DASC user guide

Haidong Yi†, Ayush T. Raman†, Han Zhang, Genevera Allen, Zhandong Liu

16 March 2017

Abstract

Batch effects are one of the major source of technical variations in high throughput studies such as omics profiling. It has been well established that batch effects can be caused by different experimental platforms, laboratory conditions, different sources of samples and personnel differences. These differences can confound the outcomes of interest and lead to spurious results. A critical input for batch correction algorithms are the knowledge of batch factors, which in many cases are unknown or inaccurate. Hence, the primary motivation of our paper is to detect hidden batch factors that can be used in standard techniques to accurately capture the relationship between expression and other modeled variables of interest. Here, we present *DASC*, a novel algorithm that is based on convex clustering and semi-NMF for the detection of unknown batch effects.

Contents

Package: *DASC* **Authors**: Haidong Yi, Ayush T. Raman **Version**: 0.1.1 **Compiled date**: 2017-03-16 **License**: MIT + file LICENSE **Prerequisites**: NMF, cvxclustr, Biobase

1 Getting started

DASC is an R package distributed as part of the Bioconductor project. To install the package, start R and enter:

```
source("http://bioconductor.org/biocLite.R")
biocLite("DASC")
```

Please, make sure to that *NMF*, *cvxclustr* and *Biobase* are installed in your system before running *DASC*. Once *DASC* is installed, it can be loaded to the R enviornment by the following command.

```
library(DASC)
```

2 Introduction

DASC is used for identifying batches and classifying samples into different batches in a high dimensional gene expression dataset. The batch information can be further used as a covariate in conjunction with other variables of interest among standard bioinformatics analysis like differential expression analysis.

2.1 Citation info

If you use *DASC* for your analysis, please cite it as here below. To cite package 'DASC' in publications use:

```
    @Manual{,
    title = {DASC: Detecting hidden batch factors through data adaptive
                adjustment for biological effects.},
    author = {Haidong Yi, Ayush T. Raman, Han Zhang, Genevera I. Allen and
                Zhandong Liu},
    year = {2017},
    note = {R package version 0.1.0},
    }
```

3 Quick Example

```
  library(DASC)
  data("esGolub")
  samples <- c(1,2,20,21,37,38)
  dat <- exprs(esGolub)[1:50,samples]
  pdat <- pData(esGolub)[samples,]
  res <- convexBatch(edata = dat, pdata = pdat, factor = pdat$Cell,
                       method='ama', type = 3, lambda = 1, rank = 2:3,
                       nrun = 50, annotation='esGolub Dataset')
Compute NMF rank= 2  ... + measures ... OK
Compute NMF rank= 3  ... + measures ... OK
```

4 Setting up the data

The first step in using DASC package is to properly format the data. For example, in case of gene expression data, it should be a matrix with features (genes, transcripts) in the rows and samples in the columns. DASC then requires the information for the variable of interest to model the gene expression data effectively. Variable of interest could be a genotype or treatment information.

4.1 Stanford Dataset

Below is an example of Stanford dataset (Chen et. al. PNAS, 2015; Gilad et. al. F1000 Research, 2015). Here, we use the filtered raw counts dataset that was published by Gilad et al. F1000 Research. 30% of genes with the lowest expression & mitochondrial genes were removed (Gilad et al.F1000 Research).

```
    ## libraries
    set.seed(99)
    library(DESeq2)
```

```
library(ggplot2)
library(pcaExplorer)

## dataset
rawCounts <- stanfordData$rawCounts
metadata <- stanfordData$metadata

## Using a smaller dataset
idx <- which(metadata$tissue %in% c("adipose", "adrenal", "pancreas",
                                    "brain", "spleen", "small_bowel"))
rawCounts <- rawCounts[,idx]
metadata <- metadata[idx,]

## raw counts
head(rawCounts)
dim(rawCounts)

## metadata
head(metadata)
dim(metadata)
```

5 Batch detection using PCA Analysis

```
## Normalizing the dataset using DESeq2
dds <- DESeqDataSetFromMatrix(rawCounts, metadata,
                                    design = ~ species + tissue)
dds <- estimateSizeFactors(dds)
dat <- counts(dds, normalized = TRUE)
rld.dds <- rlog(dds) ## this step will take some time
lognormalizedCounts <- log2(dat + 1)

## PCA plot using
pcaplot(rld.dds, intgroup = c("tissue", "species"), ntop = 1000,
            pcX = 1, pcY = 2)
```

In the PCA plot, PC1 shows the differences between the tissues. PC2 shows the differences between the species i.e. samples clustering based on species.

6 Batch detection using DASC

```
res <- convexBatch(edata = dat, pdata = metadata,
                   factor=metadata$tissue, method='ama',
                   type = 3, lambda = 1, rank = 2:10, nrun = 50,
                   annotation='Stanford Dataset')
## Consensus plot
consensusmap(res)
```

```
    ## Residual plot
    plot(res)
```

```
    ## Batches -- dataset has 6 batches
    sample.clust <- data.frame(sample.name = colnames(lognormalizedCounts),
                               clust = as.vector(predict(res$fit$`6`)),
                               batch = metadata$seqBatch)
    ggplot(data = sample.clust, aes(x = c(1:12), y = clust,
            color=factor(clust))) + geom_point(size = 4) +
             xlab("Sample Number") + ylab("Cluster Number")
```

Based on the above plots, we observe that the dataset has 6 batches. This can further be compared with the sequencing platform or `datasets$seqBatch`. The results suggest that differences in platform led to batch effects. Batch number can be used as another covariate, when differential expression analyses using `DESeq2`,`edgeR` or `limma` are performed.

7 More Examples

We are currently working on vignette and will be uploading more examples soon.

8 Session Info

```
    sessionInfo()
R version 3.3.2 (2016-10-31)
Platform: x86_64-apple-darwin13.4.0 (64-bit)
Running under: macOS Sierra 10.12.3

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

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

other attached packages:
 [1] doParallel_1.0.10   iterators_1.0.8     foreach_1.4.3
 [4] DASC_0.1.1          cvxclustr_1.1.1     igraph_1.0.1
 [7] Matrix_1.2-8        NMF_0.20.6          cluster_2.0.5
[10] rngtools_1.2.4      pkgmaker_0.22       registry_0.3
[13] Biobase_2.34.0      BiocGenerics_0.20.0 knitr_1.15.1
[16] BiocStyle_2.3.30

loaded via a namespace (and not attached):
 [1] Rcpp_0.12.9         compiler_3.3.2      RColorBrewer_1.1-2
 [4] plyr_1.8.4          tools_3.3.2         digest_0.6.12
 [7] evaluate_0.10       tibble_1.2-13       gtable_0.2.0
[10] gridBase_0.4-7      lattice_0.20-34     yaml_2.1.14
[13] stringr_1.2.0       rprojroot_1.2       grid_3.3.2
[16] rmarkdown_1.3.9004 ggplot2_2.2.1       reshape2_1.4.2
[19] magrittr_1.5        backports_1.0.5     scales_0.4.1
```

```
[22] codetools_0.2-15   htmltools_0.3.5    assertthat_0.1
[25] xtable_1.8-2       colorspace_1.3-2   stringi_1.1.2
[28] lazyeval_0.2.0     munsell_0.4.3
```