

NormExpression.....	1
1 Normalization with simple evaluation	2
1.A R package installation and data preparation	2
1.B How to produce scRNA663_factors (single-cell data)	3
1.C Normalization without evaluation (single-cell data)	4
1.D Normalization without evaluation (bulk data).....	4
1.E Using AUCVC for simply evaluation (single-cell data)	5
1.F Using AUCVC for simply evaluation (bulk data).....	6
2 Normalization with complete evaluation	6
2.A Using AUCVC for complete evaluation (single-cell data).....	6
2.B Using AUCVC for complete evaluation (bulk data)	7
2.C Using mSCC for complete evaluation (single-cell data)	8
2.D Using mSCC for complete evaluation (bulk data).....	9
3 Visualization of evaluation results	10
3.A CV threshold Curve (single-cell data).....	10
3.B CV threshold Curve (bulk data).....	11
3.C Distribution of SCCs between genes (single-cell data)	12
3.D Distribution of SCCs between genes (bulk data).....	13
3.E Hierarchical clustering of normalization factors (single-cell data).....	14
3.F Hierarchical clustering of normalization factors (bulk data).....	15
4 How to evaluate many users' methods	16
4.A For methods with normalization factors (single-cell data)	16
4.B For a general usage (single-cell data)	17
5 Package versions and FAQ	18

NormExpression

NormExpression provides a framework and a fast and simple way for researchers to evaluate methods (particularly some data-driven methods or their own methods) and then select a best one for data normalization in the gene expression analysis, based on the theory that a successful normalization method simultaneously maximizes the number of uniform genes and minimizes the correlation between the expression profiles of gene pairs. To evaluate normalization methods, NormExpression uses two metrics, which are AUCVC and mSCC [1].

NormExpression can be used in three ways: normalization without evaluation, normalization with simple evaluation or normalization with complete evaluation. (1) Normalization without evaluation is suggested to use TU to process users' data, since it has been already ranked as the best method for both scRNA-seq (**1.C**) and bulk RNA-seq data (**1.D**) [1]. (2) Normalization with simple evaluation by AUCVC is suggested to select the best method from 10 normalization methods using users' single-cell data (**1.E**) or bulk data (**1.F**), which are HG7, ERCC (if available), TN, TC, CR, NR, DESeq (RLE), UQ and TMM. (3) Normalization with complete evaluation by AUCVC and mSCC is suggested to select the best method from at least 10 normalization methods plus TU, using users' single-cell data (**2.AC**) or bulk data (**2.BD**). Normalization with simple evaluation determines the best method based on its AUCVC value, while normalization with complete evaluation need consider the consistency of evaluation results using two metrics (**the consistency of two metrics**). We suggest to use (3) as the best choice, although **it is time-consuming**. If users chose to use (1), we suggest them to conduct (2) and compare the results with our previous results [1] to validate (1). This consistency can also be used to analyze the quality of gene expression data.

Notice : (1) Users can run all the R coding as below by "copy-paste"; (2) To use NormExpression, please cite this paper.

[1] Zhenfeng Wu, Weixiang Liu, Xiufeng Jin, Deshui Yu, Hua Wang, Gustavo Glusman, Max Robinson, Lin Liu, Jishou Ruan, Shan Gao (2018) NormExpression: an R package to normalize gene expression data using evaluated methods. bioRxiv. <https://doi.org/10.1101/251140>

Two datasets scRNA663 (single-cell data) and bkRNA18 (bulk data) can be used to evaluate users' methods. These two datasets were designed to use the same library-construction and sequencing protocol to validate the consistency of evaluations by two types of data (**the consistency of two data types**). Users should select the best one for their data normalization based on the evaluate methods using their own data. The gene expression data of scRNA663 will be released after the relevant paper is published. Before the release day, users can request the data by emailing Shan Gao (gao_shan@mail.nankai.edu.cn) but need provide their names and affiliated institutions.

1 Normalization with simple evaluation

1.A R package installation and data preparation

```
# Set the working directory
setwd("d:/working_dir");

# Install R package
install.packages("NormExpression");
install.packages("ggplot2");
install.packages("dendextend");

# Load libraries
library(NormExpression);
library(ggplot2);
library(dendextend);

# Read all data
# scRNA663 will be released after the relevant paper is published.
data(scRNA663);
# Readers can request scRNA663.txt and read it by
scRNA663 <- read.table(file='scRNA663.txt', header=TRUE, row.names=1);
# scRNA663_factors contains all the pre-calculated normalization factors using scRNA663
data(scRNA663_factors);
# bkRNA18 is bulk data
data(bkRNA18);
# bkRNA18_factors contains all the pre-calculated normalization factors using bkRNA18
data(bkRNA18_factors);

# Use SCnorm to produce the normalized gene expression matrix (single-cell data)
```

```

source("https://bioconductor.org/biocLite.R");
biocLite("SCnorm");
library(SCnorm);
Conditions = rep(c(1), each= 663);
pdf("scRNA663_count-depth_norm.pdf", height=7.5, width=10.5);
DataNorm <- SCnorm(Data = scRNA663, Conditions = Conditions, FilterExpression = 4,
PrintProgressPlots = TRUE, reportSF = TRUE, NCores=1);
dev.off();
NormalizedData <- results(DataNorm);
scRNA663.SCnorm <- round(NormalizedData, 2);

# Use SCnorm to produce the normalized gene expression matrix (bulk data)
Conditions = rep(c(1), each= 18);
pdf("bkRNA18_count-depth_norm.pdf", height=7.5, width=10.5);
DataNorm <- SCnorm(Data = bkRNA18, Conditions = Conditions, FilterExpression = 5,
PrintProgressPlots = TRUE, reportSF = TRUE, NCores=1);
dev.off();
NormalizedData <- results(DataNorm);
bkRNA18.SCnorm <- round(NormalizedData, 2);

```

Results:

```

> bkRNA18[1:10,1:8]
      col3616_1 col3816_3 col3916_5 col4016_7 col4416_9 col4516_11 col4716_13 col4816_97
DDX11L1      0      0      0      0      0      0      0      0
WASH7P(1)    6      0      0      0      0      0      3      3
MIR6859-1    0      0      0      0      0      0      0      0
MIR1302-2    0      0      0      0      0      0      0      0
FAM138A      0      0      0      0      0      0      0      0
OR4G4P      0      0      0      0      0      0      0      0
OR4G11P      0      0      0      0      0      0      0      0
OR4F5        0      0      0      0      0      0      0      0
RP11-34P13.7 0      0      0      0      0      0      0      0
RP11-34P13.8 0      0      0      0      0      0      0      0

> bkRNA18_factors[1:10,1:8]
      HG7  ERCC  TN  TC  CR  NR  DESeq  UQ
col3616_1 0.86034 0.90514 0.96726 0.90562 0.90675 0.98663 0.97831 0.94329
col3816_3 0.84092 0.89020 0.90894 0.89073 0.89200 0.95924 0.88113 0.87530
col3916_5 0.82312 1.02033 1.05665 1.02031 1.02024 1.02550 1.04620 0.93422
col4016_7 1.99929 1.23959 1.30460 1.23826 1.23509 1.41115 1.36199 1.31295
col4416_9 0.70601 0.90239 0.89050 0.90287 0.90403 0.84078 0.99071 1.00163
col4516_11 0.86904 1.35563 1.19789 1.35344 1.34818 1.18330 1.11197 1.10407
col4716_13 1.13279 1.21654 1.20209 1.21537 1.21257 1.16696 1.19472 1.17058
col4816_97 0.67727 1.02185 1.03256 1.02182 1.02173 0.91903 0.87090 0.78991
col5216_17 0.92637 0.90147 0.87049 0.90195 0.90312 0.84806 0.99604 1.03360
col3616_2 1.57594 1.09422 1.11510 1.09381 1.09281 1.14343 1.13568 1.13304

```

1.B How to produce scRNA663_factors (single-cell data)

```

# bkRNA18_factors can be produced in the same way
# TU, NCS and ES are parameter dependent methods
# So, scRNA663_factors contain TU, NCS and ES normalization factors produced using the
optimal parameters
# For HG7, ERCC, TC, CR and NR
housekeeping.list <- read.table(file='housekeeping.txt', header = FALSE);

```

```
hk_name <- as.matrix(housekeeping.list)[,1];
HG7.size <- colSums(scRNA663[hk_name,]);
HG7_factors <- getFactors(data=scRNA663, lib.size=HG7.size, method="sizefactor");

# For DESeq(RLE), UQ and TMM
DESeq_factors <- getFactors(data= scRNA663, method="DESeq");

.....
scRNA663_factors <- cbind(HG7_factors, DESeq_factors.....);
colnames(scRNA663_factors)=c("HG7", "ERCC", "TN", "TC", "CR", "NR", "DESeq", "UQ",
"TMM", "TU", "NCS", "ES");
```

1.C Normalization without evaluation (single-cell data)

```
# Grid of non-zero ratios to produce AUCVCs for TU and it is time-consuming
scRNA663.AUCVCs1 <- gridAUCVC(data= scRNA663,  dataType="sc",  TU= 1,
nonzeroRatios= c(0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9));

# Look at the maximum AUCVC for TU
scRNA663.AUCVCs1;

# Find the parameters used by TU when it achieved the maximum AUCVC in bestPara.txt
bestPara <- read.table(file='bestPara.txt', header=TRUE);
bestPara;

# Using the parameters to produce the TU normalization factor
tu <- getFactors(data = scRNA663, method = "TU", pre_ratio=0.5, lower_trim=0.05,
upper_trim=0.65);

# Produce the TU-normalized gene expression matrix
TU_sc.matrix <- getNormMatrix(data = scRNA663, norm.factors = tu);
```

1.D Normalization without evaluation (bulk data)

```
# Grid of non-zero ratios to produce AUCVCs for TU and it is time-consuming
# nonzeroRatios can be set to 1 for bulk data to reduce computing time
bkRNA18.AUCVCs1 <- gridAUCVC(data= bkRNA18,  dataType="bk",  TU= 1,
nonzeroRatios= c(0.7,0.8,0.9,1));

# Look at the maximum AUCVC for TU
bkRNA18.AUCVCs1;

# Find the parameters used by TU when it achieved the maximum AUCVC in bestPara.txt
bestPara <- read.table(file='bestPara.txt', header=TRUE);
bestPara;
```

```
# Using the parameters to produce the TU normalization factor
tu <- getFactors(data = bkRNA18, method = "TU", pre_ratio=1, lower_trim=0.2,
upper_trim=0.6);

# Produce the TU-normalized gene expression matrix
TU_bk.matrix <- getNormMatrix(data = bkRNA18, norm.factors = tu);
```

Results:

```
> bkRNA18.AUCVCs1          > bestPara;
      NonzeroRatio      TU      nonzeroRatio pre_ratio lower_trim upper_trim
[1,]          0.7 0.8058438 1          0.7          1          0.2          0.6
[2,]          0.8 0.8209854 2          0.8          1          0.2          0.6
[3,]          0.9 0.8315836 3          0.9          1          0.2          0.6
[4,]          1.0 0.8262366 4          1.0          1          0.2          0.6

> tu
col3616_1 col3816_3 col3916_5 col4016_7 col4416_9 col4516_11 col4716_13 col4816_97 col5216_17 col3616_2
0.94992 0.86526 1.05861 1.49176 0.95182 1.05500 1.25776 0.84202 0.94466 1.19128
col3816_4 col3916_6 col4016_8 col4416_10 col4516_12 col4716_14 col4816_98 col5216_18
0.74389 1.25301 0.78468 0.92516 0.94365 1.09951 0.72688 1.26144

> head(TU_bk.matrix)
      col3616_1 col3816_3 col3916_5 col4016_7 col4416_9 col4516_11 col4716_13 col4816_97 col5216_17
DDX11L1 0.00000 0 0 0 0 0 0.00000 0.00000 0
WASH7P(1) 5.69952 0 0 0 0 0 3.77328 2.52606 0
MIR6859-1 0.00000 0 0 0 0 0 0.00000 0.00000 0
MIR1302-2 0.00000 0 0 0 0 0 0.00000 0.00000 0
FAM138A 0.00000 0 0 0 0 0 0.00000 0.00000 0
OR4G4P 0.00000 0 0 0 0 0 0.00000 0.00000 0
      col3616_2 col3816_4 col3916_6 col4016_8 col4416_10 col4516_12 col4716_14 col4816_98 col5216_18
DDX11L1 0.00000 0 0 0.00000 0 0.0000 0.00000 0.00000 0
WASH7P(1) 3.57384 0 0 2.35404 0 1.8873 3.29853 1.45376 0
MIR6859-1 0.00000 0 0 0.00000 0 0.0000 0.00000 0.00000 0
MIR1302-2 0.00000 0 0 0.00000 0 0.0000 0.00000 0.00000 0
FAM138A 0.00000 0 0 0.00000 0 0.0000 0.00000 0.00000 0
OR4G4P 0.00000 0 0 0.00000 0 0.0000 0.00000 0.00000 0
```

1.E Using AUCVC for simply evaluation (single-cell data)

```
# Grid of nonzero ratios to produce AUCVCs for 10 methods
# Since RLE is identical to DESeq, only normalization factors of 9 methods are required
# scRNA663_factors contains these 9 normalization factors
# TN can use one of user's normalization factors
scRNA663.AUCVCs <- gridAUCVC(data= scRNA663, dataType="sc", HG7=
scRNA663_factors$HG7, ERCC= scRNA663_factors$ERCC, TN= scRNA663_factors$TN,
TC= scRNA663_factors$TC, CR= scRNA663_factors$CR, NR= scRNA663_factors$NR,
DESeq= scRNA663_factors$DESeq, UQ= scRNA663_factors$UQ, TMM=
scRNA663_factors$TMM, nonzeroRatios= c(0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9));
```

Results(2A [1]): This need be compared with the AUCVCs of TU in 1.C

```
> scRNA663.AUCVCs
      NonzeroRatio      HG7      ERCC      TN      TC      CR      NR      DESeq      UQ      TMM
[1,]          0.2 0.9129940 0.7936516 0.6347332 0.7667326 0.7565782 0.8113003 0.7213953 0.7283275 0.7538218
[2,]          0.3 0.8259841 0.7176610 0.5550812 0.6945485 0.6630667 0.7298991 0.6877782 0.7101758 0.6812195
[3,]          0.4 0.7406932 0.6526663 0.5700770 0.6422468 0.6287480 0.6593379 0.6192707 0.6344723 0.6201311
[4,]          0.5 0.6939399 0.5977384 0.5860197 0.5771835 0.5815261 0.6234523 0.6252815 0.6202034 0.6053893
[5,]          0.6 0.6974026 0.6182677 0.5997305 0.6068003 0.5990466 0.6654869 0.6433130 0.6319271 0.6417188
[6,]          0.7 0.6673185 0.6055685 0.5186895 0.5801492 0.5608024 0.6240927 0.5966008 0.6115363 0.5750282
[7,]          0.8 0.7098559 0.6017867 0.5228242 0.5870461 0.5766571 0.6704035 0.6156916 0.6079683 0.6200288
[8,]          0.9 0.7809945 0.6682597 0.6224033 0.6643094 0.6617680 0.7468232 0.7263260 0.7180663 0.7178177
```

1.F Using AUCVC for simply evaluation (bulk data)

```
# Grid of nonzero ratios to produce AUCVCs for 10 methods
# Since RLE is identical to DESeq, only normalization factors of 9 methods are required
# scRNA18_factors contains these 9 normalization factors
# TN can use one of user's normalization factors
bkRNA18.AUCVCs <- gridAUCVC(data= bkRNA18, dataType="bk",
HG7=bkRNA18_factors$HG7, ERCC=bkRNA18_factors$ERCC, TN=bkRNA18_factors$TN,
TC=bkRNA18_factors$TC, CR=bkRNA18_factors$CR, NR=bkRNA18_factors$NR,
DESeq=bkRNA18_factors$DESeq, UQ=bkRNA18_factors$UQ,
TMM=bkRNA18_factors$TMM, GAPDH=bkRNA18_factors$GAPDH, nonzeroRatios=
c(0.7,0.8,0.9,1));
```

Results(**2B[1]**): This need be compared with the AUCVCs of TU in **1.D**

```
> bkRNA18.AUCVCs
      NonzeroRatio      HG7      ERCC      TN      TC      CR
[1,]      0.7 0.7666776 0.7999860 0.8012492 0.8000012 0.8000289
[2,]      0.8 0.7720594 0.8151265 0.8162985 0.8151376 0.8151504
[3,]      0.9 0.7777684 0.8261407 0.8280351 0.8261279 0.8260975
[4,]      1.0 0.7610417 0.8198831 0.8215724 0.8198658 0.8198135
      NR      DESeq      UQ      TMM      GAPDH
[1,] 0.8028034 0.8033671 0.8031939 0.8030954 0.7330331
[2,] 0.8162108 0.8182389 0.8180368 0.8180350 0.7435438
[3,] 0.8279361 0.8274207 0.8256893 0.8272121 0.7240643
[4,] 0.8214762 0.8220948 0.8203017 0.8218484 0.7047161
```

2 Normalization with complete evaluation

2.A Using AUCVC for complete evaluation (single-cell data)

```
# Compared to 1.E, the TU method need be added for comparison (more time-consuming)
# One more parameter need be set (TU=1) for the function gridAUCVC
# TN can use one of user's normalization factors
scRNA663.AUCVCs1 <- gridAUCVC(data= scRNA663, dataType="sc", HG7=
scRNA663_factors$HG7, ERCC= scRNA663_factors$ERCC, TN= scRNA663_factors$TN,
TC= scRNA663_factors$TC, CR= scRNA663_factors$CR, NR= scRNA663_factors$NR,
DESeq= scRNA663_factors$DESeq, UQ= scRNA663_factors$UQ, TMM=
scRNA663_factors$TMM, TU= 1, nonzeroRatios= c(0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9));

# Look at the maximum AUCVC for TU
scRNA663.AUCVCs1;

# Find the parameters used by TU when it achieved the maximum AUCVC in bestPara.txt
bestPara <- read.table(file='bestPara.txt', header=TRUE);
bestPara;
```

```
# Using the parameters to produce the TU normalization factor
tu <- getFactors(data = scRNA663, method = "TU", pre_ratio=0.5, lower_trim=0.05,
upper_trim=0.65);
```

This section is optional

```
# NCS, ES and SCnorm are not integrated into NormExpression
# Produce the NCS-normalized gene expression matrix (Nonzero ratio=0.2)
# (fraction, lower_trim_inclusive, upper_trim_inclusive) is from bestPara
# For nonzero ratios from 0.3 to 0.9, users need iteratively produce a new matrix
./normalize.pl --infile scRNA663.txt --outfile NCS --method net --verbose 1 --fraction 0.5 --
lower_trim_inclusive 5 --upper_trim_inclusive 65 --logmaxmincutoff 10 --limit_genes 0 --
is_bulk 0 > NCS.log 2>&1 &

# Produce the ES-normalized gene expression matrix (Nonzero ratio=0.2)
# (fraction, lower_trim, upper_trim) is from bestPara
./normalize.pl --infile scRNA663.txt --outfile ES --solution_file solutions_sc.tab --method
evolution_strategy --verbose 1 --fraction 0.5 --lower_trim 5 --upper_trim 65 --CoV_cutoff 0.8 -
-all_genes 0 --time_to_spend 800000 --populationsize 11 --roundswithoutimprovement 10 --
is_bulk 0 > ES.log 2>&1 &

# Produce AUCVCs for NCS, ES and SCnorm (Nonzero ratio=0.2)
NCS.matrix <- getNormMatrix(scRNA663, scRNA663_factors$NCS);
ES.matrix <- getNormMatrix(scRNA663, scRNA663_factors$ES);
scRNA663.AUCVCs2 <- gridAUCVC4Matrices(None= scRNA663, NCS=NCS.matrix,
ES=ES.matrix, SCnorm= scRNA663.SCnorm, nonzeroRatios= 0.2);

# Produce AUCVCs for NCS, ES and SCnorm (Nonzero ratio=0.3 to 0.9)
# Iteratively add new_row to the result matrix scRNA663.AUCVCs2
scRNA663.AUCVCs2=rbind(scRNA663.AUCVCs2, new_row);
.....
# Combine two matrices
scRNA663.AUCVCs <- cbind(scRNA663.AUCVCs1, scRNA663.AUCVCs2);
```

2.B Using AUCVC for complete evaluation (bulk data)

```
# Compared to 1.F, the TU method need be added for comparison (more time-consuming)
# One more parameter need be set (TU=1) for the function gridAUCVC
# TN can use one of user's normalization factors
bkRNA18.AUCVCs1 <- gridAUCVC(data= bkRNA18, dataType="bk", HG7=
bkRNA18_factors$HG7, ERCC= bkRNA18_factors$ERCC, TN=bkRNA18_factors$TN,
TC=bkRNA18_factors$TC, CR=bkRNA18_factors$CR, NR=bkRNA18_factors$NR,
DESeq=bkRNA18_factors$DESeq, UQ=bkRNA18_factors$UQ,
TMM=bkRNA18_factors$TMM, TU= 1, GAPDH=bkRNA18_factors$GAPDH,
nonzeroRatios= c(0.7, 0.8, 0.9, 1));
```

```
# Look at the maximum AUCVC for TU
bkRNA18.AUCVCs1;

# Find the parameters used by TU when it achieved the maximum AUCVC in bestPara.txt
bestPara <- read.table(file='bestPara.txt', header=TRUE);
bestPara;

# Using the parameters to produce the TU normalization factor
tu <- getFactors(data = bkRNA18, method = "TU", pre_ratio=1, lower_trim=0.2,
upper_trim=0.6);
```

This section is optional

```
# NCS, ES and SCnorm are not integrated into NormExpression
# Produce the NCS-normalized gene expression matrix (Nonzero ratio=1)
# (fraction, lower_trim_inclusive, upper_trim_inclusive) is from bestPara
# For nonzero ratios from 0.7 to 0.9, users need iteratively produce a new matrix
./normalize.pl --infile bkRNA18.txt --outfile NCS --method net --verbose 1 --fraction 1 --
lower_trim_inclusive 20 --upper_trim_inclusive 60 --logmaxmincutoff 3 --limit_genes 0 --
is_bulk 1 > NCS.log 2>&1 &

# Produce the ES-normalized gene expression matrix (Nonzero ratio=1)
# (fraction, lower_trim, upper_trim) is from bestPara
./normalize.pl --infile bkRNA18.txt --outfile ES --solution_file solutions_bk.tab --method
evolution_strategy --verbose 1 --fraction 1 --lower_trim 20 --upper_trim 60 --CoV_cutoff 0.25
--all_genes 0 --time_to_spend 800000 --populationsize 11 --roundswithoutimprovement 10 --
is_bulk 1 > ES.log 2>&1 &

# Produce AUCVCs for NCS, ES and SCnorm (Nonzero ratio=1)
NCS.matrix <- getNormMatrix(bkRNA18, bkRNA18_factors$NCS);
ES.matrix <- getNormMatrix(bkRNA18, bkRNA18_factors$ES);
bkRNA18.AUCVCs2 <- gridAUCVC4Matrices(None= bkRNA18, NCS=NCS.matrix,
ES=ES.matrix, SCnorm= bkRNA18.SCnorm, nonzeroRatios= 1);

# Produce AUCVCs for NCS, ES and SCnorm (Nonzero ratio=0.7 to 0.9)
# Iteratively add new_row to the result matrix scRNA663.AUCVCs2
bkRNA18.AUCVCs2=rbind(bkRNA18.AUCVCs2,new_row);
.....
# Combine two matrices
bkRNA18.AUCVCs <- cbind(bkRNA18.AUCVCs1, bkRNA18.AUCVCs2);
```

2.C Using mSCC for complete evaluation (single-cell data)

```
# Produce mSCCs for 11 methods (Nonzero ratio=0.2)
# Since RLE is identical to DESeq, only normalization factors of 10 methods are required
```

```
# scRNA663_factors contains normalization factors of 9 methods
# The TU normalization factor is from tu (2.A)
# TN can use one of user's normalization factors
# (pre_ratio, lower_trim, upper_trim) is from bestPara (2.A)
# Users can produce mSCCs for 11 methods (Nonzero ratio=0.3-0.9) but not necessarily
scRNA663.cors1 <- gatherCors(data= scRNA663, cor_method="spearman", HG7=
scRNA663_factors$HG7, ERCC= scRNA663_factors$ERCC, TN= scRNA663_factors$TN,
TC= scRNA663_factors$TC, CR= scRNA663_factors$CR, NR= scRNA663_factors$NR,
DESeq= scRNA663_factors$DESeq, UQ= scRNA663_factors$UQ, TMM=
scRNA663_factors$TMM, TU= tu, pre_ratio=0.5, lower_trim=0.05, upper_trim=0.65,
rounds=1000000);
```

This section is optional

```
# Produce mSCCs for NCS, ES and SCnorm
# (pre_ratio, lower_trim, upper_trim) is from bestPara (2.A)
NCS.matrix <- getNormMatrix(scRNA663, scRNA663_factors$NCS);
ES.matrix <- getNormMatrix(scRNA663, scRNA663_factors$ES);
scRNA663.cors2 <- gatherCors4Matrices(None= scRNA663, NCS=NCS.matrix,
ES=ES.matrix, SCnorm= scRNA663.SCnorm, raw_matrix= scRNA663,
cor_method="spearman", pre_ratio=0.5, lower_trim=0.05, upper_trim=0.65, rounds=1000000);

# combine all mSCCs (2C 第 1 行[1])
scRNA663.cors <- rbind(scRNA663.cors1, scRNA663.cors2);
scRNA663.cor.medians <- getCorMedians(scRNA663.cors);
```

2.D Using mSCC for complete evaluation (bulk data)

```
# Produce mSCCs for 11 methods (Nonzero ratio=1)
# Since RLE is identical to DESeq, only normalization factors of 10 methods are required
# bkRNA18_factors contains normalization factors of 9 methods
# The TU normalization factor is from tu (2.B)
# TN can use one of user's normalization factors
# (pre_ratio, lower_trim, upper_trim) is from bestPara (2.B)
# Users can produce mSCCs for 11 methods (Nonzero ratio=0.7-0.9) but not necessarily
bkRNA18.cors1 <- gatherCors(data= bkRNA18, cor_method="spearman", HG7=
bkRNA18_factors$HG7, ERCC= bkRNA18_factors$ERCC, TN= bkRNA18_factors$TN, TC=
bkRNA18_factors$TC, CR= bkRNA18_factors$CR, NR= bkRNA18_factors$NR, DESeq=
bkRNA18_factors$DESeq, UQ= bkRNA18_factors$UQ, TMM= bkRNA18_factors$TMM,
TU= tu, GAPDH= bkRNA18_factors$GAPDH, pre_ratio=1, lower_trim=0.2, upper_trim=0.6,
rounds=1000000);
```

This section is optional

```
# Produce mSCCs for NCS, ES and SCnorm
# (pre_ratio, lower_trim, upper_trim) is from bestPara (2.A)
NCS.matrix <- getNormMatrix(bkRNA18, bkRNA18_factors$NCS);
```

```

ES.matrix <- getNormMatrix(bkRNA18, bkRNA18_factors$ES);
bkRNA18.cors2 <- gatherCors4Matrices(None= bkRNA18, NCS=NCS.matrix, ES=ES.matrix,
SCnorm= bkRNA18.SCnorm, raw_matrix=bkRNA18, cor_method="spearman", pre_ratio=1,
lower_trim=0.2, upper_trim=0.6, rounds=1000000);

# combine all mSCCs (2D 第4行[1])
bkRNA18.cors <- rbind(bkRNA18.cors1, bkRNA18.cors2);
bkRNA18.cor.medians <- getCorMedians(bkRNA18.cors);

```

3 Visualization of evaluation results

3.A CV threshold Curve (single-cell data)

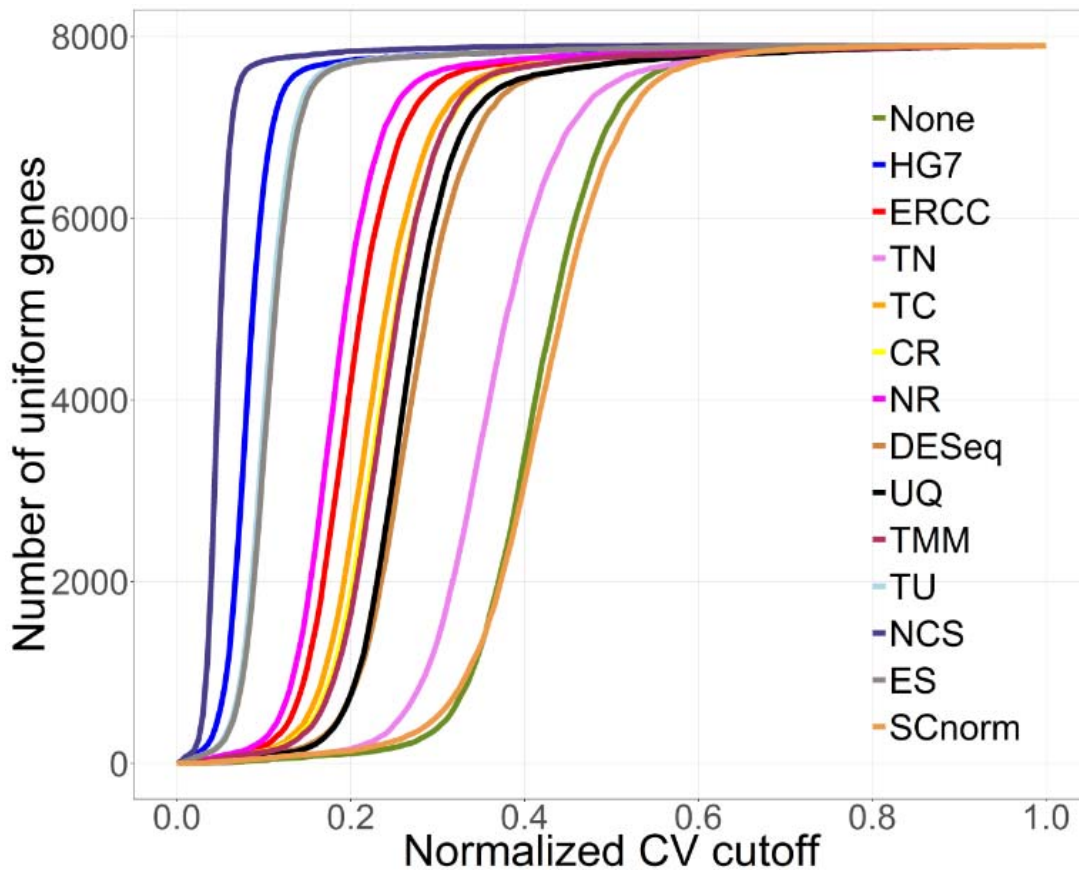
```

# scRNA663_factors contains all the pre-calculated normalization factors (Nonzero ratio=0.2)
scRNA663.cv_uniform1 <- gatherCVs(data= scRNA663, nonzeroRatio= 0.2, HG7=
scRNA663_factors$HG7, ERCC= scRNA663_factors$ERCC, TN= scRNA663_factors$TN,
TC= scRNA663_factors$TC, CR= scRNA663_factors$CR, NR= scRNA663_factors$NR,
DESeq= scRNA663_factors$DESeq, UQ= scRNA663_factors$UQ, TMM=
scRNA663_factors$TMM, TU= scRNA663_factors$TU);
NCS.matrix <- getNormMatrix(scRNA663, scRNA663_factors$NCS);
ES.matrix <- getNormMatrix(scRNA663, scRNA663_factors$ES);
scRNA663.cv_uniform2 <- gatherCVs4Matrices(None= scRNA663, NCS=NCS.matrix,
ES=ES.matrix, SCnorm= scRNA663.SCnorm, raw_matrix=scRNA663, nonzeroRatio=0.2);
scRNA663.cv_uniform <- rbind(scRNA663.cv_uniform1, scRNA663.cv_uniform2);

# plot
tiff(file = "scRNA663_cv.tif", res=300, compression =
"lzw",width=(1200*4.17),height=(960*4.17));
plotCVs(scRNA663.cv_uniform, methods=c("None", "HG7", "ERCC", "TN", "TC", "CR",
"NR", "DESeq", "UQ", "TMM", "TU", "NCS", "ES", "SCnorm"), legend.position=c(.85, .48));
dev.off();

```

Results(3A[1]):

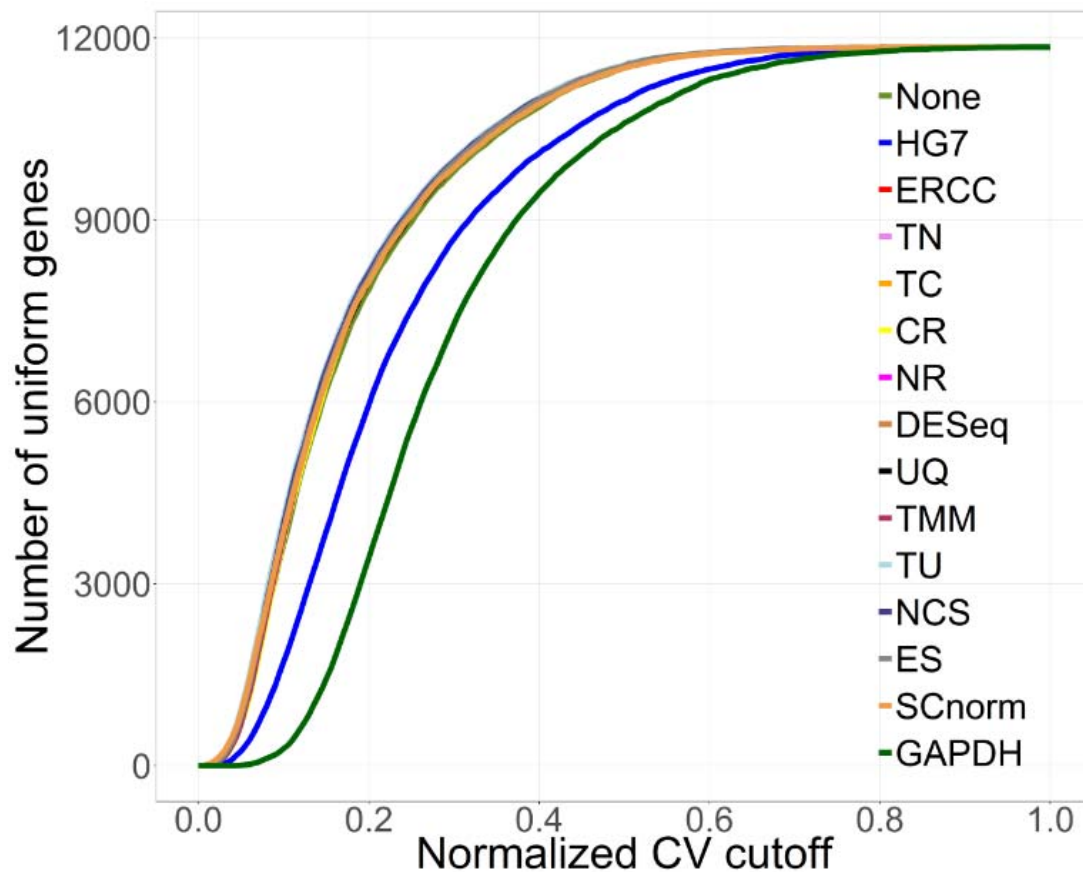


3.B CV threshold Curve (bulk data)

```
# bkRNA18_factors contains all the pre-calculated normalization factors (Nonzero ratio=1)
bkRNA18.cv_uniform1 <- gatherCVs(data= bkRNA18, nonzeroRatio= 1, HG7=
bkRNA18_factors$HG7, ERCC= bkRNA18_factors$ERCC, TN= bkRNA18_factors$TN, TC=
bkRNA18_factors$TC, CR= bkRNA18_factors$CR, NR= bkRNA18_factors$NR, DESeq=
bkRNA18_factors$DESeq, UQ= bkRNA18_factors$UQ, TMM= bkRNA18_factors$TMM,
TU= bkRNA18_factors$TU, GAPDH = bkRNA18_factors$GAPDH);
NCS.matrix <- getNormMatrix(bkRNA18, bkRNA18_factors$NCS);
ES.matrix <- getNormMatrix(bkRNA18, bkRNA18_factors$ES);
bkRNA18.cv_uniform2 <- gatherCVs4Matrices(None= bkRNA18, NCS=NCS.matrix,
ES=ES.matrix, SCnorm= bkRNA18.SCnorm, raw_matrix =bkRNA18, nonzeroRatio=1);
bkRNA18.cv_uniform <- rbind(bkRNA18.cv_uniform1, bkRNA18.cv_uniform2);

# plot
tiff(file = "bkRNA18_cv.tif", res=300, compression =
"lzw",width=(1200*4.17),height=(960*4.17));
plotCVs(bkRNA18.cv_uniform, methods=c("None", "HG7", "ERCC", "TN", "TC", "CR",
"NR", "DESeq", "UQ", "TMM", "TU", "NCS", "ES", "SCnorm", "GAPDH"),
legend.position=c(.85, .48));
dev.off();
```

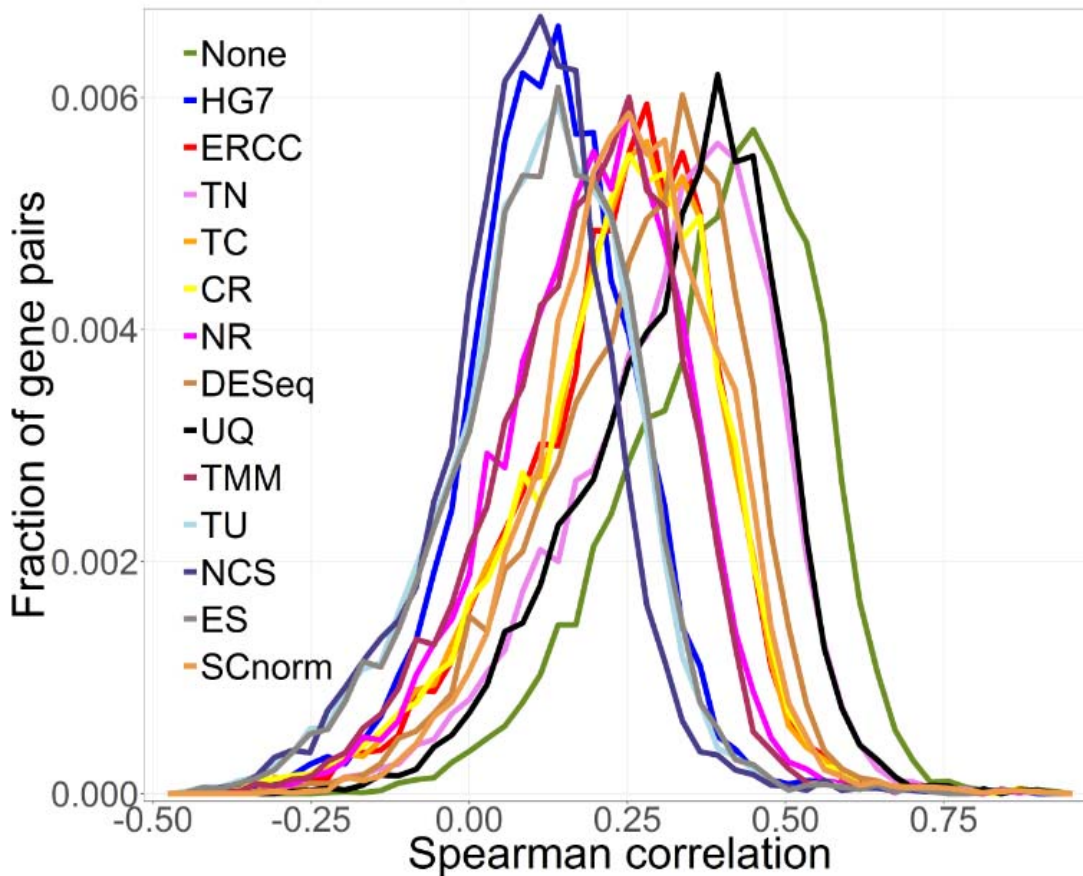
Results(**3B[1]**):



3.C Distribution of SCCs between genes (single-cell data)

```
# scRNA663.cors is from 2.C (Nonzero ratio=0.2)
tiff(file      ="      scRNA663_sp.tif",      res=300,      compression      =
"lzw",width=(1200*4.17),height=(960*4.17));
plotCors(scRNA663.cors, methods=c("None", "HG7", "ERCC", "TN", "TC", "CR", "NR",
"DESeq", "UQ", "TMM", "TU", "NCS", "ES", "SCnorm"), legend.position=c(.15, .56))
dev.off();
```

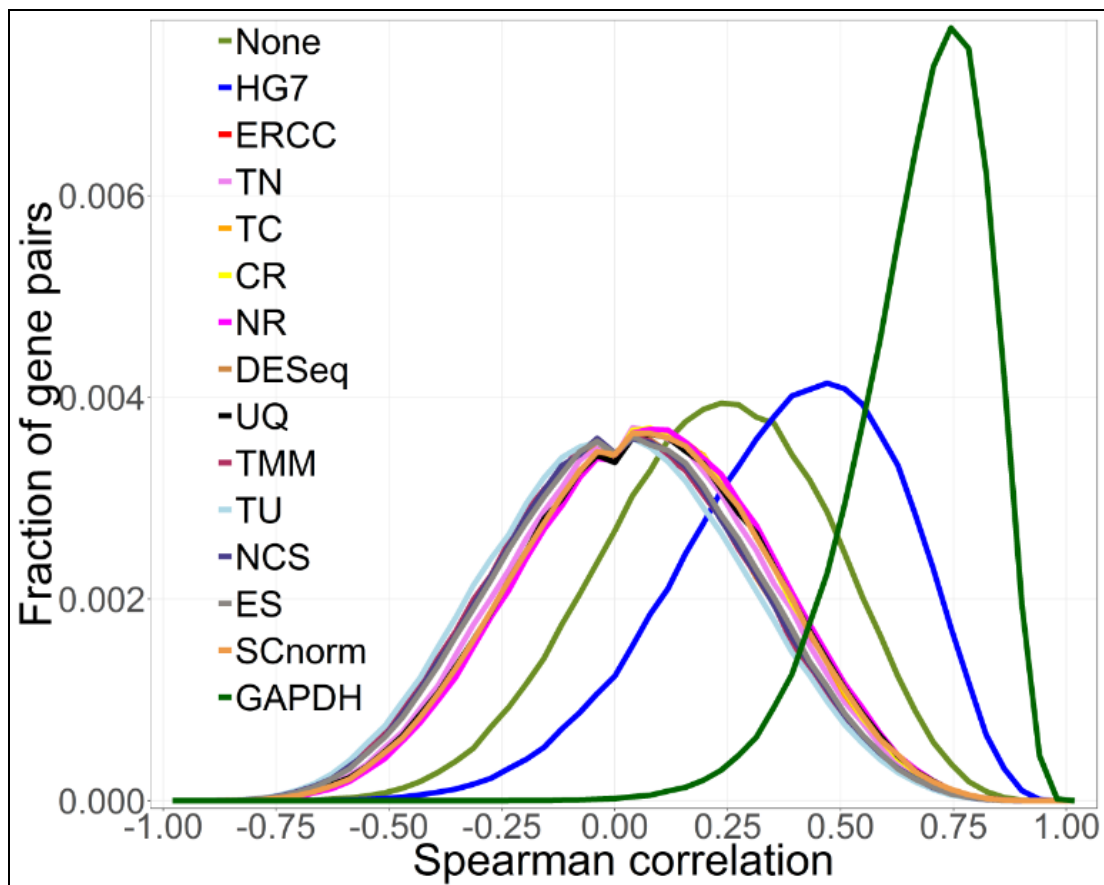
Results(**3C[1]**):



3.D Distribution of SCCs between genes (bulk data)

```
# bkRNA18.cors is from 2.D (Nonzero ratio=1)
tiff(file      =      "bkRNA18_sp.tif",      res=300,      compression      =
"lzw",width=(1200*4.17),height=(960*4.17));
plotCors(bkRNA18.cors, methods=c("None", "HG7", "ERCC", "TN", "TC", "CR", "NR",
"DESeq", "UQ", "TMM", "TU", "NCS", "ES", "SCnorm", "GAPDH"),
legend.position=c(.15, .56));
dev.off();
```

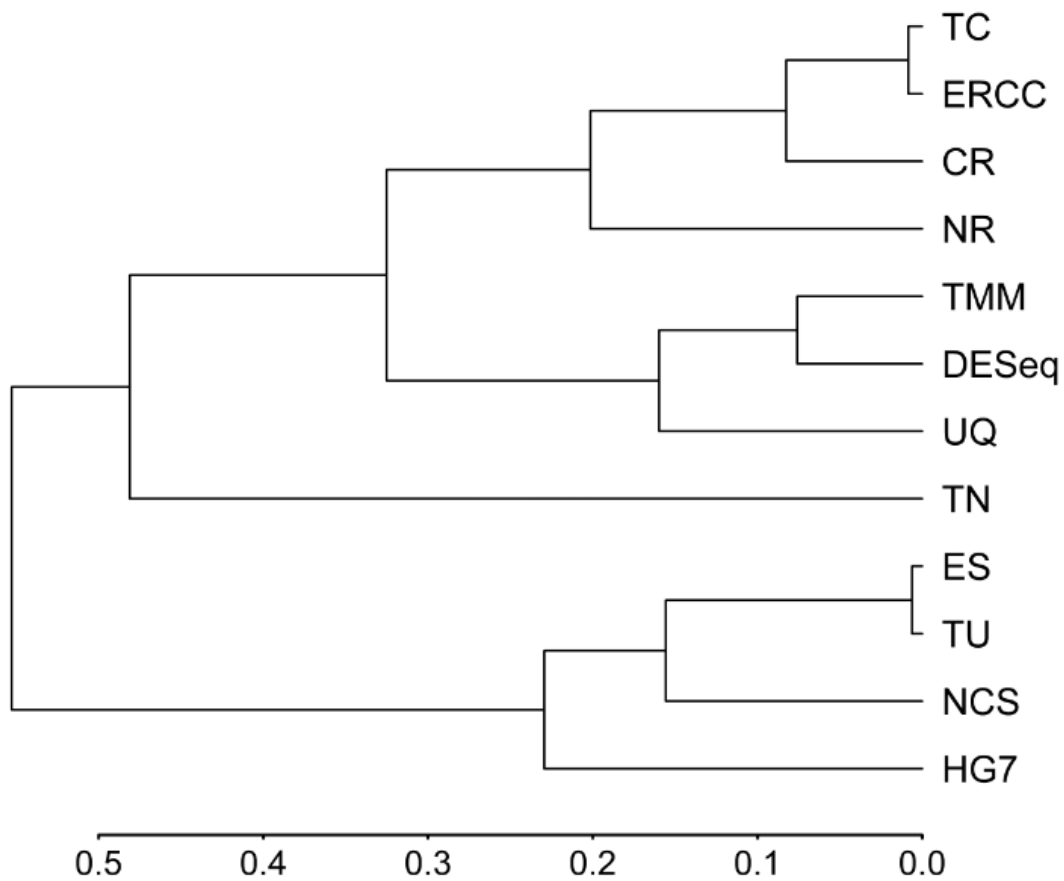
Results(**3D[1]**):



3.E Hierarchical clustering of normalization factors (single-cell data)

```
tiff(file      =      "scRNA663_hc.tif",      res=300,      compression      =
"lzw",width=(2360*4.17),height=(1960*4.17));
plotHC(scRNA663_factors, method="spearman", mar=c(9,1,0,20))
dev.off();
```

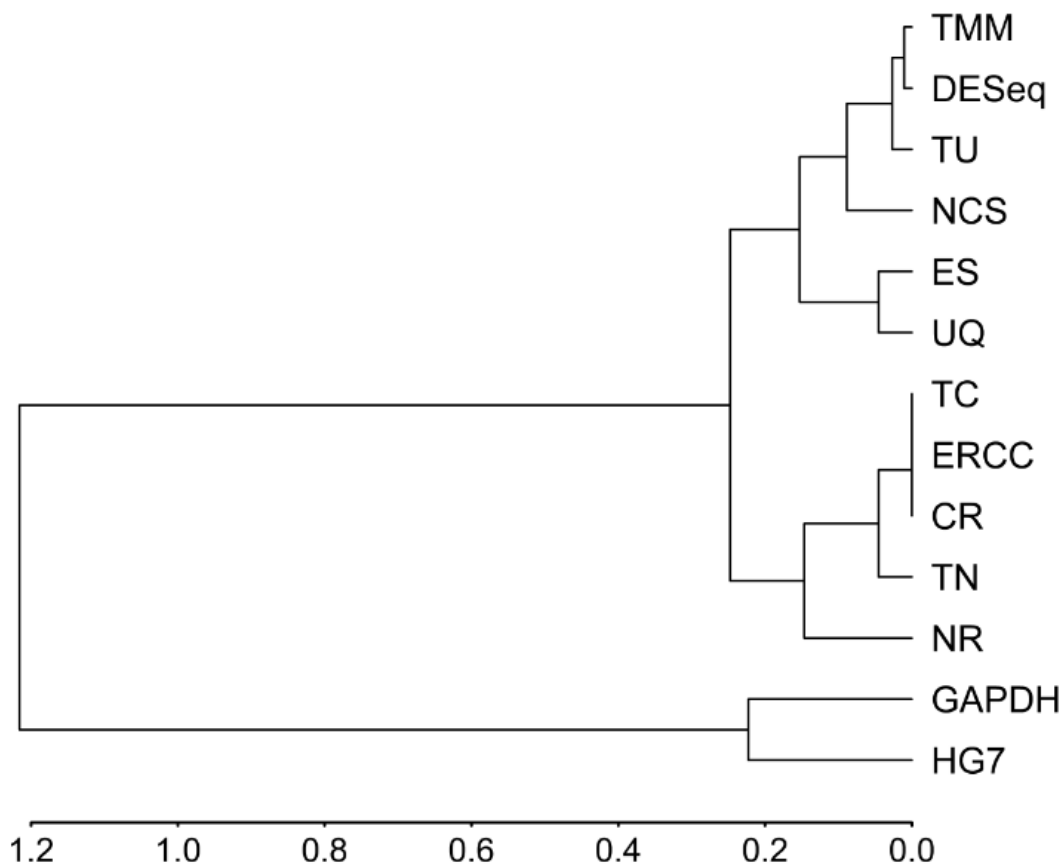
Results(**3E[1]**):



3.F Hierarchical clustering of normalization factors (bulk data)

```
tiff(file = "bkRNA18_hc.tif", res=300, compression =  
"lzw",width=(2360*4.17),height=(1960*4.17));  
plotHC(bkRNA18_factors, method="spearman", mar=c(9,1,0,20))  
dev.off();
```

Results(**3F[1]**):



4 How to evaluate many users' methods

4.A For methods with normalization factors (single-cell data)

```
# assume users' methods are named method1-9
# produce normalization factors named factor1-9
# run the same command line as 2.A
scRNA663.AUCVCs <- gridAUCVC(data= scRNA663, dataType="sc", HG7= factor1,
ERCC= factor2, TN= factor3, TC= factor4, CR= factor5, NR= factor6, DESeq= factor7, UQ=
factor8, TMM= factor9, TU= 1, nonzeroRatios= c(0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9));

# Look at the maximum AUCVC for TU
scRNA663.AUCVCs1;

# Find the parameters used by TU when it achieved the maximum AUCVC in bestPara.txt
bestPara <- read.table(file='bestPara.txt', header=TRUE);
bestPara;

# Using the parameters to produce the TU normalization factor
tu <- getFactors(data = scRNA663, method = "TU", pre_ratio=0.5, lower_trim=0.05,
upper_trim=0.65);
```

```
# Produce mSCCs for 11 methods (Nonzero ratio=0.2)
scRNA663.cors1 <- gatherCors(data= scRNA663, cor_method="spearman", HG7= factor1,
ERCC= factor2, TN= factor3, TC= factor4, CR= factor5, NR= factor6, DESeq= factor7, UQ=
factor8, TMM= factor9, TU= tu, pre_ratio=0.5, lower_trim=0.05, upper_trim=0.65,
rounds=1000000);
```

4.B For a general usage (single-cell data)

```
# assume users' methods are named method1-9
# produce normalized gene expression matrices named matrix1-9
scRNA663.AUCVCs1 <- gridAUCVC4Matrices(None= scRNA663, m1 = matrix1, m2 =
matrix2, m3 = matrix3, m4 = matrix4, m5 = matrix5, m6 = matrix6, m7 = matrix7, m8 =
matrix8, m9 = matrix9, nonzeroRatios= c(0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9));

# Grid of non-zero ratios to produce AUCVCs for TU and it is time-consuming
scRNA663.AUCVCs2 <- gridAUCVC(data= scRNA663, dataType="sc", TU= 1,
nonzeroRatios= c(0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9));

# Look at the maximum AUCVC for TU
scRNA663.AUCVCs2;

# Find the parameters used by TU when it achieved the maximum AUCVC in bestPara.txt
bestPara <- read.table(file='bestPara.txt', header=TRUE);
bestPara;

# Using the parameters to produce the TU normalization factor
tu <- getFactors(data = scRNA663, method = "TU", pre_ratio=0.5, lower_trim=0.05,
upper_trim=0.65);

# Produce the TU-normalized gene expression matrix
TU_sc.matrix <- getNormMatrix(data = scRNA663, norm.factors = tu);

# Produce mSCCs for 11 methods (Nonzero ratio=0.2)
scRNA663.cors <- gatherCors4Matrices(None= scRNA663, m1 = matrix1, m2 = matrix2, m3 =
matrix3, m4 = matrix4, m5 = matrix5, m6 = matrix6, m7 = matrix7, m8 = matrix8, m9 =
matrix9, TU = TU_sc.matrix, raw_matrix= scRNA663, cor_method="spearman",
pre_ratio=0.5, lower_trim=0.05, upper_trim=0.65, rounds=1000000);
```

5 Package versions and FAQ

```
> sessionInfo()
R version 3.4.2 (2017-08-28)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 7 x64 (build 7601) Service Pack 1

Matrix products: default

locale:
 [1] LC_COLLATE=Chinese (Simplified)_People's Republic of China.936  LC_CTYPE=Chinese (Simplified)_People's Republic of China.936
 [3] LC_MONETARY=Chinese (Simplified)_People's Republic of China.936 LC_NUMERIC=C
 [5] LC_TIME=Chinese (Simplified)_People's Republic of China.936

attached base packages:
[2] stats      graphics  grDevices  utils      datasets  methods   base

other attached packages:
[1] dendextend_1.6.0 ggplot2_2.2.1

loaded via a namespace (and not attached):
 [1] flexmix_2.3-14      Rcpp_0.12.15        cluster_2.0.6        whisker_0.3-2        magrittr_1.5         fpc_2.1-11          MASS_7.3-40          munzell_0.4.3        mclust_5.4
 [10] varidateSite_0.2-21 colorspace_1.3-2   lattice_0.20-35      rlang_0.1.6          plyr_1.8.4          preRclus_2.2-6       nnet_7.3-12          grid_3.5.2           gtable_0.2.0
 [19] modeltools_0.2-21   class_7.3-14         lazyeval_0.2.1       tidble_1.4.1         gridExtra_2.3        kernlab_0.8-25       trncluster_0.1-2     varidis_0.4.1        robustbase_0.92-8
 [28] DEoptimR_1.0-8      compiler_3.4.2       pillar_1.1.0         scales_0.5.0         dplyr_0.7.5-7       statstat_3.4.2       rsnorm_1.0-6
```