Chapter 14
Components of Spatial Information Management in Wildlife Ecology: Software for Statistical and Modeling Analysis

Hawthorne L. Beyer, Jeff Jenness, and Samuel A. Cushman

14.1 Introduction

Spatial information systems (SIS) is a term that describes a wide diversity of concepts, techniques, and technologies related to the capture, management, display and analysis of spatial information. It encompasses technologies such as geographic information systems (GIS), global positioning systems (GPS), remote sensing, and relational database management systems (RDBMS), which were once relatively independent but increasingly have been merging into cohesive, integrated systems.

Technological developments are facilitating the collection of unprecedented volumes of spatially referenced ecological data. The historical trend towards improved accuracy, resolution, precision and frequency of spatial data is likely to continue in the future. However, what have been lacking are the advances in statistical methods and analytical software that are needed to facilitate our use of this information to answer ecological questions in a rigorous manner. In Chapter 1 we presented a conceptual model for the interdependence between data, method and ecological theory. The collection of spatial data has exceeded the pace by which statistical methods and analytical software has advanced.

The task of collecting and using spatial data presents us with three general problems: how do we manage this data, how do we visualize, inspect, and process this data

S.A. Cushman (✉)
US Forest Service, Rocky Mountain Research Station, 800 E Beckwith, Missoula, MT 59801, USA
e-mail: scushman@fs.fed.us

H.L. Beyer
Faculty of Biomedical & Life Sciences, Ecology & Evolutionary Biology, Graham Kerr Building, University of Glasgow, Glasgow G12 8QQ, Scotland, UK
e-mail: hawthorne@spatialecology.com

J. Jenness
Jenness Enterprises, 3020 N. Schevene Blvd. Flagstaff, Arizona, 86004, USA
e-mail: jeffj@jennessent.com

S.A. Cushman and F. Huettmann (eds.), *Spatial Complexity, Informatics, and Wildlife Conservation*
DOI 10.1007/978-4-431-87771-4_14, © Springer 2010

efficiently, and how do we use this data to inform the decision making process and improve our understanding of ecological systems? We have three goals for this chapter. First, we will describe a range of existing kinds of software for spatial ecological analysis, and review their functionality. Second, we will discuss a number of important issues facing researchers who wish to develop their own custom software tools, and provide some general suggestions to assist their efforts. Third, we look forward to discuss the future of analytical computing, and identify key technological developments that are driving current and future progress in spatial analytical software development.

14.2 Data Storage and Management

There are several important motivations for adopting formal data management strategies. Given the increasing rate of collection of spatially referenced data it becomes imperative to automate the storage and organization of that data, enforce quality control and consistency standards, and provide a method of disseminating data to stakeholders. Also, the large volumes of data typical of spatial datasets require systems that can efficiently query and process the data, ideally by multiple concurrent users. Relational database management systems (RDMBS) are key data administration devices as they are designed to handle very large amounts of data (terabytes), efficiently process queries on huge databases, facilitate the sharing and distribution of data over networks, provide user access control and other security measures, permit simultaneous querying and editing of data by multiple users (including the ability to resolve editing conflicts), and drive network based data dissemination services (web portals and data servers). Most statistical and spatial analysis software is able to access data directly from a RDBMS.

The expense of setting up and maintaining a RDBMS is offset by the long term indirect cost savings of having a centralized, consistent, updated, permanent data warehouse that requires relatively minimal human management but that can drive a wide range of services and uses. Database management systems also help to ensure that data is appropriately archived so that it can be accessed in the future (possibly by the broader scientific community). Permanent archiving of data is something that is often neglected in the field of ecology and wildlife management. In Chapter 6 we presented a conceptual model for flexible, multi-variate, multi-scale, spatial databases to serve as the foundation for ecological monitoring and spatial and temporal analysis of ecological pattern-process relationships across scale. The construction of these databases will require the use of RDMBS within an explicitly spatial template.

Examples of RDBMS software include PostGRESQL, SQL Server, and Oracle (see Table 14.1 for details).

Table 14.1 Examples of spatial information system, statistical, modeling and spatial analysis software listed alphabetically within each section (this list is not comprehensive)

General purpose statistical and modeling software	
R[a]	http://www.r-project.org
Mathmatica	http://www.wolfram.com
MATLab	http://www.mathworks.com
WinBUGS[a]	http://www.mrc-bsu.cam.ac.uk/bugs
Software specializing in quantitative spatial analysis	
CrimeStat[a]	http://www.icpsr.umich.edu/CRIMESTAT
FRAGSTATS[a]	http://www.umass.edu/landeco/research/fragstats/fragstats.html
GUESS[a]	http://graphexploration.cond.org
PASSAGE[a]	http://www.passagesoftware.net
General purpose Geographic Information System (GIS) software	
ArcGIS	http://www.esri.com
ArcReader[a]	http://www.esri.com
GeoDa	https://www.geoda.uiuc.edu
GeoMedia	http://www.intergraph.com
Google Earth[a]	http://earth.google.com
GRASS[a]	http://grass.itc.it
IDRISI	http://www.clarklabs.org
Landserf[a]	http://www.landserf.org
Manifold	http://www.manifold.net
MapInfo	http://www.mapinfo.com
Quantum GIS[a]	http://www.qgis.org
RAMAS	http://www.ramas.com
SAGA[a]	http://www.saga-gis.org
TNTMips	http://www.microimages.com
uDIG[a]	http://udig.refractions.net
Spatial analysis software specifically designed for wildlife ecology	
ade4[a], adehabitat[a]	http://pbil.univ-lyon1.fr/ADE-4/home.php?lang=eng
CDPOP	http://LandguthResearch.dbs.umt.edu
Corridor Design[a]	http://www.corridordesign.org
Geospatial Modelling Environment[a]	http://www.spatialecology.com
HRT: Home Range Tools For ArcGIS[a]	http://blue.lakeheadu.ca/hre
Marine Geospatial Ecology Tools[a]	http://code.env.duke.edu/projects/mget
Tools for Graphics and Shapes[a]	http://www.jennessent.com/
Relational Database Management System (RDBMS) software	
Oracle	http://www.oracle.com
PostGRESQL[a]	http://www.postgresql.org
SQL Server	http://www.microsoft.com
Remote sensing software	
eCognition	http://www.definiens.com
ENVI	http://www.ittvis.com/envi/index.asp
ER Mapper	http://www.ermapper.com

(continued)

Table 14.1 (continued)

ERDAS Imagine	http://www.erdas.com
SPRING[a]	http://www.dpi.inpe.br/spring/english
PCI	http://www.pci.on.ca
Simulation software	
AnyLogic	http://www.xjtek.com/
CelLab[a]	http://www.fourmilab.ch/cellab
MASON	http://cs.gmu.edu/~eclab/projects/mason
NetLogo[a]	http://ccl.northwestern.edu/netlogo
SWARM[a]	http://www.swarm.org

[a]Open source or free software

14.3 Mapping and Visualization

A wide range of commercial and free software solutions exist for displaying and exploring spatial data. The principal features this software usually offers include: (1) on-the-fly reprojection of spatial data so that datasets with different projections are appropriately aligned without having to manually reproject the dataset, (2) the ability to access datasets stored on remote spatial data servers, (3) the ability to perform tabular and spatial queries on the data (e.g. select all features with a specified attribute value, or select all features from one dataset that overlap the features from another dataset), (4) the ability to directly read or import a wide variety of data formats, and (5) options for graphical display and map making. Although we do not offer a comparative review of mapping and visualization software here (because we focus on quantitative spatial analysis), many examples are provided in Table 14.1 (see the General Purpose GIS Software section).

14.4 Quantitative Spatial Analysis

Spatial analysis is a nebulous term that covers a wide range of statistical and numerical techniques. It ranges from incorporating spatial attributes (e.g. coordinates or distance measures) into standard statistical and modeling analyses, which handle the spatial data in a spatially implicit manner, to using techniques that have been specifically designed for application to spatial data, such as geostatistics (Cressie 1993; Diggle and Ribeiro 2007), pattern analysis, graph theory and network analysis, spatial randomization tests, and spatial stochastic simulation (Dale 1999; Fortin and Dale 2005). A wide diversity of software has been developed to facilitate these types of analysis. We briefly describe each of these categories of analysis and provide examples of software that can be used to implement them.

14.4.1 Geostatistics

Geostatistics involves modeling continuous spatial phenomena based on a set of spatially referenced samples. There are a variety of geostatistical analytical approaches; however, they all share a basic focus on predicting a quantitative response variable across space based on the pattern and autocorrelation of that variable among spatially referenced sample locations. Typically the sample data are point samples arranged in space using a regular, random, or stratified random design (e.g. point locations at which elevation is measured) and the goal of the modeling process is often spatial prediction. *Model-based geostatistics* (Diggle and Ribeiro 2007) is the term that describes the general, formal statistical framework of which commonly used techniques like simple kriging are one example.

Geostatistics often involves simple data structures: coordinates describing the spatial location of point samples, and sometimes also a boundary (polygon) that describes the area of interest. Predictions resulting from geostatistical models are often represented as raster data structures.

Examples of software that implement geostatistics include the geoR and geoRglm packages in R, GSTAT, and Geostatistical Analyst (ESRI ArcGIS) (see Table 14.1 for details).

14.4.2 Pattern Analysis

Spatial pattern analysis is a method of quantitatively describing heterogeneously distributed (patchy) phenomena. In studying pattern, we hope to learn something about the processes that have generated these patterns, i.e. to help us to develop a mechanistic understanding of ecological processes. Pattern analysis covers a broad range of techniques (see Dale 1999; Fortin and Dale 2005) and employs a variety of data structures, for instance: points (nearest neighbour analysis, point pattern analysis), quadrats and transects (quadrat variance analysis), rasters (spectral analysis and wavelets, boundary detection, landscape metrics), and polygons (polygon change analysis).

Examples of relevant software include several R packages (e.g. spatial, spatstat, splancs, waveslim, wavelets), PASSaGE, and FRAGSTATS (see Table 14.1 for details).

14.4.3 Randomization Tests

Statistical significance is quantified using a theoretically justifiable reference distribution. In the case of spatial patterns and processes we often have no preconceived notion about what a suitable reference distribution would be. Randomization techniques (bootstrapping, Monte Carlo simulation) allow us to generate appropriate reference distributions that can be used to quantify significance, and can be applied to many different sorts of problems. This approach forms the basis of techniques like Mantel's test.

Examples of relevant software include PASSAGE, R, and the Geospatial Modelling Environment (see Table 14.1 for details).

14.4.4 Stochastic Simulation

Like randomization tests, stochastic simulation helps us to quantify how likely our observed data is given a proposed process model. They can also be used to make predictions about the state of a system at a given time based on the process model, which captures the spatial and temporal dynamics of a system and range from mechanistic models to phenomenological models.

Process models tend to be specific to a particular system/question (e.g. animal movement models, landscape fire models) and it is therefore difficult to write software that can accommodate all possible model forms and variations. Most stochastic simulation involves writing simulation programs, either in a 'high-order' statistical language like R, or any of the lower level programming languages (C, C++, Java,.NET, Fortran, Pascal, etc.). However, for certain types of simulations (like cellular automata and individual based models) there is generic software available that may be useful in implementing a process model.

Examples of relevant software include R, MATLab, Mathmatica, SWARM, and CelLab (see Table 14.1 for details).

14.4.5 Graph Theory and Network Analysis

A graph is a collection of nodes connected by edges to form a network. Although graphs do not necessarily represent spatial systems (e.g. the nodes may represent producers and consumers at different trophic levels in an ecosystem and the edges represent the trophic relationships between them), here we focus on the spatial applications (cities connect by a network of roads, or foraging patches connected by a network of trails). One of the premises of network analysis is that it allows us to identify the emergent properties of complex systems that might not otherwise be apparent by direct observation. Furthermore, because many ecological processes are well represented by networks (e.g. disease transmission) they provide a powerful and often more realistic/mechanistic method of modeling the process.

Examples of relevant software include R (igraph package), GUESS, and Network Analyst (ESRI ArcGIS) (see Table 14.1 for details).

14.5 Developing Custom Geospatial Analysis Tools

Implementing a novel analysis often involves software customization. Many software providers provide a framework for adding custom functionality to their software. In the case of open source software everyone has access to the source code so we have complete freedom to customize the application (subject to open source license agreements). But many commercial products also provide powerful interfaces for customization.

Customization not only allows us to add new functionality, it allows us to improve the efficiency of existing functionality. This can be important in the context of stochastic simulations and Monte Carlo applications where a procedure must be run a large number of times, potentially requiring long processing times (weeks or months). It is not uncommon to reduce processing times by an order of magnitude or more following code optimization for a particular analysis, making it more feasible to use.

There are many different development environments that can be used to develop customized geospatial analyses and there are several issues that should be considered when selecting one. One very important consideration is to what extent you can re-use existing code. The ability to call functionality in existing tools (so that you do not have to program them yourself) is the factor that is most likely to reduce development time for an application. Understanding what functionality you require, and where it is available in existing software is a necessary first step.

Another consideration is what level of control and access you have to this existing code. With open source software the potential for code re-use is very high. With commercial environments, you are limited to the objects and methods that the company chooses to expose to you, and are somewhat at that company's mercy if they choose to change that code.

In the case of quantitative spatial analysis one must also decide whether to approach customization from the geospatial or the statistical side of the problem. Many tools exist for manipulating and exporting geospatial data, so implanting customization using statistical and modeling software (e.g. R) may be the most efficient approach.

Familiarity with a programming language is the other factor that most affects development time. If you are new to programming it is not advisable to begin with a 'low-level' programming language (e.g. C, C++, Java, .NET). Geospatial applications tend to be quite complex, involving a wide array of programming techniques and technologies. The higher order scripting languages will are easier to learn and be productive with (e.g. R, Python) because they do not require the same level of technical programming knowledge.

Examples of customized geospatial analysis tools pertaining to wildlife ecology include "CDPOP" (Landguth and Cushman in press), "Corridor Design" (Majka et al. 2007), "HRT: Home Range Extension for ArcGIS" (Rodgers et al. 2007), and the "Geospatial Modelling Environment" (see Table 14.1 for details).

14.5.1 Software Transparency, Documentation and Certification

Reporting a methodology in enough detail that it could be independently repeated is an important aspect of the scientific method. For quantitative analyses it is important to understand exactly what software is doing so that we are able to document our methodology and understand what the implications of a software program or algorithm might be on the results we obtain. However, there is little consensus on what an appropriate level of documentation should be for software. Commercial software, for which the source code is not available and algorithms are often considered proprietary and are therefore sparsely described or not described at all, is particularly problematic; we input data and get new data as output but we have little or no definitive information about how the algorithm works, hence the term "black box" software. In such cases many authors will report the name and version number of the software used, which in some sense would allow a method to be independently repeated if other people had access to the same software. However, we argue that black box software does little to improve our understanding of the systems we study and should be avoided where well documented alternatives exist.

A simple example illustrates this problem. Often an analytical technique can be implemented in a number of different ways that can result in qualitatively different solutions. For instance, bivariate kernel density estimation (KDE) is a commonly used technique in ecology. There are a number of kernels that can be used (Gaussian, quartic, etc.) and in the case of a 2D problem the bandwidth parameter is a 2×2 covariance matrix (Silverman 1986). Some commonly used GIS software packages make simplifying assumptions in the implementation of KDE by using a particular kernel and assuming an equal bandwidth for both dimensions with no covariance, but none of these details are documented. This black box approach to KDE can be misleading as these simplifying assumptions can have a profound influence on the results obtained.

In a discipline in which peer review plays such an important role in advancing understanding it is perhaps surprising that no formal peer review process has been adopted for software. In the case of "open source" software the source code is freely available so can be reviewed for logical consistency. It is therefore completely transparent, well documented (because there is no more precise documentation than the code itself) and open to peer review. In such cases a somewhat informal peer review process often takes place whereby a number of developers will review and improve an algorithm (the statistical software R is a good example of this).

However, it would be impractical to suggest that we should review source code or documentation every time we wish to use software. It would be more efficient to identify cross-disciplinary standards for scientific software and establish a system of peer review. Software certification would help to ensure that software was bug free, that appropriately detailed documentation existed, and that adequate testing had been performed. These standards would apply equally well to open source and commercial software. Although we are moving towards such a system, in the mean time we encourage users to favour software that is well documented and tested.

14.5.2 The Future of Analytical Computing

In the past spatial software was designed for large, independent workstations and used proprietary data formats that prohibited convenient software interoperability. Computing technology is now moving towards virtualization (operating systems and programs are not bound to a particular instance of computer hardware), distributed and cloud computing (data processing using remote systems, software distributed via the network or internet), and widely adopted standards that ensure software interoperability. The spatial data, processing and exchange standards being defined by the Open Geospatial Consortium (OGC; http://www.opengeospatial.org), which has broad support across many industries, are particularly important for ensuring interoperability in the future.

The ramifications of seamless software interoperability on spatial analysis are likely to be that we use a wider array of software, developed by a broad community of developers, and often accessed via the internet. Web Processing Services (WPS) are a particularly promising approach to making analyses available via the internet. WPS is a set of standards that facilitates access to an analysis algorithm or model that uses spatial data. It defines the required input data, the output data that is generated, runs the process and returns the results. The provider of the WPS service is able to maintain and improve the service independently of any software that might be installed on the client machine. The client can run the WPS process using data that is stored locally or on a network data server, and processing is performed locally.

The importance of providing transparent, peer-reviewed and high quality quantitative analysis solutions is likely to be greatly facilitated by R (R Development Core Team 2008), which has had a profound impact on the field of statistical computing. Several GIS applications already use R as the statistical and modeling engine that drives spatial analyses.

References

Cressie NAC (1993) Statistics for spatial data. Wiley, New York

Dale MRT (1999) Spatial pattern analysis in plant ecology. Cambridge University Press, Cambridge

Diggle PJ, Ribeiro PJ (2007) Model-based geostatistics. Springer, New York

Fortin M, Dale MRT (2005) Spatial analysis: a guide for ecologists. Cambridge University Press, Cambridge

Landguth EL, Cushman SA (in press) CDPOP: a spatially explicity, individual-based, population genetics model. Mol Ecol Resour. doi:10.1111/j.1755-0998.2009.02719.x

Majka D, Jenness J, Beier P (2007) CorridorDesigner: ArcGIS tools for designing and evaluating corridors. http://corridordesign.org

R Development Core Team (2008) R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3–900051–07–0, http://www.R-project.org

Rodgers AR, Carr AP, Beyer HL, Smith L, Kie JG (2007) HRT: home range tools for ArcGIS. Version 1.1. Ontario Ministry of Natural Resources, Centre for Northern Forest Ecosystem Research, Thunder Bay, Ontario, Canada

Silverman BW (1986) Density estimation for statistics and data analysis. Chapman & Hall/CRC, London, New York