

NormExpression.....	1
1 标准化与简单评估.....	1
1.A R 包安装与数据准备 .....	1
1.B 如何计算得到标准化因子矩阵（单细胞数据） .....	3
1.C 不做评估的标准化（单细胞数据） .....	3
1.D 不做评估的标准化（Bulk 数据） .....	4
1.E 应用 AUCVC 简单评估（单细胞数据） .....	5
1.F 应用 AUCVC 简单评估（Bulk 数据） .....	5
2 完整评估.....	6
2.A 应用 AUCVC 完整评估（单细胞数据） .....	6
2.B 应用 AUCVC 完整评估（Bulk 数据） .....	7
2.C 应用 mSCC 完整评估（单细胞数据） .....	8
2.D 应用 mSCC 完整评估（Bulk 数据） .....	9
3 评估结果的可视化.....	9
3.A CV 阈值曲线图（单细胞数据） .....	9
3.B CV 阈值曲线图（Bulk 数据） .....	10
3.C 基因间相关系数分布图（单细胞数据） .....	11
3.D 基因间相关系数分布图（Bulk 数据） .....	12
3.E 标准化因子层次聚类图（单细胞数据） .....	13
3.F 标准化因子层次聚类图（Bulk 数据） .....	14
4 如何评估多个用户的方法.....	15
4.A 有标准化因子的情况（单细胞数据） .....	15
4.B 更为普遍的情况（单细胞数据） .....	16
5 R 包版本和常见问题 .....	17

# NormExpression

R 软件包 NormExpression 用于在方法评估的基础上对基因表达数据进行标准化，其支持理论是**最好的评估方法可以同时最大化 uniform 基因的数量和最小化基因间相关系数**。NormExpression 用两种测度评估各类标准化方法，分别是 AUCVC 和 mSCC [1]。

NormExpression 的使用方式为：（1）直接标准化（不做任何评估），推荐使用 TU 方法对单细胞数据（见 **1.C**）或 Bulk 数据（见 **1.D**）进行标准化，TU 方法的优越性参见 [1]；（2）基于简单评估的标准化，只需要用到 AUCVC 测度，通过比较待评估的方法与 10 种常用标准化方法的 AUCVC 大小，分为单细胞数据（见 **1.E**）或 Bulk 数据（见 **1.F**）两种情况；（3）基于完整评估的标准化，需要使用两种测度并检查**两种测度的一致性**，方法比较还要增加 TU（非常耗时），分为单细胞数据（见 **2.AC**）或 Bulk 数据（见 **2.BD**）两种情况。我们推荐使用（3）；如果用户直接使用（1），我们建议同时做（2）以确认（1）；方法评估的结果与[1]中结果比较，还可以检查数据质量。

注意：（1）用户可以通过“复制粘贴”运行本文档中全部 R 代码；（2）使用 NormExpression，请引用以下文章。

[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>

本文档中示例数据集 scRNA663（单细胞数据）和 bkscRNA18（Bulk 数据）可以作为标准数据集评估新方法。这两个数据集采用相同流程建库测序，可以对比**两种数据的一致性**。用户选择**最优方法进行标准化一定要使用自己的数据进行评估**。scRNA663 单细胞数据集要等相关文章发表后释放，期间可以通过 email 索要，请联系高山（gao\_shan@mail.nankai.edu.cn）。索要数据请提供姓名和单位信息，信息不全者不予回复。

## 1 标准化与简单评估

### 1.A R 包安装与数据准备

```
#指定工作目录
setwd("d:/working_dir");

#安装全部 R 包
install.packages("NormExpression");
install.packages("ggplot2");
install.packages("dendextend");
```

```
#加载全部 R 包
library(NormExpression);
library(ggplot2);
library(dendextend);

#读入四个数据
#scRNA663 单细胞数据集要等相关文章发表后释放
data(scRNA663);
#用户可以通过 email 索要数据文件 scRNA663.txt，采用下列语句读入
scRNA663 <- read.table(file='scRNA663.txt', header=TRUE, row.names=1);
#读入预先计算好的标准化因子
data(scRNA663_factors);
#读入 bkRNA18 数据集
data(bkRNA18);
#读入预先计算好的标准化因子
data(bkRNA18_factors);

#计算得到应用 SCnorm 方法标准化后的基因表达矩阵（单细胞数据）
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);

#计算得到应用 SCnorm 方法标准化后的基因表达矩阵（Bulk 数据）
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);
```

结果展示：

```
> 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 如何计算得到标准化因子矩阵（单细胞数据）

```
#计算得到 bkRNA18_factors 的方法相同
#TU、NCS 和 ES 是参数依赖性方法，因此只保留最优参数计算的结果
#HG7、ERCC、TC、CR 和 NR 等方法
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");

#DESeq(RLE)、UQ 和 TMM 等方法
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",
"TM", "TU", "NCS", "ES");
```

## 1.C 不做评估的标准化（单细胞数据）

```
#TU 需要优化计算，消耗很长的计算时间
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));

#查看 TU 产生的 AUCVC 最大值
scRNA663.AUCVCs1;

#从 TU 优化计算的输出文件 bestPara.txt 中找到 AUCVC 最大值对应的参数
bestPara <- read.table(file='bestPara.txt', header=TRUE);
```

```
bestPara;

#根据 AUCVC 最大值对应的参数，计算得到 TU 标准化因子
tu <- getFactors(data = scRNA663, method = "TU", pre_ratio=0.5, lower_trim=0.05,
upper_trim=0.65);

#计算得到应用 TU 方法标准化后的基因表达矩阵
TU_sc.matrix <- getNormMatrix(data = scRNA663, norm.factors = tu);
```

## 1.D 不做评估的标准化 (Bulk 数据)

```
#TU 需要优化计算，消耗很长的计算时间
#nonzeroRatios 可以赋值为 1，以减少计算时间
bkRNA18.AUCVCs1 <- gridAUCVC(data= bkRNA18, dataType="bk", TU= 1,
nonzeroRatios= c(0.7,0.8,0.9,1));

#查看 TU 产生的 AUCVC 最大值
bkRNA18.AUCVCs1;

#从 TU 优化计算的输出文件 bestPara.txt 中找到 AUCVC 最大值对应的参数
bestPara <- read.table(file='bestPara.txt', header=TRUE);
bestPara;

#根据 AUCVC 最大值对应的参数，计算得到 TU 标准化因子
tu <- getFactors(data = bkRNA18, method = "TU", pre_ratio=1, lower_trim=0.2,
upper_trim=0.6);

#计算得到应用 TU 方法标准化后的基因表达矩阵
TU_bk.matrix <- getNormMatrix(data = bkRNA18, norm.factors = tu);
```

结果展示：

```
> bkRNA18.AUCVCs1
      NonzeroRatio      TU
[1,]      0.7 0.8058438  1
[2,]      0.8 0.8209854  2
[3,]      0.9 0.8315836  3
[4,]      1.0 0.8262366  4

> bestPara;
      nonzeroRatio pre_ratio lower_trim upper_trim
[1,]      0.7      1      0.2      0.6
[2,]      0.8      1      0.2      0.6
[3,]      0.9      1      0.2      0.6
[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)
      col13616_1 col13816_3 col13916_5 col14016_7 col14416_9 col14516_11 col14716_13 col14816_97 col15216_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
      col13616_2 col13816_4 col13916_6 col14016_8 col14416_10 col14516_12 col14716_14 col14816_98 col15216_18
DDX11L1      0.00000      0      0      0.00000      0      0.00000      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      0.00000      0.00000      0.00000      0
MIR1302-2    0.00000      0      0      0.00000      0      0.00000      0.00000      0.00000      0
FAM138A      0.00000      0      0      0.00000      0      0.00000      0.00000      0.00000      0
OR4G4P       0.00000      0      0      0.00000      0      0.00000      0.00000      0.00000      0
```

## 1.E 应用 AUCVC 简单评估（单细胞数据）

```
#指定一组 Nonzero ratio, 计算 10 种方法的 AUCVC 值
#RLE 与 DESeq 方法计算结果完全一样, 因此只需要计算 9 种方法的 AUCVC 值
#9 种方法的标准化因子来自 scRNA663_factors
#None（未标准化）作对照
#TN 可以使用用户指定的其他标准化因子进行比较
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));
```

结果展示（见[1]中图 2A）：建议与 1.C 结果一起比较

```
> 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 应用 AUCVC 简单评估（Bulk 数据）

```
#指定一组 Nonzero ratio, 计算 10 种方法的 AUCVC 值
#RLE 与 DESeq 方法计算结果完全一样, 因此只需要计算 9 种方法的 AUCVC 值
#9 种方法的标准化因子来自 scRNA18_factors
#None（未标准化）与 GAPDH 作对照
#TN 可以使用用户指定的其他标准化因子进行比较
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));
```

结果展示(见[1]中 2B)：建议与 1.D 结果一起比较

```
> 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 完整评估

### 2.A 应用 AUCVC 完整评估（单细胞数据）

```
#与 1.E 相比，只需要增加 TU 方法（TU=1）的比较(消耗很长的计算时间)
#TN 可以使用用户指定的其他标准化因子进行比较
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));

#查看 TU 产生的 AUCVC 最大值
scRNA663.AUCVCs1;

#从 TU 优化计算的输出文件 bestPara.txt 中找到 AUCVC 最大值对应的参数
bestPara <- read.table(file='bestPara.txt', header=TRUE);
bestPara;

#根据 AUCVC 最大值对应的参数，计算得到 TU 标准化因子
tu <- getFactors(data = scRNA663, method = "TU", pre_ratio=0.5, lower_trim=0.05,
upper_trim=0.65);
```

#### 以下内容不是必须的

```
#NCS、ES 和 SCnorm 方法未整合至 R 包中，需要单独计算再一起比较
#计算得到应用 NCS 方法标准化后的矩阵（Nonzero ratio=0.2）
#NCS 方法的参数（fraction, lower_trim_inclusive, upper_trim_inclusive）来自 bestPara
#这里只计算 Nonzero ratio=0.2 的情况，其他情况要逐个计算
./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 &

#计算得到应用 ES 方法标准化后的矩阵(Nonzero ratio=0.2)
#(fraction, lower_trim, upper_trim)来自 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 &

#计算得到 NCS、ES 和 SCnorm 三种方法的 AUCVC 值(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);

#计算其他情况下 NCS、ES 和 SCnorm 的 AUCVC 值，每次得到一行新结果 new_row
scRNA663.AUCVCs2=rbind(scRNA663.AUCVCs2, new_row);

.....
#矩阵合并
scRNA663.AUCVCs <- cbind(scRNA663.AUCVCs1, scRNA663.AUCVCs2);
```

## 2.B 应用 AUCVC 完整评估 (Bulk 数据)

```
#与 1.F 相比，只需要增加 TU 方法 (TU=1) 的比较(消耗很长的计算时间)
#TN 可以使用用户指定的其他标准化因子进行比较
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));

#查看 TU 产生的 AUCVC 最大值
bkRNA18.AUCVCs1;

#从 TU 优化计算的输出文件 bestPara.txt 中找到 AUCVC 最大值对应的参数
bestPara <- read.table(file='bestPara.txt', header=TRUE);
bestPara;

#根据 AUCVC 最大值对应的参数，计算得到 TU 标准化因子
tu <- getFactors(data = bkRNA18, method = "TU", pre_ratio=1, lower_trim=0.2,
upper_trim=0.6);
```

### 以下内容不是必须的

```
#NCS、ES 和 SCnorm 方法未整合至 R 包中，需要单独计算再一起比较
#计算得到应用 NCS 方法标准化后的矩阵 (Nonzero ratio=1)
#NCS 方法的参数 (fraction, lower_trim_inclusive, upper_trim_inclusive) 来自 bestPara
#这里只计算 Nonzero ratio=1 的情况，其他情况要逐个计算
./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 &

#计算得到应用 ES 方法标准化后的矩阵(Nonzero ratio=1)
# (fraction, lower_trim, upper_trim)来自 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 &

#计算得到 NCS、ES 和 SCnorm 三种方法的 AUCVC 值(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);

#计算其他情况下 NCS、ES 和 SCnorm 的 AUCVC 值, 每次得到一行新结果 new_row
bkRNA18.AUCVCs2=rbind(bkRNA18.AUCVCs2,new_row);
.....
#矩阵合并
bkRNA18.AUCVCs <- cbind(bkRNA18.AUCVCs1, bkRNA18.AUCVCs2);

```

## 2.C 应用 mSCC 完整评估 (单细胞数据)

```

#计算 11 种方法的斯皮尔曼秩相关系数的中位数 (mSCC)
#RLE 与 DESeq 方法计算结果完全一样, 因此只需要计算 10 种方法的 mSCC
#9 种方法的标准化因子来自 scRNA663_factors, TU 标准化因子来自 2.A 中的 tu
#参数 (pre_ratio, lower_trim, upper_trim) 来自 2.A 中的 bestPara
#这里只计算 Nonzero ratio=0.2 的情况, 其他情况要逐个计算但不是必须
#TN 可以使用用户指定的其他标准化因子进行比较
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);

```

以下内容不是必须的

```

#NCS、ES 和 SCnorm 方法未整合至 R 包中, 需要单独计算再一起比较
#参数 (pre_ratio, lower_trim, upper_trim) 来自 2.A 中的 bestPara
#这里只计算 Nonzero ratio=0.2 的情况, 其他情况要逐个计算但不是必须
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);

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

## 2.D 应用 mSCC 完整评估 (Bulk 数据)

```
#计算 11 种方法的斯皮尔曼秩相关系数的中位数 (mSCC)
#RLE 与 DESeq 方法计算结果完全一样, 因此只需要计算 10 种方法的 mSCC
#9 种方法的标准化因子来自 bkRNA18_factors, TU 标准化因子来自 2.B 中的 tu
#参数 (pre_ratio, lower_trim, upper_trim) 来自 2.B 中的 bestPara
#这里只计算 Nonzero ratio=1 的情况, 其他情况要逐个计算但不是必须
#TN 可以使用用户指定的其他标准化因子进行比较
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);
```

以下内容不是必须的

```
#NCS、ES 和 SCnorm 方法未整合至 R 包中, 需要单独计算再一起比较
#参数 (pre_ratio, lower_trim, upper_trim) 来自 2.B 中的 bestPara
#这里只计算 Nonzero ratio=1 的情况, 其他情况要逐个计算但不是必须
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);

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

## 3 评估结果的可视化

### 3.A CV 阈值曲线图 (单细胞数据)

```
#所用数据来自 scRNA663_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,
```

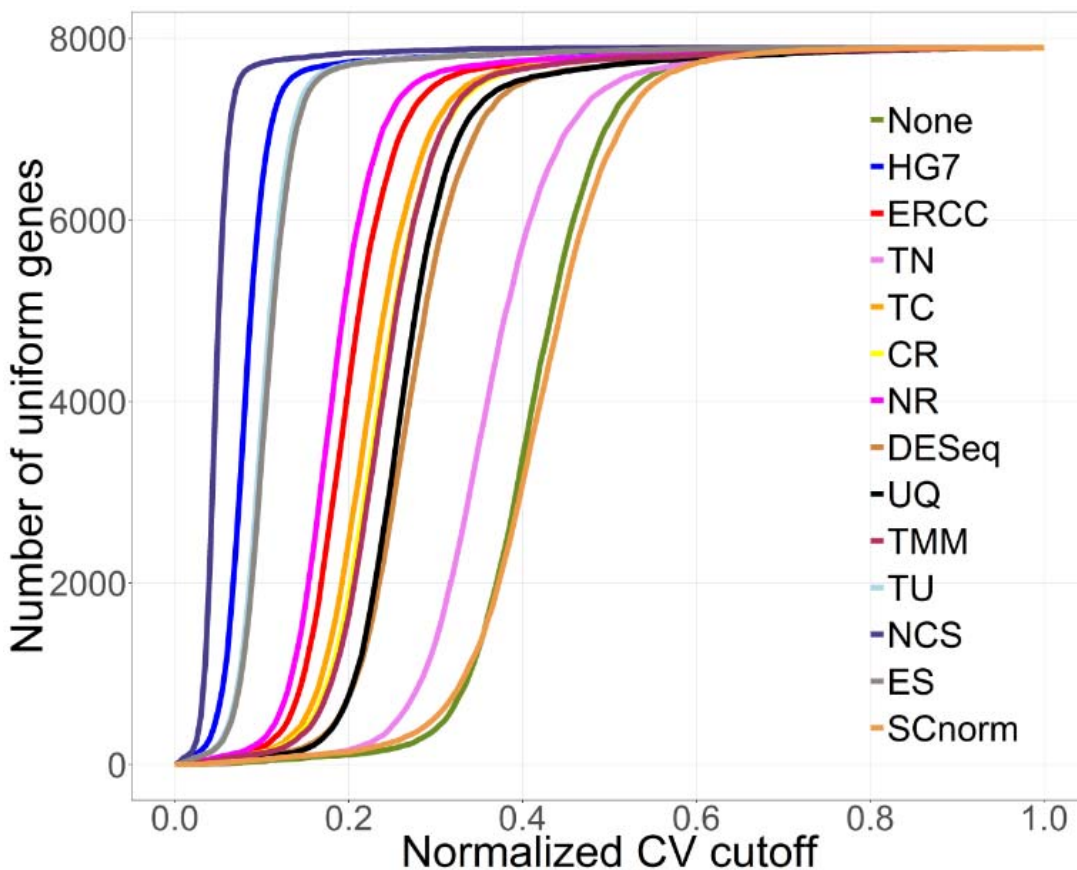
```

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);

#画图
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();

```

结果展示(见[1]中 3A):



### 3.B CV 阈值曲线图 (Bulk 数据)

```

#所用数据来自 bkRNA18_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);

```

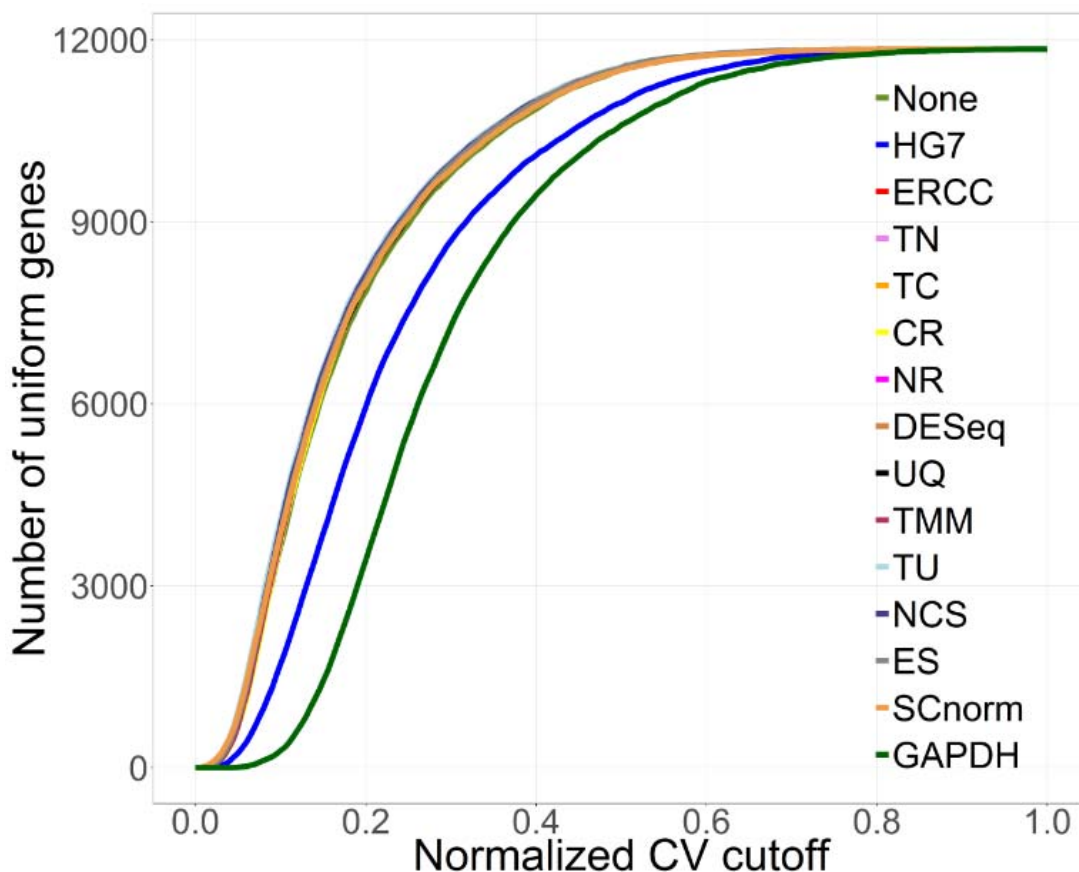
```

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);

#画图
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();

```

结果展示(见[1]图 3B):



### 3.C 基因间相关系数分布图 (单细胞数据)

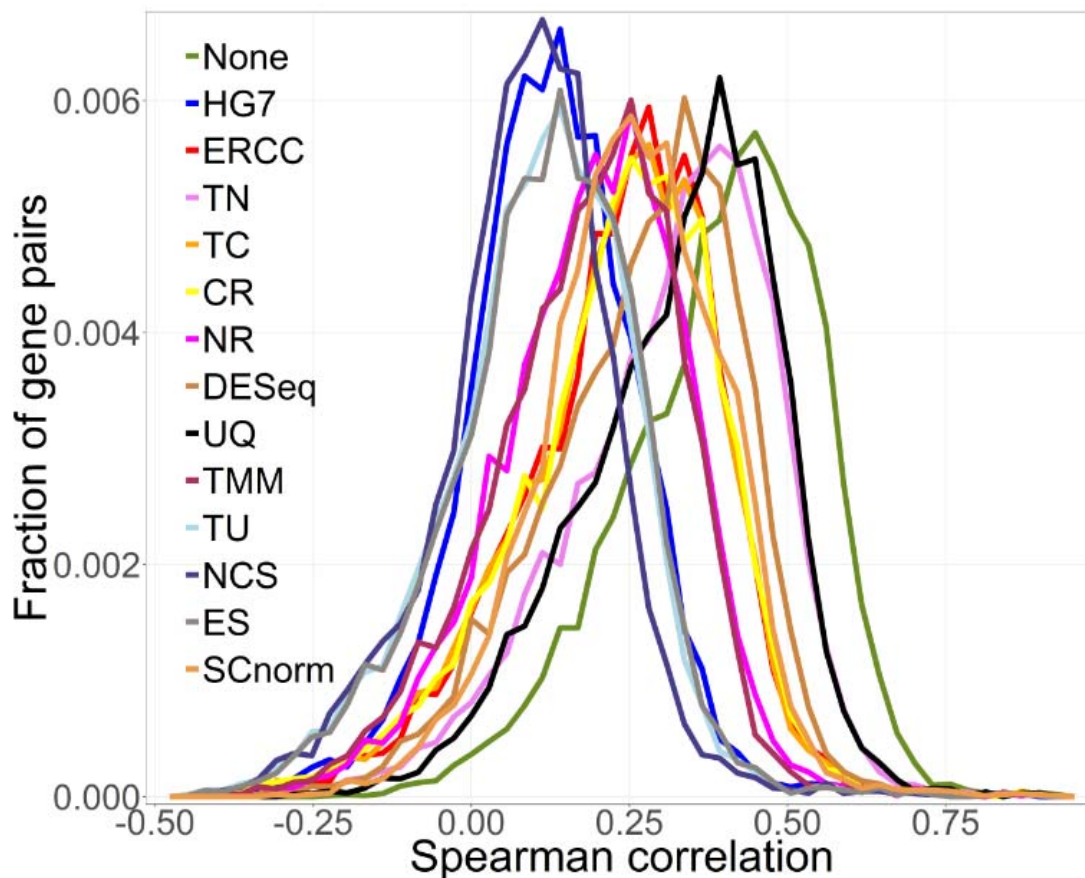
```

#所用数据来自 2.C 的 scRNA663.cors (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();
```

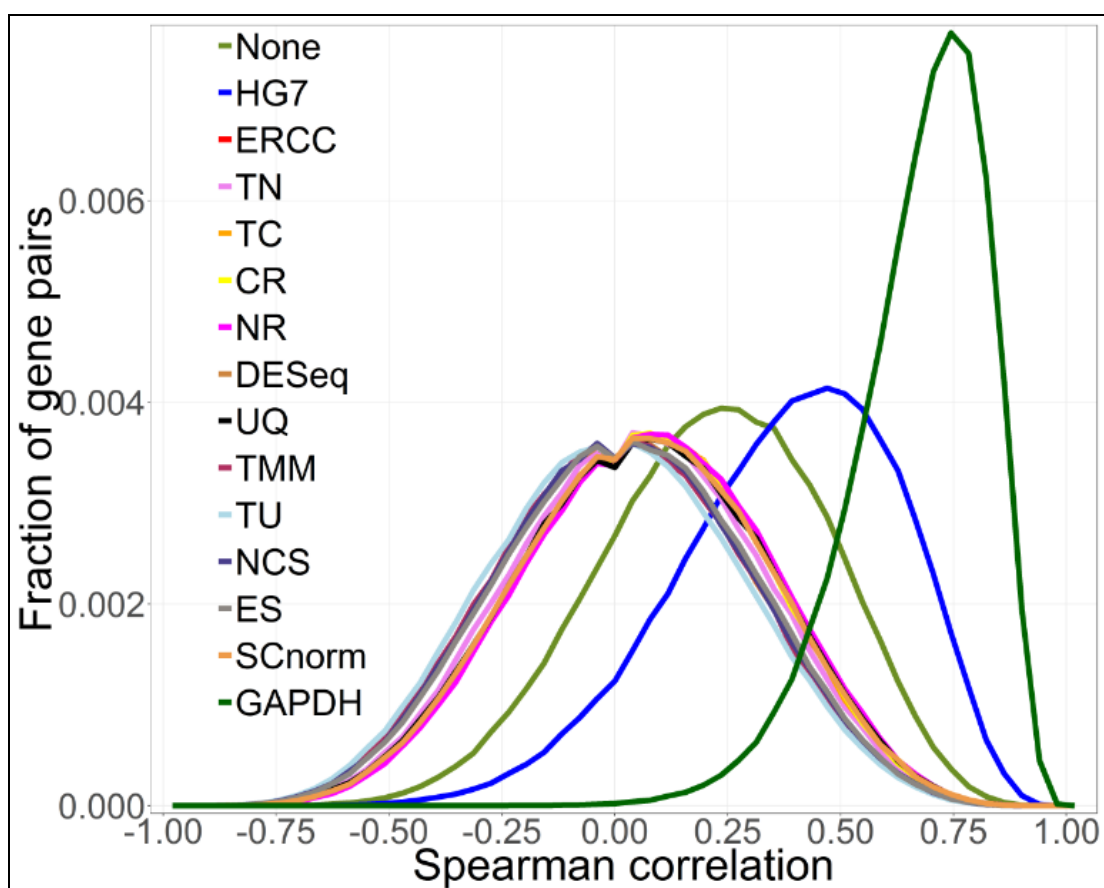
结果展示(见[1]图 3C):



### 3.D 基因间相关系数分布图 (Bulk 数据)

```
#所用数据来自 2.D 的 bkRNA18.cors (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();
```

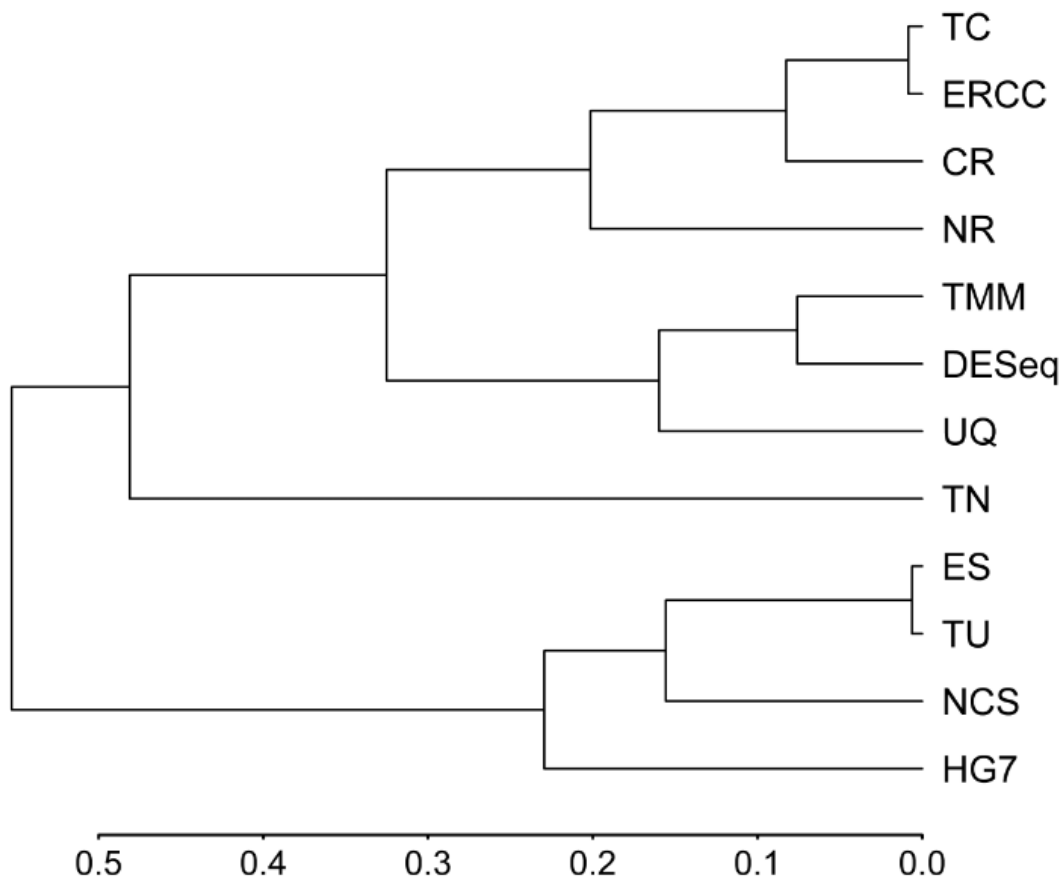
结果展示(见[1]图 3D):



### 3.E 标准化因子层次聚类图（单细胞数据）

```
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();
```

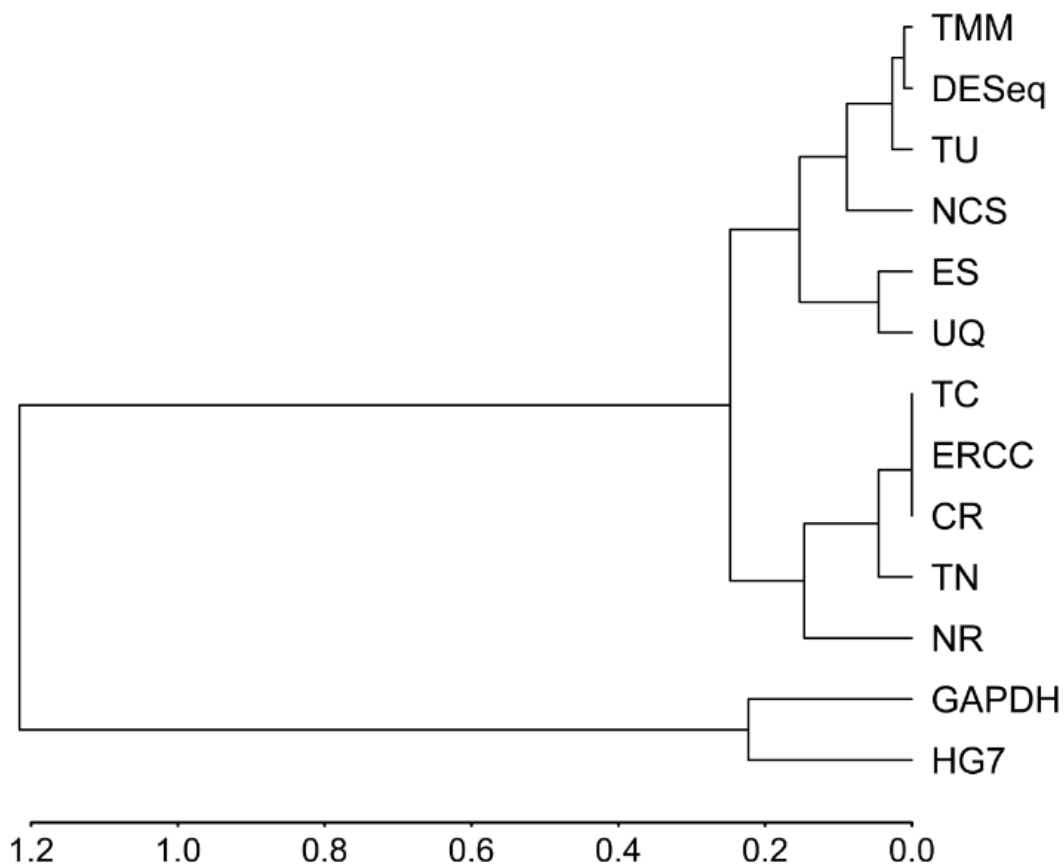
结果展示(见[1]中图 3E):



### 3.F 标准化因子层次聚类图 (Bulk 数据)

```
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();
```

结果展示(见[1]中图 3F):



## 4 如何评估多个用户的方法

### 4.A 有标准化因子的情况（单细胞数据）

```
#对于用户自己的方法：method1-9
#如果能够计算出标准化因子（向量）：factor1-9，参见 2.A
#TU 需要优化计算，消耗很长的计算时间
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));

#查看 TU 产生的 AUCVC 最大值
scRNA663.AUCVCs1;

#从 TU 优化计算的输出文件 bestPara.txt 中找到 AUCVC 最大值对应的参数
bestPara <- read.table(file='bestPara.txt', header=TRUE);
bestPara;

#根据 AUCVC 最大值对应的参数，计算得到 TU 的标准化因子
tu <- getFactors(data = scRNA663, method = "TU", pre_ratio=0.5, lower_trim=0.05,
upper_trim=0.65);
```

```
#计算 11 种方法的 mSCC(Nonzero ratio=0.2)
#这里只计算 Nonzero ratio=0.2 的情况，其他情况要逐个计算但不是必须
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 更为普遍的情况（单细胞数据）

```
#对于用户自己的方法：method1-9
#可以先求对应的标准化的基因表达矩阵：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));

#TU 需要优化计算，消耗很长的计算时间
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));

#查看 TU 产生的 AUCVC 最大值
scRNA663.AUCVCs2;

#从 TU 优化计算的输出文件 bestPara.txt 中找到 AUCVC 最大值对应的参数
bestPara <- read.table(file='bestPara.txt', header=TRUE);
bestPara;

#根据 AUCVC 最大值对应的参数，计算得到 TU 标准化因子
tu <- getFactors(data = scRNA663, method = "TU", pre_ratio=0.5, lower_trim=0.05,
upper_trim=0.65);

#计算得到应用 TU 方法标准化后的基因表达矩阵
TU_sc.matrix <- getNormMatrix(data = scRNA663, norm.factors = tu);

#计算 11 种方法的 mSCC(Nonzero ratio=0.2)
#这里只计算 Nonzero ratio=0.2 的情况，其他情况要逐个计算但不是必须
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 R 包版本和常见问题

```
> sessionInfo()
R version 3.4.2 (2017-08-28)
Platform: x64_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] varidisuite_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    trinccluster_0.1-2 varidis_0.4.1     robustbase_0.82-8
 [20] 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    mvtnorm_1.0-6
```