Computing environments for spatial data analysis

Luc Anselin

Regional Economics Applications Laboratory (REAL) and Department of Agricultural and Consumer Economics, University of Illinois, Urbana-Champaign, 326 Mumford Hall, 1301 W Gregory Drive, Urbana IL, 61801-3608, USA (e-mail: anselin@uiuc.edu)

Abstract: This paper describes the functionality and architecture of SpaceStat, the SpaceStat Extension for ArcView and the DynESDA Extension for ArcView. It compares the features of these packages to five other software implementations for spatial data analysis. Some ideas are formulated on generic requirements and future directions pertaining to computing environments for spatial data analysis.

Key words: geocomputation, spatial econometrics, spatial statistics, SpaceStat

1 Introduction

In the recent past, there has been a burst of activity focused on providing operational spatial data analytical functionality within a GIS environment. Early conceptual frameworks and basic requirements for such an integration were outlined in a by now familiar series of papers in the late 1980s and early 1990s, such as Goodchild (1987), Goodchild et al. (1992) and Anselin and Getis (1992).

These early papers focused not so much on actual implementations, but on the overall conceptual framework for integrating spatial analysis with GIS. For example, in Goodchild et al. (1992) the integration between GIS and statistical analysis was classified as either loose or close coupling, based on the way in which data and commands were passed between the GIS and the statistical software. Typically, the linkage pertained to a commercial GIS and a full-featured statistical or econometric software package, and with a few rare exceptions early implementations involved applying standard *non-spatial* statistical functions to GIS data. In these linkages, loose coupling was the rule, and consisted of establishing a type of pseudo-communication between two software packages by writing intermediate results and/or commands into text files. These text files subsequently served as a standard input file and no additional linkage mechanism was necessary beyond the usual manipulation functions (reading input data, joining and sorting tabular data, etc.). In contrast, close coupling consists of an inter-process communication between two software packages, in which commands for one can be called from the user interface of the other by means of remote procedure calls or dynamic data exchange. This is increasingly feasible in current networked computing environments based on client/server principles. In addition, with the explosion of the internet, the communication between systems can be extended to computers and data sets at different physical locations, where typically the data storage and the analysis are not carried out on the same computer.

Arguably, in more recent implementations of the link between statistical analysis and GIS, the distinction between loose and close coupling has become less relevant. Most operational approaches end up relying on some combination of these designs. Also, with the increased availability of disk caching (which speeds up loose coupling by avoiding actual disk access) as well as the explosion in the use of distributed computing across the

internet (which slows down close coupling due to network overhead), performance comparisons between the two forms of coupling are no longer straightforward.

An alternative taxonomy was offered in Anselin and Getis (1992) where a distinction is made between encompassing and modular frameworks. The first is simply an extension of the functionality of a GIS with that of a statistical package or vice versa. This is beginning to be reflected in a growing number of commercial products, for example, by the inclusion of mapping and some geostatistical functionality in statistical software such as SAS and Systat, and the extension of the Arcinfo GIS with the forthcoming geostatistical analyst. Paralleling these commercial efforts, customization by academic researchers has involved incorporating a wide range of specialized spatial statistical analyses or other forms of computational modeling that are typically not part of commercial software packages. Usually, this is carried out by relying on built-in scripts or macro commands. There are by now quite a few of such applications, making possible the calculation of global and local spatial autocorrelation indices, the estimation of spatial regressions and fitting of geostatistical models.[1] While these extensions maintain the familiar look-and-feel of the GIS or the statistical software, a drawback of the encompassing approach is that peculiarities of the scripting languages (such as Avenue for ArcView and MapBasic for MapInfo) sometimes preclude the use of the most efficient algorithms or data structures for the statistical computations. Consequently, performance is affected and few implementations can tackle realistic data sets or deliver results sufficiently fast for real-time interactive use. An alternative to the encompassing design is a modular approach, in which a framework of linked systems is constructed, each optimized for a specific functionality, such as statistical analysis, mapping, or user interaction. An growing number of such applications have come to exist as well. An important aspect of a modular design is the handling of communication between the systems, typically following a client-server paradigm, which can be readily extended to a distributed computing environment. Increasingly, such interaction can be encapsulated into software components that are based on object-oriented software design techniques. In principle, such componentization provides the potential for the development of a collection or suite of reusable spatial data analytical software pieces that can be mixed and matched by a researcher to tackle specific problems.

This special issue of the *Journal of Geographical Systems* reports on some recent developments in the efforts to extend the spatial analytical capability of GIS with sophisticated statistical and econometric functionality. The five papers included in the issue approach this question from a different perspective, which illustrates the richness of research and diversity of computing solutions that are being developed. The software solutions range from freestanding programs such as GEM in the paper by Geoffrey Jacquez, Susan Maruca and Marie-Jose Fortin, focused on a specific methodological issue (boundary analysis), to interfaces between analytical modules and various commercial GIS, such as Mapinfo, in the paper by Patrick Wall and Owen Devine, Arc/Info, in the paper by Robert Haining, Stephen Wise and Jingsheng Ma, ArcView and Grassland, in the paper by Shuming Bao, Luc Anselin, Doug Martin and Diana Stralberg,

[1] An extensive review of specific implementations is beyond the scope of the current paper. A recent review is given in Anselin (1998). See also Anselin and Bao (1997) for an earlier review of specific implementations.

as well as libraries of routines built with open source statistical toolboxes, such as R, in the paper by Roger Bivand and Albrecht Gebhardt.

Rather than simply summarizing these approaches, the purpose of this paper is to provide some perspective and outline some generic issues encountered in developing computational solutions to spatial statistical and econometric problems in current software environments. The various approaches discussed in the papers that follow are compared against this general background. Also, the comparison is extended by including a brief discussion of SpaceStat (Anselin 1992), arguably the oldest freestanding comprehensive software package for the statistical analysis of lattice (areal) data. In addition to a review of the core SpaceStat functionality and design, some discussion is provided of the recently developed DynESDA extension for exploratory spatial data analysis with the ArcView GIS (Anselin and Smirnov 1999a, b).

In the remainder of the paper, an overview is first presented of the functionality of SpaceStat and the DynESDA extension for ArcView. This is followed by a comparison and review of the various specific software solutions treated in the other papers in this issue. This comparison is carried out in the context of a discussion of a number of generic software development issues that are encountered in the implementation and dissemination of spatial data analysis functionality, with particular emphasis on the analysis of areal data. The paper concludes with some speculations on future directions.

2 SpaceStat and the SpaceStat Extension for ArcView

The SpaceStat software package for spatial data analysis was first released by the National Center for Geographic Information and Analysis (NCGIA) in 1992 as part of a general initiative to promote the dissemination of spatial analysis techniques to teaching and research (Anselin 1992). The current version (1.90) contains the functionality to manipulate spatial data and spatial weights, carry out descriptive and exploratory spatial statistics and implement spatial regression analysis.[2] The software is compiled in the Gauss toolbox of Aptech Inc. and runs as a DOS window under either Windows 95/98 or Windows NT. The Gauss version used in the construction of SpaceStat is fully 32-bit, such that, apart from the user interface (which is not Windows-based), the code is completely compatible with the latest Windows versions and can be executed in a multitasking environment. The software is freestanding, in the sense that it does not require Gauss itself, nor a particular GIS.

The design philosophy underlying SpaceStat is to avoid duplication and to specialize in functionality that is not readily available in commercial statistical and econometric software. As a result, SpaceStat purposely does not include techniques to handle point patterns or geostatistical analysis, both of which were contained in a number of commercial and academic software available in the early 1990s. It also has no mapping or graphic capability and all visualization is carried out through an extension developed for the ArcView desktop GIS. The focus is on the implementation of advanced spatial statistical and econometric methods using the most efficient algorithms.

The initial motivation for SpaceStat was to provide software tools to carry out all

[2] A full description as well as demonstrations can be found at http://www.spacestat.com.

the spatial econometric estimation methods and specification tests outlined in Anselin (1988), which consisted mostly of maximum likelihood based methods. In recent versions, this has been augmented considerably by the addition of general method of moments and instrumental variables estimators, which also broaden the application domain beyond the classical regression model to models with endogenous variables (two stage least squares estimation).

A second extension from the original focus is the inclusion of recently developed techniques for exploratory spatial data analysis (ESDA), specifically the Moran scatterplot (Anselin 1996) and local indicators of spatial association (LISA) statistics (Anselin 1995). These methods are particularly well suited for visualization in a map, which motivated the development of the SpaceStat Extension for ArcView. The latter implements a two-way interaction between the statistical functionality of SpaceStat and the visualization and data manipulation of the GIS.

Next, the core statistical and econometric functionality of SpaceStat is outlined first, followed by a description of the design and implementation of the SpaceStat Extension.

2.1 SpaceStat Core Functionality

The functionality of SpaceStat is organized into four modules, labeled Data, Tools, Explore and Regress.[3] The first two cover over 100 functions to deal with data input, manipulation and transformation, including a range of specialized operations on spatial weights files, as well as spatial smoothing and spatial filtering. Data used in SpaceStat functions consists of the combination of a data set and one or more matching spatial weights files that contain the spatial arrangement of the observations. These files are linked by means of a common key or label variable, which matches each element in the spatial weights file one-to-one to an element in the data set. The data set itself is stored as a flat file and conforms to the binary Gauss data format. Current preferred practice is to load SpaceStat data sets directly from the SpaceStat Extension for ArcView, which allows access to data in various binary formats (such as data base files). However, there is also a legacy data input function which converts ascii text files to the Gauss binary data format.

SpaceStat allows three different formats to incorporate the spatial arrangement of the data in the form of spatial weights. One is based on the binary Gauss matrix format (of dimension N by N, where N is the size of the data set), and is required for maximum likelihood estimation. The other two implement a sparse format (only non-zero elements are included in the file), either for binary contiguity (so-called GAL format) or for general spatial weights (so-called GWT format). The sparse format is the preferred approach since it avoids problems with limited workspace memory. The weights manipulation functions in SpaceStat are the most comprehensive available to date and include implementations of several methodological and algorithmic innovations (e.g., Anselin and Smirnov 1996). An overview of the Data and Tools commands is given in Tables 1 and 2, which follow the overall menu structure of the SpaceStat user interface.

---

[3] An elaborate discussion of the functionality of SpaceStat is given in the original tutorial, Anselin (1992), and subsequent updates at http://www.spacestat.com/manuals.htm.

Table 1 also lists the Report Files that are created when the ArcView link is active. In addition to the regular content of any SpaceStat data set, these files provide a means to add spatially transformed, smoothed or filtered variables to an ArcView shapefile.

The core of the statistical functionality of SpaceStat is contained in the Explore and Regress modules. Explore deals with descriptive and exploratory spatial statistics, such as outlier detection and global and local measures of spatial autocorrelation, including join counts, Moran's I, Geary's c, Moran scatterplot, local Moran, G_i and G_i^* statistics and QAP. An overview of this functionality is presented in Table 3. As in Table 1, the third column lists the Report files that are generated by the different functions. These Report files contain observation-specific results, such as LISA statistics, and provide a straightforward coupling mechanism with ArcView.

The Regress module covers the specification, estimation and diagnostic testing of spatial regression models.[4] The functionality is organized along three dimensions. First is the type of regression specification. SpaceStat recognizes a generic regression, as well as four specialized models: trend surface, spatial expansion, spatial regimes and spatial analysis of variance (dummy variable regression). These four specification allow for different forms of spatial heterogeneity to be expressed in the model, either based on discrete subsets of the data (so-called spatial regimes) or in the form of a type of hierarchical regression using polynomials in the x-y coordinates of the spatial observations. For each of the spatial heterogeneity specifications the actual model is constructed within SpaceStat without the need for the user to explicitly specify all the variables. Also, the selection of a specialized model yields a series of tests on spatial heterogeneity specific to each model (e.g., a spatial Chow test on the stability of regression coefficients across spatial regimes).

The second dimension along which the Regress module is organized pertains to the type of spatial model that is required. Five generic specifications are currently supported: classic linear regression model, spatial autoregressive error model, heteroskedastic error model, spatial lag model, and systems model. For each type of model, a number of specialized estimation methods are implemented. This constitutes the third dimension in the Regress module. Coefficient estimates and standard inference (asymptotic t-tests) are computed, as well as a range of diagnostics for spatial autocorrelation and spatial heterogeneity (such as Lagrange Multiplier tests for spatial lag and spatial error autocorrelation, Likelihood Ratio tests for spatial autocorrelation and spatial heterogeneity). Several models also incorporate a combination of spatial dependence and spatial heterogeneity in the form of groupwise heteroskedasticity, which is particularly useful for the study of spatial regimes. An overview of the spatial econometrics models and methods in SpaceStat is given in Table 4. In each instance, the Report file used to establish the link with ArcView contains observed value, predicted value and residual for each data point.

Even though its design (original under DOS) and architecture are beginning to show their age, in terms of functionality SpaceStat is still the most comprehensive package for the spatial econometric analysis of areal data, in that it includes both maximum likelihood and modern method of moments techniques and a full range of

[4] For a recent overview of the methodological issues, see Anselin and Bera (1998).

specification diagnostics.[5] The sparse weights formats allow the analysis of data sets with up to tens of thousands of observations (the actual limit depends on computer hardware, specifically, on the amount of RAM available). In the current version, maximum likelihood estimation should be limited to models with less than 1,000 observations, due to the use of eigenvalue routines in the implementation of the nonlinear optimization. In the near future, this will be complemented with sparse routines in order to remove this constraint (Smirnov and Anselin, 2000). In addition, the version of SpaceStat that is currently under development will include methods for the analysis of panel data and techniques for specification testing and estimation in spatial probit models.

2.2 The SpaceStat Extension

The SpaceStat Extension for the ArcView desktop GIS (Anselin and Smirnov 1999a) was developed to facilitate data exchange and visualization of location-specific results between SpaceStat and ArcView, with primary emphasis on techniques for exploratory spatial data analysis (ESDA). It follows the general design originally outlined in Anselin et al. (1993) for a link between SpaceStat and ArcInfo, which was earlier ported to ArcView as described in Anselin and Bao (1997). The extension is implemented by means of a combination of Avenue scripts and specialized functions written in C and organized in a dynamic link library. The latter was added to address performance issues encountered in the original all-Avenue version. The collection of scripts is installed as an ArcView "extension" which essentially adds the custom scripts to the standard ArcView installation in a transparent way.

To the user, the extension appears in the form of two additional menu items in the ArcView View interface, labeled Data and SpaceStat, as illustrated in Fig. 1 and Fig. 2. The primary role of the Data Menu is data exchange from ArcView to SpaceStat, yielding data sets and weights files in the format required by SpaceStat in a one-step operation. In addition, a small number of functions are included to facilitate the computation of "spatial" variables. Specifically, the first two menu items, identical to the earlier version in Anselin and Bao (1997, p. 43), provide the computation of respectively a selected features indicator variable and polygon centroid coordinates. The indicator variable takes on a value of one for all areal units that are selected in a View (by means of a select tool in ArcView) and forms the basis for a spatial analysis of variance in SpaceStat.[6] Polygon centroid coordinates on the other hand are necessary for the construction of distance-based spatial weights in SpaceStat (such as spatial weights based on distance bands or on k-nearest neighbors), as shown by the items in Table 2.[7]

---

[5] Other packages that are currently available either do not cover the range of spatial specifications, or do not include both method of moments techniques and maximum likelihood, or have limited (or no) diagnostic tests.

[6] See also Anselin et al. (1993) for a detailed description of the use of selected areal units in spatial analysis of variance.

[7] A third function is included to fix shapefiles that contain multiple polygons with different identifiers for the same areal unit. This function facilitates interoperability with export files generated by ArcInfo, but is seldom necessary.

The most useful features of the Data Menu are contained in the functions that build SpaceStat data sets from the data base files that correspond to ArcView Tables and that construct spatial weights from the ArcView shape files. These functions do not use Avenue, but instead call routines from a dynamic link library written in C that access the binary files directly. The ArcView user interface is employed to select variables for export, which are then passed as a list to the appropriate C routine in the dll. This routine extracts the relevant data from the dBase file corresponding to the active theme in the current View and subsequently creates a file in the binary format used by SpaceStat. The construction of spatial weights is slightly more complex, but based on a similar principle. It exploits the published binary format of shape files (ESRI 1995) and builds a topology for the areal units in a View by means of a bounding box algorithm, followed by a series of search and sort operations. This procedure has superior performance characteristics relative to an implementation that uses the built-in Avenue functions, due to the inefficiency of loop structures in Avenue. The information on the spatial arrangement of the areal units is written to an ascii file in the GAL format that SpaceStat reads. The first time the weights are loaded into SpaceStat, the values for the selected key (the value for a variable that uniquely identifies each areal unit) are transformed to the sequential values (actually row numbers in a matrix that holds all the data) that are used in the internal operations in SpaceStat. This process is transparent to the user and is accomplished in two mouse clicks (one to select the menu item and one to select the indicator variable). Weights can be computed for a rook criterion (common vertices only) as well as for a queen criterion (both vertices and corners in common; in other words, the moment two polygons have a single point in common, they are considered to be neighbors).

The remainder of the functionality in the Data Menu consists of export functions to convert ArcView data sets and boundary files to ascii format, and a generic link function that allows any SpaceStat Report file that contains a proper key variable to be joined to a Table in ArcView.

The SpaceStat Menu contains the functionality to visualize location-specific results from SpaceStat. This is a one-way transfer of information from SpaceStat to ArcView. It is implemented as a series of Avenue scripts that are invoked from the SpaceStat Menu. The scripts look for specific Report files in the current directory (generated by SpaceStat and conforming to a specific format), identify the key variable to join the information in the file to the current Table in ArcView, and draw a map with a customized legend. This is a straightforward application of Avenue programming that reduces repetitive tasks to one or two mouse clicks per application. Examples of the types of maps that can be created in this fashion are outlier and percentile maps, spatial lag bar and pie charts, maps with various spatial smoothers, Moran scatterplot maps, LISA Local Moran Map and Moran significance maps (highlighting locations with a significant Local Moran statistic as well as the type of spatial autocorrelation in that location) and maps with predicted values and residuals from spatial regression analyses. Such maps can be readily converted by ArcView to formats amenable for inclusion in other graphics packages or desktop publishing software.

3 The DynESDA Extension for ArcView

The DynESDA Extension for ArcView (Anselin and Smirnov 1999b) was inspired by the use of dynamic graphics in exploratory data analysis (e.g., Cleveland and McGill 1988; Buja et al. 1996) and its particular extension to the exploration of spatial data pioneered in the Spider and Regard packages of Haslett, Unwin and associates (Haslett et al. 1990, 1991; Unwin 1996). In this context, a map (i.e., a View in ArcView terminology) becomes one of many linked views on the data, such as a table, histogram, box plot and other statistical graphics. The views are linked in the sense that any observation highlighted in one of the views by means of a pointing device (e.g., clicking with a mouse when he cursor is on a point in a scatterplot) is also simultaneously highlighted in all the other views. Specifically, in DynESDA, linking sets up an interaction between the active theme of a View in the ArcView workspace and a series of statistical graphs constructed by the user for the data contained in the associated theme "Table". In addition to standard statistical graphs, DynESDA also implements a spatial association visualizer (Anselin 1998), which allows for the interactive recalculation of spatial association statistics for subsets of the data, as well as diagnostics for leverage and outliers.[8]

The DynESDA extension is similar in its emphasis on linking and brushing to the link between ArcView and the XploRe and XGobi packages for data exploration implemented in Cook et al. (1997) and Symanzik et al. (1997), among others. However, important differences are the focus on lattice or areal data rather than points (used as the basis for point pattern analysis or geostatistical modeling) and the multiway rather than pairwise linkages between graphs.[9] DynESDA is implemented as a collection of customized routines with limited functionality, rather than as a general purpose data exploration engine.

3.1 Functionality

A user starts the extension by means of a button that is added to the ArcView interface. Clicking on the button creates a floating toolbar (Fig. 3), from which the different statistical graphs are invoked. Currently, four graphs are fully supported and a number of others are under development. The statistical graphs include the customary histogram, boxplot and scatterplot, as well as the Moran scatterplot. By clicking on the corresponding icon on the floating toolbar (or selecting the matching menu item), a new statistical graph is created. The necessary input in terms of variable selection is carried out using the standard ArcView user interface tools and the communication between ArcView and the dynamic link library is transparent to the user.

Dynamic linking of graphs is implemented on the View side by means of the

---

[8] See also Anselin (1994, 1999a) for more extensive details on the specific ESDA tools.

[9] The nature of the inter-process communication between ArcView and respectively XGobi and XploRe limits the linkage to those graphs between which communication has been established. In Symanzik et al. (1997), this was restricted to pairwise links. In contrast, in the DynESDA Extension, the active theme in the current View in ArcView is linked with as many statistical graphs as the user opens.

standard ArcView select tools. This allows for the selection of single areal units, as well as a contiguous set or subregion. On the statistical graphics side, selection is implemented by means of a standard cursor and click and drag mouse operations. In addition to selecting individual items in the statistical graph (e.g., points in a scatterplot or a given bin for a histogram), users can also construct a box around a number of points or around an observation range in a box plot by means of standard mouse drag operations. Selecting observations in this manner results in the matching areal units being highlighted in the View. Dynamic brushing is implemented by activating a rectangle around a set of points (or an interval in a box plot) and moving it across the graph. As the box moves in one graph, all matching observations in the other graphs are highlighted. For different scatterplots or a scatterplot matrix, this is straightforward, but for the histogram and box plot, the matching observations correspond respectively to subsets of histogram bins and value ranges. In the map, the selected observations are highlighted as well as the brush moves across the graph.

The linking and brushing functionality allows for a wide range of interactive analyses, not only univariate, but multivariate as well. From a spatial analysis perspective, there are a number of particularly useful applications that would otherwise be difficult to carry out. One is the description of the distribution of a variable for a spatial subset of the observations, as selected in the map. For example, this allows the assessment of the degree to which the resulting histogram of the subset matches that for the whole set, for multiple variables simultaneously, providing important insight into the possible delineation of spatial regimes. Another important application is the visualization of spatial autocorrelation in the form of the slope of the linear smoother in the Moran scatterplot (Anselin 1996). Moran scatterplots can be constructed side by side for multiple variables or for the same variable at different points in time, allowing for the visualization of a form of space-time correlation. Each Moran scatterplot also includes a randomization routine to assess the significance of the Moran's I statistic in the form of an empirical distribution histogram for the randomly permuted data. Finally, the dynamic brushing functionality allows for the recomputation of the linear smoother in a scatterplot for a subset of the data that does not include the selected observations. As the brush moves over the scatterplot, both the original slope and the recomputed slope are shown and the smoother is redrawn as the brush moves. This works in the same fashion for standard scatterplots as well as for the Moran scatterplot, allowing for extensive sensitivity analysis (for example, to border effects) of the indication of spatial autocorrelation.

The combination of the different statistical graphs, the visualization of spatial autocorrelation and the map provides a powerful tool for exploratory data analysis. It has been applied in a number of different interdisciplinary research context, where besides introducing a methodological innovation it has also yielded important new substantive insights (e.g., Messner et al. 1999).

3.2 Architecture

The DynESDA extension is implemented as a single Avenue script that interacts with a dynamic link library of routines written in C++. The routines in the library handle all aspects related to the construction, linking and manipulation of the statistical graphics. The Avenue script initializes the routines and handles the user interaction.

After initialization, all the functions contained in the dll become available to the Avenue script. The initialization also creates the floating toolbar, which is the interface to the main driver in the dll routines. The essence of the Avenue script is a large loop structure that listens for messages from the dll and passes information back to the dll. The loop continues until a message is received that ends the operation. This message is received after the toolbar is explicitly removed (closed) and all graphics windows are closed.

There are two types of information that are passed by the dll to the Avenue script. First is an identifier for the particular statistical graph that was selected by the user from the floating toolbar (histogram, box plot, scatterplot or Moran scatterplot). For each type of statistical graph, the Avenue script generates the proper user interface to obtain the name of the variable (or variables) for which the graph should be constructed. This is implemented in the standard ArcView user interface. The second type of information pertains to any changes in the selected observations (for example, any observations that were highlighted or enclosed in a brushing box). This is implemented by means of the "Bitmap" object in Avenue, which is an efficient way to keep track of changes in the selection for a given theme. Any change in the selection that was initiated in the statistical graphics part yields an updated Bitmap on the ArcView side and results in the redrawing of the display. This is implemented very efficiently in ArcView, so that even when brushing scatterplots the user has the impression that the updating of selections is instantaneous.

The flow of information from the script back to the dll also consists of two types. First, there is a variable name (or names) that was collected from the user interface and is one of the numeric variables contained in the active theme. Only the variable name is passed to the dll since the respective functions access the data themselves directly in the binary dBase file that corresponds to the active theme. The script calls the proper function for the selected statistical graph. The second type of information pertains to any updates in the Bitmap that result from a select operation in the View. The updated Bitmap will result in all highlighted items on the statistical graphs to be updated, again simulating true dynamic linking. All functionality required for the implementation of the spatial statistics is internal to the dll and does not require further back and forth with the Avenue script. For example, the permutation routine used to assess significance of Moran's I and the construction of spatial weights are both carried out as functions in the dll. This achieves superior speed relative to using the built-in Avenue functions. Those routines tend to be geared to interactive use, but are less efficient when required in a loop structure (e.g., to compute the neighbors for all areal units in the View).

4 A Comparison of Computing Environments

The five papers included in this issue each offer a different perspective on the computational implementation of spatial data analysis. The first, by Geoffrey Jacquez, Susan Maruca and Marie-Jose Fortin focuses on methodological issues encountered in the analysis of boundaries that are conceptualized as the definition of objects on spatial fields. Apart from a review of definitional issues and specific delineation techniques, they describe the functionality of a software package, GEM, currently under development. GEM is intended to be freestanding and has its own visualization procedures, but it

contains routines to import a range of common GIS data formats. Its functionality is unique and geared to rather specialized boundary detection applications in the fields such as ecology and epidemiology.

The next two papers describe interfaces between statistical modules and a GIS with special focus on public health applications, although the accompanying computing environments are general. The paper by Patrick Wall and Owen Devine introduces MapSpat, a series of routines to carry out spatial smoothing of rates and to compute selected cluster statistics, with particular application to the detection of clusters of high incidence of a disease. MapSpat is developed as an add-on to the MapInfo desktop GIS, and exploits object linking and embedding (OLE) to construct a data and command bridge between the analytical functionality and the GIS. MapSpat has its own user interface that is launched from within MapInfo, but in essence runs in parallel to MapInfo in a multitasking MS Windows environment.

The SAGE package described in the paper by Robert Haining, Stephen Wise and Jingsheng Ma runs under the unix operating system and interfaces with the ArcInfo GIS. Similar to the design underlying SpaceStat, the GIS is used as a visualization device and specialized analytical routines are included in a separate module. SAGE contains a fair number of methods to test for spatial autocorrelation and estimate spatial regression models and is one of the few packages that integrates this with regionalization routines. SAGE implements a limited form of linking, due to limited flexibility in the use of ArcPlot as the visualization engine.

The fourth paper in the issue, by Shuming Bao, Luc Anselin, Doug Martin and Diana Stralberg takes the S-Plus software as the main engine for statistical and numerical analysis, and considers its links to two different types of geographic information systems. The first is the ArcView desktop GIS, which is integrated with S-Plus in a MS Windows environment in the form of an ArcView extension, combined with a collection of functions in a dynamic link library. These functions implement the communication between the two packages via automation technology. In essence, this allows users to call any S-Plus function from within ArcView. The extension also provides some functionality to move spatial data objects (such as spatial neighbors) between ArcView and S-Plus. The second type of linkage is quite different, in that it takes advantage of the Open Geospatial Datastore Interface (OGDI) to connect to a wide range of spatial data formats, both locally as well as over the internet. A set of application programming interface (API) routines facilitate the communication between S-Plus as a statistical server and the Grassland GIS as the visualization engine.

The final paper, by Roger Bivand and Albrecht Gebhardt outlines some examples of how spatial statistical functionality can be developed using the R toolbox, an open source clone of the familiar S (and S-Plus) environment that has been implemented for a number of operating systems, including unix, linux and MS Windows. Spatial data analysis functions include ports of existing S code for point pattern analysis and variogram modeling, as well as some new routines to construct test for spatial autocorrelation and estimate spatial regression models.

In the remainder of this section, these five approaches are compared to each other and to SpaceStat and DynESDA in terms of three important dimensions, generic to any software implementation of spatial data analysis capability. The three dimensions pertain to the overall functionality, the way in which “spatial” characteristics of the data are

handled, and the software environment in which development and delivery are implemented.[10]

4.1 Functionality

The software implementations covered here vary considerably in the range of functionality they incorporate and the types of spatial objects (points or polygons) that can be analyzed. In this context, the GEM package is somewhat of an outlier, given its special focus on boundary identification and analysis, something that is lacking in the other implementations.

DynESDA and MapSpat have a limited range of functionality and emphasize exploratory spatial data analysis, with the former stressing areal data and global and local measures of spatial autocorrelation based on Moran's I. MapSpat has no linking or brushing capability and does not explicitly incorporate the spatial arrangement of areal units in terms of contiguity. Instead distance measures are employed to identify the spatial range for rate smoothing. The cluster statistics included in MapSpat are typical of point pattern analysis software.

SAGE and SpaceStat are more comprehensive in design, with more extensive econometric methods in SpaceStat and more visualization in SAGE. SAGE is also the only package with a reasonable functionality with respect to regionalization (spatial aggregation), although spatially constrained clustering is listed in the target functionality for GEM as well.

The S-Plus links and the R routines offer the possibility of including a virtually unlimited number of functions, provided that the user develops them, or that they can be found in an existing library. The commercial S+Spatialstats set of specialized routines primarily deal with point pattern analysis and geostatistical modeling, as well as a limited degree of spatial regression. Some new tools for ESDA are incorporated in the S+ArcView link (such as LISA statistics), but the main purpose of the link is to provide the full range of statistical functions in S-Plus to the ArcView user. Several of these have been ported to R, but so far R lacks a link to a GIS as well as a specialized object to incorporate adjacency similar to the spatial neighbor object in S-Plus.

Both S-Plus based links and the R libraries offer the possibility of extending the existing functionality, although fairly sophisticated programming skills are needed to accomplish this. By contrast, MapSpat and DynESDA are point and click and require fairly little in terms of computing (and statistical) know-how on the user's part. Both SpaceStat and Sage are menu driven as well, although since they do not follow the MS Windows "standard", their interfaces are arguably somewhat unfriendlier.[11] Compared to the S-Plus and R toolboxes, the others are "closed" packages, in that there is no easy way

---

[10] While fairly representative of the types of computating environments in which spatial data analysis has been implemented, this is not intended to be a comprehensive review. A list of many other software packages and libraries of routines can be found at the software FAQ of http://www.ai-geostats.org.

[11] At the time of writing, there is not yet a general release of GEM, so that the final functionality and user interface are not known.

to add new functions to the current design.[12]

Sage, SpaceStat and DynESDA are focused on the analysis of lattice data, and consequently points are treated as discrete objects, not as a random sample. While MapSpat does deal with areal data such as county mortality rates, its primary emphasis is on rate smoothing and cluster detection for points, and the "areal" aspect of the data are not emphasized. Since both the S-Plus links and the R libraries are extendable, they can in principle include functionality to handle all types of spatial objects, although so far the main emphasis has been on points (either in point pattern analysis or in geostatistical modeling). Finally, GEM has its own conceptualization of spatial objects, which results in a specialized set of functions.

4.2 Spatial Data Handling

It is not surprising that even the little spatial data analytical capability that has become available in commercial statistical software tends to be limited to point pattern analysis and geostatistical modeling. Both sets of methods only require point coordinates as the basis for the computation of inter-observation distances that underlie the test statistics and estimators. Such distance computation can be carried out in a straightforward manner within the data models used by mainstream statistical software.[13]

Point coordinates are stored and manipulated in the same fashion as other data in a statistical package. By contrast, the analysis of lattice or areal data requires an explicit consideration of spatial arrangement, such as contiguity. This information can only be derived from the boundary files for the polygons or from the position of raster cells that represent the spatial objects of interest. Mainstream statistical software is ill-equipped to handle this complication, and if it includes lattice data analysis at all, the spatial weights are typically considered to be given. Of the examples considered here, the R routines in Bivand and Gebhardt do not deal with the construction of weights explicitly, even though they incorporate methods for lattice data analysis. S-Plus, in its S+SpatialStats add-on, handles the spatial arrangement by means of a "spatial neighbor" object, but it is only since the development of bridges between S-Plus and a GIS that such an object can be readily constructed. However, there are still serious limitations to the size of spatial data set for which the neighbor objects can be derived within a reasonable time.[14]

SpaceStat, DynESDA and Sage handle the construction of weights from a GIS internally. Since Sage is built around the ArcInfo GIS, contiguity information is derived directly from the arc-node topology stored in arc attribute tables. On the other hand, ArcView does not have built-in topology and both the SpaceStat Extension and DynESDA construct the spatial arrangement by means of a specialized algorithm applied to the shape file. SpaceStat (without the extension) also contains functions to build spatial

---

[12] In principle, it would be possible to extend the functions in MapSpat and DynESDA by adding additional scripts, although this is not straightforward and the packages are not presented as "open".

[13] Limitations crop up for large data sets, where the data model used in mainstream statistical packages becomes inadequate to store matrices of dimensions N by N.

[14] This is primarily a limitation of the Avenue scripts used to generate the neighbor object.

weights from generic boundary files, both with and without topology. However, due to memory constraints, this cannot be applied to the handling of very large spatial data sets. Neither GEM nor MapSpat deal explicitly with spatial weights.

In general, the lack of functionality to readily construct information on spatial arrangement remains a major impediment for the inclusion of lattice data analysis in mainstream statistical software. The software implementations considered here tackle this problem by developing specialized routines, or side step the issue and leave it up to the user to come up with a practical solution.

4.3 Software Environment

The software environment in which the spatial analytical functionality is developed determines to a large degree the ease with which the pool of potential users is reached. A major impediment to a wider acceptance of spatial analytical tools is no longer primarily the lack of software as such, but the reluctance of many users to leave a familiar interface and package. Software efforts therefore tend to be incremental and add functionality to an existing system, either starting from the GIS end (building onto Arcinfo, ArcView or Mapinfo, for example) or from the statistical end (using statistical toolboxes such as S-Plus or R). Of the software implementations considered here, only GEM and SpaceStat are completely self-contained in the sense that they do not rely on a particular GIS or statistical toolbox. The others only operate in conjunction with either a GIS (Arcinfo for SAGE, ArcView for DynESDA, MapInfo for MapSpat), a statistical toolbox (the R spatial library), or require both (the S+ArcView link and the S+Grassland interface).

In many respects, a freestanding package is ideal for the occasional user, provided the interface is sufficiently self-explanatory and the range of methods included comprehensive enough. Neither GEM nor SpaceStat fit that bill, nor are they intended to. On the other hand, it is unlikely as well as inefficient that a package developed from scratch will contain the range of other statistical or GIS operations that are standard in existing commercial software environments. Building onto ArcView, MapInfo or Arcinfo, as implemented in DynESDA, MapSpat and SAGE or linking to a GIS, as in the S-Plus interfaces provides a ready made solid basis for all spatial data handling and avoids reinventing the wheel. A number of recent developments suggest that similar efforts will become easier to implement in the future. For example, the components in the recently released ArcInfo 8 in principle allow the sophisticated user to develop a freestanding mini GIS with any desired spatial analytical capability. While the development of such a collection of software tools is still rather demanding, once the components become better understood and as long as they remain truly reusable, the barrier to the non-expert programmer will continue to be lowered. Similar developments are occuring on the statistical end, still mostly in the form of reusable libraries, but increasingly also combined with intuitive and customizable user interfaces, such as in the most recent release of S-Plus. On the other hand, open source libaries, such as those developed in R, can be combined with many other open source tools, although the degree of programming sophistication required from the user is still somewhat higher than in the commercial world.

5 Future Directions

Computing environments for spatial data analysis are undergoing rapid change. In part this is driven by new developments in computing technology itself, such as the explosive growth of the importance of the internet and recent emphasis on web delivery of analytical capability. In addition, the demand for spatial analysis has grown as well. Both the need to address new theoretical questions as well as the phenomenal availability of geocoded information have created a demand from researchers and scholars for ever more sophisticated analysis tools (Anselin 1999b; Goodchild et al. 2000).

It is likely that this demand will translate into a drive to expand the range of software tools available to carry out spatial analysis, either in isolation, or in conjunction with existing GIS or statistical/econometric software. In this respect, a number of important new directions for future developments can be suggested. First, new software tools will have to be modular, allowing the mixing and matching of reusable components to address the specific analytical requirements of individual researchers. This suggest the importance of open environments (not necessarily open source) in which the interfaces between software components are well documented and reasonable standards adhered to. While there is a beginning of such an environment with respect to GIS in general (the Open GIS Foundation), there has been little attention paid to this aspect on the analytical side of spatial data handling. Secondly, and related to this, new software tools will need to be able to read and manipulate spatial data from different formats. This is likely to be accomplished by middleware or specialized API that translate various formats into a common structure. Such a common structure can then form the basis for analysis, for example to build spatial weights. Thirdly, the growth of the "internet as the computer" will require considerable research to develop efficient algorithms and delivery mechanisms to overcome the current lack of speed of the internet. Important questions remain about the division of labor between the server and the client in terms of the provided analytical capability. Many technical issues must be resolved before web delivery of analysis will be standard, but it is clearly an essential aspect of the analytical software tools of the future. Fourthly, the potential in terms of added functionality that could result from the fostering of a large community of developers in an open source context should not be underestimated. While it is unlikely that spatial data analysis will attract the same degree of attention as the maintenance and refinement of an operating system such as linux, the leverage of the input and commitment of many rather than a few could be significant. However, such a community can only exist if sufficient awareness and knowledge of the methods themselves has been generated, which is still far from being accomplished. Finally, there is likely to be an increasingly strong mutual reinforcement between spatial statistical and econometric methods and the computational tools to implement them in practice. For example, superior software tools for simulation have revolutionized the estimation of complex hierarchical models. Similarly, one can expect that significant advances in software tools for spatial data analysis will open up new opportunities for methodological and theoretical advances.

The papers included in this issue are a fairly representative sample of the variety of approaches currently in use. It is hoped that a greater familiarity with the different solutions offered here will stimulate further work towards the next generation computing environments for spatial data analysis.

Acknowledgments. This research was funded in part by U.S. National Science Foundation grants SES 88-10917 (to the National Center for Geographic Information and Analysis), SBR 94-10612, and BCS 99-78058 (to the Center for Spatially Integrated Social Science).

References

Anselin L (1988) *Spatial econometrics: methods and models*. Kluwer Academic, Boston

Anselin L (1992) *SpaceStat, a software package for the analysis of spatial data*. National Center for Geographic Information and Analysis, University of California, Santa Barbara

Anselin L (1994) Exploratory spatial data analysis and geographic information systems. In: Painho M (ed) *New tools for spatial analysis*. Eurostat, Luxembourg

Anselin L (1995) Local indicators of spatial association - LISA. *Geographical Analysis* 27: 93-115

Anselin L (1996) The Moran scatterplot as an ESDA tool to assess local instability in spatial association. In: Fischer M, Scholten H, Unwin D (eds) *Spatial analytical perspectives on GIS*. Taylor & Francis, London

Anselin L (1998) Exploratory spatial data analysis in a geocomputational environment. In: Longley P, Brooks S, McDonnell R, Macmillan B (eds) *Geocomputation, a primer*. John Wiley, London

Anselin L (1999a) Interactive techniques and exploratory spatial data analysis. In Longley P, Goodchild M, Maguire D, Rhind D (eds) *Geographical information systems*, 2nd edition. John Wiley, New York

Anselin (1999b) The future of spatial analysis in the social sciences. *Geographic Information Sciences* 5: 67-76

Anselin L, Bao S (1997) Exploratory spatial data analysis linking SpaceStat and ArcView. In: Fisher M, Getis A (eds) *Recent developments in spatial analysis.* Springer, Berlin

Anselin L, Bera A (1998) Spatial dependence in linear regression models with an introduction to spatial econometrics. In: Ullah A, Giles D (eds) *Handbook of applied economic statistics*. Marcel Dekker, New York

Anselin L, Getis A (1992) Spatial statistical analysis and geographic information systems. *Annals of Regional Science* 26: 19-33

Anselin L, Smirnov O (1996) Efficient algorithms for constructing proper higher order spatial lag operators. *Journal of Regional Science* 36: 67-89

Anselin L, Smirnov O (1999a) The Spacestat extension for ArcView. http://www.spacestat.com/spex.htm

Anselin L, Smirnov O (1999b) The DynESDA extension for ArcView. http://www.spacestat.com/utilities.htm

Anselin L, Dodson R, Hudak S (1993) Linking GIS and spatial data analysis in practice. *Geographical Systems* 1: 3-23

Buja A, Cook D, Swayne D F (1996) Interactive high-dimensional data visualization. *Journal of Computational and Graphical Statistics* 5: 78-99

Cleveland W S, McGill M E (1988) *Dynamic graphics for statistics*. Wadsworth, Pacific

Grove
Cook D, Symanzik J, Majure J J, Cressie N (1997) Dynamic graphics in a GIS: more examples using linked software. *Computers and Geosciences* 23: 371-385
ESRI (1995) *ArcView version 2 shapefile technical description. White Paper*. Environmental Systems Research Institute, Redlands CA
Goodchild M F (1987) A spatial analytical perspective on geographical information systems. *International Journal of Geographical Information Systems* 1: 327-334
Goodchild M F, Haining R P, Wise S et al (1992) Integrating GIS and spatial analysis - problems and possibilities. *International Journal of Geographical Information Systems* 6: 407-423
Goodchild M F, Anselin L, Appelbaum R, Harthorn B (2000) Toward spatially integrated social science. *International Regional Science Review* 23: 139-159
Haslett J, Wills G, Unwin A (1990) Spider, an interactive statistical tool for the analysis of spatially distributed data. *International Journal of Geographical Information Systems* 4: 285-296
Haslett J, Bradley R, Craig P, Unwin A, Wills G (1991) Dynamic graphics for exploring spatial data with applications to locating global and local anomalies. *The American Statistician* 45: 234-242
Messner S, Anselin L, Baller R, Hawkins D, Deane G, Tolnay S (1999) The spatial patterning of county homicide rates: an application of exploratory spatial data analysis. *Journal of Quantitative Criminology* 15: 423-450
Smirnov O, Anselin L (2000) Fast maximum likelihood estimation of very large spatial autoregressive models: a characteristic polynomial approach. *Computational Statistics and Data Analysis* (forthcoming)
Symanzik J, Kötter T, Schmelzer S, Klinke S, Cooke D, Swayne D (1997) Spatial data analysis in the dynamically linked ArcView/XGobi/XploRe environment. *Computing Science and Statistics* 29: 561-569
Unwin A (1996) Exploratory spatial analysis and local statistics. *Computational Statistics* 11: 387-400

Category	Functionality	Report Files
Input	Creation of SpaceStat data sets and spatial weights files from Ascii input	
Merge/Select	Manipulation of SpaceStat data sets (merging data sets, adding and deleting observations or variables, subsetting); subsetting spatial weights	
Variable Create	Constructing constants, observation numbers and dummy variables, random variates, relabeling variables	
Variable Transform	Standard data transformation functions (log, exp, standardization, etc.)	
Spatial Transformation	Spatial lag, spatial moving average, spatial filter and spatial transformation (moving average and autoregressive), non-contiguous random resample	Sptran.txt
Rate Transformation	Constructing proportions, standardizations (Freeman-Tukey, arcsin, Anscombe, Empirical Bayes), smoothing (Empirical Bayes, spatial window, spatial Empirical Bayes)	Sptran.txt
Variable Algebra	Addition, subtraction, etc. of variables, trend surface polynomials, regime variables, expansion variables, principal components	
Matrix Algebra	Element by element manipulations, matrix multiplication and inverse, determinant and trace	
List	Summary of contents of SpaceStat data set and weights files, listing of contents or selected variables/observations of SpaceStat data sets, listing of contents of spatial weights files	Data.txt

Table 1. Data Functionality in SpaceStat

Category	Functionality
Weight Characteristics	Connectivity structure (most/least connected, unconnected, frequency table of neighbors), dominant root, eigenvalues, traces
Weight Transformations	Row-standardization and higher order contiguity, element by element manipulation of spatial weights, boundary shares over distance weights, dissolve areal units in weights file
Weight Conversion	Conversion between three spatial weights formats (full matrix, sparse contiguity, sparse general), relabeling and sorting elements of weights files
Distance Weights	Computing distance matrices and construction of spatial weights based on distance (contiguity, inverse distance, inverse distance power), k-nearest neighbors
Sparse Distance Weights	Same as distance weights but using sparse formats rather than full matrix
Access Measures	Computation of origin-destination pair distance and various measures of accessibility (potential, travel cost, covering)
Raster Weights	Construction of contiguity weights for regular grids using rook, bishop or queen criterion, resampling based on coding approach
GIS Functions	Generic functions to construct spatial weights and centroids from ascii input files (e.g., Arc/Info AAT files, boundary files)

Table 2. Tools Functionality in SpaceStat

Category	Functionality	Report Files
Descriptive statistics	Non-spatial descriptive statistics, quartiles, percentiles, outliers, correlations, principal components	Boxmap.txt
Join count statistics	Binary and multinomial join count statistics with inference based on a normal approximation (non-free sampling) and a permutation approach	
Moran	Moran's I statistic for global spatial autocorrelation with inference based on normal approximation, randomization and permutation, spatial correlogram, Moran scatterplot, local Moran	Morscat.txt LM_data.txt
Geary	Geary's c statistic for global spatial autocorrelation with inference based on normal approximation, randomization and permutation, spatial correlogram	
G-statistics	Global G statistic for spatial autocorrelation, local G_i and Gi* statistics	GI_data.txt
QAP	Combinatorial statistics for Moran's I, Geary's c and Sokal absolute difference, generic matrix comparison	

Table 3. Explore Functionality in SpaceStat

Model	Methods
Classic Model	Ordinary Least Squares (OLS)
	OLS Robust (White, Jackknife)
	Weighted Least Squares
Spatial Error Model	Spatial Autoregressive Error (SAR), maximum likelihood estimation (ML)
	SAR Error with groupwise heteroskedasticity (e.g., spatial regimes), ML
	SAR Error with weighted regression, ML
	Spatially weighted least squares (interactive)
	SAR Error, generalized moments (GM) estimator (two-step)
	SAR Error, GM estimator (iterated)
	SAR Error with groupwise heteroskedasticity, GM estimator
Heteroskedastic Error Model	Generic heteroskedasticity (user-specified), feasible generalized least squares (FGLS)
	Generic heteroskedasticity (user-specified), ML
	Groupwise heteroskedasticity (FGLS)
	Groupwise heteroskedasticity (ML)
	Random coefficients (FGLS)
	Random coefficients (ML)
Spatial Lag Model	Spatial Autoregressive Lag (SAR), ML
	SAR with groupwise heteroskedasticity, ML
	SAR, two stage least squares (2SLS)
	SAR with groupwise heteroskedasticity, 2SLS
	SAR, robust 2SLS
	SAR, bootstrap
Systems Model	Endogenous variables, 2SLS
	Endogenous variables with groupwise heteroskedasticity, 2SLS-GM
	Endogenous variables, robust 2SLS
	Endogenous variables with SAR error autocorrelation, GM-2SLS
	Endogenous variables with SAR error autocorrelation and groupwise heteorskedasticity, GM-2SLS

Table 4. Regress Functionality in SpaceStat - Estimation Methods

Fig. 1. The Data Menu in the SpaceStat Extension for ArcView

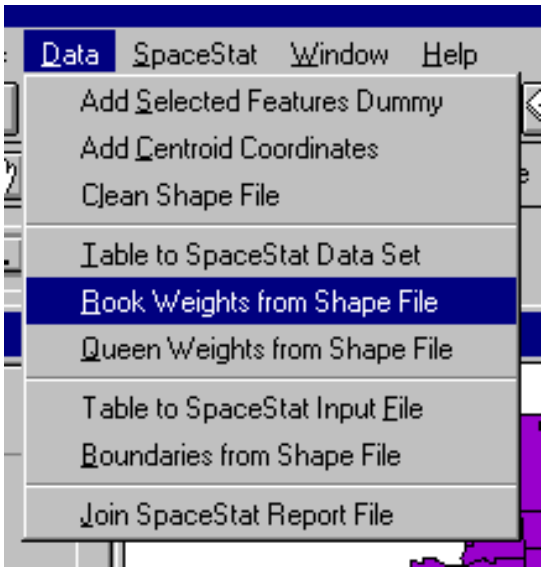

Fig 2. The SpaceStat Menu in the SpaceStat Extension for ArcView

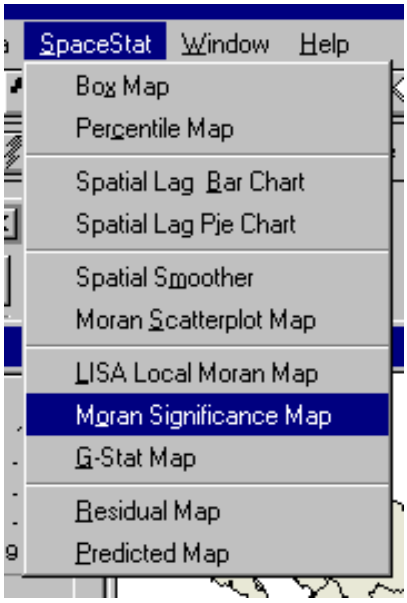

Fig. 3. The Floating Toolbar for the DynESDA Extension for ArcView