Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *http://www.jstatsoft.org/*

VRMLGen: An R-package for 3D Data Visualization on the Web

Enrico Glaab **Jonathan M. Garibaldi** **Natalio Krasnogor**
University of Nottingham, School of Computer Science

Abstract

The 3-dimensional representation and inspection of complex data is a frequently used strategy in many data analysis domains. Existing data mining software often lacks functionality that would enable users to explore 3D data interactively, especially if one wishes to make dynamic graphical representations directly viewable on the web.

In this paper we present **vrmlgen**, a software package for the statistical programming language R to create 3D data visualizations in web-formats like the Virtual Reality Markup Language (VRML) and LiveGraphics3D. **vrmlgen** can be used to generate 3D charts and bar plots, scatter plots with density estimation contour surfaces, and visualizations of height maps, 3D object models and parametric functions. For greater flexibility, the user can also access low-level plotting methods through a unified interface and freely group different function calls together to create new higher-level plotting methods. Additionally, we present a web-tool allowing users to visualize 3D data online and test some of **vrmlgen**'s features without the need to install any software on their computer.

Keywords: vrml, 3d, visualization, graph, chart, plot.

1. Introduction

3D visualizations for data mining provide an intuitive means to identify underlying structures and patterns in a data set which would otherwise often remain undetected.

Software tools to create interactive graphical data representations in 3D can be used effectively both to explore inherently 3-dimensional data and subspaces and projections of higher dimensional data. The latter task is becoming increasingly important, as companies and research institutions are striving to fully exploit their vast data resources. Moreover, in the last decade, a large number of data bases containing complex, high-dimensional data have been published for general access on the web. These include, for example, data resources for the life sciences (Edgar, Domrachev, and Lash 2002; Widera, Garibaldi, and Krasnogor 2009;

World Health Organization 2009), physical sciences (Zucker, Kishore, Sukumar, and Dragoset 2009; Oak Ridge National Laboratory, Physics Division 2009; Department of Physics and Astronomy, University of Kentucky 2009), social sciences (Council of European Social Science Data Archives (CESSDA) 2009; UK Data Archive (UKDA), University of Essex 2009), and business and finance (Pettijohn 1996; Ohio State University, Department of Finance 2009; NASDAQ OMX 2009).

For these reasons, a great variety of software tools has been made available in the past to generate 3D data representations with interactive means to explore different perspectives, analyze class membership and data density using color schemes and contour surfaces, or compare and interlink different data sources using overlay-plots and hyperlinks. The widely used statistical programming language R (Ihaka and Gentleman 1996; R Development Core Team 2009a), for example, contains several software packages for 3D data analysis including a package to interconnect R with OpenGL (Adler, Nenadic, and Zucchini 2003) and various packages for multivariate data visualization (Feng and Tierney (2005); Sarkar (2008); Ligges, U. and Maechler, M. (2003); see section 4 for a more detailed discussion).

However, though users can choose between many freely available offline-tools to inspect 3D data on their own computer, it is still a difficult task to make 3D visualizations publicly available and directly viewable on the web. Currently available software tools or packages for 3D plotting do not provide features for a web-based interactive exploration of the data. Apart from speed limits on the internet for loading large data sets, other restrictions arise from the fact that different web browsers do not support the same graphics file formats and the 3D-visualization plugins are often only compatible with a limited number of browsers.

Here, we present **vrmlgen**, a software package for the R statistical data analysis environment, which enables users to generate interactive web-based visualizations for 3D input data. Charts, graphs, bar plots and scatter plots, 3D meshes and parametric functions are visualized in two common web-formats, VRML (Virtual Reality Markup Language) and LiveGraphics3D. Files in the common VRML-format (Ames, Nadeau, and Moreland 1997) enable an interactive virtual navigation in complex 3D models and can be displayed on the web with free plugins available for every common browser platform. The Java- based LiveGraphics3D format by Kraus (2003), which is supported as an alternative to VRML, provides a somewhat reduced user interaction functionality and is based on the simple "painter's algorithm" to hide occluded surfaces, but it can be viewed in any Java-enabled browser without the need to install a plugin. **vrmlgen** is freely available subject to the GNU General Public Licence (GPL version 3).

In addition to the R package, we provide a simple web-application to illustrate some of **vrmlgen**'s main features and enable the user to easily generate colored and labeled 3D visualizations for uploaded data without the need to install any software (available at `http://www.infobiotics.org/vrml`). Similarly, the R package is employed on a bioscientific webserver for DNA microarray analysis, ArrayMining.net (Glaab, Garibaldi, and Krasnogor 2009; Infobiotics Project 2009), to visualize clustering results in a low-dimensional representation.

The structure of the paper is as follows: We will present **vrmlgen**'s features and code examples (section 2), discuss the implementation (3), extensions and related software (4), provide installation instructions (5) and we conclude the paper with a summary and conclusion section (6).

vrmlgen and the R software environment can be obtained from the Infobiotics project repos-

itory (www.infobiotics.net) and the Common R Archive Network (CRAN, http://cran.R-Project.org). VRML-files can be inspected either with offline viewers, e.g. "BS Contact VRML 7.1" (Bitmanagement 2009), or browser plugins, e.g. "Cortona 3D viewer 6.0" (ParallelGraphics 2009) (for further software alternatives, see Web3D Consortium (2005)).

2. Features and examples

In this section we will first give an overview of **vrmlgen**'s main functions and then discuss each of the methods in detail with source code examples for typical applications.

2.1. Overview of available functions

The **vrmlgen** package contains both higher-level plotting functions for creating common data visualizations like scatter plots, bar charts and mesh visualizations, and lower-level methods, enabling users to draw primitive shapes and objects through a unified, format-independent interface. Moreover, both higher- and lower-level function calls can be grouped together to build a 3D scene, using a mechanism similar to other R graphics devices (see section 2.2).

Many higher-level plotting tasks can be realized directly with a single function call of one of **vrmlgen**'s three main plotting functions:

- the *cloud3d()*-function for creating 3D scatter plots and surfaces,
- the *bar3d()*-function for generating bar plots and height map visualizations, and
- the *mesh3d()*-function for displaying parametric functions and 3D meshes.

To change the output format for a plot (VRML or Livegraphics3D) only a single parameter has to be modified. Other format-specific parameters are highlighted by a prefix ("vrml_" or "lg3d_") and are set to commonly used default values to reduce the number of required adjustments. Fig. 1 provides an overview of the main plot types and functions available in the **vrmlgen** package.

Examples of how each of these main plotting functions can be used to generate different types of 3D data visualizations and customize the properties of a plot will be given in the following sections. Apart from the web-based outputs that can be obtained with these functions, perspective plots of 3D data can also be rendered in high-quality and used in scientific publications by combining **vrmlgen** with other free software tools, for example to convert VRML-output into the POV-Ray-format (Thiessen 2004; Plachetka 1998).

In contrast to other plotting functions in R and related statistical software tools, **vrmlgen** provides features related to the interactive exploration and HTML-embedding of the 3D data. For example, the cloud3d-function allows users to specify "meta-labels" for data points in VRML-scenes, i.e. textual information which is displayed as a message box when the user moves the mouse over a data point in a 3D plot (available in all Javascript-enabled VRML-plugins). Moreover, to exploit the advantages of web-based graphics formats, hyperlinks can be added to each object in a 3D scene, and **vrmlgen** can automatically generate an HTML-template to embed the created 3D plots in a web page.

To simplify the use of **vrmlgen** for experienced R users, typical plotting features (e.g. color, format and scale settings) and the function and parameter names have been chosen so as

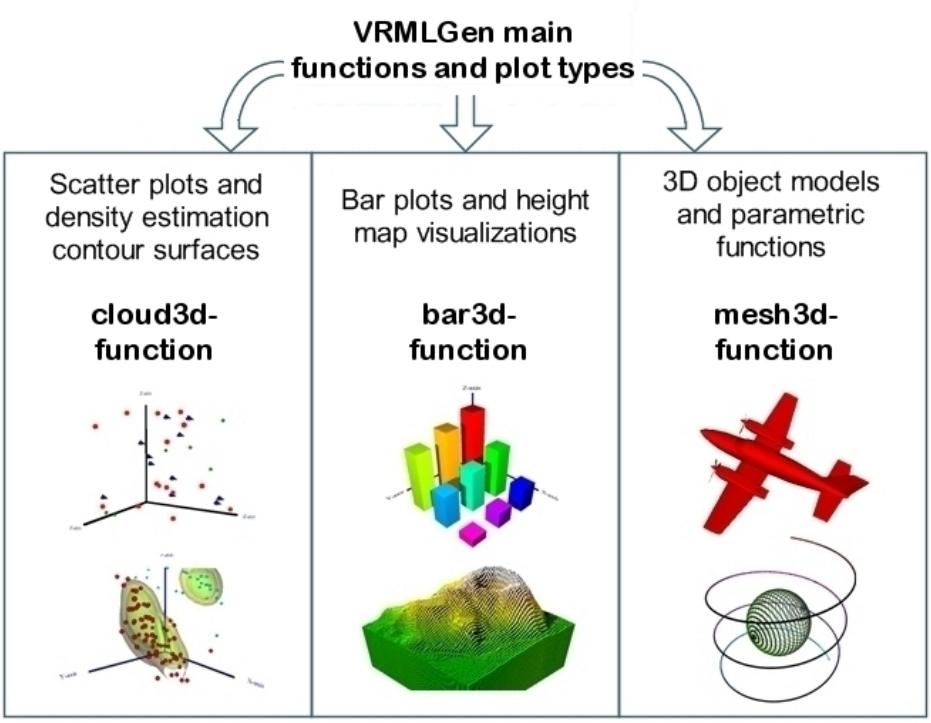

Figure 1: Overview of **vrmlgen**'s main functions and features

to maintain familiar aspects from related 2D plotting functions and "offline"-methods for 3D-visualization in R.

Below we discuss **vrmlgen**'s functions and features in detail, providing code examples for typical example applications of different types of 3D plots.

2.2. Combining primitive shapes to create complex 3D-scenes

vrmlgen provides direct access to low-level plotting functions for drawing points (*points3d()*), lines (*lines3d()*), text (*text3d()* and coordinate axes (*axis3d()*) through a format-independent interface, enabling users to switch between the VRML- and the Livegraphics3D-output format by only changing the *type*-parameter (see examples below). Alternatively, for users who wish to quickly generate common data visualizations in 3D, like scatter plots, bar charts or mesh visualizations, the higher-level functions *cloud3d()*, *bar3d()* and *mesh3d()* are available and only require the user to configure relevant parameters (these methods are discussed in detail in the following sections).

In addition to the functionality provided by the single plotting methods, both low- and high-level function calls can be combined together by placing them between a *vrml.open()* and a *vrml.close()* statement (or respectively, *lg3d.open()* and *lg3d.close()* statements to create Livegraphics3D-output instead of VRML-output). This design imitates the architecture of other graphics devices in R, like *pdf()* or *png()*, and enables the user to build a complex

3D-scene by sequentially adding new objects to it before creating the final output by calling *vrml.close()* or *lg3d.close()*. General settings for a 3D scene like the background color, the viewing perspective, or a global scaling factor to increase/decrease the size of all objects in the scene, can be configured with the parameters of the *vrml.open()* and *lg3d.open()* functions and will be inherited by all plotting methods called afterwards.

Fig. 2 shows an example 3D-scene that has been created using the vrml.open()- and vrml.close() mechanism (see source code in Fig. 3). After setting the global parameters in *vrml.open()* (here: applying a scaling factor of 5 to the scene and embedding the VRML-output in an HTML-file), a dataset of atom coordinates is loaded and the atoms are drawn into a 3D scene with the *points3d()* function. To illustrate how **vrmlgen** can profit from the built-in functionality of the R statistical programming language, lines to represent the bonding patterns between atom pairs are not drawn by reading an additional input file, but by using R statements to draw lines between all atom coordinates that fulfill a certain distance criterion (the Euclidean distance has to be smaller than 0.66 units in the VRML-scene).

Thus, by combining low-level plotting functions with **R**-based data processing and manipulation, the user can create complex 3D-object models and environments that can be interactively explored in a web-browser. Additionally, 3D-mesh models, surfaces defined by parametric functions or height maps can be added to a VRML- or Livegraphics3D-scene or the user can write self-defined higher-level plotting functions by combining existing plotting methods in new R-functions. **vrmlgen**'s pre-defined higher-level plotting methods are discussed in the following sections.

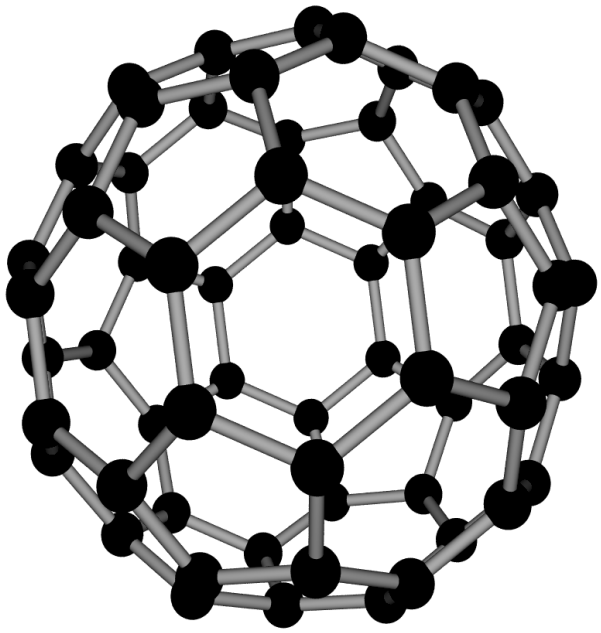

Figure 2: Visualization of a C60 molecular structure by combining primitive shapes (spheres and cylinders, see source code in Fig. 3)

```
vrml.open(file="c60.wrl", html.embed="c60example.html", scale=5)

    # load dataset
    data(c60coords)

    # plot the atoms as black spheres and add hyperlinks
    points3d(c60coords, col="black", scale = 5,
           hyperlinks=rep("http://wikipedia.org/wiki/Carbon",60))

    # plot the atom bonds as gray lines
    for(j in 1:(nrow(c60coords)-1))
    {
       for(k in (j+1):nrow(c60coords))
       {
          if(sqrt(sum((c60coords[j,]-c60coords[k,])^2)) < 0.66)
             lines3d(c60coords[c(j,k),], col="gray", lwd=3)
       }
    }
vrml.close()

# show the output in a web-browser (VRML-plugin required)
browseURL(paste("file://",file.path(getwd(),
          "c60example.html"), sep=""))
```

Figure 3: Reading atom coordinates of a C60 molecule and visualizing the atoms by black spheres (see Fig. 2). To represent the atom bonds, cylinders are drawn between all spheres with a Euclidean distance smaller than a threshold value (0.66). The *html.embed* parameter specifies an HTML-file that embeds the VRML-output, the *scale* parameter determines the size of the entire 3D scene and the *hyperlinks* parameter in the *points3d*-function attaches a web link to each plotted sphere (in the example, a link to the Wikipedia page on carbon atoms).

2.3. Scatter plots and density estimation contour surfaces

The most common format for representing 2D and 3D statistical data are scatter plot visualizations. In this section we will illustrate how the **vrmlgen** package allows users to translate 3D point data into interactive scatter plots in VRML- or LiveGraphics3D-format using the cloud3d-function. Apart from the basic functionality, additional features like meta-labels and density estimation contour surfaces will be discussed based on an example dataset from a breast cancer microarray study.

To obtain a simple 3D scatter plot, the user only needs to provide the input in a format that can be coerced to a numerical matrix with 3 columns, e.g. 3 vectors with equal length (the *xyz.coords()*-function from the R **base** package is used to provide this functionality). All other function-parameters are optional, but enable the user to add meta- information to the plot or to customize the color-, perspective- and scaling- properties according to the user's

needs. An example based on a public breast cancer microarray dataset (Spiteri 2008) after dimensionality reduction using Independent Component Analysis is shown in Fig. 4 together with the source code to reproduce the image (see Fig. 5).

By providing the cloud3d-function with the class information for the data points (here the two classes are the breast cancer tumor grades 1, for the least aggressive tumors, and 3, for the most severe tumors), samples from different classes are automatically represented by different colors and different point styles and a legend is generated above the Z-axis. Apart from the color-labeled data points, density estimation contour surfaces can optionally be added to the plot (Fig. 4, right), which represent regions of high data density in the plot (visualized by green and yellow surfaces, with green representing the highest data density). In the breast cancer example dataset, interestingly, most of the low-grade tumor samples appear to be clustered together in a high- density region of the plot, whereas the high-grade tumor samples are scattered more widely across the plot. The contour surface feature was implemented following the example of the R-package **sm** (Bowman and Azzalini 2008) and makes use of the *contour3d*-function from the **misc3d**-package (Feng and Tierney 2005). Additional textual information can be included in a plot by specifying a vector of labels for the samples using the *metalabels*- parameter. When the user moves the mouse over a data point in a VRML- visualization, the corresponding label will be shown as a tool-tip message above the Z-axis.

In order to create plots for high-dimensional datasets, common dimensionality reduction methods can be applied before passing the data to the visualization functions. Examples for freely available functions and packages in R for dimensionality reduction include *prcomp* from the **stats** package to compute the principal components of a data set and the **fastICA** package for Independent Component Analysis, which was used for the breast cancer example data.

2.4. Parametric functions and 3D meshes

Apart from plotting matrix data sets, **vrmlgen** can also be used to visualize models of 3D objects and parametric functions defining 3D surfaces. This functionality is available in the mesh3d-function, which is used in a similar manner as the cloud-function discussed in the previous section. The user can either directly specify the vertices and edges of a 3D mesh, formulate a parametric function which defines a 3D surface or load a 3D-model file in the common obj-format (the obj-input can only be used when creating VRML-output, because the Livegraphics3D-format does not support the specification of polygonal faces based on vertex-identifiers). Additionally, obj-files can be created as output, when the input data contains both vertex-coordinates and edge-indices and the parameter *write_obj* is set to "TRUE".

To visualize 3D parametric curves and surfaces, one parametric function for each dimension has to be specified (using the *xfun*, *yfun* and *zfun*-variables) together with the function parameters (*param1* and *param2*) and their value ranges (*range1* and *range2*). Additional configurations for the mesh size, the color, scaling and initial perspective can optionally be adjusted. Fig. 6 shows an example representation of a parametric function describing a spiral, which can be generated by calling the mesh3d-function (see Fig. 7).

If the data size of the obtained output is too large for a presentation on the web, the mesh-resolution can easily be reduced by choosing smaller values for the "size"-parameters.

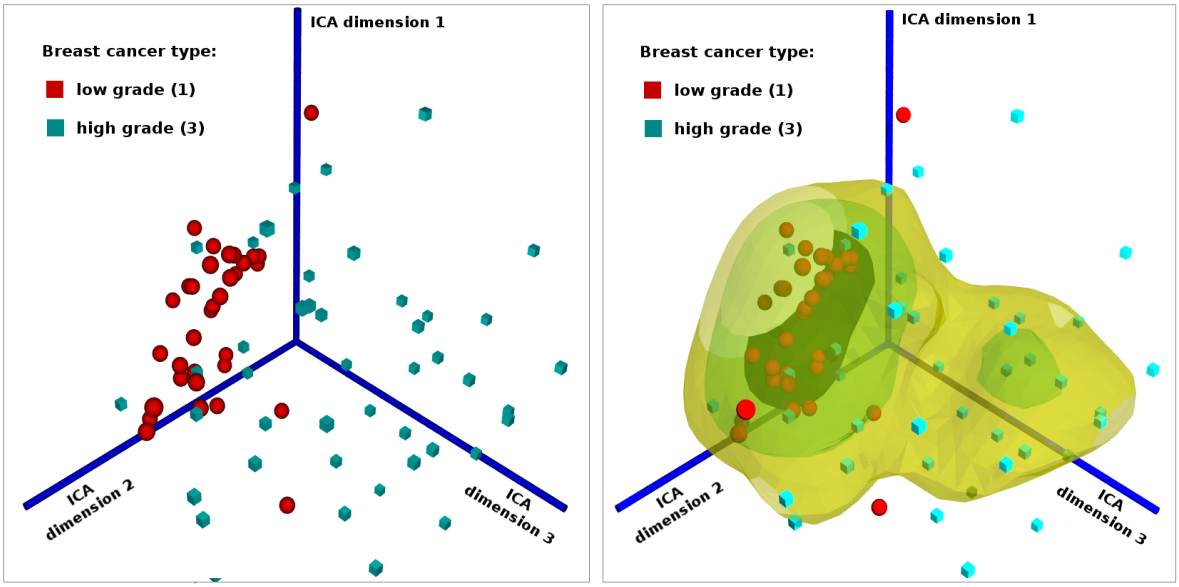

Figure 4: Left: Scatter plot visualization of breast cancer microarray samples (Spiteri 2008) after dimensionality reduction with Independent Component Analysis. Right: The same scatter plot with density estimation contour surfaces (yellow = high data density regions, green = region of highest density), see source code in Fig. 5).

In addition to plotting curves and surfaces defined by parametric functions, **vrmlgen** supports the visualization of polygon meshes representing the shape of arbitrary geometric objects. These 3D mesh models in are widely used in computer graphics and solid modeling and can be generated with the *mesh3d*-function by either manually passing the data for the vertices, edges and faces or by loading a 3D-model file in the Wavefront obj- format (Murray and Van Ryper 2005). The obj-format is one of the most common file formats for geometric modeling and is supported by the great majority of 3D designing tools. Thus, users have the possibility to create 3D models with their preferred modeling software and use the mesh3d-function in **vrmlgen** also be used as a simple conversion tool to transform obj-models into the VRML- format for publication on the web.

Fig. 8 provides an example visualization of an obj-file containing a 3D model of a city block (the model file was obtained from Genovese (2009), the source code is shown in Fig. 9). The same visualization type can be used to obtain representations of arbitrary architectural, industrial or scientific objects, e.g. engineering prototypes, prosthetics and molecular structures.

2.5. Bar plots and height map visualizations

Bar plots often provide a more simple and clear way to visually compare values for different variables than scatter plots. Therefore, the *bar3d()*- function has been added to **vrmlgen** to quickly transform a data matrix into a 3D bar plot which can be viewed from different perspectives in a web-browser. Additionally, this function supports the visualization of height maps using geographical data or gray scale image maps as input.

```
# load dataset and class labels
data(bc_dat)
data(bc_classes)

cloud3d(bc_dat, labels=paste("tumor grade",
        as.numeric(bc_classes)), vrml_showdensity=TRUE,
        metalabels = paste("sample",1:nrow(bc_dat)),
        lab.axis=paste("ICA dimension",1:3),
        filename="example2.wrl", htmlout="example2.html")

# show the output in a web-browser (VRML-plugin required)
browseURL(paste("file://",file.path(getwd(),
          "example2.html"), sep=""))
```

Figure 5: Generating a 3D VRML scatter plot for a breast cancer gene expression dataset obtained after dimensionality reduction with Independent Component Analysis (see Fig. 4). Density estimation contour surfaces (*vrml_showdensity*), meta-labels for the samples (*meta-labels*) and user-defined axis labels (*lab.axis*) are provided, and the output is embedded in an HTML-file (*htmlout*, if this parameter is not provided, only the VRML-output will be generated).

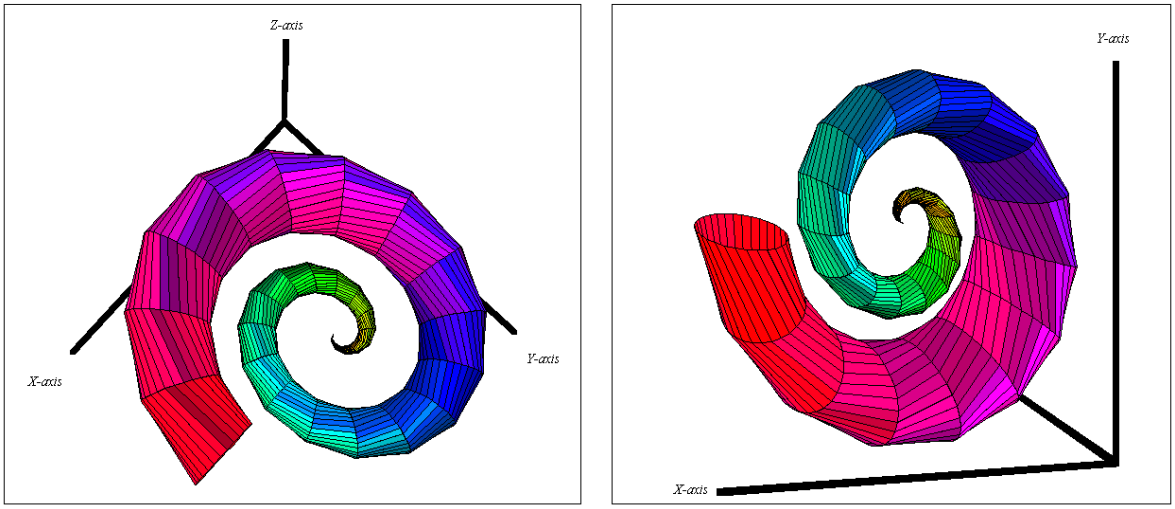

Figure 6: Mesh-visualization of a parametric function representing the surface of a spiral (seen from two different perspectives, see source code in Fig. 7)

Fig. 10 shows a simple bar plot example for a data set comparing the lung death rates in the UK for different months from 1974 to 1979 (Diggle (1990), available in the **dataset** package in R). To simplify the interpretation of the plot, colors can be set per row or per column of the input data (see source code in Fig. 11), without needing to specify the color for each single bar and axis label in the plot.

```
mesh3d(xfun="s*cos(s)*(4 + cos(t+s))",
       yfun="s*sin(s)*(4 + cos(t+s))",
       zfun="s*sin(t+s)", param1="s",
       param2="t", range1=c(0,4*pi),
       range2=c(0,2*pi), cols=NULL,
       filename="example3.m",
       type="lg3d", htmlout="example3.html")

# show the output in a web-browser
# (must be Java-enabled)
browseURL(paste("file://",file.path(getwd(),
          "example3.html"), sep=""))
```

Figure 7: Visualization of a parametric function defining the surface of a 3D spiral (see Fig. 6). The function depends on two parameters $s \in [0, 4 \cdot pi]$ and $t \in [0, 2 \cdot pi]$), for which the variable names and value ranges can be specified with the *param* and *range* parameters. The parametric surface is defined using the *xfun*, *yfun* and *zfun* parameters (if the "cols"-parameter is set to NULL, rainbow colors are used by default to visualize the surface).

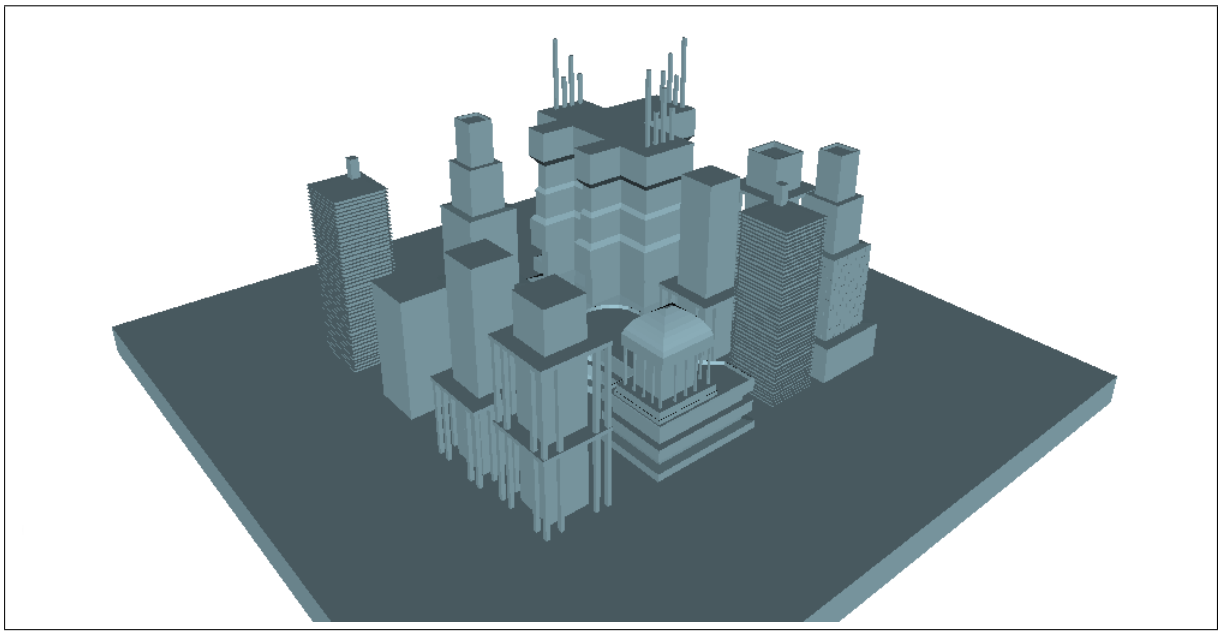

Figure 8: Visualization of a 3D mesh model of buildings in a city block (model source: Genovese (2009), see source code in Fig. 9)

An example for a height map visualization can be seen in Fig. 12 (generated from a highly simplified and rough topographic gray-scale image of the United Kingdom (Sintzel 2010) by setting all grey values below a certain threshold (0.53) to zero and using the remaining values to generate the visualization, see Fig. 13). The reader should note that complex height maps will require more processing time and depending on the size of the data, they might not

```
mesh3d(obj_infile="cityblock.obj", filename="cityblock.wrl",
       cols="lightblue", showaxis=F, htmlout="example4.html")

# show the output in a web-browser
# (VRML-plugin required)
browseURL(paste("file://",file.path(getwd(),
          "example4.html"), sep=""))
```

Figure 9: Reading a 3D-mesh input file and transforming it into a web-based 3D-model (no coordinate axes are drawn, the model is colored light-blue and embedded into a web-page, see Fig. 8).

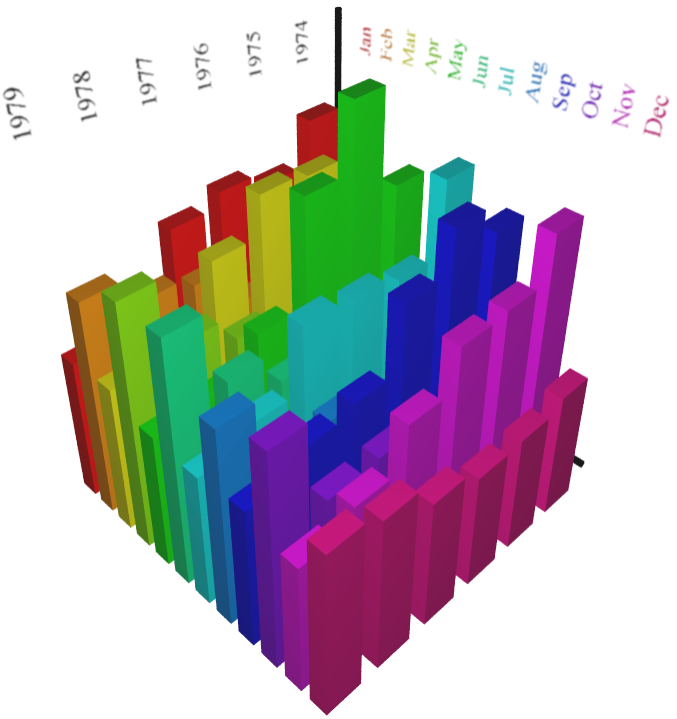

Figure 10: Bar plot showing the monthly death rates from lung diseases in the UK from 1974 to 1979 (left axis: age groups, right axis: population group; data obtained from "ldeaths" dataset in R (Diggle 1990), see source code in Fig. 11)

always be suitable for presentation on the web.

2.6. Navigation- and web-related features

When displaying 3D data on the web, the content provider might not only want to specify the visual properties of a plot but also configure how the user can navigate through the displayed data and interact with it using web-related features like hyperlinks or Javascript-based functions.

```
bar3d(matrix(ldeaths, nrow=6), row.labels=1974:1979,
       col.labels=month.abb, cex.lab=1.5,
       ccols=rainbow(12), filename="example5.wrl",
       type="vrml", lab.vertical=TRUE,
       htmlout="example5.html", autoscale=TRUE, scale=4.0)

# show the output in a web-browser
# (VRML-plugin required)
browseURL(paste("file://",file.path(getwd(),
          "example5.html"), sep=""))
```

Figure 11: Creating a 3D bar plot with column-dependent colors and vertical labels (see Fig. 10)

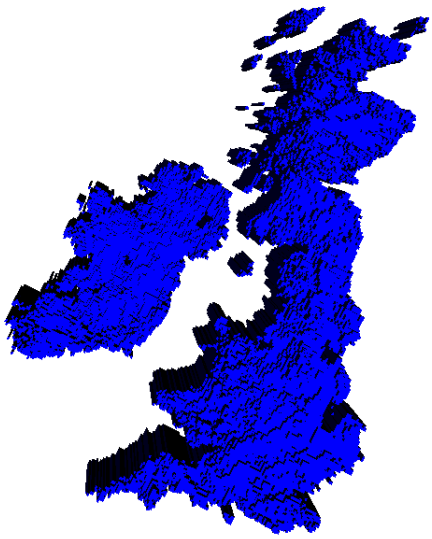

Figure 12: Height map visualization of a rough topographic map for the United Kingdom (see source code in Fig. 13)

For this reason, **vrmlgen** enables the user to specify a default mouse navigation type for VRML plots, which determines how the VRML-browser responds to mouse movements for exploring a virtual 3D space. The five available navigation types - ”EXAMINE” (default), ”WALK”, ”SLIDE”, ”FLY” or ”PAN” - can be selected with the *navigation* parameter and provide different ways to inspect a plot by applying various types of simple rotational or translational movements to the virtual camera.

In the cloud3d-function a hyperlink can be assigned to each data point displayed in a scatter plot using the *hyperlink*-parameter (for an example, see Fig. 3). Thus, a visualization can be inter-linked with additional information like text and images on user-created web-pages or with public, web- based data bases. For example, in the biosciences a typical usage scenario could be to link genes in a low-dimensional representation of gene expression data with the entries of a functional annotation data base.

```
# load height map data
data(uk_topo)

bar3d(uk_topo, autoscale=FALSE, cols="blue",
       space=0, showaxis=FALSE, filename = "example6.wrl",
       htmlout="example6.html")

# show the output in a web-browser
# (VRML-plugin required)
browseURL(paste("file://",file.path(getwd(),
          "example6.html"), sep=""))
```

Figure 13: Height map visualization of a rough topographic map for the United Kingdom (see Fig. 12). No scaling is applied (autoscale) and the bars are printed in blue (cols) without any space between them (space). The plot is drawn without axes (showaxis) and is embedded in an HTML-page (htmlout).

As an alternative to hyperlink-labels, additional textual information on each data point can also be included directly in the plot, using the *metalabels*-parameter in the cloud3d-function (see Fig. 5). This feature makes use of the Javascript facilities in VRML and provides the user with quick access to pre-defined text labels, which are displayed when the mouse cursor hovers over a data point.

2.7. Web-tool for demonstration purposes

To enable a prospective user to get an idea of some of **vrmlgen**'s features and how plots can be generated and inspected, we have created a simple web-tool based on our R-package (available at http://www.infobiotics.org/vrml). A user can paste a 3D data matrix in tab-delimited format on the web-interface, with optional columns for color labels or additional textual information (see "meta-labels" discussed in the previous section), and submit the data with a single mouse-click to obtain the outcome generated with **vrmlgen**. 3D plots will be generated both in VRML- and the LiveGraphics3D-format. Moreover, the web-application provides various example data sets which can be selected, freely modified and visualized by the user. Since the amount of data which can be uploaded is limited and not all of **vrmlgen**'s features can easily be made accessible on the web, this online tool is mainly provided for demonstration and test purposes. However, for simple visualization tasks the web-interface can be a fast and simple way to obtain a colored, labeled and navigable data plot without having to install any software on the client computer.

3. Implementation

vrmlgen is written entirely in the R programming language and can be installed on all operating systems for which the R software is available. The variable names in the function bodies have been adapted to follow common R naming conventions and help and examples for all functions can be obtained by typing `?vrmlgen` at the R command prompt. A dedicated web-

site with installation and usage instructions as well as example output pictures is available at http://www.infobiotics.org/vrmlgen. Additionally, we have implemented a web-tool for demonstration purposes (see previous section), which is based on a PHP-interface running on an Apache web- server and enables users to access some of **vrmlgen**'s main functions without the need for a local software installation.

The R package was build following the recommendations in the official "Writing R Extensions" tutorial by the R Development Core Team (2009b) and has been published in the Infobiotics project repository (www.infobiotics.org) as well as in the CRAN archive (Common R Archive Network, http://cran.R-Project.org).

4. Extensibility and related software

The **vrmlgen** package is built entirely based on functions from the R **base** package and the *contour3d*-function from the **misc3d**-package (Feng and Tierney 2005). Thus, the sources can easily be modified and extended. An installed version of **vrmlgen** can be updated to the latest build using the *update.packages*-function from the **utils** package in R.

Several related 3D visualization packages for the R environment have been developed in the past. One of the most well-known 3D-extensions for R is the **RGL**-package (Adler and Murdoch 2009), which enables users to visualize graphics with OpenGL from within an R-session. Although OpenGL enables users to exploit the capabilities of modern graphics hardware, in contrast to **vrmlgen** the **RGL**-based visualizations cannot be integrated and directly viewed in a webpage. While **RGL** provides more flexibility in controlling mouse-events via R-functions, these features are also more difficult to use for a beginner than for example the tool-tip labels in VRML, which can be configured with relatively simple parameter settings in **vrmlgen**. Apart from this, plotted data points in VRML can be interlinked with online databases by assigning hyperlinks to each data point (which is currently not possible with the **RGL**-package).

For high-dimensional data sets various other R-packages are available which provide dimensionality reduction methods and/or 3D visualization facilities, e.g. the packages **misc3d** (Feng and Tierney 2005), **lattice** (Sarkar 2008) and **scatterplot3d** (Ligges, U. and Maechler, M. 2003). Alternatively, **R**-users can choose to rely on interfaces to external software for multivariate data visualization. For example, the **xgobi** package (Maechler, Hornik, and Ripley 2009) provides an interface to the **XGobi** and **XGvis** programs for analysis and visualization of high-dimensional data (Swayne, Cook, and Buja 1998; Buja, Swayne, Littman, Dean, Hofmann, and Chen 2008). Similarly, for **GGobi** (Swayne, Lang, Buja, and Cook 2003), the successor of **XGobi**, an R-package has been developed to link the R environment with the features of **GGobi**.

Apart from these general-purpose tools, there are several other more specialized R-packages for multivariate data visualization. Two notable examples are the **denpro** package, which uses level set trees, tail and shape trees to visualize multivariate data, density functions and level sets, and the **plotrix**-package (Lemon, Bolker, Oom, Klein, Rowlingson, Wickham, Tyagi, Eterradossi, Grothendieck, Toews *et al.* 2009), which provides a great variety of special plotting functions including 3-dimensional methods like 3D pie charts.

While all these packages are very useful for visualization purposes on a single computer, their generated output cannot be made accessible directly in a browser on the web, where a large number of external users might be interested in exploring the data. However, since the above

packages provide many features which are not included in **vrmlgen** and use the more powerful graphics and rendering capabilities of the local hardware, **vrmlgen** is intended to complement, not replace the existing packages, by allowing R-users to make some of the most common types of 3D plots available on the web for interactive data exploration.

5. Installation instructions

vrmlgen requires a recent pre-installed version of the R statistical programming environment, which is freely available for all major operating systems at http://www.r-project.org including both source code and binaries.

On a computer with internet connection and no firewall/proxy-related restrictions, **vrmlgen** can easily be installed and loaded by starting an R session and typing the following two lines at the command prompt:

```
install.packages("vrmlgen")
library("vrmlgen")
```

Alternatively, the package can also be downloaded from the CRAN-archive or at http://www.infobiotics.org/vrmlgen and installed manually, or an internet proxy can be specified in the R terminal using

```
Sys.setenv("http_proxy" = "hostname:port")
```

where "hostname" is the hostname or IP of the proxy server and "port" is the proxy's port number. Help and example code can be obtained with the command "?vrmlgen" in a running R terminal afer having loaded the **vrmlgen** package.

To view created VRML-models in a web-browser, a VRML-plugin needs to be pre- installed. Many plugins for this purpose are available as freeware on the web for different browsers and platforms, however, to exploit all of **vrmlgen**'s features, the installed VRML viewer has to support VRML 2.0 and Javascript. Two examples which fulfill these requirements and have been used by the authors of the paper are the freeware plugin "Cortona 3D viewer 6.0" (ParallelGraphics 2009) and the offline viewer "BS Contact VRML 7.1" (Bitmanagement 2009). A list of alternative software can be found on the webpage of the Web3D Consortium (2005).

6. Summary and conclusions

vrmlgen is a new R software package to generate 3D visualizations for interactive data exploration on the web. It provides functions for the creation of scatter and bar plots, and visualizations of 3D meshes, parametric functions and height maps, as well as access to low-level plotting functions. Calls of different plotting functions can be grouped together to generate complex 3D-scenes, and the user can write new higher-level plotting methods based on the already existing functionality. Additional features include the possibility to assign hyperlinks and meta-labels to the data points and to visualize regions of high data density by means of contour surfaces. All outputs are provided in two common web-formats, the VRML- and LiveGraphics3D-format. Apart from using the obtained 3D models for data presentation on the web, these files can also serve as input for freeware rendering software to generate high-quality perspective plots for scientific publications.

For demonstration purposes, **vrmlgen** has been inter-linked with a PHP- interface to enable users to generate visualizations of uploaded 3D data matrices on the web without the need to install any software (http://www.infobiotics.org/vrml). Moreover, the R-package is employed in practice on a bioinformatics web-server for DNA-chip analysis (Glaab *et al.* 2009; Infobiotics Project 2009). These example applications show that **vrmlgen** can readily be used within web-applications to generate 3D visualizations dynamically in response to user activity on a web site.

7. Acknowledgements

We thank the reviewers for their suggestions, which have helped us to improve the software package and the manuscript. We acknowledge support by the Marie- Curie Early-Stage-Training programme (grant MEST-CT-2004-007597), by the UK Engineering and Physical Sciences Research Council (EP/E017215/1) and the Biotechnology and Biological Sciences Research Council (BB/F01855X/1).

References

Adler D, Murdoch D (2009). "**RGL**: 3D Visualization Device System (OpenGL)." *http://cran.r-project.org/web/packages/rgl/index.html*.

Adler D, Nenadic O, Zucchini W (2003). "**RGL**: A R-library for 3D Visualization with OpenGL." In *Proceedings of the 35th Symposium of the Interface: Computing Science and Statistics, Salt Lake City*, volume 35.

Ames A, Nadeau D, Moreland J (1997). *The VRML 2.0 Sourcebook*. John Wiley & Sons, Inc. New York, NY, USA.

Bitmanagement (2009). "BS Contact VRML 7.1." *http://www.bitmanagement.de/en/products/interactive-3d-clients/bs-contact*.

Bowman A, Azzalini A (2008). "**SM**: Nonparametric Smoothing Methods." *http://cran.r-project.org/web/packages/sm/index.html*.

Buja A, Swayne D, Littman M, Dean N, Hofmann H, Chen L (2008). "Data Visualization with Multidimensional Scaling." *Journal of Computational and Graphical Statistics*, **17**(2), 444–472.

Council of European Social Science Data Archives (CESSDA) (2009). "CESSDA Data Portal." *http://www.cessda.org/accessing*.

Department of Physics and Astronomy, University of Kentucky (2009). "Atomic Data for Astrophysics." *http://www.pa.uky.edu/~verner/atom.html*.

Diggle P (1990). *Time series: a biostatistical introduction*. Oxford University Press.

Edgar R, Domrachev M, Lash A (2002). "Gene Expression Omnibus: NCBI Gene Expression and Hybridization Array Data Repository." *Nucleic acids research*, **30**(1), 207.

Feng D, Tierney L (2005). "**misc3d**: Miscellaneous 3D Plots." http://cran.r-project.org/web/packages/misc3d/index.html.

Genovese A (2009). "3D City Block Buildings." http://artist-3d.com/free_3d_models/dnm/model_disp.php?uid=2030.

Glaab E, Garibaldi J, Krasnogor N (2009). "ArrayMining: A Modular Web-Application for Microarray Analysis Combining Ensemble and Consensus Methods with Cross-Study Normalization." *BMC Bioinformatics*, **10**(1), 358.

Ihaka R, Gentleman R (1996). "R: A Language for Data Analysis and Graphics." *Journal of Computational and Graphical Statistics*, **5**(3), 299–314.

Infobiotics Project (2009). "Infobiotics Bioinformatics Servers." http://www.infobiotic.org.

Kraus M (2003). "LiveGraphics3D Documentation." http://wwwvis.informatik.uni-stuttgart.de/~kraus/LiveGraphics3D/index.html.

Lemon J, Bolker B, Oom S, Klein E, Rowlingson B, Wickham H, Tyagi A, Eterradossi O, Grothendieck G, Toews M, *et al.* (2009). "**Plotrix**: Various Plotting Functions." http://cran.r-project.org/web/packages/plotrix/index.html.

Ligges, U and Maechler, M (2003). "**Scatterplot3d**." *Journal of Statistical Software*, **8**(11), 1–20.

Maechler M, Hornik K, Ripley B (2009). "**Xgobi**: Interface to the XGobi and XGvis Programs for Graphical Data Analysis." http://cran.r-project.org/web/packages/xgobi/index.html.

Murray J, Van Ryper W (2005). "Wavefront OBJ File Format Summary." http://www.fileformat.info/format/wavefrontobj/egff.htm.

NASDAQ OMX (2009). "NASDAQ Data Store." https://data.nasdaq.com.

Oak Ridge National Laboratory, Physics Division (2009). "Controlled Fusion Atomic Data Center (CFADC)." http://www-cfadc.phy.ornl.gov/.

Ohio State University, Department of Finance (2009). "Financial Data Finder." http://fisher.osu.edu/fin/osudata.htm.

ParallelGraphics (2009). "Cortona 3D Viewer 6.0." http://www.parallelgraphics.com/products/cortona3d.

Pettijohn J (1996). "A Guide to Locating Financial Information on the Internet." *Financial Practice and Education*, **6**(2), 102–10.

Plachetka T (1998). "POV Ray: Persistence of Vision Parallel Raytracer." In L Szirmay-Kalos (ed.), *Proceedings of Spring Conference on Computer Graphics, Budmerice, Slovakia*, pp. 123–129.

R Development Core Team (2009a). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org.

R Development Core Team (2009b). "Writing R Extensions." *http://cran.r-project.org/doc/manuals/R-exts.html*.

Sarkar D (2008). **Lattice**: *Multivariate Data Visualization with R*. Springer Verlag, New York.

Sintzel P (2010). "European topographic maps." *http://maps.simutrans.com/europe.html*.

Spiteri I (2008). "Transcription profiling of human primary breast cancer panel (Nottingham)." *http://www.ebi.ac.uk/microarray-as/ae/browse.html?keywords=E-TABM-576*.

Swayne D, Cook D, Buja A (1998). "**XGobi**: Interactive Dynamic Data Visualization in the X Window System." *Journal of Computational and Graphical Statistics*, **7**(1), 113–130.

Swayne D, Lang D, Buja A, Cook D (2003). "**GGobi**: Evolving from XGobi into an extensible framework for interactive data visualization." *Computational Statistics and Data Analysis*, **43**(4), 423–444.

Thiessen P (2004). "VRML2POV Parser." *http://www.chemicalgraphics.com/paul/vrml2pov/index.html*.

UK Data Archive (UKDA), University of Essex (2009). "Economic and Social Science Data Service." *http://www.esds.ac.uk*.

Web3D Consortium (2005). "VRML Viewers, Browsers & Plug-ins." *http://www.web3d.org/x3d/vrml/tools/viewers_and_browsers*.

Widera P, Garibaldi J, Krasnogor N (2009). "GP Challenge: Evolving Energy Function for Protein Structure Prediction." *Genetic Programming and Evolvable Machines (to appear)*.

World Health Organization (2009). "Global Health Atlas - Data Query." *http://apps.who.int/globalatlas/DataQuery/default.asp*.

Zucker M, Kishore A, Sukumar R, Dragoset R (2009). "NIST Physics Laboratory - Elemental Data Index." *http://physics.nist.gov/PhysRefData/Elements/cover.html*.

Affiliation:

Enrico Glaab
School of Computer Science
University of Nottingham
NG8 1BB
E-mail: e.glaab@cs.nott.ac.uk
URL: http://www.infobiotics.org/vrmlgen/

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume VV, Issue II *Submitted:* yyyy-mm-dd
MMMMMM YYYY *Accepted:* yyyy-mm-dd