

Investigating Spatial Prediction of Blister Rust Infection Rates in Whitebark Pine

Matt Tyers

April 29, 2014

Contents

1	Introduction	2
1.1	Ecological Background	2
1.2	Study Design	2
2	Evidence of Spatial Autocorrelation	3
2.1	Graphical Assessment - Mapping Site Infection Rates	3
2.2	Graphical Assessment - Mapping Site-level Model Residuals	5
2.3	Tests for Spatial Autocorrelation	6
2.3.1	Moran's I	7
2.3.2	Mantel's Test	7
2.4	Graphical Assessment of Correlation & Covariance	7
2.5	Modeling Spatial Correlation	9
3	Spatial Prediction	9
3.1	Introduction to Kriging	10
3.2	Regression Kriging using Residuals	11
3.3	A Bayesian Approach for Generalized Linear Geostatistical Models	13
4	Discussion	15
5	Graphical functions developed for this analysis	16
6	Acknowledgements	17
7	References	17
8	Appendix - R Code Used	18

1 Introduction

1.1 Ecological Background

Whitebark pine (*Pinus albicaulus*) is a conifer species native to the Greater Yellowstone Ecosystem (GYE), and occupies a subalpine niche habitat, growing almost exclusively in the region immediately below timberline. Whitebark pine produce large, nutrient-rich cones, an important food source for the Clark's nutcracker (*Nucifraga columbiana*), red squirrel (*Sciurus vulgaris*), and notably the black and grizzly bear (*Ursus americanus* and *Ursus arctos horribulus*). With the Yellowstone grizzly currently federally listed as an endangered species, conservation of the whitebark pine is a particularly sensitive issue.

Whitebark pine in the GYE has sustained damage from two biological agents. Recent years have seen a large-scale outbreak of attack from the native mountain pine beetle (*Dendroctonus ponderosae*), thought to be linked to a recent pattern of warmer winters. In addition to this more recent threat, the GYE whitebark pine has also experienced infection over the past several decades from blister rust (*Cronartium ribicola*), an exotic fungus introduced to North America in the 1920's, which can impact all five-needled pine species. Blister rust disrupts the cambium layer of infected trees, and will eventually kill branches or whole trees. It has also been hypothesized that mountain pine beetle attack may occur more frequently in stands of whitebark that have been weakened by blister rust infection.

When blister rust spore dissemination occurs, the spores are suspended on atmospheric water droplets (clouds or fog), making the patterns of spread subject to climate or microclimate, which is in turn influenced by local topography. Infection does not spread tree-to-tree; rather, it is spread by means of an intermediary host, typically shrubs of the *Ribes* genus (currant or gooseberry) but certain forbs, such as Indian paintbrush (*Castilleja*) and elephanthead (*Pedicularis groenlandica*), have also been shown to transmit blister rust infection.

1.2 Study Design

During the summer and fall field seasons of 1995-2010, the Gardiner district field crew opportunistically sampled whitebark pine in and around the Absaroka-Beartooth Wilderness of the GYE, as part of a project intended to characterize the spread and severity of blister rust infection, as well as assessing the possible relationships between environmental variables and infection rates. At each sampling location, the health of each whitebark pine tree within a 300 foot by 10 foot belt transect was assessed, and all sources of damage were recorded and classified by severity. In addition to tree health, site characteristics were also recorded: elevation (found using a handheld GPS), slope (found using a clinometer or compass), aspect (found using a compass), forest type (according to the Mattson-Despain cover type key), and *Ribes* presence.

Sampling was not random. Instead, sampling areas were chosen so as to provide as much spatial coverage of the primary study area (A-B Wilderness and surrounding area with white-

bark pine) as possible. In each area, sampling site (transect) placement was conducted as time and workflow allowed, with the intent of sampling stands of trees that were “representative” of the immediate area. Infection is often difficult to see, and the majority of the forest canopy represented in a 300 foot transect is simply not visible from the start of the line. Even though sampling area selection and transect placement were not random, this effective “blinding” likely mitigated any potential field technician bias related to blister rust, that is, choosing sampling sites with high or low amounts of blister rust. The distribution of sites sampled is shown in Figure 1.

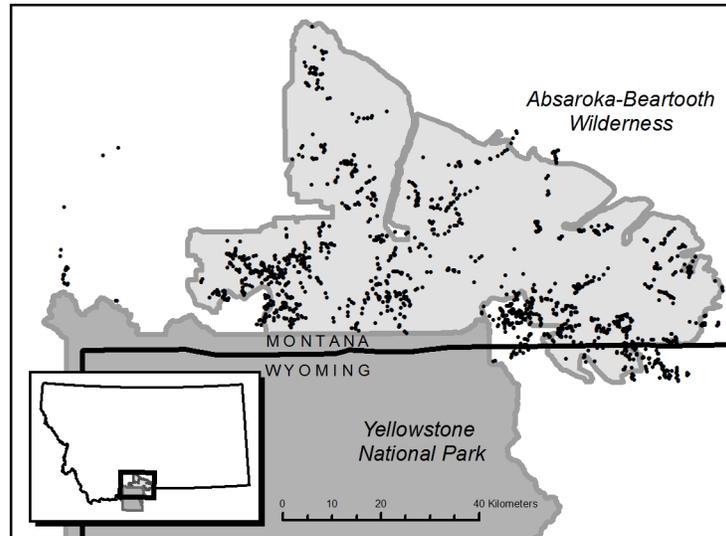


Figure 1: Sites sampled during the years 1995-2010.

Sampling was conducted over a period of 16 field seasons, with each season representing a relatively broad spatial coverage of the study area. However, due to the logistics of back-country travel, it was often advantageous to sample heavily in one valley before moving to the next, and so on. Resulting temporal patterns in sampling efforts are shown in Figures 2 and 3, first showing the spatial distribution of sampling sites for 2-3 year intervals, then as a single map with colors on a color gradient representing sampling year.

2 Evidence of Spatial Autocorrelation

2.1 Graphical Assessment - Mapping Site Infection Rates

Without any further analysis, it can certainly be surmised that infection rates will exhibit a degree of spatial autocorrelation, that is, that sampling sites closer together will have similar infection rates. The rate of infection can be expected to be related to the degree of spore presence at a particular site, and the degree of spore presence is directly driven by the presence of fruiting bodies at active infection sites. Therefore, we can expect to see a pattern

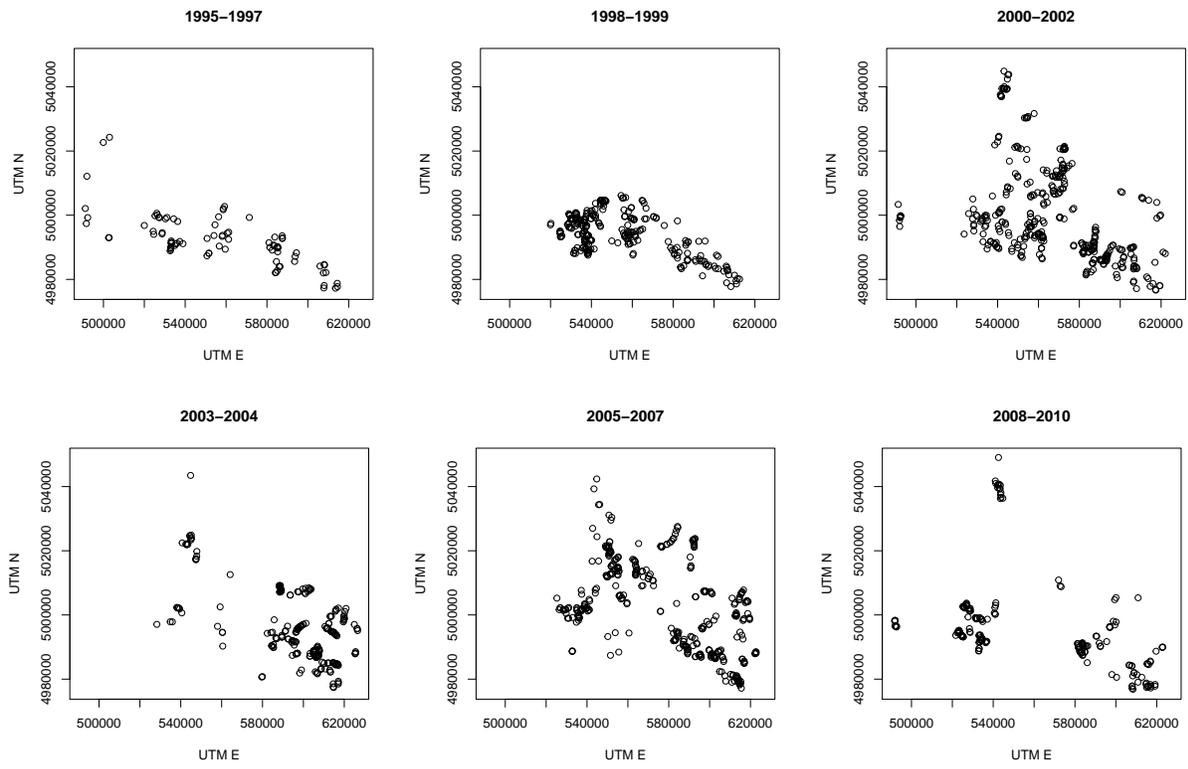


Figure 2: Sampling sites are shown, divided into three-year intervals.

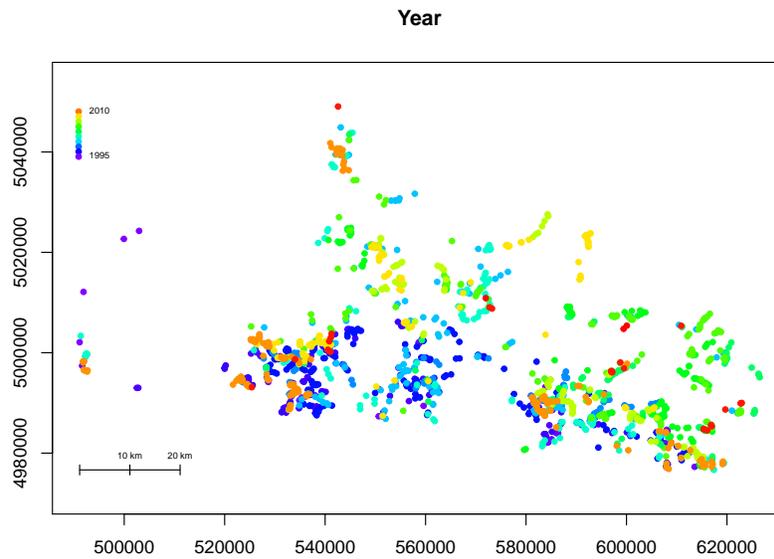


Figure 3: All sampling sites are shown, with colors corresponding to sampling year.

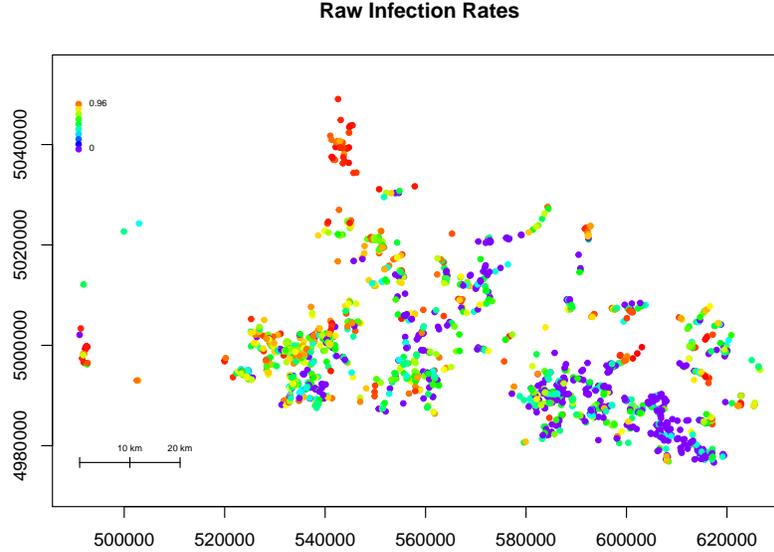


Figure 4: Proportion of infected trees for each sampling site.

of “hot spots” and “cold spots”: areas in which there is a high degree of infection due to a natural “feedback” cycle, and areas in which this has not yet occurred.

Plotting the raw infection rates (defined as $\frac{\#InfectedTrees}{\#TotalTrees}$ at a given sampling site), the pattern of spatial autocorrelation is immediately evident (see Figure 4), particularly the consistently high rates of infection in the northwestern and southwestern portions of the study area (the Absaroka front, south of Livingston, and the Tom Miner/Buffalo Horn Pass areas respectively) and the consistently low rates of infection in the southeastern portion of the study area (the southern Beartooth range, near Cooke City).

2.2 Graphical Assessment - Mapping Site-level Model Residuals

An initial intent of this research was to explore possible relationships between site-level variables (elevation, aspect, etc.) and infection rates. A binomial logistic regression model was fit, with the form given below, where y_i = number of infected whitebark pine trees at site i , m_i = total number of whitebark pine trees at site i .

$$y_i \sim Bin(m_i, \pi_i)$$

$$logit(\pi_i) = \sum_{k=1}^p \beta_k d_k(x_i)$$

In this notation, each $d_k(x_i)$ refers to covariate d_k observed at location x_i . The covariates used in this model were recorded on-site. Elevation (km) and slope (degrees) were treated

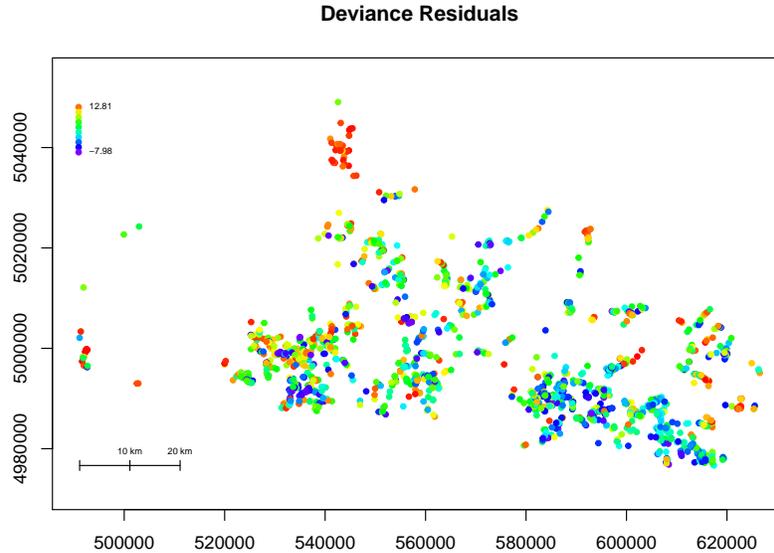


Figure 5: Deviance residuals from the site-level variable model.

as continuous. *Ribes* presence was recorded as a categorical variable, with levels defined as *Yes*, *No*, and *Undetermined*. Aspect was treated as a categorical variable, classified into quadrants: *North*, *South*, *East*, and *West*. Forest type was recorded on-site according to the Mattson/Despain forest cover type key, but was aggregated into the following categories: *Krummholtz/tundra*, *Lodgepole*, *Mixed forest*, *Mix/open*, *Old successional Whitebark*, *Pure Whitebark*, *Young successional Whitebark*, and *Whitebark/open*. Interactions thought to capture local weather patterns (which drive spore dispersal) were also included, specifically $Slope \times Aspect$ and $Elevation \times Aspect$.

This site-level model seems to show evidence of relationships between site-level variables and infection rates at sampled sites. However, mapping the residuals from this model (see Figure 5) shows a spatial pattern remains in the residuals (even after accounting for the site-level variables.)

2.3 Tests for Spatial Autocorrelation

The spatial autocorrelation in the site-level model residuals is evident from the map (Figure 4). It is common, however, to see tests of spatial autocorrelation presented, so the most common of these were investigated. In order to investigate the presence of spatial autocorrelation in the residuals, both Moran's I and Mantel's test were employed.

2.3.1 Moran’s I

Moran’s I is a summary measure quantifying autocorrelation. It is defined as $I(d) = \left(\frac{1}{W(d)}\right) \frac{\sum_{(i=1, i \neq j)}^n \sum_{(j=1, j \neq i)}^n w_{ij}(d)(x_i - \bar{x})(x_j - \bar{x})}{\frac{1}{n} \sqrt{\sum_{i=1}^n (x_i - \bar{x})^2}}$ (Fortín & Dale 2005), in which each x_i is an observed value at location i , $w_{ij}(d)$ represents a “weight” corresponding to a pair of observations i and j . The weight is typically the inverse or inverse square of the euclidean distance between observations i and j , and $W(d)$ is the sum of all $w_{ij}(d)$ values. Moran’s I can be thought of as a weighted average of all pairwise correlations in the observed values, with weights defined by the corresponding euclidean distances between observed values. Moran’s I will have a value between -1 and 1, with positive values indicating positive spatial autocorrelation and 0 indicating no spatial autocorrelation. The expected value of the Moran’s I can be calculated as $-\frac{1}{n-1}$ under no spatial autocorrelation.

Using Moran’s I with an inverse square distance weight matrix to test for spatial autocorrelation in the residuals from the site-level model described above, yields $I = 0.40$. Under a null hypothesis of no spatial autocorrelation, we can calculate $E(I) = -0.00065$, and $SD(I) = 0.0185$. Using a normal approximation, this leads to a p-value < 0.0001 . Therefore, using Moran’s I, we have conclusive evidence of spatial autocorrelation in the residuals.

2.3.2 Mantel’s Test

Mantel’s test calculates the correlation between the spatial distance matrix and the distance matrix of the responses (in this case, the residuals), and uses a permutation approach to evaluate the evidence of a correlation between the two matrices (Mantel 1967). Mantel’s test was also employed to test for for spatial autocorrelation in the residuals from the above model. A Mantel test statistic of $r = 0.1502$ was calculated, and performing 999 permutations gave a permutation p-value of 0.001. The upper permutation quantiles for the Mantel statistic (under the null model of no spatial autocorrelation) were $q_{0.95} = 0.0154$, and $q_{0.99} = 0.0220$. Therefore, this test also yields conclusive evidence of spatial autocorrelation in the residuals, implying spatial autocorrelation remains, even after accounting for site-level covariates available.

2.4 Graphical Assessment of Correlation & Covariance

Additional techniques of graphical assessment of the spatial autocorrelation in site-level model residuals are perhaps more illustrative, and useful for subsequent modeling. Shown in Figure 6 (left) is a correlogram, which depicts distance on the x-axis and correlation on the y-axis, for the binned residuals. The autocorrelation coefficient r_h for lag h is calculated as $r_h = \frac{c_h}{c_0}$, in which c_h is the autocovariance function $c_h = \frac{1}{N} \sum_{t=1}^{N-h} (y_t - \bar{y})(y_{t+h} - \bar{y})$ for some spatial process y , and c_0 is the variance function, defined as $c_0 = \frac{1}{N} \sum_{t=1}^N (y_t - \bar{y})^2$ (Box & Jenkins 1976). Function `correlog()` within package `ncf` (Bjornstad 2013) calculates these correlations for binned distances. In this instance, the residuals still show a large amount

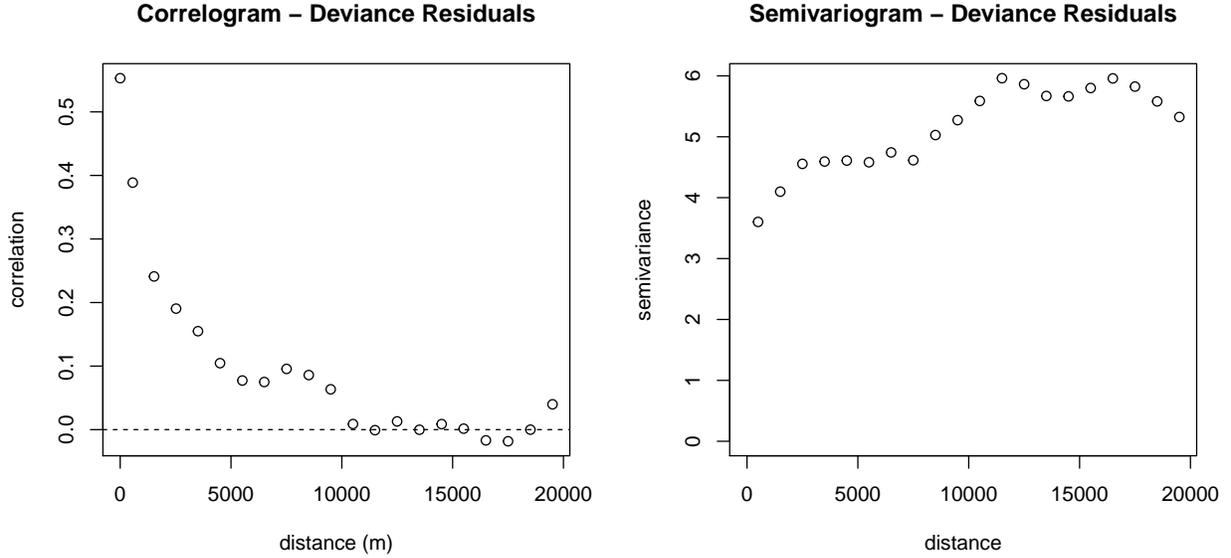


Figure 6: Correlogram (left) and semivariogram (right) calculated from the site-level model residuals.

of correlation at close distances, with the degree of correlation diminishing as distance increases, with essentially no observed correlation beyond approximately 10 km. Because of this, only distances less than 20 km are displayed here.

Also shown in Figure 6 (right) is an empirical semivariogram, calculated from the residuals. Again, the x-axis depicts distance. The y-axis gives an estimate of $\gamma(x, y) = \frac{1}{2} \text{var}(S(x) - S(y))$ in which $S(\cdot)$ represents the stochastic process as realized at locations x and y . Under the assumption of stationarity, in which the expectation and variance of the spatial stochastic process is the same for all locations, and correlation depends only on distance (not direction), the semivariogram simplifies to $\gamma(u) = \sigma^2(1 - \rho(u))$ in which u denotes distance, and $\rho(u)$ denotes the value of a correlation function (the same as that approximated above) for a given distance (Diggle & Ribeiro 2007).

The estimates of $\gamma(x, y)$ are calculated as $\hat{\gamma}(h) = \frac{1}{2N(h)} \sum_{i,j \in N(h)} (z_i - z_j)^2$ for all pairs of observations (z_i, z_j) within the set $N(h)$ of pairs of observations within a lag distance h . In this case, the values of observations used are the site-level model residuals, and bins were defined in 1 km intervals. Again, this shows a relatively small degree of variability between close points, with variability increasing as distance increases.

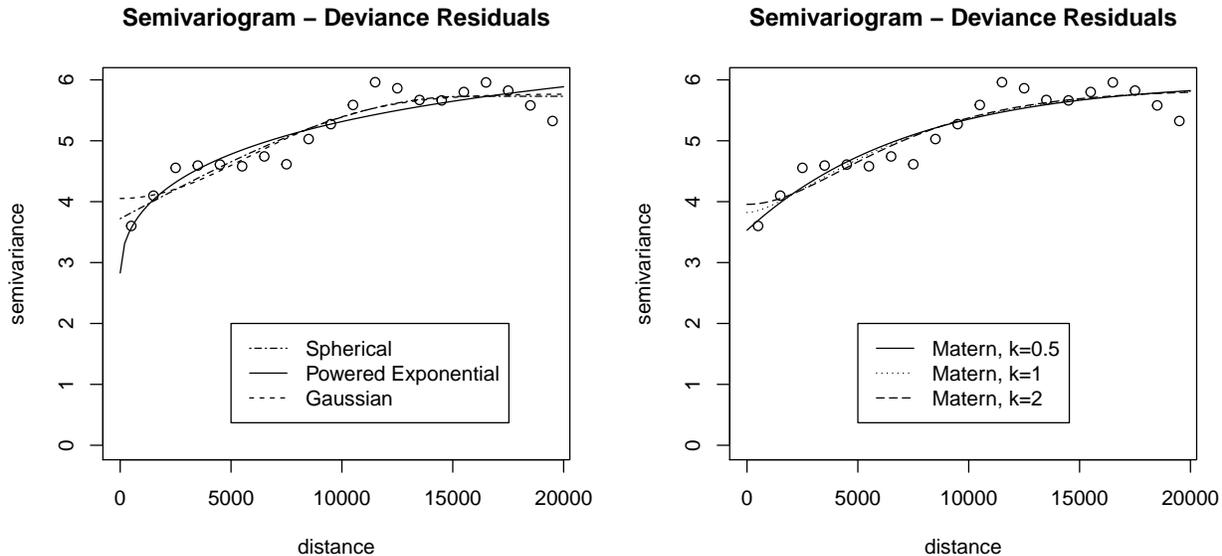


Figure 7: Correlation function forms considered.

2.5 Modeling Spatial Correlation

With the strong degree of spatial autocorrelation in the site-level model residuals, any analysis of this dataset should account for this. In order to do so, the form of the spatial correlation must be estimated. Several forms of correlation functions have been developed to model correlation (and therefore covariance) as a function of distance between locations (Diggle & Ribeiro 2007). Shown in Figure 7 are least-squares estimates of the correlation function for the residuals (as binned above). Shown left are three families of correlation functions (spherical, powered exponential, and Gaussian) and to the right, three forms of the Matérn function, with different values of the smoothing parameter κ : $\kappa = 0.5$ (also referred to as the exponential correlation function), $\kappa = 1$, and $\kappa = 2$. The exponential correlation function was ultimately selected for modeling, based on a visual assessment of fit, particularly at close distances.

3 Spatial Prediction

Perhaps what is most useful to Forest managers is a spatial prediction of the blister rust infection rates for the full study area, that is, the A-B Wilderness and surrounding area. For this, it is possible to use the spatial covariance modeled above, within some spatial prediction technique. Presented here are two methods for spatial prediction of infection rates on the probability scale: a traditional Kriging approach, and a Bayesian method.

3.1 Introduction to Kriging

Kriging, first developed by Danie Krige (1951) and further refined by Georges Matheron (1963), was initially intended for mining applications, in order to estimate the spatial distribution of gold deposits based on samples from a few boreholes (Schabenberger & Gotway 2005). Kriging is a spatial prediction technique under the assumption of an underlying multivariate Gaussian surface, with a spatial covariance structure that must be estimated using the spatial covariance of the data used, in this case, the model residuals. In order to do this, the form of the correlation function must be specified, along with the parameters of the chosen correlation function. Kriging assumes first- and second-order stationarity, that is, that the expectation and variance of the random spatial process be equal in all locations, with correlation depending only on distance. Using typical methods, this necessitates the assumption that no directional effects present (isotropy), however, Kriging methods have been developed that account for the presence of directional effects (anisotropy).

Three commonly-used forms of Kriging have been developed, each with a different assumption of the expectation of the underlying multivariate Gaussian surface. *Simple Kriging* assumes the expectation of the Gaussian surface to be equal to some known value at all locations. *Ordinary Kriging* assumes the expectation to be equal at all locations, but allows the mean to be unknown. *Universal Kriging* extends this, and uses the assumption that the mean is equal to some linear function of site-level covariates, i.e. $\sum_{j=1}^q \beta_j d_j(x_i)$ (Schabenberger & Gotway 2005).

The underlying structure of the model used by Universal Kriging can be expressed in the form of a generalized geostatistical model, expressed as shown below for location x_i , and employing the logit-link used in this analysis (Diggle & Ribeiro). The set of β_j 's represent a set of regression parameters, with associated values of $d_j(x_i)$ for each observation of site-level explanatory variables at location x_i (elevation, slope, aspect, etc.) All of these elements are analogous to the pieces of a classical generalized linear regression model. What a geostatistical model adds is denoted $S(x_i)$, the realization of an assumed zero-mean multivariate Gaussian stochastic process at location x_i . Adding $S(x_i)$ is analogous to adding a random effect with a form of correlation depending on spatial distance between observations.

$$y_i \sim \text{Bin}(m_i, \pi_i)$$

$$\text{logit}(\pi_i) = \sum_{j=1}^q \beta_j d_j(x_i) + S(x_i)$$

The $\sum_{j=1}^q \beta_j d_j(x_i)$ piece of the geostatistical model (the mean used in Universal Kriging) can be predicted from the same set of site-level predictors at some new site, using the estimated regression coefficients from a binomial logistic regression model (discussed later). The spatial component $S(x_i)$ can be predicted from the residuals of the same model using Kriging.

3.2 Regression Kriging using Residuals

Spatial prediction using traditional Kriging methods was performed for the points on a prediction grid, which was drawn at 1km intervals within a minimum convex polygon drawn from the locations of the sampling sites. For each point on the prediction grid, values of slope, elevation, and aspect were extracted from a 30m digital elevation model using ArcGIS. Unfortunately, not all measured site-level variables were able to be extracted for each prediction location. Specifically, forest type and *Ribes* presence were unavailable.

With the vegetation variables unavailable for the prediction points, it was necessary to formulate a binomial logistic regression model with new covariates, using available information as a surrogate for unavailable information. Since patterns in vegetation tend to occur in accordance with site topography, an attempt was made to use the site topography variables (slope, elevation, and aspect; all covariates in the original model) to account for the missing vegetation variables. This was done in the hope of being able to account for the variation in infection rate due to site vegetation variables (forest type, *Ribes* presence) that would themselves be driven by varying combinations of site topography. To accomplish this, the quantitative site topography variables (slope and elevation) were treated as 10-level categorical variables instead, and interaction between elevation (10 categories) and aspect (still treated as 4 categories) were included. Treating the topography variables as categorical rather than continuous is likely a more accurate description of what would influence site vegetation, which likely responds to discrete combinations of site topography, rather than linear relationships with either variable.

Prediction of the spatial random effect was done using the `geoR` package (Ribeiro & Diggle 2001), using the deviance residuals from the topography-variable binomial logistic regression model, fit separately using `glm()`. The correlation function used was the exponential (Matérn with $\kappa = 0.5$), with parameters estimated through maximum likelihood estimation from the topography-variable model residuals, using `likfit()`, a function in `geoR`. Prediction locations outside the ranges of slope and elevation found in sampling sites were excluded from prediction. The linear model (mean) component and spatial component were added for each prediction location, and back-transformed to the probability scale, with the resulting predicted infection probabilities plotted in Figure 8.

Any form of estimation is incomplete without an estimate of uncertainty, which must be given for all prediction locations. The Kriging algorithm implemented in `geoR` gives the estimated Kriging variance (Figure 9), which can be thought of as the variability in estimation of the random surface. The Kriging variance is characterized precisely as we might expect, with estimated prediction variance increasing as distance increases from a sampling location.

Naïve variance estimates for the prediction at each prediction location were combined by adding the estimated linear model prediction variance to the estimated Kriging variance, making a naïve assumption of no covariance. Confidence intervals were then calculated for each prediction location. Endpoints of approximate 95% confidence intervals were constructed using the naïve standard errors (Figure 10).

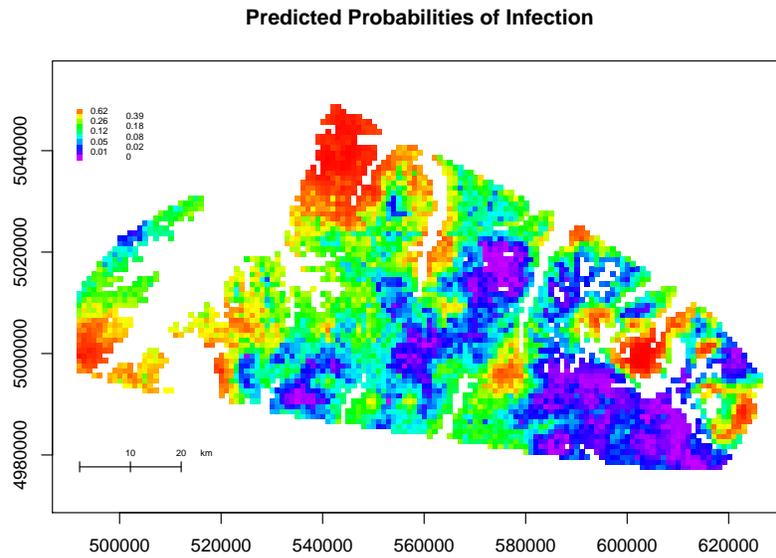


Figure 8: Predicted probabilities of infection for a 1-km grid. This prediction grid was produced by adding the logit-scale predicted values from the topography variables to the logit-scale predicted values from simple Kriging performed on the topography variable model, back-transformed to the probability scale.

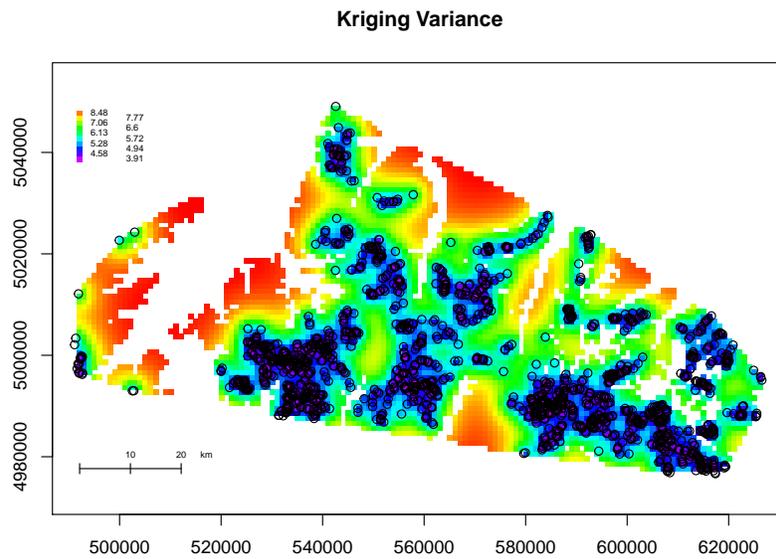


Figure 9: Estimated Kriging variance

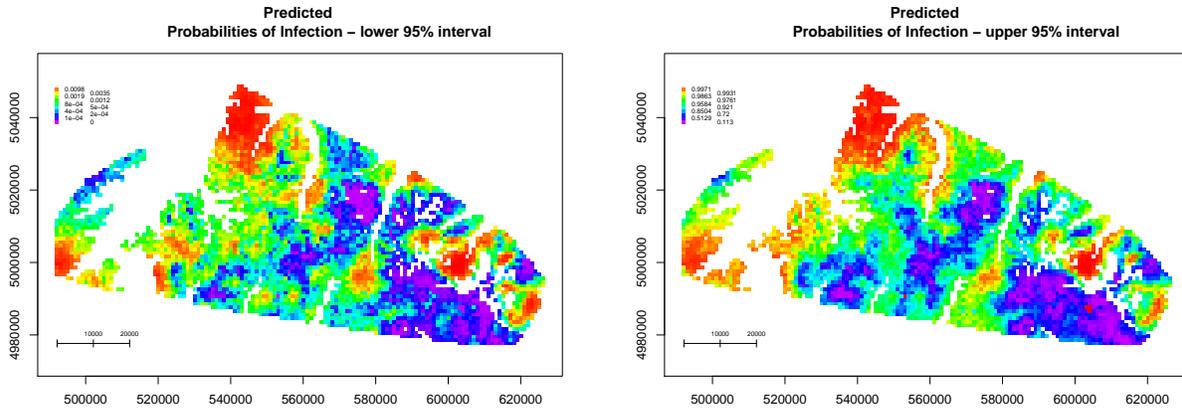


Figure 10: Approximate 95% confidence interval for infection probabilities at prediction locations, from Kriging on residuals.

3.3 A Bayesian Approach for Generalized Linear Geostatistical Models

Diggle & Ribeiro (2007) developed a Bayesian approach to universal Kriging (that is, Kriging when the mean is defined by a set of site-specific covariates) for non-normal responses, that is, a GLM setting. This lends the elegance of unifying estimation of spatial covariance parameters and logistic regression parameters used to predict at new locations, with a non-normal model for the mean. The benefit to be derived from this approach is the allowance for uncertainty in estimation of the spatial covariance parameters in prediction, whereas under the classical Kriging approach, prediction is performed under the assumption that the estimated spatial covariance parameters are the “truth.” Another benefit to the Bayesian approach is the relative simplicity of generating posterior intervals for each prediction location. Prediction was performed using the `geoRglm` package (Christiansen & Ribeiro 2002). Posterior medians and 0.025 & 0.975 quantiles of the posterior distributions at each location in the prediction grid are provided in Figures 11 and 12.

A look at the plots of the upper and lower quantiles reveals that something strange is happening at a UTM Northing coordinate of 5,000,000. At this point, I can only offer speculation as to what might be causing this “break” in the prediction values. However, we must remember that predictions were made directly from site topography variables, treated as categorical, and allowing interactions. These site topography variables were themselves derived quantities, calculated from a digital elevation model. As an example, the derived slopes are plotted in Figure 13. It’s not as obvious, but I believe the same horizontal line at UTMN = 5,000,000 to be visible.

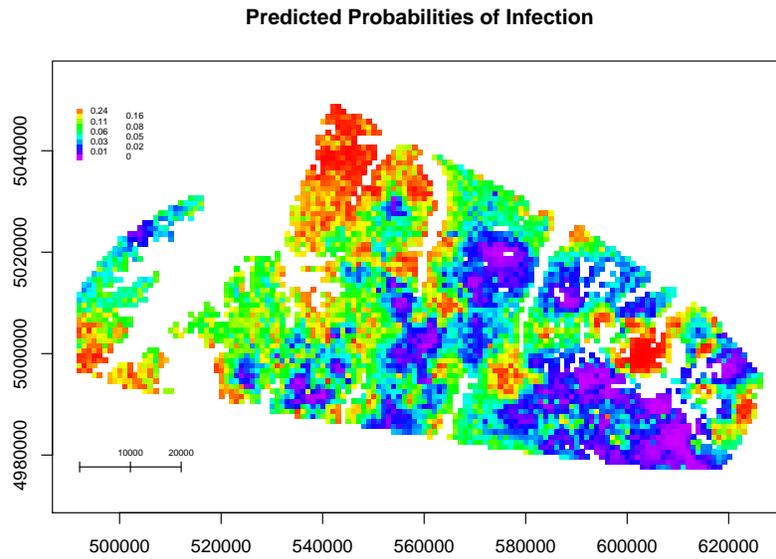


Figure 11: Predicted probabilities of infection (posterior medians) for a 1-km grid, made using Bayesian spatial prediction methods.

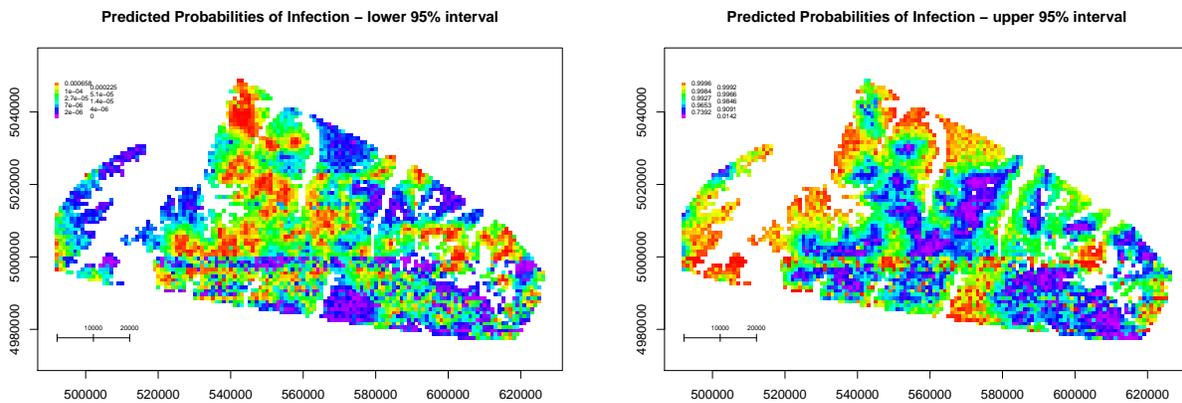


Figure 12: Approximate 95% Posterior interval for infection probabilities at prediction locations.

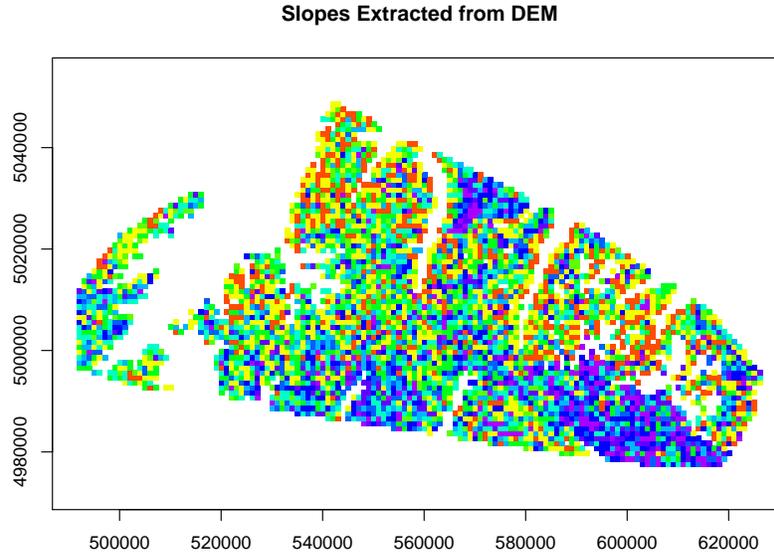


Figure 13: Values of slope extracted from the digital elevation model, again showing a horizontal line artifact at a UTM Northing coordinate of 5,000,000

To go looking for an assignable cause in this instance is an exercise in pure speculation, and extreme caution is required. However, it's within conjecture that what we're observing is some form of breakpoint in the source data. Perhaps the elevation model was processed in multiple batches, and what we're observing is a slight difference in smoothing. If this were true, a very slight perturbation in the elevation model could have the effect of larger perturbations in any derived quantities (even for no other reason than the fact that they are derived quantities), particularly at the edge. Then, any complex analysis using those derived quantities would potentially be even more subject to those perturbations, and exacerbated by the inclusion of interactions. Future work may include further investigation of this with graphical techniques.

4 Discussion

Any inference drawn from this analysis should be made with some caution. Any form of prediction to new sites is a form of inference to a larger population. Since sampling was not random, we must make the assumption that the sampling locations were representative of whitebark pine stands in the study area, and not subject to bias. If there were any relationship between infection rates and site characteristics related in some way to accessibility, there is certainly the potential for bias. In addition, since sampling sites were selected by field technicians, it's certainly within conjecture that well-meaning technicians may have selected so as to include areas of infection. This is unlikely, due to the relative difficulty of seeing infection at a distance, but the potential does exist.

Another serious limitation to this study is the potential confounding of space with time. Even though the spread of blister rust is a relatively slow process, the data collection phase lasted sixteen years, potentially enough for infection rates to increase appreciably. Sampling in any given field season was widespread across the study area, but a number of sampling site “clusters” over space were also clustered in time, particularly in areas where access was difficult (e.g. one trip was made to sample Saderbalm Basin, in the fall of 2008). This could potentially cause problems in two ways: first, any spatial analysis considered here only accounts for spatial distance between sampling sites, not temporal. It is thus potentially problematic to treat the sampling of a region in the early years of the data collection as equivalent to sampling in the later years of the study. In addition, since spatial prediction was performed using data from the past decades, caution should perhaps be advised in inferring these results to the present day or the future.

In addition, the spatial prediction methodology used in this paper was subject to the assumption of isotropy, which might be inappropriate to assume. Since blister rust spore dispersal occurs atmospherically, any prevailing weather patterns might make blister rust infection in sites directly downwind from one another have higher degrees of correlation than for sites oriented perpendicular to weather patterns.

5 Graphical functions developed for this analysis

This analysis necessitated plotting of spatial point-referenced data with an additional (z) component - observed or predicted infection rates. For intuitive simplicity, this was expressed using a color gradient. However, graphical methods for expressing raster (grid) data typical of a GIS environment - that is, with an incomplete grid - were not easily obtained. Because of this and some additional functionality desired and the format of the results, custom functions were developed and made adaptable for other, similar analyses.

The `rasterblaster` function displays raster data as a colored grid, and simply requires a vector of (x,y) coordinates for the grid cell midpoints and a corresponding vector of quantitative data. It automatically detects the grid scale and computes a color ramp according to a user-specified type, and adds the options of a legend and scale bar. The `dotsplot` function displays point data, again with the input of (x,y) coordinates and a vector of quantitative data, also computing a color ramp and adding mapping elements such as a legend and scale bar.

With the relative simplicity of use of these functions for mapping spatial data, these functions and others will be published as the `rasterblaster` package, for the benefit of others performing similar analyses.

6 Acknowledgements

A deep debt of gratitude is owed to:

Dr. Megan Higgs, for the continued guidance,

Dr. Dan Tyers, for the opportunity and for the fabulous data set, and

All the field technicians that have been part of the data collection, for all our shared blood, sweat, tears, and ramen.

7 References

Ottar N. Bjornstad (2013). ncf: spatial nonparametric covariance functions. R package version 1.1-5. <http://CRAN.R-project.org/package=ncf>

Box & Jenkins. *Time Series Analysis: Forecasting and Control*. Holden-Day. 1976.

Christensen, O.F. & Ribeiro Jr., P.J. (2002) geoRglm: A package for generalised linear spatial models. R-NEWS, Vol 2, No 2, 26-28.

Diggle, P.J. & Ribeiro Jr, P.J. *Model Based Geostatistics*. Springer, New York, 2007.

Fortín & Dale. *Spatial Analysis: A Guide for Ecologists*. Cambridge 2005.

Mantel, N. (1967). “The detection of disease clustering and a generalized regression approach”. *Cancer Research* 27 (2): 209-220

Oksanen, Blanchet, Kindt, Legendre, Minchin, O’Hara, Simpson, Solymos, Stevens, and Wagner (2013). vegan: Community Ecology Package. R package version 2.0-10. <http://CRAN.R-project.org/package=vegan>

Paradis E., Claude J. & Strimmer K. 2004. APE: analyses of phylogenetics and evolution in R language. *Bioinformatics* 20: 289-290.

Jose Pinheiro, Douglas Bates, Saikat DebRoy, Deepayan Sarkar and the R Development Core Team (2012). nlme: Linear and Nonlinear Mixed Effects Models. R package version 3.1-105.

Ramsey & Schafer. *The Statistical Sleuth*, second edition. Duxbury 2002.

Paulo J. Ribeiro Jr & Peter J. Diggle geoR: a package for geostatistical analysis R-NEWS, 1(2):15-18. June, 2001.

Schabenberger & Gotway. *Statistical Methods for Spatial Data Analysis*. Chapman & Hall 2005.

8 Appendix - R Code Used

```
# Loading data
setwd("~/Matt/classes/Writing_project")
load("Project_Data_4_5.Rdata")
# Functions that will be used
logit <- function(x) log(x/(1-x))
expit <- function(x) (exp(x))/(1+exp(x))
rasterblaster.bw <- function(coords,z,x=coords[,1],y=coords[,2],zmin=min(z,na.rm=T)
                             ,zmax=max(z,na.rm=T)){
  xdismat <- dist(x)
  xdismat[xdismat==0] <- max(xdismat)
  xside <- min(xdismat)
  #print(xside)
  ydismat <- dist(y)
  ydismat[ydismat==0] <- max(ydismat)
  yside <- min(ydismat)
  #print(yside)
  plot(x,y,col="white",asp=1)
  for(i in 1:length(z)){
    if(is.na(z[i])==F){
      polygon(c(x[i]-xside/2,x[i]+xside/2,x[i]+xside/2,x[i]-xside/2),
              c(y[i]-yside/2,y[i]-yside/2,y[i]+yside/2,y[i]+yside/2),
              col=gray((z[i]-zmin)/(zmax-zmin)),border=NA)
    }
  }
}
rasterblaster <- function(coords,z,x=coords[,1],y=coords[,2],invert=F,scale=F,
                          scale.mult=1,scale.units="",scale.dist=10000,main="",
                          xlab="",ylab="",legend=T,leg.round=2,add=F){
  xdismat <- dist(x)
  xdismat[xdismat==0] <- max(xdismat)
  xside <- min(xdismat)
  #print(xside)
  ydismat <- dist(y)
  ydismat[ydismat==0] <- max(ydismat)
  yside <- min(ydismat)
  #print(yside)
  if(add==F) plot(x,y,col="white",asp=1,main=main,xlab=xlab,ylab=ylab)
  cols<-rainbow(length(z),start=0,end=0.8)
  zrank<-rank(z,na.last=T)
  if(invert==TRUE) {
    cols1<-rep(NA,length(z))
    for(i in 1:length(z)) {
      cols1[i] <- cols[length(z)-i+1]
    }
    cols<-cols1
  }
  for(i in 1:length(z)){
```

```

    if(is.na(z[i])==F){
      #polygon(c(x[i]-xside/2,x[i]+xside/2,x[i]+xside/2,x[i]-xside/2),
      #       c(y[i]-yside/2,y[i]-yside/2,y[i]+yside/2,y[i]+yside/2),
      #       col=cols[zrank[i]],border=NA)
      rect(x[i]-xside/2,y[i]-yside/2,x[i]+xside/2,y[i]+yside/2,col=cols[zrank[i]],
          border=NA)
    }
  }
}
if(legend==T) {
  for(i in 1:10) {
    polygon(c(min(x)-xside/2,min(x)+xside/2,min(x)+xside/2,min(x)-xside/2),
            c(max(y)-i*yside-yside/2,max(y)-i*yside-yside/2,max(y)-i*yside+yside/2,
              max(y)-i*yside+yside/2),col=cols[floor((10-i)/10*length(z))+1],
            border=NA)
    if(is.na(z[zrank==floor((10-i)/10*length(z))+1])==F){#floor(i/2)!=i/2 &
      text(x=min(x)+(floor(i/2)==i/2)*7*yside,y=max(y)-i*yside,
          round(z[zrank==floor((10-i)/10*length(z))+1],leg.round),pos=4,cex=.5)
    }
  }
}
}
if(scale==T) {
  lines(c(min(x),min(x)+2*scale.dist),rep(min(y),2))
  lines(rep(min(x),2),min(y)+c(-1000,1000))
  lines(rep(min(x)+scale.dist,2),min(y)+c(-1000,1000))
  lines(rep(min(x)+2*scale.dist,2),min(y)+c(-1000,1000))
  text(x=min(x)+scale.dist,y=min(y),scale.dist*scale.mult,pos=3,cex=.5)
  text(x=min(x)+2*scale.dist,y=min(y),2*scale.dist*scale.mult,pos=3,cex=.5)
  text(x=min(x)+2.5*scale.dist,y=min(y),scale.units,pos=3,cex=.5)
}
}
dotsplot <- function(x,y,z,main="",xlab="",ylab="",colramp="rainbow",invert=FALSE,
                    scale=F,scale.dist=10000,legend=T,add=F) {
  if(add==F) plot(x,y,col="white",asp=1,main=main,xlab=xlab,ylab=ylab)
  if(colramp=="rainbow") cols<-rainbow(length(z),start=0,end=0.75)
  if(colramp=="terrain") cols<-terrain.colors(length(z))
  if(colramp=="topo") cols<-topo.colors(length(z))
  if(invert==TRUE) {
    cols1<-rep(NA,length(z))
    for(i in 1:length(z)) {
      cols1[i] <- cols[length(z)-i+1]
    }
    cols<-cols1
  }
  zrank<-rank(z,na.last=T,ties.method="min")
  for(i in 1:length(z)) {
    points(x[i],y[i],pch=20,col=cols[zrank[i]])
  }
  if(legend==T) {
    for(i in 1:10) {
      points(min(x)-200,max(y)-1000*i,pch=20,col=cols[floor((10-i)/10*length(z))+1])
      #if(is.na(z[zrank==floor((10-i)/10*length(z))+1])==F){#floor(i/2)!=i/2 &
      # text(x=min(x)+(floor(i/2)==i/2)*7000,y=max(y)-i*1000,
      # round(z[zrank==floor((10-i)/10*length(z))+1],2),pos=4,cex=.5)
    }
  }
}

```

```

#}
}
text(min(x)-200,max(y)-1000,round(max(z),2),pos=4,cex=.5)
text(min(x)-200,max(y)-10000,round(min(z),2),pos=4,cex=.5)
}
if(scale==T) {
  lines(c(min(x),min(x)+2*scale.dist),rep(min(y),2))
  lines(rep(min(x),2),min(y)+c(-1000,1000))
  lines(rep(min(x)+scale.dist,2),min(y)+c(-1000,1000))
  lines(rep(min(x)+2*scale.dist,2),min(y)+c(-1000,1000))
  text(x=min(x)+scale.dist,y=min(y),"10 km",pos=3,cex=.5)
  text(x=min(x)+2*scale.dist,y=min(y),"20 km",pos=3,cex=.5)
}
}

```

```

library(ncf)
corr.bin<-correlog(x=site.data$UTM.E,y=site.data$UTM.N,z=resids,
  increment=1000,resamp=1)
#plot(corr.bin)
plot(corr.bin$mean.of.class[1:20],corr.bin$correlation[1:20],xlab="distance (m)",
  ylab="correlation",main="Correlogram - Deviance Residuals",type="b")
#lines(corr.bin$mean.of.class[1:100],corr.bin$correlation[1:100])
abline(h=0)
library(ade4)
library(ape)
dists<-as.matrix(dist(cbind(site.data$UTM.N, site.data$UTM.E)))
dists.inv <- 1/dists
diag(dists.inv) <- 0
dists.inv[is.infinite(dists.inv)] <- 0
dists.inv2 <- dists.inv^2
gearymoran(dists.inv,resids)
Moran.I(resids,dists.inv2)
v1.expon<-variofit(v1)
lines(v1.expon)
# prediction using GLM residuals...
site.coords <- data.frame(site.data$UTM.N,site.data$UTM.E)
set.seed(1)
site.coords.j <- data.frame((site.data$UTM.N+runif(1550,-20,20)),
  (site.data$UTM.E+runif(1550,-20,20)))
inf.rate <- site.data$prop.all
length(inf.rate[inf.rate==0])/length(inf.rate)
logit.inf <- log((inf.rate+.02)/(1-(inf.rate+.02)))
hist(inf.rate)
hist(logit.inf)
gridpts <- read.csv("gridpts_elev_aspect_slope.csv",head=T)
names(gridpts)
num.cat <- 10
elev.cat <- rep(NA,1550)
elev.q<- quantile(site.data$elev_km,seq(0,1,by=(1/num.cat)))
for(i in 1:num.cat) {
  elev.cat[(site.data$elev_km<elev.q[i+1])&(site.data$elev_km>=elev.q[i])] <- i
}

```

```

elev.cat[894] <- num.cat
plot(elev.cat,site.data$elev_km)
num.cat <- 10
slope.cat <- rep(NA,1550)
slope.q<- quantile(site.data$Slope.deg,seq(0,1,by=(1/num.cat)))
for(i in 1:num.cat) {
  slope.cat[(site.data$Slope.deg<slope.q[i+1])&(site.data$Slope.deg>=slope.q[i])] <- i
}
slope.cat[1463] <- num.cat
plot(slope.cat,site.data$Slope.deg)
site.data$slope.cat <- slope.cat
site.data$elev.cat <- elev.cat
library(geoR)
library(geoRglm)
# setting up the prediction grid points
gr.elev.cat <- rep(NA,5959)
gr.slope.cat <- rep(NA,5959)
num.cat<-10
gridpts$elev_km <- gridpts$elev_m/1000
for(i in 1:num.cat) {
  gr.elev.cat[(gridpts$elev_km<elev.q[i+1])&(gridpts$elev_km>=elev.q[i])] <- i
}
plot(gr.elev.cat,gridpts$elev_km)
for(i in 1:num.cat) {
  gr.slope.cat[(gridpts$slope<slope.q[i+1])&(gridpts$slope>=slope.q[i])] <- i
}
plot(gr.slope.cat,gridpts$slope)
gr.NSEW <- rep(NA,5959)
gr.NSEW <- as.character(ifelse((gridpts$aspect<45|gridpts$aspect>=315),"N",
                              ifelse((gridpts$aspect<135&gridpts$aspect>=45),"E",
                              ifelse((gridpts$aspect<225&gridpts$aspect>=135),
                              "S",ifelse((gridpts$aspect<315&
                              gridpts$aspect>=225),"W",0))))))
gr.NSEW[gridpts$slope==0] <- 0
gr.NSEW <- as.factor(gr.NSEW)
#slope.cat[is.na(slope.cat)==T] <- -1
#elev.cat[is.na(elev.cat)==T] <- -1
#gr.slope.cat[is.na(gr.slope.cat)==T] <- -1
#gr.slope.cat[17] <- -1
#gr.elev.cat[is.na(gr.elev.cat)==T] <- -1
slope.cat <- as.factor(slope.cat)
elev.cat <- as.factor(elev.cat)
gr.slope.cat <- as.factor(gr.slope.cat)
gr.elev.cat <- as.factor(gr.elev.cat)
gridpts$slope.cat<-gr.slope.cat
gridpts$elev.cat<-gr.elev.cat
gridpts$NSEW<-gr.NSEW
#inserting the new model residuals, i hope it works
mod.cat<-glm(cbind(subset.blist,subset.noblist)~
  factor(slope.cat)+factor(elev.cat)*NSEW,data=site.data,family=binomial(link="logit"))
#hist(mod.cat$residuals)
site.resids <- data.frame(site.coords.j,(resid(mod.cat,type="deviance")))
#predicting the deterministic part

```

```

hist(logit(mod.cat$fitted))
#hist(fitted(mod.cat,type="link"))
grid.determ <- predict.glm(mod.cat,newdata=gridpts.c,se.fit=T)
hist(grid.determ$fit)
hist(grid.determ$se.fit)
#site.resids <- data.frame(site.coords.j,mod10$residuals)
names(site.resids) <- c("N","E","resid")
site.resids <- as.geodata(site.resids,coords.col=2:1,data.col=3)
plot(variogram(site.resids),type = "b")
ml <- likfit(site.resids, ini=c(5, 3000), nug=0)
# likfit: estimated model parameters:
#   beta      tausq      sigmasq      phi
# "  0.4845" "  1.2083" "  1.6398" "5035.5085"
# "  0.4835" "  1.2077" "  1.6342" "5000.0000"
# "  0.4846" "  1.2083" "  1.6398" "5035.7233"
# Practical Range with cor=0.05 for asymptotic range: 15085.04
# likfit: maximised log-likelihood = -2621
# likfit: maximised log-likelihood = -2621
# likfit: maximised log-likelihood = -2621
## performing the spatial prediction
grid.loc <- as.matrix(cbind(gridpts$Var1,gridpts$Var2))
KC <- krige.control(type="sk", obj.mod=ml)
kr <- krige.conv(site.resids, loc=grid.loc, krige=KC)
kr.c <- krige.conv(site.resids, loc=grid.loc.c, krige=KC)
#borders=cbind(ab.poly.pts$site.data.UTM.E,ab.poly.pts$site.data.UTM.N),
str(kr)
hist(kr$predict)
hist(sqrt(kr$krige.var))
plot(site.resids)
plot(grid.loc,col="white")
points(grid.loc,col=gray((kr$predict-min(kr$predict))/
(range(kr$predict)[2]-range(kr$predict)[1])))
grid.pred <- grid.determ$fit + kr.c$predict
hist(grid.pred)
grid.err <- sqrt(kr.c$krige.var+(grid.determ$se.fit^2))
hist(grid.err)
plot(site.resids)
lo.pred.1se <- grid.pred-grid.err
hi.pred.1se <- grid.pred+grid.err
lo.pred.95 <- grid.pred-2*grid.err
hi.pred.95 <- grid.pred+2*grid.err
hist(lo.pred.95)
hist(hi.pred.95)
hist(expit(lo.pred.95))
hist(expit(hi.pred.95))
hist(lo.pred.1se)
hist(hi.pred.1se)
hist(expit(lo.pred.1se))
hist(expit(hi.pred.1se))
p.grid.pred <- expit(grid.pred)
p.lo.pred.1se <- expit(lo.pred.1se)
p.hi.pred.1se <- expit(hi.pred.1se)
#The Bayesian Version

```

```

grid.loc <- as.matrix(cbind(gridpts$Var1,gridpts$Var2))
site.binom <- data.frame(site.coords.j,site.data$subset.blist,site.data$subset)
names(site.binom) <- c("N","E","blist","all")
site.binom <- as.geodata(site.binom,coords.col=2:1,data.col=3)
site.binom$units.m <- site.data$subset
#try a subset
sub.poly<-cbind(c(600000,620000,620000,600000),c(5010000,5010000,4990000,4990000))
site.binom.sub <- subarea(site.binom,xlim=c(600000,620000),ylim=c(4990000,5010000))
grid.loc.sub <- polygrid(grid.loc, bor=sub.poly)
grid.loc.subsub <- grid.loc.sub[c(123,135),]
plot(grid.loc.sub)
points(grid.loc.subsub,pch=2)
points(site.binom.sub,add=T)
MCb <- mcmc.control(n.iter=10000,S.scale=.2,phi.scale=.6,burn.in=1000)
OCb <- output.glm.control(sim.predict=T)
PCb <- prior.glm.control(phi.prior="uniform",sigmasq.prior="uniform",
                        phi.discrete=seq(4000,6000,50))
set.seed(268)
kr.b <- binom.krige.bayes(site.binom.sub, loc=grid.loc.sub, mcmc = MCb,prior = PCb)
#convergence
par(mfrow=c(2,2))
plot(kr.b$posterior$phi$sample,type="l")
acf(kr.b$posterior$phi$sample)
plot(kr.b$posterior$sim[1,],type="l")
acf(kr.b$posterior$sim[1,])
par(mfrow=c(1,1))
hist(kr.b$predictive$median)
str(kr.b)
hist(kr.b$predictive$simulations[300,])
grid.medians <- rep(NA,dim(grid.loc.sub)[1])
for(i in 1:dim(grid.loc.sub)[1]) {
  grid.medians[i] <- median(kr.b$predictive$simulations[i,])
}
hist(grid.medians)
rasterblaster.col(grid.loc.sub,grid.medians)
points(site.binom.sub$coords,pch='+')
rasterblaster.col(grid.loc.sub,kr.b$predictive$median)
points(site.binom.sub$coords,pch='+')
hist(kr.b$posterior$simulations[70,])
abline(v=site.binom.sub$data[70]/site.binom.sub$units.m[70])
#try the whole thing
plot(grid.loc)
points(site.binom,col=2,add=T)
MCb <- mcmc.control(n.iter=1000,S.scale=.01,phi.scale=.6,burn.in=1000)
OCb <- output.glm.control(sim.predict=T)
PCb <- prior.glm.control(phi.prior="fixed",sigmasq.prior="uniform",phi=5000)
set.seed(268)
kr.b.full <- binom.krige.bayes(site.binom, loc=grid.loc, mcmc = MCb,prior = PCb)
rasterblaster.col(grid.loc,kr.b.full$predictive$median)
rasterblaster.col(grid.loc,kr.b.full$predictive$quantiles$q0.025)
rasterblaster.col(grid.loc,kr.b.full$predictive$quantiles$q0.975)
rasterblaster.col(grid.loc,kr.b.full$predictive$uncertainty)
points(site.binom$coords,pch='+')

```

```

rasterblaster(grid.loc,kr.b.full$predictive$median)
#try the whole thing plus predictors
gridpts.c <- na.omit(gridpts)
grid.loc.c <- as.matrix(cbind(gridpts.c$Var1,gridpts.c$Var2))
plot(grid.loc.c)
points(site.binom,col=2,add=T)
MCb <- mcmc.control(n.iter=1000,S.scale=.005,phi.scale=.6,burn.in=1000)
OCb <- output.glm.control(sim.predict=F)
PCb <- prior.glm.control(phi.prior="fixed",sigmasq.prior="uniform",phi=5000)
ModCb <- model.glm.control(trend.d= ~factor(site.data$slope.cat)+
                           factor(site.data$elev.cat)*factor(site.data$NSEW),
                           trend.l= ~factor(gridpts.c$slope.cat)+
                           factor(gridpts.c$elev.cat)*factor(gridpts.c$NSEW))

set.seed(268)
kr.b.full.c <- binom.krige.bayes(site.binom, loc=grid.loc.c, mcmc = MCb,
                               prior = PCb,model=ModCb,output=OCb)

#try it again
set.seed(149)
kr.b.full.c1 <- binom.krige.bayes(site.binom, loc=grid.loc.c, mcmc = MCb,
                                  prior = PCb,model=ModCb,output=OCb)

rasterblaster.col(grid.loc.c,kr.b.full.c$predictive$median,invert=T)
rasterblaster.col(grid.loc.c,kr.b.full.c$predictive$quantiles$q0.025)
rasterblaster.col(grid.loc.c,kr.b.full.c$predictive$quantiles$q0.975)
rasterblaster.col(grid.loc.c,kr.b.full.c$predictive$uncertainty)
points(site.binom$coords,pch='+')

```

```

plot(site.data$UTM.E,site.data$UTM.N,col="white",xlab="UTM E",ylab="UTM N",
     main="1995-1997")
for(i in 1995:1997) {
  points(site.data$UTM.E[site.data$Year==i],site.data$UTM.N[site.data$Year==i])
}
plot(site.data$UTM.E,site.data$UTM.N,col="white",xlab="UTM E",ylab="UTM N",
     main="1998-1999")
for(i in 1998:1999) {
  points(site.data$UTM.E[site.data$Year==i],site.data$UTM.N[site.data$Year==i])
}
plot(site.data$UTM.E,site.data$UTM.N,col="white",xlab="UTM E",ylab="UTM N",
     main="2000-2002")
for(i in 2000:2002) {
  points(site.data$UTM.E[site.data$Year==i],site.data$UTM.N[site.data$Year==i])
}
plot(site.data$UTM.E,site.data$UTM.N,col="white",xlab="UTM E",ylab="UTM N",
     main="2003-2004")
for(i in 2003:2004) {
  points(site.data$UTM.E[site.data$Year==i],site.data$UTM.N[site.data$Year==i])
}
plot(site.data$UTM.E,site.data$UTM.N,col="white",xlab="UTM E",ylab="UTM N",
     main="2005-2007")
for(i in 2005:2007) {
  points(site.data$UTM.E[site.data$Year==i],site.data$UTM.N[site.data$Year==i])
}
plot(site.data$UTM.E,site.data$UTM.N,col="white",xlab="UTM E",ylab="UTM N",

```

```

    main="2008-2010")
for(i in 2008:2010) {
  points(site.data$UTM.E[site.data$Year==i],site.data$UTM.N[site.data$Year==i])
}

```

```

dotsplot(site.data$UTM.E,site.data$UTM.N,site.data$prop.all,
  main="Raw Infection Rates",invert=T,scale=T)

```

```

#dotsplot(site.data$UTM.E,site.data$UTM.N,resid(mod.cat,type="deviance"),
#main="Deviance Residuals from Top Model",invert=T,scale=T)
mod1 <- glm(prop.all~Ribies+Cov_class_1+NSEW+elev_km+year_c+Slope.deg.,
  data=site.data,family=binomial(link="logit"),weights=subset)
mod1a <- glm(prop.all~Ribies+Cov_class_1+NSEW*elev_km+year_c+NSEW*Slope.deg.,
  data=site.data,family=binomial(link="logit"),weights=subset)
dotsplot(site.data$UTM.E,site.data$UTM.N,resid(mod1a,type="deviance"),
  main="Deviance Residuals",invert=T,scale=T)
resids<-resid(mod1a,type="deviance")
#resids<-resid(mod.cat,type="deviance")

```

```

library(ade4)
library(ape)
library(vegan)
#library(spdep)
dists<-as.matrix(dist(cbind(site.data$UTM.N, site.data$UTM.E)))
dists.inv <- 1/dists
diag(dists.inv) <- 0
dists.inv[is.infinite(dists.inv)] <- 0
dists.inv2 <- dists.inv^2
Moran.I(resids,dists.inv2)
#geary.test(resids,dists.inv2)
gearymoran(dists.inv2,resids)
resid.dist<-dist(resids)
mantel(resid.dist,dists,permutations=999)

```

```

plot(corr.bin$mean.of.class[1:21],corr.bin$correlation[1:21],xlab="distance (m)",
  ylab="correlation",main="Correlogram - Deviance Residuals")
abline(h=0,lty=2)
#library(nlme)
library(geoR)
#glsfit<-gls(resids~1)
#plot(Variogram(glsfit,form=~site.data$UTM.E+site.data$UTM.N,max=20000))
#plot(Variogram(object=resids,distance=dists,max=20000))
breaks = seq(0, 20000, l = 21)
v1 <- variog(coords=site.binom$coords,data=resids, breaks = breaks)
plot(v1,main="Semivariogram - Deviance Residuals")

```

```

plot(v1,main="Semivariogram - Deviance Residuals")
v1.spher <- variofit(v1,cov.model="spherical")
lines(v1.spher,lty=6)
# parameter estimates:
#   tausq   sigmasq   phi
#   3.5088   3.7335 18464.6988
# Practical Range with cor=0.05 for asymptotic range: 18464.7
v1.pow.exp <- variofit(v1,cov.model="powered.exponential")
lines(v1.pow.exp,lty=7)
#parameter estimates:
#   tausq   sigmasq   phi
#   2.5345   35.6044 864702.6670
#Practical Range with cor=0.05 for asymptotic range: 7760198
v1.gauss <- variofit(v1,cov.model="gaussian")
lines(v1.gauss,lty=2)
#parameter estimates:
#   tausq   sigmasq   phi
#   4.1253   3.2673 9865.6747
#Practical Range with cor=0.05 for asymptotic range: 17075.69
legend(5000,2,legend=c("Spherical", "Powered Exponential", "Gaussian"),lty=c(6,7,2))
plot(v1,main="Semivariogram - Deviance Residuals")
v1.expon <- variofit(v1,cov.model="exponential")
lines(v1.expon)
#parameter estimates:
#   tausq   sigmasq   phi
#   3.4662   5.3155 13979.6978
#Practical Range with cor=0.05 for asymptotic range: 41879.43
v1.matern1 <- variofit(v1,cov.model="matern",kappa=1)
lines(v1.matern1,lty=3)
#parameter estimates:
#   tausq   sigmasq   phi
#   3.8585   4.1806 7015.1962
#Practical Range with cor=0.05 for asymptotic range: 28050.42
v1.matern2 <- variofit(v1,cov.model="matern",kappa=2)
lines(v1.matern2,lty=5)
#parameter estimates:
#   tausq   sigmasq   phi
#   4.0233   3.6987 4210.0323
#Practical Range with cor=0.05 for asymptotic range: 22601.03
legend(5000,2,legend=c("Matern, k=0.5", "Matern, k=1", "Matern, k=2"),lty=c(1,3,5))

```

```

library(nlme)
set.seed(1)
site.data$UTM.N.j<-site.data$UTM.N+runif(1550,-20,20)
site.data$UTM.E.j<-site.data$UTM.E+runif(1550,-20,20)
gls.mod<-gls(logit(prop.all)~Ribies+Cov_class_1+NSEW+elev_km+year_c+Slope.deg.,
             data=site.data,corr=corSpher(form=~UTM.E.j+UTM.N.j))

```

```

lo.pred.50 <- grid.pred-0.675*grid.err
hi.pred.50 <- grid.pred+0.675*grid.err
lo.pred.95 <- grid.pred-2*grid.err

```

```
hi.pred.95 <- grid.pred+2*grid.err
rasterblaster(grid.loc.c,expit(grid.pred),invert=T,scale=T,scale.units="km",
              scale.mult=0.001,main="Predicted Probabilities of Infection")
```

```
rasterblaster(grid.loc.c,kr.c$krige.var,invert=T,scale=T,scale.units="km",
              scale.mult=0.001,main="Kriging Variance")
points(site.data$UTM.E,site.data$UTM.N)
```

```
#rasterblaster(grid.loc.c,expit(lo.pred.50),invert=T,scale=T,
#main="Predicted Probabilities of Infection - lower 50% interval")
#rasterblaster(grid.loc.c,expit(hi.pred.50),invert=T,scale=T,
#main="Predicted Probabilities of Infection - upper 50% interval")
rasterblaster(grid.loc.c,expit(lo.pred.95),invert=T,scale=T,main="Predicted
              Probabilities of Infection - lower 95% interval",leg.round=4)
rasterblaster(grid.loc.c,expit(hi.pred.95),invert=T,scale=T,main="Predicted
              Probabilities of Infection - upper 95% interval",leg.round=4)
```

```
rasterblaster(grid.loc.c,kr.b.full.c$predictive$median,invert=T,scale=T,
              main="Predicted Probabilities of Infection")
```

```
rasterblaster(grid.loc.c,kr.b.full.c$predictive$quantiles$q0.025,invert=T,scale=T,main
              ="Predicted Probabilities of Infection - lower 95% interval",leg.round=6)
rasterblaster(grid.loc.c,kr.b.full.c$predictive$quantiles$q0.975,invert=T,scale=T,main
              ="Predicted Probabilities of Infection - upper 95% interval",leg.round=4)
```

```
rasterblaster(grid.loc.c,as.numeric(gridpts.c$slope.cat),legend=F,
              main="Slopes Extracted from DEM",invert=T)
```