

DASC user guide (PDF)

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

7 May 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.

Package version: DASC 0.99.11

Contents

1	Getting started	1
2	Introduction	1
2.1	Citation info	2
3	Quick Example	2
4	Setting up the data	2
4.1	Stanford RNA-Seq Dataset	2
5	Batch detection using PCA Analysis	3
6	Batch detection using DASC	4
7	Session Info	7

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")
```

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(20,21,28,30)
dat <- exprs(esGolub)[1:100,samples]
pdat <- pData(esGolub)[samples,]

## use nrun = 50 or more for better convergence of results
res <- DASC(edata = dat, pdata = pdat, factor = pdat$Cell, method = 'ama',
  type = 3, lambda = 1, rank = 2:3, nrun = 5,
  annotation="esGolub Dataset")
#consensusmap(res)
#plot(res)
```

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 RNA-Seq Dataset

Below is an example of Stanford gene expression dataset (Chen et. al. PNAS, 2015; Gilad et. al. F1000 Research, 2015). It is a filtered raw counts dataset which 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(99999)
library(DESeq2)
library(ggplot2)
library(pcaExplorer)

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

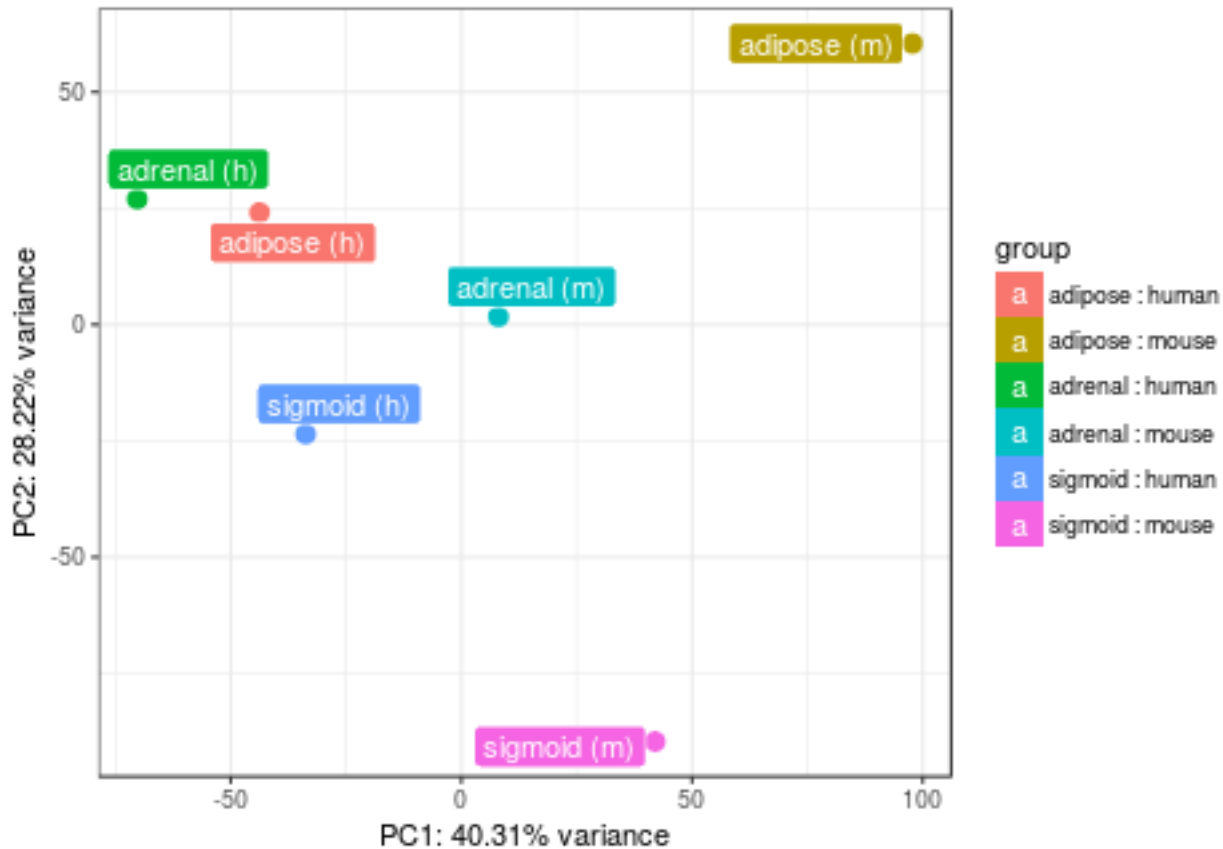
```
## Using a smaller dataset
idx <- which(metadata$tissue %in% c("adipose", "adrenal", "sigmoid"))
rawCounts <- rawCounts[,idx]
metadata <- metadata[idx,]

head(rawCounts)
##      adipose (h) adrenal (h) sigmoid (h) adipose (m) adrenal (m)
## STAG2      1430      4707      4392      3223      8235
## STAG1       835      2362      1687      2750      2732
## GOSR2       142       891       97      1599      1430
## C1orf43     1856     9591     2611       706       498
## ART5         1         4         0         0         0
## ART1         0         0         0         0         1
##      sigmoid (m)
## STAG2      10435
## STAG1      2833
## GOSR2       887
## C1orf43     753
## ART5         0
## ART1         0
head(metadata)
##      setname      seqBatch species tissue
## adipose (h) adipose (h) D87PMJN1:253:D2GUAACXX:8  human adipose
## adrenal (h) adrenal (h) D87PMJN1:253:D2GUAACXX:8  human adrenal
## sigmoid (h) sigmoid (h) D87PMJN1:253:D2GUAACXX:8  human sigmoid
## adipose (m) adipose (m) D4LHBFN1:276:C2HKJACXX:4  mouse adipose
## adrenal (m) adrenal (m) D4LHBFN1:276:C2HKJACXX:4  mouse adrenal
## sigmoid (m) sigmoid (m) D4LHBFN1:276:C2HKJACXX:4  mouse sigmoid
```

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)
lognormalizedCounts <- log2(dat + 1)

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

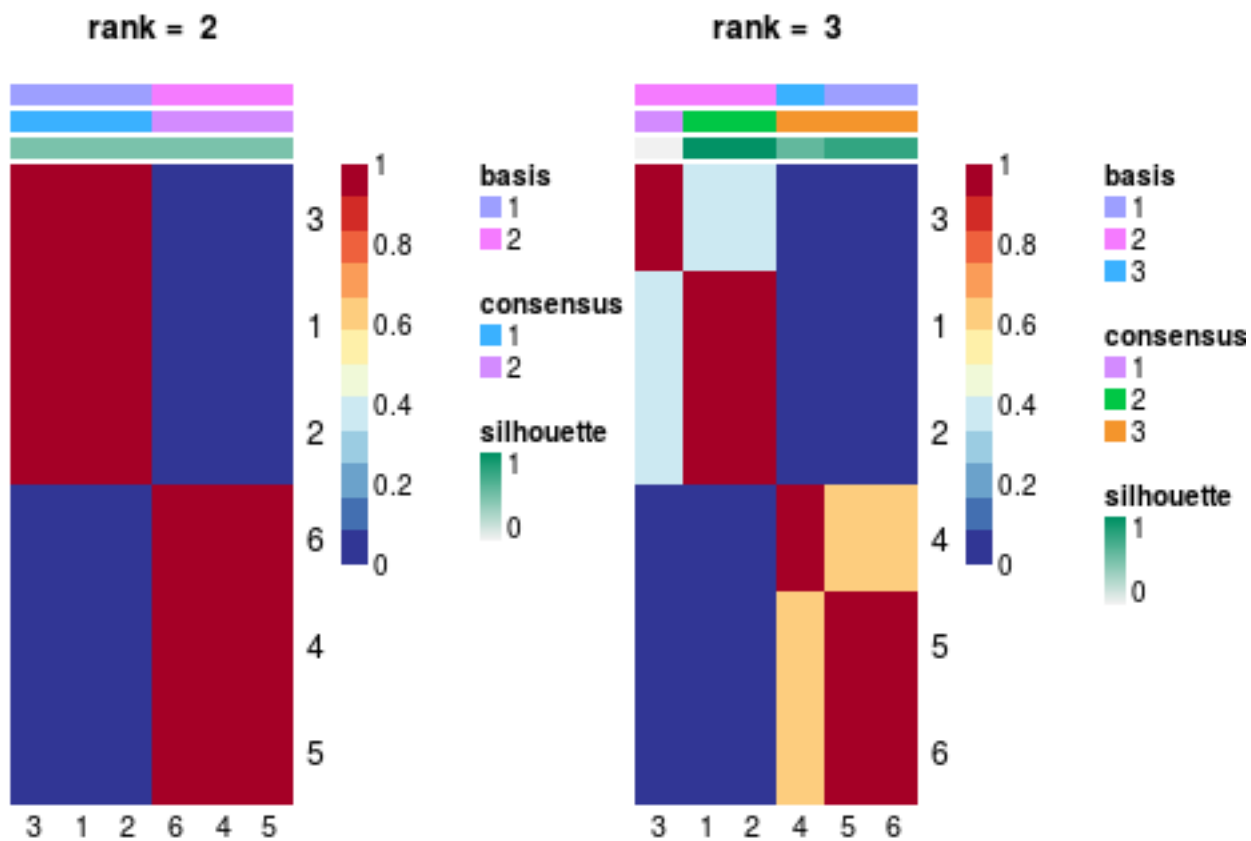


In the PCA plot, PC1 shows the differences between the species. PC2 shows the differences between the species i.e. samples clustering based on tissues.

6 Batch detection using DASC

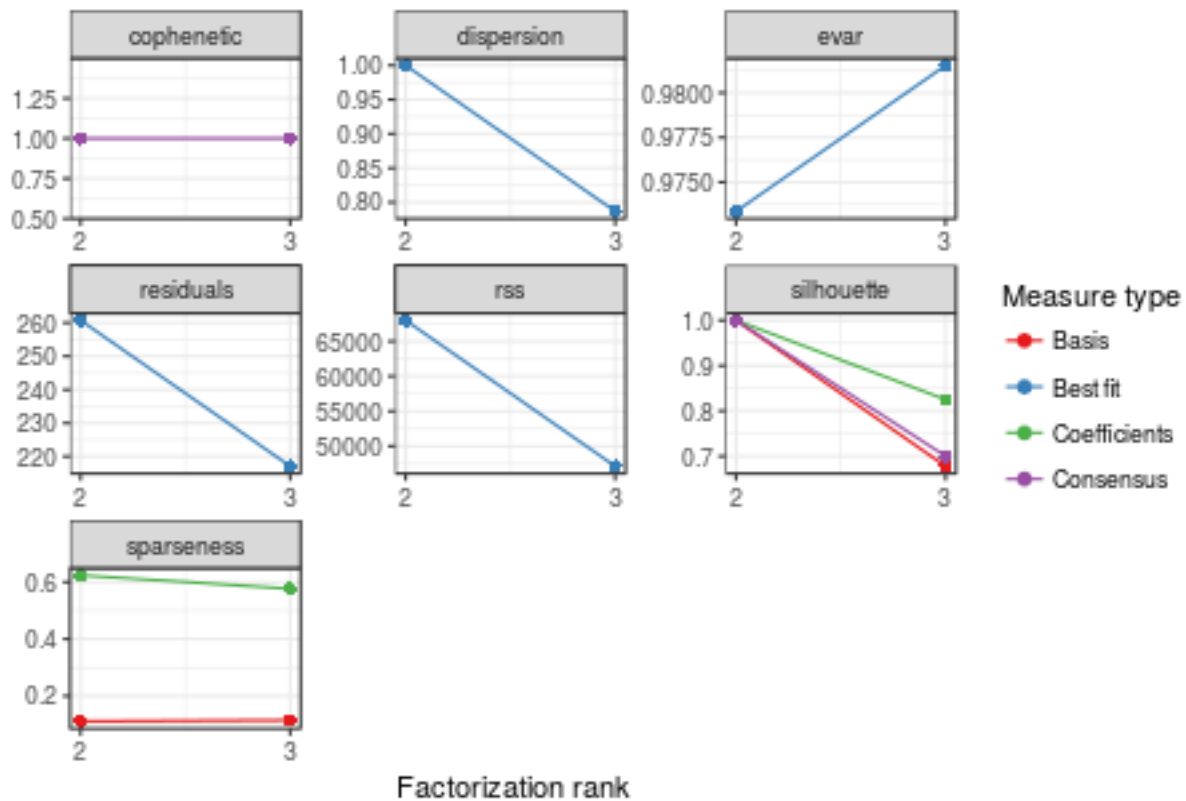
```
res <- DASC(edata = dat, pdata = metadata, factor = metadata$tissue,
            method = 'ama', type = 3, lambda = 1, rank = 2:3, nrun = 10,
            annotation = 'Stanford Dataset')
## Compute NMF rank= 2 ... + measures ... OK
## Compute NMF rank= 3 ... + measures ... OK

## Consensus plot
consensusmap(res)
```

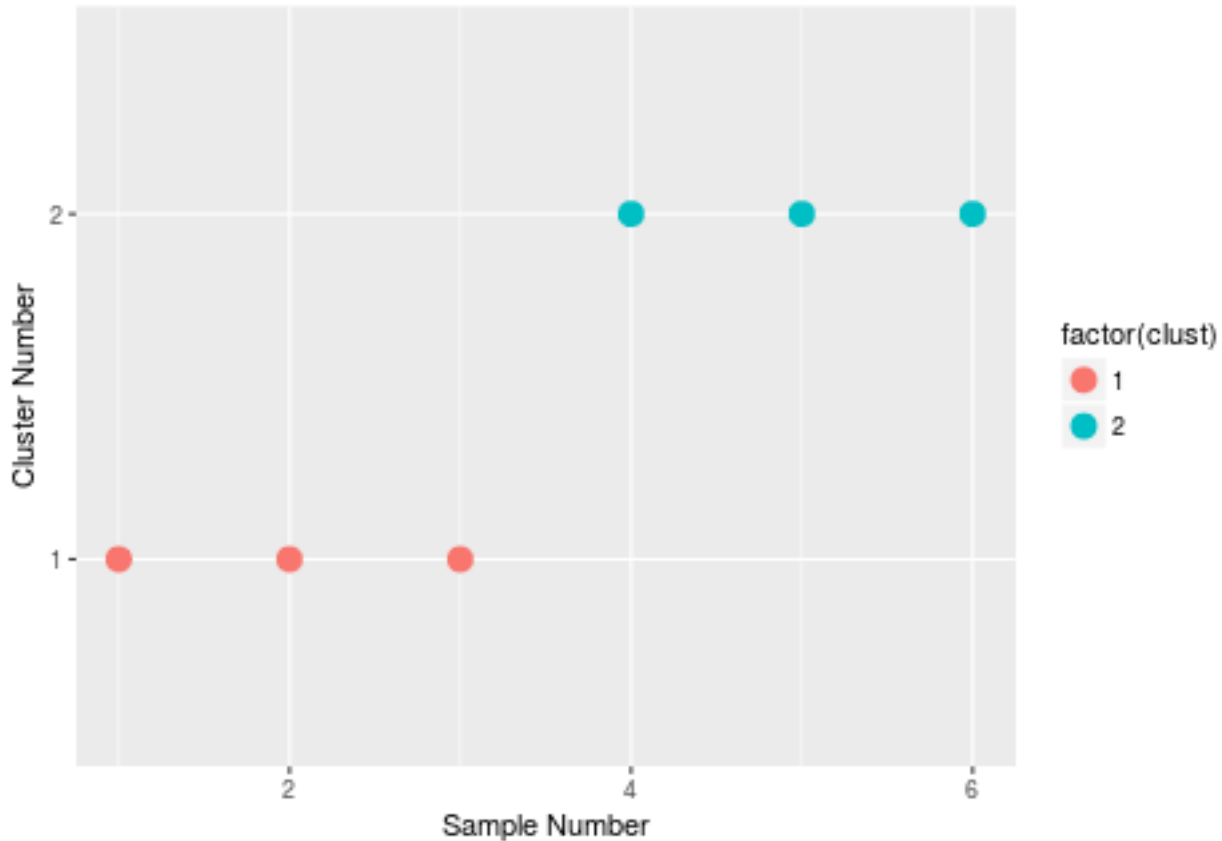


```
## Residual plot
plot(res)
```

NMF rank survey



```
## Batches -- dataset has 6 batches
sample.clust <- data.frame(sample.name = colnames(lognormalizedCounts),
                           clust = as.vector(predict(res$fit$`2`)),
                           batch = metadata$seqBatch)
ggplot(data = sample.clust, aes(x=c(1:6), 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 2 batches. This can further be compared with the sequencing platform or `metadata$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 Session Info

```

sessionInfo()
## R version 3.4.0 (2017-04-21)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 16.04.2 LTS
##
## Matrix products: default
## BLAS: /home/biocbuild/bbs-3.6-bioc/R/lib/libRblas.so
## LAPACK: /home/biocbuild/bbs-3.6-bioc/R/lib/libRlapack.so
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8      LC_COLLATE=C
##  [5] LC_MONETARY=en_US.UTF-8  LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8     LC_NAME=C
##  [9] LC_ADDRESS=C             LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:

```

```

## [1] stats4      parallel  stats      graphics  grDevices  utils      datasets
## [8] methods      base
##
## other attached packages:
## [1] RColorBrewer_1.1-2          pcaExplorer_2.3.0
## [3] ggplot2_2.2.1              DESeq2_1.17.1
## [5] SummarizedExperiment_1.7.2 DelayedArray_0.3.4
## [7] matrixStats_0.52.2         GenomicRanges_1.29.3
## [9] GenomeInfoDb_1.13.1        IRanges_2.11.1
## [11] S4Vectors_0.15.1          doParallel_1.0.10
## [13] iterators_1.0.8           foreach_1.4.3
## [15] DASC_0.99.11              cvxclustr_1.1.1
## [17] igraph_1.0.1              Matrix_1.2-10
## [19] NMF_0.20.6                bigmemory_4.5.19
## [21] bigmemory.sri_0.1.3       cluster_2.0.6
## [23] rngtools_1.2.4            pkgmaker_0.22
## [25] registry_0.3              Biobase_2.37.2
## [27] BiocGenerics_0.23.0       knitr_1.15.1
## [29] BiocStyle_2.5.0
##
## loaded via a namespace (and not attached):
## [1] colorspace_1.3-2          rprojroot_1.2
## [3] htmlTable_1.9             XVector_0.17.0
## [5] base64enc_0.1-3          d3heatmap_0.6.1.1
## [7] topGO_2.29.0             ggrepel_0.6.5
## [9] DT_0.2                    AnnotationDbi_1.39.0
## [11] codetools_0.2-15         splines_3.4.0
## [13] geneplotter_1.55.0       Formula_1.2-1
## [15] jsonlite_1.4             gridBase_0.4-7
## [17] annotate_1.55.0          GO.db_3.4.1
## [19] png_0.1-7                pheatmap_1.0.8
## [21] shinydashboard_0.5.3     graph_1.55.0
## [23] shiny_1.0.3              compiler_3.4.0
## [25] GOstats_2.43.0          backports_1.0.5
## [27] assertthat_0.2.0        lazyeval_0.2.0
## [29] limma_3.33.2            acepack_1.4.1
## [31] htmltools_0.3.6         prettyunits_1.0.2
## [33] tools_3.4.0             gtable_0.2.0
## [35] GenomeInfoDbData_0.99.0 Category_2.43.0
## [37] reshape2_1.4.2          Rcpp_0.12.10
## [39] stringr_1.2.0           mime_0.5
## [41] XML_3.98-1.7            shinyAce_0.2.1
## [43] zlibbioc_1.23.0         scales_0.4.1
## [45] shinyBS_0.61            RBGL_1.53.0
## [47] SparseM_1.77            yaml_2.1.14
## [49] memoise_1.1.0          gridExtra_2.2.1
## [51] biomaRt_2.33.1         rpart_4.1-11
## [53] latticeExtra_0.6-28     stringi_1.1.5
## [55] RSQLite_1.1-2          genefilter_1.59.0
## [57] checkmate_1.8.2        BiocParallel_1.11.1
## [59] bitops_1.0-6           evaluate_0.10
## [61] lattice_0.20-35        labeling_0.3
## [63] htmlwidgets_0.8        GSEABase_1.39.0

```



```
## [65] AnnotationForge_1.19.0  plyr_1.8.4
## [67] magrittr_1.5              bookdown_0.3
## [69] R6_2.2.0                  Hmisc_4.0-3
## [71] DBI_0.6-1                 foreign_0.8-68
## [73] survival_2.41-3          RCurl_1.95-4.8
## [75] nnet_7.3-12              tibble_1.3.0
## [77] rmarkdown_1.5            progress_1.1.2
## [79] locfit_1.5-9.1           grid_3.4.0
## [81] data.table_1.10.4        threejs_0.2.2
## [83] digest_0.6.12            xtable_1.8-2
## [85] tidyr_0.6.2              httpuv_1.3.3
## [87] munsell_0.4.3
```