BIOINFORMATICS APPLICATIONS NOTE
Vol. 27 no. 16 2011, pages 2309–2310
doi:10.1093/bioinformatics/btr364

Systems biology
Advance Access publication June 17, 2011

Menu-driven cloud computing and resource sharing for R and

Hamid Bolouri*,†, Rajiv Dulepet‡ and Michael Angerman
Division of Biology, California Institute of Technology, Pasadena, CA 91125, USA
Associate Editor: Alex Bateman

ABSTRACT

Summary: We report CRdata.org, a cloud-based, free, open-source web server for running analyses and sharing data and R scripts with others. In addition to using the free, public service, CRdata users can launch their own private Amazon Elastic Computing Cloud (EC2) nodes and store private data and scripts on Amazon's Simple Storage Service (S3) with user-controlled access rights. All CRdata services are provided via point-and-click menus.

Availability and Implementation: CRdata is open-source and free under the permissive MIT License (opensource.org/licenses/mit-license.php). The source code is in Ruby (ruby-lang.org/en/) and available at: github.com/seerdata/crdata.

Contact: hbolouri@fhcrc.org

Received on April 4, 2011; revised on June 2, 2011; accepted on June 8, 2011

1 INTRODUCTION

High-throughput technologies and integrative systems biology have led to an increasing need for high-performance computing. Cloud computing has emerged as an attractive solution to issues of maintenance, administration and obsolescence (Stein, 2010).

The Bioconductor (bioconductor.org) and R (cran.r-project.org) projects offer a rich, open-source computational environment with over 3000 'packages' (high-level libraries) covering data analysis (e.g. sequencing, ChIP/RNA-seq), simulation modeling (e.g. stochastic modeling, ODEs and PDEs) and network integration, analysis and visualization. Compatibility with other resources is provided via bridging packages such as RSBML and Rgraphviz.

2 RESULTS

Here, we report the development of a web-based resource (CRdata.org) that addresses three current challenges: First, CRdata provides a means with which people inexpert in R syntax can execute R scripts using a simple web-based graphical user interface. Secondly, to facilitate sharing datasets and scripts, CRdata automatically generates a graphical user interface for submitted scripts. Moreover, users can make data and scripts available to selected collaborators or the world using simple menus. Thirdly, to avoid processing bottlenecks, we provide menu-driven access to Amazon's Elastic Computing Cloud (EC2, aws.amazon.com/ec2/) and its Simple Storage Service (S3, aws.amazon.com/s3/). CRdata users can launch any number of private EC2 processor nodes and/or store their private and shared data and scripts in S3.

In addition to individual use, CRdata users can create groups and share data, scripts and Cloud Computing resources within groups. This functionality enables computational biologists to provide targeted private resources (e.g. customized scripts and analysis results) to collaborators and subscribers. It also enables script users to run multiple analyses using different algorithms and/or parameters. Similarly, authors of systems biology models can provide online versions of their models for interactive exploration.

CRdata users can also send feedback to the owner of a shared file and rate/review the resource to help others. Moreover, the usage history of any file can be explored by users with access rights.

Figure 1 shows example views of CRdata in use. Figure 1A shows the directory listing for an example CRdata group. Each group has an administrator, who can accept/reject user applications to join (arrow). This example group has four members, as shown.

Figure 1B shows the user interface for a script shared within the above group. The dialog boxes (and their default values) are automatically generated by CRdata based on specifications by the script author. As shown, the user is prompted to choose the processing queue (Public, or a user's private queue), specify the input file name and provide algorithmic parameter values (including the choice of output data).

To help users understand the algorithm and choose parameter values, each script is accompanied by an HTML help page (supplied by the script author via CRdata's HTML help file editor). Figure 1C shows a portion of the help file for the script in Figure 1B. Users can also view the script code through a read-only viewer (a portion is shown in Figure 1D).

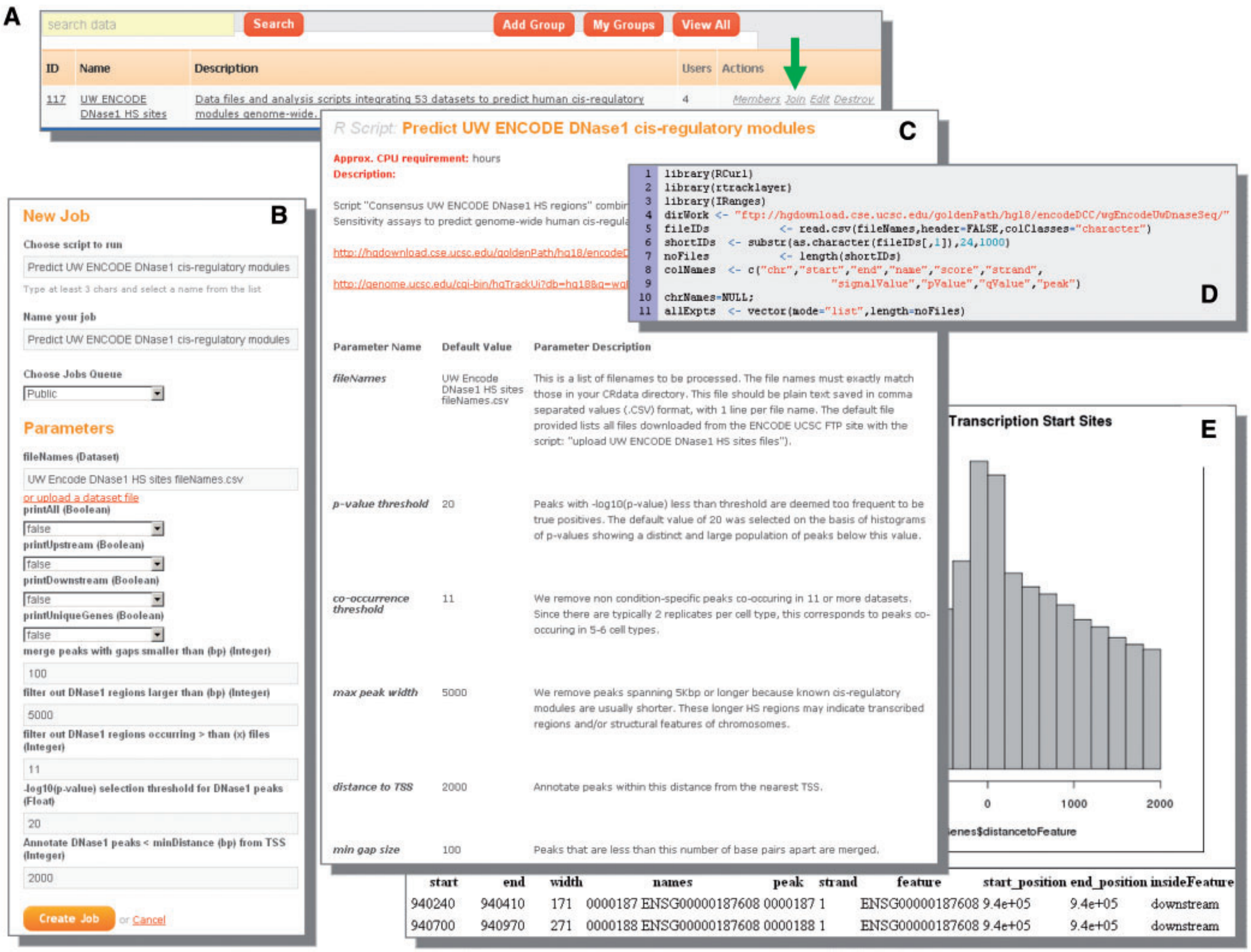
Output of analyses are provided in two forms: data files that can be downloaded or used as inputs to other CRdata scripts, and HTML pages containing text, figures, tables, etc. A portion of the HTML output of the example script is shown in Figure 1E.

To enable CRdata to process output statements, scripts must be annotated with HTML-like tags that declare output statements and their type. The tags are treated as comments by R. For example, a command to output some text is tagged as: #<crdata_text> output text </crdata_text>. CRdata replaces the output declaration tags with HTML commands using R2HTML (tinyurl.com/R2HTML). See URLs 1, 2 and 4 in Supplementary Material for details.

*To whom correspondence should be addressed.

†Present address: D4-100, Div. Human Biology, Fred Hutchinson Cancer Research Center, 1100 Fairview Avenue North, Seattle, WA 98109-1024, USA.

‡Present address: WiseWindow Corporation, 17748 Skypark Circle, Suite 220, Irvine, CA 92614, USA.



ID	Name	Description	Users	Actions
117	UW ENCODE DNase1 HS sites	Data files and analysis scripts integrating 53 datasets to predict human cis-regulatory modules genome-wide.	4	Members Join Edit Destroy

R Script: Predict UW ENCODE DNase1 cis-regulatory modules

Approx. CPU requirement: hours
Description:

Script "Consensus UW ENCODE DNase1 HS regions" combir
Sensitivity assays to predict genome-wide human cis-regula

http://hgdownload.cse.ucsc.edu/goldenPath/hg18/encodeD

http://genome.ucsc.edu/cgi-bin/hgTrackUi?db=hg18&g=wg

```
library(RCurl)
library(rtracklayer)
library(IRanges)
dirWork <- "ftp://hgdownload.cse.ucsc.edu/goldenPath/hg18/encodeDCC/wgEncodeUwDnaseSeq/"
fileIDs          <- read.csv(fileNames,header=FALSE,colClasses="character")
shortIDs  <- substr(as.character(fileIDs[,1]),24,1000)
noFiles          <- length(shortIDs)
colNames  <- c("chr","start","end","name","score","strand",
                    "signalValue","pValue","qValue","peak")
chrNames=NULL;
allExpts  <- vector(mode="list",length=noFiles)
```

Parameter Name	Default Value	Parameter Description
fileNames	UW Encode DNase1 HS sites fileNames.csv	This is a list of filenames to be processed. The file names must exactly match those in your CRdata directory. This file should be plain text saved in comma separated values (.CSV) format, with 1 line per file name. The default file provided lists all files downloaded from the ENCODE UCSC FTP site with the script: "upload UW ENCODE DNase1 HS sites files").
p-value threshold	20	Peaks with -log10(p-value) less than threshold are deemed too frequent to be true positives. The default value of 20 was selected on the basis of histograms of p-values showing a distinct and large population of peaks below this value.
co-occurrence threshold	11	We remove non condition-specific peaks co-occuring in 11 or more datasets. Since there are typically 2 replicates per cell type, this corresponds to peaks co-occuring in 5-6 cell types.
max peak width	5000	We remove peaks spanning 5Kbp or longer because known cis-regulatory modules are usually shorter. These longer HS regions may indicate transcribed regions and/or structural features of chromosomes.
distance to TSS	2000	Annotate peaks within this distance from the nearest TSS.
min gap size	100	Peaks that are less than this number of base pairs apart are merged.

New Job

Choose script to run
Predict UW ENCODE DNase1 cis-regulatory modules
Type at least 3 chars and select a name from the list

Name your job
Predict UW ENCODE DNase1 cis-regulatory modules

Choose Jobs Queue
Public

Parameters

fileNames (Dataset)
UW Encode DNase1 HS sites fileNames.csv
or upload a dataset file
printAll (Boolean)
false
printUpstream (Boolean)
false
printDownstream (Boolean)
false
printUniqueGenes (Boolean)
false
merge peaks with gaps smaller than (bp) (Integer)
100
filter out DNase1 regions larger than (bp) (Integer)
5000
filter out DNase1 regions occurring > than (x) files (Integer)
11
-log10(p-value) selection threshold for DNase1 peaks (Float)
20
Annotate DNase1 peaks < minDistance (bp) from TSS (Integer)
2000

```
 start     end     width   names                          peak  strand  feature          start_position end_position insideFeature
940240    940410    171   0000187 ENSG00000187608 0000187 1        ENSG00000187608 9.4e+05        9.4e+05      downstream
940700    940970    271   0000188 ENSG00000187608 0000188 1        ENSG00000187608 9.4e+05        9.4e+05      downstream
```

Fig. 1. Example views of the CRdata user interface. See text for details.

Over 20 example scripts and associated data files are provided in the CRdata Public space (see Supplementary Material).

CRdata architecture is modular and extensible. A Ruby-on-Rails (http://rubyonrails.org/) server node handles all user, processing node, jobs queue, data and script management, leaving CRdata processing nodes free for data processing. All processing nodes are copies of a Master node which is preloaded with R and Bioconductor libraries and stored as an EC2 processor instance. In this way, nodes can be launched dynamically on demand (typically, initialization of a new EC2 node takes a few minutes).

A Staging Node provides a means to update CRdata nodes without interrupting the operation of the server and its active processing nodes. Apart from packages requiring third-party software or interactive graphics, all packages from R2.12 and Bioconductor2.7 are preinstalled on CRdata nodes. Unpublished packages can be submitted to CRdata using a simple R script.

CRdata supports all Amazon processing node sizes and configurations (see aws.amazon.com/ec2/instance-types/). To minimize running costs, we currently offer two processing node types: small and medium. Three free processing nodes are offered by CRdata to allow users to test scripts and perform small analysis tasks. For jobs likely to take more than a few minutes, we request that users launch private EC2 nodes. A menu-driven interface in CRdata makes this task straightforward. Extra nodes can be launched automatically when there are jobs waiting in a user's queue, and idle nodes terminated.

ACKNOWLEDGEMENTS

CRdata was developed as part of a collaborative project with the laboratory of Dr Ellen Rothenberg (Caltech), with code contributions by SparkTech Software (sparktechsoft.com/), and Bioconductor contributions by Dr Martin Morgan (FHCRC).

Funding: NHLBI grant (HL089102 to H.B.).

Conflict of Interest: none declared.

REFERENCE

Stein,R.D. (2010) The case for cloud computing in genome informatics. *Genome Biol.*, **11**, 207.