The SDAMS package

Yuntong Li[1], Chi Wang[2,3]*, Li Chen[2,3]†

[1]Department of Statistics , University of Kentucky,Lexington, KY;

[2]Markey Cancer Center, University of Kentucky, Lexington, KY;

[3]Division of Cancer Biostatistics, Department of Internal Medicine;

liyuntong0704@gmail.com

chi.wang@uky.edu

lichenuky@uky.edu

October 24, 2023

Abstract

This vignette introduces the use of the Bioconductor package SDAMS, which is designed for differential abundance analysis for metabolomics and proteomics data from mass spectrometry and differential expression analysis for single-cell RNA sequencing data. These data may contain a large fraction of zero values and the non-zero part may not be normally distributed. SDAMS considers a two-part semi-parametric model, a logistic regression for the zero proportion and a semi-parametric log-linear model for the non-zero values. A kernel-smoothed likelihood method is proposed to estimate regression coefficients in the two-part model and a likelihood ratio test is constructed for differential abundant/expression analysis.

*to whom correspondence should be addressed

†to whom correspondence should be addressed

Contents

1 Citation 3

2 Quick Start 3

3 Data Input 4
3.1 Create SummarizedExperiment object from csv files 4
3.2 Create SummarizedExperiment object from seperate matrix 6

4 Data Analysis 7
4.1 Proteomics example data 7
4.2 Single-cell RNA sequencing example data 9

5 Theory for SDAMS 10
5.1 A two-part semi-parametric model 10
5.2 Identification of differentially abundant features based on data from mass spectrometry . 11
5.3 Identification of differentially expressed genes based on single-cell RNA sequencing data 11

6 Session Info 12

1 Citation

The package SDAMS implements statistical methods from the following publication. If you use SDAMS in the published research, please cite:
Li, Y., Fan, T.W., Lane, A.N. et al. SDA: a semi-parametric differential abundance analysis method for metabolomics and proteomics data. BMC Bioinformatics 20, 501 (2019).
Yuntong Li, Chi Wang and Li Chen: SDAMS: an R Package for differential expression analysis of single-cell RNA sequencing data (Manuscript).

2 Quick Start

This section show the most basic SDAMS work flow for a differential abundance analysis for metabolomics and proteomics data from mass spectrometry or differential expression analysis for single-cell RNA sequencing data:

1. Create a SummarizedExperiment object using function createSEFromMatrix or createSEFromCSV. In this section we use an example SummarizedExperiment object directly, which is an object of SummarizedExperiment class named exampleSumExp contained in this package.
2. Perform a differential abundance analysis or differential expression analysis using SDA.

```
> library("SDAMS")
> data("exampleSumExp")
> results <- SDA(exampleSumExp)
```

Here, the SummarizedExperiment class object exampleSumExp contained in the package is the proteomics dataset, which a matrix-like container for proteomic features with experimental subject grouping information. There are 560 features for 202 experimental subjects with 49 prostate cancer subjects and 153 healthy subjects (0 for healthy control and 1 for patient in this case). This is a 10% subsample of the original dataset. The features are stored as a matrix in the assay slot. Each row in this matrix represents a proteomic feature and each column represents a subject. See Reference [1] for detailed information regarding this dataset.

```
> data("exampleSingleCell")
> results_SC <- SDA(exampleSingleCell)
```

The SDAMS package also provides an example for single-cell RNA sequencing data in a SummarizedExperiment class object exampleSingleCell. There are 92 single cells (48 mouse embryonic stem (ES) cells and 44 mouse embryonic fibroblasts (MEF) cells) that were analyzed. This example data in the form of

TPM (transcripts per kilobase million) values contains 10% of genes which are randomly sampled from the original dataset. See Reference [2] for detailed information regarding this dataset.

3 Data Input

3.1 Create SummarizedExperiment object from csv files

The proteomics or metabolomics data is stored as a matrix with each row being a feature and each column corresponding to a subject. All data in this matrix are non-negative. Another information required is the phenotype covariates. Here we focus on the binary grouping information, for example, numeric 1 for control group and 0 for case group. But it can also be characters, such as "healthy" and "disease". To utilize `SDAMS` package, we should have two separate csv files (for example 'feature.csv' and 'group.csv') as inputs for `createSEfromCSV` to creat a `SummarizedExperiment` object.

Note:

1. The 1^{st} column in 'feature.csv' represents feature names and the 1^{st} row represents subject codes.
2. The 1^{st} column in 'group.csv' represents subject codes, for example, Subject1, Subject2....

The format for "csv files" should look like as Figure 1 and Figure 2:

	9308	9324	9447	9449	9457	9459	9464	9466	9467	9471
54	0	0	116.9700012	0	0	0	0	19.01000023	0	0
94	52.40000153	0	0	139.1199951	82.37000275	0	58.22999954	0	22.32999992	25.09000015
97	0	0	53.45000076	62.72999954	47.15999985	0	0	24.44000053	0	22.77000046
191	0	0	22.18000031	170.6499939	0	0	0	0	0	26.52000046
219	58.47000122	0	0	64.5	28.11000061	0	39.77999878	7.429999828	0	5.230000019
286	0	0	0	19.26000023	89.55999756	0	162.8200073	109.8600006	75.48999786	115.6999969
366	0	0	0	0	28.97999954	0	0	0	0	0
388	29.43000031	0	0	0	0	0	0	32.20000076	0	59.97000122
420	65.79000092	0	87.44000244	314.0100098	160.4400024	0	0	0	0	26.85000038
463	0	0	0	12.15999985	38.52999878	0	0	0	0	0
499	0	0	0	26.12999916	14.22000027	0	0	0	0	1.919999957
583	43.74000168	0	0	59.29000092	7.099999905	0	0	0	0	3.809999943
605	0	0	0	657.8400269	661.9500122	0	0	32.22999954	0	0
624	0	96.94999695	0	97.40000153	167.0599976	232.9499969	89.54000092	55.5	35.15999985	0
652	0	0	9.68999958	0	0	0	0	0	0	3.599999905
709	0	0	38.45000076	0	0	0	0	8.229999542	0	6.380000114
889	0	0	0	0	0	0	0	0	0	0
912	569.3200073	0	0	0	0	0	23.57999992	120.6800003	73.69000244	100.4100037

Figure 1: Example of 'feature.csv' pattern

After creating the two csv files, we need the paths for the two csv files:

	grouping
9308	0
9324	0
9447	0
9449	0
9457	0
9459	0
9464	0
9466	0
9467	0
9471	0
9495	0
9497	0
9507	0
9512	0
9959	1
9960	1

Figure 2: Example of 'group.csv' pattern

```
> path1 <- "/path/to/your/feature.csv/"
> path2 <- "/path/to/your/group.csv/"
```

Here for demonstration purpose, we use the data stored in `inst/extdata` directory. This is the csv format of the data in exampleSumExp which is a `SummarizedExperiment` object we described before.

```
> directory1 <- system.file("extdata", package = "SDAMS", mustWork = TRUE)
> path1 <- file.path(directory1, "ProstateFeature.csv")
> directory2 <- system.file("extdata", package = "SDAMS", mustWork = TRUE)
> path2 <- file.path(directory2, "ProstateGroup.csv")
```

then use the function `createSEFromCSV` after loading the `SDAMS` package.

```
> library("SDAMS")
> exampleSE1 <- createSEFromCSV(path1, path2)
> exampleSE1
```

```
class: SummarizedExperiment
dim: 560 202
metadata(0):
assays(1): counts
rownames(560): 93922 87209 ... 180624 130855
rowData names(0):
```

```
colnames(202): 9512 9963 ... 49341 49586
colData names(1): grouping
```

The feature data and grouping information can be accessed using SummarizedExperiment commands:

```
> head(assay(exampleSE1)[,1:10])
```

```
        9512   9963  9965  9975   9979  9997 10015  10034  10044 10047
93922   0.00   0.00  0.00  0.00   0.00  0.00 68.97   0.00   0.00  0.00
87209   0.00   0.00  0.00  0.00   0.00  0.00  0.00   0.00   0.00 43.87
29633   0.00   0.00  0.00  0.00   0.00  0.00  0.00  57.21   0.00  0.00
40225   0.00 226.40  0.00 19.65   0.00  0.00  0.00   0.00   0.00  0.00
126342  0.00   0.00  0.00  0.00   0.00 20.43  0.00   0.00 109.93  0.00
42832  52.32 137.76 70.25  0.00 453.23 92.20  0.00 352.55 496.71  0.00
```

```
> head(colData(exampleSE1)$grouping)
```

```
[1] 0 1 1 1 1 1
```

3.2 Create SummarizedExperiment object from seperate matrix

If the two datasets have already been cleaned and loaded into R as matrices, then we can use createSEFromMatrix to create a SummarizedExperiment object. Note that groupInfo is the design matrix. The first colomn of this design matrix is the cell subpopulation, and the following columns could be additional covariates.

```
> set.seed(100)
> featureInfo <- matrix(runif(8000, -2, 5), ncol = 40)
> featureInfo[featureInfo<0] <- 0
> rownames(featureInfo) <- paste("gene", 1:200, sep = '')
> colnames(featureInfo) <- paste('cell', 1:40, sep = '')
> groupInfo <- data.frame(grouping=matrix(sample(0:1, 40, replace = TRUE),
+                         ncol = 1))
> rownames(groupInfo) <- colnames(featureInfo)
> exampleSE2 <- createSEFromMatrix(feature = featureInfo, colData = groupInfo)
> exampleSE2
```

```
class: SummarizedExperiment
dim: 200 40
```

```
metadata(0):
assays(1): counts
rownames(200): gene1 gene2 ... gene199 gene200
rowData names(0):
colnames(40): cell1 cell2 ... cell39 cell40
colData names(1): grouping
```

```
> head(assay(exampleSE2)[,1:10])
```

```
          cell1     cell2     cell3    cell4    cell5     cell6     cell7
gene1 0.1543628 0.5871725 1.5786619 0.000000 0.000000 0.0000000 4.8142065
gene2 0.0000000 4.6942599 0.0000000 2.057938 0.000000 0.0000000 3.4021976
gene3 1.8662570 4.3950366 0.5245985 0.000000 1.105383 2.3676080 1.4648599
gene4 0.0000000 3.7633544 1.0626950 0.000000 1.609934 2.6975726 0.7482424
gene5 1.2798450 0.2363751 3.6214671 0.000000 4.474260 0.5612595 0.0000000
gene6 1.3863951 4.1439023 1.6442681 4.393398 0.000000 0.0000000 0.4924940
         cell8     cell9    cell10
gene1 2.857807 0.6332892 1.9930932
gene2 0.000000 0.0000000 1.3506317
gene3 1.655411 4.7851556 0.1505902
gene4 3.693774 0.0000000 4.1987980
gene5 4.413107 2.2631995 1.7927709
gene6 4.624719 0.0000000 0.9549322
```

```
> head(colData(exampleSE2)$grouping)
```

```
[1] 0 1 1 1 0 1
```

```
>
```

4 Data Analysis

4.1 Proteomics example data

Finally, we perform differential abundance analyais using `SummarizedExperiment` object created in previous section. This can be done by using function `SDA`. The theory behind `SDA` can be reached at section 5. A list with point estimates, p-values, q-values and corresponding feature names is returned. Below are the results generated by using the `SummarizedExperiment` object exampleSE1.

```
> results <- SDA(exampleSE1)
> head(results$gamma[,1])
[1]  0.1100009  0.8629447 -0.7151261  0.2876821 -0.1251631  0.6292037
> head(results$pv_gamma[,1])
[1] 0.8290701 0.1703578 0.2356975 0.5571851 0.7227301 0.1248023
> head(results$qv_gamma[,1])
[1] 0.4092097 0.1914385 0.2191259 0.3505466 0.3887992 0.1651784
```

In this example , there is only one group covariate applied to each subject. Here $\mathbf{X}_i$ is one dimension. The covariate effect on the fraction of zero values for certain feature is γ, which is estimated to be 0.11 for the first feature, and 0.86 for the second feature, etc. The corresponding hypothesis is H_0: $\gamma = 0$ vs. H_1: $\gamma \neq 0$. The p-values calculated from likelihood ratio test are returned in `pv_gamma`. Users can determine their own significance level to make inference, such as 0.05 nominal level. We also provide a FDR adjustment method [3] used in `SDA` for multiple comparison issues. Those results for γ are stored in `qv_gamma`.

```
> head(results$beta[,1])
[1] -0.04912170 -1.11354659 -1.30566809  0.02484749  0.53967121 -0.22075205
> head(results$pv_beta[,1])
[1] 0.88988924 0.11334266 0.06081223 0.95500620 0.06770261 0.13832035
> head(results$qv_beta[,1])
[1] 0.7466379 0.2979437 0.2525717 0.7599717 0.2673415 0.3235222
```

The model parameter β is the log fold change in the non-zero abundance comparing different values of the single group covariate for certain feature. The corresponding two-sided hypothesis is H_0: $\beta = 0$ vs. H_1: $\beta \neq 0$. Again, `SDA` will return p-values and adjusted p-values (q-values) for parameter β, and they are stored in `pv_beta` and `qv_beta` respectively.

```
> head(results$pv_2part[,1])
[1] 0.96764635 0.11153821 0.08538208 0.84038731 0.17695215 0.10266642
> head(results$qv_2part[,1])
[1] 0.4622205 0.1380966 0.1192246 0.4323376 0.1806273 0.1345612
```

Hypothesis testing on overall effect of group covariate on certain feature is performed by assessing γ and β. The null hypothesis $H_0 : \gamma = 0$ and $\beta = 0$ against alternative hypothesis H_1 : at least one of the two parameters is non-zero. The p-values are calculated based on chi-square distribution with 2 degrees of freedom. And the corresponding q-values are calculated using the same procedure as in one-part test.

```
> head(results$feat.names)
```

```
[1] "93922"  "29633"  "40225"  "126342" "42832"  "127351"
```

A vector of feature names is returned for convenience which corresponds to the results in the other components.

4.2 Single-cell RNA sequencing example data

SDAMS can also perform differential expression analysis for single-cell RNA sequencing data.

In this section, we will use the example data generated in Section 3.2. This toy data set has 200 genes and 40 cells. The first column of the design matrix is the cell subpopulation, and additional covariates can be added into the design matrix. We are interested in identifying genes that are differentially expressed in two different cell subpopulations, which is quantified by γ_Z and β_Z. The γ_Z is the log odds ratio comparing rate of expression, and β_Z is the log fold change comparing the mean gene expression of the expressed cells between the two cell subpopulations.

```
> results_SC <- SDA(exampleSE2)
> head(results_SC$pv_gamma[,1])
```

```
[1] 0.49647690 0.86968362 0.49647690 0.91895776 0.05805587 0.78112357
```

```
> head(results_SC$qv_gamma[,1])
```

```
[1] 0.9547633 0.9716152 0.9547633 0.9716152 0.6450652 0.9716152
```

```
> head(results_SC$pv_beta[,1])
```

```
[1] 0.4309475 0.5854066 0.8645949 0.6506803 0.7257676 0.7586484
```

```
> head(results_SC$qv_beta[,1])
```

```
[1] 0.9878665 0.9878665 0.9878665 0.9878665 0.9878665 0.9878665
```

```
> head(results_SC$pv_2part[,1])
```

```
[1] 0.5819540 0.8502325 0.7821025 0.8978987 0.1560542 0.9177733
```

```
> head(results_SC$qv_2part[,1])

[1] 0.998765 0.998765 0.998765 0.998765 0.998765 0.998765

> head(results_SC$feat.names)

[1] "gene1" "gene2" "gene3" "gene4" "gene5" "gene6"
```

Three types of hypotheses can be tested through function `SDA`: $H_0^1 : \beta_Z = 0$, $H_0^2 : \gamma_Z = 0$, and $H_0^3 : \gamma_Z = 0 \text{ and } \beta_Z = 0$. Likelihood ratio test is conducted for each test and p-value is returned by `SDA` for each single gene. The corresponding q-values are also calculated to address multiple comparison correction. For Hypothesis H_0^1, the p-value and corresponding q-value for each gene are returned in `pv_beta` and `qv_beta`. For Hypothesis H_0^2, the p-value and corresponding q-value for each gene are returned in `pv_gamma` and `qv_gamma`. For example, the p-value for testing the covariate effect on comparing the proportion of drop-outs for Gene 1 is 0.496, which is not significant under 0.05 level. The corresponding q-value is 0.955. For testing whether there is difference in either the rate of expression or mean expression for the expressed cells, H_0^3, the p-value and corresponding q-value for each gene are returned in `pv_2part` and `qv_2part`.

5 Theory for SDAMS

As mentioned in the abstract, metabolomics and proteomics data from mass spectrometry or single-cell RNA sequencing data maybe a mixture of zero intensity values and possibly non-normally distributed non-zero intensity values. Therefore, the differential abundance/expression analysis needs to be performed to compare both the zero proportion and the mean of non-zero values between groups and also allows adjustment of covariates. SDA is a two-part model which addresses these issues that uses a logistic regression model to characterize the zero proportion and a semiparametric model to characterize non-zero values.

5.1 A two-part semi-parametric model

The differential abundance/expression analysis in SDAMS has the following forms. For binary part:

$$\log(\frac{\pi_i}{1-\pi_i}) = \gamma_0 + \boldsymbol{\gamma}\boldsymbol{X}_i,$$

For continuous non-zero part:

$$\log(Y_i) = \boldsymbol{\beta}\boldsymbol{X}_i + \varepsilon_i,$$

where Y_i $(i = 1, 2, ..., N)$ is a random variable and $\pi_i = Pr(Y_i = 0)$. $\boldsymbol{X}_i$ is a vector of covariates. The corresponding model parameters $\boldsymbol{\gamma}$ quantify the covariates effects on the fraction of zero values and γ_0 is the intercept. $\boldsymbol{\beta}$ are the model parameters quantifying the covariates effects on the non-zero values, and ε_i $(i = 1, 2, ..., N)$ are independent error terms with a common but completely unspecified density function f. Importantly, we do not impose any distributional assumption on f. Without assuming a specific parametric distribution for ε_i, this model is much more flexible to characterize data with unknown and possibly non-normal distribution. We replace f by its kernel density estimator in the likelihood function. The maximum likelihood estimator is obtained through a trust region maximization algorithm.

5.2 Identification of differentially abundant features based on data from mass spectrometry

In this case, Y_i represents the abundance of certain feature for subject i, $\pi_i = Pr(Y_i = 0)$ is the probability of point mass. $\boldsymbol{X}_i = (X_{i1}, X_{i2}, ..., X_{iQ})^T$ is a Q-vector covariates that specifies the treatment conditions applied to subject i. The corresponding Q-vector of model parameters $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, ..., \gamma_Q)^T$ and $\boldsymbol{\beta} = (\beta_1, \beta_2, ..., \beta_Q)$ quantify the covariates effects for certain feature.

For each feature, the likelihood ratio test is performed on the null hypothesis $H_0 : \gamma_q = 0$ and $\beta_q = 0$ against alternative hypothesis H_1 : at least one of the two parameters is non-zero. We also consider the hypotheses for testing $\gamma_q = 0$ and $\beta_q = 0$ separately. To adjust for multiple comparisons across features, the false discovery discovery rate (FDR) q-value is calculated based on the `qvalue` function in `qvalue` package in R/Bioconductor (See Reference [3] for details).

5.3 Identification of differentially expressed genes based on single-cell RNA sequencing data

In this case, Y_i represents the expression (TPM value) of certain gene in the ith cell, $1-\pi_i = Pr(Y_i > 0)$ is the rate of expression. $\boldsymbol{X}_i = (Z_i, \boldsymbol{W}_i)^T$ is a vector of covariates with Z_i being a binary indicator of the cell population under comparison and $\boldsymbol{W}_i$ being a vector of other covariates, e.g. batch and cellular detection rate, and $\boldsymbol{\gamma} = (\gamma_Z, \boldsymbol{\gamma}_W)$ and $\boldsymbol{\beta} = (\beta_Z, \boldsymbol{\beta}_W)$ are model parameters.

We are interested in identifying genes that are differentially expressed in two different cell subpopulations, as quantified by γ_Z and β_Z. For each gene, three null hypotheses, i.e. $\beta_Z = 0$, $\gamma_Z = 0$, and $\beta_Z = 0$ and $\gamma_Z = 0$, are examined based on likelihood ratio tests. Multiple comparison correction is addressed by calculating the false discovery discovery rate (q-values) (See Reference [3] for details).

6 Session Info

```
> toLatex(sessionInfo())
```

- R version 4.3.1 (2023-06-16), x86_64-pc-linux-gnu
- Locale: LC_CTYPE=en_US.UTF-8, LC_NUMERIC=C, LC_TIME=en_GB, LC_COLLATE=C, LC_MONETARY=en_US.UTF-8, LC_MESSAGES=en_US.UTF-8, LC_PAPER=en_US.UTF-8, LC_NAME=C, LC_ADDRESS=C, LC_TELEPHONE=C, LC_MEASUREMENT=en_US.UTF-8, LC_IDENTIFICATION=C
- Time zone: America/New_York
- TZcode source: system (glibc)
- Running under: Ubuntu 22.04.3 LTS
- Matrix products: default
- BLAS: /home/biocbuild/bbs-3.18-bioc/R/lib/libRblas.so
- LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.10.0
- Base packages: base, datasets, grDevices, graphics, methods, stats, stats4, utils
- Other packages: Biobase 2.62.0, BiocGenerics 0.48.0, GenomeInfoDb 1.38.0, GenomicRanges 1.54.0, IRanges 2.36.0, MatrixGenerics 1.14.0, S4Vectors 0.40.0, SDAMS 1.22.0, SummarizedExperiment 1.32.0, matrixStats 1.0.0
- Loaded via a namespace (and not attached): DelayedArray 0.28.0, GenomeInfoDbData 1.2.11, Matrix 1.6-1.1, R6 2.5.1, RCurl 1.98-1.12, Rcpp 1.0.11, S4Arrays 1.2.0, SparseArray 1.2.0, XVector 0.42.0, abind 1.4-5, bitops 1.0-7, cli 3.6.1, colorspace 2.1-0, compiler 4.3.1, crayon 1.5.2, dplyr 1.1.3, fansi 1.0.5, generics 0.1.3, ggplot2 3.4.4, glue 1.6.2, grid 4.3.1, gtable 0.3.4, lattice 0.22-5, lifecycle 1.0.3, magrittr 2.0.3, munsell 0.5.0, pillar 1.9.0, pkgconfig 2.0.3, plyr 1.8.9, qvalue 2.34.0, reshape2 1.4.4, rlang 1.1.1, scales 1.2.1, splines 4.3.1, stringi 1.7.12, stringr 1.5.0, tibble 3.2.1, tidyselect 1.2.0, tools 4.3.1, trust 0.1-8, utf8 1.2.4, vctrs 0.6.4, zlibbioc 1.48.0

References

[1] Justyna Siwy, William Mullen, Igor Golovko, Julia Franke, and Petra Zürbig. Human urinary peptide database for multiple disease biomarker discovery. *PROTEOMICS-Clinical Applications*, 5(5-6):367–374, 2011.

[2] Saiful Islam, Una Kjällquist, Annalena Moliner, Pawel Zajac, Jian-Bing Fan, Peter Lönnerberg, and Sten Linnarsson. Characterization of the single-cell transcriptional landscape by highly multiplex

rna-seq. *Genome research*, 21(7):1160–1167, 2011.

[3] John D Storey and Robert Tibshirani. Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciences*, 100(16):9440–9445, 2003.