B.I.R.O.

Best Information through Regional Outcomes

A Public Health Project funded by the European Commission, DG-SANCO 2005

WP 8

Final Report

Statistical Engine

April 2009

A technical report
produced by Serectrix s.n.c., Pescara,
as subcontractor of the Coordinating Centre, The BIRO Project
Department of Internal Medicine, University of Perugia, Perugia

Statistical software referenced in this report
available as integral part of Workpackage 8
at the restricted area of the Project Website

Authors

Fabrizio Carinci
Senior Biostatistician,
Serectrix s.n.c.
e-mail: research@fabcarinci.net

Luca Rossi,
Statistician, Consultant
e-mail: redsluke@gmail.com

Citation

Carinci F, Rossi L, “Statistical Engine”,
Workpackage 8, The BIRO Project
Pescara, 2nd April 2009

Address for correspondence

Fabrizio Carinci
Via Gran Sasso 79
65121 Pescara – ITALY
Ph/Fax. +39 085 4429188
Email: research@fabcarinci.net

Project Website

http://www.biro-project.eu

CONTENTS

ABSTRACT

Introduction

The “Health Information Strand” funded by the European Commission under the Public Health Programme aims at creating sustainable solutions for the routine provision of strategic data across Europe. The general objective of BIRO is to build a common infrastructure for standardized information exchange in diabetes care, to monitor, update and disseminate evidence on a regular basis. The proposed “Shared Evidence-based Diabetes Information System” (SEDIS) is based on the application of two consecutive data processing steps, locally and centrally, each one involving key statistical procedures. A “Statistical Engine” is specifically required to derive aggregate tables from databases held at the regional level, to be sent towards a central BIRO server.

Objectives

To run the same specialised statistical software in each partner region, directly tapping into a standardized PostgreSQL database extracted from local data, formatted according to common definitions specified in a BIRO concept/data dictionary.

To implement and disseminate use of advanced statistical methods to collect and analyse population-based data stored in diabetes registries through a fully documented repository of open source statistical software that will allow users to replicate and further extend the approach.

Materials and Methods

R software has been adopted as a development platform for the statistical engine, launched directly by script command files or with the aid of a GUI interface. The engine connects to the local database using R Postgres drivers. The concept of “statistical object” has been introduced as “an element of a distributed information system that carries essential data in the form of embedded, partial aggregate components, required to compute a summary measure or relevant parameter for the whole population from multiple sites”. Objects are created as tables including statistical aggregations of local data (e.g. the arithmetic mean, percentile, variance, etc.), stored as flat text comma delimited files. A taxonomy has been specified to provide details of all objects being implemented. Specifications provided by the report template have been used to process data and deliver objects as small datasets. Graphical functions and Latex are used to produce individual centre outputs and full local reports in the form of .html files and .pdf documents. A compressed folder is created to deliver all statistical objects produced by local runs of the statistical engine, stored in a directory named with datetime/centre id, transmitted to the central server.

Results

The statistical engine has been successfully developed and tested on both Vista and Linux. Average hardware allowed completing a full BIRO report from a test sample of more than 2,000 patients and several thousands episodes in less than 8 minutes. Installation of the software is identical regardless of the hardware, and requires R>1.8, Latex, Java 6.0 and PostgreSQL plus various additional libraries/packages that are included in its distribution. All R functions are released under the GPL license.

Discussion

The statistical engine provides a platform for accurate benchmarking that currently does not exist at the point of health care provision. It may serve multiple users, from the European Union, to provide updated benchmarking of key indicators on a routine basis, and the local physician, to monitor the status of patients in a modern standardized procedure. The system may improve, through a shared infrastructure, the validity and completeness of information available. Existing registers may be optimised on the basis of common standards, and new ones can be created with a fostered structure. Advantages proposed by the system should be part of a progressive approach through which statistical functions are constantly improved. Users, once inducted to using the software, can apply it independently and submit better aggregate data to the central server, at the same time safeguarding privacy at the highest level of protection, as a result of the application of rigorous rules set by the BIRO privacy impact assessment (WP5).

Conclusions

The application of the statistical engine in regional and individual clinical units can help evaluating clinical practice more rapidly and efficiently. Prevention strategies and health services may be planned more carefully on the basis of factual information, making clinicians more accountable through the availability of up-to-date, well structured information. The development of the statistical engine provides the basis for an expandable open product that through its availability at no charge can crucially help disseminating the BIRO approach across Europe.

1. INTRODUCTION

"*Best Information Through Regional Outcomes*" (BIRO) is a three years public health project started in 2005, funded under the EC Public Health Programme 2003-2008. The project, coordinated by the University of Perugia (Italy) includes as partners: Un.Dundee (Scotland), Joanneum Research (Austria), Un.Bergen (Norway), Paulescu Institute (Romania), Un. Malta (Malta), Cyprus Ministry of Health (Cyprus), also supported by Serectrix (Italy), NOKLUS (Norway) and Telemedica (Romania).

The general objective of BIRO is to build a common European infrastructure for standardized information exchange in diabetes care, to monitor, update and disseminate evidence on the application and clinical effectiveness of best practice guidelines on a regular basis.

The project involves the identification of target parameters and indicators; definition of a common dataset and a data dictionary with a schema for representation; development of a report template, database and statistical engines to deploy results in printed and web format; validation of a secure protocol for international communication and shared data analysis; construction of a web portal to test the dissemination of European estimates on a routine basis.

The technology associated to the construction of the system is centred on the definition of the "Shared Evidence-based Diabetes Information System" (SEDIS), whose general architecture is based on the application of two consecutive data processing steps.

At the basic level, a general version of the system runs in each single register[1] ("*local SEDIS"*) to produce initial estimates that are valid for the local population. All partners in the network, using the same standardized procedures, repeat the process at their best convenience. All regional estimates are sent towards a central server that compiles all "partial" results into a global report that is valid for the European level[2].

The functionality of the basic level of the system is ensured by three fundamental elements.

The first is the *concept and data dictionary* (CDD), storing all common definitions adopted to collect and exchange data across the network. The CDD represents the evidence-based component in the model chain.

The second, the *report* template, is located at the opposite end of the chain, and it determines the selection of data procedures and statistical methods required to estimate all results for the health report.

The third component is represented by core engines operated by local administrators on the local databases. The overall model (*global SEDIS)* directly follows from the local implementation: once statistical objects are available from each register, they are sent to the server using a secure transmission via specialised *communication software*.

The level of aggregation chosen for each object is a trade-off among formal agreement, legislation, ethical values and practical limits, all aspects that are properly investigated in the framework of the BIRO project. The general design has been progressively implemented through the definition of candidate architectures submitted to a formal evaluation process coordinated by the *Privacy Impact Assessment* (PIA).

Modern technology is used to disseminate statistical results through a dedicated *Web Portal*. The whole process is followed up by a team of representatives from EU New Member States, analysing barriers and obstacles to *technology transfer* at each step*. Dissemination* includes the preparation of a monograph. An *evaluation* is included by design through the involvement of independent reviewers at each phase of the project.

The present report documents the activity related to BIRO Workpackage 8 (WP8), "Statistical Engine", whose main responsibility is to deliver statistical "*objects*" (*tables, parameters, graphs*) to be derived by databases held at the level of the local centre, e.g. a clinical site or regional register. In BIRO, partial results contribute to an overarching framework collecting aggregate data from all regions. As all statistical results are amalgamated by a central server, "twin" statistical routines to match those used by the statistical engine have been developed to produce pooled estimates of diabetes indicators. These components are covered by the WP10 report, Central Engine.

2. OBJECTIVES

The Workpackage must provide the BIRO Consortium with a set of powerful statistical tools that will transform data gathered in different regions into usable information and customized reports.

The main product is planned to constitute a central part of the Intranet Statistical System (ISS)[3] that will operate statistical routines in connection with tools deploying results on the web portal.

Primary objective of the BIRO "*Statistical Engine*" is to run specialised, standardized software in each participating region on top of a well formatted PostgreSQL database, based upon common definitions included in the BIRO concept/data dictionary. Outputs will be standardized, based upon specifications provided by a Report Template whose scope is to present information according to an agreed standard that will be the same for the local register and the European collaboration. The exchange of aggregate data is at the base of this process.

Secondary objectives include:

- implementation and dissemination of modern database techniques and advanced statistical methods to collect and analyse population-based data stored in diabetes registries. Statistical models must include mainstream methods for the standardization of the diabetic population, including multivariate models, e.g. Logistic Regression, GEEs, Multilevel models[4], taking into account different sources of variation. Whenever appropriate, a meta-analytical approach[5] is used to bypass data transfer of excessive micro data across countries.
- creation of a fully documented repository of open source statistical software[6] that will allow user to replicate and further extend the application of specialised software.

3. MATERIALS AND METHODS

3.1 TECHNOLOGICAL ARCHITECTURE OF THE BIRO SYSTEM

The selected BIRO Architecture is defined by three consecutive steps, logically organized in two different parts: *local* and *global* (**Fig. 1**)[7].

The *local* part of BIRO Architecture includes the set of software tools required by each collaborating centre to undertake two basic operations:

1) to produce a standardized BIRO local report
2) to transmit data to the BIRO server for the production of the global report.

Step 1) involves **client data processing and statistical analysis**.

A BIRO "*Adaptor"* is used to establish a connection to the local database and export data from any format used by the local diabetes register to the standardized format complying with specifications agreed for the BIRO common dataset.

Standardized instructions (XML Schema) have been specifically developed to implement common BIRO definitions into a uniformly defined database allowing the use and pooling of data collected from different centres

A "*Metadata Dictionary*" has been realized in XML to incorporate a broad set of diabetes-related concepts and to derive new variables from the original ones that would be incorporated into the overall BIRO dataset.

A flat text file (XML export) is produced by each centre through the combined and repeated use of Java tools and the JDBC driver. This operation needs some basic pre-processing of local data to comply with basic requirements (e.g. storing one record for each individual subject in the production of the so-called "Merge Table").

A configuration file is needed for the BIRO Adaptor to apply specific options to the relevant driver. Such operation will be further simplified using a user friendly visual application.

The BIRO "*Database Manager*" reads XML files and subsequently stores records into a local (Postgres) database that is used to organize local data in an optimal way, so that they could be automatically processed by the statistical engine.

The Java language and tools e.g. Castor and Hibernate are used for the scope, as well as a configuration file.

The BIRO "*Statistical Engine*" connects to the local BIRO Postgres database and runs statistical functions to create "*statistical objects",* i.e. elements that carry essential data from local units to carry out the calculation of a specific parameter of indicator for a whole population (for more details refer to Section 3.3).

The BIRO *report template* precisely defines all outputs to be produced by the statistical engine.

The same structure is used to automate the production of both the individual centres and the global BIRO reports, a feature that is extremely convenient, as it allows using the same set of basic statistical functions for multiple, repeated applications.

The statistical engine connects to the local database using the open source statistical R software with proper Postgres drivers.

According to the specifications given by the report template, and the associated relevant definitions of the statistical objects, it processes the database to deliver statistical objects in the form of small CSV datasets, to be further processed to output individual centre and complete local reports in the form of pdf and html files, using the Latex software.

A compressed CSV folder is created to include all statistical objects produced by each run of the local reporting system, classified by date and centre id. This operation completes step 1) of the local engine.

Step 2) involves **data transmission**.

Specialized communication software has been developed to securely transmit the CSV folder including statistical objects from the local to the Central BIRO system.

Web services have been used to comply with basic requirements, including availability of an open platform-independent standard, XML support, usability over Internet protocols, open source implementation and comprehensive security support.

For the scope, World Wide Web consortium standards have been applied, based upon SOAP (Simple Object Access Protocol) for messaging, HTTP (Hypertext Transfer Protocol) for Internet transport and XML (eXtensible Markup Language) together with its security extensions XMLenc (encryption) and XMLsig (digital signatures).

Two J2EE server applications (sender and receiver) were set up for secure data exchange using the open source framework Apache Axis 2 together with Apache Rampart available for the Java 2 Enterprise Edition platform.

Security services (according to ISO/OSI 7498-2) have been carefully implemented.

For authentication, digital certificates trusted by a common certification authority were exchanged and installed in both servers. Access control has been configured so that only trusted identities are authorized to connect to services.

Confidentiality has been ensured by using encryption and data integrity, as well as non-repudiation provided by digital signatures.

Two alternative ways have been applied for encryption and digital signatures:

- Transport layer security using HTTPS, i.e. HTTP protocol together with SSL (Secure Sockets Layer) to protect the entire data stream exchanged between sender/receiver.
- SOAP messages encryption and digital signatures, utilizing XMLenc and XMLsig respectively, could be applied to protect well defined chunks of data, giving the application full control over further utilization, storage and processing of digital signatures and other security related information.

The *central* part of BIRO Architecture includes the set of software tools required by the BIRO server to undertake Step 3: **global statistical analysis.**

Step 3) involves several operations including **database processing and statistical analysis**.

At the central level, individual data are no longer required as the BIRO system only requires aggregate data, so all database specifications include meta-data mainly referred to the concept of statistical objects.

A specialised application (BIRO CSV Importer) has been developed in Java to read CSV files embedding statistical objects and to store them as separate tables of the Central BIRO database.

As for the Adaptor and Database Manager, a configuration file is required to allocate proper options.

Related statistical objects, transmitted by separate centres, are appended to the same table to form a global collection of local aggregate data.

The BIRO Database component of the Central Engine has been specifically developed to load and to organize all central aggregate data, as well as to perform basic data processing. Elementary Postgres functions have been used to compute a "*cumulative component*" for each statistical object as a pooled estimate of multiple "*local*" statistical objects.

Advanced statistical analysis in the Central Engine is performed by specific R functions. The cumulative components of statistical objects are processed to deliver all elements of the global report required to deliver the same template used for the local analysis. The template will be populated with results referring to the whole universe of BIRO collaborative centres.

Outputs of the Central Engine include a complete pdf report (as defined in the template), an html report (following specifications in the web portal), and CSV data, all produced using R and Latex software.

The final section of software development involves integration of the BIRO architecture into a unique, integrated software.

The BIRO process will be triggered by a simple "*local*" user friendly (GUI) application, allowing the user to:

- export local data stored into a local database to XML files running the BIRO Adaptor
- import XML files to the local database using the BIRO Database Manager
- produce the local statistical report
- send the local statistical objects to the Central BIRO System

A "central" GUI application will allow the user to:

- import statistical objects stored as csv files
- run the global statistical analysis
- produce the global BIRO report

The BIRO architecture will require for the Central Engine to be managed by the BIRO coordinator, which would evidently ensure compliance with all national and international security rules for the maintenance of the server.

A specific directory structure has been identified to allocate all different components of the BIRO system and drive the construction of a comprehensive software installation for both the client and server side (**Figure 2**).

Figure 1. BIRO Software Procedural Flow

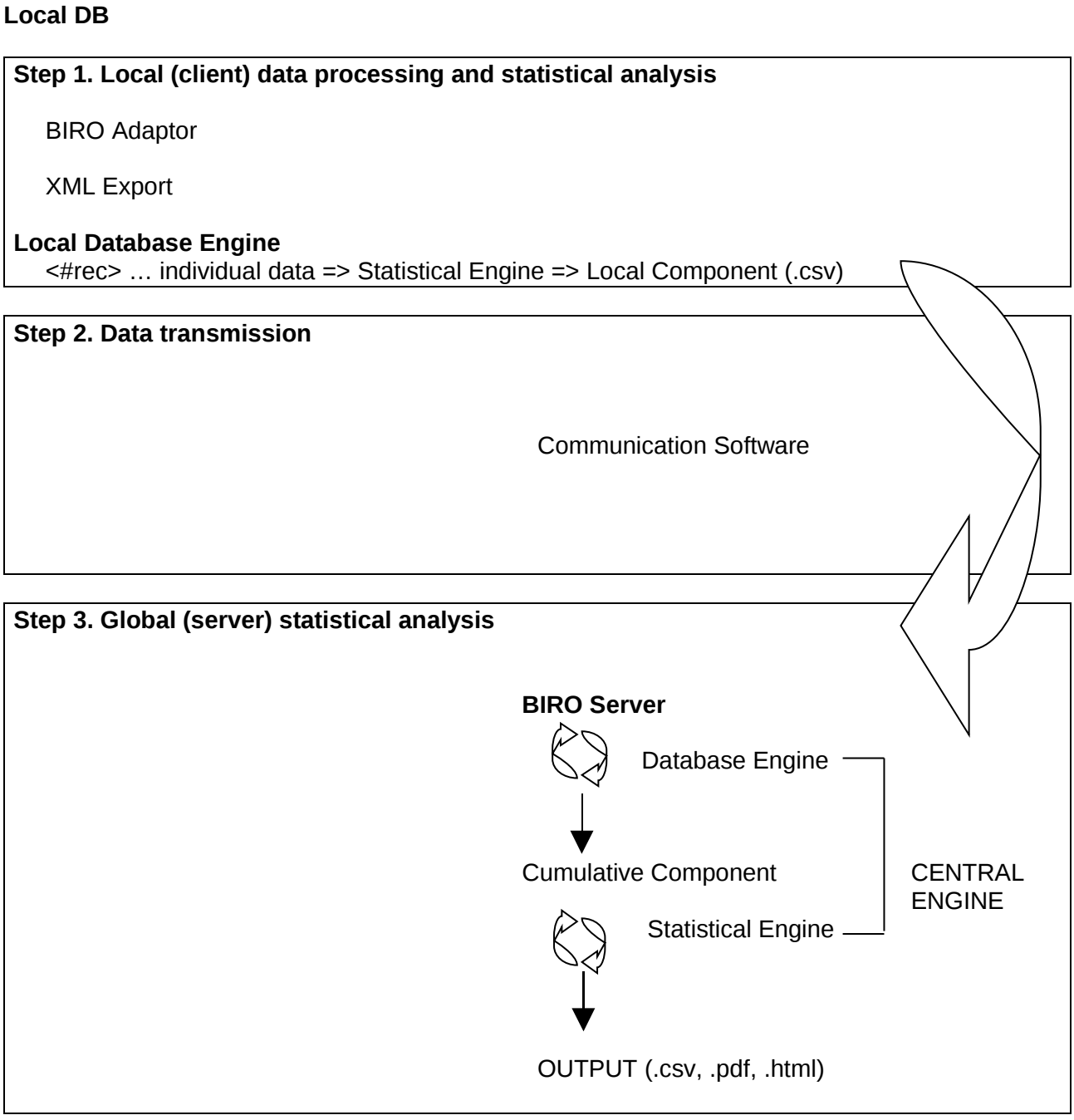


Figure 2. BIRO system directory structure

BIRO	**software**	**lib**	**html**					
			latex					
			maps					
			db					
			R	**source**	**biro**			
					packages	**linux**		
						pdf		
						vignette		
						win		
		_de_	**data**	**<centre_id>**				
			source	**gui**				
				adaptor				
				postgres				
			output	**xml**	**<datetime>**	**<year>**	**<centre_id>**	
		_se_	**data**	**<datetime>**	**<year>**	**<centre_id>**	patient.csv	
							episode.csv	
							pop.csv	
							mortality.csv	
			source	**R**	**backup**			
					formats			
					include			
					main			
					scripts			
					BIRO_se_run.r			
			output	**data**	**<datetime>**	**<year>**	**<centre_id>**	<local_comp.csv>
				reports	**<datetime>**	**<year>**	**<centre_id>**	**graphs**
								tables
								html
								images
								pdf
								report.aux,,.tex,.toc,.pdf,.html
		_cs_	**source**					
		_ce_	**data**	**<datetime>**	**<year>**	**<centre_id>**	<local_comp.csv>	
							<cum_comp.csv>	
			source	**R**				
			output	**data**	**<datetime>**	**<year>**	**<statobjects>**	
				reports	**<datetime>**	**<year>**	**graphs**	
							tables	
							html	
							images	
							pdf	
							report.aux,,.tex,.toc,.pdf,.html	

Figure 2 (continued). BIRO system directory structure

BIRO	**web**	**html**		
	docs	**guides**		
		deliverables	**wp7**	**d7.1**
				d7.2
	misc	**video**		
		audio		

3.2 GENERAL DESIGN OF THE STATISTICAL ENGINE

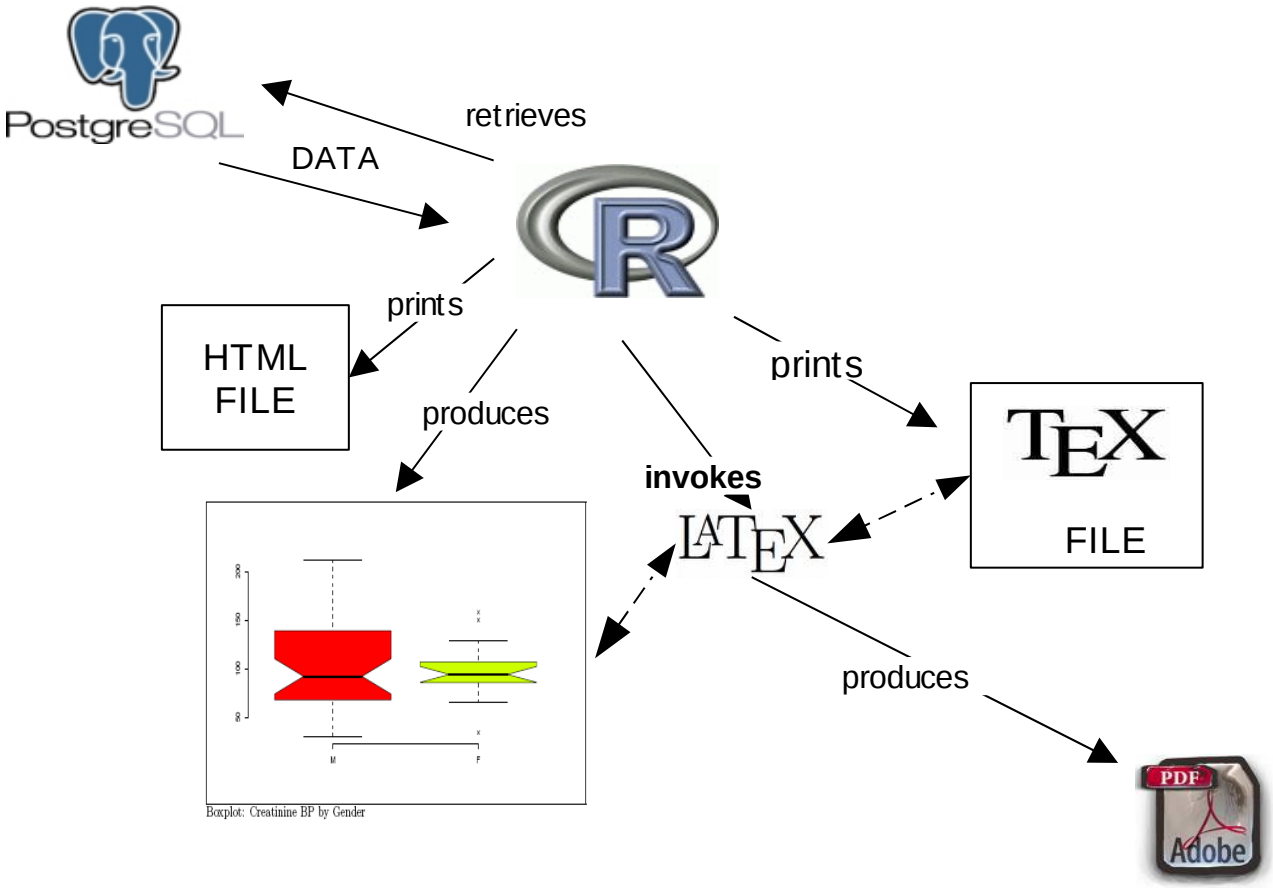
The design of the statistical engine revolves around the application of R software, playing a central role in the whole process of data loading, transformation and analysis (**Figure 3**).

Application of R is triggered by the user, either through a script command file or with the aid of a GUI interface. Software connects to the local database using proper Postgres drivers.

According to the specifications given by the *report template,* and the associated definitions of statistical objects, R functions process the Postgres database to deliver statistical objects in the form of small CSV datasets. Such datasets are further processed to produce individual centre outputs and full local reports in the form of pdf and html files, by using different graphical drivers and the high quality typographical software Latex.

A compressed CSV folder is created to deliver all statistical objects produced by each local run of the statistical engine, that will be stored in a directory properly named with the current datetime and centre id, ready to be transmitted as a compressed file to the central server.

Figure 3. BIRO Statistical Engine Design



3.3 DEFINITION OF STATISTICAL OBJECT

The statistical engine consists basically into a set of functions specifically designed to create and manipulate "*statistical objects"* according to an original definition provided to deliver the BIRO approach.

A statistical object is defined as "*an element of a distributed information system that carries essential data in the form of embedded, partial aggregate components, required to compute a summary measure or relevant parameter for the whole population from multiple sites*".

The definition of statistical objects is central to the functioning of BIRO, as it allows using pre-determined datasets as basic elements of statistical analysis ran on top of aggregate data to produce individual centre reports. Such partial results are then transmitted over the network for the production of global reports. This solution allows bypassing many possible risks and restrictions imposed by privacy legislation, as defined by the best architecture, avoiding the exchange of individual records.

Basically, statistical objects are tables that contain statistical aggregations of local data (arithmetic mean, percentile, variance, linear and logistic regression, bar plot data, histogram data, box pot data, etc), stored as flat text comma delimited files (CSV).

Statistical objects are organized according to a dictionary that includes basic components of frequency tables, measures of location, measures of dispersion, graphical elements, regression, and standardization.

Criteria agreed by the Delphi panel for the definition of the best architecture are duly taken into account in the specifications of statistical objects.

Table 1. Meta-data of BIRO Statistical Objects

Code	Sequential code based on the taxonomy of the statistical objects dictionary
Statistical Object	Name of the statistical object
Description	Short description of the principal content and output of the statistical object, and the main properties
Variables	Type of variables (categorical, continuous)
Properties	Mathematical and statistical properties in a distributed data environment
Local Component	OUTPUT OF THE LOCAL STATISTICAL ENGINE Technical characteristics of the statistical object that is produced from each data repository, to be sent to the central engine. Data section includes details on the format of the csv output.
Cumulative Component	CUMULATIVE DATASET PROCESSED BY CENTRAL ENGINE Technical characteristics of the procedure implemented in the central engine to produce the statistical object for the overall sample of connected repositories Data section includes details on the format of the csv output.
Output	STATISTICAL OUTPUT OF THE CENTRAL ENGINE Includes the list of components that will be computed and stored in the statistical object (ex: n, relative risks + confidence intervals, graph elements). Data section includes details on the format of the csv output. Defined codes are attributed to the list of electronic elements (e.g. XML tags, or csv tables)

Table 2. Statistical Objects Dictionary

SECTION 1. FREQUENCY TABLES

1.1 Univariate Frequency Distribution
1.2 Outliers
1.3 Contingency Table

SECTION 2. MEASURES OF LOCATION

2.1 Arithmetic Mean
2.2 Percentile
2.3 Range

SECTION 3. MEASURES OF DISPERSION

3.1 Variance
3.2 Interquartile Distance

SECTION 4. GRAPHICAL ELEMENTS

4.1 Bar plot
4.2 Histogram
4.3 Partial boxplot
4.4 Overall boxplot
4.5 Line plots
4.6 XY Plots
4.7 Webplot
4.8 Maps
4.9 Forest plot

SECTION 5. REGRESSION

5.1 Linear regression
5.2 Logistic regression
5.3 Meta-analysis

SECTION 6. STANDARDIZATION

6.1 Standardized rate
6.2 O-E

Table 3. GLOSSARY OF OUTPUT COLUMNS

Variable Name	Description
x	Variable X
y	Variable Y
n	number of non missing values for each cell
sum_x	
n_x	is the total number of non missing values for variable x

3.4 TAXONOMY OF STATISTICAL OBJECTS

SECTION 1. FREQUENCY TABLES

Code	1.1
Statistical Object	Frequency Distribution
Description	Ordered list of values, eventually ranked by n Can be univariate or multivariate
Variables	CONTINUOUS
Properties	Distribution from overall sample is identical to the distribution of the sum of individual frequencies
Local Component	Data matrix including all non-zero frequencies for each level of a target variable. Optionally the original variable can be rounded based on a desired interval (e.g. the one determined by privacy protection rules) Ex: 52=48=50 (>45,< 55) ~ 50; round to the nearest integer; day ~ month DATA: <1.1.a>id, date, stratum, x, n
Cumulative Component	Data matrix including the sum of all local data matrices DATA: <1.1.a> id, date, stratum, x,n
Output	List of frequencies for each level of the target variable DATA: <1.1.a> id, date, stratum, x,n Ordinary lines or XY plot etc. Histogram

Code	1.2
Statistical Object	Outliers
Description	List of values of a target variable above and/or below a specified threshold
Variables	CONTINUOUS
Properties	Outliers must be computed ex novo for each sample
Local Component	Data matrix including all non-zero frequencies for each level of a target variable DATA: <1.2.a> id, date, stratum, x,n
Cumulative Component	Data matrix including the sum of all local data matrices DATA: <1.2.a> id, date, stratum, x,n
Output	List of outlying values of the target variable <1.2.a> id, date, stratum, x

Code	1.3
Statistical Object	Contingency Table
Description	Each cell includes the number of subjects or percentage in a specific class obtained by combining levels of a series of factors
Variables	CATEGORICAL
Properties	The sum of n from tables from multiple centres is equal to the table that can be obtained by the overall rowtable Property applies to n and percentages (row,column,total)
Local Component	Data matrix including all non-zero frequencies for all combinations of a predefined list of variables DATA: <1.3.a>id, date, stratum, x, y, n, pct, evrate, l_evrate, u_evrate, rr, l_rr, u_rr, or, l_or, u_or <1.3.b> id, date, stratum, chisq, p_chisq,df
Cumulative Component	Data matrix including the sum of all local data matrices DATA: <1.3.a>id, date, stratum, x, y, n, pct, evrate, l_evrate, u_evrate, rr, l_rr, u_rr, or, l_or, u_or <1.3.b> id, date, stratum, chisq, p_chisq,df
Output	1) Table including n values for each combination, with percentages. Percent values are row percent within the inner column factor, Crude event rate + 95% confidence interval, Relative Risk + 95% confidence interval, Odds Ratio + 95% confidence interval 2) Bars: Row Bars grouped by the outer column factor (colors are based on different levels of the row variable) DATA: <1.3.a>stratum, x, y, n, pct, evrate, l_evrate, u_evrate, rr, l_rr, u_rr, or, l_or, u_or 3) Partial Bars: Row Bars grouped by centre, outer column factor (colors are based on different levels of the row variable) DATA: <1.3.b>id, date, stratum, x, y, n, pct, evrate, l_evrate, u_evrate, rr, l_rr, u_rr, or, l_or, u_or

SECTION 2. MEASURES OF LOCATION

Code	2.1
Statistical Object	Arithmetic Mean
Description	Weighted average of a single characteristic, with weights equal to the number of observations for each specific value of the target variable
Variables	CONTINUOUS
Properties	The mean of the overall sample is equal to the weighted mean of the arithmetic means from all local repositories
Local Component	Data vector composed of two quantities: sum of the values of the target variable; total number of observations DATA: <2.1.a>id, date, stratum, sum_x, n
Cumulative Component	Sum of the sum of values from each local object DATA: <2.1.a> id, date, stratum, sum_x, n
Output	Single value of the overall arithmetic mean: cumulative object, divided by the sum of the total number of observations from each local object DATA: <2.1.a>mean Single value of the arithmetic mean by centre: cumulative object, divided by the sum of the total number of observations from each local object, for each centre, for each stratum DATA: <2.1.b>id, date, stratum, mean

Code	2.2
Statistical Object	Percentile (Median=50%)
Description	Value that includes the desired percent (Median=Central) of observations in the weighted ordered list of a target variable. If the desired percentile lies between two values, then the percentile is equivalent to the arithmetic mean of the two adjacent values.
Variables	CONTINUOUS
Properties	The percentile of the overall sample is obtained from the complete ordered list, including n from all levels of the target variable in each local object
Local Component	Data vector composed of two quantities: value for each level of the target variable; total number of observations in the specific level DATA: <2.2.a>id, date, stratum, x, n
Cumulative Component	Sum of all ordered lists from each local object DATA: <2.2.a>id, date, stratum, x, n
Output	Single parameter value that includes the desired percent (Median=Central) of observations in the weighted ordered list of the target variable, obtained as a sum of all ordered lists from each local object DATA: <2.2.a>date, stratum, pcl_x Single parameter value by centre DATA: <2.2.b>id, date, stratum, pcl_x

Code	2.3
Statistical Object	Range
Description	List of two values: min and max in the ordered list of values of a target variable
Variables	CONTINUOUS
Properties	The min, max of the overall sample are the min,max of min,max obtained from each local measurement
Local Component	Data vector composed of two quantities: min, max DATA: <2.3.a>id, date, stratum, min_x, max_x
Cumulative Component	Data vector composed of the list of all unique values of min,max obtained from all local vectors DATA: <2.3.a>id, date, stratum, min_x, max_x
Output	Data vector composed of two quantities: min, max, computed as min,max from all local vectors Two values of the target variable DATA: <2.3.a>date, stratum, min_x, max_x Two values by centre <2.3.b>id, date, stratum, min_x, max_x

SECTION 3. MEASURES OF DISPERSION

Code	3.1
Statistical Object	Variance
Description	Sum of squared deviations from the mean divided by the total number of observations minus one. Can be interpreted as the average squared distance from the mean.
Variables	CONTINUOUS

Properties	The overall variance is equal to the sum of two components: the variance "within" data repositories, expressed as the weighted average of the variances in each data repository, and the variance "between", expressed as the weighted average difference between the mean at each data repository and the overall mean.
	
	Formula for the overall variance:
	`(weighted.mean(var,n)*(sum_n-length(unique(id)))`
	`+sum(((mean-weighted.mean(mean,n))^2)%*%n))/`
	`(length(x)-1)`

Local Component	List of values of the arithmetic mean, variance, total number of observations for each stratum of interest DATA: <3.1.a>id, date, stratum, mean, var, n
Cumulative Component	Appended list of values of the arithmetic mean, variance, total number of observations for each stratum of interest for all data repositories DATA: <3.1.a>id, date, stratum, mean, var, n
Output	Variance parameter and number of observations for each stratum DATA: <3.1.a>date, stratum, var, sum_n Variance by local data repository DATA: <3.1.a>date, stratum, var, sum_n

Code	3.2
Statistical Object	Interquartile range
Description	Overall difference between the upper and lower quartile (75%-25%)
Variables	CONTINUOUS
Properties	The 25%, 75% percentiles of the overall sample are obtained from the complete ordered list, including n from all levels of the target variable in each local object
Local Component	Data vector composed of two quantities: value for each level of the target variable; total number of observations in the specific level DATA: <2.2.a>id, date, stratum, x, n
Cumulative Component	Sum of all ordered lists from each local object DATA: <2.2.a>id, date, stratum, x, n
Output	List of two parameters corresponding to values including the 25%, 75% observations in the weighted ordered list of the target variable, obtained as a sum of all ordered lists from each local object DATA: <2.2.a>date, stratum, pcl_25x, pcl_75x, iqr Parameter values by centre DATA: <2.2.b>id, date, stratum, pcl_25x, pcl_75x, iqr

SECTION 4. GRAPHICAL ELEMENTS

Code	4.1
Statistical Object	Barplot
Description	Plot in which each bar represents the total number of observations or the percentage in a specific class obtained by combining levels of a series of factors
Variables	COUNT,CONTINUOUS
Properties	Each bar of the overall Barplot is equal to the total number or percentage of observations in a specific class summing up results from individual repositories
Local Component	Plot in which each bar represents the total number of observations or the percentage in a specific class obtained by combining levels of a series of factors DATA: <4.1.a>id, date, stratum, height
Cumulative Component	Plot of a series of bars stratified by data repository (region, centre) or target strata. <4.1.a>id, date, stratum, height Overall Barplot Each bar of the overall Barplot is equal to the total number of observations or the percentage observed in a specific class, summing up results obtained from all local repositories <4.1.b>date, stratum, height
Output	<4.1.a>,<4.1.b> PNG, GIF, JPG files

Code	4.2
Statistica Object	Histogram
Description	Plot in which each area of a specific bar represents the number of observations or the percentage in a specific class obtained by combining levels of a series of factors
Variables	CONTINUOUS
Properties	Each area of a specific bar of the overall Barplot is equal to the total number of observations or percentage observed in a specific class, obtained by summing up all results from local repositories Each area can be obtained as a product of the width of each class multiplied by the number of observations in the specific class unit
Local Component	Plot in which each area of a specific bar represents the total number of observations or the percentage in a specific class obtained by combining levels of a series of factors DATA: <4.2.a>id, date, stratum, width, density
Cumulative Component	Comparison across individual data repositories. Histograms are grouped by centre with areas corresponding to individual centres <4.2.a>id, date, stratum, width, density Overall Histogram Each area of a specific bar of the overall Histogram is equal to the total number of observations or the percentage observed in a specific class, obtained as a sum of values from local repositories <4.2.b>date, stratum, width, density
Output	<4.2.a>,<4.2.b> PNG, GIF, JPG files

Code	4.3
Statistical Object	Partial Boxplot
Description	Simultaneous graphical representation of measures of location and dispersion to represent the statistical distribution of a continuous variable. The graph includes the mean, median, interquartile range, two derived measures of deviation from the centre of the distribution, defined as “wiskers”, and extremely deviant observations, also known as “outliers”. Wiskers are calculated using the following formulas: Upper wisker = 75% percentile + 1.5 (interquartile range) Lower wisker = 25% percentile – 1.5 (interquartile range) Outliers are presented as values of a target variable above and/or below wiskers.
Variables	CONTINUOUS
Properties	Overall representation is a mere display of individual values obtained by local data repositories. No further data processing is required to produce the output display
Local Component	DATA: <4.3.a> id,date,stratum, mean, median, pcl_25x,pcl_75x, l_wisk, u_wisk, outlie_x
Cumulative Component	Appended list of boxplots submitted by individual data repositories: DATA: <4.3.a> id,date,stratum, mean, median, pcl_25x,pcl_75x, l_wisk, u_wisk, outlie_x
Output	<4.3.a> PNG, GIF, JPG files

Code	4.4
Statistical Object	Overall Boxplot
Description	Simultaneous graphical representation of measures of location and dispersion to represent the statistical distribution of a continuous variable. The graph includes the mean, median, interquartile range, two derived measures of deviation from the centre of the distribution, defined as “wiskers”, and extremely deviant observations, also known as “outliers”. Wiskers are calculated using the following formulas: Upper wisker = 75% percentile + 1.5 (interquartile range) Lower wisker = 25% percentile – 1.5 (interquartile range) Outliers are presented as values of a target variable above and/or below wiskers.
Variables	CONTINUOUS
Properties	Overall boxplot is computed by appending individual frequency distributions, summing up all frequencies for the union of levels observed, and computing the graphical representation from the weighted cumulative distribution.
Local Component	Data matrix including all non-zero frequencies for each level of a target variable. Optionally the original variable can be rounded based on a desired interval (e.g. the one determined by privacy protection rules) Ex: 52=48=50 (>45,< 55) ~ 50; round to the nearest integer; day ~ month DATA: <4.4.a>id, date, stratum, x, n
Cumulative Component	Overall boxplot obtained from weighted cumulative distribution DATA: <4.4.a> date,stratum, mean, median, pcl_25x,pcl_75x, l_wisk, u_wisk, outlie_x
Output	<4.4.a> PNG, GIF, JPG files

Code	4.5
Statistical Object	Line plot
Description	Overall or individual lines for each data repository or stratum, obtained by connecting dots relative to point estimates for a target Y variable (e.g. the average over a time interval), corresponding to increasing levels of a continuous X variable (e.g. time rounded by year).
Variables	COUNT,CONTINUOUS
Properties	Each line represents a one-to-one correspondence (bijective function) between X-Y values for a particular strata combination (centre, etc). An overall plot represents Y values for the mean/sum of data values from local repositories
Local Component	DATA: <4.5.a> id,date,stratum,x,y
Cumulative Component	Partial plot DATA: <4.5.a> id,date,stratum,x,y Overall plot DATA: <4.5.b> date,stratum,x,y
Output	<4.5.a> PNG, GIF, JPG files

Code	4.6
Statistical Object	XY bubble plot
Description	Graphical representation where a single dot is displayed for each combination of two target variables over orthogonal axes. A circle is plotted around each dot, with a radius weighted by the number of observations found in each combination.
Variables	CONTINUOUS
Properties	The overall XY plot is produced on top of the cumulative distribution obtained by summing up all results of individual bivariate distributions.
Local Component	Data matrix including all non-zero frequencies for each level of a combination of target X,Y variables. Optionally the original variables can be rounded based on a desired interval (e.g. the one determined by privacy protection rules) Ex: 52=48=50 (>45,< 55) ~ 50; round to the nearest integer; day ~ month DATA: <4.6.a>id, date, stratum, x, y, n
Cumulative Component	Data matrix including the sum of all local data matrices DATA: <4.6.a> id, date, stratum, x, y, n Overall plot DATA: <4.6.b> date, stratum, x, y,n
Output	<4.6.a> PNG, GIF, JPG files

Code	4.7
Statistical Object	Spider Plot
Description	Multidimensional graph depicting multiple synthetic values of a small number of variables (x,…,z) for different entities (e.g. centres), eventually grouped by overlapping strata, where each ray corresponds to a single variable ampng those chosen at the outset. All values are normalized over a specified range, usually based on min, max values across entities to be compared.
Variables	CONTINUOUS
Properties	The overall Spider Plot is produced on top of the cumulative distribution obtained by summing up values of target variables, by different strata levels, each corresponding to a target entity to be graphed as a spiderweb (ex: centres, gender, different risk level, etc).
Local Component	Data matrix including all non-zero frequencies for each level of a combination of target set of variables. Optionally the original variables can be rounded based on a desired interval (e.g. the one determined by privacy protection rules) Ex: 52=48=50 (>45,< 55) ~ 50; round to the nearest integer; day ~ month DATA: <4.7.a>id, date, stratum, id_ent, sum_x, n_x,…, sum_z, n_z
Cumulative Component	Data matrix obtained as a weighted sum of all data matrices, for each entity to be compared, normalized on the overall observed distribution of values (min, max over all spider objects). DATA: <4.7.a>id, date, stratum, id_ent, mean_x, ..., mean_z
Output	<4.7.a> PNG, GIF, JPG files

Code	4.8
Statistical Object	Maps
Description	Graphical representation of a value (e.g. a rate, mean, total number of observations), for each geographical cluster (e.g. region), plotted as a closed polygon.
Variables	CONTINUOUS
Properties	The overall map is produced on top of the cumulative distribution obtained by summing up values of the target variable for each geographical cluster
Local Component	Data matrix including all non-zero frequencies for each level of a target variable, for each geographical cluster. DATA: <4.8.a>id, date, stratum, id_geo, sum_x, n
Cumulative Component	Data matrix obtained as a weighted sum of all data matrices, for each geographical cluster DATA: <4.8.a>id, date, stratum, id_geo, sum_x, n Overall map DATA: <4.8.b>date, stratum, id_geo, sum_x, n
Output	<4.8.a> PNG, GIF, JPG files

Code	4.9
Statistical Object	Forest plot
Description	Graphical representation of a list of parameter estimates (e.g. ORs, means, O-Es), each corresponding to a predefined stratum (e.g. centre, or target variable category e.g. males, high levels of BMI etc), together with their 95% confidence interval limits. Point estimates are represented by dots, connected to lower and upper limits by a line. Optionally (e.g. in meta-analysis), a dot can be replaced by a square, whose side is proportional to the inverse of the variance of the estimate
Variables	CONTINUOUS
Properties	There are two different methods: exact or approximated. In the exact method, individual components required to compute the overall parameter estimates are saved in the format required to perform a generalized linear model (glm). The overall model is performed on top of the data matrix obtained by summing up all data contributed by individual data repositories. In the approximated method (meta-analysis), the target measure (e.g. OR) is computed at each local site together with its variance, for each predefined stratum, and a weighted average is computed to produce the mirroring overall estimates (fixed effect model). Optionally, a random effect model (Der Simonian – Laird) can be also used. Homogeneity test is saved accordingly (chi-square or Q).
Local Component	Exact method Multivariate distribution of confounders (X) and outcome variable (Y), weighted by the number of non missing observations DATA: <4.9.a>id,date,stratum,y, x,…,z, n Approximated method: DATA: <4.9.a>id,date,stratum,stat,variance
Cumulative Component	Overall estimates: DATA: <4.9.a>date, stratum, stat, l_stat, u_stat <4.9.b>date, test
Output	<4.9.a> PNG, GIF, JPG files

SECTION 5. REGRESSION

Code	5.1
Statistical Object	Generalized Linear Models (Exact method)
Description	Regression model built on top of a multivariate data matrix including a target outcome variable, Y, which can be dichotomous or continuous, and a list of X explanatory variables, of any kind. Models can include families and link functions in a variety of ways, allowing to estimate parameters based upon Poisson, Logistic, Linear regression etc
Variables	COUNT, CATEGORICAL, CONTINUOUS
Properties	The regression model is obtained from a cumulative multivariate table, computed as a sum of all local tables, including the number of observations for each unique combination of levels
Local Component	Multivariate distribution of confounders (X) and outcome variable (Y), weighted by the number of non missing observations DATA: <5.1.a>id,date,stratum,y, x,…,z, n
Cumulative Component	Sum of all tables including the multivariate distribution of confounders (X) and outcome variable (Y), weighted by the number of non missing observations DATA: <5.1.a>date,stratum,y, x,…,z, n
Output	Overall estimates: DATA: <5.1.a>date, stratum, stat, l_stat, u_stat <5.1.b>date, test

Code	5.2
Statistical Object	Meta-Analysis (Approximated method)
Description	Weighted average of regression parameters obtained from partial regression models built locally, using a multivariate data matrix including a target outcome variable, Y, which can be dichotomous or continuous, and a list of X explanatory variables, of any kind. Local models can include families and link functions in a variety of ways, allowing to estimate parameters based upon Poisson, Logistic, Linear regression etc
Variables	COUNT, CATEGORICAL, CONTINUOUS
Properties	Global regression parameters are obtained as an average of the relevant local regression parameter, weighted by the inverse variance. Confidence intervals are computed on the basis of meta-analytic theory. Random effect models may be applied, based on the Der Simonian-Laird method.
Local Component	Vector of parameter estimates, with variance DATA: <5.2.loc.a>id,date,stratum,beta,variance
Cumulative Component	Vector of parameter estimates, with variance DATA: <5.2.cum.a>id,date,stratum,beta,variance
Output	Vector of parameter estimates, with variance DATA: <5.2.out.a>date,stratum,beta,variance <5.2.out.b>date, test

SECTION 6. STANDARDIZATION

Code	6.1
Statistical Object	Standardized Rate
Description	Adjustment of the crude rate, computed as the original rate, plus the difference between the value of the indicator obtained from predicted values, based on a specified regression model, and the average population rate
Variables	CONTINUOUS, BOUNDED [0,1]
Properties	Any model equation can be applied to local and/or global samples to produce predicted values. Standardized rates are obtained summing up predicted values for each sample
Local Component	Multivariate distribution of confounders (X) and outcome variable (Y), weighted by the number of non missing observations DATA: <6.1.a>id,date,stratum,y, x,…,z, n
Cumulative Component	Sum of all tables including the multivariate distribution of confounders (X) and outcome variable (Y), weighted by the number of non missing observations DATA: <6.1.a>date,stratum,y, x,…,z, n <6.1.b>id,date,stratum,y, x,…,z, n
Output	Overall estimates: DATA: <6.1.out.a>date, stratum, beta, l_beta, u_beta <6.1.out.b>date, test Multivariate distribution of confounders (X) and outcome variable (Y), weighted by the number of non missing observations DATA: <6.1.out.c>id, date, stratum, rate, l_rate, u_rate

Code	6.2
Statistical Object	O-E
Description	In the case of a dichotomous outcome, is equal to the difference between the total number of events observed in a specific class (e.g. centre), and the sum of predicted values for the same class, based on a specified regression model,
Variables	CONTINUOUS, BOUNDED [0,1]
Properties	Any model equation can be applied to local and/or global samples to produce predicted values. Standardized rates are obtained summing up predicted values for each sample
Local Component	Multivariate distribution of confounders (X) and outcome variable (Y), weighted by the number of non missing observations DATA: <6.2.a>id,date,stratum,y, x,…,z, n
Cumulative Component	Sum of all tables including the multivariate distribution of confounders (X) and outcome variable (Y), weighted by the number of non missing observations DATA: <6.2.a>date,stratum,y, x,…,z, n <6.2.b>id,date,stratum,y, x,…,z, n
Output	Overall estimates: DATA: <6.2.out.a>date, stratum, beta, l_beta, u_beta <6.2.out.b>date, test Multivariate distribution of confounders (X) and outcome variable (Y), weighted by the number of non missing observations DATA: <6.2.out.c>id, date, stratum, oe, l_oe, u_oe

4. RESULTS

4.1 COMPONENTS OF THE STATISTICAL ENGINE

The statistical engine is characterised by the following structure (pseudo-code):

```
Start
1. Setup environment
2. Compute Indicator Statistics
 For each indicator in the Report Template:
 Loop Start
     Reference Indicator
IF i-th statistical procedure is TRUE then
 Apply Statistical Procedure
 Output production
     END
Loop End
3. Compile results
End
```

The loop is presented as a flow chart in **Figure 4**.

The first step relates to the definition of the workspace, data preparation, and output formatting (**Box 1**).

Execution starts with a fresh setup of the complete environment, including a check of the local OS version, any required installation of additional R packages, and the definition of global variables.The BIRO database is formatted by applying definitions in the data dictionary: new variables are created using a predefined set of cutoffs, new tables are created by merging and linking the original datasets into a new format amenable to statistical analysis. Finally, html and tex (pdf) outputs are initialized and formatted where required.

A second step is required to compute all indicator statistics (**Box 2**).

The complete list of BIRO indicators is read from the report template, along with definitions included in the data dictionary. An indicator “cohort” is automatically constructed, based upon the agreed specifications relative to the particular category of patients that must be included in each indicator.

Appropriate database and statistical procedure are executed to reproduce algorithms foreseen for each indicator, until the complete list of tasks is finalised and the set of planned outputs is entirely produced.

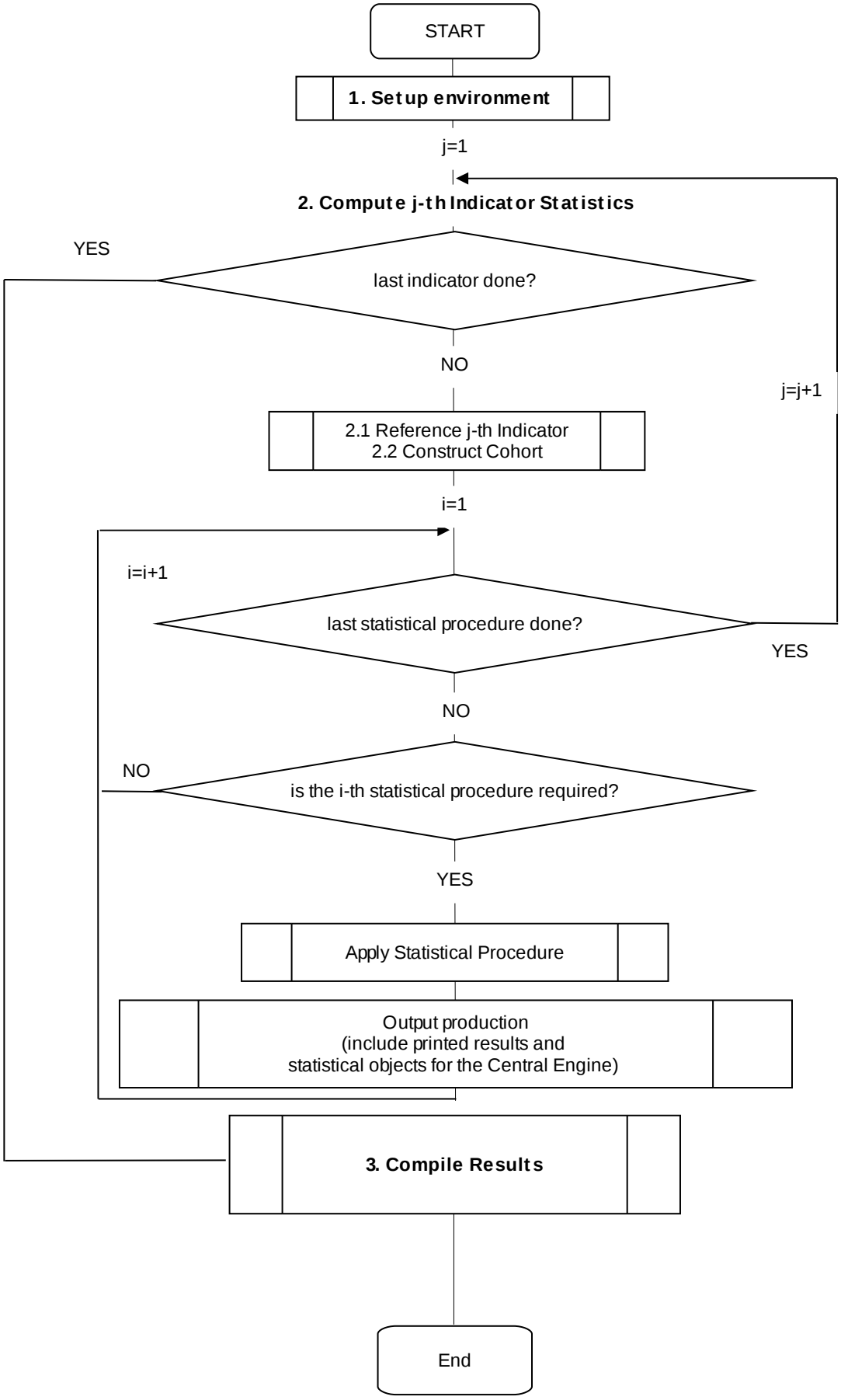
The loop ends when the complete list of indicators in the BIRO report template is produced (**Box 3**).

All results are compiled into an overall report that is produced in PDF and HTML format for the local centre site, including output files that include raw data, text listings (individual html tables) and graphical outputs.

Results are stored in a directory with a unique timestamp, whose content is sent by invoking a BIRO routine towards the central server, where they are used by the central engine to produce European results from a part or all BIRO participating centres.

The complete list of functions specifically created to realise the statistical engine, along with their location in storage files, is presented in detail in **Box 4.**

Figure 4. Statistical Engine Flow Chart



Box 1. Components and Files of the Statistical Engine (1): "Setup Environment"

`BIRO_se_.r`

`BIRO_se_setup.r`

- Clean active Workspace
- Load Environment Parameters (db drivers, directories, R libraries)
- Check for the existence of SE's directories, create missing ones
- Load Utilities
    - `BIRO_aggregate.r`
    - `BIRO_demographics.r`
- Load BIRO R libraries

`BIRO_se_.r`

`BIRO_se_datastep.r`

- Check for the existence of stored R data frames (Merge Table)
    - Connect to the BIRO database, transform it into R data frames and Merge Table, save transformations into .csv files
    - Load BIRO R data frames from transformed .csv files
- Apply thresholds, limits, levels for categorical variables from BIRO R libraries
- Apply date parameters stored in BIRO R libraries

`BIRO_se_.r`

`BIRO_se_report.r`

- Pre-production of Tex File
    - Open Tex file, write report cover page including authors, logo etc.

`BIRO_se_.r`

Box 2. Components and Files of the Statistical Engine (2): "Compute Indicator Statistics"

`BIRO_se_.r`

For each indicator in the Report Template:

Loop Start

Reference indicator (read relevant parameters)

`BIRO_se_report.r`
- Open Tex Indicator Section, write section cover page

`BIRO_se_indicator_<section>.r`
- Construct indicator data frame (valid cohort)
- Apply relevant indicator parameters to statistical procedure call

IF a possible statistical procedure among:

- Measures of location (`BIRO_se_location.r`)
- Measures of dispersion (`BIRO_se_dispersion.r`)
- Contingency Tables (`BIRO_se_tables.r`)
- Histograms (`BIRO_se_histograms.r`)
- Boxplots (`BIRO_se_boxplots.r`)
- Historical Trend (`BIRO_se_trend.r`)
- Forest plot (`BIRO_se_forest.r`)
- Trellis (`BIRO_se_trellis.r`)
- Webplots (`BIRO_se_webplots.r`)
- Maps (`BIRO_se_maps.r`)
- Regression (`BIRO_se_regression.r`)
- Standardization (`BIRO_se_standardize.r`)

IS TRUE then

Call and apply i-th Statistical Procedure
Output Production (`see relevant .r files above`)
- Save i-th set of produced statistical object as .csv
- Save i-th set of statistical objects as .csv
- Save i-th set of statistical tables as .html
- Save i-th set of produced graphs as .png

End

Loop End

Box 3. Components and Files of the Statistical Engine (3): "Compile Results"

`BIRO_se_.r`

`BIRO_se_report.r`

- Post-production of Tex file
  - close tex file and execute Latex to create report .pdf file

`BIRO_se_.r`

Box 4. List of Statistical Engine functions by storage file

biro_se_run.r
biro_se_datastep.r
BIRO_data_format
BIRO_loaddata
biro_se_.r
BIRO_se
biro_se_setup.r
BIRO_setenv
BIRO_dircreate
biro_se_recode.r
biro_se_boxplots.r
BIRO_boxplot
BIRO_plotbox
biro_se_dispersion.r
BIRO_range
BIRO_out
BIRO_variance
biro_se_histograms.r
BIRO_barplot
BIRO_drawbars
BIRO_pie
biro_se_location.r
BIRO_mean
biro_se_report.r
BIRO_report
BIRO_report_toc
open_tex
new_chapter
new_section
new_sub_section
include_tex
import_png
import_large_png
close_tex
import_png_html
new_section_html
new_chapter_html
new_sub_section_html
biro_se_tables.r
BIRO_fd
BIRO_table_format
BIRO_table_compute
BIRO_table_print
BIRO_df2html
biro_se_trend.r
BIRO_lines
BIRO_plotlines
biro_webplots.r
BIRO_spider
biro_se_indicator_clinical.r
biro_se_indicator_demographic.r
biro_se_indicator_health_system.r
biro_se_indicator_population.r
biro_se_indicator_risk_adjusted.r
biro_aggregate.r
BIRO_aggregate

Box 4 (continued). List of Statistical Engine functions by storage file

biro_demographics.r
BIRO_demographic
biro_explife.r
BIRO_explife
biro_forest.r
BIRO_forest
biro_maps.r
BIRO_maps
fabselreg
BIRO_map
BIRO_patmap
biro_regression.r
regression
biro_standardize.r
BIRO_standardize
biro_trellis.r
BIRO_histtrellis
BIRO_densitytrellis
BIRO_boxtrellis
biro_util.r
varclass
classlabel
classlabellist
BIRO_dframe

Box 5. List of files in the Statistical Engine

```
biro/software/_se_/source/r:
 biro_se_run.r

biro/software/_se_/source/r/main:
 biro_se_datastep.r
 biro_se_.r
 biro_se_setup.r

biro/software/_se_/source/r/formats:
 biro_se_recode.r

biro/software/_se_/source/r/include:
 biro_se_boxplots.r
 biro_se_dispersion.r
 biro_se_histograms.r
 biro_se_location.r
 biro_se_report.r
 biro_se_tables.r
 biro_se_trend.r
 biro_webplots.r

biro/software/_se_/source/r/scripts:
 biro_se_indicator_clinical.r
 biro_se_indicator_demographic.r
 biro_se_indicator_health_system.r
 biro_se_indicator_population.r
 biro_se_indicator_risk_adjusted.r

biro/software/lib/r/source/biro:
 biro_aggregate.r
 biro_demographic.r
 biro_explife.r
 biro_forest.r
 biro_maps.r
 biro_regression.r
 biro_standardize.r
 biro_trellis.r
 biro_util.r

biro/software/lib/db:
 postgresql-8.2-504.jdbc3.jar

biro/software/lib/html:
 biro-logo01.jpg
 layout_close.html
 layout_open.html

biro/software/lib/Latex:
 layout.tex
```

Box 5 (continued). List of files in the Statistical Engine

```
biro/software/lib/maps:
 admin98.avl
 admin98.dbf
 admin98.sbn
 admin98.sbx
 admin98.shp
 admin98.shx
 eurnuts0.avl
 eurnuts0.dbf
 eurnuts0.sbn
 eurnuts0.sbx
 eurnuts0.shp
 eurnuts0.shp.xml
 eurnuts0.shx
 eurnuts1.avl
 eurnuts1.dbf
 eurnuts1.sbn
 eurnuts1.sbx
 eurnuts1.shp
 eurnuts1.shp.xml
 eurnuts1.shx
 eurnuts2.avl
 eurnuts2.dbf
 eurnuts2.sbn
 eurnuts2.sbx
 eurnuts2.shp
 eurnuts2.shp.xml
 eurnuts2.shx
 eurnuts3.avl
 eurnuts3.dbf
 eurnuts3.sbn
 eurnuts3.sbx
 eurnuts3.shp
 eurnuts3.shp.xml
 eurnuts3.shx

biro/software/lib/r/source/packages/linux:
 Cairo_1.4-4.tar.gz
 DBI_0.2-4.tar.gz
 Epi_1.0.8.tar.gz
 epicalc_2.8.0.0.tar.gz
 Hmisc_3.4-4.tar.gz
 lattice_0.17-17.tar.gz
 maptools_0.7-16.tar.gz
 R2HTML_1.59.tar.gz
 rJava_0.6-0.tar.gz
 RJDBC_0.1-5.tar.gz
 rmeta_2.14.tar.gz
 sp_0.9-28.tar.gz
```

Box 5 (continued). List of files in the Statistical Engine

```
biro/software/lib/r/source/packages/pdf:
 Cairo.pdf
 dbi.pdf
 epicalc.pdf
 Epi.pdf
 Hmisc.pdf
 lattice.pdf
 maptools.pdf
 r2html.pdf
 rJava.pdf
 rjdbc.pdf
 rmeta.pdf
 sp.pdf

biro/software/lib/r/source/packages/vignette:
 sp_vignette.pdf

biro/software/lib/r/source/packages/win:
 Cairo_1.4-4.zip
 DBI_0.2-4.zip
 Epi_1.0.8.zip
 epicalc_2.8.0.0.zip
 Hmisc_3.4-4.zip
 lattice_0.17-15.zip
 maptools_0.7-15.zip
 R2HTML_1.59.zip
 rJava_0.6-0.zip
 RJDBC_0.1-5.zip
 rmeta_2.14.zip
 sp_0.9-28.zip
```

Geographical representation

Geographical information may be required to identify clusters of observations highlighting abnormal values for target indicators in diabetes. Maps can be created by categorizing areas according to a limited number of classes, either by using simple statistical measures e.g. percentiles, or advanced methods e.g. cluster analysis, regression trees etc, or by directly specifying cut-offs for the definition of classes.

Geographical areas are normally represented by closed polygons linking bi-dimensional points whose coordinates are stored as latitude and longitude in geographical libraries that in many cases are freely available.

Polygons may refer to entities of different size and nature e.g. cities, provinces, states, or entire continents. A fundamental problem hampering the uniform geographical representation of epidemiological data across Europe is the heterogeneity of such areas in different states. To map different definitions to a uniform representation, a specific taxonomy is required.

The European Union has developed the Nomenclature of Territorial Units for Statistics (NUTS) as a geocode standard for referencing the administrative divisions of countries for statistical purposes. A NUTS code begins with a two-letter code referencing the country and is available beyond the EU, with a two-letter code for a continent, two numbers for the country, and for the USA, Canada and Australia the states, provinces, and territories, separate numbers.

NUTS regions are based on the existing national administrative subdivisions. In countries where only one or two regional subdivisions exist, or where the size of existing subdivisions is too small, a second and/or third level is created. This may be on the first level (ex. France, Italy, Greece, and Spain), on the second (ex. Germany) and/or third level (ex. Belgium).

In smaller countries, where the entire country would be placed on the NUTS 2 or even NUTS 3 level (ex. Luxembourg, Cyprus, Ireland), levels 1, 2 and/or 3 are identical to the level above and/or to the entire country. Indicative thresholds are 3-7 millions for NUTS 1, 800,000-3 millions for NUTS2, and 150,000-800-000 for NUTS 3[8].

In BIRO, we recognize that some extra levels may be worth to be included for the specific organizational levels of health systems, and added them to the NUTS classification. In setting up the BIRO database, each centre is asked to specify a relevant classification of geographical information that is either included as a reference to the place of residency of the patient (patient dataset), or the location of the centre (data source dataset).

In BIRO, a total maximum number of 8 nested levels is generally considered for the purpose of recording and mapping geographical information. Each country must supply a transcoding table, to link across all codes. If a variable does not exist in one country, the coding of the first non missing variable at the higher level is applied.

BIRO geographical levels include the following definitions (along with an example for Italy):

Table 4. Regional Geographical Taxonomy Table

Variable	Description	Class	*Example for Italy*
continent	name of continent	BIRO-0	European Union
country	name of the country	NUTS-0	Italy
macroarea	group of subnational areas	NUTS-1	Macro-regions
region	name of region	NUTS-2	Region
lha	local health authority	BIRO-1	ASL
province	province	NUTS-3	Province
dhu	district health unit	BIRO-2	Health District
postcode	name of subprovincial unit	BIRO-3	Commune

Each class can be linked to a particular class of shapefiles, depending upon the level available. It is possible that a standard shapefile is not available for the specific level of detail: open repositories offer libraries e.g. the ESRI “admin98”, which only provides NUTS 3 levels for some countries. Few countries may have maps available at the BIRO-2, BIRO-3 levels.

Mapping is carried out in BIRO by processing geographical information stored for the patient or specific clinical unit. To simplify the process, BIRO considers only two variables for the scope, i.e. one to be available for the patient, and another for the centre. These variables do not share necessarily the same level of detail: patient references may be available in terms of postcode, while provinces may be used at the level of centre location.

The centre data descriptor must clearly indicate which of the 8 variables are used.

The statistical engine processes information stored in the database engine, producing aggregates by groups of patients and/or centre location.

The taxonomy is needed for two different reasons.

The first one relates specifically to the statistical engine (local processing): the aggregate table resulting from the statistical analysis (see statistical object 4.8) is merged to the regional taxonomy table through the appropriate data source descriptor, and the variable corresponding to the level available for the polygon ID in the target shapefile (which of course must be same detail or coarser) is used for mapping. If necessary, the level must be rescaled (see figure 4).

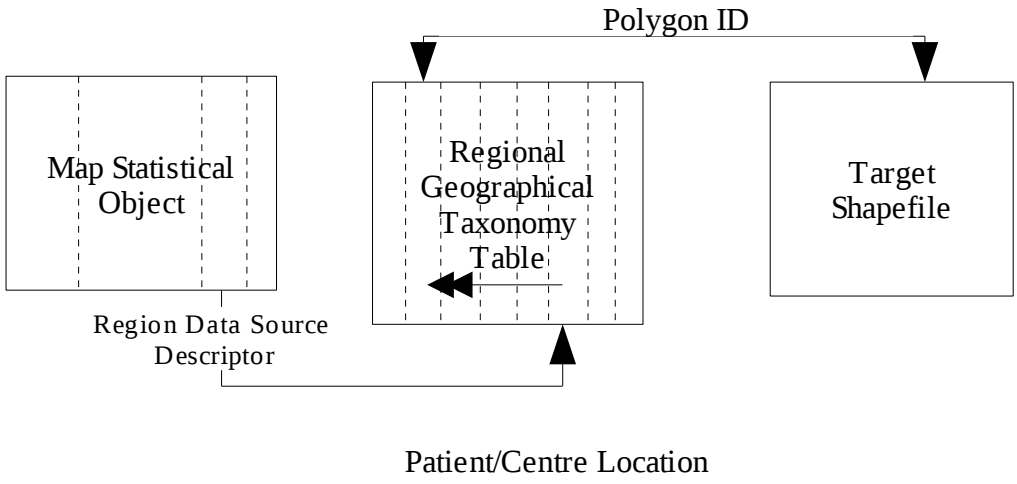


Figure 4. Linking geographical references to a target shapefile in BIRO statistical engine

In the central engine (global processing) the main problem arises when maps from different regions/countries must be produced, with heterogeneous levels recorded by different registers.

Since only one target shapefile is chosen to map all regions at the same time, the level of detail of the polygon ID in the shapefile determines the target geographical level for all regions.

Each portion of the overall cumulative table must be extracted and merged to the relevant regional taxonomy table through the relevant regional data descriptor (see figure).

Levels must be then rescaled to the one in target shapefile, and extracted tables from different regions can be appended to a unique table that is used to produce an overall map.

In some cases, a decision can be made to change original options selected, to optimise mapping. If there are regions with very little detail (e.g. countries or NUTS 9), either they are dropped or only the coarser subdivision is used. In any case, aggregate tables must be all linked to the same shapefiles.

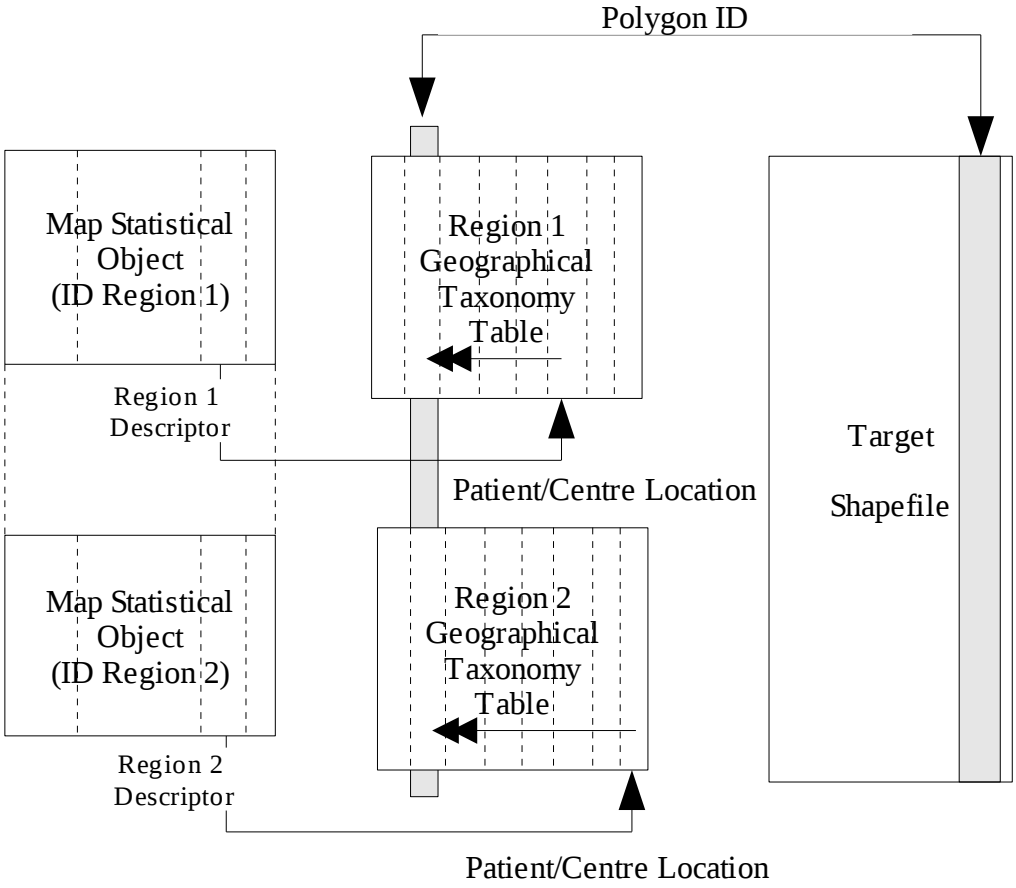


Figure 5. Linking geographical references to a target shapefile in BIRO central engine

Software/Hardware specifications and performance

Statistical engine has been successfully developed without noticeable deviations from the original plan and has been successfully tested on both major OS Windows (Vista) and Linux (Fedora 10).

Hardware consisted of average Intel-based PCs/Notebooks, the least powerful with the following specifications: CPU speed 2.0GhZ, 1Gb RAM, hard disk capacity of 100Gb.

Figures from a test run on data from the Umbria register for the production of an annual *local report* showed the following execution times on the same machine:

Centre	N Patients	N episodes	Elapsed Time
1	17,552	92,237	24' 25''
2	5,315	19,434	7' 01''
3	7,846	60,274	12' 20''
4	7,827	45,345	10' 51''
5	5,008	10,994	5' 22''

Outputs occupy an average storage space of about 30Mb, including data to be transmitted to the central server.

Installation of the software is identical regardless of the hardware, and requires R>1.8, Latex, Java 6.0 and PostgreSQL plus various additional libraries/packages that are included in its distribution.

Software is released using the GPL license and is authored by F.Carinci and L.Rossi.

4. CONNECTION TO THE CENTRAL ENGINE

Routines of the statistical engine produce the atomic elements required by the central engine to operate once they are transmitted to the central server.

Here we describe the statistical components of the central engine only in general: more details are provided in the specific report on WP10, Central Engine.

The first part of the engine is dedicated to the creation of an overall database from multiple aggregate tables, also known as the “Pile-Up Database” (**Figure 5**).

A compressed directory of “partial results” in .csv format is uploaded by each participating centre, after the application of the statistical engine. The central engine processes a predefined list of statistical objects required for the production of a specific indicator, checking for the presence in the “partial” directory.

The engine appends each object to the specific table formed by all same statistical objects that have been transferred by BIRO centres for a particular reference time interval. The database component of the central engine is invoked through a Java function specifically developed to load .csv objects in a PostgreSQL database (CSV “Importer”). The loop is completed once all objects are allocated to a PostgreSQL table.

The key component of the central engine operates once all tables from all centres have been allocated (**Figure 6**).

Basically, a set of routines “twin” to the statistical engine have been developed in a loop that is almost identical to the partial one. The major difference is that statistical procedures of the central engine operate on top of the PostgreSQL set of aggregate tables, and for these reasons they are similar in scope, but based on slightly different algorithms running on top of the “cumulative component” (see Table 1, Meta-data of BIRO Statistical Objects).

Outputs produced by the central engine are mostly identical both in format and content to those delivered by the statistical engine, with the exception of variations among regions that usually does not appear in local reports.

Figure 6. Pile-up database Flow Chart

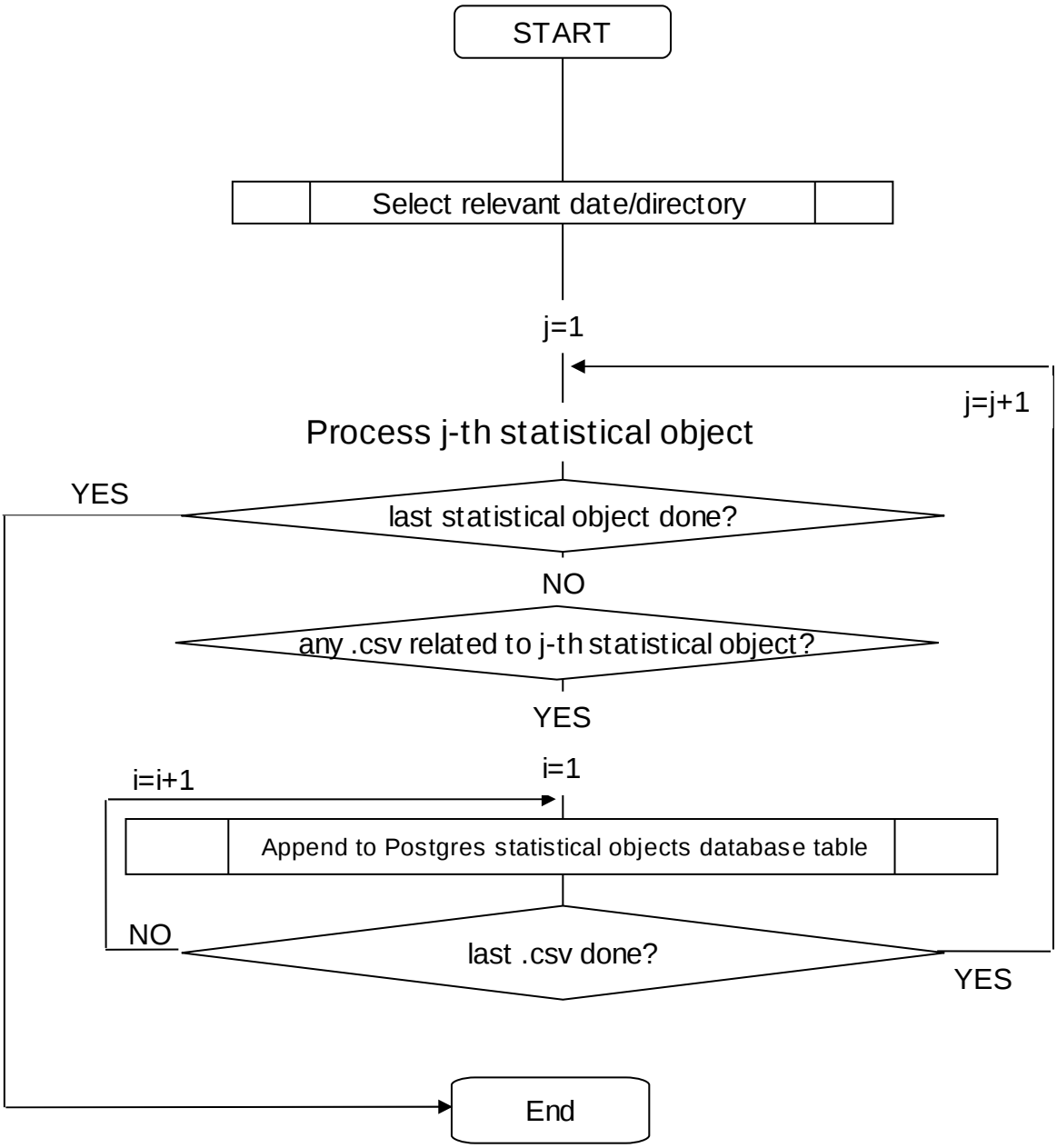
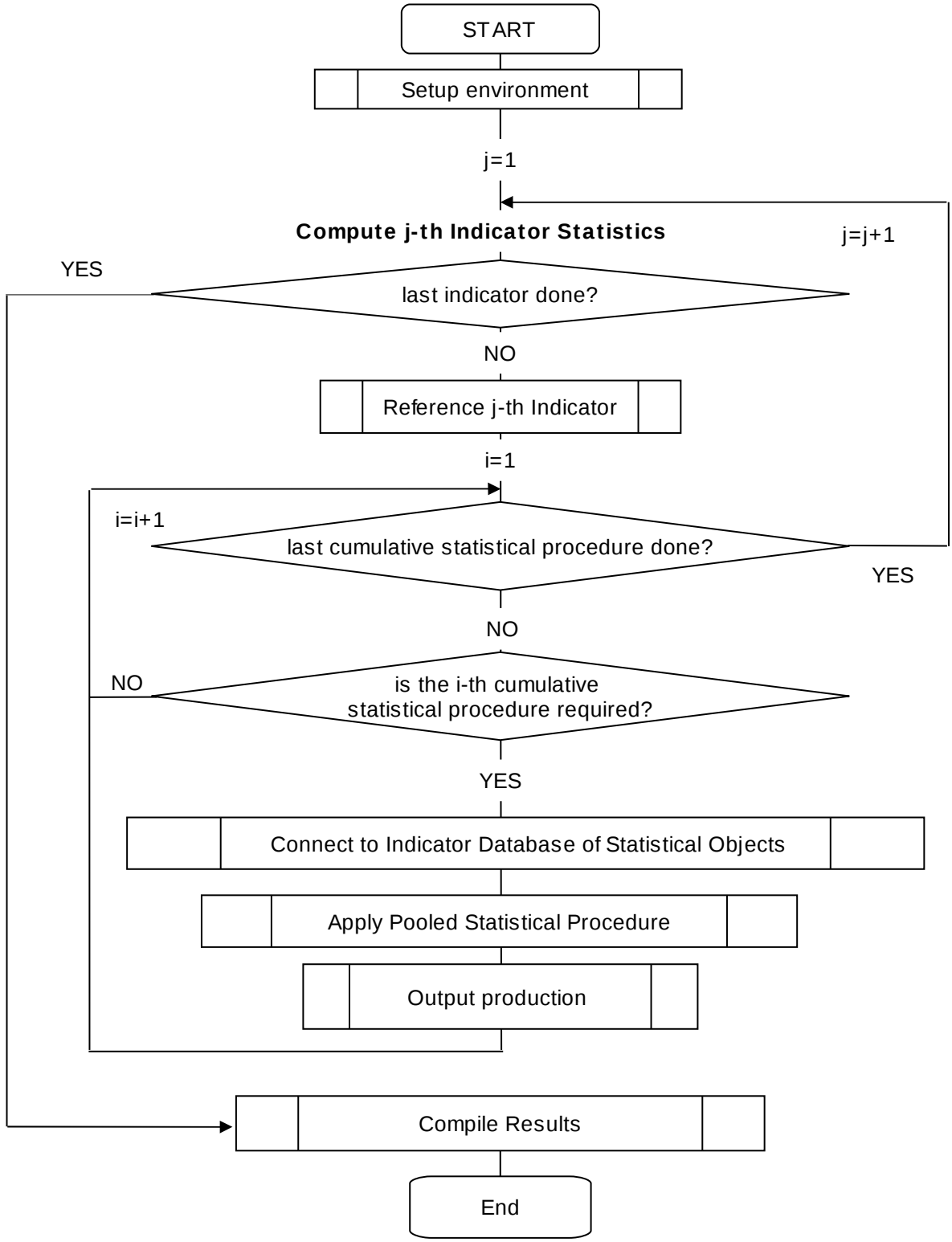


Figure 7. Central Engine Flow Chart



5. DISCUSSION

The ultimate aim of the BIRO Information System (SEDIS) is to link policy-makers, practitioners and end users through secure Internet software that is specifically designed to provide reports from the analysis of a unique distributed data-warehouse.

Here we will focus on the specific advantages offered by the (local) application of the statistical engine, leaving the presentation of the global application at the European level to the report specifically dedicated to the central engine.

The statistical engine represents a fundamental feature of SEDIS, as it delivers by design a range of outputs from descriptive frequencies targeted at clinical diabetologists, to population-based standardized analysis best suited for public health specialists.

The advantages offered by this component of BIRO are primarily due to the unique features of the overall design of the project, based upon the specific information-intensive management of chronic diseases.

In diabetes, definitions and practice guidelines change often, involving updates to the analytical software that must be re-run to get relevant up-to-date results.

BIRO allows to update its standardized information by linking statistical variables to a rich knowledge repository that uses evidence in context through a central “concept and data dictionary”. The dictionary is directly translated into database and statistical software, allowing to apply such definitions directly for the rapid production of new reports.

Through the generalisation of its data model, the same statistical results from the overall collaboration are saved into standard definitions, allowing to set appropriate terms of reference through which each region connected can apply the statistical engine to benchmark average results against own data.

As a matter of fact, statistical modelling in BIRO allows to create an average population resulting from connected centres almost in real time. Currently the same operation requires a long time to be realized, usually through ad hoc epidemiological studies.

The interesting aspect of the statistical engine is that it both serves the European Union to provide updated data in a sustainable manner, as well as the local user that through it can monitor in an inexpensive manner the clinical status of the served population.

In fact, traditional epidemiological studies usually do not return much information back to the data collection unit. This is a controversial aspect that in many collaborations have a direct impact on data quality and participation. Through its shared infrastructure, BIRO may also help to improve validity and completeness of information available.

The statistical engine may also help to enhance the statistical ability down to the level of individual clinical centres. It is important to highlight that a “region” in BIRO is not intended as an administrative entity, but as a network of centres sharing at a very low level a homogeneous set of organizational aspects, including the definition of individual data items and the way to measure and collecting them. BIRO sets itself at an upper level, through the definition of common standards that do not imply a change in the way information is treated at a lower level. It is the responsibility of the regional level to map local definitions against the common format that must be produced independently.

This way the use of the statistical engine can be moved further down to the level of the individual clinical centre, thus involving the individual practitioner directly.

The statistical engine provides a platform for accurate benchmarking that currently usually does not exist at the point of health care provision. BIRO allows answering very rapidly to questions e.g.: what is the average difference in glycated haemoglobin that a system can achieve within six months, with this therapy, in similar conditions? What is the average length of stay for a particular procedure? What outcomes can clinicians achieve for this particular population of patients? Why regions experience different variations? Why average outcomes are so different from my direct experience?

To make this information effective, the results obtained must be easy to interpret and use.

Both policy makers and physicians may gain particular advantage from browsing health reports made available in common formats, e.g. html and pdf files. Outputs may also be customized for own use. Tables and graphs can be imported in own presentations. Special reports can be produced for a class of physicians vs. a subset of regions/centres. A physician may inspect the average variation of glycated haemoglobin over time for a class of patients in a region, as opposed to the same results across different regions. Variation in clinical practice may be directly inspected.

On the other hand, it will be possible to choose from a range of possible standardization criteria those more adequate for a specific subgroup of patients, for instance using results from a multivariate model to define the relative risk of a particular category of patients.

The availability of a well-constructed and validated model represents an important step in the construction of a novel infrastructure that is capable of involving many of these aspects and can be equally applied to different geographical areas and collaborative networks.

An important element of the BIRO framework that is worth to be highlighted is the development policy, entirely based upon open source software that has very little to envy from its commercial counterparts. .

The implementation of a distributed system represents a valid scheme that exploits recent advancements in the field of IT and launches the adoption of new standards based upon low-cost platforms in the landscape of EU public health applications.

Finally, limitations of the current system are worth to be highlighted.

The statistical engine is based upon techniques for standardization and risk adjustment that do not allow to control for the potential bias that is indeed one of the major pitfalls of disease registers. As data is collected from automated sources, the inclusion of patients cannot guarantee about their level of representativeness.

Furthermore, no random selection process is put in place to get unbiased statistical estimates. The system relies upon collected data, on top of which it applies usual case-mix adjustment techniques.

Finally, bayesian techniques have not been developed to adjust for random variations in clinical centres characterised by small sample sizes.

Nevertheless, the progressive approach of disease registers must see the use of the statistical engine in perspective.

Firstly, the BIRO project aims at involving more and more centres in the collaboration, and within the EUBIROD project recently started it already grew up to twenty-two centres, from the seven originally involved.

Secondly, the BIRO data specifications include specific items that take into account data quality, including the concept of “validated diabetic patient” that must be taken as an important parameter to monitor a clean composition of the population under study.

Finally, the BIRO system, with its flexible data model, encourages further use of data linkage locally to pool clinical data with different administrative data sources (hospitals, diabetic clinics, GPs, pharmaceutical expenditures, pathology tests, etc), progressively covering the overall diabetic population in an exhaustive way, as it has never been possible before.

Once this will be realised, sampling may be specifically used to monitor quality and precision of regional registers, which in the meantime had become the gold standard in statistical information, from both perspectives of sustainability and speed of use.

The range of statistical routines to be included in the engine will be expanded in the framework of the WP “Epidemiological analysis” in the EUBIROD project.

6. CONCLUSIONS

The statistical engine is the core component of BIRO dedicated to the production of core outputs for the entire system.

The results of its application serve both the production of aggregate data that are the basic elements for the European analysis, as well as the needs of the local clinicians and policy makers.

Application of the statistical engine in regional and individual clinical units can be used in different ways.

Through it, networks of professionals may self-evaluate more rapidly and efficiently and implement *clinical governance* more convincingly. Prevention strategies and health services may be planned more carefully on the basis of factual information, making clinicians more accountable through the availability of up-to-date, well structured information.

Disease management is the key instrument that links patients and clinicians to the BIRO application. The cycle is virtuous and can generate synergies that result in improved health outcomes for the patients as well as improved information for the European Union through more accurate registers.

Each individual clinician, once inducted to using the software, can apply it independently and contribute to the European network through the production and submission of aggregate data to the central server. This way privacy is safeguarded at the highest level of protection, as a result of the application of a rigorous process of Privacy Impact Assessment (WP5).

Through the engine, researchers can deploy sophisticated statistical models for ordinary use and deliver more accurate benchmarks through multivariate risk adjustment.

The development of the statistical engine offers an open product available at no charge that will allow disseminating the BIRO technology more rapidly and effectively.

REFERENCES

1. M. Benedetti, F. Carinci, M. Federici, The Umbria Diabetes Register, *Diabetes Research and Clinical Practice*, Volume 74, S200-S204.
2. F. Carinci, M. Federici, M. Benedetti, Diabetes registers and prevention strategies: towards an active use of health information, *Diabetes Research and Clinical Practice*, Volume 74, S215-S219.
3. F.Carinci, Southern Population Health Information System (SPHIS). An integrated Population-based Data Warehouse, http://www.fabcarinci.net/projects/chsr/sphis.html
4. Rice N, Leyland A, Multilevel models: applications to health data, *J Health Serv Res Policy*, 1996; 1(3), 154-164.
5. A. Whitehead, J. Whitehead, A general parametric approach to the meta-analysis of randomized clinical trials, Stat Med (1991) , 1665–1677.
6. Churches T, Carinci F, Open source at the interface between policy and academia: towards evidence-based information systems, *4th International Conference on the Scientific Basis of Health Services Research*, Sydney, 22-25 September 2001.
7. Carinci F, Rose W, Advantages of Distributed Statistical Processing in Health Information Systems: the H+MetaBase project, Technical Report, Monash Institute of Health Services Research, Melbourne, Australia, 2003.
8. Wikipedia: Nomenclature of Territorial Units for Statistics, available at: http://en.wikipedia.org/wiki/Nomenclature_of_Territorial_Units_for_Statistics

APPENDIX

Statistical Software

Source Code