DASC user guide

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

5 March 2017

Abstract

Batch effects are one of the major source of technical variations that affect the measurements 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 is 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

1 Package Details and Pre-requisite

Package: DASC 0.1.0

Authors: Haidong Yi, Ayush T. Raman

Version: 0.1.0

Compiled date: 2017-03-05

License: MIT + file LICENSE

Prerequisites: NMF, cvxclustr, Biobase

2 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")
```

Also, 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")
```

3 Introduction

DASC is used for identifying batches in a 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.

3.1 Citation info

If you use *DASC* for your analysis, please cite it as here below:

To cite package 'DASC' in publications use:

Haidong Yi, Ayush T. Raman, Han Zhang, Genevera I. Allen and Zhandong Liu (2017). **DASC: Detecting hidden batch factors through data adaptive adjustment for biological effects.** R package version 0.1.0.

4 Setting up the data

The first step in using cvxbatch package is to properly format the data. For example, in case of gene expression data, it should be a matrix with features (genes, transcripts, voxels) in the rows and samples in the columns. cvxbatch then requires the information for the variable of interest to model the gene expression data effectively. For example, variable of interest could be a genotype or treatment information. Below is an example of Stanford dataset (Chen et. al. PNAS, 2015; Gilad et. al. F1000 Research, 2015).

```
## Libraries
library("NMF")
library("cvxclustr")

## filtered raw counts data from Gilad et al. F1000 Research
## i.e filtering out lower 30% + mitochondrial genes
data("rawCounts")
data("datasets")

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

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

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

5 Batch detection using PCA Analysis

```
## Normalizing the dataset using DESeq2
library(DESeq2)
library(ggplot2)
dds <- DESeqDataSetFromMatrix(rawCounts, datasets, 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
library(pcaExplorer)
pcaplot(rld.dds, intgroup = c("tissue", "species"), ntop = 1000, pcX = 1,
                pcY = 2, title = "PCA suggesting batch")
```

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

6 Batch detection using DASC

```
res <- convexBatch(edata = dat, pdata = datasets,
                        factor = datasets$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 = datasets$seqBatch)
ggplot(data = sample.clust, aes(x = c(1:12), y = clust, color = factor(clust))) +
         geom_point(size = 3) + 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`, which suggest that difference in platform led to batch effects. Batch number can be used as another covariate, when the differential expression analyses are performed.

7 More Examples

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

8 Session Info

```
sessionInfo()
```