# Integration of R Statistical Environment into ICT Infrastructure of GMP and GENASIS

Richard Hůlek, Jiří Kalina, Ladislav Dušek, Jiří Jarkovský



# Integration of R statistical environment into ICT infrastructure of GMP and GENASIS

Richard Hůlek, Jiří Kalina, Ladislav Dušek, and Jiří Jarkovský

Institue of Biostatistics and Analyses,
Kamenice 126/3, 625 00 Brno, Czech Republic
hulek@iba.muni.cz,kalina@mail.muni.cz,
dusek@iba.muni.cz,jarkovsky@iba.muni.cz
http://www.iba.muni.cz

**Abstract.** The Global Monitoring Plan (GMP) was established as a tool for providing a worldwide overview of Stockholm Convention (SC) compliance by monitoring and evaluation of SC 22 persistent organic pollutants (POPs) concentration levels and their trends. In order to evaluate a dataset on POPs concentrations from the initial GMP campaign, it was essential to use advanced statistical methods which are not incorporated in commonly used database languages. Instead of a complete realization of these methods in the main programming language, in which the application is developed, this language was used only as an interface to the server version of the powerful statistical software R. The involvement of the R language into the environmental pollution data assessment infrastructure of the Global Environmental Assessment Information System (GENASIS) adopted for the GMP data makes easier to avoid disambiguities in data analysis and brings a powerful tool for advanced statistical analysis and visualization of GMP and GENASIS data.

**Keywords:** R, statistical computing, web application, system architecture, Stockholm Convention, POPs, GMP, GENASIS.

## 1 Background

The Stockholm Convention (SC) was signed by 128 parties in 2001 in order to protect both the environment and human health from persistent organic pollutants (POPs). It has been in force since 2004, changing an approach to POPs all around the world by the elimination or a significant restriction of their manufacturing and application. Most of the production and use of listed compounds was abandoned, but a huge amount of POPs, although still slowly decreasing in general, is still present in the environment. Thus a detailed and reliable analysis of the still developing situation is needed for the evaluation of SC effectiveness.

An arrangement of the Global Monitoring Plan (GMP) was established as a tool for providing a worldwide overview of the Stockholm Convention (SC) compliance by monitoring and evaluation of SC 22 persistent organic pollutants (POPs) concentration levels and their trends.

The results of GMP reporting are assessed and visualised via an international part of the Global Environmental Assessment Information System (GENASIS) which encompasses a wider set of projects on environmental contamination by chemicals, namely POPs. Run in the cooperation between the Research Centre for Toxic Compounds in the Environment (RECETOX) and Institute of Biostatistics and Analyses (IBA) at Masaryk University, the GENASIS system combines expertise, validated data from partner institutions, inputs from regular environmental monitoring programs and provides data repository, analytical tools and data management crowned by online web visualization platform [1].

A prime related data mining and a further treatment on the dataset of POPs concentrations from the GMP and other projects needs to employ several advanced statistical methods, which are not incorporated in any of the commonly used database languages neither are standard distributions of main languages. Thus, several different possibilities come under consideration on how to treat the task of a sufficiently robust and effective development and running of statistical computations.

There are several options to use instant solutions on database level (i.e. Oracle DBMS_STAT_FUNC package [11], Oracle Data Mining (ODM) extension [12], PostgreSQL MADlib library [13] and a few others) which enable several data mining functionalities as native SQL functions.

Another possibility is to involve some of the wide range of main programming language statistical libraries such as PHP Statistics extension [14] or Commons Math for Java [15], but usually they are less efficient than SQL language extensions.

However, both of these solutions have certain shortcomings given by either a limited scope (in particular, a small number of incorporated statistical techniques and options of their customization) or a high price. Nevertheless, even the most expensive ready-made packages are principally limited by the set of functions that were included by their creators and the development of a new function is thus impossible or very difficult.

Similar imperfections occur when trying to create one's own main programming language statistical library. The development, optimization and validation of algorithms and their perpetual modifications and improvements represent a rather complicated way to achieve satisfactory results of environmental data mining, especially in the changing environment of different programming languages and different requirements of infrastructure projects.

Finally, all of the above-mentioned solutions are part of an area that is easily accessible for experts from the IT field but generally unintelligible for (bio)statisticians, who are primarily involved in environmental data mining and evaluation.

These conclusions led us to a decision to prefer the use of a main programming language (PHP) only as an interface to the server version of a powerful statistical software R, especially in the scope of further statistical analyses carried out at the Institute of Biostatistics and Analyses in the framework of the GENASIS project. The involvement of the R language into the environmental pollution data

assessment infrastructure makes the development considerably more efficient not only by engaging a wider group of experts into the process of data analysis and makes available a powerful tool, but also facilitates to avoid ambiguities in data, allowing an advanced visualization and analysis of environmental data.

### 1.1 R

R is the name of a computational environment based on the S language developed in 1970's in Bell Laboratories. It comprises an extensive pool of statistical techniques, which can be easily extended via a wide range of packages available online through the Comprehensive R Archive Network (CRAN). Due to the fact, that R is available as a Free Software under the terms of the Free Software Foundation's GNU General Public Licence in source code form [2], it achieved enormous popularity in recent years and the number of its users is estimated to exceed 2,000,000 worldwide.

R compiles and runs on a wide variety of UNIX platforms, Windows, MacOS and related systems. The environment itself is built in the R language, wherefore any extensions of existing and development of new functions can be conducted easily recurrently from basic elements of the language up to advanced statistical techniques.

Another advantage of R lies not only in the fact, that it is widely used in the statistical community, but due to its open-source character, it allows to share an enormous amount of experience, methods and ideas suitable for statistical computations. In the field of environmental sciences, a lot of different projects were solved using the R language and numerous methods were published in a form of R packages [15].

On the other hand, the R environment has one substantial disadvantage: from the very beginning, it was constructed as a single-user system intended primarily for a desktop use; therefore its use for solving multi-user tasks is rather complicated and has certain limitations, as well as a partial incompatibility of R environment data structures and formats used during data transmission between the clients and server.

Although the use of R as a tool providing advanced statistical techniques solves all problems listed above and allows the use of an almost unlimited set of algorithms and functions for calculations, server implementation remains quite challenging with regrds to its desktop character. Therefore, in terms of efficiency, R should be implemented into computational infrastructure in its widest meaning in the frame of projects solved at IBA. Thus, its use should not be confined to the processing of data from GMP, but it should be used in all applications under the information system GENASIS in the widest possible context.

## 2 Integration of R into ICT infrastructure

The integration of R into web applications is not a trivial task. A rather complex structure of the R language suggests for such form of integration that will be

both most usable across applications and flexible enough to deal with various requirements on computing tasks. It should also meet many other criteria that are primarily implied by the general characteristics of web applications and their development process.

### 2.1 Requirements

Web applications operate on a client – server communication architecture. All the application logic is located at the server. Individual users send requests to a central server through their web browsers (clients). Requests are being interpreted and evaluated. The output is processed and sent back to the client computer in the form of response (so called the request – response communication). It is obvious that the R software integration needs to be at the server side due to the fact that today's trend is to create a so called thin client that does not perform any computational operations and displays only server outputs. On the other hand, the thick clients contain a substantial portion of application logic. This kind of architecture (communication model client-server, where client means thin client) supports a wide range of client devices from conventional desktop computers or laptops to recently highly popular mobile phones and tablets where R support would not be possible.

Another important characteristic, which determines a certain set of requirements for the integration method, is the fact that web applications are typically accessed simultaneously by multiple users. Multi-user environment is the main difference from the other applications, where R is run locally on a desktop computer or laptop. Due to the existence of multiple users, the server integration of R has to support concurrent processing in multiple processes so that required system response time would be met. Concurrent processes have to be isolated among each other and not to share for example program variables or other resources.

Using R on local computers program scripts, helper libraries and R packages are written in a way where the output of particular computing functions prints result to a standard or error output (print on a screen), saves a file to a local hard drive or returns a value with pointer to the system memory. Such a different approach to returning result values is not fully applicable in the server environment because, e.g., sharing data among servers on hard drives is not always possible.

In general the particular form of R software integration into the server environment should support maximum flexibility and effective use in different web applications. For that reason it is efficient to integrate it as a component in commonly used ICT infrastructure and a set of helper communication code libraries rather than building web applications with one-off R support solutions.

### 2.2 Methods of integration

R can be effectively embedded with other programming languages and environments. Implementations of interfaces to R exist for most of the main programming languages such as Java [3], C++ or Python [4]. The interface for

communication with C and Fortran is already implemented in R. A module for Apache2 web server RApache (mod_r) [5] is the important technology for the development of web applications, which enables the R code to be run directly from the web server environment.

There are various approaches to R software integration into web applications. They differ mainly in the general concept.

First, an extension for R support in a main programming language, in which the application is developed, can be used. The main programming language is used to create the application logic, access to data sources, data transformation and the definition of application design. The interface then allows for calling the computation and statistical functions directly from the given programming language. Such solution is simple and relatively easy to manage. There are, however, limited possibilities of scaling and moreover, when further web applications are developed using different programming languages it is necessary to implement the whole R support again.

The integration of the R support on a more general level is much more flexible, so that the access to computation functions would not be limited by using one or another programming language or platform – i.e. the choice of integration in a form of web services. Web service in general play an important role in building service-oriented systems and infrastructures. The communication between individual system components runs by means of the universal and widely supported communication protocol (e.g., Hypertext Transfer Protocol – HTTP), which is also used for sending requests between the components for processing individual tasks. A concrete internal design (implementation) of the components remains fully hidden behind a strictly defined programme interface (API), through which the components communicate to each other. Making the computation services of R language accessible in a form of web services, which can be called via web interface by a system of unique addresses (URL), will enable a flexible development of web applications, which might use computation possibilities easily and independently on the used programming language, platform or form of internal implementation. The separation of computation and application server also facilitates system scaling, security, or, e.g., caching. This is because the computation backend is integrated within the infrastructure at one place only and system resources can be assigned effectively, so that the sufficient system response would be ensured and the resources would be effectively used.

During the integration of R to the GENASIS system infrastructure we preferred the inclusion of a computation server as a separated component communication on a SOA principle [6].

### 2.3 Computation backend architecture

Computation backend server is accessible through a central proxy which enables load balancing and – if implemented – caching functionality as well. Thanks to using the load balancing proxy it is easy to scale the computation backend just by adding additional resources (servers).

Proxy, cache and computation servers are situated in a protected infrastructure environment behind a firewall which protects them from threats from the Internet. Services of computation backend are accessible via client applications or via direct calls of particular R functions and scripts mapped to URL addresses.

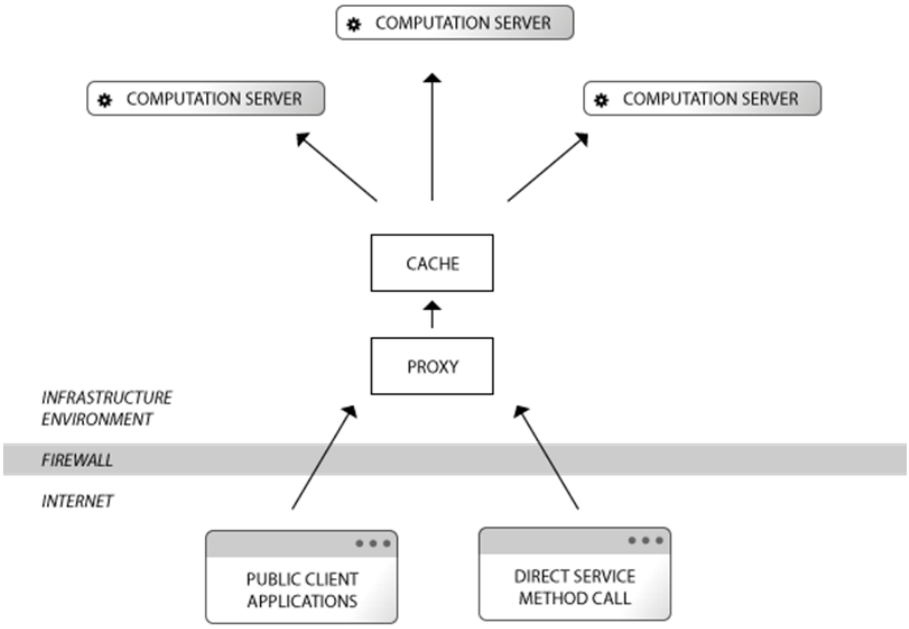


**Fig. 1.** Computation backend architecture

### 2.4 Technologies used

Due to a very quick progress in recent years there are technologies available which enables integration of R into server environment.

RApache is a module extension of popular Apache2 web server enables executing R scripts on the server. It supports multiuser access by running isolated processes for the each request. Since Apache2 is HTTP server, RApache communicates with its clients over HTTP as well. RApache works very well together with R package Brew [7] which enables mixing R and HTML code together in one document. Such approach speeds up the development rapidly.

RApache and Brew are sufficient technologies to create web applications directly just by writing R and HTML code. But for more complicated applications

it is more reliable to build web application in the main language such as PHP or Java and to use R functions published via RApache as remote method calls.

Following this approach, one can very easily realize there are many repeating steps: mapping functions to particular URL addresses, reading parameters from request, transforming results into the requested format before sending a response back to the client and many more.

To implement computation backend a much smarter solution can be used: OpenCPU [8]. It is built on the top of the RApache. It offers REST (Representational State Transfer) API with an automated mapping of URL addresses onto available R functions and custom scripts. OpenCPU transforms the output into various formats such as the plain text, JSON or as a downloadable picture.

Client applications build on the top of the GENASIS ICT infrastructure are usually developed in PHP, HTML and JavaScript or Apache Flex framework.

### 2.5 Details of client-server communication

The communication between web applications and the computational server is realized via the HTTP protocol. Request parameters are passed into R scripts using GET and POST methods. To create a request on the client side of the application, which desires to call R-based computation backend, it is necessary to create HTTP request programmatically, which means namely: targeting the request to a proper URL address; serializing all data structures (simple variables, data arrays or complex objects as well as binary files) into a transferable format and making them ready to be sent as parameters.

When a request is sent to computation, the backend application must wait until the result is calculated and sent back in the server response. Results must be unserialized, data structures must be converted into a proper format of the target language and passed into program variables.

In order to resolve all of the above mentioned tasks, a custom code library which makes it easier to communicate PHP web applications and R-based computational backends has been developed. The library helps to convert all data and data structures into JSON format, resolves minor (but important) differences of implementation of JSON in PHP and R and helps to create proper HTTP request using the curl library available in PHP language. All binary data are encoded by base64 [9] algorithm in order to be easily transferrable over the text-based HTTP protocol. Since the JSON implementation on the R side (JSONIO [10]) is a significant consumer of system resources, the helper library in addition offers an optimized way to transfer large amounts of data to the computation backend.

## 3 Results

### 3.1 GENASIS ICT infrastructure

The R-based computation server is an important part of the GENASIS system architecture. It provides R capabilities in the form of web services accessible via web interface to any application developed in this infrastructure.

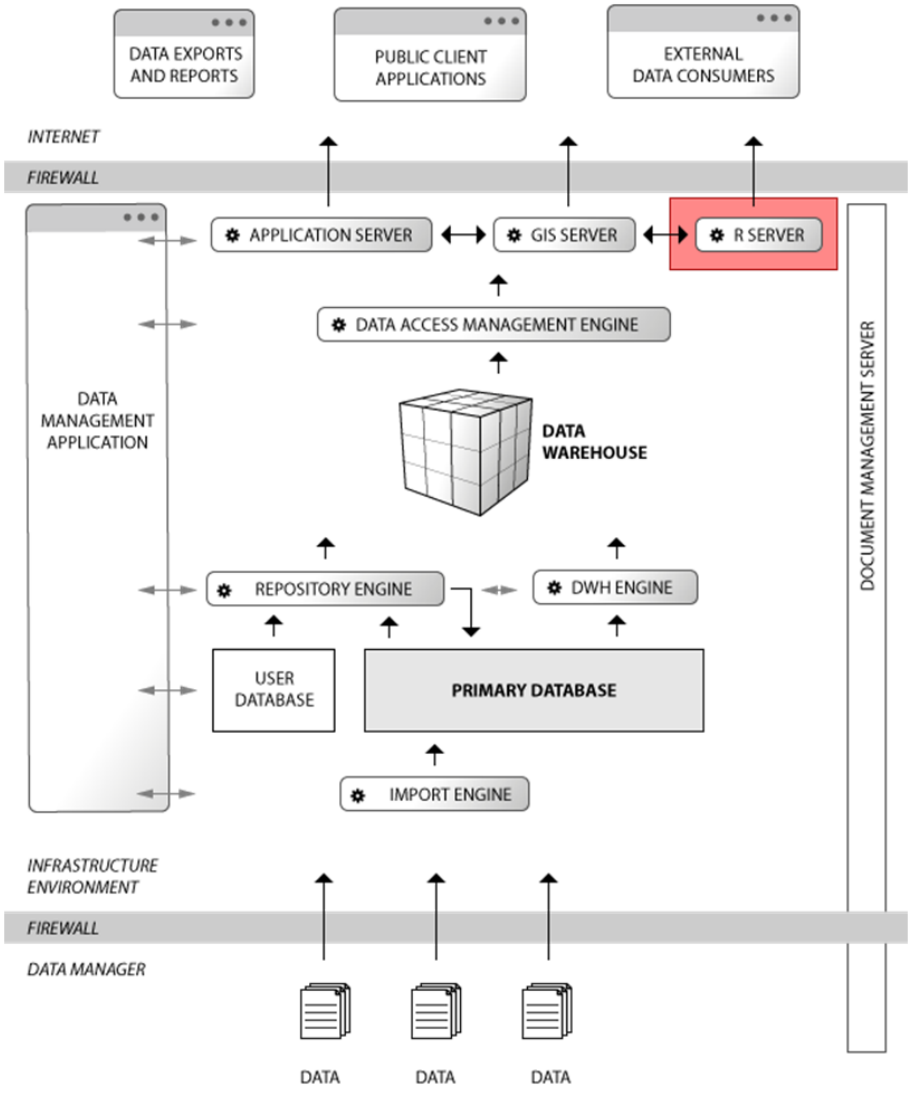


**Fig. 2.** GENASIS ICT infrastructure

### 3.2 Using of R-based computation services

Environmental issues, including the tasks of air pollution, are in the centre of R users attention from the very first stages of its development. Over time, numerous packages containing functions relevant for the use in the evaluation of environmental data were published. GENASIS system procedures and, in particular, the evaluation of data from the GMP is largely implemented using functions included in the packages supplied with the standard R distribution (base, compiler, datasets, graphics, grDevices, grid, methods, parallel, splines, stats, stats4, tcltk), so descriptive statistics, correlation and linear trends etc. do not require additional functions. Nevertheless, the situation in time series analysis and outlier exclusion is more complicated and requires the use of some extending packages.

Several R packages relevant to GMP and GENASIS statistical tasks are listed in Table 1.

Table 1: R packages relevant to GMP and GENASIS computations

| Name | Content | Year published |
|---|---|---|
| *forecast* | Methods and tools for displaying and analysing univariate time series forecasts including exponential smoothing via state space models and automatic ARIMA modelling. | 2009 |
| *Kendall* | A modest package for Kendall rank correlation and Mann-Kendall trend test computations, both used in GMP trend analysis. | 2005 |
| *dyn* | Another package for time series regression. The regression functions to be used with time series including specifications that may contain lags, diffs and missing values. | 2005 |
| *openair* | Tools to analyse, interpret and understand air pollution data. Data are typically hourly time series and both monitoring data and dispersion model output can be analysed. Many functions can also be applied to other data, including meteorological and traffic data. | 2010 |
| *tseries* | Package originally created for time series analysis and computational finance, but useful also in environmental applications. | 1999 |
| *mvtnorm* | Package allowing to compute multivariate normal and t probabilities, quantiles, random deviates and densities used in linear trend confidence intervals. | 2000 |
| *car* | Package accompanying a book of J. Fox and S. Weisberg, An R Companion to Applied Regression contains i.a. the Cooks distance used in outlier exclusion. | 2001 |

For the purpose of a reliable estimation of the mean overall POPs concentrations and related statistical measures, it was necessary to apply a six-step validation procedure on the annually aggregated database of GMP I records (which contains either mean, median, minimum, maximum or any combination of these

statistics). This procedure was developed in order to maintain a high predictive value of the GMP records and to avoid any bias in estimates of the background concentration values. It consists of the elimination of useless data records (inappropriately aggregated data, incomplete values, outliers) and employs several R base functions (both parametric and non-parametric) such as min, max, median, quantiles, geometric and arithmetic mean, standard deviation and several data transformations) as well as advanced methods of outlier exclusion and description as $\chi^2$ testing or Cook's distance involved in R package *car*.

Further steps in GMP and GENASIS data analysis could be divided into 5 sections employing differnet kinds of R functions:

**Baseline concentration** - a set of descriptive statistics used to an elementary description of all the data available for each parameter (compound or set of compounds). The qualitative data include the number of entries and the time span of their recording (measurement), quantitative non-parametric statistics embraces the median and the pair of percentiles (5th and 95th). In terms of parametric description, the geometric mean seems to be the best description of the central tendency, which corresponds approximately to a lognormal distribution of the concentration of pollutants in the air. The confidence interval of the geometric mean was also computed, using log-normal distribution of values and concept of geometric standard deviation. As additional statistics, the arithmetic mean and standard deviation were also determined.

**Detectable difference** was expressed by two different concepts. The first alternative was the minimal annual concentration decline (increase) statistically significantly different from 0 based on standard t-test procedure on mean annual differences on all reasonable sites (i.e. on sites, where at least two concentration values in different times were available, the mean difference was computed as the total difference between initial and final value divided by the length of the time series. Such a value was repeated according to this length.).

The second alternative of detectable difference considers the minimal statistically significant slope of concentration line using a computation of a confidence interval for linear regression (or exponential regression in the case of logarithmic transformation). This computations requires the use of multivariate normal distribution parameters included in the package *mvtnorm*, quantile functions of normal and log-normal distributions and linear regression modelling tool, which are really complicated to implement properly without R.

**Identification of time trend** - several different methods are being used for the identification of statistically significant time trend, from which non-parametric Mann-Kendall test was chosen for GMP due to its robustness and low demands on the length of time series and its data distribution.

The test compares the relative magnitudes of concentration rather than the data values themselves for each site individually. One benefit of this test is

that the data do not need to conform to any particular distribution. Moreover, data reported as not-detected can be included by assigning them a common value which is smaller than the smallest measured value in the data set. The procedure assumes that only one data value exists per time period, which is ensured by annual aggregations on sites. The final statistics of Mann-Kendall test S is normalized and compared with the critical value of normal distribution.

There is also the possibility of employing other methods of trend identification such as the non-parametric Spearman's rank correlation trend test, parametric Pearson's correlation trend test and again the linear and exponential models. Moreover, short-term predictions of time series are in progress, employing several advanced time series modelling methods (exponential smoothing and ARIMA models). For this section, not only the *Kendall* package, but also the time series modelling packages were used as well as correlation functions included in the base set of R functions.

**Trend quantification** - in the case of trend quantification, two different approaches are implemented, using parametric and non-parametric statistical techniques. The least square method based on linear and exponential regression models is employed for the parametric estimation of mean annual decrease/increase of POPs concentrations in the first case, whereas the non-parametric approach uses the Theil-Sen estimator. This robust linear regression method provides the value of trend slope as a median value of slopes of all pairs of time points inside the time series. The computed slope value corresponds well to the Mann-Kendall trend test, since the Kendall correlation coefficient between the measurements and modeled values converges to zero. The intercept of the Theil-Sen model is computed likewise as a median of differences between (without any intercept) modeled and measured values.

After a cancellation of *mblm* package from the R CRAN repository, the best option for employing the Theil-Sen estimator is to use the more complex and embracing package *openair*.

**Visualization techniques** - the current visualization techniques within the GENASIS and GMP web platforms comprise interactive histograms of concentrations, time series plots, trends visualisations including confidence intervals, detectable alternative dependence plots (power analysis), outliers exclusion plots, autocorrelation plots and several others. With the presence of plot drawing libraries as graphics or several more specific such as lattice, there are many possibilities to draw the whole spectrum of these plots within the R environment. Nevertheless, data transfer of the results in the form of jpeg/png files and restricted possibilities of plots graphic design resulted in the variant, that R is used only for precomputation of the plots, while the plots themselves are generated inside the main programming language by an appropriate graphic library.

## 4 Discussion

Although the implementation of the server version R environment is far from being a trivial task, after its completion it became the essential part of the environmental pollution data infrastructure, which provides a practically unlimited space for statistical computations and visualization of results.

There are several effects induced by the possibility to run statistical computations inside the web applications in the R environment:

- The R environment offers a huge pool of ready-made packages for almost all applications in the field of environmental statistics.
- The open source character and recurrent form of the R environment allows an easy creation of new functions and packages for sharing experiences both inside and outside the institution.
- The separation of computing and application layers of infrastructure allows a larger number of experts to participate in the development of statistical algorithms, without having to completely control the main programming language environment - especially (bio)statisticians without a deeper in-sight into IT technologies.
- Using the proven functions and procedures makes it easier to avoid ambiguities in the data analysis and saves a substantial part of the development capacity used for validation and correction processes.
- In the case of large volumes of data processing, the stability of the system is at a significantly higher level due to the use of optimized algorithms in the R environment.

Undoubtedly the high increase of development efficiency, worth taking the difficulties associated with the implementation of R into the data infrastructure.

**Acknowledgment.** This work was supported by the TACR project No. TB010-MZP058 "Development of the system for spatial evaluation of the environmental contamination".

## References

1. GENASIS environmental data repository, `http://www.genasis.cz/index-en.php`
2. R Core Team. R: A language and environment for statistical computing, `http://www.r-project.org/`
3. rJava: Low-level R to Java interface. R package version 0.9-4 `http://cran.r-project.org/web/packages/rJava/index.html`
4. rpy, a robust Python interface to the R Programming Language `http://rpy.sourceforge.net/rpy.html`
5. Horner J. Embedding R within the Apache Web Server: What's the Use? `http://biostat.mc.vanderbilt.edu/wiki/pub/Main/RApacheProject/paper.pdf`
6. Arsanjani, A., Zhang, L.J., Ellis, M., Allam, A., Channabasavaiah, K.: A service-oriented reference architecture. IT Professional 9(3), 10–17 (2007)

7. Framework for Report Generation. R package version 1.0-6 http://cran.r-project.org/package=brew
8. OpenCPU https://public.opencpu.org/pages/
9. Tools for base64 encoding. R package version 0.1-1. http://cran.r-project.org/package=base64enc
10. Serialize R objects to JSON, JavaScript Object Notation. R package version 1.0-3. http://cran.r-project.org/package=RJSONIO
11. Oracle Database PL/SQL Packages and Types Reference 102 DBMS_STAT_FUNCS, http://docs.oracle.com/cd/B19306_01/appdev.102/b14258/d_stat_f.htm
12. Oracle Data Mining http://www.oracle.com/technetwork/database/options/advanced-analytics/odm/index.html
13. MADlib http://madlib.net
14. PHP Statistics http://www.php.net/manual/en/intro.stats.php
15. CRAN Task View: Analysis of Ecological and Environmental Data http://cran.r-project.org/web/views/Environmetrics.html