

Rwui: A web application to create user friendly web interfaces for R scripts

by R. Newton and L. Wernisch

Summary

Rwui is used to create web applications for running R scripts. All the code for the web application is generated automatically so that a fully functional web interface for an R script can be downloaded and up and running in a matter of minutes.

Although of general applicability, Rwui is designed primarily with bioinformatics applications in mind; aimed at bioinformaticians who are developing a statistical analysis of experimental data for collaborators and who want to automate their analysis in a user friendly way. Rwui may also be of use for creating teaching applications.

Rwui may be found at <http://rwui.cryst.bbk.ac.uk>

Introduction

R is widely used in the field of bioinformatics. The Bioconductor project (Gentleman et al., 2004) contains R packages specifically designed for this field. However many potential users of bioinformatics programs written in R come from a non-bioinformatics background and are unfamiliar with the language. One solution to this problem is to provide user-friendly web interfaces for R scripts. Values for variables and data files for processing are entered by the user on a web form. The application then runs the R script on a server, out of sight of the user, and returns the results of the analysis to the user's web page.

A web interface for an R script means that the script can be used by anyone, even if they have no knowledge of R. Because the web interface runs on a server the application can be accessed remotely so the user does not need to have R installed on their machine. And updates to the script need only be made to the copy on the server.

Rwui (R Web User Interface) is a web application that creates web applications for running R scripts. Code for the web application is generated automatically so that a fully functional web interface for an R script can be implemented in a matter of minutes.

The completed web applications run on Tomcat servers. Tomcat is a free and widely used server software, obtainable from <http://tomcat.apache.org/> and easy to install on both Unix and Windows machines. If remote access to applications is not required, Tomcat can be installed on a stand-alone machine and web applications accessed in a browser on

the machine via the 'localhost' URL.

Using Rwui

The information that Rwui requires in order to create a web application for running an R script is entered on a sequence of forms. After entering a title and introductory text for the application, the user selects the input items that will appear on the application's web page. Input items may be Numeric or Text entry boxes, Checkboxes, Drop-down lists, Radio Buttons, File Upload boxes and a Multiple/Replicate File Upload page. Each of the input variables of the R script, that is, those variables that require a value supplied by the user, must have a corresponding input item on the application's web page.

Section headings can also be added if required. Rwui displays a facsimile of the web page that the user has created as items are added to the page. Input items are given a number so that items can be deleted and new items inserted between existing items. After uploading the R script, Rwui generates the web application which can be downloaded as a zip or tgz file.

Once uncompressed the download contains the complete application, including all the source code, in a number of directories. All the files are included for the benefit of anyone who would like to make modifications of their own to the application. Normally however no modifications are necessary, in which case only one of the files is required, namely, the .war file in the 'deploy' directory.

The completed applications will run on a Tomcat server. All that needs to be done to use the downloaded web application is to place the .war file in the Tomcat 'webapps' directory. In addition, the file permissions of an included shell script must be changed to executable.

An application description file is included in the download. This is useful if the application requires modification at a later date. The details of the application can be re-entered automatically into Rwui by uploading the application description file. The application can then be edited and rebuilt within Rwui. Alternatively, since the application description file is written in XML, an existing application can be modified by editing the XML, prior to uploading the file to Rwui for rebuilding.

If modifications are required which involve changes to the application's source code, for example changing the format of the Results page, the download contains an Ant script (<http://ant.apache.org/>) for rebuilding and redeploying the application.

System Requirements

In order to use the web applications created by Rwui a machine is required with Tomcat version 5.0 or later, Java version 1.5 and an R version compatible with the R script(s).

Ant will be needed if changes are to be made to the downloaded application and the Ant script, included in the download, is used for rebuilding. Although a server running a Unix operating system is preferable, the completed applications will work without modification on a Tomcat server running Windows XP.

Using applications created by Rwui

A demonstration application created by Rwui can be accessed or downloaded via links on the 'Help' page of Rwui.

Applications created by Rwui can include a login page. Access can be controlled either by a single password, or by username/password pairs.

If a Multiple/Replicate File upload page is included in an application created by Rwui then the application consists of two web pages on which the user enters information. On the first web page the user uploads multiple files one at a time. Once completed, a button takes the user to a second web page where singleton data files and values for all other variables are entered. The 'Analyse' button on this page submits the values of variables, uploads any data files and runs the R script. If a Multiple/Replicate File upload page has not been included in an application, then the application consists of this second web page only.

During an analysis the application first checks the validity of the values that the user has entered and returns an error message to the page if any are invalid. Only numeric values or NA can be entered into Numeric entry boxes. Numeric, Text and File Upload boxes cannot be left blank. Multiple/Replicate Files must be unique, that is, the same file cannot be entered more than once.

When creating the application validation may be turned off, although this is not recommended. There is also the option of setting up parent-child linking of validation. Child input items can be linked with parent radio buttons and/or checkboxes. In the completed web application, if a parent radio button or check box is not 'ticked' then its child input items are ignored and won't cause validation problems if they are currently empty.

On completion of the analysis, a link to a Results page appears at the bottom of the web page. The user can change data files and/or the values of any of the variables and re-analyse, and the new results will appear as a second link at the bottom of the page, and so on. Clicking on a link brings up the Results page

for the corresponding analysis. The layout of these pages works best in browsers other than Internet Explorer which always puts scrollbars around text files, whatever their size, and won't display png images.

The user can download individual results files by clicking on the name of the appropriate file on a Results page. Alternatively, each Results page also contains a link which will download all the results files from the page and the html of the page itself. In this way the user can view offline saved Results pages with their associated results files.

Structure of the applications created by Rwui

Rwui creates Java based applications that use the Apache Struts framework. Struts is an open source and popular Java based framework for constructing web applications.

The Apache Struts framework separates the three main components of a web application; the View (the way in which information is presented to the user), the Controller (controlling the flow of the application) and the Model (the data processing). This produces a web application that is well-organised, stable and extensible.

In web applications created by Rwui, the Model part of the application, a Java program, passes the information entered by the user on the application's web pages to the R script and then runs the script. The application waits for the script to finish and then displays the results on the web page. If the script takes some time to run, the web application can display progress information for the user.

Running the R script

The R script is run using R batch mode. The batch command is placed in a shell script.

```
#!/bin/sh
R CMD BATCH -slave -no-restore -no-save $1 $2
```

The shell script is run as a Process using the application's instance of the Runtime class.

```
Runtime r = Runtime.getRuntime();
String cmd = "shell_script " +
            "Rscript.R " + "Rscript.Rout";
Process p = r.exec(cmd);
p.waitFor();
```

The Runtime class allows a Java application to interface with the environment in which it is running. A Java application has one instance of this class. The Process class has a method `waitFor()` that causes the current thread to wait until the Process has completed.

Passing values to the R script

Before the R script is run the values of the variables that the user entered on the web pages are passed to the R script. To do this the application writes the variables and their values, as R assignments, into a text file which is concatenated with the main R script prior to execution:

Numeric entry box If, for example, an R script variable named `my_num` was associated with a Numeric entry box when the application was created with `Rwui`, and the user of the completed application enters the number 1234 into this Numeric entry box, then the line `my_num = 1234` will be added automatically to the beginning of the R script prior to execution.

Text entry box Similarly for a Text entry box; if, for example, an R script variable named `my_text` was associated with a Text entry box and the user of the completed application enters `hello` in this box, then the application will automatically add the line `my_text = "hello"` to the beginning of the R script. Drop-down lists, Radio buttons and Checkboxes work in a similar fashion except the choices available to the user have been pre-defined when the application was created with `Rwui`.

File upload box These are dealt with in a similar fashion. If, for example, an R variable named `my_data_file` was associated with a File upload box when the application was created, then the completed web application will assign the name of the file that the user uploads with this File upload box, to the R variable `my_data_file`. For example, if the user uploads a file named `data.txt`, then the application will automatically add the line `my_data_file = "data.txt"` to the beginning of the R script prior to execution. An important point to note is that the application does not add a line to the script to actually read the file into R. The R script uploaded when the application was created with `Rwui`, must include code to read the data file(s). In this example the R script needs to include a line to read the file whose name is stored in the R variable `my_data_file`, for example, `my_data = scan(file=my_data_file)`. When the file `data.txt` is uploaded from the web page it will be stored in the correct working directory on the server, so no path to the file is required.

Multiple/Replicate File upload box The completed web application will assign to the R variable associated with a Multiple/Replicate File upload page, in the form of a list, the names of the files that the user uploads with this page. For example, if the user uploads three

files called, `replicate1.txt`, `replicate2.txt` and `replicate3.txt` and the R variable associated with the page is `my_data_replicates` then the application will automatically add the line `my_data_replicates = c("replicate1.txt", "replicate2.txt", "replicate3.txt")` to the beginning of the R script prior to execution. But, as explained in the previous paragraph about the File upload box, the application does not add any lines to the script to actually read the files into R. In this example, the R script, uploaded when the application was created with `Rwui`, needs to include code to read the files whose names are stored in the variable `my_data_replicates`.

Multiple/Replicate File upload pages may optionally contain a text box for specifying the group that each of the multiple/replicate files belongs to. If the page includes a group text box then the completed application will automatically add a further line to the beginning of the R script prior to execution. This line assigns to a variable named 'groups', in the form of a list, the group names that the user enters in the box. For example, if the user entered `control` in the text box when submitting `replicate1.txt`, `diseased` when submitting `replicate2.txt` and `diseased` when submitting `replicate3.txt`, then the application will automatically add the line `groups = c("control", "diseased", "diseased")` to the beginning of the R script.

Displaying the results

Each time the analyse button is pressed a uniquely named working directory is created on the server. In order to pass the results generated by the R script back to the web page, the R script writes the results to files in this directory. On completion of the R script the web application looks at what files have been produced and lists them on the Results page as links, which the user can click on to view and/or download. The application also uploads any data files for processing into this directory which means that these files will also be listed as links on the Results page, giving the user the opportunity to check that the correct data has been submitted and has been uploaded correctly.

Optionally the contents of some or all of the results files can be displayed on the Results page straight away, to save the user the trouble of having to click on a link. The names of the results files that are to be displayed are pre-defined when the application is created with `Rwui`. Files which are to be displayed in this way must end in either `.txt` or, for images, `.png` or `.jpg`. Internet Explorer won't display `png`'s, so `jpg`'s will be necessary if the completed application may be accessed with Internet Explorer.

SessionListener

Every time a new analysis is performed, any data files will be copied to the new working directory, even if the data files have not changed between analyses. If the user presses the 'Clear Page' button then their current working directory and all the working directories from their previous submissions are deleted from the server. But if the user exits the application by closing the browser window without pressing the 'Clear Page' button, then these directories and their contents will remain on the server. To prevent the server from becoming clogged with data, the completed web application includes a SessionListener. The SessionListener detects when a session is about to expire. This usually occurs 30 minutes from when the session was last accessed, but the session timeout can be changed in the Tomcat server configuration if required (in the 'Default Session Configuration' section of the file '*tomcat.home/conf/web.xml*'). On detecting a session expiring the SessionListener removes all the working directories created during the session from the server. The SessionListener can be turned off by commenting out the relevant section in the file '*application.home/web/WEB-INF/web.xml*'.

Progress information

Optionally the completed application can display progress information for the user while the R script is running. If progress information is required then the R script, uploaded when the application was created with Rwui, must append information to a text file, called 'process_info.txt', at stages during its execution (for example: `capture.output(expr = rma(data), append=TRUE, file="process_info.txt")`). This text file is displayed for the user by a JavaScript pop-up window which refreshes at fixed intervals. This feature will only work if the user's browser has JavaScript enabled.

JavaScript is also used to disable the analyse button while the R script is running, in order to prevent multiple submissions. However applications created by Rwui also use a synchronizer token based method (Guay, 2003) to ensure multiple submissions are not possible, even when JavaScript is turned off.

Validation and error messages

The web application validates data as it is entered by the user and if necessary, returns error messages to the web page. Errors that occur whilst the R program is running are also displayed. The R options command is automatically set at the beginning of the script so that if an error occurs the error message is written to a text file. The web application checks this file when the Process terminates and if it is not

empty, displays the message on the web page.

R script Requirements

An R script requires a few simple modifications in order to be run by a web application created by Rwui:

- The input variables of the R script must be named according to the rules of both R and Java variable naming. The input variables are those variables that require a value supplied by the user, so correspond to an input item on the application's web page. Their names may only contain letters from the alphabet, \$ and underscore. The names must start with a letter of the alphabet, which needs to be lower-case. If the second character in the variable name is a letter then it must also be lower-case.

(If the input variables of an existing R script are incorrectly named then, rather than change them all, it may be simpler to add reassignments at the beginning of the script. For example, if an R script requires a value for a variable named MY.input.variable then, when creating the application with Rwui, associate `my_input_variable`, rather than the incorrectly named `MY.input.variable`, with the input item on the web page. Then add the line `MY.input.variable = my_input_variable` at the beginning of the R script, prior to uploading the script to Rwui).

Those variables in the R script that are not input variables do not need to conform to the rules of Java variable naming.

- The R script must not set the working directory.
- Any results to be made available to the user must be written to files. If the files are to be displayed, rather than just downloaded from the Results page, they must be suffixed with either .txt or, for images, .png or .jpg.
- If progress information is to be displayed, the R script needs to periodically append lines to a file ('process_info.txt') with messages for the user.

Bibliography

R. C. Gentleman, V. J. Carey, D. M. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, J. Gentry, K. Hornik, T. Hothorn, W. Huber, S. Iacus, R. Irizarry, F. Leisch, C. Li, M. Maechler, A. J. Rossini, G. Sawitzki, C. Smith, G. Smyth, L. Tierney, J. Y. H. Yang, and

- J. Zhang. Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology*, 5:R80, 2004. URL <http://genomebiology.com/2004/5/10/R80>
- R. Guay. Protect web application control flow, 2003. URL http://www.javaworld.com/javatips/jw-javatip136_p.html/

School of Crystallography
Birkbeck College
University of London
`r.newton@mail.cryst.bbk.ac.uk`
`l.wernisch@mail.cryst.bbk.ac.uk`