistical reports using literate data analysis. In Härdle, W. and Rönz, B., editors, *Compstat 2002 — Proceedings in Computational Statistics*, pages 575–580. Physica Verlag, Heidelberg.

Minasny, B., Vrugt, J. A., and McBratney, A. B. (2011). Confronting uncertainty in model-based geostatistics using Markov Chain Monte Carlo simulation. *Geoderma*, 163(3-4):150–162.

Pebesma, E. J. (2004). Multivariable geostatistics in S: the gstat package. *Computers & Geosciences*, 30:683–691.

Ribeiro Jr., P. and Diggle, P. J. (2001). `geoR`: A package for geostatistical data analysis using the R sofware. *R-News* 1(2). Available at: http://cran.r-project.org/doc/rnews.

van Kekem, A. J., Hoogland, T., and van der Horst, J. B. F. (2005). Uitspoelingsgevoelige gronden op de kaart; werkwijze en resultaten. Technical Report 1080, Alterra.

Webster, R. and McBratney, A. B. (1989). On the Akaike Information Criterion for choosing models for variograms of soil properties. *Journal of Soil Science*, 40(3):493–496.