caBIG Software REQUIREMENTS SPECIFICATION TEMPLATE

DOCUMENT CHANGE HISTORY

Version Number	Date	Description
1.0	May 2006	Draft

Integrating Bioconductor and R into caBIG

Bioconductor / caBIG

Software Requirements Specification

Version 1.0

[Insert approval date of document]

Document Change Record

Version Number	Date	Description
0.1	*May 2006*	Initial document

TABLE OF CONTENTS

1. INTRODUCTION

This SRS captures the complete software requirements for the *Bioconductor/RWebServices* 1.0 release.

The purpose of this module is to provide tools to expose existing Bioconductor 'packages' as ca GRID services.

[The introduction of the SRS should provide an overview of the entire SRS. It should include the purpose, scope, references, and overview of the release being documented.]

1.1 SCOPE

Bioconductor is a collection of open-source software components based on the R programming language. Bioconductor is used for gene expression and other high-throughput analysis in molecular biology. R packages are collections of algorithms grouped to facilitate particular analyses.

This module allows R package developers to expose the functionality of their package as analytic services on caGRID. Operationally this will be performed at two different, but related, levels. One level is to expose Bioconductor packages on caGrid. A second level is to expose particular pipelines as 'exemplars'. Use cases outlining these goals are available online[1].

The *primary concern* of this project is to develop tools for converting existing Bioconductor packages to caGRID analytic services. This is in contrast to other modules, which may focus on providing specific functionality as analytic services. This portion of the project involves development of software components to wrap functionality written in the R programming language as Java components, and to expose these Java components as analytic web services.

The *secondary concern* of this project is to expose particular Bioconductor pipelines as 'exemplars'. Exemplars are chosen from three Bioconductor packages. Candidate packages include *affy* (used for microarray analysis) *PROcess* (SELDI-TOF proteomics data) and *DNAcopy* (array CGH data).

1.1.1 Identification

Bioconductor / caBIG 1.0.

1.1.2 System Overview

The purpose of the system and software is to provide tools that facilitate packaging of existing Biocondcutor packages as web services, exposure of web-services enabled packages as caGRID services, and discovery and invocation of web-services enabled packages as caGRID analytic services.

[1] http://cabigcvs.nci.nih.gov/viewcvs/viewcvs.cgi/bioconductor/AdministrativeDeliverables/Use_Case.pdf

The tools in RWebServices represent new software development, sponsored by the National Cancer Institute through the cancer Biomedical Informatics Grid (caBIG). End users include caBIG participants, as well as members of the public. Module development is conducted at the Fred Hutchinson Cancer Research Center in Seattle, WA, under the overall direction of Dr. Robert Gentleman.

Software developed in this project will be installed and used at our caBIG adopter sites (Northwestern University) and will be available for installation and use at other caGRID locations.

1.1.3 Document Overview

This SRS captures the complete software requirements for *Bioconductor / caBIG* 1.0 as part of the *Integrating Bioconductor and R into caBIG* project.

There are no security or privacy considerations directly associated with use of this software. However, software underlying the project cannot be considered secure; security must be provided by other caGRID software layers.

The underlying R software is open source and licensed under the GPL. Additional license restrictions apply to individual Bioconductor packages.

The remaining SRS sections are organized as follows:

- **Section 2. Project Description**: Describes the general factors that affect *Bioconductor / RWebServices* 1.0
- **Section 3. Requirements**: Describes all the software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements.
- **Appendix A. Acronym List:** Defines the abbreviations used on the project.

1.2 REFERENCE DOCUMENTS

For additional project and requirement specific information, please consult the administrative deliverables and documentation available at:

- http://cabigcvs.nci.nih.gov/viewcvs/viewcvs.cgi/bioconductor/AdministrativeDeliverables

2. PROJECT DESCRIPTION

2.1 PROJECT PERSPECTIVE

There are several software product components to be produced by this project. In broad terms, collaborations between three R packages (to be developed here) allow R functions to be wrapped as Java beans. Java beans are then converted (using existing technologies) to web services. Script and other methodologies (incorporating existing technologies, augmented with documentation and development here) transform the web service enabled Java to caGRID-enabled components. Additional scripting and installation documentation (incorporating existing technologies, and developed here) facilitate invocation of Bioconductor functions as caGRID analytic services.

The delivery platform being targeted is a current linux operating system; this does not preclude deployment on Windows operating systems. Specific constraints are listed below.

2.2 PROJECT FUNCTIONS

TypeInfo is an R package to provide strong typing and introspection on function argument and return types. TypeInfo is used to annotate functions in existing Bioconductor packages, and is a necessary first step in making the functions available as web services.

RWebServices is an R package (with some C code) that produces Java beans encapsulating the function and data type signatures of TypeInfo-annotated functions. The package uses type specification and R introspection facilities to determine the underlying data structure of R objects. The data structure is then translated into a corresponding Java class representation.

SJava is an existing technology consisting of R, Java, and C code. It allows R functions to be called from Java, and vice versa. SJava recognizes Java class objects, and translates data represented in Java classes to their corresponding R representation. Data translation occurs at the C level. RWebServices augments the data translation facilities of SJava. The Java beans produced by RWebServices are fully interoperable with SJava.

A diversity of introspection tools allow expression of Java beans as web services, automatically generating appropriate WSDL and other data descriptions as necessary. Additional standard tools allow these Java beans to be exposed as web services. The tools for Java introspection and exposure of Java beans as web services are existing technologies, and are not a software product of the current work.

To be published as caGRID analytic services, standard web services require semantic annotation. The tools required for semantic annotation are not a software product of the current project. However, some steps involved in semantic annotation (particularly, combining manually curated semantic annotations with programmatically generated WSDL) are amenable to automation. Scripts and other technologies will be developed to automate the portions of this process that are *uniquely* relevant to providing Bioconductor packages as caGRID analytic services.

Installation of appropriate modified Bioconductor packages as caGRID analytic services involves steps identical to those required to install any collection of Java objects as analytic services, so the tools required for this step are not a software product of the current project. Steps specific to installing Bioconductor packages as web services will be documented; scripts or other

functionality to automate specific steps in this process (e.g., coordinating installation of Bioconductor packages, their Java bean representation, WSDL, and semantic annotation information) will be developed.

Invocation of Bioconductor functionality as an analytic service requires correct installation of R, Bioconductor, and SJava, in addition to standard requirements for caGRID analytic services. Our project will document recommended ways of installing R, Bioconductor, and SJava, and will provide installation software to automate this process, as appropriate.

2.3 USER CHARACTERISTICS

Primary users of this software include Bioconductor package developers and caGRID node administrators. An secondary user is the individual invoking Bioconductor functionality as a web service.

The Biocondcutor package developer is computationally- and statistically capable individual wishing to produce or modify an R package to be deployed as a web service in caGRID. The individual is familiar with R, but has no special skills related to web services. The individual needs to be informed of the steps required for function and data type specification, and for producing Java beans corresponding to their Bioconductor package.

The caGRID node administrator is familiar with caGRID architecture and semantic annotation requirements of caBIO. The role of the administrator is to facilitate semantic annotation, and to install and deploy Biocondcutor packages as web services. This individual may have limited familiarity with R, requiring that installation steps unique to R need to be documented.

2.4 CONSTRAINTS

a. Architecture. Our project is directed toward a current linux operating system with standard development tools. These tools include those[2] required to build R. The system has versions of Java required for SJava[3] and caBIO[4]. Deployment and invocation tools require the web service infrastructure specified by caGRID[5]. Extension to use on Windows requires installation of specific but readily available software tools for R development, as indicated in footnote 2; Java and caGRID requirements on this platform are detailed in the documents cited earlier.

b. Data standards. Bioconductor packages analyze a diversity of data types; ensuring that these correspond to existing data types accessible as caGRID data services is the responsibility of individual package developers.

---

[2] http://cran.r-project.org/doc/manuals/R-admin.html

[3] http://www.omegahat.org/RSJava/

[4] https://cabig.nci.nih.gov/

[5] https://cabig.nci.nih.gov/workspaces/Architecture/caGrid

Analytic services frequently have 'tuning' and other parameters required for function evaluation but largely irrelevant to semantic interoperability. Creative solutions to conforming such parameters to semantic annotation requirements are being actively pursued.

c. Programming language versions. R packages developed during this project will require the current release version of the language. Requirements on Java are those specified by caGRID.

2.5 QUALIFICATION PROVISIONS

a. Specific code-based test facilities will be constructed to qualify requirements 3.1.*, 3.2.*, and 3.5.*.1-3.5.*.3 (* represents all subsections).

2.6 ASSUMPTIONS AND DEPENDENCIES

This project relies on services provided by caDSR (data type curation) and caGRID (analytic service infrastructure).

3. REQUIREMENTS

There are four requirement types indicated below. A 'component' is software unit responsible for particular functionality. A 'utility' denotes shell or other scripts to facilitate particular project objectives. 'Exemplars' are specific Bioconductor functionality to be exposed as proof-of-methodology.

3.1 TYPE SPECIFICATION (TYPEINFO)

Table 3.1-1: Type Specification (TypeInfo) Requirements

Req ID	Requirement Type	Requirement Description	CCR # (Optional)	Trace from User Requirement/ Trace to System Requirement (Optional)
3.1.1	Component	Apply type specification to standard R function arguments and return values.		
3.1.2	Component	Allow multiple type specification for each function call.		
3.1.3	Component	Allow introspection, viz, query function for argument and return type information.		
3.1.4	Component	Allow introspection to facilitate type specification of S4 methods.		

3.2 R AS WEB SERVICES (RWEBSERVICES)

Table 3.1-1: R as Web Services (RWebServices) Requirements

Req ID	Requirement Type	Requirement Description	CCR # (Optional)	Trace from User Requirement/ Trace to System Requirement (Optional)
3.2.1	Component	Produce Java classes corresponding to S4 or core R data types		
3.2.2	Component	Produce Java beans encapsulating function signature(s)		
3.2.3	Component	Provide (in conjunction with SJava) data translation services between Java and R internal representations of core R data types.		

3.3 CAGRID ANALYTIC SERVICE DEPLOYMENT

Deployment of analytic services as caGRID-accessible web services, including methods for correctly installing R and Bioconductor, will be provided as documentation rather than software tools.

3.4 EXEMPLAR FUNCTIONALITY

Our tools enable annotation and publication of Bioconductor packages as web services, rather than wholesale conversion of Bioconductor packages for this purpose. Nonetheless, our project

includes exemplars chosen from three different areas of Bioconductor functionality. The goal is to expose at least one function from each package, as a caGRID web service.

3.4.1 Affy

Table 3.5.1-1: Affy Exemplar Requirements

Req ID	Requirement Type	Requirement Description	CCR # (Optional)	Trace from User Requirement/ Trace to System Requirement (Optional)
3.5.1.1	Exemplar	Expose specific functionality as a web service, using TypeInfo, RWebServices, and SJava		
3.5.1.2	Exemplar	Invoke and perform analytic service, reporting correct result to user		
3.5.1.3	Exemplar	Report errors during execution		

3.4.2 PROcess

Table 3.5.2-1: PROcess Exemplar Requirements

Req ID	Requirement Type	Requirement Description	CCR # (Optional)	Trace from User Requirement/ Trace to System Requirement (Optional)
3.5.2.1	Exemplar	Expose specific functionality as a web service, using TypeInfo, RWebServices, and SJava		
3.5.2.2	Exemplar	Invoke and perform analytic service, reporting correct result to user		
3.5.2.3	Exemplar	Report errors during execution		

3.4.3 DNAcopy

Table 3.5.3-1: DNAcopy Exemplar Requirements

Req ID	Requirement Type	Requirement Description	CCR # (Optional)	Trace from User Requirement/ Trace to System Requirement (Optional)
3.5.3.1	Exemplar	Expose specific functionality as a web service, using TypeInfo, RWebServices, and SJava		
3.5.3.2	Exemplar	Invoke and perform analytic service, reporting correct result to user		
3.5.3.3	Exemplar	Report errors during execution		

APPENDIX A – ACRONYM LIST

[Insert an alphabetical listing of all acronyms and abbreviations, and their meanings as used in this document.]

Term/ Abbreviation	Description
SJava	Java / R bridge software and client/server framework
DBA	Database Administrator
PM	Project Manager
RM	Requirements Manager
RTM	Requirements Traceability Matrix
SCM	Software Configuration Management
SDLC	Software Development Lifecycle
SDP	Software Development Plan
SE	Software Engineering
SEPG	Software Engineering Process Group
SM	Software Manager
SOP	Standard Operating Procedure
SPI	Software Process Improvement
SQA	Software Quality Assurance
SW	Software
TBD	To Be Determined
TM	Test Manager