Overview of the Glimma package

Shian Su, Matthew E. Ritchie

28 January 2016
Revised 27 February 2016

Contents

1 Introduction 1
2 Getting Started 2
3 Interactive Multidimensional Scaling Plots 3
4 Interactive Mean-Difference Plots 4

1 Introduction

Glimma is a Bioconductor package for interactive visualization of the results from a differential expression analysis of RNA-sequencing data. This functionality is intended to enhance reporting capabilities so that results can be explored more easily by end-users.

Glimma (which loosely stands for Interactive Graphics from the limma package) extends some of the popular plotting capabilities in limma [1] such as multidimensional scaling (MDS) plots, that can be useful for visualising relationships between samples, and mean-difference plots (MD plots) for summarising results from comparisons of interest. The choice of displays and layouts used was inspired by visualisations in the Degust software [2].

The Glimma package is designed to handle RNA-seq differential expression results from the limma, edgeR [3] and DESeq2 [4] packages. Figure 1 below provides an overview of this functionality.

Figure 1: Overview of package workflow.
In this vignette we demonstrate the two main plotting capabilities of this package using a published RNA-seq dataset [5].

## 2 Getting Started

We first perform a basic differential expression analysis on an RNA-seq data set involving Lymphoma cell-lines with either wild-type or null levels of the gene Smchd1 [5]. This experiment was analysed using a limma-voom pipeline [6, 7] that tests for differential expression relative to a particular fold-change using treat [8]. The plots that Glimma extends are shown in Figure 2.

```r
> library(edgeR)
> library(limma)
> library(Glimma)
> data(lymphomaRNAseq)
> x <- lymphomaRNAseq
> class(x)
[1] "DGEList"
attr(,"package")
[1] "edgeR"

> ## Filter out genes with low counts
> sel = rowSums(cpm(x$counts)>0.5)>=3
> x = x[sel,]
> ## Make MDS plot
> genotype = relevel(x$samples$group, "Smchd1-null")
> par(mfrow=c(1,2))
> plotMDS(x, label=1:ncol(x), main="MDS plot", col=as.numeric(genotype))
> legend("topright",legend=c("Smchd1-null","Wild Type"), pch=20, col=1:2, text.col=1:2)
> ## Normalize the data using TMM
> x = calcNormFactors(x, method="TMM")
> ## Set up design matrix
> des = model.matrix(~genotype)
> des

   (Intercept) genotypeWT
1       1         0
2       1         0
3       1         0
4       1         0
5       1         1
6       1         1
7       1         1

attr(,"assign")
[1] 0 1
attr(,"contrasts")
attr(,"contrasts")$genotype
[1] "contr.treatment"

> ## Apply voom with sample quality weights and fit linear model
> v=voomWithQualityWeights(x, design=des, normalization="none", plot=FALSE)
> vfit = lmFit(v,des)
> ## Apply treat relative to a fold-change of 1.5
> vtfit=treat(vfit,lfc=log2(1.5))
> vfit= eBayes(vfit)
```
Figure 2: Examples of static MDS plot (left) and mean-difference plot (right) for the Lymphoma RNA-seq data set. Glimma extends these functions, adding various interactive features to each.

```r
> results = decideTests(vfit, p.value=0.01)
> summary(results)

         (Intercept) genotypeWT
  -1       61       45
  0      2065      12209
  1     10177       49
```

```r
> # Make a mean-difference (MD) plot of the results
> plotMD(vfit, col=2, status=results[,2], hl.col=c("red", "blue"),
>         legend="topright", main="MD plot: Wild-type vs Smcd1")
```

## 3 Interactive Multidimensional Scaling Plots

The first extension display extends plotMDS from limma. Multidimensional scaling is a dimension reduction technique that is popular in gene expression analysis as a check that the samples are well behaved (i.e. whether replicate samples cluster together and to check for the presence of batch effects).

The g1MDSPlot function generates an html page with two panels, with an MDS plot on the left and a on the right and a bar plot of the eigenvalues (a measure of the magnitude of variation explained by each dimension) on the left.

An example from the Lymphoma data set [5] is given below, with Figure 3 showing a screen shot of the output. The interactive nature of this display allows users to hover over points to reveal the sample information for particular points (left panel) and to switch between dimensions using the bar plot in the right panel. The html page generated by g1MDSPlot is intended to make it easier for end users look for relationship between samples in different dimensions.

For further information on the arguments it accepts, see ?g1MDSPlot. By default, the data is saved in a directory named glimma-plots, with the file MDS-Plot.html opening the interactive display shown above. This function works with
Overview of the Glimma package

4 Interactive Mean-Difference Plots

A second extension extends plotMD from limma. The glMDPLOT function generates an html page with two panels and a search bar that allows individual genes to be looked up and highlighted on the summary mean-difference plot which shows average expression and log-fold-change in the first panel alongside the transformed counts in the second. Genes that pass a particular significance threshold can be highlighted on the mean-difference plot.

An example from the Lymphoma data set [5] is given below, with Figure 4 showing a screenshot of the output. The plot on the right is the mean-difference plot with the same genes highlighted in Figure 2. The plot on the right shows the log₂ transformed counts-per-million (CPM) by experimental group for the gene selected. The interactive nature of both plots allows users to hover over points to reveal the values and Gene IDs in the left panel and sample information in the right panel. The html page generated by glMDPLOT is intended to make it easier for end users look up genes of interest in their differential expression results. For further information on the arguments it accepts, see ?glMDPLOT. By default, the data is saved in a directory named glimma-plots, with the file MD-Plot.html opening the interactive display shown above. This function works with output from limma, edgeR and DESeq2.

> glMDPLOT(vfit, counts=x$counts, anno=x$genes, groups=genotype, samples=1:7, status=results[,2], + display.columns=c("Symbols", "GeneID", "GeneName"), folder="Smchd1-Lymphoma", + main="MD plot: Wild-type vs Smchd1", launch=FALSE)

A default version of glMDPLOT accepts a data.frame of results (e.g. an unsorted top table from limma - note that the genes must be in the same order as they appear in the counts and anno objects). This is intended to allow results from any arbitrary differential expression analysis to be handled using this tool. The user can then specify the relevant columns to display in the summary plot. For example the code below makes a volcano plot by choosing the log-odds (B versus the log-fold-change (logFC) columns.

> topt = topTable(vfit, coef=2, number=Inf, sort="none")
> glMDPLOT(topt, xval="logFC", yval="B", counts=x$counts, anno=x$genes, groups=genotype, + samples=1:7, status=results[,2], display.columns=c("Symbols", "GeneID", "GeneName"), + folder="Smchd1-Lymphoma", html="volcano", launch=FALSE)
Overview of the Glimma package

Figure 4: Screen shot of interactive mean-difference (MD) plot generated by the glMDPlot. The gene Smchd1 has been searched for and highlighted in the left-most plot, and it’s expression in sample 1 has been highlighted in the right panel.

Figure 5: Screen shot of interactive volcano plot created using the glMDPlot by providing an unsorted data.frame of results from a differential expression analysis (limma-voom in this case) and selecting the columns to plot via the xval and yval arguments.

References


Overview of the Glimma package


> sessionInfo()

R version 3.3.1 (2016-06-21)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 16.04.1 LTS

locale:
[1] LC_CTYPE=en_US.UTF-8 LC_NUMERIC=C LC_TIME=en_US.UTF-8
[4] LC_COLLATE=C LC_MONETARY=en_US.UTF-8 LC_MESSAGES=en_US.UTF-8
[7] LC_PAPER=en_US.UTF-8 LC_NAME=C LC_ADDRESS=C
[10] LC_TELEPHONE=C LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] stats graphics grDevices utils datasets methods base

other attached packages:
[1] Glimma_1.2.1 edgeR_3.16.2 limma_3.30.2

loaded via a namespace (and not attached):
[1] Rcpp_0.12.7 RColorBrewer_1.1-2 GenomeInfoDb_1.10.1
[4] plyr_1.8.4 XVector_0.14.0 bitops_1.0-6
[7] tools_3.3.1 zlibbioc_1.20.0 digest_0.6.10
[10] rpart_4.1-10 RSQLite_1.0.0 annotate_1.52.0
[13] gtable_0.2.0 htmlTable_1.7 lattice_0.20-34
[16] Matrix_1.2-7.1 DBI_0.5-1 parallel_3.3.1
[19] gridExtra_2.2.1 genefilter_1.56.0 stringr_1.1.0
[22] knitr_1.14 cluster_2.0.5 S4Vectors_0.12.0
[25] IRanges_2.8.1 locfit_1.5-9.1 stats4_3.3.1
[28] grid_3.3.1 nnet_7.3-12 data.table_1.9.6
[31] Biobase_2.34.0 AnnotationDbi_1.36.0 XML_3.98-1.4
[34] survival_2.40-1 BiocParallel_1.8.1 foreign_0.8-67
[37] latticeExtra_0.6-28 Formula_1.2-1 genepattern_1.52.0
[40] DESeq2_1.14.0 ggplot2_2.2.1 magrittr_1.5
[43] htmltools_0.3.5 Hmisc_4.0-0 scales_0.4.0
[46] splines_3.3.1 BiocGenerics_0.20.0 GenomicRanges_1.26.1
[49] SummarizedExperiment_1.4.0 xtable_1.8-2 BiocStyle_2.2.0
[52] colorspace_1.2-7 stringi_1.1.2 acepack_1.4.1
[55] RCurl_1.95-4.8 munsell_0.4.3 chron_2.3-47