Ecological Boundary Detection Using Bayesian Areal Wombling

ECOLOGICAL BOUNDARY DETECTION USING BAYESIAN AREAL WOMBLING

Matthew C. Fitzpatrick*[,1,2,3], Evan L. Preisser[2], Adam Porter[4], Joseph Elkinton[4], Lance A. Waller[5], Bradley P. Carlin[6], Aaron M. Ellison[3]

[1] *University of Maryland Center for Environmental Science, Appalachian Lab, Frostburg, MD 21532*

[2]*Department of Biological Sciences, University of Rhode Island, Kingston, RI 02881*

[3]*Harvard University, Harvard Forest, Petersham, MA 01366*

[4]*Department of Plant, Soil, and Insect Sciences, University of Massachusetts at Amherst, Amherst, Massachusetts*

[5]*Department of Biostatistics, Rollins School of Public Health, Emory University*

[6]*Division of Biostatistics, School of Public Health, University of Minnesota*

*Author to whom correspondence should be addressed:

Matt Fitzpatrick

University of Maryland Center for Environmental Science

Appalachian Lab

301 Braddock Road, Frostburg, MD 21532

e-mail: mfitzpatrick@umces.edu; phone: (301) 689-7131; fax: (301) 689-7200

Manuscript type: *Statistical Report*

ABSTRACT

The study of ecological boundaries and their dynamics is of fundamental importance to much of ecology, biogeography, and evolution. Over the past two decades, boundary analysis (often termed wombling) has received considerable research attention, resulting in multiple approaches for the quantification of ecological boundaries. Nonetheless a number of issues remain unresolved, notably the inability of most methods to (*i*) analyze spatially-homogenized datasets (i.e., areal data in the form of polygons rather than point-reference data); (*ii*) account for spatial structure in these data and uncertainty associated with them; and (*iii*) objectively assign probabilities to boundaries once detected. Here we describe a method for ecological boundary detection used in public health that employs a Bayesian hierarchical framework and which addresses these issues. As examples, we analyze simulated data and the historic pattern of spread of an invasive species, the hemlock woolly adelgid (*Adelges tsugae*), across eastern North America, using county-level dates of first infestation and several covariates potentially important to influencing the observed spread dynamics.

KEYWORDS: boundary analysis, ecotones, edge detection, invasive species, spatial statistics

INTRODUCTION

A central challenge in ecology is determining the factors influencing species distributions and how these factors change across space and time (Holt and Keitt 2005). The increasingly serious threats to natural systems posed by global change emphasize the practical importance of identifying the environmental factors associated with range edges (e.g., Gavin and Hu 2006) and of determining how environmental changes may affect movement of both native and invasive

47 species across heterogeneous landscapes. At its core, understanding the dynamics of species
48 distributions is both a statistical problem of identifying boundaries between where a species is
49 present (or abundant) and absent (or rare), and an ecological problem of determining
50 environmental factors associated with these boundaries (Gaston 2003, Fortin et al. 2005).

51 Two major challenges limit detailed analysis of ecological and evolutionary processes
52 underlying the formation, persistence, and change of range edges. First, the spatiotemporal data
53 required for inference are lacking (Parmesan et al. 2005) or when available, are often spatially
54 homogenized as summaries over geopolitical or ecological regions such as counties, states, or
55 biomes. Such aggregation obscures fine-scale spatiotemporal characteristics in the data. Second,
56 data arising from neighboring regions are often more highly correlated than those from distant
57 neighbors. The spatial structure inherent in the data is often of ecological interest, but must be
58 accounted for to make valid inferences (Legendre 1993). Acknowledging spatial structure is
59 particularly important when considering the spread of invasive species because ecological
60 dynamics are inherently correlated in space and time.

61 Over the last decade a large body of ecological research has addressed boundary analysis
62 (sometimes called 'wombling' in recognition of William H. Womble, a pioneer in the field,
63 Womble 1951), with a corresponding increase in the number of analytical approaches available
64 for detecting and analyzing boundaries (see Jacquez et al. 2000 and Fagan et al. 2003 for recent
65 reviews and Jacquez et al. 2008 for a recent special issue on the topic). Wombling is a technique
66 for determining zones of abrupt change on a spatial surface that separate areas of lower and
67 higher values of a georeferenced unit (Fortin and Dale 2005). A common secondary concern is to
68 assign statistical significance or probabilities to the identified boundaries. At present, much of
69 the published literature on boundary analysis in ecology considers point-referenced data (i.e.,

70 geostatistical data comprised of spatial locations of points with known coordinates, such as
71 latitude-longitude) that are either regularly (lattice or grid) or irregularly spaced. Although point-
72 referenced data are becoming increasingly accessible (Graham et al. 2004), ecological data
73 covering broad spatial and temporal scales are more commonly available as summaries over
74 geographic regions. For example, herbaria data and records from the USDA PLANTS database
75 (http://plants.usda.gov) are provided as county- or state-level summaries. Boundary analysis of
76 such data, often term *areal* data, is well-developed in public health fields, but it has received
77 minimal attention in ecology. Further, most of the boundary analysis approaches in current use in
78 ecology assign significance or probabilities to detected boundaries using null distributions or
79 arbitrary thresholds; such inferences are relative to predetermined and often subjective choices.
80 Here we describe a promising technique for ecological analysis of areal data developed
81 by public heath researchers (e.g., Lu and Carlin 2005, Ma et al. 2006, Wheeler and Waller 2008)
82 that has as yet seen little use by ecologists. The method employs a Bayesian hierarchical
83 framework that (*i*) uses areal data; (*ii*) accounts for spatial structure in these data and the spatial
84 and nonspatial uncertainty associated with them; and (*iii*) provides a natural means of assigning
85 probabilities to boundaries using posterior estimates of the modeled parameters. As an example,
86 we analyze the historic pattern of spread of an invasive species, the hemlock woolly adelgid
87 ('HWA', *Adelges tsugae* Annand). Although this pest threatens hemlock forests (both eastern
88 hemlock, *Tsuga canadensis* (L.) Carr., and Carolina hemlock, *Tsuga caroliniana* Englemann, are
89 susceptible) throughout eastern North America (Orwig et al. 2002) and is of great concern to
90 both researchers and land managers, data on HWA spread exists primarily as county-level data
91 documenting the first reported HWA infestation in that area. Our goal is to strengthen links
92 between observed spread pattern and underlying ecological processes by identifying boundaries

93 across which spread is slower than expected and to determine whether such boundaries are
94 associated with environment features.
95
96 METHODS
97 *Study system* – HWA is a small (1 mm adult) flightless insect native to Asia that was first
98 collected from hemlock in the eastern United States in spring of 1951, in Richmond, VA. New
99 HWA infestations were collected next in Philadelphia, Pennsylvania in 1969, followed by
100 counties southwest of Richmond, VA (Fig. 1a, see Appendix A for a detailed description of these
101 data). The observed pattern of county-level spread following these early events largely mimics a
102 diffusive process although outlying infestations also have appeared in northwestern New York
103 State. As an exploratory tool, ordinary kriging on the county-level spread pattern (Fig. 1b) shows
104 slow initial spread from the three distinct early infestations, followed by spread to the northeast
105 and southwest. Compressed contours along the Appalachian Mountains suggest that
106 environmental or topographic aspects of this feature may be associated with reduction of spread
107 rate to the west. In contrast, spread has been relatively rapid in the southeastern Appalachians,
108 where contours are spaced broadly (Fig. 1b), suggesting topography alone may not influence
109 spread rate. Despite their proximity to the initial infestation, counties south of Richmond, VA
110 remain uninfested presumably because of a lack of hemlock.
111 Although population and dispersal dynamics of HWA remain poorly understood, we
112 expect the pattern of spread to be a function of both environmental and social factors.
113 Environmental factors such as hemlock abundance and winter temperature (Paradis et al. 2008,
114 Trotter and Shields 2009) may alter spread rate by influencing population and dispersal
115 dynamics. Social factors such as human population density may influence the pattern of spread

both by altering the environment (e.g., by reducing forest cover or planting hemlocks as landscape trees) and by influencing the detection and reporting of HWA infestations. To account for these processes, we generated a set of covariates for each county that could influence the spread and detection of the advancing HWA front, including mean winter temperature, human population density, and hemlock abundance (See Appendix A for details regarding the calculation of these variables). We did not consider physical barriers to spread such as rivers or mountains (e.g., Wheeler and Waller 2008) in this analysis because passive dispersal of HWA by wind and birds is unlikely to be directly influenced by such features at the county level.

Bayesian areal wombling – We follow recent work by Lu and Carlin (2005) and use a Bayesian hierarchical model to perform areal wombling. Wheeler and Waller (2008) extended Lu and Carlin's (2005) research on human disease incidence to the spread of rabies using county-level reporting of rabid raccoons. Following Wheeler and Waller (2008), we modeled Y_i, the number of months elapsed between the first reported HWA infestation in 1951 and the first reported HWA infestation in each county i as

$$Y_i \sim \mathrm{N}\left(\mu_i, \frac{1}{\tau}\right), \qquad (1)$$

where

$$\mu_i = \alpha + \boldsymbol{x}_i\beta + \phi_i \qquad (2)$$

is the expected number of months elapsed to first reported HWA infestation in county i, α is an intercept, τ is the precision, $\boldsymbol{x}_i$ is a vector of the covariates, and ϕ_i is a spatial random effect. The spatial random effect ϕ_i is given an intrinsic conditionally autoregressive (CAR) prior expressed as

$$\phi \sim \mathrm{CAR}(\tau_C), \quad (3)$$

$$\phi \mid \phi_{j \neq i} \sim \mathrm{N}\left(\bar{\phi}_i, \frac{1}{(\tau m_i)}\right), \quad (4)$$

where m_i is the number of counties neighboring county i and τ_C is the precision. The use of a CAR prior for the random effects serves two functions. Foremost, invasive spread is a spatial process, with neighboring counties more similar in date of first infestation than distant counties. Second, the CAR prior provides a degree of spatial smoothing and thereby may prevent the erroneous detection of barriers that arise from spurious departures from the overall spatial trend. For example, uncertainty in detection and therefore reporting of HWA infestations could be higher in counties where HWA populations remain at low densities (Fitzpatrick et al. 2009) because of scarcity of hemlock or where winter temperatures cause high mortality (Paradis et al. 2008, Trotter and Shields 2009). In our analysis, we consider counties to be neighbors if they share a common boundary; more sophisticated choices such as inverse distance weighting warrant investigation.

The above framework provides a smoothed expected value for the number of months to first HWA infestation in each county. Although spread rate is itself of ecological interest, our goal is to identify barriers that separate counties with substantially different times to first infestation and to assign probabilities to these boundaries. A boundary likelihood value (BLV) for boundary (i, j) can be defined as the absolute difference in months (Lu and Carlin 2005) of first HWA infestation reported in neighboring counties i and j as,

$$\Delta_{ij} = \left| Y_i - Y_j \right|. \quad (5)$$

Estimates of Δ_{ij} can be obtained using a Markov chain Monte Carlo (MCMC) algorithm to draw G samples of the modeled response $\mu_i^{(g)}$, $g = 1, \ldots, G$ from the posterior distribution $\mathrm{p}(\mu_i|\mathbf{y})$ for

each county i and each MCMC iteration g to obtain

$$\Delta_{ij}^{(g)} = \left| \mu_i^{(g)} - \mu_j^{(g)} \right|. \qquad (6)$$

Boundary probabilities are then determined by simply counting the number of samples of $\Delta_{ij}^{(g)}$ that exceed a threshold c, where c is some number of months. For example, if we wanted to know which county boundaries were associated with preventing spread for five years (i.e., difference in date of first detected HWA between adjacent counties is five years), c would equal 60 months. The boundary probability is then simply the ratio of this count ($\#\Delta_{ij}^{(g)} > c$) to the total number of samples G (2000 in our analyses), or

$$\hat{p}_{ij} \equiv \hat{P}(\Delta_{ij} > c \mid \boldsymbol{y}) = \frac{\#\Delta_{ij}^{(g)} > c}{G}. \qquad (7)$$

This approach to determining boundary probabilities is known as fuzzy wombling. Alternatively, crisp wombling can be performed if boundaries are assigned a value of 1 when the BLV exceeds some predetermined threshold (e.g., 0.5) or 0 otherwise.

Although BLVs based on the expected values μ_i offer one means of investigating boundary probabilities, a potentially more informative approach is to calculate BLVs using the spatial random effects ϕ_i. In essence, the ϕ_i can be interpreted as spatial residuals. High-probability boundaries based on residuals delineate adjacent regions that differ in their unmodeled heterogeneity and thus highlight regions where the covariates do not explain detected boundaries. In contrast, if no significant boundaries exist in a map of residual-based boundaries, then the covariates explain (or are at least correlated with factors that explain) detected boundaries. Close examination of boundary probabilities based on spatial residuals could prove extremely useful in ecological studies where the goal is to elucidate the factors determining range edges and how these vary across space.

182 The model described above can be fit in WinBUGS (Spiegelhalter et al. 2003) and output
183 analyzed and plotted in R (R Development Core Team 2009). For all models described below we
184 used a burn-in period of 100,000 iterations and an additional 100,000 iterations were used to
185 estimate model parameters. For calculation of BLVs, we subsampled 2000 iterations from the
186 posterior distributions of μ and ϕ. We assessed model convergence using the Gelman-Rubin
187 potential scale reduction statistic (Brooks and Gelman 1998). Details of model construction and
188 selection of priors are available from the code provided in Appendix B.
189
190 EXAMPLE ANALYSES
191 *Simulation study* – Our first example considers an analysis of simulated county-level spread data.
192 We simulated, with added noise, the number of months to first infestation as a linear function of
193 distance from Richmond, VA (Fig. 2a). By design, counties surrounding York County,
194 Pennsylvania do not follow this pattern (Fig. 2b). Because distance from Richmond should not
195 explain the detected boundaries around these outlier counties, even after smoothing, we expect
196 high probability boundaries in the vicinity of York County, PA for both μ- and ϕ-based BLVs.
197 We found the expected pattern: nearly all of the detected boundaries (Fig. 2c) are explained by
198 the covariate other than those surrounding York County, Pennsylvania (Fig. 2d).
199
200 *Historic spread of HWA* – A model fit to the observed HWA spread data incorporated three
201 covariates: human population density, mean winter temperature, and hemlock abundance. This
202 model suggests several features of the spread of HWA (Fig. 3a). Most notably, boundary
203 probabilities are highest (1) in the vicinity of counties where HWA first established and where
204 spread may have been slow due to lag effects (Kowarik 1995) related to HWA population

dynamics, (2) along ridges of the Appalachian Mountains north of Tennessee, and (3) in the northernmost portions of HWA's range in New England. In contrast there are few barriers south of Virginia's southern border, where spread has been rapid. However, mean winter temperature and hemlock abundance are not significantly associated with barriers to spread; only the coefficient for human population density emerged as significantly different from zero. Except in for some northern counties and those in central Pennsylvania, boundary probabilities based on the spatial residuals (Fig. 3b) largely reflect those calculated using the expected value μ (Fig. 3a).

In retrospect, the failure of temperature and hemlock abundance to explain barriers to spread may not be surprising. Global covariates, though useful in detecting and visualizing boundaries, do not couple regional heterogeneity in environmental conditions to local barriers to spread. For example, HWA can spread rapidly under warm temperatures only where hemlock is available. In addition, spread patterns are strongly a function of where propagules are first introduced. In the case of HWA, the earliest dates of infestation are found in counties with little or no naturally-occurring hemlock.

To better model the landscape influences that hinder spread, Bayesian spatially-varying coefficient models (Banerjee et al. 2004) can be used for wombling (e.g., Wheeler and Waller 2008), although these models offer greater technical challenges. Alternatively, rather than modeling the data arising from areal units, wombling can be performed on the county borders themselves (Ma et al. 2006, Ma et al. 2009). In this approach, every boundary segment is a data point and the response for each segment is the difference in the modeled value of interest between adjacent units. In the context of invasive spread, 'local edge wombling' is likely to be ecologically more sensible because differences (or similarities) between adjacent areal units may

be more important for, and therefore may better explain, spread dynamics than mean values of covariates within counties. This approach also provides a more straightforward means to represent physical barriers such as rivers, mountains or urban areas as binary indicator variables.

We modified our model (equations 1-3) for local edge wombling by examining the difference in months to first infestation between adjacent counties:

$$D_{ij} = Y_i - Y_j, \quad (8)$$

$$D_{ij} \sim \mathrm{N}\left(\delta_{ij}, \frac{1}{\tau}\right), \; i \text{ adjacent to } j, \quad (9)$$

where

$$\delta_{ij} = \alpha + \boldsymbol{x_{ij}}\beta + \psi_{ij}. \quad (10)$$

As before, a spatial random effect (ψ) is included and is given a CAR prior. The vector of covariates $\boldsymbol{x_{ij}}$ in this version represents *differences* in covariates across borders and/or indicators variables corresponding to known barriers. Because the response is the difference in months to first infestation across borders, the calculation of BLVs is simplified slightly because they are determined using the absolute values of the posterior estimates of δ_{ij} (or ψ_{ij}) themselves (as opposed to *post hoc* calculation of these differences, Eq. 6) using a constant c. Code for fitting this model is provided in Appendix B.

A local edge wombling model incorporating as covariates differences in population density, mean winter temperature, and hemlock abundance across county borders reveals similar results to those derived from the areal wombling model: high probability boundaries are concentrated in the east and northeast (Fig. 4a, c). However, the covariates in the local edge wombling model have more influence on the detected boundaries for BLV thresholds of both three (Fig. 4b) and five years (Fig. 4d). The coefficients for hemlock abundance and population

density are significantly from zero. As before, boundaries associated with early spread in the eastern portion of the study region remain after accounting for the effects of the covariates, potentially reflecting demographic lag effects unrelated to environmental factors (Kowarik 1995).

CONCLUSIONS

Bayesian areal wombling is promising approach for analyzing ecological boundaries and the spread of invasive species. Many other applications for areal wombling can be envisioned. For example, wombling is commonly used in public health research to identify boundaries where disease incidence is higher/lower than expected. The same principle can be applied in ecology to understand patterns of both invasive species richness and distribution as well as patterns of distribution and abundance of native species. Important targets for future improvement of these models in ecology include exploration of alternate parameterizations for spatial smoothing, such as distance weighting or to estimate smoothing parameters from the data (Ma et al. 2009).

The strengths of wombling in a Bayesian framework should be clear. Beyond making good use of data with relatively coarse spatial and temporal resolution – data commonly available to ecologists – the Bayesian model easily incorporates uncertainty and provides a natural means of assigning probabilities to detected boundaries. Although there is not yet a single software package or R library that can be used to perform Bayesian areal wombling analyses of the sort described here, the code provided in Appendix B illustrates how to integrate several software packages to implement areal wombling models. Additional statistical challenges remain. The use of a CAR prior encourages local smoothing of dates of first infestation toward those of neighboring counties. Ideally, this accounts for uncertainty in detection, if, for example,

a single county reports a much later date of first infestation than its neighbors. Local smoothing can, however, have unanticipated effects. For example, a county that is colonized early but that is surrounded by counties with much later dates of colonization could have a modeled (smoothed) later date of first infestation. Although it is possible for the actual date of first infestation to be earlier than the reported date, it is unlikely that the actual date of first infestation would be later than the reported date (barring misidentification or data entry errors). Finally, the incorporation of spatially-correlated errors may alter estimates of fixed-effects coefficients in ways that are only beginning to be explored and which could lead to misinterpretation of residual-based wombling maps. Despite these issues, Bayesian areal wombling should be considered a complement to existing methods for ecological boundary analysis as one of the few techniques that can effectively utilize the coarse resolution datasets common in ecology and biogeography.

ACKNOWLEDGEMENTS -

We thank D. Wheeler and M. Levine for assistance with coding and for statistical advice. Charlie Burnham, Barbara Burns, Jason Denham, James Johnson, Tom McAvoy, Gary Miller, Dave Orwig, Bradley Regester, and Ann Steketee for access to HWA spread datasets. This research was funded by NSF DEB-0715504 to EP and by funding from UMCES to MCF.

LITERATURE CITED

Banerjee, S., B. P. Carlin, and A. E. Gelfand. 2004. Hierarchical Modeling and Analysis for Spatial Data. Chapman & Hall/CRC, Boca Raton, FL.

Brooks, S. P., and A. Gelman. 1998. General Methods for Monitoring Convergence of Iterative Simulations. Journal of Computational and Graphical Statistics 7:434-455.

296 Fagan, W. F., M. J. Fortin, and C. Soykan. 2003. Integrating edge detection and dynamic
297 modeling in quantitative analyses of ecological boundaries. BioScience 53:730-738.
298 Fitzpatrick, M. C., E. L. Preisser, A. M. Ellison, and J. S. Elkinton. 2009. Observer bias and the
299 detection of low-density populations. Ecological Applications 19:1673-1679.
300 Fortin, M. J., T. H. Keitt, B. A. Maurer, M. L. Taper, D. M. Kaufman, and T. M. Blackburn.
301 2005. Species' geographic ranges and distributional limits: pattern analysis and statistical
302 issues. Oikos 108:7-17.
303 Fortin, M., and M. R. T. Dale. 2005. Spatial Analysis. Cambridge University Press, Cambridge.
304 Gaston, KJ 2003. The structure and dynamics of geographic ranges. Oxford Unv. Press, Oxford.
305 Gavin, DG and FS Hu. 2006. Spatial variation of climatic and non-climatic controls on species
306 distribution: the range limit of *Tsuga heterophylla*. Journal of Biogeography 33:1384-1396.
307 Graham, C. H., S. Ferrier, F. Huettman, C. Moritz, and A. T. Peterson. 2004. New developments
308 in museum-based informatics and applications in biodiversity analysis. Trends in Ecology &
309 Evolution 19:497-503.
310 Holt, R. D., and T. H. Keitt. 2005. Species' borders: a unifying theme in ecology. Oikos 108:3-6.
311 Jacquez, G., M. Fortin, and P. Goovaerts. 2008. Preface to the special issue on spatial statistics
312 for boundary and patch analysis. Environmental and Ecological Statistics 15:365-367.
313 Jacquez, G. M., S. Maruca, and M. Fortin. 2000. From fields to objects: A review of geographic
314 boundary analysis. Journal of Geographical Systems 2:221-241.
315 Kowarik, I. 1995. Time lags in biological invasions with regard to the success and failure of
316 alien species. *in* P. Pysek, K. Prach, M. Rejmanek, and M. Wade, editors. Plant Invasions -
317 General Aspects and Special Problems. SPB Academic Publishing, Amsterdam.
318 Legendre, P. 1993. Spatial Autocorrelation - Trouble or New Paradigm. Ecology 74:1659-1673.

Lu, H., and B. P. Carlin. 2005. Bayesian Areal Wombling for Geographical Boundary Analysis. Geographical Analysis 37:265-285.

Ma, H., B. P. Carlin, and S. Banerjee. In Press. Hierarchical and Joint Site-Edge Methods for Medicare Hospice Service Region Boundary Analysis. Biometrics.

Ma, H., B. A. Virnig, and B. P. Carlin. 2006. Spatial methods in areal administrative data analysis. Italian Journal of Public Health 3:94-103.

Orwig, DA, D Foster, DL Mausel. 2002. Landscape patterns of hemlock decline in New England due to the introduced hemlock woolly adelgid. Journal of Biogeography 29:1475-1487.

Paradis, A., J. Elkinton, K. Hayhoe, and J. Buonaccorsi. 2008. Role of winter temperature and climate change on the survival and future range expansion of the hemlock woolly adelgid (*Adelges tsugae*) in eastern North America. Mitigation and Adaptation Strategies for Global Change 13:541-554.

Parmesan, C., S. Gaines, L. Gonzalez, D. M. Kaufman, J. Kingsolver, A. Townsend Peterson, and R. Sagarin. 2005. Empirical perspectives on species borders: from traditional biogeography to global change. Oikos 108:58-75.

R Development Core Team. 2009. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria.

Spiegelhalter, D., A. Thomas, N. Best, and D. Lunn. 2003. WinBUGS Users Manual, v 1.4.3.

Trotter, R. T., and K. S. Shields. 2009. Variation in Winter Survival of the Invasive Hemlock Woolly Adelgid (Hemiptera: Adelgidae) Across the Eastern United States. Environmental Entomology 38:577-587.

Wheeler, D., and L. Waller. 2008. Mountains, valleys, and rivers: The transmission of raccoon rabies over a heterogeneous landscape. Journal of Agricultural, Biological, and Environmental

Statistics 13:388-406.

Womble, W. H. 1951. Differential Systematics. Science 114:315-322.

FIGURE LEGENDS

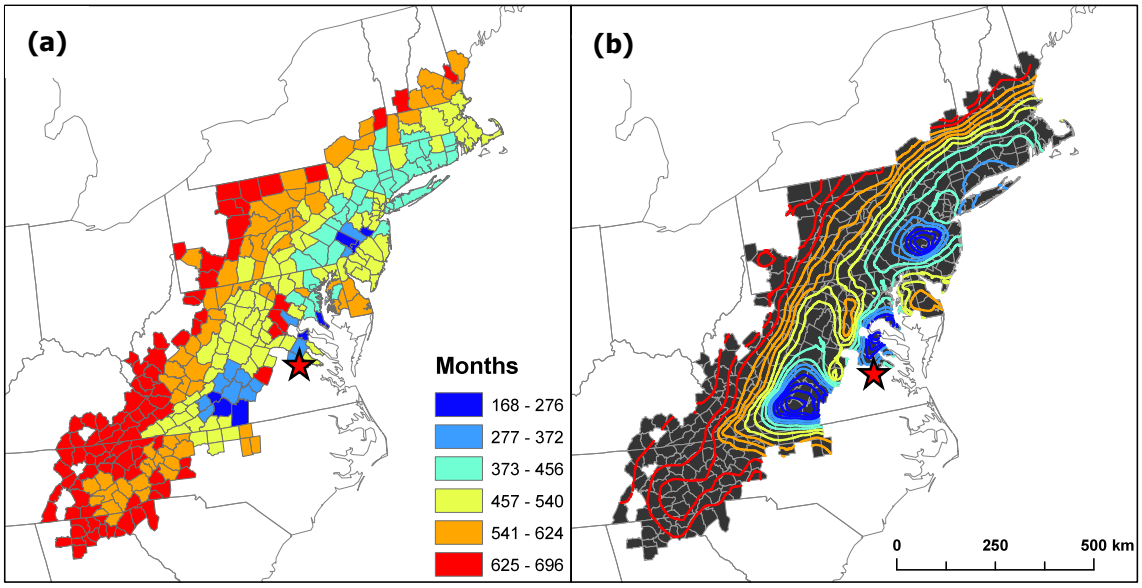
Figure 1. Observed pattern of spread of the hemlock woolly adelgid at (a) the county level and (b) smoothed using ordinary kriging of these dates. Colors represent the number of months elapsed since the first reported infestation in Richmond, VA (red star) in 1951 and the first reported infestation in each county.

Figure 2. Bayesian areal wombling on (a) simulated dates of first infestation; and (b) a single simulated covariate related to distance from Richmond, VA, with a cluster of outlier counties centered on York County, PA (red shading). Panels (c) and (d) show posterior probabilities for boundaries for the expected values μ and the spatial residuals ϕ respectively and a threshold of 60 months. Darker shades of red indicate high boundary probabilities.

Figure 3. Posterior probabilities for Bayesian areal wombling boundaries calculated using either (a) the expected values μ or (b) the spatial residuals ϕ and a threshold of 60 months. Darker shades of red indicate high boundary probabilities.

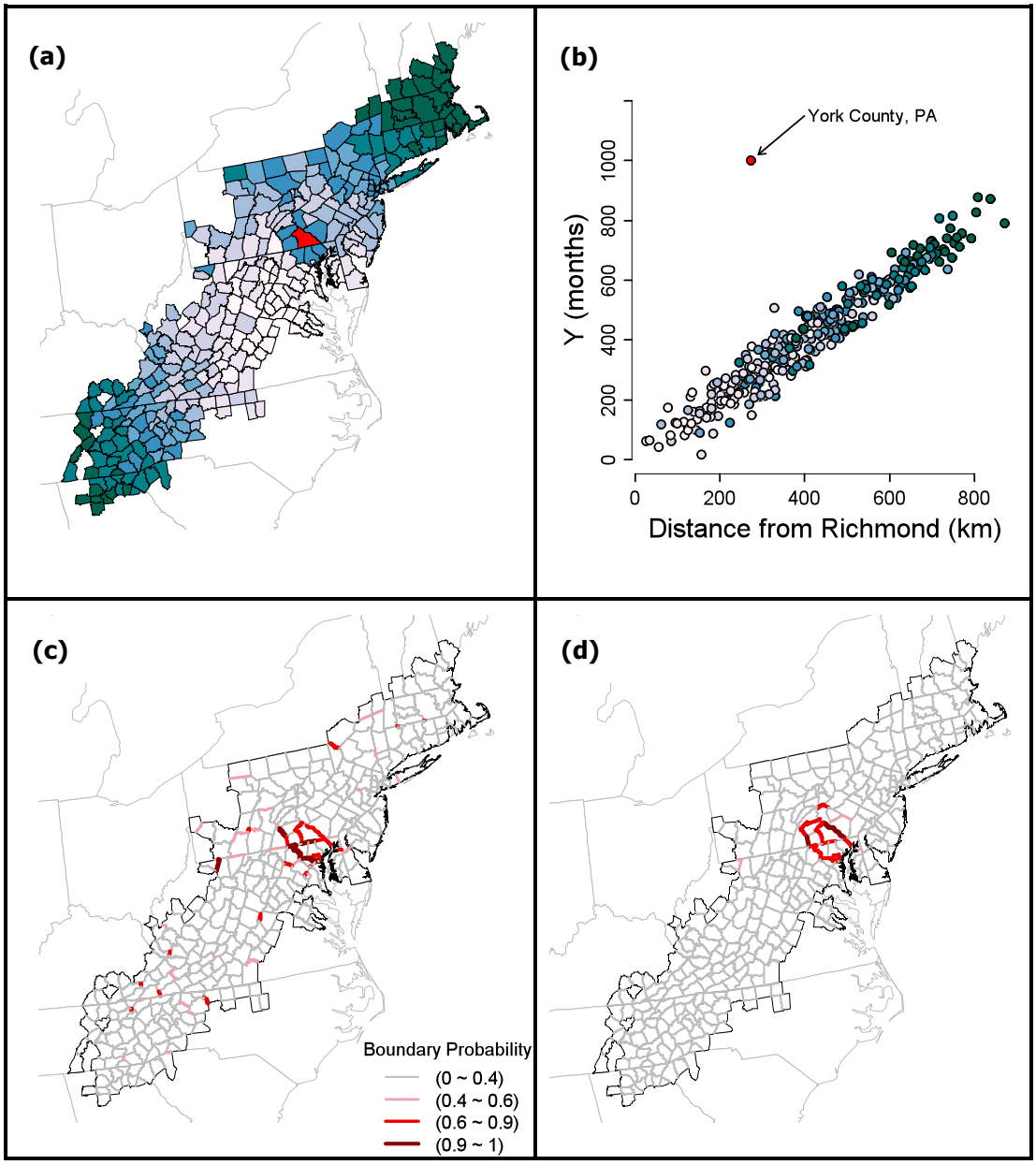
Figure 4. Posterior probabilities for Bayesian local edge wombling boundaries calculated using either (a) the expected values δ or (b) the spatial residuals ψ and a threshold of 36 months. Panels (c) and (d) show the same, but using a threshold of 60 months. Darker shades of red indicate high boundary probabilities.

365 Figure 1.



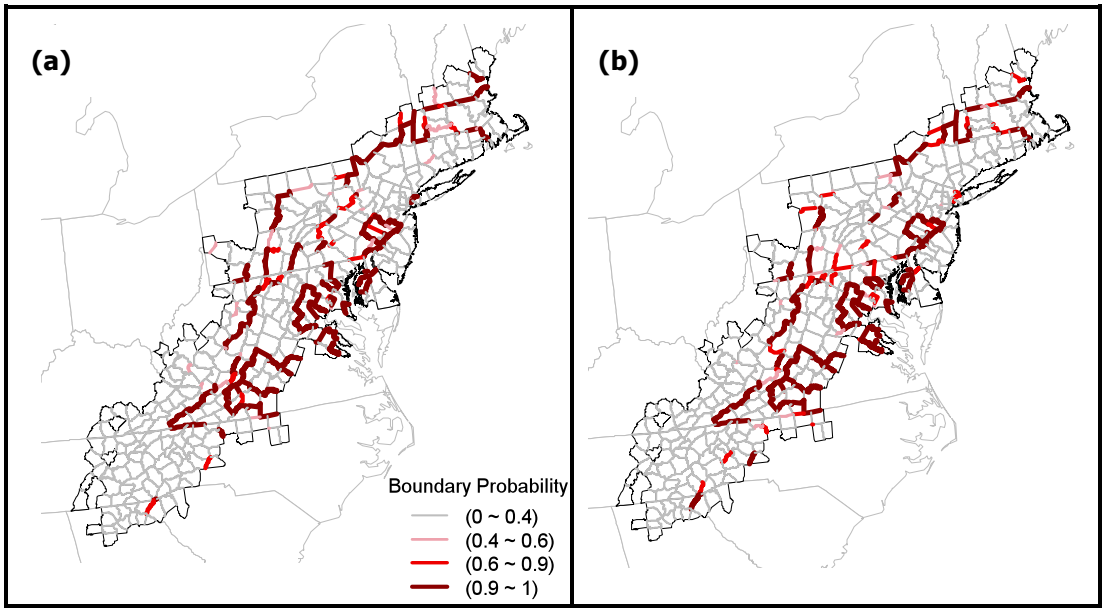
366

367 Figure 2.



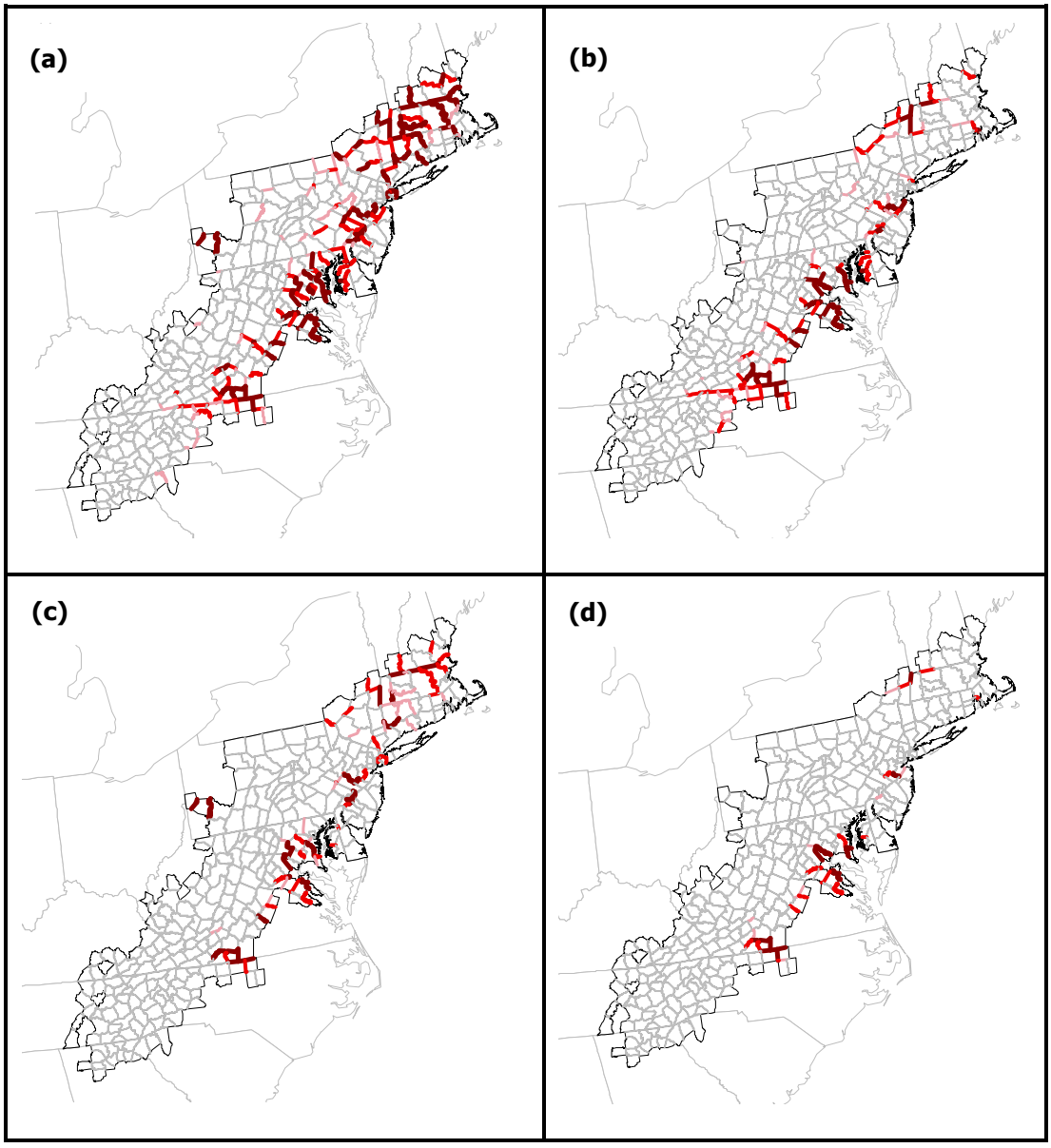
368

368 Figure 3.



369

369 Figure 4.



370

APPENDIX A – Description of datasets

County-level spread records - We derived the dynamics of HWA's spread for the years 1951 through 2009 using county-level records compiled by Forest Service, US Department of Agriculture, Forest Health Protection personnel (http://na.fs.fed.us/fhp/hwa/maps/distribution.shtm). We updated these county-level records with more localized records drawn from multiple sources, including: the National Entomological Collection at the Smithsonian Institute (G. Miller), the Pennsylvania General Hemlock Survey executed by the Pennsylvania Department of Conservation of Natural Resources (B. Regester), township-level records for Massachusetts (C. Burnham) and New York (J. Denham), surveys performed by the Georgia Forestry Commission (J. Johnson), stand-level surveys for southwestern Virginia (T. McAvoy), surveys in southern Vermont by the Vermont Department of Forests, Parks, & Recreation (B. Burns), and stand-level surveys in Connecticut and Massachusetts (D. Orwig). When these more local surveys indicated an earlier date of first infestation than the county-level records, we updated the county-level records as necessary. Finally, to simplify coding of the models, we removed 12 "island" counties, (i.e., counties with no infested neighbors possibly infested by long-distance jump dispersal). The final dataset comprised 322 counties with dates of first infestation ranging from 1951 to 2009.

Estimates of hemlock abundance - To produce a map of hemlock abundance we used the randomForests algorithm (Liaw and Wiener 2002) in R 2.9.1 (R Development Core Team 2009) to relate observed hemlock abundance (basal area, $m^2\ ha^{-1}$) from the USDA Forest Inventory and Analysis (FIA) database (comprised of 16,084 occurrences) to 26 environmental predictor

variables. Environment predictors included 23 bioclimatic variables describing minimum, maximum, and seasonality in temperature and precipitation and water balance (Hijmans et al. 2005, Svenning and Skov 2005), two topographic variables (slope and compound topography index) from the USGS HYDRO1k dataset (http://eros.usgs.gov/#/Find_Data/Products_and_Data_Available/gtopo30/hydro), and an index of net primary productivity (Zhao et al. 2005). All variables were manipulated in ArcGIS 9.3 such that they were spatially congruent, had a common resolution of 1 km, and were projected using and equidistance conic projection to preserve distance characteristics between locations.

We used the resulting model to predict hemlock abundance across eastern North America. Although Carolina hemlock (*Tsuga caroliniana*) is also susceptible to HWA, we did not model its distribution as it is relatively rare and narrowly distributed and its distribution falls entirely within the range of eastern hemlock. To account for the fact that most cells were not 100% forested, we multiplied the map of hemlock abundance by a corresponding remotely-sensed estimate of percent forest cover. The result was a map of hemlock abundance adjusted for forest cover that corresponds well with its known distribution and abundance.

Estimate of human population density & mean winter temperature – Estimates of human population density were derived from 2000 U.S. census data (http://www.census.gov/main/www/cen2000.html). Estimates of mean winter temperature (December, January, February, March) at 1km spatial resolution were downloaded from the Worldclim database (http://www.worldclim.org/, Hijmans et al. 2005). For all covariates, we used the Zonal Statistics tool in ArcGIS 9.3 to calculate summaries of covariates for each county.

47 LITERATURE CITED

48 Hijmans, R. J., S. E. Cameron, J. L. Parra, P. G. Jones, and A. Jarvis. 2005. Very high resolution
49 interpolated climate surfaces for global land areas. International Journal of Climatology
50 25:1965-1978. doi: 10.1002/joc.1276.

51 Liaw, A., M. Wiener 2002. Classification and regression by Random Forest. R News. 2: 18-22.

52 R Development Core Team. 2009. R: A language and environment for statistical computing. R
53 Foundation for Statistical Computing, Vienna, Austria. Retrieved from http://www.R-
54 project.org.

55 Svenning, J. C., and F. Skov. 2005. The relative roles of environment and history as controls of
56 tree species composition and richness in Europe. Journal of Biogeography 32:1019-1033.

57 Zhao, M., F. A. Heinsch, R. R. Nemani, and S. W. Running. (2005). Improvements of the
58 MODIS terrestrial gross and net primary production global data set. Remote Sensing of
59 Environment. 95: 164.176.

APPENDIX B – WinBUGS Code

```
# MODEL: three global covariates, spatial error

# WinBUGS model to perform Bayesian areal wombling (boundary detection) with
# global covariates

# Y is time to first infestation: the number of months elapsed, for each county
# i, since the first report of hemlock woolly adelgid in eastern North America
# in 1951 (e.g., if a county was found to be infested in 1981, Y = 360)

# code is called from R using R2WinBUGS

model{
  # Likelihood
  for (i in 1:n.areas){
    Y[i] ~ dnorm(mu[i], tau.err)

    mu[i] <- beta[1] + beta[2]*X1[i] + beta[3]*X2[i] + beta[4]*X3[i] + phi[i]

    # vector for plotting
    SLDRhat[i] <- mu[i] # SLDR, standardized late detection ratio,
                        # is legacy terminology from B. Carlin's code and
                        # has no meaning in this context
  }

  # CAR prior for the spatial random effects
  phi[1:n.areas] ~ car.normal(adj[], weights[], num[], tau.phi)   # CAR prior
  for (k in 1:sumNumNeigh){weights[k] <- 1}

  # Other priors
  beta[1] ~ dflat()
  beta[2] ~ dnorm(0, 0.000001)
  beta[3] ~ dnorm(0, 0.000001)
  beta[4] ~ dnorm(0, 0.000001)

  tau.phi <-1/pow(sdphi, 2)
  tau.err <- 1/pow(sdy, 2)
  sdphi ~ dunif(0,150)
  sdy ~ dunif(0,100)
}
```

```
# WinBUGS model to perform Bayesian local edge wombling (boundary detection)
# with three covariates & spatial error

# Y is the DIFFERENCE in time to first infestation

# Covariates are differences in values across edges

# Must have separate chunks of code for each edge without neighbors,
# 15 in this example

# code is called from R using R2WinBUGS

model{
  # Likelihood

  Y[1] ~ dnorm(mu[1], tau.err)
  mu[1] <- beta[1] + beta[2]*X1[1] + beta[3]*X2[1] + beta[4]*X3[1] + psi[1] + phi[1]
  #psi term is to account for island edges that have no neighbors

  # vector for plotting
  SLDRhat[1] <- mu[1]} # SLDR, standardized late detection ratio,
                       # is legacy terminology from B. Carlin code and has no
                       # meaning in this context

  for (i in 2:38){
    Y[i] ~ dnorm(mu[i], tau.err)
    mu[i] <- beta[1] + beta[2]*X1[i] + beta[3]*X2[i] + beta[4]*X3[i] + phi[i]
    SLDRhat[i] <- mu[i]}

  Y[39] ~ dnorm(mu[39], tau.err)
  mu[39] <- beta[1] + beta[2]*X1[39] + beta[3]*X2[39] + beta[4]*X3[39] + psi[2] + phi[39]
  SLDRhat[39] <- mu[39]

  for (i in 40:46){
    Y[i] ~ dnorm(mu[i], tau.err)
    mu[i] <- beta[1] + beta[2]*X1[i] + beta[3]*X2[i] + beta[4]*X3[i] + phi[i]
    SLDRhat[i] <- mu[i]}

  Y[47] ~ dnorm(mu[47], tau.err)
  mu[47] <- beta[1] + beta[2]*X1[47] + beta[3]*X2[47] + beta[4]*X3[47] + psi[3] + phi[47]
  SLDRhat[47] <- mu[47]

  for (i in 48:110){
    Y[i] ~ dnorm(mu[i], tau.err)
    mu[i] <- beta[1] + beta[2]*X1[i] + beta[3]*X2[i] + beta[4]*X3[i] + phi[i]
    SLDRhat[i] <- mu[i]}

  Y[111] ~ dnorm(mu[111], tau.err)
  mu[111] <- beta[1] + beta[2]*X1[111] + beta[3]*X2[111] + beta[4]*X3[111] + psi[4] + phi[111]
  SLDRhat[111] <- mu[111]

  for (i in 112:155){
    Y[i] ~ dnorm(mu[i], tau.err)
    mu[i] <- beta[1] + beta[2]*X1[i] + beta[3]*X2[i] + beta[4]*X3[i] + phi[i]
    SLDRhat[i] <- mu[i]}

  Y[156] ~ dnorm(mu[156], tau.err)
  mu[156] <- beta[1] + beta[2]*X1[156] + beta[3]*X2[156] + beta[4]*X3[156] + psi[5] + phi[156]
```

```
SLDRhat[156] <- mu[156]

for (i in 157:276){
  Y[i] ~ dnorm(mu[i], tau.err)
  mu[i] <- beta[1] + beta[2]*X1[i] + beta[3]*X2[i] + beta[4]*X3[i] + phi[i]
  SLDRhat[i] <- mu[i]}

Y[277] ~ dnorm(mu[277], tau.err)
mu[277] <- beta[1] + beta[2]*X1[277] + beta[3]*X2[277] + beta[4]*X3[277] + psi[6] + phi[277]
SLDRhat[277] <- mu[277]

for (i in 278:282){
  Y[i] ~ dnorm(mu[i], tau.err)
  mu[i] <- beta[1] + beta[2]*X1[i] + beta[3]*X2[i] + beta[4]*X3[i] + phi[i]
  SLDRhat[i] <- mu[i]}

Y[283] ~ dnorm(mu[283], tau.err)
mu[283] <- beta[1] + beta[2]*X1[283] + beta[3]*X2[283] + beta[4]*X3[283] + psi[7] + phi[283]
SLDRhat[283] <- mu[283]

for (i in 284:370){
  Y[i] ~ dnorm(mu[i], tau.err)
  mu[i] <- beta[1] + beta[2]*X1[i] + beta[3]*X2[i] + beta[4]*X3[i] + phi[i]
  SLDRhat[i] <- mu[i]}

Y[371] ~ dnorm(mu[371], tau.err)
mu[371] <- beta[1] + beta[2]*X1[371] + beta[3]*X2[371] + beta[4]*X3[371] + psi[8] + phi[371]
SLDRhat[371] <- mu[371]

for (i in 372:445){
  Y[i] ~ dnorm(mu[i], tau.err)
  mu[i] <- beta[1] + beta[2]*X1[i] + beta[3]*X2[i] + beta[4]*X3[i] + phi[i]
  SLDRhat[i] <- mu[i]}

Y[446] ~ dnorm(mu[446], tau.err)
mu[446] <- beta[1] + beta[2]*X1[446] + beta[3]*X2[446] + beta[4]*X3[446] + psi[9] + phi[446]
SLDRhat[446] <- mu[446]

for (i in 447:473){
  Y[i] ~ dnorm(mu[i], tau.err)
  mu[i] <- beta[1] + beta[2]*X1[i] + beta[3]*X2[i] + beta[4]*X3[i] + phi[i]
  SLDRhat[i] <- mu[i]}

Y[474] ~ dnorm(mu[474], tau.err)
mu[474] <- beta[1] + beta[2]*X1[474] + beta[3]*X2[474] + beta[4]*X3[474] + psi[10] + phi[474]
SLDRhat[474] <- mu[474]

for (i in 475:580){
  Y[i] ~ dnorm(mu[i], tau.err)
  mu[i] <- beta[1] + beta[2]*X1[i] + beta[3]*X2[i] + beta[4]*X3[i] + phi[i]
  SLDRhat[i] <- mu[i]}

Y[581] ~ dnorm(mu[581], tau.err)
mu[581] <- beta[1] + beta[2]*X1[581] + beta[3]*X2[581] + beta[4]*X3[581] + psi[11] + phi[581]
SLDRhat[581] <- mu[581]

for (i in 582:673){
  Y[i] ~ dnorm(mu[i], tau.err)
  mu[i] <- beta[1] + beta[2]*X1[i] + beta[3]*X2[i] + beta[4]*X3[i] + phi[i]
```

```
    SLDRhat[i] <- mu[i]}

  Y[674] ~ dnorm(mu[674], tau.err)
  mu[674] <- beta[1] + beta[2]*X1[674] + beta[3]*X2[674] + beta[4]*X3[674] + psi[12] + phi[674]
  SLDRhat[674] <- mu[674]

  for (i in 675:698){
    Y[i] ~ dnorm(mu[i], tau.err)
    mu[i] <- beta[1] + beta[2]*X1[i] + beta[3]*X2[i] + beta[4]*X3[i] + phi[i]
    SLDRhat[i] <- mu[i]
  }

  Y[699] ~ dnorm(mu[699], tau.err)
  mu[699] <- beta[1] + beta[2]*X1[699] + beta[3]*X2[699] + beta[4]*X3[699] + psi[13] + phi[699]
  SLDRhat[699] <- mu[699]

  for (i in 700:723){
    Y[i] ~ dnorm(mu[i], tau.err)
    mu[i] <- beta[1] + beta[2]*X1[i] + beta[3]*X2[i] + beta[4]*X3[i] + phi[i]
    SLDRhat[i] <- mu[i]}

  Y[724] ~ dnorm(mu[724], tau.err)
  mu[724] <- beta[1] + beta[2]*X1[724] + beta[3]*X2[724] + beta[4]*X3[724] + psi[14] + phi[724]
  SLDRhat[724] <- mu[724]

  for (i in 725:739){
    Y[i] ~ dnorm(mu[i], tau.err)
    mu[i] <- beta[1] + beta[2]*X1[i] + beta[3]*X2[i] + beta[4]*X3[i] + phi[i]
    SLDRhat[i] <- mu[i]}

  Y[740] ~ dnorm(mu[740], tau.err)
  mu[740] <- beta[1] + beta[2]*X1[740] + beta[3]*X2[740] + beta[4]*X3[740] + psi[15]+ phi[740]
  SLDRhat[740] <- mu[740]

  for (i in 741:793){
    Y[i] ~ dnorm(mu[i], tau.err)
    mu[i] <- beta[1] + beta[2]*X1[i] + beta[3]*X2[i] + beta[4]*X3[i] + phi[i]
    SLDRhat[i] <- mu[i]}

  # CAR prior for the spatial random effects
  phi[1:n.areas] ~ car.normal(adj[], weights[], num[], tau)   # CAR prior
  for (k in 1:sumNumNeigh){weights[k] <- 1}

  # prior for edges without neighbors
  for (j in 1:15) {psi[j] ~ dnorm(0, tau.psi)}

  # Other priors
  beta[1] ~ dflat()
  beta[2] ~ dnorm(0, 0.000001)
  beta[3] ~ dnorm(0, 0.000001)
  beta[4] ~ dnorm(0, 0.000001)

  tau <- 1/pow(sdphi,2) #per Andrew Lawson
  tau.err <- 1/pow(sdy,2)
  tau.psi <- 1/pow(sdpsi,2)
  sdphi ~ dunif(0, 100)
  sdy ~ dunif(0, 100)
  sdpsi ~ dunif(0, 100)
}
```

```r
##############################################################################
# R code to prepare data, call winBUGS code to perform Bayesian wombling,
# and plot output.
#
# 21 April 2010
# M. C. Fitzpatrick
# mfitzpatrick@umces.edu
#
# Most of this code is based on hard work by Brad Carlin & his students/post-
# docs. I have simply assembled many pieces into one place.
# In some instances, files available from:
# http://www.biostat.umn.edu/~brad
/software.html are needed. See comments below.
#
# The anlyses require a shapefile with a response of interest (in this case
# month of first infestation) and corresponding covariates
##############################################################################


###     chunk 1 - set wd and load libraries       ################################
setwd("...")
source("womblingFuncs.R")
library(R2WinBUGS)
library(maptools)
library(spdep)
library(coda)
library(RColorBrewer)
library(classInt)
library(sp)
##############################################################################


###     chunk 2 - build adjacency & edge info    ################################
#  Given a shapefile, this R code creates:
#    (1) an areal adjacency matrix using maptools,
#    (2) an edge adjacency matrix (indicating which edges touch each other)

# based on code provided from B. Carlin's website:
# http://www.biostat.umn.edu/~brad/software/getEdges_code.txt

# Also will need following two .exe files from B. Carlin's website
# (1) matchzip.exe
# (2) edgeneig.exe
# downloaded from: http://www.biostat.umn.edu/~brad/software/tutorial.zip

setwd(".../edgefolder")
# output files will be saved under current directory
# copy "matchzip.exe" and "edgeneig.exe" to directory "edgefolder"

```

```
map <- readShapeSpatial(".../hwa_wombling.shp")

# Use maptools to get polygon adjacency matrix
# two areas are neighbors if they share common edges with length > 0
nb.r <- poly2nb(map, queen=F)
mat <- nb2mat(nb.r, style="B") # mat is the 0/1 adjacency matrix
n.site <- dim(mat)[1]          # n.site: number of areas
n.edge <- sum(mat)/2           # n.edge: number of unique pairs


SEind1 <- SEind2 <- 0
matmy <- mat
for(i in 1:(n.site-1)){
    for(j in (i+1):n.site){
        if (mat[i,j]>0) {SEind1<-c(SEind1,i)
            SEind2<-c(SEind2,j)
            matmy[i,j]<-matmy[j,i]<-length(SEind1)-1
        }
    }
}

SEind1 <- SEind1[-1] # edges sorted by row of upper triangle of the adj. matrix
SEind2 <- SEind2[-1] # SEind1[k]=i and SEind2[k]=j => kth edge is edge ij

dput(SEind1,"SEind1.txt")
dput(SEind2,"SEind2.txt")
dput(mat, "W.txt")

# create adjacency information needed for WinBUGS
mkAdj <- function(W){
    n <- nrow(W)
    adj <- 0
    for(i in 1:n){
        for(j in 1:n){
            if(W[i,j]==1){adj<-append(adj,j)
            }
        }
    }
adj <- adj[-1]
return(adj)
}

dput(mkAdj(mat),"Sadj.txt")
dput(as.vector(rowSums(mat)),"Snum.txt")

# Create adj. matrix for the edges

# 1. prepare needed files and save them #
#    under the directory where you have #
#    matchzip.exe and edgeneig.exe      #

```

```
# Dump out the coordinates (by polygon) #
# Default order for polygons is by first column of map@data #

for(i in 1:n.site){
    write(t(map@polygons[[i]]@Polygons[[1]]@coords), paste(i,".txt",sep=""),
          ncolumns=2)
}

# Dump out SEind, the site-edge correspondence table #
write(rbind(SEind1, SEind2), paste("SEind.txt",sep=""), ncolumns=2 )

# 2. Double click "matchzip.exe". A dos window will pop up. Type in SEind.txt. #
#  This will produce many (n.edge) files at the current directory.              #
#  Then use the following code to prepare the edge plotting code.               #
#  "edgelines" should be dumped out and called in later when make edge plots.  #

edgelines <- vector(mode="list",length=n.edge)

for(i in 1:n.edge){
    edgelines[[i]] <- read.table(paste("output-",i,".txt",sep=""), header=F,
                                          na.strings="*")
}

edges <- edgelines
dput(edgelines,"edgelines.txt")

# 3. Double click "edgeneig.exe". A dos window will pop up. Type in SEind.txt  #
#  Two files will be produced (may take a while):                               #
#  file 1 is the upper triangular of the W matrix for the edges;                #
#  file 2 is the number of 1s for each row of the upper triangular matrix.     #

#  4. Produce the Wstar matrix which provides neighborhoods of the edges       #

tempn <- scan(paste("file2.txt",sep=""))
tempneig <- scan(paste("file1.txt",sep=""))

Wstar <- matrix(0, nrow=n.edge, ncol=n.edge)
start <- 1
end <- 0

for(i in 1:n.edge){
    if (tempn[i]>0){
        start <- end+1
        end <- start + tempn[i]-1
        neig <- tempneig[start:end]
        l <- tempn[i]
        for (k in 1:l){
            Wstar[i,neig[k]]<-Wstar[neig[k],i]<-1
        }
    }
}
```

```
151
152   edge.adj <- mkAdj(Wstar)
153   dput(edge.adj, "edge.adj.txt")
154   edgeSum <- as.vector(rowSums(Wstar))
155   dput(edgeSum, "edgeSum.txt")
156   dput(Wstar,"Wstar.txt")
157
158   # 5. Remove files not needed any more #
159   for(i in 1:n.edge){
160       unlink(paste(i,".txt",sep=""))
161       unlink(paste("output-",i,".txt",sep=""))
162   }
163
164   unlink(paste("file1.txt",sep=""))
165   unlink(paste("file2.txt",sep=""))
166   unlink(paste("SEind.txt",sep=""))
167   ##############################################################################
168
169
170   ###     chunk 3 - Areal wombling model      ###################################
171   # Three global covariates, spatial error
172   # mu = beta0 + beta1*X1 + beta2*X2 + beta3*X3 + phi
173
174   map <- readShapeSpatial(".../hwa_wombling.shp")
175
176   temp <- map@data$WINTERTEMP
177   pop <- log(map@data$POP2000)
178   hemlock <- log(map@data$HEMLOCK)
179
180   Y <- (map$YEARINFEST-1951)*12
181   X1 <- pop
182   X2 <- hemlock
183   X3 <- temp
184
185
186   n.areas = length(Y)
187   adj <- dget(".../edgefolder/Sadj.txt")
188   num = dget(".../edgefolder/Snum.txt")
189   sumNumNeigh = sum(num)
190
191   # indexes required for plotting
192   ind1 <- ind2 <- rep(0,length(num))
193   ind1[1] <- 1
194   for(i in 1:length(num)){j <- i+1; ind1[j] <- num[i] + ind1[i]}
195   ind1 <- ind1[1:length(num)]
196   for(i in 1:length(num)){j <- i-1; ind2[i] <- ind1[i] + num[i]-1}
197
198   params <- 5 # number of parameters in the model
199
200   # initial values
201   phi1 <- rep(-10,n.areas)
```

```
phi2 <- rep(0,n.areas)
phi3 <- rep(10,n.areas)

inits.mod5 <- list(list(phi=phi1, sdy=15, sdphi=5, beta=rep(-5, params)),
list(phi=phi2, sdy=25, sdphi=5, beta=rep(0, params)), list(phi=phi3, sdy=35,
sdphi=5, beta=rep(10, params)))

dat.in.mod5 <- list("Y", "n.areas", "num", "sumNumNeigh", "adj", "X1", "X2",
                    "X3", "X4")

# call to Winbugs
mod <- bugs(data=dat.in.mod5, inits.mod5,
            model.file=".../areal_wombling.bug",
            parameters.to.save=c("SLDRhat", "beta", "tau.err", "phi",
            "tau.phi"), n.chains = length(inits.mod5), n.iter=200000,
            n.burnin=100000, save.history=F, debug=TRUE,
            bugs.directory=".../WinBUGS14/", working.directory="...")

# read & summarize coda files
mod.coda <- read.coda.interactive()
# codaIndex.txt, coda1.txt, coda2.txt, coda3.txt
dimnames(mod.coda$coda1.txt)
samps <- mcmc.list(mcmc(mod.coda$coda1.txt[,c(323:328,651,652)]),
mcmc(mod.coda$coda2.txt[,c(323:328,651,652)]),
mcmc(mod.coda$coda3.txt[,c(323:328,651,652)]))
xyplot(samps)
gelman.plot(samps)
densityplot(samps)

merge.chains <- (mod.coda$coda1.txt + mod.coda$coda2.txt + mod.coda$coda3.txt)/3
SLDRhat <- merge.chains[,1:322]

rows <- nrow(SLDRhat)
cols <- ncol(SLDRhat)
phi <- merge.chains[,329:650]

# calculate posterior estimates of mu (sldrhat)
SLDRhat.samp <- matrix(0, ncol=cols, nrow=rows)
phi.samp <- matrix(0, ncol=cols, nrow=rows)
for(i in 1:n.areas){
    from<-(i-1)*rows+1
    to<-i*rows
    SLDRhat.samp[,i]<- SLDRhat[from:to]
    phi.samp[,i] <- phi[from:to]
}

#c alculate differences in spread dates across county edges
delta.sldr <- matrix(0, ncol=sum(num)/2,nrow=rows)
delta.phi <- matrix(0, ncol=sum(num)/2,nrow=rows)
k <- 0
for( i in 1:n.areas){
```

```
        for(j in ind1[i]:ind2[i]){
            if(adj[j]>i){
                k<-k+1
                delta.sldr[,k]<-abs(SLDRhat.samp[,i] - SLDRhat.samp[,adj[j]])
                delta.phi[,k]<-abs(phi.samp[,i] - phi.samp[,adj[j]])}}}

    # Boundary likelihood values
    p.sldr.5years <- apply(apply(delta.sldr,2,cut.func.5years)/rows,2,sum)
    p.phi.5years <- apply(apply(delta.phi,2,cut.func.5years)/rows,2,sum)

    # color palette for plotting boundaries
    n.col = 4
    col.br <- colorRampPalette(c("gray", "lightpink2", "red2", "red4"))
    col.pal <- col.br(n.col)

    # breaks for boundary groupings & legend text
    br <- c(0.0,0.4,0.6,0.9,1.0)
    leg.txt <- paste("(",br[n.col]," ~ ",br[n.col+1],")",sep="")
    for(i in (n.col-1):1){
        leg.txt <- append(leg.txt, paste("(", br[i], " ~ ", br[i+1], ")", sep=""),)
    }
    leg.txt <- rev(leg.txt)

    # Plot maps with boundary probabilities
    edgelines <- dget(".../edgefolder/edgelines.txt")

    # map of mu-based booundaries
    probPlot(map, edgelines, p.sldr.5years, n.col, add=F, col.pal=col.pal)
    legend(locator(), legend=leg.txt, col=col.pal, lty="solid", lwd=c(2,3,4,5),
           cex=1.8, ncol=1,
    bty="n", title="Boundary Probability")

    # map of phi-based boundaries
    probPlot(map, edgelines, p.phi.5years, n.col, add=F, col.pal=col.pal)
    legend(locator(), legend=leg.txt, col=col.pal, lty="solid", lwd=c(2,3,4,5),
           cex=1.8, ncol=1,
    bty="n", title="Boundary Probability")
    ##############################################################################


    ###     chunk 4 - Local edge wombling model      ################################
    # Three global covariates, spatial error
    # model will not run using R2WinBUGS for some reason
    # must copy and paste model, inits, data and run directly in winBUGS


    map <- readShapeSpatial(".../hwa_wombling.shp")
    Y <- (map$YEARINFEST-1951)*12
    mapDat <- map@data[,7:9]

    # prepare data for edge wombling
```

```
304
305   # edge deltas
306   SEind1 <- dget(".../edgefolder/SEind1.txt")
307   SEind2 <- dget(".../edgefolder/SEind2.txt")
308   edgelines <- dget(".../edgefolder/edgelines.txt")
309   n.edge <- length(SEind1)
310
311   deltaY <- weight.calculate(Y, SEind1, SEind2, n.edge)
312
313   deltaCov <- matrix(NA, n.edge, ncol(mapDat))
314   for(i in 1:ncol(mapDat)){
315       covar <- mapDat[,i]
316       dx <- delta.calculate(covar, SEind1, SEind2, n.edge)
317       deltaCov[,i] <- dx
318   }
319
320   colnames(deltaCov) <- names(mapDat)
321
322   Y <- deltaY
323
324   adj = dget("...edgefolder/edge.adj.txt")
325   num = dget("...edgefolder/edgeSum.txt")
326   n.areas = length(Y)
327   sumNumNeigh = sum(num)
328
329   X1 <- deltaCov[,"POP2000"]
330   X2 <- deltaCov[,"HEMLOCK"]
331   X3 <- deltaCov[,"WINTERTEMP"]
332
333   params <- 4 # number of parameters in the model
334
335   # initial values
336   phi1 <- rep(10, n.areas)
337   phi2 <- rep(0, n.areas)
338   phi3 <- rep(-1, n.areas)
339
340   phi1[which(num==0)] <- NA # for edges with no neighbors
341   phi2[which(num==0)] <- NA
342   phi3[which(num==0)] <- NA
343
344   psi1 <- rep(10, length(which(num==0))) # for edges with no neighbors
345   psi2 <- rep(1, length(which(num==0)))
346   psi3 <- rep(0, length(which(num==0)))
347
348   inits.mod10 <- list(list(psi=psi1, phi=phi1, sdy=15, sdphi=5, sdpsi=1,
349                            beta=rep(5, params)), list(psi=psi2, phi=phi2, sdy=25,
350                            sdphi=5, sdpsi=5, beta=rep(-2, params)), list(psi=psi3,
351                            phi=phi3, sdy=35, sdphi=5, sdpsi=5, beta=rep(-50,
352                            params)))
353   edit(inits.mod10)
354
```

```
dat.in.mod10 <- list(sumNumNeigh=sumNumNeigh, n.areas=n.areas, Y=Y,  num=num,
                     adj=adj, X1=X1, X2=X2, X3=X3)
edit(dat.in.mod10)


#call to Winbugs
mod10 <- bugs(data=dat.in.mod10, inits.mod10, model.file="...edge_womble.bug",
parameters.to.save=c("SLDRhat", "beta", "tau.err", "phi", "psi"),
n.chains = length(inits.mod10), n.iter=20000, n.burnin=10000,
save.history=F, debug=TRUE, bugs.directory="../WinBUGS14/", codaPkg=T,
working.directory="...")

# read & summarize coda files
mod.coda <- read.coda.interactive()
# codaIndex.txt, coda1.txt, coda2.txt, coda3.txt
dimnames(mod.coda$coda1.txt)
samps <- mcmc.list(mcmc(mod.coda$coda1.txt[,c(794:798, 1592:1594)]),
mcmc(mod.coda$coda2.txt[,c(794:798, 1592:1594)]),
mcmc(mod.coda$coda3.txt[,c(794:798, 1592:1594)]))
xyplot(samps)
gelman.plot(samps)
densityplot(samps)

merge.chains <- (mod.coda$coda1.txt + mod.coda$coda2.txt + mod.coda$coda3.txt)/3
SLDRhat <- merge.chains[,1:793]
phi <- merge.chains[,799:1576]
psi <- merge.chains[,1577:1591]

rows <- nrow(SLDRhat)

# Boundary likelihood values at 5 years
p.sldr.5years <- apply(apply(abs(SLDRhat),2,cut.func.5years)/rows,2,sum)
p.phi.5years <- apply(apply(abs(phi),2,cut.func.5years)/rows,2,sum)
p.psi.5years <- apply(apply(abs(psi),2,cut.func.5years)/rows,2,sum)

num1 <- ifelse(num==0,0,1)

#combine island vector with index and sort
indx <- seq(1:length(Y))
srt <- as.data.frame(cbind(num1, indx))
srt <- srt[order(srt$num1, srt$indx),]

# bind phi and psi and then to srt df
phiX <- c(p.psi.5years, p.phi.5years)
srt <- cbind(srt,phiX)

# sort to original order and extract new psi vector
phi.df <- srt[order(srt$indx),]
p.phi.5years <- phi.df$phiX

# color palette for plotting boundaries
```

```
n.col = 4
col.br <- colorRampPalette(c("gray", "lightpink2", "red2", "red4"))
col.pal <- col.br(n.col)

# breaks for boundary groupings & legend text
br <- c(0.0,0.4,0.6,0.9,1.0)
leg.txt <- paste("(",br[n.col]," ~ ",br[n.col+1],")",sep="")
for(i in (n.col-1):1){
    leg.txt <- append(leg.txt, paste("(", br[i], " ~ ", br[i+1], ")", sep=""),)
}
leg.txt <- rev(leg.txt)

# Plot maps with boundary probabilities
edgelines <- dget(".../edgefolder/edgelines.txt")

# map of mu-based booundaries
probPlot(map, edgelines, p.sldr.5years, n.col, add=F, col.pal=col.pal)
legend(locator(), legend=leg.txt, col=col.pal, lty="solid", lwd=c(2,3,4,5),
       cex=1.8, ncol=1, bty="n", title="Boundary Probability")

# map of phi-based boundaries
probPlot(map, edgelines, p.phi.5years, n.col, add=F, col.pal=col.pal)
legend(locator(), legend=leg.txt, col=col.pal, lty="solid", lwd=c(2,3,4,5),
       cex=1.8, ncol=1, bty="n", title="Boundary Probability")

# Boundary likelihood values at 3 years
p.sldr.3years <- apply(apply(abs(SLDRhat),2,cut.func.3years)/rows,2,sum)
p.phi.3years <- apply(apply(abs(phi),2,cut.func.3years)/rows,2,sum)
p.psi.3years <- apply(apply(abs(psi),2,cut.func.3years)/rows,2,sum)

num1 <- ifelse(num==0,0,1)

# combine island vector with index and sort
indx <- seq(1:length(Y))
srt <- as.data.frame(cbind(num1, indx))
srt <- srt[order(srt$num1, srt$indx),]

# bind phi and psi and then to srt df
phiX <- c(p.psi.3years, p.phi.3years)
srt <- cbind(srt,phiX)

# sort to original order and extract new psi vector
phi.df <- srt[order(srt$indx),]
p.phi.3years <- phi.df$phiX

# map of mu-based booundaries
probPlot(map, edgelines, p.sldr.3years, n.col, add=F, col.pal=col.pal)
legend(locator(), legend=leg.txt, col=col.pal, lty="solid", lwd=c(2,3,4,5),
       cex=1.8, ncol=1, bty="n", title="Boundary Probability")

# map of phi-based boundaries
```

```
probPlot(map, edgelines, p.phi.3years, n.col, add=F, col.pal=col.pal)
legend(locator(), legend=leg.txt, col=col.pal, lty="solid", lwd=c(2,3,4,5),
       cex=1.8, ncol=1, bty="n", title="Boundary Probability")
##############################################################################


# functions needed to format data & results and make plots

probPlot <- function(map, edgelines, y, n.col, add, col.pal){
  require(classInt)
  polylist <- map@polygons
  br <- c(0, 0.4, 0.6, 0.9, 1)
  y.grp <- findInterval(y, vec=br, rightmost.closed = TRUE, all.inside = TRUE)
  y.shad <- col.pal[y.grp]
  linewd <- y.grp + 1

  plot(map, axes=F, auxvar=Y, add=add)

  for (i in 1:length(edgelines)){
    lines(as.matrix(edgelines[[i]]),col=y.shad[i],lwd=linewd[i])
  }
}


#functions to calculate boundary probs at different thresholds
cut.func.1years <- function(x){
    c.ind<-as.numeric(x>12)
    return(c.ind)}

cut.func.2years <- function(x){
    c.ind<-as.numeric(x>24)
    return(c.ind)}

cut.func.3years <- function(x){
    c.ind<-as.numeric(x>36)
    return(c.ind)}

cut.func.4years <- function(x){
    c.ind<-as.numeric(x>48)
    return(c.ind)}

cut.func.5years <- function(x){
    c.ind<-as.numeric(x>60)
    return(c.ind)}

cut.func.6years <- function(x){
    c.ind<-as.numeric(x>72)
    return(c.ind)}

cut.func.8years <- function(x){
    c.ind<-as.numeric(x>96)
```

```r
    return(c.ind)}

cut.func.10years <- function(x){
    c.ind<-as.numeric(x>120)
    return(c.ind)}

# functions need to calculate deltas across boundries
delta.calculate <- function(x, ind1, ind2, n){
  delta<-rep(0,n)
  for (k in 1:n){
    i <- ind1[k]
    j <- ind2[k]
    delta[k] <- x[j] - x[i]
  }
  return(delta)
}

weight.calculate <- function(x, ind1, ind2, n){
  delta<-rep(0,n)
  for (k in 1:n){
    i <- ind1[k]
    j <- ind2[k]
    delta[k] <- x[j] - x[i]
  }
  return(delta)
}
```