

Tutorial for R package *scDeconv*

Yu Liu

2022-09-03

Contents

Introduction	1
Package overview	1
Citation	2
Installation	3
Data preparation	3
RNA reference generation and bulk RNA data deconvolution	5
Bulk DNAm data deconvolution with RNA reference	8
Session information	11

Introduction

Many DNA methylation (DNAm) data are from tissues composed of various cell types; hence, cell deconvolution methods are needed to infer their cell compositions accurately. However, a bottleneck for DNAm data is the lack of cell-type-specific DNAm references. On the other hand, scRNA-seq data are being accumulated rapidly with various cell type transcriptomic signatures characterized, and also, many paired bulk RNA-DNAm data are publicly available currently. Hence, I developed the R package *scDeconv* to use these resources to solve the reference deficiency problem of DNAm data and deconvolve them from scRNA-seq data in a trans-omics manner.

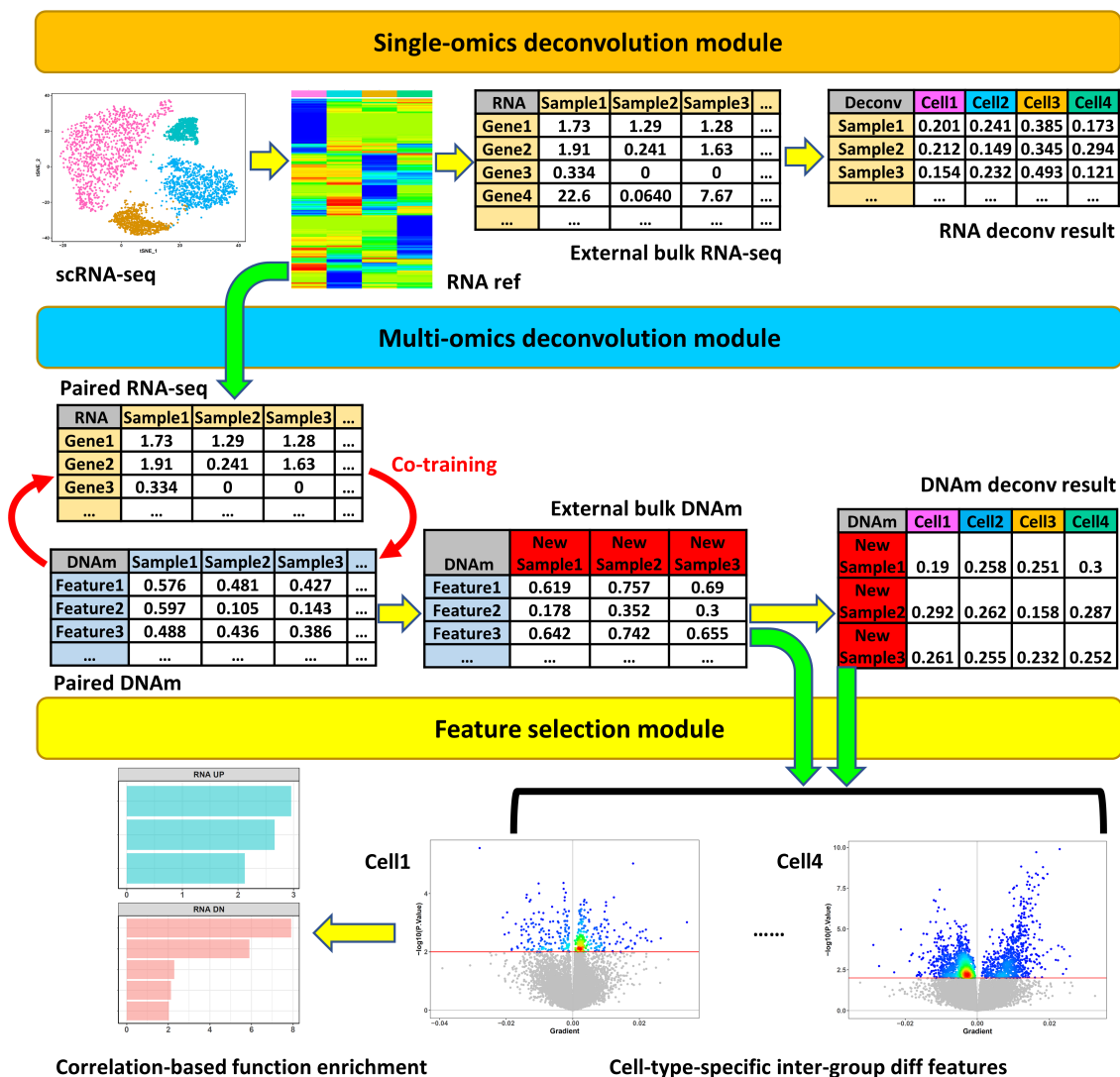
It assumes that paired samples have similar cell compositions. So the cell content information deconvolved from the scRNA-seq and paired RNA data can be transferred to the paired DNAm samples. Then an ensemble model is trained to fit these cell contents with DNAm features and adjust the paired RNA deconvolution in a co-training manner. Finally, the model can be used on other bulk DNAm data to predict their relative cell type abundances. If given an appropriate paired dataset, *scDeconv* can also deconvolve other omics, such as ATAC-seq data. Furthermore, the package also contains other functions, such as identifying cell-type-specific inter-group differential features from bulk DNAm data.

Package overview

The package has three modules. The first is a single-omics deconvolution module. Its function `scRef` can construct cell transcriptomic references from scRNA-seq data. Next, the reference is transferred to the function `refDeconv`. It performs a recursion process to solve a constrained linear model to deconvolve bulk RNA data with the RNA reference.

The second module is a multi-omics deconvolution module. Its function `epDeconv` deconvolves bulk DNAm microarray data via an RNA reference. It also needs another dataset with paired RNA-DNAm data. Because of the pairing, I assume the cell contents of the RNA and DNAm samples are similar. Hence, the deconvolution results from the RNA reference and RNA samples can be shared with the paired DNAm samples to train a cell contents prediction model with DNAm features.

The third module is a feature selection module. It accepts the deconvolution results from the former modules. Then, it uses them to identify cell-type-specific inter-group differential features from the bulk data.



Citation

Liu Y. *scDeconv*: an R package to deconvolve bulk DNA methylation data with scRNA-seq data and paired bulk RNA-DNA methylation data. *Brief Bioinform.* 2022 May 13;23(3):bbac150. doi: 10.1093/bib/bbac150. PMID: 35453146; PMCID: PMC9271220.

Installation

The code of *scDeconv* is freely available at <https://github.com/yuabrahamliu/scDeconv>.

The following commands can be used to install this R package via GitHub.

```
library(devtools)

install_github('yuabrahamliu/scDeconv')
```

It can also be installed via Bioconductor.

```
if (!require("BiocManager", quietly = TRUE))
  install.packages("BiocManager")

# The following initializes usage of Bioc devel
BiocManager::install(version = 'devel')

BiocManager::install("scDeconv")
```

Data preparation

We will use the data that accompany with *scDeconv* package in this tutorial. They contain a *Seurat* object generated from scRNA-seq data and preprocessed by the R package *Seurat*. It is a subset of a human placenta scRNA-seq dataset in ArrayExpress, with experiment code E-MTAB-6701 (droplet-based data), and covers 1388 cells and 21737 genes. We will deconvolve 4 main placental cell types in it, including extravillous trophoblasts (EVTs), fibroblasts (FBs), Hofbauer cells (HBs), and villous cytotrophoblasts (VCTs). Among them, EVT and VCT are epithelial trophoblasts with similar origins, while HB cells are fetal macrophages in the placenta. The cell type information is contained in the metadata of the *Seurat* object. In addition, this *Seurat* object also has the gene read count data and normalized data. We will use it to generate an RNA deconvolution reference via *scDeconv*.

On the other hand, the data that need to be deconvolved are bulk DNAm data with 311 human placenta samples and 18626 probes. They are collected from 9 different GEO datasets based on the platforms of Illumina 27K and 450K and have gone through preprocessing with batch difference adjusted and the shared high-quality probes retained.

To deconvolve these DNAm data with the scRNA-seq data, *scDeconv* also needs a paired bulk RNA-bulk DNAm dataset to fulfill the trans-omics deconvolution, which is also in the accompanying data, with 48 human placenta samples. Its RNA part contains 22188 genes, and its DNAm part contains 18626 probes, the same as the ones in the DNAm data to be deconvolved. Because batch difference exists between the paired DNAm data and the ones to be deconvolved, they have been adjusted with the *ComBat* function in the R package *sva*, using the paired DNAm set as the reference batch.

The paired RNA data are from the platform Affymetrix Human Gene 1.0 ST Array, and the gene expression values are library size normalized values with log2 transformation. In contrast, that in the DNAm datasets are beta values. The samples in both the paired dataset and the external DNAm dataset to be deconvolved can be divided into 2 groups. One is the normal sample group, and the other is the disease group with a preeclampsia pregnancy complication. This information and the original GEO datasets of the samples can be found in the metadata frame coupled with this package.

Now, attach *scDeconv* to the R session and take a look at these data.

```

library(scDeconv)

scrRNA <- system.file('extdata', 'scrRNAseqdat.rds', package = 'scDeconv')
scrRNA <- readRDS(scrRNA)

prRNA <- system.file('extdata', 'pairedRNA.dat.rds', package = 'scDeconv')
prRNA <- readRDS(prRNA)

pdNAM <- system.file('extdata', 'pairedDNAM.dat.rds', package = 'scDeconv')
pdNAM <- readRDS(pdNAM)

externalDNAM <- system.file('extdata', 'externalDNAM.dat.rds', package = 'scDeconv')
externalDNAM <- readRDS(externalDNAM)

DNampd <- system.file('extdata', 'DNampd.rds', package = 'scDeconv')
DNampd <- readRDS(DNampd)

```

The summary or beginning parts of these data are shown below.

```

#The scrRNA-seq data
scrRNA
#> An object of class Seurat
#> 21737 features across 1388 samples within 1 assay
#> Active assay: RNA (21737 features, 0 variable features)
head(scrRNA@meta.data)
#>
#> orig.ident nCount_RNA nFeature_RNA Fetus location
#> FCA7196226_CATGCCTGTCCCTTGT FCA7196226 23559 4736 F27 Placenta
#> FCA7196226_AGCGGTCAGCTGCAAG FCA7196226 8203 3194 F27 Placenta
#> FCA7196226_AATCCAGCATTGGCGC FCA7196226 18224 3602 F27 Placenta
#> FCA7196226_GAACATCTCTTGTACT FCA7196226 18266 4077 F27 Placenta
#> FCA7196226_AGATCTGGTTGCCTCT FCA7196226 17689 4169 F27 Placenta
#> FCA7196226_CTCGAGGAGGGCACTA FCA7196226 6314 2796 F27 Placenta
#>
#> annotation
#> FCA7196226_CATGCCTGTCCCTTGT VCT
#> FCA7196226_AGCGGTCAGCTGCAAG VCT
#> FCA7196226_AATCCAGCATTGGCGC VCT
#> FCA7196226_GAACATCTCTTGTACT VCT
#> FCA7196226_AGATCTGGTTGCCTCT VCT
#> FCA7196226_CTCGAGGAGGGCACTA VCT

```

```

#The paired RNA microarray data
prRNA[1:6,1:6]
#>
#> GSM1940495 GSM1940496 GSM1940499 GSM1940500 GSM1940501 GSM1940502
#> A1BG 7.376765 7.413560 7.285113 7.352853 7.488182 7.300325
#> A1CF 5.522374 5.537723 5.463703 5.671208 5.581101 5.413397
#> A2M 11.622694 11.120252 11.230030 11.327457 11.541988 11.442168
#> A2ML1 5.994014 6.101948 5.982931 6.033332 6.075942 6.036454
#> A3GALT2 6.150027 6.216505 6.046596 6.320198 6.233631 6.119112
#> A4GALT 7.752668 8.059492 7.935501 7.676856 7.830030 7.920985

```

```

#The paired DNAM data
pdNAM[1:6,1:6]
#>
#> GSM1940495 GSM1940496 GSM1940499 GSM1940500 GSM1940501 GSM1940502

```

```
#> cg00000292 0.73358449 0.59264315 0.59496274 0.63138095 0.66699830 0.67046169
#> cg000002426 0.69265004 0.51667233 0.47864586 0.45470146 0.51774379 0.59661450
#> cg000003994 0.16837433 0.12105939 0.22729146 0.22437398 0.32077141 0.24005318
#> cg000007981 0.04251201 0.02582752 0.03510695 0.03843307 0.04369758 0.03930951
#> cg000008493 0.41044308 0.32724144 0.39277370 0.30326993 0.48137703 0.48915648
#> cg000008713 0.05892651 0.04065540 0.06138317 0.08263494 0.09078148 0.08007630
```

```
#The external DNAm data to be deconvolved
```

```
externalDNAm[1:6,1:6]
```

```
#>
#> GSM788417 GSM788419 GSM788420 GSM788421 GSM788414 GSM788415
#> cg00000292 0.65961366 0.67591141 0.65709651 0.66077820 0.66847653 0.67436406
#> cg000002426 0.53516824 0.53883284 0.53683120 0.53990206 0.53804637 0.53543344
#> cg000003994 0.17674229 0.16432771 0.16631494 0.16803355 0.16416337 0.16852233
#> cg000007981 0.03553198 0.02934814 0.03189098 0.02765783 0.02768899 0.02764183
#> cg000008493 0.43619912 0.42998023 0.43761396 0.44973257 0.46281094 0.45935308
#> cg000008713 0.07431056 0.06367040 0.06764089 0.06218142 0.05267028 0.05397539
```

```
#The metadata for the paired samples
```

```
head(subset(DNAmPd, type == 'paired'))
```

```
#> sampleid Samplegroup Gestwk dataset type
#> 1 GSM1940495 Preeclampsia 37 GSE98224 paired
#> 2 GSM1940496 Preeclampsia 29 GSE98224 paired
#> 3 GSM1940499 Preeclampsia 35 GSE98224 paired
#> 4 GSM1940500 Preeclampsia 31 GSE98224 paired
#> 5 GSM1940501 Preeclampsia 29 GSE98224 paired
#> 6 GSM1940502 Preeclampsia 37 GSE98224 paired
```

```
table(subset(DNAmPd, type == 'paired')$Samplegroup)
```

```
#>
```

```
#> Control Preeclampsia
```

```
#> 18 30
```

```
#The metadata for the external samples to be deconvolved
```

```
head(subset(DNAmPd, type == 'external'))
```

```
#> sampleid Samplegroup Gestwk dataset type
#> 49 GSM788417 Control 8 GSE31781 external
#> 50 GSM788419 Control 8 GSE31781 external
#> 51 GSM788420 Control 8 GSE31781 external
#> 52 GSM788421 Control 9 GSE31781 external
#> 53 GSM788414 Control 12 GSE31781 external
#> 54 GSM788415 Control 12 GSE31781 external
```

```
table(subset(DNAmPd, type == 'external')$Samplegroup)
```

```
#>
```

```
#> Control Preeclampsia
```

```
#> 240 71
```

RNA reference generation and bulk RNA data deconvolution

We will first use the scRNA-seq and paired RNA data to construct an RNA deconvolution reference. It can be achieved via the function `scRef` in the package.

We provide the scRNA-seq data to `scRef` via its parameter `Seuratobj` and set another parameter `targetcelltypes` as `c('EVT', 'FB', 'HB', 'VCT')`, meaning these 4 cell types in the scRNA-seq data will be covered to generate the reference. The parameter `celltypecolname` is set as “annotation”, indicating

that the cell type information for every single cell is stored in the column “annotation” of the scRNA-seq metadata frame.

Because the first step of reference making is to synthesize several pseudo-bulk RNA-seq samples for each cell type from the scRNA-seq data, the parameter `pseudobulknum` is used to set how many such samples will be made for each cell type, and we set it as 100 here, meaning each cell type will get 100 pseudo-bulk RNA samples via sampling from the scRNA-seq data, and the 4 cell types will totally get $100 \times 4 = 400$ such samples.

It should be noted that for large scRNA-seq data, this pseudo-bulk sample synthesis step of `scRef` can be very slow. In this case, I suggest using another function, `prepseudobulk`, in the package to conduct this step first. Then save the output and transfer it to `scRef` via its parameter `pseudobulkdat` to construct the reference. It is because the same scRNA-seq data may be needed to deconvolve various bulk RNA or DNAm datasets, so saving the pseudo-bulk samples can avoid repeating their synthesis for different deconvolution tasks.

Then, the synthesized samples will go through several steps to get the final reference, while if the bulk RNA data need to be deconvolved (the paired RNA data `pRNA` here) is provided to the parameter `targetdat`, a batch adjustment step will be included to remove the batch difference or platform difference between the scRNA-seq and the bulk RNA data. If no data is provided to it, this step will be skipped. Because the values in `pRNA` are log2 transformed values, we set the parameter `targetlogged` as `TRUE`.

```
#Generate the reference with scRef directly
refres <- scRef(Seuratobj = scRNA,
#           targetcelltypes = c('EVT', 'FB', 'HB', 'VCT'),
#           celltypecolname = 'annotation',
#           pseudobulknum = 100,
#           targetdat = pRNA,
#           targetlogged = TRUE)

#Or use the function prepseudobulk to synthesize the pseudo-bulk data first,
#and then transfer the pseudo-bulk data to scRef to generate the reference,
#which is more suggested
pseudobulk <- prepseudobulk(Seuratobj = scRNA,
                           targetcelltypes = c('EVT', 'FB', 'HB', 'VCT'),
                           celltypecolname = 'annotation',
                           pseudobulknum = 100)

refres <- scRef(Seuratobj = scRNA,
               targetcelltypes = c('EVT', 'FB', 'HB', 'VCT'),
               celltypecolname = 'annotation',
               pseudobulkdat = pseudobulk,
               targetdat = pRNA,
               targetlogged = TRUE)

#Save the synthesized pseudo-bulk dataset so that it can be used repetitively
#for different tasks
saveRDS(pseudobulk, 'pseudobulk.rds')
```

The result `refres` is a list containing 2 slots. The one named “ref” is the RNA reference generated.

```
head(refres$ref)
#>           EVT           FB           HB           VCT
#> ZNF490 15.350183 14.510841 14.510841 14.510841
#> CP      21.066156 20.387364 20.387364 20.387364
```

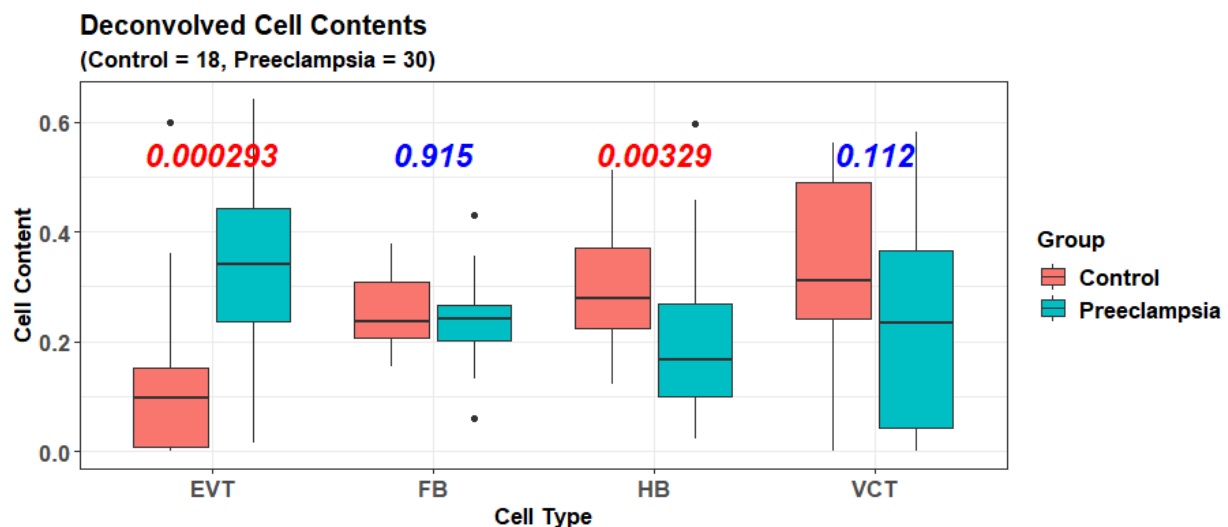
```
#> ABCC2 14.769732 14.266494 14.266494 14.266494
#> SULF1 12.160773 12.160773 16.972445 12.160773
#> PADI1 26.177981 10.876944 6.304278 4.723937
#> TLR8 8.157669 7.737533 25.150155 8.579287
```

The other slot named “targetnolog” is the adjusted bulk RNA data to be deconvolved, and the values in it are non-log transformed values.

```
refres$targetnolog[1:6,1:6]
#> GSM1940495 GSM1940496 GSM1940499 GSM1940500 GSM1940501 GSM1940502
#> ZNF490 18.416815 13.522850 14.930505 10.946805 13.748425 8.916959
#> CP 0.000000 0.000000 149.774824 0.000000 0.000000 0.000000
#> ABCC2 14.707393 7.098110 7.169431 16.389123 12.370262 10.784374
#> SULF1 9.567271 11.716441 34.076833 2.233104 9.400352 3.464205
#> PADI1 3.824240 7.766002 10.525161 15.920500 18.052632 3.175283
#> TLR8 7.326088 14.035631 16.531961 4.667870 4.065535 15.483962
```

After getting the RNA reference and the adjusted bulk RNA data, they can be transferred to the function `refDeconv` in the package to deconvolve the bulk RNA data with the RNA reference. The obtained RNA reference `refres$ref` and the adjusted bulk RNA data `refres$targetnolog` can be transferred to the parameters `ref` and `targetdat`, and because the values of `refres$targetnolog` are non-log transformed, the parameter `targetlogged` is set as `FALSE`. Another parameter `rescale` can be set as `TRUE` so that the sum of all the deconvolved cell contents for each sample will be 1.

```
rnares <- refDeconv(ref = refres$ref,
  targetdat = refres$targetnolog,
  targetlogged = FALSE,
  resscale = TRUE,
  plot = TRUE,
  pddat = subset(DNAmpd, type == 'paired'))
```



```
#Result for the bulk RNA microarray data
head(rnares)
#>
#> GSM1940495 EVT FB HB VCT
#> GSM1940495 0.33828645 0.1931764 0.22473700 0.2438002
```

```
#> GSM1940496 0.34555565 0.2599395 0.17158354 0.2229213
#> GSM1940499 0.38832639 0.2543652 0.35730844 0.0000000
#> GSM1940500 0.38614136 0.2566712 0.02217642 0.3350110
#> GSM1940501 0.26575362 0.3056125 0.11986884 0.3087651
#> GSM1940502 0.05083727 0.2008974 0.34875269 0.3995126
```

It is noteworthy that the result of **refDeconv** is very accurate and outperforms all other popular RNA deconvolution algorithms. The details can be found in my original publication for *scDeconv* (Liu Y. *scDeconv*: an R package to deconvolve bulk DNA methylation data with scRNA-seq data and paired bulk RNA-DNA methylation data. *Brief Bioinform.* 2022 May 13;23(3):bbac150. doi: 10.1093/bib/bbac150. PMID: 35453146; PMCID: PMC9271220).

Bulk DNAm data deconvolution with RNA reference

After getting the RNA reference, we use it to deconvolve the external DNAm data **externalDNAm** via the function **epDeconv**. It also needs the paired bulk RNA-bulk DNAm dataset, and for the RNA part, it is the adjusted bulk RNA data returned by **scRef**, and we provide it to the parameter **rnamat**, while for the DNAm part, we provide the data **pDNAm** to **methylnmat**. Because the values in the adjusted RNA data are non-log transformed values, we set the parameter **rnamatlogged** as **FALSE**.

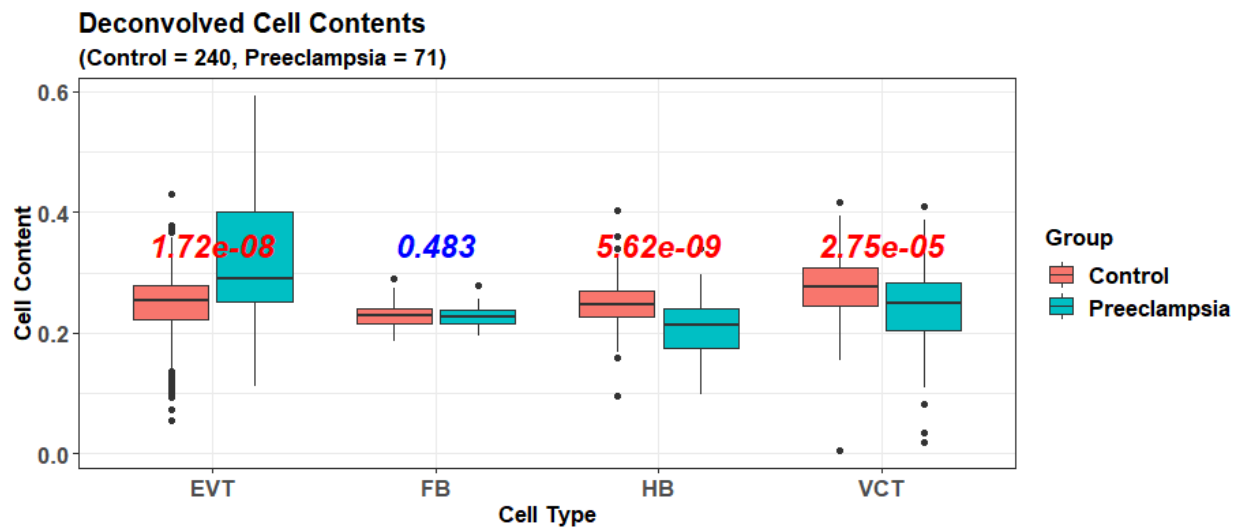
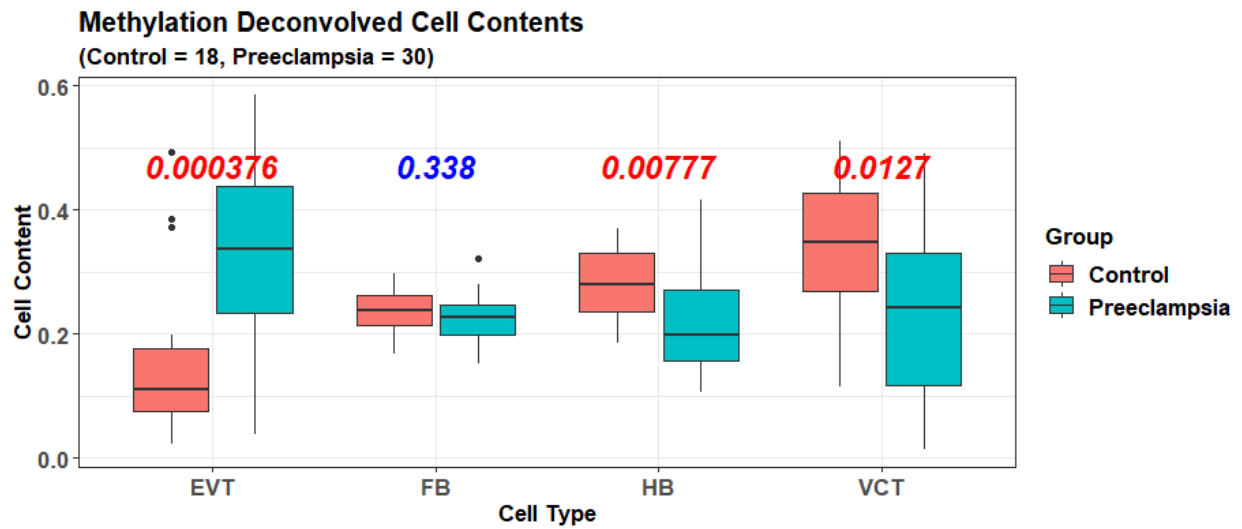
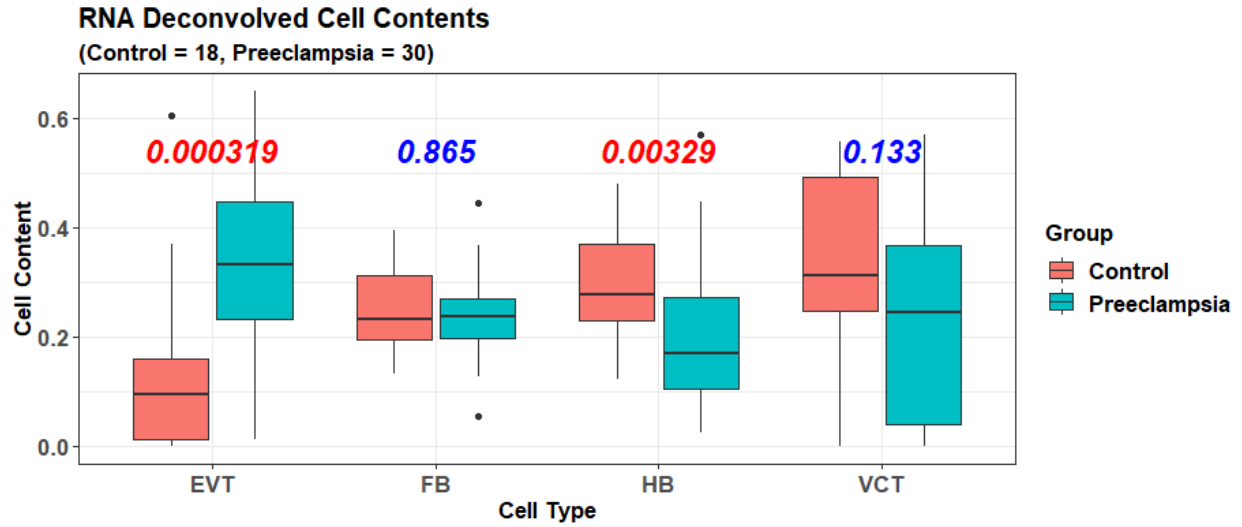
To deconvolve **externalDNAm** with the RNA reference **refres\$ref**, **epDeconv** will construct an ensemble model on the paired RNA and DNAm data in a co-training manner and then use this model to predict the cell contents for **externalDNAm**. For the number of base learners of the ensemble, it is defined by the parameter **learnernum**, and we set it as 10 here. While because we want the 4 cell contents deconvolved can have a sum of 1 for each DNAm sample, we set the parameter **resscale** as **TRUE**.

If we want box plots to show the deconvolution results for the paired RNA, paired DNAm, and external DNAm data, we can set the parameter **plot** as **TRUE**. At the same time, we can provide the metadata frame of the paired samples to **pddat**, and that of the external samples to **targetmethylnpddat**, so that the sample group information can be transferred to the function, and the cell content difference can also be shown in the plot.

```
dnamres <- epDeconv(rnaref = refres$ref,
                    rnamat = refres$targetnolog,
                    rnamatlogged = FALSE,

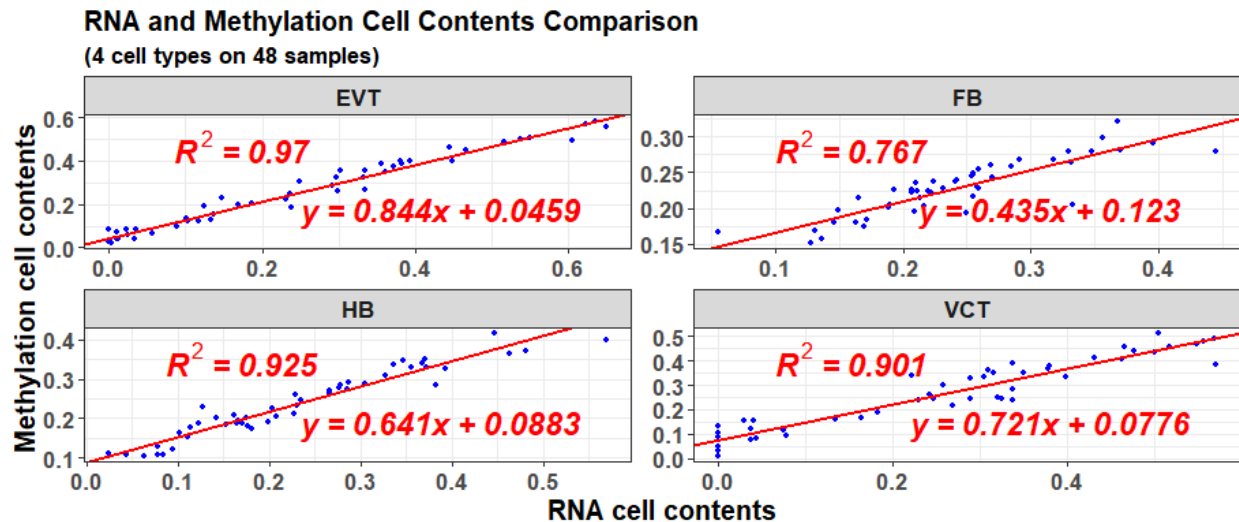
                    methylnmat = pDNAm,
                    learnernum = 10,
                    resscale = TRUE,

                    targetmethylnmat = externalDNAm,
                    plot = TRUE,
                    pddat = subset(DNAmPd, type == 'paired'),
                    targetmethylnpddat = subset(DNAmPd, type == 'external'))
```

From the box plots, we can see that in all the 3 datasets, preeclampsia samples have a much higher EVT cell content than normal samples, while their HB and VCT cells are largely reduced.

In addition, 4 scatter plots are also generated for the cell types as `plot` is set as `TRUE`, and they compare the deconvolution results between the paired RNA data and the paired DNAm data. Because `epDeconv` constructs the model based on the assumption that paired samples have similar cell compositions, the results predicted by it should show a high correlation between the RNA and DNAm data.



The concrete values of the deconvolution results can be seen from the result `dnamres`. Its slots “`rnacellconts`” and “`methylcellconts`” contain the results for the paired data, while the slot “`methyltargetcellconts`” contains the results for the external DNAm data.

```
#Result for the paired RNA microarray data
head(dnamres$rnacellconts)
#>           EVT           FB           HB           VCT
#> GSM1940495 0.33367213 0.1931336 0.22831875 0.2465819
#> GSM1940496 0.33135710 0.2586181 0.16554716 0.2454366
#> GSM1940499 0.39360023 0.2416980 0.37026306 0.0000000
#> GSM1940500 0.38089687 0.2550985 0.02477827 0.3369090
#> GSM1940501 0.24784508 0.3179029 0.11330019 0.3205049
#> GSM1940502 0.05616103 0.2068742 0.35558339 0.3797657
```

```
#Result for the paired DNAm data
head(dnamres$methylcellconts)
#>           EVT           FB           HB           VCT
#> GSM1940495 0.26947568 0.2267247 0.2616417 0.24215789
#> GSM1940496 0.32627289 0.2291105 0.1934914 0.25112522
#> GSM1940499 0.39789689 0.2396586 0.3487757 0.01366883
#> GSM1940500 0.40208801 0.2484923 0.1127943 0.23662538
#> GSM1940501 0.30573949 0.2670643 0.1772920 0.24990424
#> GSM1940502 0.06658484 0.2262086 0.3310046 0.37620193
```

```
#Result for the external DNAm data
head(dnamres$methyltargetcellcounts)
#>           EVT           FB           HB           VCT
#> GSM788417 0.2817118 0.2298730 0.2091309 0.2792843
#> GSM788419 0.3122537 0.2109420 0.2037660 0.2730383
#> GSM788420 0.2960548 0.2214043 0.2059084 0.2766325
#> GSM788421 0.2852733 0.2101249 0.1959775 0.3086243
```

```
#> GSM788414 0.2978430 0.2068120 0.1911144 0.3042306
#> GSM788415 0.3099221 0.2017879 0.1769487 0.3113413
```

The trained ensemble model is also in **dnamres**; its slot “modellist” is the base learners of the model, while “normweights” is the base learner weights for the ensemble.

If **externalDNAm** is not provided to **epDeconv**, it will not influence the ensemble model training on the paired data, and the same model can still be returned. Then, **externalDNAm** can be transferred with the **epDeconv** result to the function **methypredict** to predict the external sample cell contents.

```
externalcellconts <- methypredict(model = dnamres,
                                   targetmethyldat = externalDNAm,
                                   resscale = TRUE,
                                   adjustminus = TRUE)
```

```
head(externalcellconts)
#>           EVT           FB           HB           VCT
#> GSM788417 0.2817118 0.2298730 0.2091309 0.2792843
#> GSM788419 0.3122537 0.2109420 0.2037660 0.2730383
#> GSM788420 0.2960548 0.2214043 0.2059084 0.2766325
#> GSM788421 0.2852733 0.2101249 0.1959775 0.3086243
#> GSM788414 0.2978430 0.2068120 0.1911144 0.3042306
#> GSM788415 0.3099221 0.2017879 0.1769487 0.3113413
```

In addition, *scDeconv* also contains other useful functions such as **celldiff**, which selects cell-type-specific inter-group differential features from bulk data; and **enrichwrapper**, which annotates differential DNAm feature function using a correlation-based method. To make this tutorial simple and clear, they will not be covered here, but the users can explore them via the help documents and the original publication for this package.

My original publication on *scDeconv* contains more examples of its functions and is recommended to read (Liu Y. *scDeconv*: an R package to deconvolve bulk DNA methylation data with scRNA-seq data and paired bulk RNA-DNA methylation data. *Brief Bioinform.* 2022 May 13;23(3):bbac150. doi: 10.1093/bib/bbac150. PMID: 35453146; PMCID: PMC9271220).

Session information

The session information for this vignette is as follows.

```
sessionInfo()
#> R version 4.0.5 (2021-03-31)
#> Platform: x86_64-w64-mingw32/x64 (64-bit)
#> Running under: Windows 10 x64 (build 19042)
#>
#> Matrix products: default
#>
#> locale:
#> [1] LC_COLLATE=English_United States.1252
#> [2] LC_CTYPE=English_United States.1252
#> [3] LC_MONETARY=English_United States.1252
#> [4] LC_NUMERIC=C
#> [5] LC_TIME=English_United States.1252
#>
```

```

#> attached base packages:
#> [1] stats      graphics  grDevices  utils      datasets  methods    base
#>
#> other attached packages:
#> [1] Seurat_4.0.1      SeuratObject_4.0.0 scDeconv_1.0.0
#>
#> loaded via a namespace (and not attached):
#>  [1] plyr_1.8.6              igraph_1.2.6
#>  [3] lazyeval_0.2.2          splines_4.0.5
#>  [5] BiocParallel_1.24.1     listenv_0.8.0
#>  [7] scattermore_0.7         GenomeInfoDb_1.26.7
#>  [9] ggplot2_3.3.3           scater_1.18.6
#> [11] sva_3.38.0              digest_0.6.27
#> [13] foreach_1.5.1           htmltools_0.5.1.1
#> [15] viridis_0.6.0           fansi_0.4.2
#> [17] magrittr_2.0.1          memoise_2.0.0
#> [19] tensor_1.5              cluster_2.1.1
#> [21] ROCR_1.0-11             limma_3.46.0
#> [23] annotate_1.68.0         globals_0.14.0
#> [25] matrixStats_0.58.0     spatstat.sparse_2.0-0
#> [27] colorspace_2.0-0       blob_1.2.1
#> [29] ggrepel_0.9.1           xfun_0.22
#> [31] dplyr_1.0.5             crayon_1.4.1
#> [33] RCurl_1.98-1.3          jsonlite_1.7.2
#> [35] genefilter_1.72.1       spatstat.data_2.1-0
#> [37] iterators_1.0.13        survival_3.2-10
#> [39] zoo_1.8-9               glue_1.4.2
#> [41] polyclip_1.10-0         gtable_0.3.0
#> [43] zlibbioc_1.36.0         XVector_0.30.0
#> [45] leiden_0.3.7            DelayedArray_0.16.3
#> [47] BiocSingular_1.6.0      shape_1.4.5
#> [49] future.apply_1.7.0      SingleCellExperiment_1.12.0
#> [51] BiocGenerics_0.36.0     abind_1.4-5
#> [53] scales_1.1.1            pheatmap_1.0.12
#> [55] edgeR_3.32.1            DBI_1.1.1
#> [57] miniUI_0.1.1.1          Rcpp_1.0.7
#> [59] viridisLite_0.4.0       xtable_1.8-4
#> [61] reticulate_1.19         spatstat.core_2.1-2
#> [63] rsud_1.0.5              bit_4.0.4
#> [65] stats4_4.0.5            glmnet_4.1-1
#> [67] htmlwidgets_1.5.3       httr_1.4.2
#> [69] RColorBrewer_1.1-2      ellipsis_0.3.2
#> [71] ica_1.0-2               farver_2.1.0
#> [73] XML_3.99-0.6            pkgconfig_2.0.3
#> [75] scuttle_1.0.4           uwot_0.1.10
#> [77] deldir_1.0-6            locfit_1.5-9.4
#> [79] utf8_1.2.1              labeling_0.4.2
#> [81] tidyselect_1.1.2        rlang_1.0.1
#> [83] reshape2_1.4.4          later_1.1.0.1
#> [85] AnnotationDbi_1.52.0    munsell_0.5.0
#> [87] tools_4.0.5             cachem_1.0.4
#> [89] cli_2.4.0               generics_0.1.0
#> [91] RSQLite_2.2.5           ggribes_0.5.3

```

```

#> [93] evaluate_0.14
#> [95] fastmap_1.1.0
#> [97] goftest_1.2-2
#> [99] knitr_1.34
#> [101] fitdistrplus_1.1-3
#> [103] RANN_2.6.1
#> [105] pbapply_1.4-3
#> [107] nlme_3.1-152
#> [109] compiler_4.0.5
#> [111] plotly_4.9.3
#> [113] spatstat.utils_2.3-0
#> [115] stringi_1.5.3
#> [117] Matrix_1.3-2
#> [119] pillar_1.7.0
#> [121] spatstat.geom_2.3-2
#> [123] BiocNeighbors_1.8.2
#> [125] data.table_1.14.0
#> [127] bitops_1.0-6
#> [129] httpuv_1.5.5
#> [131] GenomicRanges_1.42.0
#> [133] promises_1.2.0.1
#> [135] gridExtra_2.3
#> [137] IRanges_2.24.1
#> [139] codetools_0.2-18
#> [141] assertthat_0.2.1
#> [143] sctransform_0.3.2
#> [145] GenomeInfoDbData_1.2.4
#> [147] parallel_4.0.5
#> [149] grid_4.0.5
#> [151] tidyr_1.1.3
#> [153] rmarkdown_2.16
#> [155] Rtsne_0.15
#> [157] shiny_1.6.0
stringr_1.4.0
yaml_2.2.1
org.Hs.eg.db_3.12.0
bit64_4.0.5
purrr_0.3.4
sparseMatrixStats_1.2.1
future_1.21.0
mime_0.10
beeswarm_0.3.1
png_0.1-7
tibble_3.1.0
lattice_0.20-41
vctrs_0.3.8
lifecycle_1.0.1
lmtest_0.9-38
RcppAnnoy_0.0.18
cowplot_1.1.1
irlba_2.3.3
patchwork_1.1.1
R6_2.5.0
KernSmooth_2.23-18
vipor_0.4.5
parallelly_1.25.0
MASS_7.3-53.1
SummarizedExperiment_1.20.0
S4Vectors_0.28.1
mgcv_1.8-34
beachmat_2.6.4
rpart_4.1-15
DelayedMatrixStats_1.12.3
MatrixGenerics_1.2.1
Biobase_2.50.0
ggbeeswarm_0.6.0

```