Rwui: A Web Application to Create User Friendly Web Interfaces for R Scripts

by Richard Newton and Lorenz Wernisch

Summary

The web application Rwui is used to create web interfaces for running R scripts. All the code 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.

Rwui is aimed at R script writers who have scripts that they want people unversed in R to use. The script writer uses Rwui to create a web application that will run their R script. Rwui allows the script writer to do this without them having to do any web application programming, because Rwui generates all the code for them.

The script writer designs the web application to run their R script by entering information on a sequence of web pages. The script writer then downloads the application they have created and installs it on their own server. This is a simple matter of copying one file from the download onto the server. The script writer now has a web application on their server that runs their R script, that other people can use over the web.

Although of general applicability, Rwui was 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, who come from a non-bioinformatics background, are unfamiliar with the language. One solution to this problem is for developers of R scripts to provide user-friendly web interfaces for their scripts.

Rwui (R Web User Interface) is a web application that the developer of an R script can use to create a web application for running their script. All the code for the web application is generated automatically. This is the key feature of Rwui and means that it only takes a few minutes for someone who is entirely unfamiliar with web application programming to design, download and install on their server a fully functional web interface for an R script.

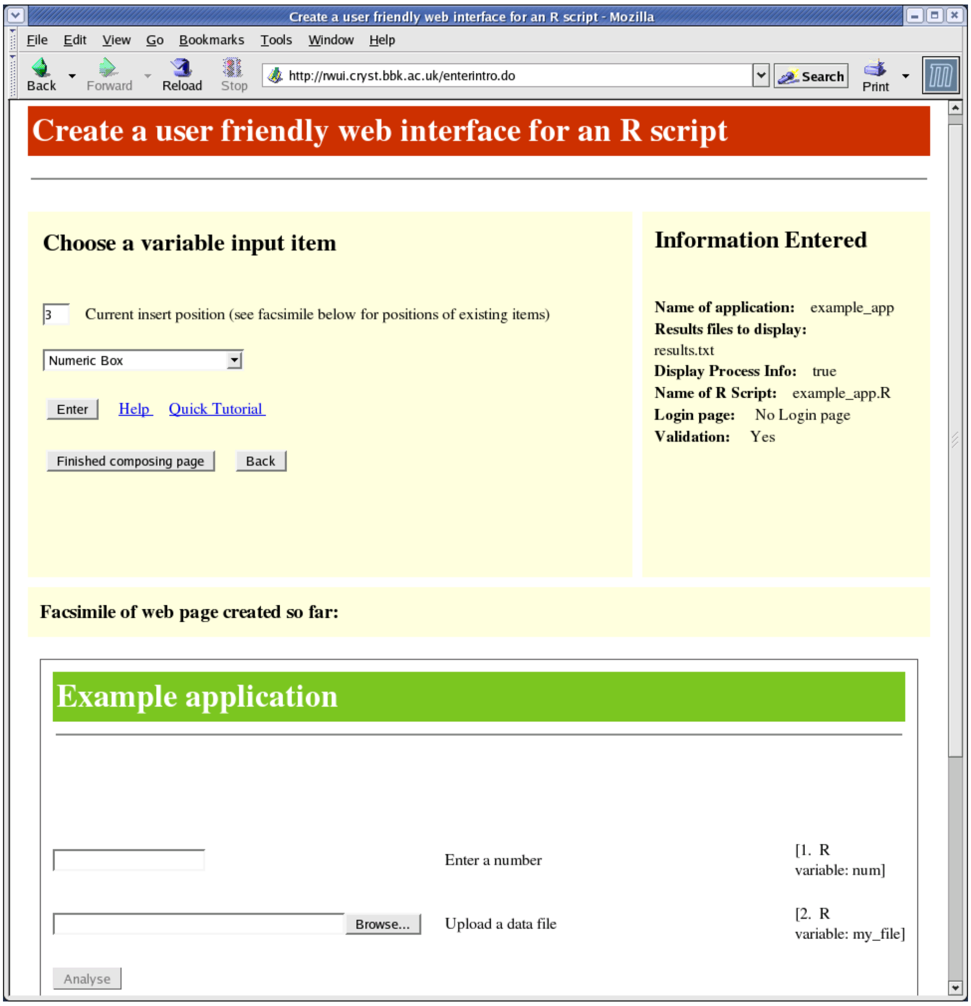
A web interface for an R script means that the script can be used by anyone, even if they have no knowledge of R. Instead of using the R script directly, values for variables and data files for processing are first submitted by the user on a web form. The application then runs the R script on a server, out of sight of the user, and returns the results of the analysis to the user's web page. Because the web application runs on a server it can be accessed remotely and 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.

Although of general applicability, Rwui has been designed with bioinformatics applications in mind. To this end the web applications created by Rwui can include features often required in bioinformatics data analysis, such as a page for uploading replicate data files and their group identifiers. Rwui is typically aimed at bioinformaticians who have developed an R script for analysing experimental data and who want to make their method immediately accessible to collaborators who are unfamiliar with R. Rwui enables bioinformaticians to do this quickly and simply, without having to concern themselves with any aspects of web application programming.

The completed web applications run on Tomcat servers (http://tomcat.apache.org/). Tomcat is free and widely used server software, very easy to install on both Unix and Windows machines, and with an impeccable security record. There have been no known instances of the safety of data on Tomcat servers being compromised despite its widespread use. But if data cannot be sent over the internet then either the web applications created by Rwui can be installed on a local Tomcat server for internal use, situated behind a firewall, or Tomcat and the web applications created by Rwui can be installed on individual stand-alone machines, in which case the web applications are accessed in a browser on the machine via the 'localhost' URL. Instructions on installing and running Tomcat can be accessed from a link on the 'Help' page of Rwui.

A number of approaches to providing web interfaces to R scripts exist. A list with links can be found on the R website (http://cran.r-project.org/doc/FAQ/R-FAQ.html) in the section 'R Web Interfaces'. These fall into two main categories. Firstly there are projects which allow the user to submit R code to a remote server. With these methods, the end user must handle R code. In contrast, users of applications created by Rwui have no contact with R code.

Secondly there are projects which facilitate pro-


Create a user friendly web interface for an R script - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://rwui.cryst.bbk.ac.uk/enterintro.do
Search
Print

Create a user friendly web interface for an R script

Choose a variable input item

3 Current insert position (see facsimile below for positions of existing items)

Numeric Box

Enter Help Quick Tutorial

Finished composing page Back

Information Entered

Name of application: example_app
Results files to display: results.txt
Display Process Info: true
Name of R Script: example_app.R
Login page: No Login page
Validation: Yes

Facsimile of web page created so far:

Example application

Enter a number [1. R variable: num]

Browse... Upload a data file [2. R variable: my_file]

Analyse

Figure 1: Screenshot showing the web page of Rwui where input items are added.

gramming web applications that run R code. With these methods, whoever is creating the application must write web application code. In contrast, when using Rwui no web application code needs to be written; a web interface for an R script is created simply by selecting options from a sequence of web pages.

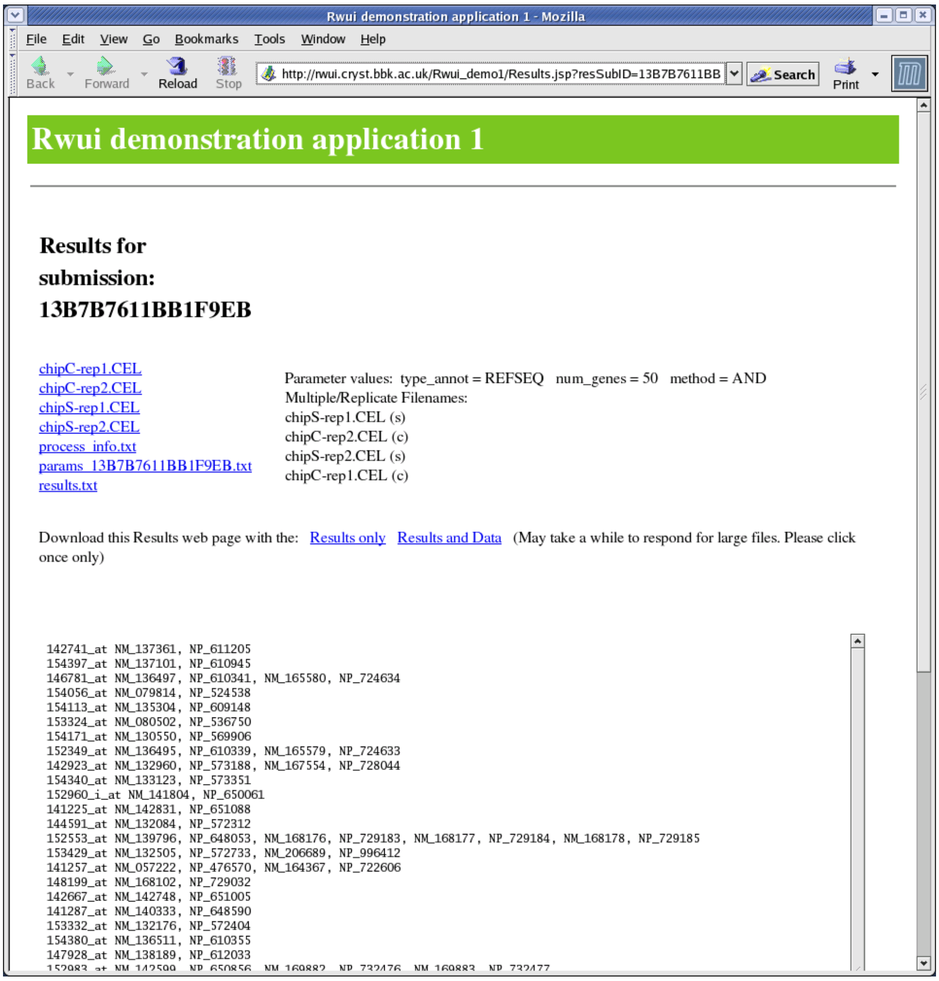
Using the Application

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 input items that will appear on the application's web page are selected. 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 in the script that require a value supplied by the user, must have a corresponding input item on the application's web page. Figure 1 shows the web page of Rwui on which the input items that will appear in the application are added.

Section headings can also be added if required. Rwui displays a facsimile of the web page that has been created as items are added to the page. This can be seen in the lower part of the screenshot shown in Figure 1. 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.

Rwui creates Java based applications that use the Apache Struts framework (http://struts.apache.org/). Struts is open source and a popular framework for constructing well-organised, stable and extensible web applications.

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 application's 'web application archive' file, which is contained in the download, into a subdirectory of the Tomcat server. In addition, the file permissions of an included shell script must be changed to executable.


Rwui demonstration application 1 - Mozilla
File
Edit
View
Go
Bookmarks
Tools
Window
Help
Back
Forward
Reload
Stop
http://rwui.cryst.bbk.ac.uk/Rwui_demo1/Results.jsp?resSubID=13B7B7611BB
Search
Print

Rwui demonstration application 1

Results for submission: 13B7B7611BB1F9EB

chipC-rep1.CEL
chipC-rep2.CEL
chipS-rep1.CEL
chipS-rep2.CEL
process_info.txt
params_13B7B7611BB1F9EB.txt
results.txt

Parameter values: type_annot = REFSEQ num_genes = 50 method = AND
Multiple/Replicate Filenames:
chipS-rep1.CEL (s)
chipC-rep2.CEL (c)
chipS-rep2.CEL (s)
chipC-rep1.CEL (c)

Download this Results web page with the: Results only Results and Data (May take a while to respond for large files. Please click once only)

```
142741_at NM_137361, NP_611205
154397_at NM_137101, NP_610945
146781_at NM_136497, NP_610341, NM_165580, NP_724634
154056_at NM_079814, NP_524538
154113_at NM_135304, NP_609148
153324_at NM_080502, NP_536750
154171_at NM_130550, NP_569906
152349_at NM_136495, NP_610339, NM_165579, NP_724633
142923_at NM_132960, NP_573188, NM_167554, NP_728044
154340_at NM_133123, NP_573351
152960_i_at NM_141804, NP_650061
141225_at NM_142831, NP_651088
144591_at NM_132084, NP_572312
152553_at NM_139796, NP_648053, NM_168176, NP_729183, NM_168177, NP_729184, NM_168178, NP_729185
153429_at NM_132505, NP_572733, NM_206689, NP_996412
141257_at NM_057222, NP_476570, NM_164367, NP_722606
148199_at NM_168102, NP_729032
142667_at NM_142748, NP_651005
141287_at NM_140333, NP_648590
153332_at NM_132176, NP_572404
154380_at NM_136511, NP_610355
147928_at NM_138189, NP_612033
152983_at NM_142599, NP_650856, NM_169882, NP_732476, NM_169883, NP_732477
```

Figure 2: Screenshot showing an example of a results web page of an application created using Rwui.

An application description file, written in XML, 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 into Rwui by uploading the application description file. The application can then be edited and rebuilt within Rwui.

Further information on using Rwui can be found from links on the application's web pages. The 'Quick Tour' provides a ten minute introductory example of using Rwui. The 'Technical Report' gives a technical overview of the application. The 'Help' link accesses the manual for Rwui which contains detailed information for users.

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). Although a server running a Unix operating system is preferable, the applications will work without modification on a Tomcat server running Windows XP.

R script Requirements

An R script needs little or no modification in order to be run from a web application created by Rwui. There are three areas where the script may require attention.

The R script receives input from the user via R variables, which we term the input variables of the script. The values of input variables are entered by the user on the web page of the application. The input variables must be named according to the rules of both R and Java variable naming. These rules are given on the 'Help' page of Rwui. But those variables in the R script that are not input variables do not need to conform to the rules of Java variable naming.

Secondly, in order to make the results of the analysis available to the user, the R script must save the results to files. Thirdly, and optionally, the R script may also write progress information into a file which can be continously displayed for the user in a Javascript pop-up window.

Using applications created by Rwui

A demonstration application created by Rwui can be accessed from 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 an application created by Rwui includes a Multiple/Replicate File upload page, 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 is not included, the application consists of this second web page only.

Before running the R script 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. During the analysis, progress information can be displayed for the user. To enable this the R script must append information to a text file at stages during its execution. This text file is displayed for the user in a JavaScript pop-up window which refreshes at fixed intervals.

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. Figure 2 shows an example of a results page.

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. Any uploaded data files are also linked to on the Results page, giving the user the opportunity to check that the correct data has been submitted and has been uploaded correctly.

The applications include a SessionListener which detects when a user′s session is about to expire. The SessionListener then removes all the working directories created during the session from the server, to prevent it from becoming clogged with data. By default a session expires 30 minutes after it was last accessed by the user, but the delay can be changed in the server configuration.

All the code of a complete demonstration application created by Rwui can be downloaded from a link on the 'Help' page of Rwui.

Acknowledgment

RN was funded as part of a Wellcome Trust Functional Genomics programme grant.

Bibliography

Gentleman,R.C., Carey,V.J., Bates,D.M., Bolstad,B., Dettling,M., Dudoit,S., Ellis,B., Gautier,L., Ge,Y., Gentry,J., Hornik,K., Hothorn,T., Huber,W., Iacus,S., Irizarry,R., Leisch,F., Li,C., Maechler,M., Rossini,A.J., Sawitzki,G., Smith,C., Smyth,G., Tierney,L., Yang,J.Y.H., Zhang,J. (2004) Bioconductor: Open software development for computational biology and bioinformatics, *Genome Biology*, **5**, R80.

Richard Newton
School of Crystallography
Birkbeck College, University of London, UK
r.newton@mail.cryst.bbk.ac.uk
Lorenz Wernisch
MRC Biostatistics Unit, Cambridge, UK
lorenz.wernisch@mrc-bsu.cam.ac.uk

The np Package: Kernel Methods for Categorical and Continuous Data

by Tristen Hayfield and Jeffrey S. Racine

Readers of R News are certainly aware of a number of nonparametric kernel[1] methods that exist in R base (e.g., `density`) and in certain R packages (e.g., `locpoly` in the **KernSmooth** package). Such functionality allows R users to nonparametrically model a density or to conduct nonparametric local polynomial regression, to name but two applications of kernel methods. Nonparametric kernel approaches are appealing to applied researchers because they often reveal features in the data that might be missed by classical parametric methods. However, traditional nonparametric kernel methods presume that the underlying data is continuous, which is frequently not the case.

[1] A 'kernel' is simply a weighting function.