VLADIMIR BARBOSA CARLOS DE SOUZA

ANÁLISE DE DADOS DE RNA-SEQ COM DIFERENTES NÚMEROS DE FATORES E REPETIÇÕES

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Estatística Aplicada e Biometria, para obtenção do título de *Magister Scientiae*.

VIÇOSA
MINAS GERAIS – BRASIL
2015

Ficha catalográfica preparada pela Biblioteca Central da Universidade Federal de Viçosa - Câmpus Viçosa

T
S729a
2015

Souza, Vladimir Barbosa Carlos de, 1982-
Análise de dados de RNA-Seq com diferrentes números de fatores e repetições / Vladimir Barbosa Carlos de Souza. – Viçosa, MG, 2015.
xi, 74f. : il. (algumas color.) ; 29 cm.

Orientador: Luiz Alexandre Peternelli.
Dissertação (mestrado) - Universidade Federal de Viçosa.
Referências bibliográficas: f.71-74.

1. Biologia molecular. 2. Expressão gênica. 3. Bioinformática. 4. Análise de dados (Estatísticas). 5. Biometria. I. Universidade Federal de Viçosa. Departamento de Estatística. Programa de Pós-graduação em Estatística Aplicada e Biometria. II. Título.

CDD 22. ed. 572.8633

VLADIMIR BARBOSA CARLOS DE SOUZA

ANÁLISE DE DADOS DE RNA-SEQ COM DIFERENTES NÚMEROS DE FATORES E REPETIÇÕES

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Estatística Aplicada e Biometria, para obtenção do título de *Magister Scientiae*.

APROVADA: 22 de julho de 2015.

_____
Moyses Nascimento
(Coorientador)

_____
Talles Eduardo Ferreira Maciel

_____
Luiz Alexandre Peternelli
(Orientador)

“Existem muitas hipóteses em ciência que estão erradas. Isso é perfeitamente aceitável, elas são a abertura para achar as que estão certas”. — Carl Sagan

Agradecimentos

Aos meus pais Sebastião Carlos de Souza e Edirlene Barbosa Carlos de Souza, por acreditarem em mim e por todo apoio, carinho e amor.

À minha noiva Bethânia Gomes de Souza, por estar sempre ao meu lado, me incentivando e apoiando, e por todo seu carinho, amizade e amor.

Aos meus irmãos Vinícius Barbosa Carlos de Souza e Vitor Barbosa Carlos de Souza, pela amizade e incentivo.

Especialmente ao meu orientador Luiz Alexandre Peternelli, pela paciência, dedicação e amizade.

Ao meu coorientador Moyses Nascimento, pela dedicação e amizade.

À ajuda dos amigos Talles Eduardo Ferreira Maciel e Eveline Teixeira Caixeta.

À Carla Zinato Campos, por sempre estar disposta a ajudar.

A todos meus amigos de república e frequentadores, pelos bons momentos de convívio e amizade.

A todos meus amigos da Biologia, UFV, pelos bons momentos e experiências que passamos juntos e pela amizade.

A todos meus amigos do Departamento de Estatística da UFV, pela contribuição e amizade.

À FAPEMIG, pela concessão da bolsa de estudo.

Sumário

Lista de Figuras

Lista de Tabelas

RESUMO

SOUZA, Vladimir Barbosa Carlos de, M.Sc., Universidade Federal de Viçosa, Julho de 2015. **Análise de dados de RNA-Seq com diferentes números de fatores e repetições**. Orientador: Luiz Alexandre Peternelli. Coorientador: Moyses Nascimento.

A tecnologia RNA-Seq mostrou-se ser revolucionária para o estudo de expressão gênica. Porém, mais estudos na literatura sobre a análise de dados de RNA-Seq são necessários, até mesmo porque se trata de um método de elevado custo. Devido a este alto custo, é importante o aproveitamento das amostras disponíveis para concluir sobre mais fatores e suas interações. Este trabalho tem como objetivo realizar um comparativo do desempenho da análise de identificação de DEGs (genes diferencialmente expressos) em experimentos com diferentes números de fatores e repetições, mas todos com o mesmo número de amostras, ou seja, com o mesmo custo. Para as análises, foram simulados conjuntos de dados provenientes de experimentos com diferentes números de fatores e repetições. Para a realização dessas simulações foi utilizado o pacote `TCC`, desenvolvido para o *software* livre R, para a normalização dos dados também foi utilizado o `TCC`, e para a identificação dos DEGs foi utilizado o pacote `DESeq`, também desenvolvido para o R. Por último, o desempenho das análises de cada experimento foi calculado utilizando-se curvas ROC (*Receiver Operating Characteristics*), usando-se o pacote `ROCR`, também disponível para o R. Após o cumprimento da metodologia, pôde-se observar que, na ausência de interação entre fatores, não ocorre perda de desempenho das análises ao adicionar mais fatores, e, quando existe interação entre fatores, ocorre essa perda. Portanto, o uso de mais fatores, ao custo de se ter menos repetições, pode ser vantajoso.

ABSTRACT

SOUZA, Vladimir Barbosa Carlos de, M.Sc., Universidade Federal de Viçosa, July, 2015. **Analysis of RNA-Seq data with different numbers of factors and replicates**. Adviser: Luiz Alexandre Peternelli. Co-adviser: Moyses Nascimento.

The RNA-Seq technology show to be revolutionary for gene expression studies. However, more studies in literature about the analysis of RNA-Seq data are necessary, even because it is a costly method. Because of that high cost, it is important to take the full advantage of the available samples to conclude about more factors and its interactions. The aim of this work is to perform a comparative of the performance of DEGs (differential expression genes) identification analysis in experiments with different numbers of factors and replicates, but all of them with the same number of samples, or, in other words, with the same cost. For the analysis, was simulated a dataset from experiments with different numbers of factors and replicates. The package TCC, developed to the free software R, was used to perform that simulation. For the normalization of the data, TCC was also used, and for the DEGs identification the package DESeq was used, also developed to R. Finally, the performance of the analysis of each experiment was calculated with the use of ROC (Receiver Operating Characteristics) Curves, using the package ROCR, also available for R. After the implementation of the methodology, it was possible to observe that, when absence of interactions between factors, do not occur loss of analysis's performance when more factors are added, and, when there are interactions of factors, that loss happens. Therefore, the use of more factors, to the cost of having less replicates, may be advantageous.

1 Introdução

A análise de expressão gênica é de fundamental importância em vários campos de pesquisas biológicas. Um maior conhecimento dos padrões de expressão gênica tem possibilitado um melhor entendimento básico de mecanismos biológicos, auxiliando desde em pesquisas relacionadas a tratamentos de doenças quanto a programas de melhoramento genético.

No final da década de 1990 e começo da década de 2000, a principal tecnologia utilizada para estudo de expressão gênica foi a chamada *microarray*. Tecnologia esta que possibilitou um grande avanço na área, tornando possível quantificar o nível de expressão de milhares de genes simultaneamente.

Porém, no final da década de 2000, a nova tecnologia RNA-Seq, baseada em tecnologias de Sequenciamento de Nova Geração (NGS), emergiu como revolucionária na criação de dados de expressão gênica. Esta tecnologia tem se mostrado vantajosa, em relação a *microarray*, em vários aspectos como: i) não está limitada a detectar apenas aqueles transcritos que correspondam ao material genético de referência; (ii) consegue identificar genes com baixos ou elevados níveis de expressão; (iii) apresenta maior acurácia na quantificação do nível de expressão; (iv) são necessários um número menor de amostras de RNA para o uso da técnica; (v) promove mais informações para identificar expressão alelo-específica; (vi) revela novos promotores e isoformas; (vii) oferece menos ruído e maior rendimento (Wang L *et al*, 2010; Wang Z *et al*, 2009; Oshlack *et al*, 2010). Por causa desses motivos, RNA-Seq passou a ser a principal forma de estudo da expressão gênica.

Apesar das tecnologias de NGS possibilitarem um menor custo em relação a outras tecnologias de sequenciamento (Hayden, 2009), o custo para sequenciamentos via RNA-Seq ainda permanece elevado, principalmente quando não se tem o sequenciamento do genoma da espécie em estudo para servir como genoma de referência. Devido à limitação de recursos, é muito comum o uso de poucas amostras em pesquisas sobre expressão gênica, realizando poucas repetições ou até mesmo nenhuma. Portanto, é extremamente importante um bom planejamento experimental, para que seja extraído o máximo de informações dos dados, obtidos por poucas amostras disponíveis. Baseando-se nesse fato como motivação, e também pela falta de material na literatura sobre experimentos fatoriais de RNA-Seq, este trabalho objetiva

comparar o desempenho da análise de experimentos com diferentes números de fatores e repetições, porém todos eles com o mesmo número de amostras, ou seja, com o mesmo custo, para que assim possa ser sugerido experimentos fatoriais em vez de experimentos com apenas um fator. Assim será possível, caso seja de interesse do pesquisador, concluir sobre mais fatores e suas interações, conseguindo maior informação sem aumentar o custo do experimento.

Suponha-se que um pesquisador queira testar o efeito do tratamento *A* nos níveis de expressão gênica. Sendo assim, ele prepara todas as suas amostras sob uma mesma condição fixa, exceto variando os níveis de *A*. Essas condições fixas que não entram na análise do experimento são constituídas por vários fatores (sejam eles os fatores *B*, *C*, *D*, …, *Z*), por exemplo: temperatura, pH, pressão, taxa de fornecimento de um substrato em uma cultura de células, entre outros. E, sendo assim, o pesquisador conclui sobre o tratamento *A*. Porém, ele apenas pode concluir quando aqueles níveis utilizados nos fatores *B*, *C*, *D*, …, *Z*. Pois, é possível que, quando o experimento for conduzido em outros níveis de *B*, *C*, *D*, …, *Z*, a conclusão sobre *A* ser completamente diferente. Para minimizar esse problema, o pesquisador poderia fazer uso de um experimento fatorial, utilizando como fatores o fator *A* (que é seu alvo principal na análise) e outros fatores que julgar serem importantes ou, principalmente, que interagem significativamente com *A*. Porém, para que essa análise fatorial tenha o mesmo custo da análise que utiliza apenas *A*, é necessário diminuir o número de repetições na medida em que mais fatores são adicionados. O número de fatores a ser adicionado está limitado pelo número de amostras disponíveis. Com isso, o pesquisador poderá ter uma melhor conclusão sobre *A*, além de também poder concluir sobre outros fatores.

Nesse trabalho, deseja-se verificar o que acontece com o desempenho da análise de expressão gênica (análise de identificação de genes diferencialmente expressos (DEGs)) em função do fator *A*, na medida em que outros fatores independentes de *A*, ou seja, os efeitos de suas interações são não significativos, são adicionados na análise, e também no caso quando a interação entre os fatores for significativa. Ao adicionar mais fatores, o número de repetições deverá diminuir, de modo a manter o mesmo números de amostras entre os experimentos. Pode-se, assim, comparar o desempenho de experimentos com o mesmo custo. Assim, será possível concluir se é vantajoso usar um planejamento como experimento fatorial ao invés do experimento de um tratamento. Caso exista interação entre os fatores, pode-se considerar que já foi obtido sucesso no planejamento do experimento ao optar por um experimento fatorial, pois, sendo os

efeitos dos fatores dependentes, eles não podem ser testados separadamente.

2 Objetivos e justificativa

2.1 Objetivo geral

- Realizar um comparativo do desempenho da análise de identificação de DEGs entre experimentos com diferentes números de fatores e repetições, mas todos eles com o mesmo número de amostras, ou seja, mesmo custo, quando as interações entre os fatores forem significativas, e quando não forem significativas.

2.2 Objetivos específicos

- Fazer um levantamento bibliográfico sobre metodologias atuais empregadas na análise de expressão gênica baseada em dados de RNA-Seq, tais como o método de normalização dos dados e de identificação de DEGs;
- Apresentar as linhas de comando que devem ser utilizadas no *software* R para a realização das análises de expressão gênica em dados de RNA-Seq com a utilização dos pacotes apropriados.

2.3 Justificativa

- Considerando a dificuldade na obtenção de número satisfatório de amostras em experimentos de RNA-Seq, devido principalmente ao elevado custo, é importante a utilização de um desenho experimental que possibilite extrair o máximo de informações das poucas amostras disponíveis. Ou seja, poder concluir sobre o fator em estudo sob um maior número de condições, além de concluir sobre mais fatores e suas interações, ao inserir mais fatores nas análises,

porém, sem aumentar o custo do experimento.

- Existe uma atual falta de material na literatura sobre experimentos fatoriais de RNA-Seq.

3 Referencial Teórico

3.1 Expressão Gênica

Dentro das células, o DNA é a molécula responsável pelo armazenamento da informação genética. Nele estão os genes que conferem as características de um indivíduo. Os genes são regiões do DNA que entram num processo chamado transcrição, que consiste na produção de moléculas de RNAs através do DNA. Essas moléculas de RNAs, também chamada de transcritos, podem apresentar diferentes funções, uma delas é servir de molde para um outro processo, denominado tradução. A molécula de RNA que serve de molde para a tradução é chamada de RNA mensageiro (mRNA), e a tradução é a produção de proteínas através do código presente nas moléculas de mRNAs. A expressão de um gene é justamente a formação de transcritos a partir de um determinado gene, podendo seu nível ser influenciado por algum fator, como por exemplo, o ambiente, que pode alterar o metabolismo de um indivíduo, e esse influenciar, inibindo ou estimulando, o nível de expressão de alguns genes relacionados a tal mudança (Zaha *et al*, 2003).

3.2 Como são obtidos os dados de RNA-Seq

Entre as tecnologias de RNA-Seq voltadas para o estudo de expressão gênica, as principais plataformas de NGS que são amplamente aceitas comercialmente são: Illumina's Genome Analyzer, e Applied Biosystems' SOLiD (Auer & Doerge, 2010). A tecnologia Illumina (apenas esta será apresentada) consiste em, primeiramente, isolar as moléculas de mRNA do material biológico de estudo. Em seguida, esses mRNAs são fragmentados em milhões de pedaços. Esses fragmentos de mRNAs são então utilizados para a formação de cDNAs, a partir da reação de transcrição reversa. Logo após,

fragmentos de cDNA de tamanho entre 35 a 300 pares de bases (pb) são selecionados e colocados para serem ampliados por PCR (conjunto de reações químicas que multiplica a quantidade de material genético disponível em uma solução). Esses cDNAs amplificados são colocados para serem sequenciados em uma máquina específica. Após o sequenciamento, os fragmentos de cDNA são alinhados a um genoma (ou transcriptoma) de referência, caso exista. E assim, finalmente pode-se calcular o nível de expressão de cada gene. Para isso, deve-se contar quantos fragmentos de cDNA alinharam-se com o gene em questão (Nunes, 2014). A Figura 1 exemplifica como é feito o cálculo da abundância (expressão) de um gene qualquer. Para obter informações sobre como planejar apropriadamente experimentos de RNA-Seq, consulte Auer & Doerge (2010).

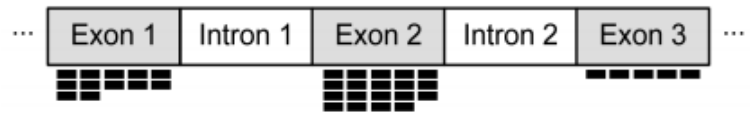

Figura 1: Exemplo de contagens alinhadas de dados de RNA-Seq. A expressão gênica é definida como o número de fragmentos lidos (pequenos retângulos negros) que são mapeados aos *exons* (grandes retângulos cinzas) em cada gene. Neste exemplo, a expressão gênica digital é 12 + 19 + 5 = 36. Fonte: Nunes (2014).

Portanto, dados de RNA-Seq são dados de contagem. Sendo assim, é fácil perceber que dados de RNA-Seq não possuem a mesma natureza dos dados de *microarray*, já que estes são contínuos. Devido a isso, as técnicas desenvolvidas para um, não devem ser usadas para a análise do outro. Porém, existe uma metodologia chamada *voom* que, com ela, os métodos desenvolvidos para análise de dados de *microarray* podem ser utilizados para realizar a análise de dados de RNA-Seq. Para mais detalhes sobre essa metodologia, consulte Law *et al* (2014).

3.3 Simulação de dados de RNA-Seq usando o pacote TCC

Além de realizar outras funções, o pacote TCC (Sun *et al*, 2013) também simula dados de RNA-Seq. Essa simulação gera dados de uma distribuição binomial negativa, cujos parâmetros utilizados são os estimados da distribuição empírica dos dados de RNA-Seq de *Arabidopsis*, disponível no pacote NBPSeq (Di *et al*, 2011). Esse conjunto de dados é constituído por três repetições biológicas para cada um dos tratamentos tratado e não tratado (Sun *et al*, 2015).

Abaixo são apresentados, como exemplo, os comandos no R para realizar a simulação dos dados de um experimento com as seguintes características: 3 tratamentos (*multi-group data*); 1000 genes (`Ngene=1000`); os 40% primeiros genes são DEGs (`PDEG=0.4`); nos 30% primeiros genes que são DEGs, a expressão é 4 vezes maior (do que um certo valor basal) no primeiro tratamento; nos 50% próximos genes que são DEGs, a expressão é 6 vezes maior no segundo tratamento; nos 20% últimos genes que são DEGs, a expressão é 2 vezes menor no terceiro tratamento (`DEG.assign=c(0.3,0.5,0.2)` e `DEG.foldchange=c(4,6,0.5)`); duas repetições para o primeiro e o segundo tratamento, e 3 repetições para o terceiro tratamento (`replicates=c(2, 2, 3)`).

```
> tcc<-simulateReadCounts(Ngene = 1000,
+                         PDEG = 0.4,
+                         DEG.assign = c(0.3, 0.5, 0.2),
+                         DEG.foldchange = c(4, 6, 0.5),
+                         replicates = c(2, 2, 3))
> head(tcc$count)          #Seis linhas de dados gerados pela simulacao.
       G1_rep1 G1_rep2 G2_rep1 G2_rep2 G3_rep1 G3_rep2 G3_rep3
gene_1       0      36       0       1       2       0       1
gene_2     304     375      79      71      61     134      87
gene_3     358     324      35     132      66      91      20
gene_4      27      23       0      15       2       4       3
gene_5      70      85      29      29      23      32      19
gene_6     257     185      53      57      52      54      51
```

Também é mostrado como é feita a simulação de dados de um experimento fatorial. Como exemplo, será mostrado a simulação de dados de um experimento com: 2 fatores; 2 repetições; 1000 genes (`Ngene=1000`); os 50% primeiros genes são DEGs (`PDEG=0.5`); nos 45% primeiros genes que são DEGs a interação entre os fatores é não significativa e nos 55% últimos genes que são DEGs a interação entre os fatores é significativa (de

acordo com `DEG.assign=c(0.45,0.55)` e `DEG.foldchange=DEG.foldchange`, explicação no próximo parágrafo).

```
> group<-data.frame(A=rep(paste('a',1:2,sep=''),c(4,4)),
+                   B=rep(rep(paste('b',1:2,sep=''),c(2,2)),2))
> group         #Combinacoes dos niveis dos fatores.
   A  B
1 a1 b1
2 a1 b1
3 a1 b2
4 a1 b2
5 a2 b1
6 a2 b1
7 a2 b2
8 a2 b2
> DEG.foldchange<-data.frame(fc1=c(1,1,4,4,4,4,7,7),  #Sem interacao dos fatores
+                            fc2=c(1,1,4,4,4,4,1,1))  #Com interacao dos fatores
> DEG.foldchange
  fc1 fc2
1   1   1
2   1   1
3   4   4
4   4   4
5   4   4
6   4   4
7   7   1
8   7   1
> tcc<-simulateReadCounts(Ngene = 1000,
+                         PDEG = 0.5,
+                         DEG.assign = c(0.45, 0.55),
+                         DEG.foldchange = DEG.foldchange,
+                         group = group)
> head(tcc$count)
      a1b1_rep1 a1b1_rep2 a1b2_rep1 a1b2_rep2 a2b1_rep1 a2b1_rep2 a2b2_rep1 a2b2_rep2
gene_1        63        55       197       251       208       241       352       395
gene_2        26        50        86        23       331       128        63       685
gene_3       375       487       900       701       841       978      1657      2140
gene_4       236       283      1057      1116      1139      1090      1916      1799
gene_5         3        20        10         6         7        20         1         0
gene_6        42        53       163       107       203       117       315       307
```

Como pode-se ver, na simulação anterior (experimento fatorial), existem dois fatores, cada um deles com dois níveis e cada combinação destes níveis com duas repetições (todas essas informações foram fornecidas à função `simulateReadCounts()` pelo argumento `group=group`). Quando um experimento fatorial é simulado com o `TCC`, o argumento `DEG.foldchange` sempre será um *data frame* (diferentemente de quando for um experimento com um único fator (*two-group* ou *multi-group*), em que o argumento `DEG.foldchange` é um vetor). O primeiro elemento do vetor atribuído ao argumento

`DEG.assign`, está relacionado à primeira coluna do *data frame* atribuído ao argumento `DEG.foldchange`, o segundo elemento à segunda coluna, e assim por diante. Portanto o número de elementos de `DEG.assign` sempre será igual ao número de colunas de `DEG.foldchange`. Além disso, o número de linhas de `DEG.foldchange` deve ser igual ao número de linhas de `group`. No exemplo, quer dizer que a expressão dos 45% primeiros genes que são DEGs estão de acordo com os valores de *fold-change* dados na primeira coluna de `DEG.foldchange`, sendo que cada um desses oito valores de *fold-change*, estão relacionados a uma combinação de níveis dos fatores, ou seja, a cada uma das 8 linhas de `group`. Já os últimos 55% dos genes que são DEGs, suas expressões estão relacionadas aos valores de *fold-change* da segunda coluna de `DEG.foldchange`.

O pacote `TCC` também realiza a identificação de DEGs, entre outras funções. Para outras informações sobre como usar o pacote `TCC` e seus comandos, consulte o manual em Sun *et al* (2015).

3.4 O método TCC para normalização de dados de RNA-Seq

Estudos têm mostrado que a normalização dos dados de RNA-Seq é um passo crítico para a obtenção de uma análise mais acurada de expressão gênica diferencial (Kadota *et al*, 2012). Essa prática consiste em remover efeitos técnicos sistemáticos que ocorrem nos dados, garantindo que o viés atribuído a estes tenha mínimo impacto nos resultados (Robinson & Oshlack, 2010). Sun *et al* (2013) vai além, afirmando que o método de normalização causa um maior impacto no ranqueamento de DEGs do que a própria metodologia de identificação de DEGs. Esses indicam a grande importância da normalização dos dados em experimentos de RNA-Seq.

O pacote `TCC`, desenvolvido para o R, propõe uma nova estratégia para normalização, baseada em outros pacotes já existentes, chamada de *DEG Elimination Strategy* (DEGES). Em Sun *et al* (2013) é mostrado ser vantajoso o uso desse método proposto pelo pacote. Essa vantagem é devida ao fato de, quando existir um desbalanceamento entre o número de DEGs que são *up-regulated* (genes que apresentam uma maior expressão) e *down-regulated* (genes que apresentam uma menor expressão) em um certo grupo, a análise de identificação de DEGs apresenta um melhor desempenho quando o método de normalização utilizado for o proposto em `TCC`. Portanto, a DEGES é um método robusto que melhor consegue lidar com dados que

possuem este tipo de desbalanceamento, diferentemente dos outros métodos que tem como pressuposição um balanceamento entre DEGs *up-regulated* e *down-regulated*, algo que muitas vezes não acontece na prática.

O método DEGES de normalização, proposto pelo `TCC`, consiste em três passos: (i) primeiramente, realiza-se a normalização dos dados (todos os genes) através de algum método de normalização a escolher; (ii) após, são identificados os genes que são DEGs, utilizando alguma metodologia de identificação de DEGs a escolher, e esses genes são eliminados; (iii) e por último, uma nova normalização é feita, apenas naqueles genes que não foram eliminados na etapa anterior. As etapas (ii) e (iii) podem entrar em um *loop* e serem repetidas até que haja convergência, ou seja, não ocorrer mais a eliminação de algum gene na etapa (ii), ou ainda, começará a ocorrer a normalização dos mesmos genes na etapa (iii) em cada *loop*. Essa estratégia que faz uso de iteração recebe o nome de *Iterative DEG Elimination Strategy* (iDEGES).

Como o `TCC` utiliza funções já existentes por outros pacotes, o usuário deve escolher qual função será utilizada para realizar a normalização na etapa (i) e (iii). Para isso, pode-se escolher a opção TMM, que é o método de normalização utilizado no pacote `edgeR` (Robinson *et al*, 2010), ou pode-se escolher o método de normalização utilizado pelo pacote `DESeq` (Anders & Huber, 2010), também chamado de DESeq. Além disso, o usuário também deve escolher qual o método de identificação de DEGs será utilizado na etapa (ii). Para isso, pode-se escolher os métodos edgeR, DESeq ou baySeq (Hardcastle & Kelly, 2010), cada um pertencente ao pacote referente ao seu respectivo nome. E por último, deve-se definir se o processo será iterativo e quantos *loops* serão realizados. Portanto, o `TCC` consiste em realizar combinações de funções já existentes pelos pacotes `edgeR`, `DESeq` e `baySeq`, alcançando um melhor resultado daqueles obtidos utilizando esses pacotes individualmente (Sun *et al*, 2013). Todas essas informações sobre como é realizada a normalização usando o método TCC, estão contidas na Figura 2 para um melhor entendimento.

Seguem abaixo os comandos no R para realizar a normalização:

```
> library(TCC)      #Carregar o pacote TCC. O pacote já deve estar instalado.
                    #  Veja como instalar corretamente o pacote TCC em
                    #  (Sun et al, 2015).
> data(hypoData)    #Cria o data frame hypoData com os dados hipoteticos.
```

```
> head(hypoData)
        G1_rep1 G1_rep2 G1_rep3 G2_rep1 G2_rep2 G2_rep3
gene_1      34      45     122      16      14      29
gene_2     358     388      22      36      25      68
gene_3    1144     919     990     374     480     239
gene_4       0       0      44      18       0       0
gene_5      98      48      17       1       8       5
gene_6     296     282     216      86      62      69

#hypoData eh um data frame dos dados de um experimento hipotetico com dois
#  grupos (tratamentos) e 3 repeticoes.
> group<-c(1,1,1,2,2,2)   #Vetor dos tratamentos.
> tcc<-new('TCC',         #Criar objeto do tipo tcc.
+          hypoData,
+          group)
> tcc<-calcNormFactors(tcc,
+                      norm.method = 'tmm',      #Ou 'deseq'.
+                      test.method = 'edger',    #Ou 'deseq' ou 'bayseq'.
+                      iteration = 3)            #Definir o numero de iteracoes.
> tcc$norm.factors              #Fatores de normalizacao calculados.
  G1_rep1   G1_rep2   G1_rep3   G2_rep1   G2_rep2   G2_rep3
0.8766053 0.8450605 0.8346595 1.0842097 1.1538160 1.2056491

> head(getNormalizedData(tcc))     #Dados normalizados.
          G1_rep1   G1_rep2   G1_rep3    G2_rep1    G2_rep2    G2_rep3
gene_1   33.51453  45.02768 119.80558  16.172931  14.318555  28.978172
gene_2  352.88829 388.23870  21.60429  36.389094  25.568847  67.948817
gene_3 1127.66536 919.56539 972.19284 378.042253 490.921870 238.820107
gene_4    0.00000   0.00000  43.20857  18.194547   0.000000   0.000000
gene_5   96.60070  48.02953  16.69422   1.010808   8.182031   4.996237
gene_6  291.77355 282.17349 212.11480  86.929502  63.410742  68.948064
```

3.5 DESeq

O pacote `DESeq` realiza a análise de identificação de genes com expressão diferencial em duas maneiras diferentes: (i) quando o experimento tiver apenas um fator, é realizado um teste baseado no teste exato de Fisher, utilizando a função `nbinomTest()`, esse teste é demostrado em Anders & Huber (2013); (ii) quando o experimento for fatorial, é utilizado modelos lineares generalizados e comparado os modelos completo e reduzido por meio do teste da razão de verossimilhanças, utilizando a função `nbinomGLMTest()`. A metodologia citada em (ii) também pode ser utilizada quando apenas um fator (apenas esta metodologia será descrita neste trabalho).

Considere a matriz dos dados de contagem dos *reads* (dados de RNA-Seq), de um experimento com dois tratamentos $(\tau_1$ e $\tau_2)$, dada por

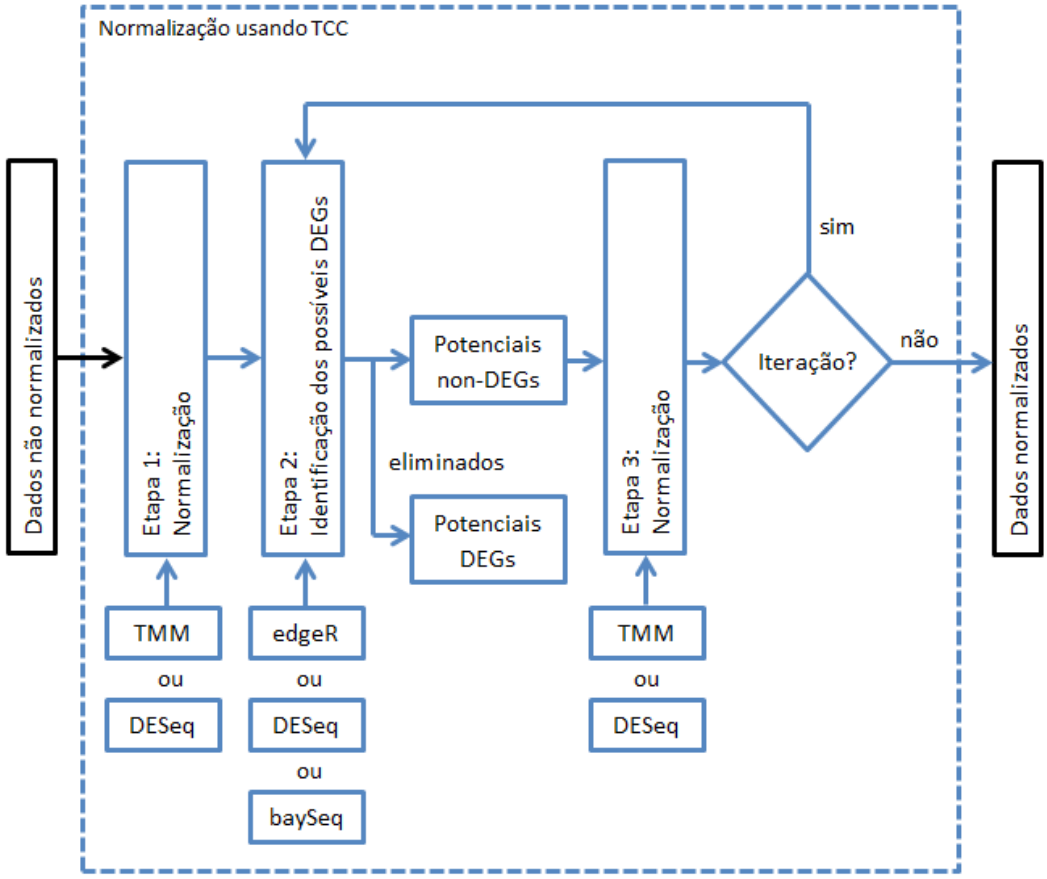

Figura 2: Esquema representando o algoritmo utilizado para normalização de dados usando o método TCC. Figura baseada em Sun et al (2013).

$$\begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1m_0} & y_{1(m_0+1)} & \cdots & y_{1m} \\ y_{21} & y_{22} & \cdots & y_{2m_0} & y_{2(m_0+1)} & \cdots & y_{2m} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ y_{n1} & y_{n2} & \cdots & y_{nm_0} & y_{n(m_0+1)} & \cdots & y_{nm} \end{bmatrix},$$

em que,

y_{ij} , com $i=1,2,\ldots,n$ e $j=1,2,\ldots,m_0,m_0+1,\ldots,m$, é o valor de contagem de *reads* da *i*-ésima linha e *j*-ézima coluna;

cada linha de 1 a n representam um gene;

cada coluna de 1 a m representa uma amostra;

as colunas de 1 a m_0 são repetições do tratamento τ_1 ; e

as colunas de m_0+1 a m são repetições do tratamento τ_2 .

Além disso, seja definido o modelo completo dado pelo modelo linear generalizado

$$y_{ij} = \mu_i + \tau_{i,c(j)} + e_{ij} \quad ,$$

em que,

y_{ij} é o valor observado da variável aleatória Y_{ij} que segue distribuição binomial negativa e que representa a contagem de *reads* da *j*-ésima amostra do *i*-ésimo gene;

μ_i é o parâmetro média de Y_{ij} , para o *i*-ésimo gene;

$c(j)$ é uma função que retorna $c(j)=1$ quando $j=1,2,\ldots,m_0$, e $c(j)=2$ quando $j=m_0+1, \ldots, m$;

τ_{ic} é o efeito do tratamento τ_ρ $(\rho=1,2)$, *no i*-ésimo gene; e

e_{ij} é o valor observado da variável aleatória erro experimental.

E seja definido o modelo reduzido dado pelo modelo linear generalizado

$$y_{ij} = \mu_i + e_{ij} \quad .$$

Considere a situação em que se deseja testar se existe diferença entre os tratamentos, para cada gene *i*, ou seja, testar as hipóteses

H_0: $\tau_{i1} = \tau_{i2}$, e

H_a: $\tau_{i1} \neq \tau_{i2}$.

Para isso, basta testar se existe diferença significativa do quanto que o modelo completo explica de Y_{ij} , e do quanto é explicado pelo modelo reduzido.

Para comparar estes dois modelos, usa-se o teste da razão de verossimilhanças. A estatística desse teste é dada por

$$\Lambda(\underset{\sim}{Y}) = \frac{\underset{\theta\in\omega}{sup} \; L_0(\theta|\underset{\sim}{Y})}{\underset{\theta\in\Omega}{sup} \; L_a(\theta|\underset{\sim}{Y})} \quad , \qquad (1)$$

em que,

$\underset{\sim}{Y}$ é o vetor aleatório dos valores de contagem dos *reads*;

L_0 e L_a são as funções de verossimilhança sob a hipótese nula e alternativa, respectivamente;

θ é o vetor dos parâmetros;

ω é o conjunto de todos os valores possível de θ, baseando-se na hipótese nula; e

Ω é o conjunto complementar de ω, ou seja, o conjunto de todos os valores possível de θ, baseando-se na hipótese alternativa;

Porém, ao invés de usar o valor de θ que maximiza cada uma das duas funções de verossimilhança em (1), θ é calculado segundo a metodologia de estimativa proposta em `DESeq`. Essa metodologia é descrita a seguir baseando-se em Anders & Huber (2013).

Em Anders & Huber (2013) assume-se que cada valor y_{ij} provém de uma distribuição binomial negativa com parâmetros média e variância próprios, ou seja,

$$Y_{ij} \sim \mathrm{BN}\left(\mu_{ij}, \sigma_{ij}^2\right) \quad , \qquad (2)$$

e que, devido ao fato de dados de RNA-Seq tipicamente possuírem poucas repetições, para que estimativas úteis possam ser obtidas, deve-se fazer algumas suposições sobre a modelagem. São elas:

1) $\mu_{ij} = q_{i,\rho(j)}\, s_j$,

em que,

μ_{ij} é a média ou valor esperado da contagem observada de *reads* para o gene *i* e amostra *j*;

$\rho(j)$ é a condição (tratamento) atribuída à amostra *j*;

$q_{i,\rho(j)}$ é proporcional ao valor esperado da verdadeira concentração de fragmentos do gene *i* sob a condição $\rho(j)$; e

s_j é o fator de tamanho (*size factor*), que representa a cobertura da biblioteca (amostra) *j*, e é proporcional ao valor esperado de todas as contagens da amostra *j*.

2) $\sigma_{ij}^2 = \mu_{ij} + \underbrace{s_j^2 \, v_{i,\rho(j)}}_{\textit{variância bruta}}$,

em que,

σ_{ij}^2 é a variância do valor de contagem do gene *i* e amostra *j*; e

$s_j^2 \, v_{i,\rho(j)}$ é a variância bruta (*raw variance*).

3) $v_{i,\rho(j)} = v_\rho\left(q_{i,\rho(j)}\right)$,

em que,

$v_{i,\rho(j)}$ é uma função suave (*smooth function*) de $q_{i,\rho(j)}$.

A terceira suposição deve-se ao fato de o número de repetições ser tipicamente muito pequeno para poder ter uma estimativa precisa da variância do gene *i*, baseando-se apenas nos dados que o corresponda. Ao invés disso, essa suposição permite usar dados de genes com níveis de expressão similares para estimar a variância.

Dada as suposições, deve-se fazer as estimativas dos parâmetros. O parâmetro s_j é estimado por

$$\hat{s}_j = \underset{i}{mediana}\left\{\frac{y_{ij}}{\left(\prod_{v=1}^{m} y_{iv}\right)^{\frac{1}{m}}}\right\} ,$$

e este é importante para a normalização dos dados, que é realizada dividindo-se cada valor y_{ij} , de uma coluna j , por $\hat{s}_j$.

A estimativa de $q_{i\rho}$ é dada por

$$\hat{q}_{i\rho} = \frac{1}{m_\rho - 1} \sum_{j:\rho(j)=\rho} \frac{y_{ij}}{\hat{s}_j} ,$$

em que,

m_ρ é o número de repetições da condição ρ ; e

o somatório corre por essas repetições.

Para estimar a v_ρ , deve-se calcular a variância amostral

$$w_{i\rho} = \frac{1}{m_\rho - 1} \sum_{j:\rho(j)=\rho} \left(\frac{y_{ij}}{\hat{s}_j} - \hat{q}_{i\rho} \right)^2$$

e definir

$$z_{i\rho} = \frac{\hat{q}_{i\rho}}{m_\rho} \sum_{j:\rho(j)=\rho} \frac{1}{\hat{s}_j} \quad .$$

Assim, $w_{i\rho} - z_{i\rho}$ poderia ser o estimador de v_ρ . Porém, quando o número de repetições é pequeno, o que é típico em dados de RNA-Seq, os valores de $w_{i\rho}$ são altamente variados, o que faz com que $w_{i\rho} - z_{i\rho}$ não seja um estimador útil. Portanto, o método utiliza um modelo linear generalizado, da família gama, para uma regressão local no gráfico $(q_{i\rho}, w_{i\rho})$, para obter a função suave $w_\rho(q)$. E finalmente, v_ρ é estimado por

$$\hat{v}_\rho(\hat{q}_{i\rho}) = w_\rho(\hat{q}_{i\rho}) - z_{i\rho} \quad .$$

Portanto, ao concluir toda essa metodologia, obtém-se as estimativas dos parâmetros μ_{ij} e σ_{ij}^2 da distribuição em (2) para cada valor y_{ij} .

Porém, para realizar o teste da razão de verossimilhanças, deve-se supor que, segundo o modelo reduzido,

$$Y_{i1}, Y_{i2}, \ldots, Y_{im} \overset{\text{i.i.d.}}{\sim} \text{BN}(\mu_{iR}, \sigma_{iR}^2) \quad , \tag{3}$$

ou seja, não existe diferença entre os tratamentos e, sendo assim, todas as amostras referente ao gene *i* provém de uma mesma distribuição. E que, segundo o modelo completo,

$$Y_{i1}, Y_{i2}, \ldots, Y_{im_0} \overset{\text{i.i.d.}}{\sim} \text{BN}(\mu_{iA}, \sigma_{iA}^2) \tag{4}$$

e

$$Y_{i(m_0+1)}, Y_{i(m_0+2)}, \ldots, Y_{im} \overset{\text{i.i.d.}}{\sim} \text{BN}\left(\mu_{iB}, \sigma^2_{iB}\right) \quad , \tag{5}$$

ou seja, existe diferença entre os tratamentos e, por isso, as amostras $(y_{i1}, y_{i2}, \ldots, y_{im_0})$, referente ao gene *i*, que receberam um determinado tratamento (τ_1) , não provém de uma mesma distribuição das amostras $(y_{i(m_0+1)}, y_{i(m_0+2)}, \ldots, y_{im})$ que receberam o outro tratamento (τ_2) .

Para estimar os parâmetros das distribuições (3), (4) e (5), primeiramente define-se

$$K_{iA} = \sum_{j:\rho(j)\in A} Y_{ij} \quad , \quad K_{iB} = \sum_{j:\rho(j)\in B} Y_{ij} \quad , \quad K_{iT} = \sum_{j:\rho(j)\in T} Y_{ij} \quad ,$$

em que,

A é a condição devida à atribuição do tratamento τ_1 ,

B é a condição devida à atribuição do tratamento τ_2 , e

$T = A \cup B$.

Anders & Huber (2013) comentam que a distribuição da variável aleatória $(K_{i\rho})$, obtida pela soma de variáveis aleatórias (Y_{ij}) com distribuição binomial negativa, pode ser aproximada por uma também distribuição binomial negativa com parâmetros estimados por

$$\hat{\mu}_{K_{i\rho}} = \sum_{j:\rho(j)\in\rho} \hat{s}_j \, \hat{q}_{io} \quad ,$$

em que, $\hat{q}_{io} = \sum_{j:\rho(j)\in T} \frac{y_{ij}}{\hat{s}_j}$,

e

$$\hat{\sigma}^2_{K_{i\rho}} = \sum_{j:\rho(j)\in\rho} \hat{s}_j \, \hat{q}_{io} + \hat{s}_j^2 \, \hat{v}_\rho(\hat{q}_{io}) \quad .$$

Como valores de Y_{ij} são independentes, as estimativas dos parâmetros das distribuições (3), (4) e (5) são dados por

$$\hat{\mu}_{iR} = \hat{\mu}_{K_{iT}}/m \quad , \quad \hat{\sigma}^2_{iR} = \hat{\sigma}^2_{K_{ip}}/m \quad ,$$

para o modelo reduzido, e

$$\hat{\mu}_{iA} = \hat{\mu}_{K_{iA}}/m \quad , \quad \hat{\sigma}^2_{iA} = \hat{\sigma}^2_{K_{iA}}/m \quad ,$$
$$\hat{\mu}_{iB} = \hat{\mu}_{K_{iB}}/m \quad , \quad \hat{\sigma}^2_{iB} = \hat{\sigma}^2_{K_{iB}}/m \quad ,$$

para o modelo completo.

Portanto, a razão das verossimilhanças entre os modelos reduzido e completo, $\Lambda(\underset{\sim}{Y})$, definida em (1), para uma determinada amostra $\underset{\sim}{y}$, é

$$\Lambda(\underset{\sim}{y}) = \frac{L_0(\hat{\mu}_{iR}, \hat{\sigma}_{iR} | \underset{\sim}{y})}{L_a(\hat{\mu}_{iA}, \hat{\sigma}_{iA}, \hat{\mu}_{iB}, \hat{\sigma}_{iB} | \underset{\sim}{y})} \quad .$$

Como $-2\ln(\Lambda(\underset{\sim}{Y})) \sim \chi^2_s$, onde *s* é o grau de liberdade da distribuição de χ^2 , dado pela diferença do número de parâmetros estimados entre os modelos completo e reduzido, pode-se calcular o *p-valor* obtido pelo teste.

3.6 Curvas ROC

Curvas ROC (*Receiver Operating Characterestic*) é uma técnica para visualizar, organizar e selecionar classificadores baseando-se no seu desempenho (Fawcett, 2006). Essa técnica tem sido muito utilizada em trabalhos recentes que objetivam comparar o desempenho de diferentes metodologias de identificação de DEGs, por exemplo: Kvam *et al* (2012), Soneson & Delorenzi (2013), dentre outros. Porém, Curvas ROC podem ser empregadas não apenas para avaliar o desempenho de análises de expressão gênica, mas também, para avaliar o desempenho de qualquer outro classificador binário.

Seja *X* o verdadeiro valor de classificação de uma instância, e que *X* assuma apenas

dois valores possíveis (classificação binária). Pode-se generalizar, chamando esses dois valores possíveis de *positivo* e *negativo* (no caso da análise de expressão gênica, os valores que *X* poderia assumir seriam: DEG, para *positivo*; e *non*-DEG (genes não diferencialmente expressos), para *negativo*). Além disso, seja *Y* o valor predito por um modelo, e que *Y* possa assumir um valor real que, quanto maior for, mais força o classificador tem para classificar uma determinada instância como *positivo*, portanto *Y* pode também ser chamado de *valor de positividade*. Também é possível o uso de Curvas ROC quando *Y* assume algum outro tipo de valor (*e.g.*, classe predita), porém neste trabalho será tratado apenas do caso em que *Y* assume valores reais contínuos.

Sendo assim, um bom classificador seria aquele que, ao fazer um "ranqueamento" de todas as instâncias, baseando-se no valor de *Y* atribuído a cada uma delas, consiga colocar no topo de uma lista, as instâncias que seu verdadeiro valor (valor de *X*) seja *positivo* e, no final dessa lista, as instâncias que seu verdadeiro valor seja *negativo* (Prati *et al*, 2008).

Para desenhar a Curva ROC, foram utilizados, como exemplo, os valores de *X* e *Y* da Tabela 1. Percebe-se que pode ser definido um valor arbitrário *k* como o valor limiar que determinará que aquelas instâncias que possuem um valor de $Y > k$ serão consideradas pertencentes à classe *positivo*, e aquelas instâncias que tiverem seu valor de $Y < k$, à classe *negativo*. Assim, pode-se transformar *Y* para valores categóricos (*positivo* ou *negativo*), ou seja, mesma natureza de *X*, e com isso, calcular a *taxa de falso positivo* (FPR) e a *taxa de verdadeiro positivo* (TPR), em que,

$$\text{FPR} = \frac{\text{nº de instâncias classificadas erroneamente como } positivo}{\text{nº total de instâncias com valor real igual a } positivo} \quad ,$$

$$\text{TPR} = \frac{\text{nº de instâncias classificadas corretamente como } positivo}{\text{nº total de instâncias com valor real igual a } positivo} \quad .$$

O Gráfico ROC é construído com os valores de FPR no eixo horizontal (abscissas) e valores de TPR no eixo vertical (ordenadas). Porém, não é de interesse utilizar apenas um valor fixo arbitrário para *k*, e assim obter apenas um único ponto no Gráfico ROC. Pois assim será representando o desempenho do classificador baseando-se apenas naquele valor para *k*. Ao invés disso, deve-se variar *k* para todos os valores reais desde $-\infty$ até ∞ e, com isso, "plotar" cada ponto (FPR, TPR) obtido a cada valor

atribuído a *k*, formando assim uma curva no Gráfico ROC, a Curva ROC, que é independente do valor de *k* (Fawcett, 2006; Prati *et al*, 2008; Prati *et al*, 2011). No entanto, pode-se perceber que, na verdade, *k* não precisa variar entre todos os infinitos valores reais, basta que *k* varie entre todos os valores de *Y*, pois outros valores de *k* gerariam pontos (FPR, TPR) já existentes no Gráfico ROC.

Foi utilizado o pacote `ROCR` (Sing *et al*, 2005), desenvolvido para o *software* R, para desenhar a Curva ROC referente aos dados hipotéticos da Tabela 1.

Tabela 1: Exemplo de valores reais (*X*) e valores de positividade (*Y*) preditos por um classificador para 20 instâncias.

Instância	*X*	*Y*	Instância	*X*	*Y*
1	*positivo*	0,9	11	*positivo*	0,4
2	*positivo*	0,8	12	*negativo*	0,39
3	*negativo*	0,7	13	*positivo*	0,38
4	*positivo*	0,6	14	*negativo*	0,37
5	*positivo*	0,55	15	*negativo*	0,36
6	*positivo*	0,54	16	*negativo*	0,35
7	*negativo*	0,53	17	*positivo*	0,34
8	*negativo*	0,52	18	*negativo*	0,33
9	*positivo*	0,51	19	*positivo*	0,3
10	*negativo*	0,505	20	*negativo*	0,1

Obs.: Valores retirados de Fawcett (2005).

Os comandos utilizados para desenhar a curva ROC estão demostrados abaixo e a curva pode ser visualizada na Figura 3.

```
> library(ROCR)
> y<-c(0.9, 0.8, 0.7, 0.6, 0.55, 0.54, 0.53, 0.52, 0.51, 0.505, 0.4, 0.39, 0.38,
+      0.37, 0.36, 0.35, 0.34, 0.33, 0.3, 0.1)
> x<-c(1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0)
> pred<-prediction(y,x)
> perf<-performance(pred,'tpr','fpr')
> plot(perf)
```

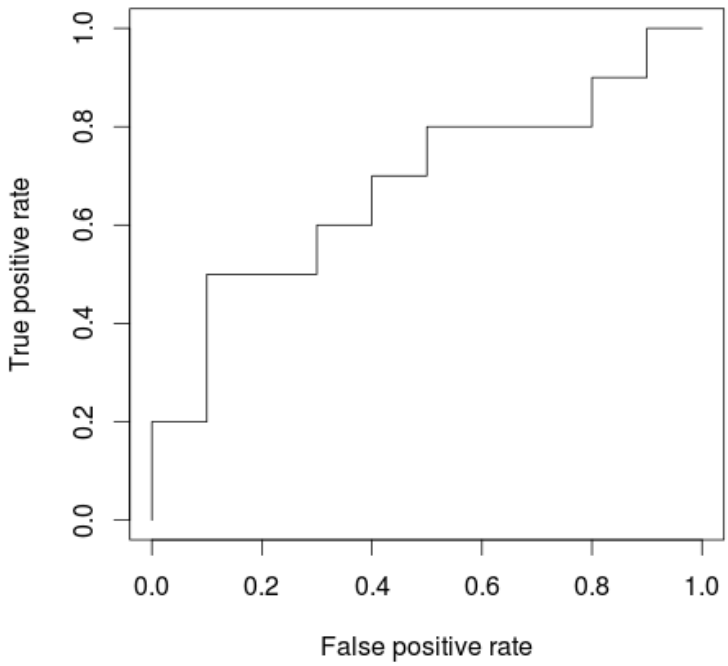

Figura 3: Curva ROC referente aos dados da Tabela 1. Obtida com o uso do pacote `ROCR` desenvolvido para o R. *False positive rate* são os valores de FPR e *True positive rate* são os valores de TPR.

Quanto melhor for o classificador, mais próximo do eixo TPR e da linha definida pelos pontos (0,1) e (1,1) a Curva ROC passará, ou seja, maior será a área abaixo da curva (AUC). Portanto, uma boa medida para mensurar o desempenho de um classificador é o seu valor AUC. Para calcular a AUC também pode-se usar o pacote `ROCR`. Para isso, o comando é apresentado abaixo.

```
> auc<-performance(pred,"auc")
> auc<-unlist(slot(auc, "y.values"))
> auc
[1] 0.68
```

Uma leitura sobre outras metodologias gráficas, além de Curvas ROC, para avaliar o desempenho de uma análise, pode ser encontrada em Prati *et al* (2011).

4 Metodologia

A metodologia deste trabalho consiste em duas partes. Primeiramente, em avaliar o desempenho da análise de identificação de DEGs, devido a um determinado fator *A*, na medida em que o número de fatores de um experimento é aumentado, enquanto o número de repetições é diminuído, mas de tal modo que o número de amostras em todos os experimentos seja o mesmo, quando os efeitos de todos os tipos de interações entre os fatores são não significativos. Em segundo lugar, avaliar o mesmo processo, porém quando os efeitos das interações entre os fatores são significativos. Para isso, foram simulados os dados dos experimentos e o desempenho de suas análises foram comparadas conforme explicado nos tópicos a seguir. Um esquema mostrando todo o processo empregado pela metodologia, resumidamente, é mostrado na Figura 4.

Os dados utilizados foram simulados e normalizados com o uso do pacote `TCC` (Sun *et al*, 2013). Para as análises de identificação dos DEGs, foi utilizado o pacote `DESeq` (Anders & Huber, 2010). Outros importantes pacotes para a realização dessas análises são: `edgeR` (Robinson *et al*, 2010), `baySeq` (Hardcastle & Kelly, 2010), entre outros. Todos os pacotes citados estão disponibilizado pelo projeto Bioconductor (Gentleman *et al*, 2004), desenvolvidos para o *software* livre R (R Core Team, 2014). Uma comparação entre os métodos desses e de outros principais pacotes para a análise de identificação de DEGs pode ser encontrado em Kvam *et al* (2012) e Soneson & Delorenzi (2013). Para mensurar o desempenho das análises, usou-se a *Área Abaixo da Curva* (AUC, do inglês *Area Under the Curve*) obtida através da Curva ROC (Fawcett, 2005; Prati *et al*, 2011). O cálculo da AUC foi feito através do pacote `ROCR` (Sing *et al*, 2005), também desenvolvido para o R.

4.1 Quando interação não significativa entre os fatores

4.1.1 Simulação dos dados

Os dados foram simulados por meio do pacote `TCC`, conforme os números de fatores e repetições indicados na Tabela 2 para cada experimento a comparar. Para tanto, primeiramente foi criado um único conjunto de dados com 3 fatores e 4 repetições, em que os efeitos da interação entre fatores são não significativos, e, a partir desse conjunto

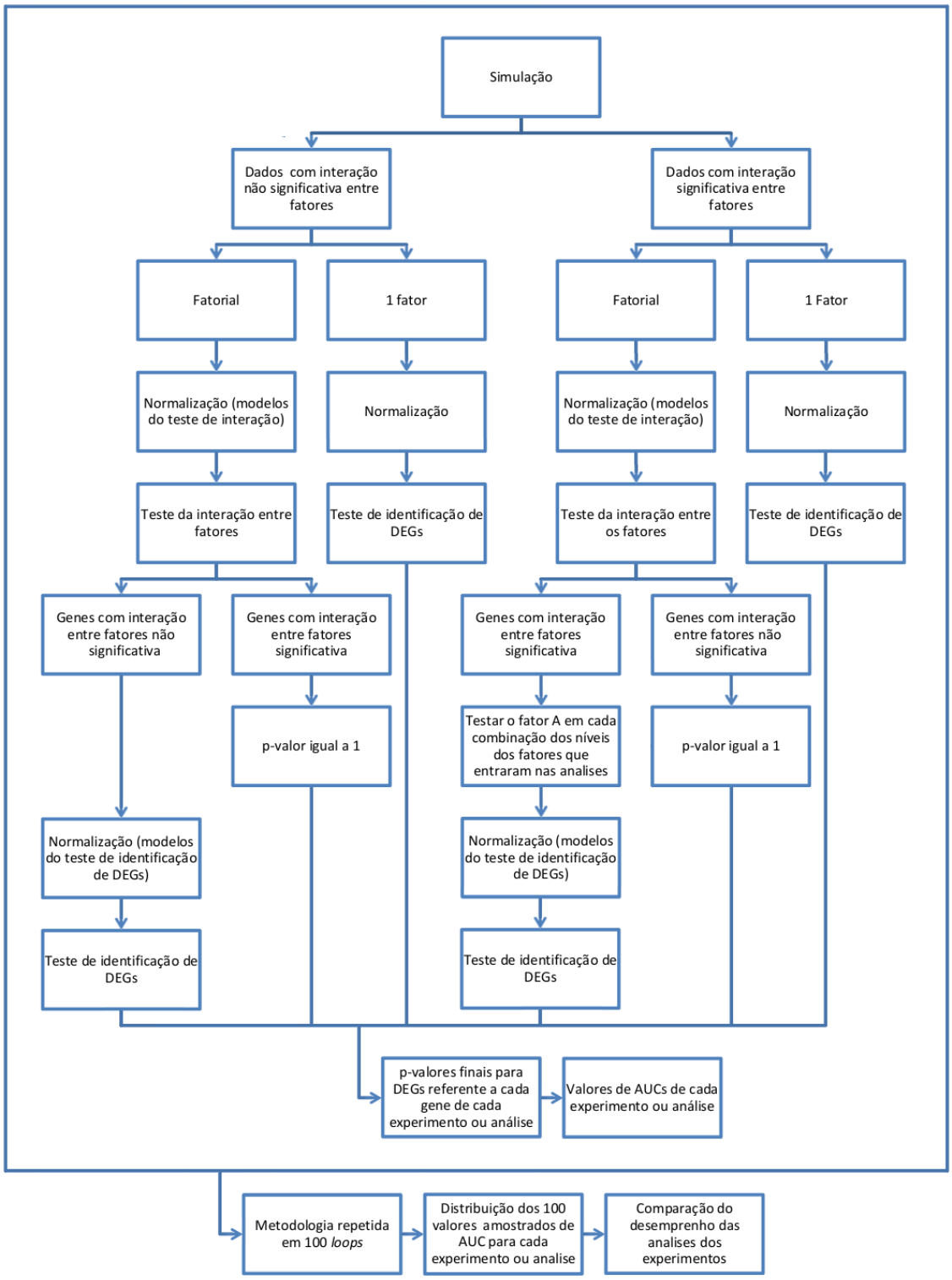

Figura 4: Esquema mostrando, resumidamente, todo processo empregado pela metodologia, para comparar o desempenho das análises quando os experimentos foram simulados de forma que os efeitos de interação entre fatores sejam não significativos e quando esses são significativos.

de dados em forma de matriz, colunas foram selecionadas de forma apropriada a formar os dados de 7 experimentos com diferentes números de fatores e repetições, nos quais foram comparados o desempenho de suas análises.

A Tabela 2 apresenta as características de cada um desses experimentos. Nela pode-se perceber que os experimentos de 1 a 4 são experimentos que apresentam apenas um fator. Para que os dados de um experimento com um fator (fator *A*) possa ser extraído de um conjunto de dados de 3 fatores, deve-se fixar os níveis dos outros dois fatores (fatores *B* e *C*) que não entrarão nas análises. Isso simula o que ocorre na prática, pois, quando um pesquisador planeja um experimento para testar um fator, ele deve definir e fixar os níveis dos outros fatores que não entrarão nas análises. Os experimentos de 1 a 4 simulam experimentos de um fator onde pesquisadores planejaram seus experimentos fixando níveis diferentes dos fatores que não entraram nas análises, ou seja, pesquisadores que querem concluir sobre o mesmo fator, porém estão seguindo protocolos diferentes. A mesma ideia segue para formar os experimentos 5 e 6, porém esses são experimentos com dois fatores (*A* e *B*), onde em cada um deles, um nível do fator *C* foi fixado. No experimento 7 todos os três fatores entraram nas análises (*A, B* e *C*) e, portanto, nenhum fator ficou de fora das análises para ser fixado.

Tabela 2: Descrição dos experimentos que tiveram o desempenho de suas análises comparadas.

	Nº de fatores[1]	Nº de repetições[2]	Total de amostras[3]	Níveis de *B* e *C* que foram fixados
Experimento 1	1	4	8	*b1c1*
Experimento 2	1	4	8	*b2c1*
Experimento 3	1	4	8	*b1c2*
Experimento 4	1	4	8	*b2c2*
Experimento 5	2	2	8	*c1*
Experimento 6	2	2	8	*c2*
Experimento 7	3	1	8	-

[1]Cada fator com dois níveis.
[2]Para cada nível do fator.
[3]O total de amostras em cada experimento é dado por: $n = i^f \cdot r$, em que i é o número de níveis do fator; f é o número de fatores; e r é o número de repetições.

Para a simulação do conjunto de dados com os efeitos da interação entre fatores não significativos, foram utilizados os seguintes comandos:

```
> library(TCC)
> group<-data.frame(A=rep(rep(paste('a',1:2,sep=''),c(16,16)),1),
+                   B=rep(rep(paste('b',1:2,sep=''),c(8,8)),2),
+                   C=rep(rep(paste('c',1:2,sep=''),c(4,4)),4))

#Dados sem interacao:
> DEG.foldchange<-data.frame(fc1=rep(c(1,4,4,7,4,7,7,10),rep(4,8)),
+                            fc2=rev(rep(c(1,4,4,7,4,7,7,10),rep(4,8))))
> tcc<-simulateReadCounts(Ngene = 1000,
+                         PDEG = .5,
+                         DEG.assign = c(.5,.5),
+                         DEG.foldchange = DEG.foldchange,
+                         group = group)
+ dados<-tcc$count
> colnames(dados)     #nome das colunas de dados
 [1] "a1b1c1_rep1" "a1b1c1_rep2" "a1b1c1_rep3" "a1b1c1_rep4"
 [5] "a1b1c2_rep1" "a1b1c2_rep2" "a1b1c2_rep3" "a1b1c2_rep4"
 [9] "a1b2c1_rep1" "a1b2c1_rep2" "a1b2c1_rep3" "a1b2c1_rep4"
[13] "a1b2c2_rep1" "a1b2c2_rep2" "a1b2c2_rep3" "a1b2c2_rep4"
[17] "a2b1c1_rep1" "a2b1c1_rep2" "a2b1c1_rep3" "a2b1c1_rep4"
[21] "a2b1c2_rep1" "a2b1c2_rep2" "a2b1c2_rep3" "a2b1c2_rep4"
[25] "a2b2c1_rep1" "a2b2c1_rep2" "a2b2c1_rep3" "a2b2c1_rep4"
[29] "a2b2c2_rep1" "a2b2c2_rep2" "a2b2c2_rep3" "a2b2c2_rep4"
```

Portanto, foi simulado dados de um experimento com: 1000 genes (`Ngene=1000`); os 500 primeiros genes são DEGs e os 500 últimos genes são *non*-DEGs (`PDEG=.5`); a expressão dos 50% primeiros DEGs (gene 1 até o gene 250) está de acordo com os valores de *fold change* indicados pela coluna `fc1` de `DEG.foldchange`, e a expressão dos 50% últimos genes que são DEGs (gene 250 até o gene 500) está de acordo com os valores de *fold change* indicados pela coluna `fc2` de `DEG.foldchange` (`DEG.assign=c(.5,.5)`); e existem 3 fatores e 4 repetições (`group=group`).

Como pode ser observado, a ordem dos valores de *fold change* dos 50% últimos DEGs é o inverso da ordem dos valores de *fold change* dos 50% primeiros DEGs. O porque de fazer isso é para evitar que as contagens não fiquem sempre maiores (*up-regulated*) quando um mesmo tratamento, pois, quando acontece esse desbalanceamento, é comum que as análises apresentam um baixo desempenho. Esse problema também é reportado em Robinson & Oshlack (2010). O método de normalização TCC, apesar de conseguir lidar com o desbalanceamento entre DEGs *up-regulated* e *down-regulated*, também não consegue impedir o baixo desempenho da

análise quando 100% dos DEGs são *up-regulated* (ou *down-regulated*) em um mesmo tratamento.

A seguir, será mostrado que, ao criar os dados dos experimentos de 1 a 7, conforme a Tabela 2, a partir dos dados simulados acima em `tcc`, a ordem dos valores de *fold change* dos 50% últimos DEGs também deverá ser o inverso dos valores de *fold change* dos 50% primeiros DEGs. O fato da ordem desses valores serem o inverso uma da outra, garantirá que o nível de diferenciação será o mesmo em todos os DEGs e apenas mudará em qual tratamento ocorre o *up-regulated* e o *down-regurated*.

Para que os experimentos de 1 a 7, estejam de acordo com a Tabela 2 e que a ordem dos valores de *fold change* dos 50% últimos DEGs seja o inverso da ordem dos valores de *fold change* dos 50% primeiros DEGs, selecionou-se colunas do *data frame* `tcc$count` do seguinte modo:

- Para o Experimento 1:

```
> dados<-rbind(tcc$count[1:250,c(c(1,2,3,4),c(17,18,19,20))],
+              tcc$count[251:500,c(c(13,14,15,16),c(29,30,31,32))],
+              tcc$count[501:1000,c(c(1,2,3,4),c(17,18,19,20))])
> condition<-gl(2,4)       #para 1 fator e 4 repeticoes

#Funcao para criar objeto do tipo TCC a partir de dados existentes:
> tcc1<-new('TCC',
+           dados,
+           condition)


> colnames(dados)
[1] "a1b1c1_rep1" "a1b1c1_rep2" "a1b1c1_rep3" "a1b1c1_rep4" "a2b1c1_rep1"
"a2b1c1_rep2" "a2b1c1_rep3" "a2b1c1_rep4"
```

Portanto o único fator que está variando entre os tratamentos é o fator *A*. Para o experimento 1, `tcc1` contém dados de um experimento com apenas 1 fator (*A*) e 4 repetições. O fator *B* está fixo no nível *b1* e o fator *C* no nível *c1*, por isso, não entraram nas análises.

- Para o Experimento 2:

```
> dados<-rbind(tcc$count[1:250,c(c(9,10,11,12),c(25,26,27,28))],
+              tcc$count[251:500,c(c(5,6,7,8),c(21,22,23,24))],
+              tcc$count[501:1000,c(c(9,10,11,12),c(25,26,27,28))])
> condition<-gl(2,4)       #para 1 fator e 4 repeticoes
```

```
> tcc1<-new('TCC',
+           dados,
+           condition)
> colnames(dados)
[1] "a1b2c1_rep1" "a1b2c1_rep2" "a1b2c1_rep3" "a1b2c1_rep4" "a2b2c1_rep1"
"a2b2c1_rep2" "a2b2c1_rep3" "a2b2c1_rep4"
```

Para o experimento 2, `tcc1` contém dados de um experimento com 1 fator (*A*) e 4 repetições. Os fatores *B* e *C* estão fixados nos níveis *b2* e *c1*, respectivamente.

- Para o Experimento 3:

```
> dados<-rbind(tcc$count[1:250,c(c(5,6,7,8),c(21,22,23,24))],
+              tcc$count[251:500,c(c(9,10,11,12),c(25,26,27,28))],
+              tcc$count[501:1000,c(c(5,6,7,8),c(21,22,23,24))])
> condition<-gl(2,4)       #para 1 fator e 4 repeticoes
> tcc1<-new('TCC',
+           dados,
+           condition)
> colnames(dados)
[1] "a1b1c2_rep1" "a1b1c2_rep2" "a1b1c2_rep3" "a1b1c2_rep4" "a2b1c2_rep1"
"a2b1c2_rep2" "a2b1c2_rep3" "a2b1c2_rep4"
```

Para o experimento 3, `tcc1` contém dados de um experimento com 1 fator (*A*) e 4 repetições. Os fatores *B* e *C* estão fixados nos níveis *b1* e *c2*, respectivamente.

- Para o Experimento 4:

```
> dados<-rbind(tcc$count[1:250,c(c(13,14,15,16),c(29,30,31,32))],
+              tcc$count[251:500,c(c(1,2,3,4),c(17,18,19,20))],
+              tcc$count[501:1000,c(c(13,14,15,16),c(29,30,31,32))])
> condition<-gl(2,4)       #para 1 fator e 4 repeticoes
> tcc1<-new('TCC',
+           dados,
+           condition)
> colnames(dados)
[1] "a1b2c2_rep1" "a1b2c2_rep2" "a1b2c2_rep3" "a1b2c2_rep4" "a2b2c2_rep1"
"a2b2c2_rep2" "a2b2c2_rep3" "a2b2c2_rep4"
```

Para o experimento 4, `tcc1` contém dados de um experimento com 1 fator (*A*) e 4 repetições. Os fatores *B* e *C* estão fixados nos níveis *b2* e *c2*, respectivamente.

- Para o Experimento 5:

```
> dados<-rbind(tcc$count[1:250,c(c(1,2),c(9,10),c(17,18),c(25,26))],
+              tcc$count[251:500,c(c(5,6),c(13,14),c(21,22),c(29,30))],
+              tcc$count[501:1000,c(c(1,2),c(9,10),c(17,18),c(25,26))])

#2 fatores e 2 repeticoes:
> design<-data.frame(A=rep(rep(paste('a',1:2,sep=''),c(4,4)),1),
+                    B=rep(rep(paste('b',1:2,sep=''),c(2,2)),2))

> tcc1<-new('TCC',
+           dados,
+           design)

> colnames(dados)

[1] "a1b1c1_rep1" "a1b1c1_rep2" "a1b2c1_rep1" "a1b2c1_rep2" "a2b1c1_rep1"
"a2b1c1_rep2" "a2b2c1_rep1" "a2b2c1_rep2"
```

Portanto, apenas os fatores *A* e *B* estão variando entre os tratamentos. Para o experimento 5, `tcc1` contém dados de um experimento com 2 fatores (*A* e *B*) e 2 repetições. O fator *C* está fixo no nível c*1* e, por isso, não entrou nas análises.

- Para o Experimento 6:

```
> dados<-rbind(tcc$count[1:250,c(c(5,6),c(13,14),c(21,22),c(29,30))],
+              tcc$count[251:500,c(c(1,2),c(9,10),c(17,18),c(25,26))],
+              tcc$count[501:1000,c(c(5,6),c(13,14),c(21,22),c(29,30))])

> design<-data.frame(A=rep(rep(paste('a',1:2,sep=''),c(4,4)),1),
+                    B=rep(rep(paste('b',1:2,sep=''),c(2,2)),2))

> tcc1<-new('TCC',
+           dados,
+           design)

> colnames(dados)

[1] "a1b1c2_rep1" "a1b1c2_rep2" "a1b2c2_rep1" "a1b2c2_rep2" "a2b1c2_rep1"
"a2b1c2_rep2" "a2b2c2_rep1" "a2b2c2_rep2"
```

Para o experimento 6, `tcc1` contém dados de um experimento com 2 fatores (*A* e *B*) e 2 repetições. O fator *C* está fixado no nível *c2*.

- Para o Experimento 7:

```
> dados<-tcc$count[,c(1,5,9,13,17,21,25,29)]

> design<-data.frame(A=rep(rep(paste('a',1:2,sep=''),c(4,4)),1),
+                    B=rep(rep(paste('b',1:2,sep=''),c(2,2)),2),
+                    C=rep(rep(paste('c',1:2,sep=''),c(1,1)),4))

> tcc1<-new('TCC',
+           dados,
+           design)
```

```
> colnames(dados)
[1] "a1b1c1_rep1" "a1b1c2_rep1" "a1b2c1_rep1" "a1b2c2_rep1" "a2b1c1_rep1"
"a2b1c2_rep1" "a2b2c1_rep1" "a2b2c2_rep1"
```

Portanto, os três fatores *A*, *B* e *C* estão variando entre os tratamentos. Para o experimento 7, `tcc1` contém dados de um experimento com 3 fatores (*A*, *B* e *C*) e apenas 1 repetição. Nenhum fator foi preciso ser fixado.

A Figura 5 mostra os valores de *fold change* resultantes para cada um dos experimentos de 1 a 7, após a realização de toda a metodologia descrita acima.

4.1.2 Normalização dos dados e identificação dos genes em que os efeitos da interação entre os fatores são não significativos

Quando o experimento é fatorial, antes de testar quais genes são DEGs, deve-se primeiro testar se existe interação entre os fatores. Pois, se a interação for significativa, não se pode testar os fatores separadamente e sim testar o efeito de um fator em cada nível dos outros fatores. Além disso, ao trabalhar com dados de RNA-Seq, é extremamente recomendado normalizar os dados. Sendo assim, antes de testar o efeito da interação, deve-se fazer uma normalização apropriado a esse teste. Como os experimentos de 1 a 4 possuem apenas um tratamento e os experimentos de 5 a 7 são fatoriais, a linha de comando difere um pouco entre esses dois casos. Para realizar a normalização dos dados foram utilizadas as funções do pacote `TCC`, e para testar o efeito da interação foram utilizadas as funções implementadas pelo pacote `DESeq`.

4.1.2.1 Quando experimento com apenas 1 fator

Para os experimentos de 1 a 4, que possuem apenas um fator, não é preciso testar o efeito da interação, portanto, já se pode realizar a normalização dos dados de acordo com o teste de identificação de DEGs, para isso, os comandos utilizados foram apresentados no tópico (4.1.3).

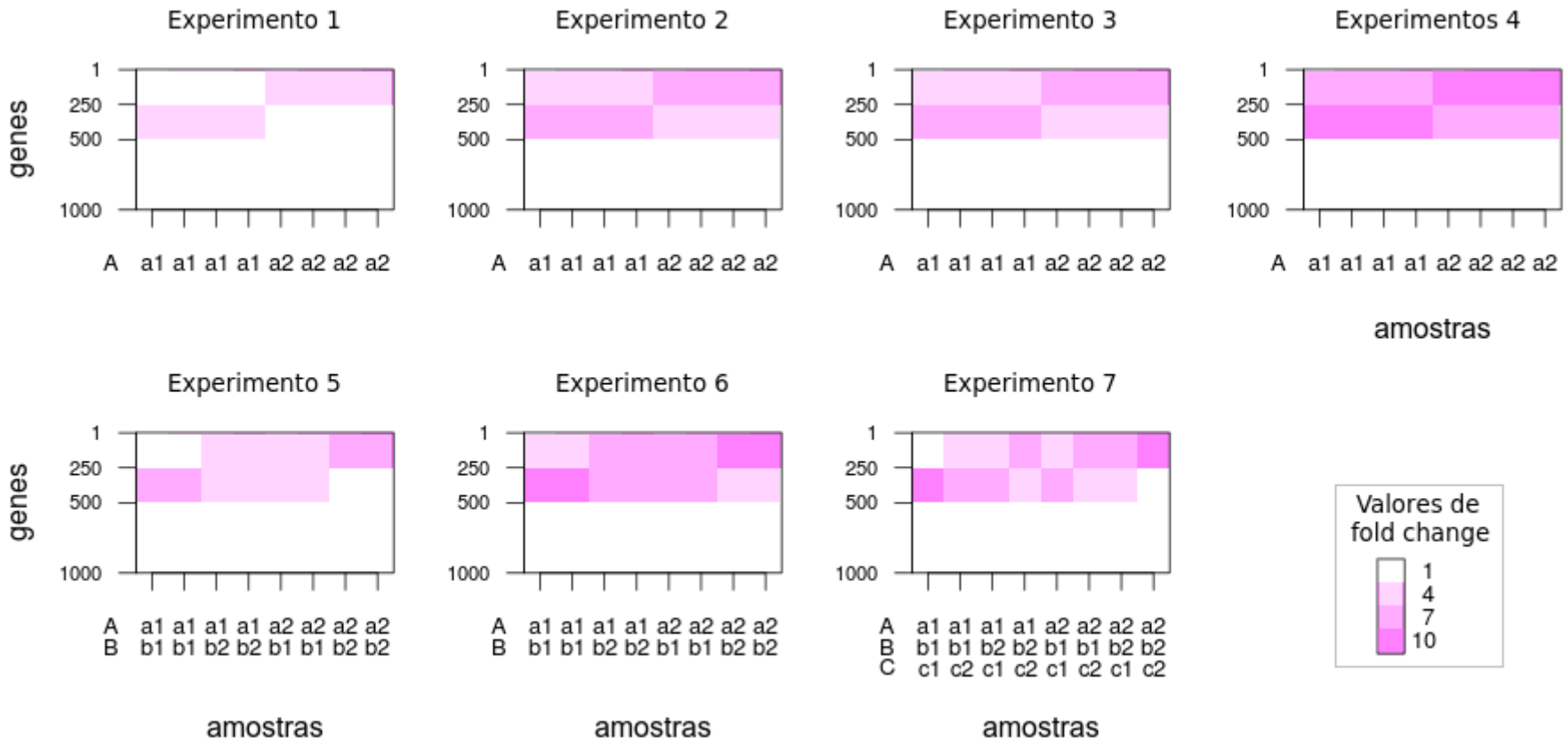

Figura 5: Representação gráfica dos valores resultantes de *fold change* usados como parâmetros para a simulação dos dados dos experimentos de 1 a 7, onde os efeitos de interação entre fatores são não significativos. O eixo vertical indica os 1000 genes simulados para cada experimento, e o eixo horizontal indica os tratamentos (combinações dos níveis dos fatores). Maiores tons de rosa indicam um maior valor de *fold change*. (Figura desenvolvida com o auxílio da função `plotFCPseudocolor()` do pacote `TCC`, porém com alterações.)

4.1.2.2 Quando experimento fatorial

Já os experimentos de 5 a 7 são experimentos fatoriais (experimentos 5 e 6 com dois fatores e experimento 7 com três fatores). Portanto, para esses experimentos, deve-se realizar uma normalização dos dados de acordo com o teste de interação e, em seguida, testar a interação. Ou seja, para realizar a normalização, deve-se indicar o modelo completo (`modelo.completo`) e o modelo reduzido (`modelo.reduzido`) como os mesmos que serão utilizados ao testar se o efeito da interação é significativo.

4.1.3 Normalização dos dados e identificação dos genes em que os efeitos da interação entre os fatores são não significativos

Quando o experimento é fatorial, antes de testar quais genes são DEGs, deve-se primeiro testar se existe interação entre os fatores. Pois, se a interação for significativa, não se pode testar os fatores separadamente e sim testar o efeito de um fator em cada nível dos outros fatores. Além disso, ao trabalhar com dados de RNA-Seq, é extremamente recomendado normalizar os dados. Sendo assim, antes de testar o efeito da interação, deve-se fazer uma normalização apropriado a esse teste. Como os experimentos de 1 a 4 possuem apenas um tratamento e os experimentos de 5 a 7 são fatoriais, a linha de comando difere um pouco entre esses dois casos. Para realizar a normalização dos dados foram utilizadas as funções do pacote `TCC`, e para testar o efeito da interação foram utilizadas as funções implementadas pelo pacote `DESeq`.

4.1.3.1 Quando experimento com apenas 1 fator

Para os experimentos de 1 a 4, que possuem apenas um fator, não é preciso testar o efeito da interação, portanto, já se pode realizar a normalização dos dados de acordo com o teste de identificação de DEGs, para isso, os comandos utilizados foram apresentados no tópico (4.1.3).

4.1.3.2 Quando experimento fatorial

Já os experimentos de 5 a 7 são experimentos fatoriais (experimentos 5 e 6 com dois fatores e experimento 7 com três fatores). Portanto, para esses experimentos, deve-se realizar uma normalização dos dados de acordo com o teste de interação e, em seguida, testar a interação. Ou seja, para realizar a normalização, deve-se indicar o modelo completo (`modelo.completo`) e o modelo reduzido (`modelo.reduzido`) como os mesmos que serão utilizados ao testar se o efeito da interação é significativo.

4.1.3.2.1 Quando experimento com 2 fatores

Para testar o efeito da interação dos experimentos 5 e 6, que possuem dois fatores, o modelo

$$y_{ijkg} = \mu_g + \alpha_{ig} + \beta_{jg} + \delta_{ijg} + e_{ijkg} \quad ,$$

em que,

- y_{ijkg} é o valor de contagem observado quando *i*-ésimo nível do fator *A*, *j*-ésimo nível do fator *B*, *k*-ésima repetição, para o gene *g*;
- μ_g é a média dos valores de contagem de todas as amostras do gene *g*;
- α_{ig} é o efeito do *i*-ésimo nível do fator *A*, no gene *g*;
- β_{jg} é o efeito do *j*-ésimo nível do fator *B*, no gene *g*;
- δ_{ijg} é o efeito da interação entre o *i*-ésimo nível do fator *A* e o *j*-ésimo nível do fator *B*, no gene *g*;
- e_{ijkg} é o erro aleatório quando *i*-ésimo nível do fator *A*, *j*-ésimo nível do fator *B*, *k*-ésima repetição, para o gene *g*,

é o modelo completo, e o modelo

$$y_{ijkg} = \mu_g + \alpha_{ig} + \beta_{jg} + e_{ijkg} \quad ,$$

é o modelo reduzido.

No R, os objetos `modelo.completo` e `modelo.reduzido`, do tipo *formula*, foram

formados da seguinte forma:

```
> modelo.completo<-count~A*B          #Para o modelo completo
> modelo.reduzido<-count~A+B          #Para o modelo reduzido
```

Para realizar a normalização dos dados, anterior ao teste da interação, dos experimentos 5 e 6, foram utilizados os seguintes comandos:

```
#Realizando a normalizacao:
> tcc1<-calcNormFactors(tcc1,
+                       norm.method = 'deseq',    #Etapa 1 da normalizacao
+                       test.method = 'deseq',    #Etapa 2 da normalizacao
+                       iteration = 3,            #Usando 3 iteracoes.
+                       fit1 = modelo.completo,
+                       fit0 = modelo.reduzido)

#Como o TCC (assim como o edgeR) trabalho com Normalization Factors e o
#  DESeq trabalha com Size Factors, eh preciso fazer uma transformacao antes
#  de poder usar as funcoes do DESeq.
> lib.sizes <- tcc1$norm.factors * colSums(tcc1$count)
> size.factors <- lib.sizes / mean(lib.sizes)    #Size Factors obtidos a partir
                                                 #  dos Normalization Factors.

#A partir daqui usa-se as funcoes do pacote DESeq.
> cds<-newCountDataSet(tcc1$count,design)        #Funcao do DESeq para entrada
                                                 #  dos dados.
> sizeFactors(cds) <- size.factors               #Atribuindo os valores de Size
                                                 #  Factors e a normalizacao
                                                 #  fica concluida.
```

E a seguir são mostrados os comandos (pacote `DESeq`) para testar o efeito da interação entre os fatores:

```
> cds<-estimateDispersions(cds)                #Calcula a estimativa do parametro
                                               #  de dispersao para cada gene.
> fit1<-fitNbinomGLMs(cds,modelo.completo)     #Calcula a estimativa dos
                                               #  coeficientes e a deviance do
                                               #  modelo linear generalizado
                                               #  completo para cada gene.
> fit0<-fitNbinomGLMs(cds,modelo.reduzido)     #Calcula a estimativa dos
                                               #  coeficientes e a deviance do
                                               #  modelo linear generalizado
                                               #  reduzido para cada gene.
> pvalsGLM<-nbinomGLMTest(fit1,fit0)           #Retorna o p-valor para a
                                               #  comparacao do modelo completo
                                               #  com o reduzido.
> pvalsGLM[is.na(pvalsGLM)]<-1                 #Eh comum ocorrer o valor NA para
                                               #  alguns p-valores, e estes devem
                                               #  ser modificados para o valor 1.
```

```
> padjGLM<-p.adjust(pvalsGLM,method="BH")      #Ajustamento do p-valor para
                                               #  testes multiplos, utilizando o
                                               #  metodo Benjamini-Hochberg.
```

Então `padjGLM` é um vetor com os *p-valores*, ajustados para testes múltiplos pelo método Benjamini-Hochberg (Benjamini & Hochberg, 1995), resultantes do teste da interação, referente a cada gene. Para concluir se o efeito da interação é significativo, utilizou-se o valor 0,1 como valor crítico para os *p-valores* ajustados, sendo assim, rejeitou-se a hipótese dos fatores serem independentes (efeito da interação entre fatores ser não significativo) quando o *p-valor* ajustado for menor ou igual a 0,1 e não rejeita-se essa hipótese caso contrário. Após testar o efeito da interação, foi selecionado aqueles genes onde se concluiu que a interação é não significativa, para que possa ser testado, em apenas esses, o efeito do fator *A* no nível de expressão gênica, separadamente.

Para selecionar os genes em que a interação foi concluída como não significativa, foram utilizados os seguintes comandos:

```
> linhas<-padjGLM > 0.1
> dados<-tcc1$count[linhas,]    #Nova matriz dos dados, com apenas os genes em
                                #  que a interacao foi concluida como não
                                #  significativa.
> tcc1<-new('TCC',              #Criando novamente o objeto tcc1
+           dados,
+           design)
```

4.1.3.2.2 Quando experimento com 3 fatores

Quando o experimento tem 3 fatores (experimento 7), primeiramente deve-se testar a interação de ordem tripla. Para isso, o modelo

$$y_{ijlkg} = \mu_g + \alpha_{ig} + \beta_{jg} + \gamma_{lg} + \delta_{ijg} + \delta_{ilg} + \delta_{jlg} + \phi_{ijlg} + e_{ijlkg} \quad ,$$

em que,

y_{ijlkg} é o valor de contagem observado quando *i*-ésimo nível do fator *A*, *j*-ésimo nível do fator *B*, *l*-ésimo nível do fator *C*, *k*-ésima repetição, para o gene *g*;

μ_g é a média dos valores de contagem de todas as amostras do gene *g*;

α_{ig} é o efeito do *i*-ésimo nível do fator *A*, no gene *g*;

β_{jg} é o efeito do *j*-ésimo nível do fator *B*, no gene *g*;

γ_{lg} é o efeito do *l*-ésimo nível do fator *C*, no gene *g*;

δ_{ijg} é o efeito da interação dupla entre o *i*-ésimo nível do fator *A* e o *j*-ésimo nível do fator *B*, no gene *g*;

δ_{ilg} é o efeito da interação dupla entre o *i*-ésimo nível do fator *A* e o *l*-ésimo nível do fator *C*, no gene *g*;

δ_{jlg} é o efeito da interação dupla entre o *j*-ésimo nível do fator *B* e o *l*-ésimo nível do fator *C*, no gene *g*;

ϕ_{ijlg} é o efeito da interação tripla entre o *i*-ésimo nível do fator *A*, *j*-ésimo nível do fator *B* e o *l*-ésimo nível do fator *C*, no gene *g*;

e_{ijlkg} é o erro aleatório quando *i*-ésimo nível do fator *A*, *j*-ésimo nível do fator *B*, *l*-ésimo nível do fator *C*, *k*-ésima repetição, para o gene *g*,

é o modelo completo, e o modelo

$$y_{ijlkg} = \mu_g + \alpha_{ig} + \beta_{jg} + \gamma_{lg} + \delta_{ijg} + \delta_{ilg} + \delta_{jlg} + e_{ijlkg} \quad ,$$

é o modelo reduzido.

No R, os objetos `modelo.completo` e `modelo.reduzido`, foram formados da seguinte forma:

```
> modelo.completo<-count~A*B*C                #Para o modelo completo
> modelo.reduzido<-count~A*B*C-A:B:C          #Para o modelo reduzido
```

Para realizar a normalização dos dados (anterior ao teste da interação tripla) e testar se a interação tripla é significativa, para cada gene, do experimento 7, foram utilizados os mesmos comandos mostrados no tópico (4.1.2.2.1), porém com os objetos `modelo.completo` e `modelo.reduzido` formados conforme acima.

Então `padjGLM` é um vetor com os *p-valores* ajustados, resultantes do teste da interação tripla, referente a cada gene. Assim, foram selecionados apenas aqueles genes que a interação tripla foi concluída como não significativa, ou seja, apenas as linhas da matriz `dados` onde `padjGLM` $\geq 0{,}1$, para que possa ser testado, em apenas esses, o efeito das interações duplas *A*x*B* e *A*x*C*, separadamente.

Para testar a interação dupla *A*x*B*, o modelo

$$y_{ijlkg} = \mu_g + \alpha_{ig} + \beta_{jg} + \gamma_{lg} + \delta_{ijg} + \delta_{ilg} + \delta_{jlg} + e_{ijlkg}$$

é o modelo completo, e o modelo

$$y_{ijlkg} = \mu_g + \alpha_{ig} + \beta_{jg} + \gamma_{lg} + \delta_{ilg} + \delta_{jlg} + e_{ijlkg}$$

é o modelo reduzido. Ou seja, em comandos no R:

```
> modelo.completo<-count~A*B*C-A:B:C              #Para o modelo completo
> modelo.reduzido<-count~A*B*C-A:B:C-A:B          #Para o modelo reduzido
```

E para testar a interação dupla *A*x*C*, o modelo

$$y_{ijlkg} = \mu_g + \alpha_{ig} + \beta_{jg} + \gamma_{lg} + \delta_{ijg} + \delta_{ilg} + \delta_{jlg} + e_{ijlkg}$$

também é o modelo completo, e o modelo

$$y_{ijlkg} = \mu_g + \alpha_{ig} + \beta_{jg} + \gamma_{lg} + \delta_{ijg} + \delta_{jlg} + e_{ijlkg}$$

é o modelo reduzido. Em comandos no R:

```
> modelo.completo<-count~A*B*C-A:B:C              #Para o modelo completo
> modelo.reduzido<-count~A*B*C-A:B:C-A:C          #Para o modelo reduzido
```

Portanto, a matriz `dados`, que será utilizada para criar `tcc1`, possui apenas as linhas referentes aos genes onde foram concluídos que todas os tipos interação que envolve o fator *A* (tripla ou duplas), são não significativas. E, sendo assim, nesses genes, o fator *A* pode ser analisado separadamente.

4.1.4 Normalização dos dados e análise do efeito do fator *A* no nível de expressão Gênica

Quando o experimento tiver apenas um fator, basta: (i) realizar uma única normalização; (ii) e, em seguida, identificar os DEGs (no caso deste trabalho, deseja-se identificar os genes que são DEGs devido ao fator *A*). E quando o experimento for fatorial deve-se: (i) realizar uma primeira normalização (tópico (4.1.2.2)), usando os modelos completo e reduzido que também serão utilizados para testar os efeitos das interações; (ii) testar as interações (tópico (4.1.2.2)); (iii) e depois fazer uma segunda normalização, usando os modelos completo e reduzido que serão utilizados para a análise de identificação de DEGs devido ao fator em questão; (iv) e, por fim, para apenas aqueles genes que foi concluído que a interação é não significativa, realizar o teste de identificação de DEGs, concluindo sobre o fator *A*, separadamente dos outros fatores.

4.1.4.1 Quando experimento com apenas 1 fator

Para realizar a normalização dos dados dos experimentos de 1 a 4, foram utilizados os seguintes comandos:

```
#Realizando a normalizacao:
> tcc1<-calcNormFactors(tcc1,
+                       norm.method = 'deseq',      #Etapa 1 da normalizacao
+                       test.method = 'deseq',      #Etapa 2 da normalizacao
+                       iteration = 3)              #Usando 3 iteracoes.

#Transformacao de Normalization Factors para  Size Factors:
> lib.sizes<-tcc1$norm.factors*colSums(tcc1$count)
> size.factors<-lib.sizes/mean(lib.sizes)

#A partir daqui usa-se as funcoes do pacote DESeq:
> cds<-newCountDataSet(tcc1$count,condition)
> sizeFactors(cds)<-size.factors                    #Normalizacao concluida.
```

Para realizar a identificação de DEGs dos experimentos de 1 a 4, deve-se comparar o modelo completo

$$y_{ijlkg} = \mu_g + \alpha_{ig} + e_{ijlkg} \quad ,$$

e o modelo reduzido

$$y_{ijlkg} = \mu_g + e_{ijlkg} \quad .$$

Para isso, foram utilizados os seguintes comandos, funções do pacote `DESeq`:

```
> cds<-estimateDispersions(cds)                 #Estimando os parametros de
                                                #  dispersao.
> fit1<-fitNbinomGLMs(cds,count~condition)      #Modelagem segundo o modelo linear
                                                #  generalizado completo.
> fit0<-fitNbinomGLMs(cds,count~1)              #Modelagem segundo o modelo linear
                                                #  generalizado reduzido.
> pvalsGLM<-nbinomGLMTest(fit1,fit0)            #Comparando o modelo completo com
                                                #  o reduzido.
> pvalsGLM[is.na(pvalsGLM)]<-1                  #Atribuindo 1 para p-valores
                                                #  iguais a NA.
> padjGLM<-p.adjust(pvalsGLM,method="BH")       #Ajustamento dos p-valores.
```

Então, quando o experimento tiver apenas um fator, `padjGLM` é um vetor com os *p-valores* (ajustados) resultantes da análise de identificação de DEGs. São estes que serão utilizados para mensurar o desempenho da análise, através de Curvas ROC.

4.1.4.2 Quando experimento fatorial

Para realizar a normalização (anterior à identificação dos DEGs) dos dados dos experimentos de 5 a 7, deve-se definir os modelos completo e reduzido:

Quando experimento com 2 fatores (experimentos 5 e 6), o modelo completo é

$$y_{ijlkg} = \mu_g + \alpha_{ig} + \beta_{jg} \quad ,$$

e o modelo reduzido é

$$y_{ijlkg} = \mu_g + \beta_{jg} \quad .$$

Em comandos no R:

```
> modelo.completo<-count~A+B            #Para o modelo completo.
> modelo.reduzido<-count~B              #Para o modelo reduzido.
```

Quando experimento com 3 fatores (experimento 7), o modelo completo é

$$y_{ijlkg} = \mu_g + \alpha_{ig} + \beta_{jg} + \gamma_{lg} \quad ,$$

e o modelo reduzido é

$$y_{ijlkg} = \mu_g + \beta_{jg} + \gamma_{lg} \quad .$$

Em comandos no R:

```
> modelo.completo<-count~A+B+C          #Para o modelo completo.
> modelo.reduzido<-count~B+C            #Para o modelo reduzido.
```

Assim, a normalização foi feita utilizando os seguintes comandos:

```
#Realizando a normalizacao:
> tcc1<-new('TCC',                 #O objeto tcc1 recebe o novo conjunto de
                                   #  dados que possui apenas aquelas linhas
                                   #  (genes) que a interacao foi concluida
                                   #  como nao significativa.
+           dados,
+           design)
> tcc1<-calcNormFactors(tcc1,
+                       norm.method='deseq',
+                       test.method='deseq',
+                       iteration=3,
+                       fit1=count~modelo.completo,
+                       fit0=count~modelo.reduzido)
> lib.sizes<-tcc1$norm.factors*colSums(tcc1$count)
> size.factors<-lib.sizes/mean(lib.sizes)
> cds<-newCountDataSet(tcc1$count,design)
> sizeFactors(cds)<-size.factors                 #Normalizacao concluida.
```

Para realizar a identificação de DEGs, devido ao fator *A*, nos experimentos de 5 a 7, foram utilizados os seguintes comandos:

```
> cds<-estimateDispersions(cds)
> fit1<-fitNbinomGLMs(cds,count~modelo.completo)
> fit0<-fitNbinomGLMs(cds,count~modelo.reduzido)
```

```
> pvalsGLM<-nbinomGLMTest(fit1,fit0)
> pvalsGLM[is.na(pvalsGLM)]<-1
> padjGLM<-p.adjust(pvalsGLM,method="BH")
```

Portanto, `padjGLM` é o vetor com os *p-valores* do teste de identificação de DEGs, para somente aqueles genes em que a interação foi concluída como não significativa. Porém, para que seja feito a Curva ROC, e com ela o desempenho da análise ser mensurado, atribuiu-se o *p-valor* igual a 1 para aqueles genes em que a interação (dupla ou tripla) foi concluída como significativa (falso positivo para interação significativa). Para isso foram utilizados os comandos:

```
> p<-rep(1,1000)
> p[linhas]<-padjGLM     #O objeto linhas eh um vetor com valores do tipo
                         #  logical, indicando verdadeiro ou falso para a
                         #  conclusao da interacao (dupla no caso quando 2
                         #  fatores, e dupla e tripla no caso quando 3
                         #  fatores) ser não significativa em cada gene.
                         #  O objeto p eh um vetor com 1000 p-valores,
                         #  referente a cada um dos genes 1000 genes
                         #  simulados inicialmente.
```

Então `p` é um vetor com os *p-valores* (ajustados) para o teste de identificação de DEGs. Eles são os valores resultantes no final da análise e são os valores que serão utilizados para mensurar o desempenho da análise utilizando Curvas ROC.

4.1.5 Calculando o desempenho das análises

Para mensurar o desempenho de cada análise, foi utilizado Curvas ROC, através do pacote `ROCR`. Para *valores de positividade* (em que DEG é a classe *positivo* e *non*-DEG é a classe *negativo*), foi utilizado o valor complementar do *p-valor* (1 – *p-valor*) retornado da análise para identificação de DEGs para cada gene, ou seja, foi utilizado o vetor `1-padjGLM` quando o experimento tiver apenas um fator e o vetor `1-p` quando o experimento for fatorial. Sendo assim, para cada um dos experimentos de 1 a 7 foi calculado seu valor de AUC. Os comandos utilizados são mostrados a seguir.

Quando experimento com apenas um fator (experimento de 1 a 4):

```
> real<-rep(c(1,0),c(500,500))
> pred<-prediction(1-padjGLM,real)     #Vetor 1-padjGLM eh usado como valores
                                       #  de positividade
> perf<-performance(pred,"tpr","fpr")
> auc<-performance(pred,"auc")
> auc<-unlist(slot(auc,"y.values"))
```

Quando experimento fatorial (experimento de 5 a 7):

```
> real<-rep(c(1,0),c(500,500))
> pred<-prediction(1-p,real)              #Vetor 1-p eh usado como valores
                                          #  de positividade
> perf<-performance(pred,"tpr","fpr")
> auc<-performance(pred,"auc")
> auc<-unlist(slot(auc,"y.values"))
```

Toda metodologia descrita no tópico (4.1) foi realizada em um *loop* de 100 vezes, formando assim 100 valores de AUC para cada um dos 100 experimentos de 1 a 7 simulados. Esses valores foram representados por uma matriz de ordem 100x7, em que cada coluna representa um experimento, e cada linha representa um *loop*. Com isso, pode-se comparar os experimentos de acordo com a distribuição de seus valores calculados para AUC.

4.2 Quando interação significativa entre os fatores

4.2.1 Simulação dos dados

Para a criação dos dados de experimentos com efeitos de interação entre fatores significativos, foi simulado um conjunto de dados com 3 fatores e 4 repetições, em que a interação é significativa, e, a partir dele, foram selecionadas as colunas apropriadas para a formação de cada um dos experimentos a terem o desempenho de suas análises comparados. Como no caso quando a interação é não significativa, as características desses experimentos também seguem conforme a Tabela 2. No entanto, devido à significância da interação entre fatores, não teremos apenas 7 análises para comparar o desempenho, e sim, um total de 12 análises a comparar. Pois, nos experimentos de 5 a 7,

que são fatoriais, deve-se analisar o efeito do fator *A* em cada nível dos outros fatores que entraram nas análises. A Tabela 3 mostra as características dessas análises (de 1 a 12) a comparar seu desempenho.

Tabela 3: Descrição das análises, que tiveram seu desempenho comparados, sobre experimentos em que os efeitos de interação entre fatores são significativos.

Análise	Experimento	Nº de fatores	Nº de repetições	Níveis dos fatores que não entraram na análise[1]	Níveis dos fatores que entraram na análise[2]
1	1	1	4	*b1c1*	-
2	2	1	4	*b2c1*	-
3	3	1	4	*b1c2*	-
4	4	1	4	*b2c2*	-
5	5	2	2	*c1*	*b1*
6	5	2	2	*c1*	*b2*
7	6	2	2	*c2*	*b1*
8	6	2	2	*c2*	*b2*
9	7	3	1	-	*b1c1*
10	7	3	1	-	*b2c1*
11	7	3	1	-	*b1c2*
12	7	3	1	-	*b2c2*

[1]Para formar os dados de um experimento, a partir de um conjunto de dados maior, deve-se fixar os níveis dos fatores que não iram entrar na análise.
[2]Quando o experimento for fatorial e existir interação entre os fatores, deve-se avaliar os efeitos do fator *A* em cada nível dos fatores que entraram na análise.

Para a simulação do conjunto de dados com interação significativa, de onde foram obtidos os dados dos 7 experimentos, foram utilizados os seguintes comandos:

```
> library(TCC)
> group<-data.frame(A=rep(rep(paste('a',1:2,sep=''),c(16,16)),1),
+                   B=rep(rep(paste('b',1:2,sep=''),c(8,8)),2),
+                   C=rep(rep(paste('c',1:2,sep=''),c(4,4)),4))

#Com interacao:
> DEG.foldchange<-data.frame(fc1=rep(c(1,4,4,1,4,1,1,4),rep(4,8)),
+                            fc2=rev(rep(c(1,4,4,1,4,1,1,4),rep(4,8))))
```

```
> tcc<-simulateReadCounts(Ngene = 1000,
+                         PDEG = .5,
+                         DEG.assign = c(.5,.5),
+                         DEG.foldchange = DEG.foldchange,
+                         group = group)
> dados<-tcc$count
```

Portanto, pode-se perceber que a única diferença entre o conjunto de dados gerado acima (com interação) e o conjunto de dados gerado no tópico (4.1.1) (sem interação) é o *data frame* `DEG.foldchange`, que é o argumento da função `simulateReadCounts()` que define se existe o efeito da interação entre os fatores.

Para realizar as análises de 1 a 12, deve-se formar os dados dos experimentos de 1 a 7 conforme a Tabela 2, de modo similar aos comandos apresentados no tópico (4.1.1). Porém não foi preciso se preocupar com que a ordem dos valores de *fold change* dos 50% últimos DEGs fossem o inverso da dos 50% primeiros DEGs, pois isso já acontece naturalmente, devido a como o *data frame* `DEG.foldchange` foi definido. Assim, para formar os experimentos de 1 a 7, foram utilizados os mesmos comandos descritos no tópico (4.1.1) para esse propósito, porém selecionando as colunas do conjunto de dados `tcc$count` conforme comandos apresentdos na Tabela 4.

Sendo assim, os valores resultantes de *fold change*, usados como parâmetros para a simulação, após o uso de toda essa metodologia, para a formação dos dados dos experimentos de 1 a 7, em que os efeitos de interação entre fatores são significativos, são os valores conforme mostrados na Figura 6.

Tabela 4: Comandos utilizados para selecionar colunas do conjunto de dados simulado, de forma apropriada a formar os dados de cada experimento.

Experimento	Colunas selecionadas do conjunto de dados[1]
Experimento 1	`dados<-tcc$count[,c(1,2,3,4, 17,18,19,20)]`
Experimento 2	`dados<-tcc$count[,c(9,10,11,12, 25,26,27,28)]`
Experimento 3	`dados<-tcc$count[,c(5,6,7,8, 21,22,23,24)]`
Experimento 4	`dados<-tcc$count[,c(13,14,15,16, 29,30,31,32)]`
Experimento 5	`dados<-tcc$count[,c(1,2, 9,10, 17,18, 25,26)]`
Experimento 6	`dados<-tcc$count[,c(5,6, 13,14, 21,22, 29,30)]`
Experimento 7	`dados<-tcc$count[,c(1, 5, 9, 13, 17, 21, 25, 29)]`

[1]Comandos do *software* R.

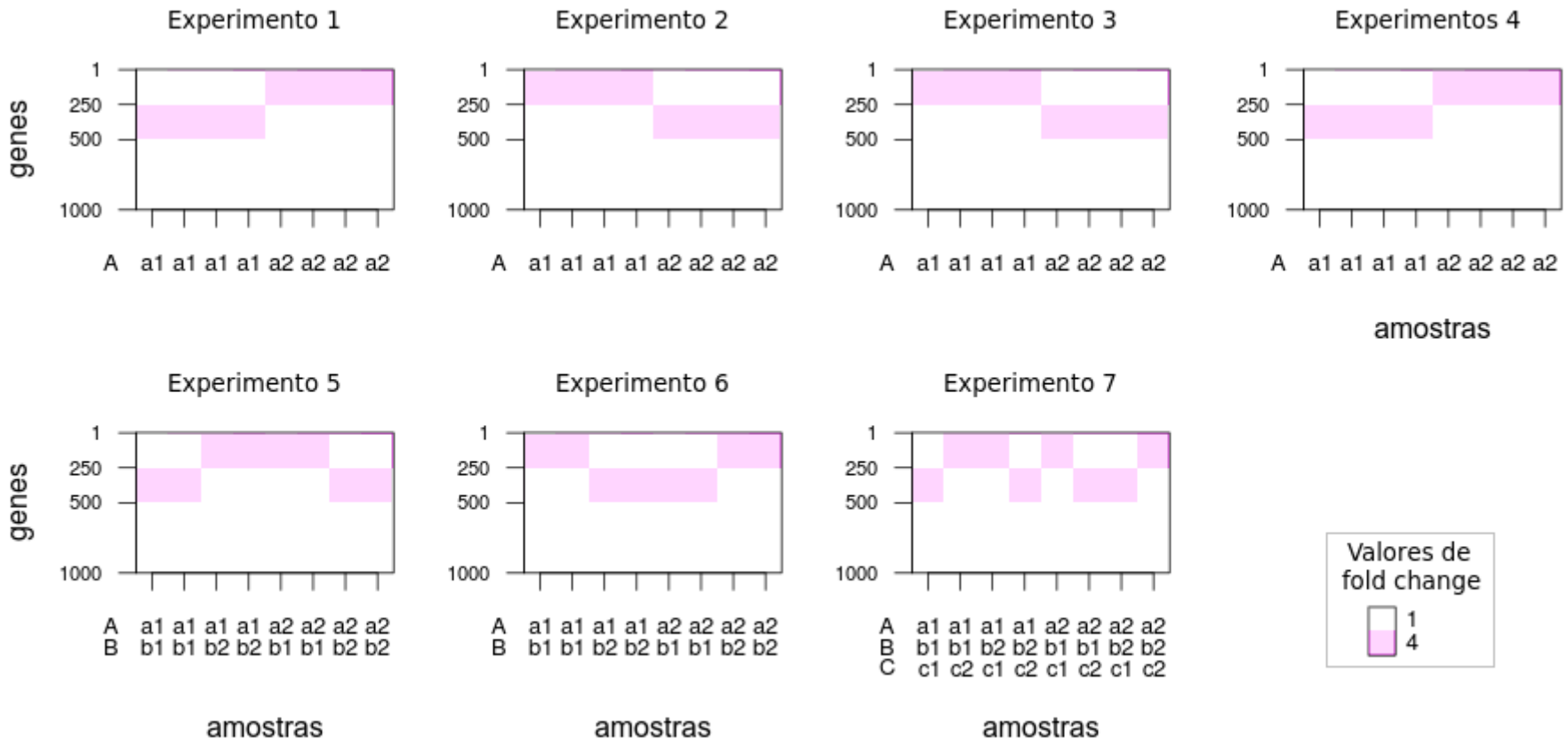

Figura 6: Representação gráfica dos valores resultantes de *fold change*, usados como parâmetros para a simulação, dos dados dos experimentos de 1 a 7, onde os efeitos de interação entre fatores são significativos. O eixo vertical indica os 1000 genes simulados para cada experimento, e o eixo horizontal indica os tratamentos (combinações dos níveis dos fatores). Maiores tons de rosa indicam um maior valor do *fold change*. (Figura desenvolvida com o auxílio da função `plotFCPseudocolor()` do pacote `TCC`, porém com alterações.)

4.2.2 Normalização dos dados e identificação dos genes em que os efeitos da interação entre os fatores são significativos

4.2.2.1 Quando experimento com apenas 1 fator

Assim como no tópico (4.1.2.1), para os experimentos de 1 a 4, que possuem apenas um fator, não é preciso testar a significância dos efeitos de interação entre fatores. Sendo assim, para esses experimentos, pode-se avançar diretamente para o próximo passo, que é a já identificação dos DEGs, procedimento que será mostrado no tópico (4.2.3.1).

4.2.2.2 Quando experimento fatorial

Para a normalização dos dados (normalização visando o teste da interação) e teste da significância do efeito de interação entre fatores, o procedimento realizado foi o mesmo utilizado com os dados com interação não significativa (tópico (4.1.2.2)). Porém, ao contrário de selecionar os genes que a interação é não significativa, deseja-se selecionar aqueles genes em que a interação é significativa, para que nestes genes possa-se testar o efeito do fator *A* em cada nível dos outros fatores que entraram na análise. Sendo assim, após a criação do vetor com os *p-valores* ajustados `padjGLM`, foram utilizados os seguintes comandos:

```
> linhas<-padjGLM < 0.1          #Valor TRUE para genes com interacao
                                 #  significativa.
> dados<-dados[linhas,]          #Nova matriz dos dados, com apenas os genes em
                                 #  que a interacao foi concluida como
                                 #  significativa.
```

A nova matriz `dados` passa a conter apenas aquelas linhas, dos dados de um determinado experimento, referentes aos genes nos quais foi concluído que o efeito da interação entre fatore é significativo. É essa matriz de dados que foi utilizada na próxima etapa para fazer a análise de identificação de DEGs.

4.2.3 Normalização dos dados e análise do efeito do fator *A* no nível de expressão Gênica

4.2.3.1 Quando experimento com apenas 1 fator

Para realizar a análise de identificação de DEGs quando se tem apenas um tratamento (fator *A*), ou seja, no caso dos experimentos de 1 a 4, o mesmo foi realizado quando no caso que a interação é não significativa (tópico (4.1.3.1)). Como existe apenas um fator, não é preciso analisá-lo em cada nível dos outros fatores e, sendo assim, tem-se apenas uma análise para cada um dos 4 experimentos. As análises de 1 a 4 referem-se aos experimentos de 1 a 4, respectivamente.

4.2.3.2 Quando experimento fatorial

Para realizar a análise de identificação de DEGs dos experimentos 5 e 6, que são experimentos com 2 fatores (fatores *A* e *B*), deve-se avaliar o efeito do fator *A*, no nível de expressão, em cada nível do fator *B* (nível *b1* e *b2*). Assim, tem-se duas análises para cada um desses experimentos. As análises 5 e 6 referem-se ao experimento 5 e as análises 7 e 8 referem-se ao experimento 6. Já no experimento 7, que é um experimento com 3 fatores (*A*, *B* e *C*), deve-se avaliar o fator *A* em cada combinação de níveis de *B* e *C* (*b1c1*, *b2c1*, *c2b1*, *b2c2*). Portanto, tem-se 4 análises para o experimento 7, que serão as análises de 9 a 12. Os comandos utilizados para formar os dados para as análises de 5 a 12, a partir da nova matriz `dados` criada no tópico (4.2.2.2), são mostrados na Tabela 5.

A Figura 7 mostra os valores de *fold change*, usados como parâmetros para a simulação, para os dados das análises de 1 a 12. Como pode ser observado, a única diferença entre as análises com 1, 2 ou 3 fatores são seus respectivos números de repetições. Quanto mais fatores estiverem envolvidos nas análises, menor será o número de repetições.

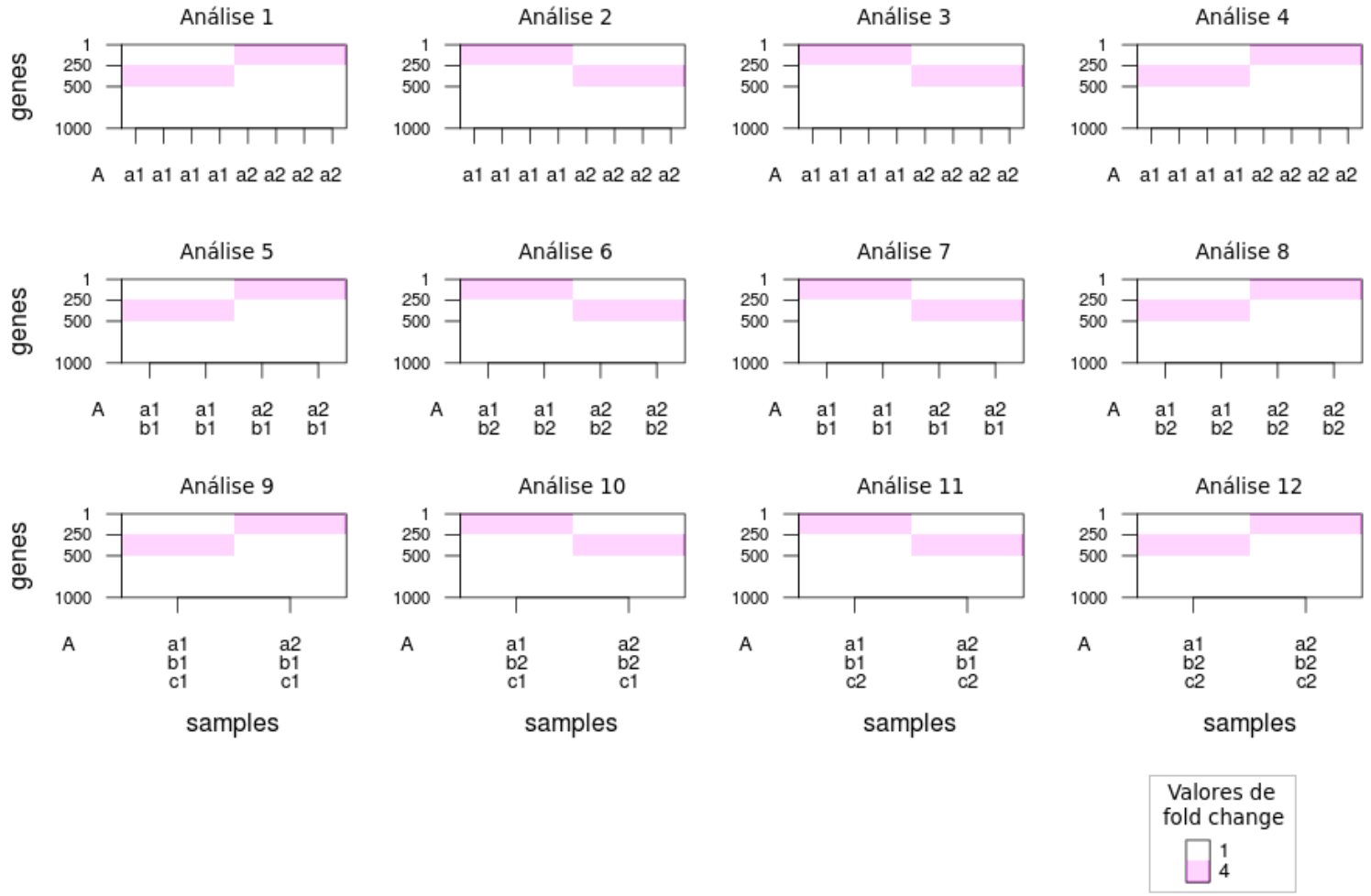

Figura 7: Representação gráfica dos valores de *fold change* usados como parâmetros para a simulação dos dados das análises de 1 a 12, quando o efeito do fator *A* está sendo testado em cada nível dos outros fatores que entraram nas análises. O eixo vertical indica os 1000 genes simulados para cada experimento, e o eixo horizontal indica os tratamentos (combinações dos níveis dos fatores). Maiores tons de rosa indicam um maior valor de *fold change*. (Figura desenvolvida com o auxílio da função `plotFCPseudocolor()` do pacote `TCC`, porém com alterações.)

Tabela 5: Comandos utilizados para selecionar as colunas dos dados de um experimento fatorial, quando efeito de interação entre fatores é significativo, para que o fator *A* possa ser analisado dentro dos níveis dos outros fatores que entraram na análise.

Experimento	Análises	Níveis dos fatores que entraram na análise[1]	Colunas selecionadas da matriz `dados`[2]
Experimento 5	Análise 5	*b1*	`dados1<-dados[,c(1,2, 5,6)]`
Experimento 5	Análise 6	*b2*	`dados1<-dados[,c(3,4, 7,8)]`
Experimento 6	Análise 7	*b1*	`dados1<-dados[,c(1,2, 5,6)]`
Experimento 6	Análise 8	*b2*	`dados1<-dados[,c(3,4, 7,8)]`
Experimento 7	Análise 9	*b1c1*	`dados1<-dados[,c(1,5)]`
Experimento 7	Análise 10	*b2c1*	`dados1<-dados[,c(3,7)]`
Experimento 7	Análise 11	*b1c2*	`dados1<-dados[,c(2,6)]`
Experimento 7	Análise 12	*b2c2*	`dados1<-dados[,c(4,8)]`

[1]Como existe interação, o fator A deve ser analisado dentro de cada nível dos fatores que entraram na análise.
[2]Comandos utilizados para formar os dados para analisar o fator *A* quando o respectivo nível dos fatores que entraram na análise. A matriz `dados` é a matriz dos dados de um experimento que possui apenas aquelas linhas que correspondam aos genes em que foi concluído que o efeito de interação entre fatores é significativo (conforme tópico (4.2.2.2)).

Uma importante observação a ser feita, é que existe uma situação em que o uso do pacote `TCC`, para realizar a normalização, provoca uma perda do desempenho da análise. Isso acontece quando, em um experimento fatorial, deve-se analisar separadamente os genes em que a conclusão sobre o efeito de interação entre fatores é não significativo e os genes em que a conclusão é significativo. Para aqueles genes em que a conclusão é não significativo, pode-se realizar a normalização (que visa a identificação de DEGs) utilizando o método proposto em TCC e, em seguida, realizar o teste de identificação de DEGs utilizando alguma metodologia que queira. Porém, para aqueles genes em que a conclusão sobre a interação é significativo, ao analisar um fator em cada nível dos outros fatores, a utilização do `TCC` para a normalização anterior à análise de identificação de DEGs, não é aconselhada. Isso se deve ao fato de, ao selecionar apenas aqueles genes em que a conclusão sobre o efeito da interação foi significativo, descarta-se a maioria dos genes que são *non*-DEGs, selecionando apenas aqueles *non*-DEGs que foram falso positivo para interação. Sendo assim, quando se deseja fazer uma análise

apenas sobre os genes em que existe interação entre os fatores, existirá um grande desbalanceamento entre o número de DEGs e *non*-DEGs, de forma que o número de DEGs seja maior e, em nossas simulações, descobrimos que, quando acontece esta situação, o uso do `TCC` causa uma grande perda do desempenho nas análises. Esse problema foi reportado aos criadores do `TCC`, J. Sun e K. Kadota, e, após discutirmos sobre o assunto, chegamos à conclusão que, nessa situação, o uso do `TCC` provoca um pior desempenho.

Portanto, para a normalização (visando o teste de identificação de DEGs) dos dados referentes às análises de 5 a 12, não foi utilizado o pacote `TCC`, e sim a normalização implementado no pacote `DESeq`, conforme comandos mostrados a seguir. Para isso, primeiramente, deve-se definir a condição do experimento, que será indicado pelo objeto `condition`.

Quando experimento com 2 fatores (experimentos 5 e 6 ou análises de 5 a 8):

```
> condition<-gl(2,2)              #Pois o fator A tem dois niveis e duas
                                  #  repeticoes.
```

Quando experimento com 3 fatores (experimento 7 ou análises de 9 a 12):

```
> condition<-gl(2,1)              #Pois o fator A tem dois niveis e
                                  #  uma repeticao.
```

Após, segue-se com a normalização, utilizando o método de normalização proposto em DESeq (função `estimateSizeFactors()`). Neste caso, não foi utilizada a normalização proposta em TCC, pois, como deseja-se realizar o teste de identificação de DEGs em apenas naqueles genes em que o efeito de interação entre fatores é significativo, o uso do `TCC` é não apropriado.

```
> cds<-newCountDataSet(dados1,condition)       #Objeto de entrado do DESeq.
> cds<-estimateSizeFactors(cds)                #Normalizacao usando o DESeq.
```

Após realizada a normalização, deve-se calcular a estimativa da dispersão dos dados. Para isso, é usada a função `estimateDispersions()`, com argumentos diferentes quando existe e quando não existe repetição.

Quando experimento com 2 fatores (experimentos 5 e 6 ou análises de 5 a 8):

```
> cds<-estimateDispersions(cds)                #Existe repeticao.
```

Quando não existe repetição, o que acontece no experimento com 3 fatores (experimento 7 ou análises de 9 a 12), deve-se usar os argumentos conforme indicados abaixo:

```
> cds<-estimateDispersions(cds, method='blind',
                           sharingMode='fit-only', fitType="local")
```

Para realizar o teste de identificação de DEGs, ou seja, testar o efeito do fator *A* na expressão gênica, em cada nível dos outros fatores que entraram nas análises, o procedimento é similar quando apenas um fator. Os comandos que foram utilizados são mostrados a seguir:

```
> fit1<-fitNbinomGLMs(cds,count~condition)
> fit0<-fitNbinomGLMs(cds,count~1)
> pvalsGLM<-nbinomGLMTest(fit1,fit0)
> pvalsGLM[is.na(pvalsGLM)]<-1
> padjGLM<-p.adjust(pvalsGLM,method="BH")
> p<-rep(1,1000)              #Os valores que serao utilizados como valores
                              #  de positividade, para fazer a Curva ROC,
                              #  serao os valores do objeto p, e não do
                              #  objeto padjGLM. Pois deve-se mensurar o
                              #  desempenho da analise sobre todos os 1000
                              #  genes simulados inicialmente, e nao apenas
                              #  sobre aqueles genes em que a interacao foi
                              #  concluida como significativa. Para aqueles
                              #  genes falso negativo para interacao
                              #  significativa, seu p-valor para o teste
                              #  de identificacao de DEGs serah 1.
> p[linhas]<-padjGLM
```

4.2.4 Calculando o desempenho das análises

Para mensurar o desempenho, foi calculado o valor de AUC de cada análise. Os comandos foram os mesmos citados no tópico (4.1.4).

Toda metodologia descrita no tópico (4.2) foi realizada em um *loop* de 100 vezes, formando assim 100 valores de AUC para cada uma das análises de 1 a 12. Esses valores foram representados por uma matriz de ordem 100x12, em que cada coluna representa uma análise, e cada linha representa um *loop*. Com isso, foi possível

comparar os experimentos de acordo com a distribuição de seus valores calculados para AUC.

4.3 Analisando os valores de AUCs quando interação não significativa

Após a realização de toda metodologia, foi obtido uma matriz de ordem 100 x 7 dos valores de AUC, para quando os efeitos da interação entre fatores são não significativos, em que, as colunas representam os experimentos de 1 a 7, e cada linha representa um dos 100 *loops* realizados. Com isso, obteve-se uma amostra com 100 observações dos valores de AUC para cada experimento.

Com o intuito de realizar um teste de médias para comparar o desempenho das análises em cada experimento, primeiramente deve-se testar se existe homocedasticidade entre as variâncias dos dados de cada tratamento (experimentos de 1 a 7). Para a escolha certa do teste de homocedasticidade, foi realizado, anteriormente, o teste de normalidade de Lilliefors (Lilliefors, 1967) para testar se os valores de AUC, em cada tratamento, seguem distribuição normal. Para isso, foi utilizada a função `lillie.test()`, do pacote `nortest` (Gross & Ligges, 2015) para o R. Com isso, o teste de homocedasticidade escolhido foi o teste de Levene (Levene, 1960).

Para realizar o teste de Levene para homocedasticidade, foi utilizado a função `leveneTest()`, do pacote `car` (Fox & Weisberg, 2011) para o R. Além disso, foi utilizado o teste de normalidade de Lilliefors para testar a normalidade do erro experimental do modelo

$$y_{ij} = m + t_i + e_{ij}$$

em que,

y_{ij} é o valor observado da AUC obtida para o *i*-ésimo (i = 1, 2, …, 7) tratamento, em sua *j*-ésima (j = 1, 2, ..., 100) repetição;

m é a média de todos os valores possíveis de y_{ij} ;

t_i é o efeito do tratamento *i* no valor observado y_{ij} ; e

e_{ij} é o erro experimental associado ao valor observado y_{ij} .

Para comparar o desempenho das análises em cada experimento, optou-se em utilizar a alternativa não-paramétrica através do uso do teste de Kruskal-Wallis para concluir sobre as medianas, testando a hipótese

$$H_0:\ Md_1 = Md_2 = \ldots = Md_7\ ,$$

em que, $Md_1, Md_2, \ldots, Md_7$ são as medianas dos valores de AUC dos tratamentos de 1 a 7, respectivamente, contra a hipótese

$$H_a:\ Md_i \neq Md_j\ ;\ \forall\ i \neq j\ ,\ i, j = 1,2,\ldots,7$$

(Triola, 2013). O teste foi realizado através do uso da função `kruskal.test()` do R.

Sendo assim, deve-se agora identificar quais foram os experimentos que apresentaram um maior desempenho de suas análises e quais os de menor desempenho. Para isso, realizou-se o teste de Kruskal-Wallis para comparações múltiplas (Hollander & Wolfe, 1973), utilizando a função `kruskal()`, do pacote `agricolae` (Mendiburu, 2014) para o R, a $\alpha = 0{,}05$ de nível de significância e ajustamento para testes múltiplos dos *p-valores* através do método de Bonferroni (Bonferroni, 1936).

4.3.1 Analisando os valores de AUC de experimentos com o mesmo número de fatores, como se fossem de uma mesma amostra

Com o intuito de comparar experimentos com diferentes números de fatores (e repetições) em geral, considerou-se os seguintes tratamentos:

- Tratamento 1: como os valores de AUC de todos os experimentos com 1 fator e 4 repetições (experimentos de 1 a 4), portanto, para o tratamento 1, o tamanho da amostra é de 400 observações;
- Tratamento 2: como os valores de AUC de todos os experimentos com 2 fator e 2 repetições (experimentos 5 e 6), portanto, para o tratamento 2, o tamanho da amostra é de 200 observações; e
- Tratamento 3: como os valores de AUC do experimento com 3 fator e 1 repetições (experimento 7), portanto, para o tratamento 3, o tamanho da amostra

é de 100 observações.

Para comparar os tratamentos 1, 2 e 3, ou seja, experimentos quando 1, 2 e 3 fatores, respectivamente, foi utilizado o teste de Kruskal-Wallis.

4.4 Analisando os valores de AUCs quando interação significativa

Em seguida, o mesmo foi feito para quando os efeitos de interação entre os fatores são significativos. Após a finalização de toda metodologia, também foi criada uma matriz dos valores de AUC quando interação significativa. Essa matriz é de ordem 12 x 100, em que, cada coluna representa as análises de 1 a 12, respectivamente, de acordo com a Tabela 3 do tópico (4.2.1), e cada linha representa um dos 100 *loops* realizados.

Para comparar o desempenho de cada análise de 1 a 12, primeiramente realizou-se o teste de normalidade de Lilliefors em cada uma delas. Em seguido, optou-se em realizar o teste de homocedasticidade das variâncias entre as 12 análises através do teste de Levene. Além disso, também foi realizado o teste de Lilliefors para verificar normalidade do erro experimental. Finalmente, optou-se em comparar o desempenho de cada análise de 1 a 12 através da alternativa não paramétrica, com o uso do teste de Kruskal-Wallis.

Sendo assim, deve-se então identificar quais foram as análises que apresentaram um maior desempenho e quais apresentaram um menor desempenho. Para isso realizou-se o teste de Kruskal-Wallis para comparações múltiplas.

5 Resultados e Discussões

5.1 Quando interação não significativa

A Figura 8 mostra as distribuições, através de histogramas, dos valores amostrados da AUC para cada um dos experimentos de 1 a 7, quando os efeitos de interação entre fatores são não significativos (matriz de ordem 7x100 dos valores de AUCs). Os *p-valores* retornados pelo teste de Lilliefors, para testar normalidade dos valores de AUCs

de cada experimento, são mostrados na Tabela 6. Como foi concluído que não acontece a distribuição normal dos valores de AUC em todos os tratamentos (experimentos de 1 a 7), devido ao experimento 4 apresentar um *p-valor* considerado pequeno, o uso do teste de Bartlett (Bartlett, 1937) para homocedasticidade não é aconselhável, pois apresenta-se sensível quanto a violação da pressuposição de normalidade e, sendo assim, optou-se em realizar o teste de Levene, devido a sua robustez quanto à não normalidade dos dados (Riboldi, 2014).

O *p-valor* retornado pelo teste de Levene foi de 2,2e-16, indicando a não homogeneidade das variâncias entre os tratamentos. Além disso, o teste de normalidade de Lilliefors para testar a normalidade do erro experimental, e_{ij} , retornou o *p-valor* igual a 2,2e-16, indicando distribuição não normal. Portanto, não se deve realizar o teste paramétrica ANOVA, devido a essas duas violações de suas pressuposições (Ribeiro Júnior, 2012), e sim, o teste não-paramétrico Kruskal-Wallis, para concluir sobre as medianas dos tratamentos. O *p-valor* retornado pelo teste de Kruskal-Wallis foi de 2,2e-16, o que nos permite concluir em rejeitar a hipótese de que todas as medianas dos tratamentos são iguais, ou seja, existe pelo menos um tratamento que possui sua mediana diferente das demais. Portanto, pode-se concluir qual experimento obteve um melhor e qual obteve um menor desempenho, baseando-se em suas medianas. Essas conclusões são mostradas na Tabela 7, em que, aqueles tratamentos que receberam uma mesma letra são considerados iguais, baseando-se em suas medianas, e aqueles que receberam letras diferentes são considerados diferentes. Além disso, uma boa visualização dessas igualdades e diferenças entre as medianas dos tratamentos (experimentos de 1 a 7), pode ser obtida pela Figura 9, onde é feito um *boxplot* dos valores de AUC de cada experimento.

Observando a Tabela 7 ou a Figura 9, pode-se notar que o desempenho das análises de experimentos que possuem os mesmos números de fatores e repetições é extremamente influenciado pela escolha daqueles fatores que não entraram nas análises. Por exemplo, o experimento 1 obteve o melhor desempenho, baseando-se em sua mediana, e o experimento 4 apresentou-se ser o de menor desempenho, mesmo que ambos sejam experimentos com o mesmo número de fatores (1 fator) e de repetições (4 repetições). Essa discrepância entre eles ocorre pois, ao observar como foi gerado seus respectivos dados pela simulação, pode-se notar que as únicas diferenças entre eles são seus valores de *fold change* para cada uma de suas amostras (conforme Figura 5, do tópico (4.1.1)). O experimento 1 apresenta dados com grandes diferenças entre os dois

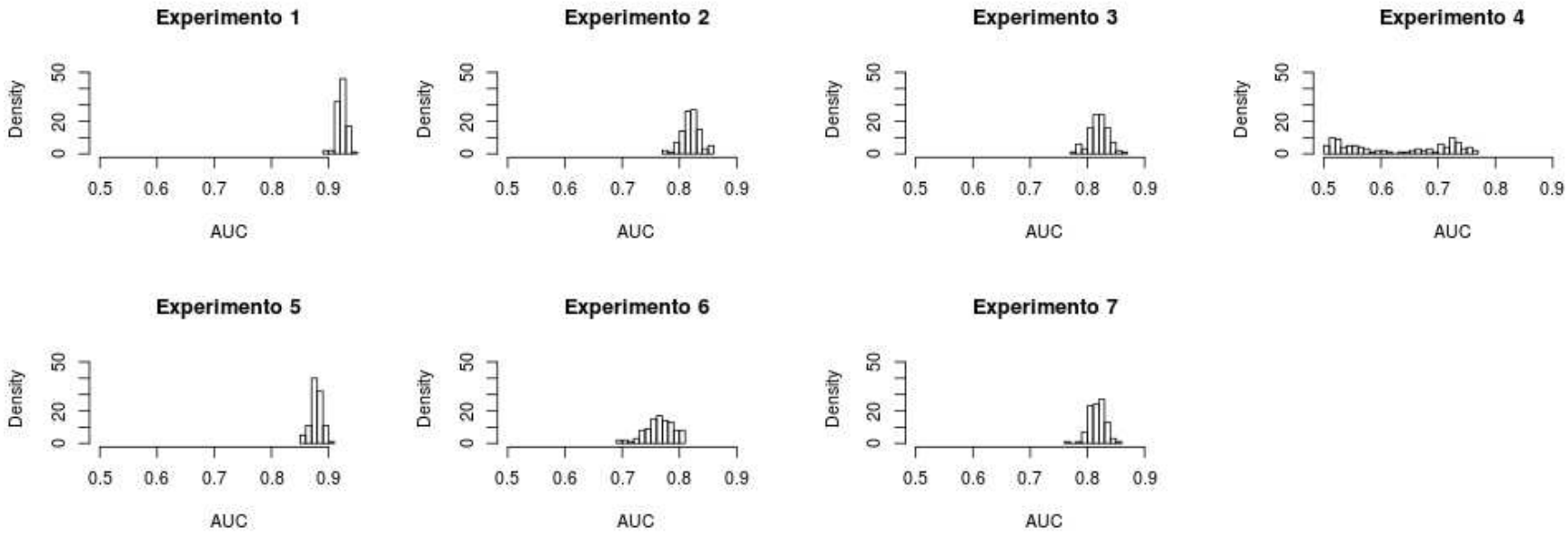

Figura 8: Histogramas das 100 observações de AUC para cada um dos experimentos de 1 a 7, quando os efeitos de interação são não significativos.

Tabela 6: *p-valores* para o teste de normalidade de Lilliefors em cada um dos experimentos de 1 a 7, quando os efeitos de interação entre fatores são não significativos.

Experimento	*p-valor*	Conclusão*
1	0,8219	não rejeita H_0
2	0,9779	não rejeita H_0
3	0,9248	não rejeita H_0
4	**3,3e-7**	**rejeita H_0**
5	0,3049	não rejeita H_0
6	0,5178	não rejeita H_0
7	0,8492	não rejeita H_0

*Conclusão feita sobre a hipótese H_0: dados segue distribuição normal , com nível de significância α = 0,05.

Tabela 7: Conclusões obtidos pelo teste de Kruskal-Wallis para comparações múltiplas.

Experimento	Médias dos postos*	
1	650,40	A
5	550,37	B
3	354,74	C
2	353,17	C
7	336,40	C
6	153,33	D
4	55,09	E

*Médias obtidas dos postos para cada experimento (tratamento), quando todos os valores de AUC de todos experimentos são ordenados juntos.
Experimentos/tratamentos que receberam uma mesma letra, são considerados iguais, baseando-se na mediana. Os que receberam letras diferentes, são considerados diferentes. Diferentes cores representam experimentos com diferentes número de fatores: vermelho ≡ 1 fator; azul ≡ 2 fatores; verde ≡ 3 fatores.

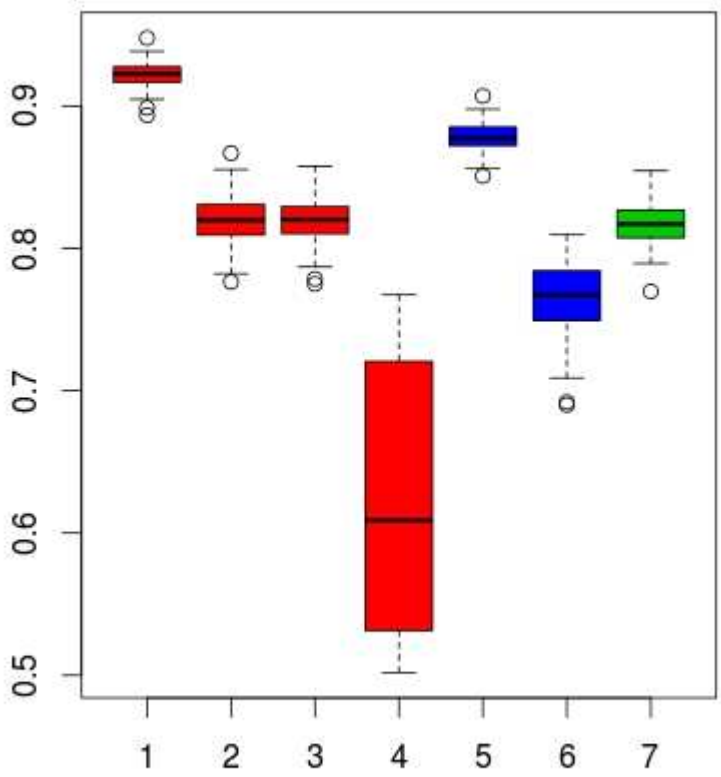

Figura 9: *Boxplot* dos valores de AUC de cada experimento (de 1 a 7) quando os efeitos de interação são não significativos. Diferentes cores representam experimentos com diferentes número de fatores: vermelho ≡ 1 fator; azul ≡ 2 fatores; verde ≡ 3 fatores.

níveis do fator em estudo (o efeito do nível *a2* do fator em estudo *A*, confere um valor de contagem dos *reads* (valor de expressão gênica digital), $4 \div 1 = 4$ vezes maior do que o efeito do nível *a1*), o que simula genes que estão expressando de forma altamente diferencial entre eles. Sendo assim, o método consegue facilmente identificar os genes que são DEGs e, consequentemente, a análise apresenta alto desempenho. Já no experimento 4, as diferenças entre os níveis do fator em estudo é pequena (o efeito do nível *a2* do fator em estudo *A*, confere um valor de contagem dos *reads*, $10 \div 7 \approx 1{,}43$ vezes maior do que o efeito do nível *a1*), o que simula os dados de um experimento desenvolvido para testar o mesmo do que no experimento 1, porém, os níveis dos fatores que não entraram nas análises são diferentes, e estes fazem com que os valores

de contagem em todas as observações tendem a serem maiores, fazendo com que a razão entre os níveis *a2* e *a1* seja menor. Com isso, a identificação dos genes que são DEGs, pelo método, é difícil, apresentando um baixo desempenho das análises.

Esse problema é muito comum de acontecer, pois pesquisadores em todo mundo podem não seguir o mesmo protocolo para o cultivo dos espécimes a serem estudados. Principalmente, tendo em vista estudos realizados sobre organismos (*e.g.*, bactérias) que podem viver sob condições diversas e, com isso, a escolha dos níveis daqueles fatores que não iram entrar nas análises (*e.g.*, temperatura, pressão, pH, etc.) é ampla e, não só pode, mas deverá influenciar nas conclusões sobre os fatores em estudo. Os experimentos 1 e 4 (assim como os experimentos 5 e 6) simulam o caso em que dois pesquisadores estão realizando as mesmas análises, mas estão seguindo protocolos diferentes.

5.1.1 Considerando os valores de AUC de experimentos com o mesmo número de fatores, como se fossem de uma mesma amostra

Seja as amostras de um tratamento todos os valores de AUCs de experimentos com o mesmo número de fatores e, por consequência, mesmo número de repetições, ou seja: tratamento 1, experimentos com 1 fator; tratamento 2, experimentos com 2 fatores; e tratamento 3, experimentos com 3 fatores. A Figura 10 mostra os histogramas dos valores de AUC observados em cada um desses tratamento.

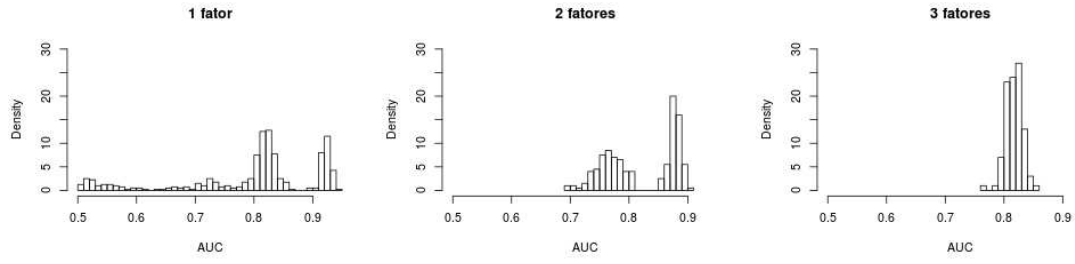

Figura 10: Histogramas para todos os valores de AUC quando experimentos com 1 fator (experimentos de 1 a 4), 2 fatores (experimentos 5 e 6), e 3 fatores (experimento 7).

O teste de Kruskal-Wallis, para testar a hipótese de que esses três tratamentos

possuem o mesmo valor para suas medianas, retornou o *p-valor* igual a 0,7503, o qual pode ser considerado alto e concluir que as medianas de todos os três tratamentos são iguais. Foi escolhido o teste de Kruskal-Wallis para comparar os tratamentos pois, visivelmente (Figura 9), os dados dos tratamentos não são normalmente distribuídos (salvo o tratamento 3), suas variâncias não são homogêneas, e o erro experimental não segue distribuição normal. Confirmando isso, o teste de Lilliefors retornou os *p-valores* 2,2e-16 , 2,2e-16 , e 0,8492 , para os respectivos tratamentos de 1 a 3; o teste de Levene retornou o *p-valor* 2,2e-16 ; e o *p-valor* do teste de Lilliefors para o erro experimental foi de 2,2e-16.

Sendo assim, pode-se dizer que, ao incluir mais fatores à análise, e diminuir o número de repetições, mas de forma que o número de amostras continue constante, o desempenho da análise não muda (baseando-se na mediana dos valores de AUC). Isso indica que o uso de experimentos fatoriais, quando a interação entre os fatores é não significativa, para a análise de dados de RNA-Seq, é viável. Pois, assim, pode-se concluir para mais fatores, sem perder desempenho das análises.

Além disso, e mais importante, o teste de Levene diz que não existe homocedasticidade entre as variâncias, tanto no caso anterior, onde os experimentos foram analisados separadamente (como 7 tratamentos distintos), quando no caso em que os valores de AUC de experimentos com o mesmo número de fatores foram considerados como pertencentes a uma mesma amostra (3 tratamentos distintos). E, para avaliar se uma análise é boa, não se deve simplesmente considerar se a ela é atribuído um valor de desempenho que tende a ser alto, mas também, o quanto esse valor pode variar. O teste não paramétrico nos diz qual tratamento é o melhor, mas apenas baseando-se no quão alto seus valores tendem a ser, baseando-se na mediana, não levando em conta a variabilidade desses valores em cada tratamento. No entanto, é muito lógico dizer que, é pior uma análise que pode apresentar seu desempenho (valores de AUC) muito variado. A variância calculada para os valores amostrados de AUC quando 1, 2 e 3 fatores são, respectivamente, iguais a 0,0138 , 0,0036 , e 0,0002. Portanto, a variância dos valores de AUC quando 1 fator é aproximadamente 70 vezes maior quando 3 fatores. Isso indica fortemente a vantagem de experimentos fatoriais.

5.1.2 Outros importantes resultados

Uma outra discussão apropriada é sobre o poder do teste, ou *taxa de verdadeiro positivo* (TPR), e a taxa de ocorrência do erro tipo 1, ou *taxa de falso positivo* (FPR), obtidos em cada uma das análises dos experimentos de 1 a 7. A Figura 11 e a Figura 12 mostram a distribuição dos valores de TPR e FPR, respectivamente, através de histogramas, para cada experimento, quando ajustamento dos *p-valores* para testes múltiplos através do método de Benjamini-Hochberg, e uso de $\alpha = 0{,}1$ de nível de significância. Esse valor para o nível de significância foi utilizado por ser um valor padrão, utilizado em vários outros trabalhos, tais como Sun *et al* (2015) e Anders & Huber (2013).

Ao analisar a Figura 11, pode-se perceber que os valores de TPR foram extremamente baixos, apenas nos experimentos 1 e 5 esses valores se apresentaram maiores. Isso se deve pois, ao calcular a média dos valores de *fold change* entre as amostras que receberam o nível *a1* do fator *A* (fator em estudo), e a média dos valores de *fold change* das que receberam o nível *a2*, pode-se perceber que: para o experimento 1, a média dos valores de *fold change* das amostras que receberam o nível *a2* é 4 (pois todas essas amostras receberam *fold change* igual a 4), e a média dos valores de *fold change* das amostras que receberam o nível *a1* é 1 (pois em todas elas é 1), portanto, as amostras que receberam o nível *a2* tendem a ter seus valores de contagem $4 \div 1 = 4$ vezes maior que os valores de contagem das amostras que receberam o nível *a1*; para o experimento 5, as amostras que receberam o nível *a2* tendem a ser 2,20 vezes maior que as que receberam o nível *a1*; para os experimentos 2, 3 e 7, em *a2* tende a ser 1,75 vezes maior que em *a1*; para o experimento 6, em *a2* tende a ser 1,56 vezes maior que em *a1*; e para o experimento 4, em *a2* tende a ser 1,43 vezes maior que em *a1*. Portanto, pode-se observar que, a metodologia DESeq não consegue identificar as pequenas diferenças entre as amostras que receberam níveis diferentes do fator em estudo. Se a diferença for menor do que duas vezes maior (aproximadamente) em um nível em relação ao outro, a TPR é muito pequena. Ou seja, para que a metodologia consiga identificar as diferenças, com maior poder, as amostras que receberam um determinado nível do fator em estudo, devem apresentar-se, no mínimo, 2 vezes (aproximadamente) maiores do que aquelas que receberam o outo nível.

Ao analisar a Figura 12, também pode-se perceber que os valores de FPR foram muito baixos para todos os experimentos, o que indica, juntamente com o fato dos valores de TPR serem baixos, o quanto o teste é conservador ao nível de significância de $\alpha = 0{,}1$.

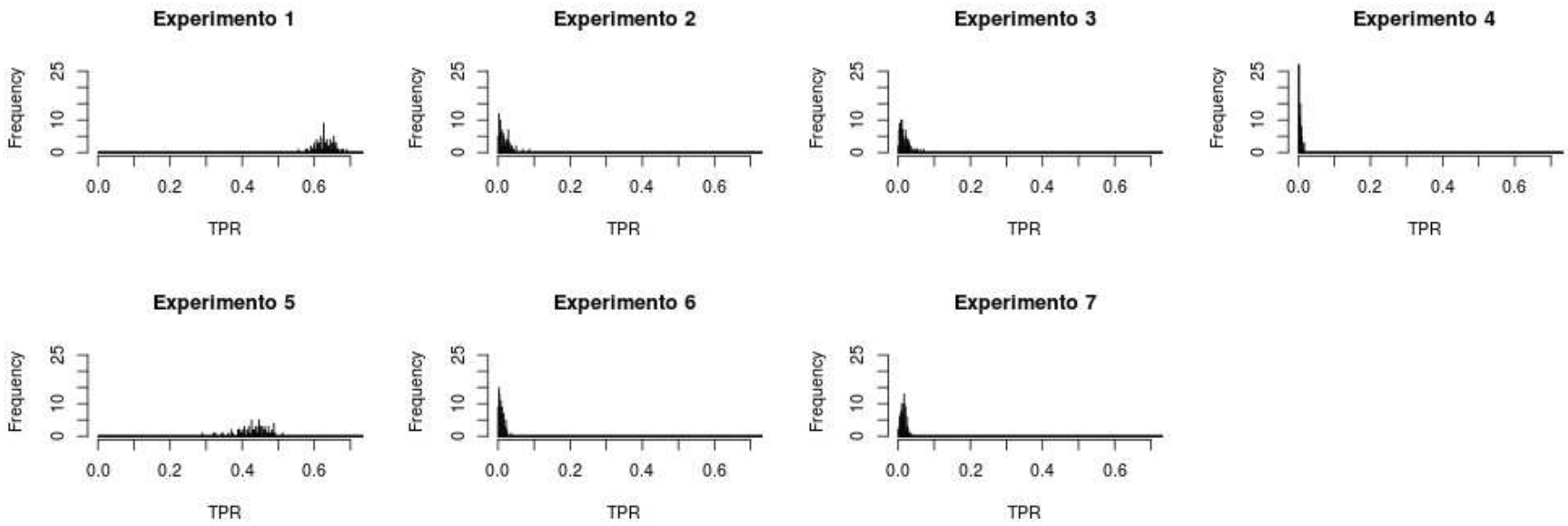

Figura 11: Histogramas dos valores de taxa de verdadeiro positivo (TPR) para cada um dos experimentos de 1 a 7, quando os efeitos de interação entre os fatores são não significativos.

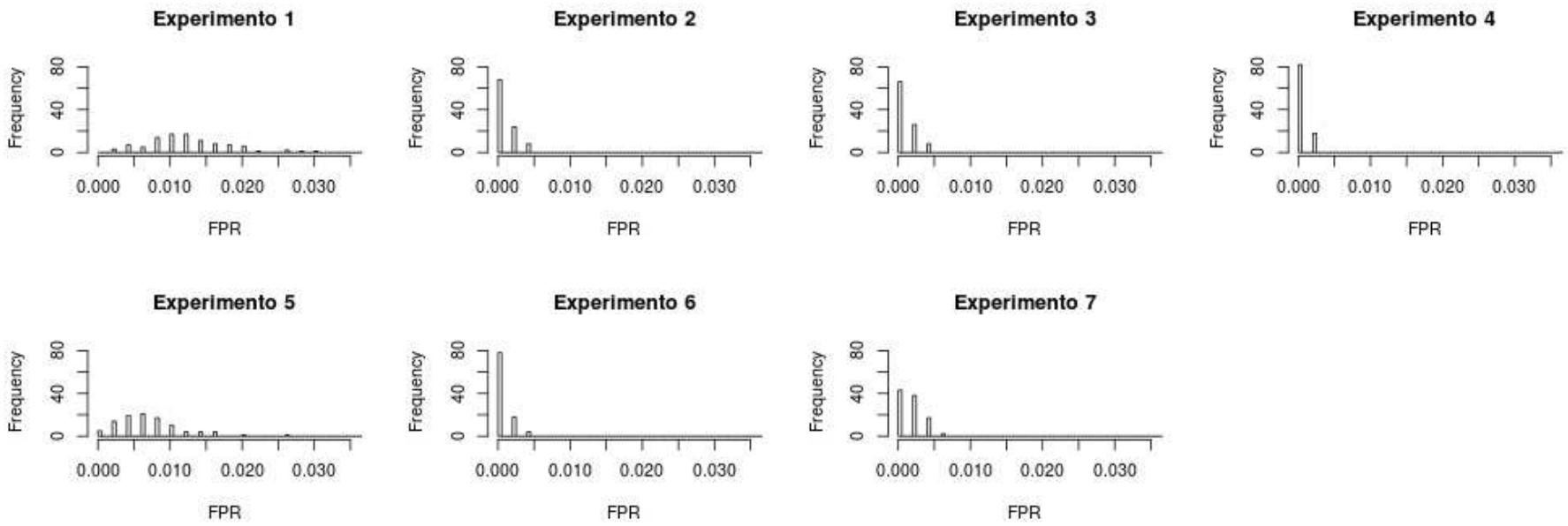

Figura 12: Histogramas dos valores de taxa de falso positivo (FPR) para cada um dos experimentos de 1 a 7, quando os efeitos de interação entre os fatores são não significativos.

5.2 Quando interação significativa

A Figura 13 mostra a distribuição, através de histogramas, dos valores amostrados da AUC para cada uma das análises de 1 a 12, de acordo com a Tabela 3 do tópico (4.2.1), quando os efeitos de interação entre os fatores são significativos (matriz é de ordem 12 x 100 dos valores de AUCs). Os *p-valores* retornados pelo teste de Lilliefors para verificar normalidade dos valores de AUCs, em cada das 12 análises, é mostrado na Tabela 8. Portanto, pode-se concluir que existem tratamentos (análises de 1 a 12) que seus valores de AUC não seguem distribuição normal. Devido a isso, optou-se realizar o teste de homocedasticidade das variâncias entre as 12 análises através do teste de Levene, e este retornou *p-valor* igual a 2,2e-16, através do qual pode-se concluir que as variâncias dos valores de AUC em cada análise não são homogêneas. Além disso, o teste de Lilliefors para verificar normalidade do erro experimental retornou o *p-valor* igual a 0,0469, valor esse que considerado baixo e, portanto, assumiu-se distribuição não normal para o erro experimental. Portanto, devido a violações de pressuposições da ANOVA, optou-se em comparar o desempenho de cada análise de 1 a 12 através da alternativa não-paramétrica, com o uso do teste de Kruskal-Wallis. O *p-valor* retornado pelo teste não paramétrico foi de 2,2e-16, nos permitindo rejeitar

$$H_0:\ Md_1 = Md_2 = \ldots = Md_{12}\ ,$$

em que, $Md_1, Md_2, \ldots, Md_{12}$ são as medianas dos valores de AUC das análises de 1 a 12, respectivamente, e concluir que existe pelo menos uma análise que tende a obter maiores valores de AUC, baseando-se na mediana.

Sendo assim, para identificar quais foram as análises que apresentaram um maior e quais apresentaram um menor desempenho, foi realizado o teste de Kruskal-Wallis para comparações múltiplas, e essas conclusões obtidas são mostradas na Tabela 9. Em que, aquelas análises que receberam uma mesma letra são considerados iguais, baseando-se em suas medianas, e aqueles que receberam letras diferentes são considerados diferentes. Além disso, uma boa visualização dessas igualdades e diferenças entre suas medianas, pode ser obtida pela Figura 14, onde é feito um *boxplot* dos valores de AUC de cada análise.

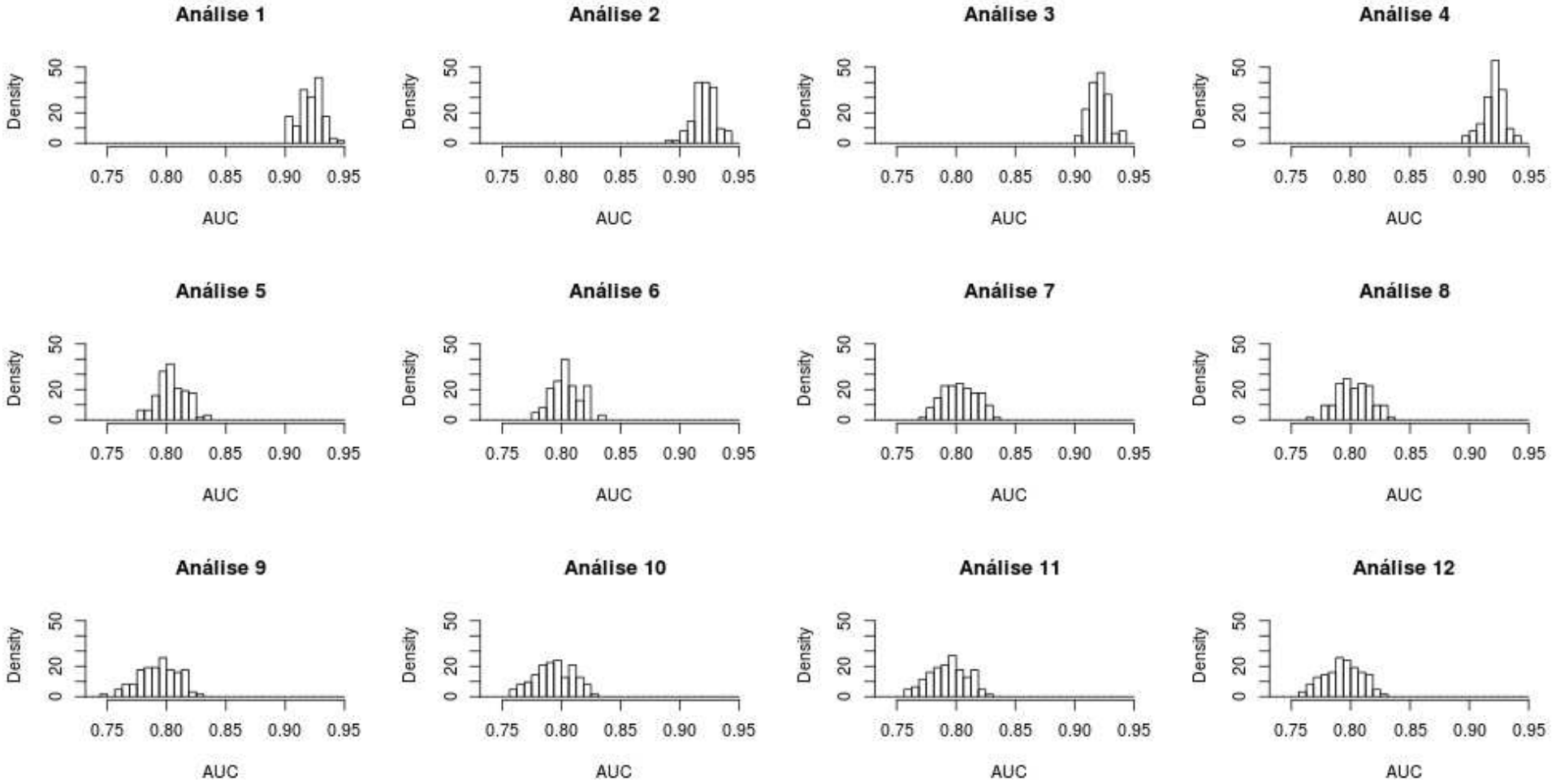

Figura 13: Histogramas das 100 observações de AUC para cada uma das análises de 1 a 12, quando os efeitos de interação são significativos.

Tabela 8: *p-valores* para o teste de normalidade de Lilliefors em cada uma das análises de 1 a 12, quando os efeitos de interação entre fatores são significativos.

Análise	*p-valor*	Conclusão*
1	0,3202	não rejeita H_0
2	0,9025	não rejeita H_0
3	0,4167	não rejeita H_0
4	**0,0379**	**rejeita H_0**
5	**0,0288**	**rejeita H_0**
6	**0,0059**	**rejeita H_0**
7	0,3306	não rejeita H_0
8	0,7768	não rejeita H_0
9	0,5285	não rejeita H_0
10	0,4402	não rejeita H_0
11	0,2686	não rejeita H_0
12	0,8232	não rejeita H_0

*Conclusão feita sobre a hipótese H_0: dados segue distribuição normal , com nível de significância α = 0,05.

Tabela 9: Conclusões obtidos pelo teste de Kruskal-Wallis para comparações múltiplas.

Análise	Médias dos postos*			
1	1009,14	A		
4	1001,04	A		
2	995,98	A		
3	995,84	A		
6	484,11		B	
5	483,46		B	
7	462,83		B	
8	461,76		B	
10	328,88			C
11	328,48			C
12	327,92			C
9	326,55			C

*Médias obtidas dos postos para cada análise, quando todos os valores de AUC de todas análises são ordenados juntos.
Análises que receberam uma mesma letra, são considerados iguais, baseando-se na mediana. Os que receberam letras diferentes, são considerados diferentes. Diferentes cores representam experimentos com diferentes número de fatores: vermelho ≡ 1 fator; azul ≡ 2 fatores; verde ≡ 3 fatores.

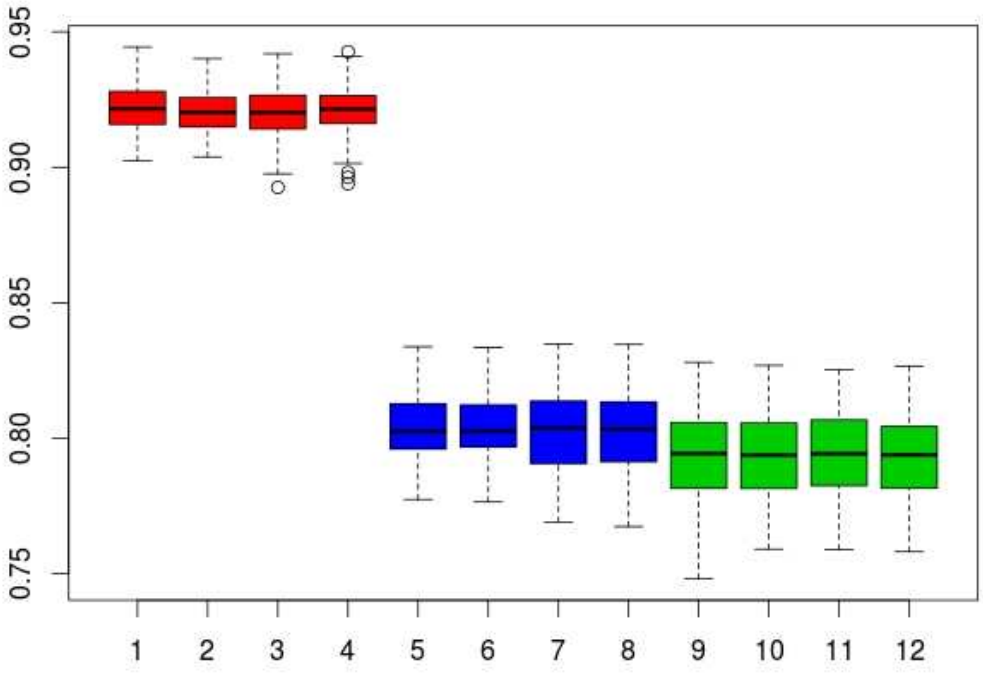

Figura 14: *Boxplot* dos valores de AUC de cada análise (de 1 a 12) quando os efeitos de interação são significativos. Diferentes cores representam experimentos com diferentes número de fatores: vermelho ≡ 1 fator; azul ≡ 2 fatores; verde ≡ 3 fatores.

Analisando a Tabela 9 ou a Figura 14, pode-se perceber que, ao aumentar o número de fatores e diminuir o número de repetições, mas de forma que o número de amostras permaneça constante, o desempenho da análise vai diminuindo. Isso se deve pois, ao olhar como foram criados os dados, percebe-se que a única diferença nas simulações entre os dados das análises que envolvem 1 fator (análise 1 a 4), 2 fatores (análise 5 a 8), e 3 fatores (análise 9 a 12), é o fato do número de repetições ir diminuindo, respectivamente (veja Figura 7, do tópico (4.2.3.2)). Além disso, quando o experimento é fatorial (caso das análises de 5 a 12), deve-se, primeiramente, testar se os efeitos de interação entre fatores são significativos, e como o método é conservador, ocorrem muitos falsos negativos para a interação, e, no algoritmo desenvolvido por este trabalho, quando não se rejeita a hipótese de que o efeito da interação é não significativo, para o caso de dados simulados com interação significativa, é atribuído para o referente gene o *p-valor* igual a 1 para o teste de identificação de DEGs. Isso faz com que muitos genes

que são DEGs, fiquem ranqueados juntos com genes que são *non*-DEGs. Diferentemente do caso de experimentos que não são fatoriais, pois nestes é realizado apenas o teste de identificação de DEGs, sem a necessidade de antes testar a interação. Isso explica o porquê das análises de 5 a 12 apresentarem valores de AUC tão menores que as análises de 1 a 4.

Porém, embora os experimentos com menos fatores apresentam um melhor desempenho em suas análises, pois possuem mais repetições, pode-se considerar que a escolha do experimento fatorial seja a apropriada, pois, se existe interação entre os fatores, não se pode analisá-los separadamente. Quando se conclui sobre os fatores de uma análise, essa conclusão é limitada apenas quando naquela condição determinado pelo experimento. Qualquer alteração realizada nos outros fatores que não entraram nas análises, pode causar conclusões completamente contrastantes a respeito dos fatores que entraram nas análises. Pois o efeito destes podem depender daqueles (interação significativa). Por exemplo, um pesquisador que realizou a análise 1, pode concluir que o nível de expressão gênica aumenta quando se passa do nível *a1* para o nível *a2* do fator *A*, porém, um outro pesquisador que realizou a análise 2, pode concluir, sobre o mesmo fator *A*, que o nível de expressão gênica diminui quando se passa do nível *a1* para o nível *a2*. Isso se deve pois, o nível dos fatores que não entraram nas análises são diferentes para os dois casos (na análise 1, o nível do fator *B* é *b1* e o do *C* é *c1*; e na análise 2 o nível do fator *B* é *b2* e o do *C* é *c1*).

No entanto, se o objetivo da pesquisa for apenas identificar quais os genes que são DEGs, sem se preocupar se a expressão está aumentando (*up-regulated*) ou se está diminuindo (*down-regulated*), é preferível planejar um experimento com apenas um fator, pois assim, tem-se mais repetições.

5.2.1 Outros importantes resultados

Para avaliar os valores de TPR e FPR em cada análise, quando o efeito da interação entre os fatores são significativos, a Figura 15 e a Figura 16 mostram, respectivamente, as distribuições desses valores calculados, quando ajustamento dos *p-valores* para testes múltiplos através do método de Benjamini-Hochberg, e uso de $\alpha = 0{,}1$ de nível de significância.

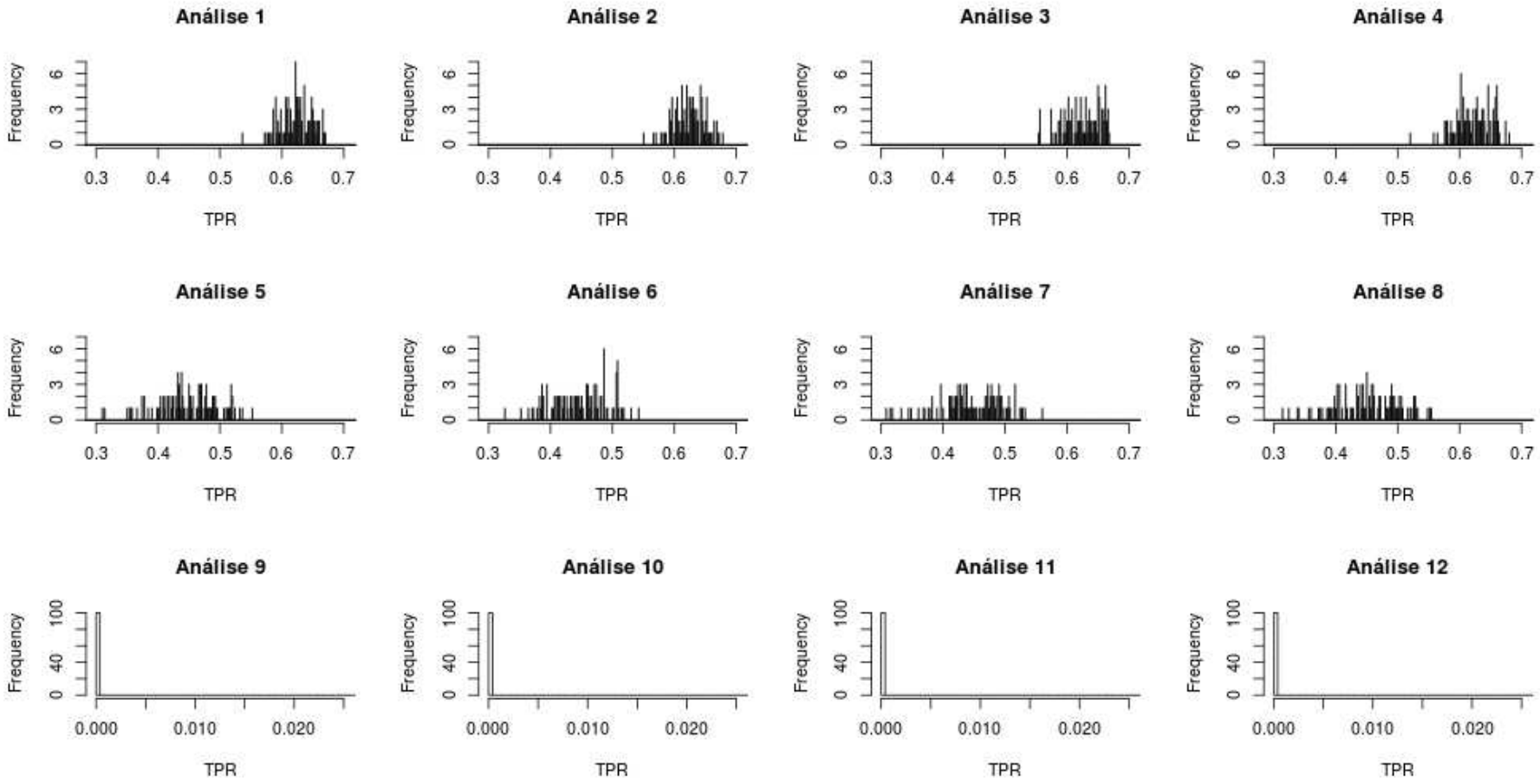

Figura 15: Histogramas dos valores de taxa de verdadeiro positivo (TPR) para cada uma das análises de 1 a 12, quando os efeitos de interação entre os fatores são significativos.

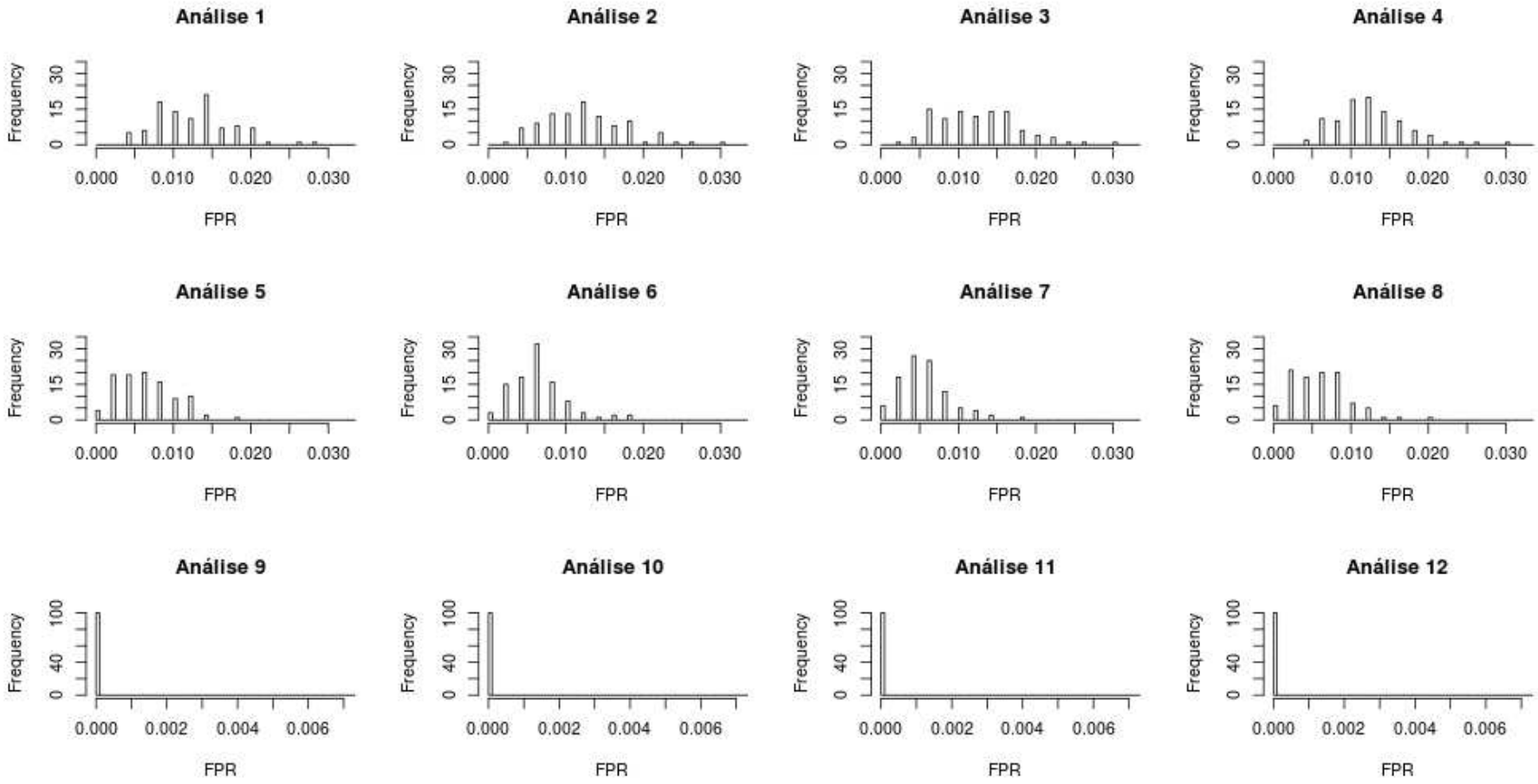

Figura 16: Histogramas dos valores de taxa de falso positivo (FPR) para cada uma das análises de 1 a 12, quando os efeitos de interação entre os fatores são significativos.

Na Figura 15 pode-se perceber que os valores de TPR são menores naquelas análises com mais fatores. Isso se deve ao fato das análises com mais fatores terem menos repetições. Uma vez que, uma das maneiras de aumentar o poder de um teste, é o uso de mais repetições (Auer & Doerge, 2010). Também é possível perceber uma grande queda dos valores de TPR das análises com 2 fatores em relação às com apenas 1 fator. Além do motivo mencionada anteriormente, também pode ser atribuir a isso, o fato de, quando um experimento fatorial e o efeito da interação entre fatores é significativo, também deve-se testar essa interação. Portanto, quando diante de um gene que é DEG, para que ocorra um verdadeiro positivo, não basta apenas rejeitar H_0 quando é realizado o teste de identificação de DEGs, antes disso, deve-se também rejeitar H_0 quando é realizado o teste para significância da interação entre os fatores. Já uma possível explicação para a acentuada queda dos valores de TPR para as análises que envolvem 3 fatores, é o fato de não haver repetição. Para essa situação, a estimativa do parâmetro de dispersão, proposta pelo método DESeq, é menos precisa do que quando existe repetição e, apesar do teste mostrar-se ser razoavelmente capaz de ranquear os genes que são DEGs (razoáveis valores de AUC), seu desempenho em identificar DEGs foi insatisfatório.

Na Figura 16 pode-se perceber que, assim como no caso quando a interação é não significativa, os valores de FPR das análises foram pequenos. Mostrando novamente a metodologia ser conservadora, principalmente quando não existe repetição. Pois, para esta situação, o teste retorna apenas altos *p-valores*, que são aumentados mais ainda após seu ajustamento para testes múltiplos e, com isso, não consegue classificar um gene como DEG.

6 Conclusão

Quando os efeitos de interação entre fatores são não significativos, as análises de experimentos com diferentes números de fatores e repetições, mas com o mesmo número de amostras, mostraram-se ter o mesmo desempenho esperado, baseando-se na mediana de seus valores de AUC. Portanto, pode-se planejar experimentos com mais fatores, e assim concluir sobre mais fatores, sem o aumento do custo da pesquisa, e sem perder o desempenho das análises.

Além disso, como o desempenho da análise é influenciado pela escolha dos níveis dos fatores que não entraram nas análises, o desempenho quando experimentos com menos fatores, e caso interação não significativa, mostrou-se ser mais variado. Reforçando a preferência por experimentos com mais fatores.

Já no caso quando os efeitos de interação entre fatores são significativos, ao aumentar o número de fatores e diminuir o número de repetições, mas de forma que o número de amostras seja contante, o valor de AUC da análise tende a diminuir (pois menos repetições). Porém, ao planejar experimentos com menos fatores, corre-se o risco de obter conclusões erradas quanto ao efeito do fator em estudo, pois existe interação com outros fatores, e estes ficaram de fora das análises. De forma que, é completamente possível, por exemplo, concluir que o efeito de um nível de um fator causa uma maior expressão em um gene, e, no entanto, se outros níveis tivessem sido atribuídos para aqueles fatores que não entraram nas análises, o efeito do nível de tal fator em estudo causaria uma menor expressão.

Portanto, pode-se dizer que a escolha de um experimento fatorial ou apenas um fator, e o número de fatores analisados, dependerá do objetivo visado pela pesquisa. Caso o pesquisador queira apenas identificar quais os genes que estão expressando de forma diferencial (DEGs), sem se importar se o nível de expressão está sendo maior ou menor devido ao efeito de um fator de interesse, é preferível usar experimentos com apenas um fator, pois assim se pode ter mais repetições com um determinado custo estabelecido. Entretanto, caso o interesse seja em identificar como esse fator está influenciando a expressão, deve-se utilizar experimentos fatoriais, para que assim não se corra o risco de tomar conclusões generalizadas erradas.

7 Referências

Anders S, Huber W (2010). **Differential expression analysis for sequence count data**. *Genome Biology*. 11:R106.

Anders S, Huber W (2013). **Differential expression of RNA-Seq data at the gene level - the DESeq package**. Disíponvel em http://bioconductor.org/packages/release/bioc/vignettes/DESeq/inst/doc/DESeq.pdf. Acesso em 10/12/2014.

Auer PL, Doerge RW (2010). **Statistical Design and Analysis of RNA Sequencing Data**. *Genetics Society of America*. Jun; 185(2):405-16.

Bartlett MS (1937). **Properties of sufficiency and statistical tests**. *Proceedings of the Royal Statistical Society*, Series A 160, 268–282. 1937.

Benjamini Y, Hochberg Y (1995). **Controlling the false discovery rate: a practical and powerful approach to multiple testing**. *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 57, No., pp. 289-300.

Bonferroni CE (1936). **Teoria statistica delle classi e calcolo delle probabilità**. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze* 8, 3-62.

Di Y, Schafer DW, Cumbie JS, Chang JH (2011). **The NBP negative binomial model for assessing differential gene expression from RNA-Seq**. *Stat Appl Genet Mol Biol*, 10:art24.

Fawcett T (2006). **An introduction to ROC analysis**. *Pattern Recognition Letters*, 27, 861–874.

Filho DF (2009). **Estudo de expressão gênica em citros utilizando modelos lineares**. Dissertação de Mestrado. Escola Superior de Agricultura “Luiz de Queiroz”,

ESALQ/USP. Disponível em http://www.teses.usp.br/teses/disponiveis/11/11134/tde-16032010-111945/publico/Diogenes_Ferreira_Filho.pdf. Acesso em 21/01/2015.

Fox J, Weisberg S (2011). An R Companion to Applied Regression, Second Edition. Thousand Oaks CA: Sage. URL: http://socserv.socsci.mcmaster.ca/jfox/Books/Companion.

Gentleman RC, Carey VJ, Bates DM, Bolstad B, Dettling M, Dudoit S, Ellis B, Gautier L, Ge Y, Gentry J, Hornik K, Hothorn T, Huber W, Iacus S, Irizarry R, Leisch F, Li C, Maechler M, Rossini AJ, Sawitzki G, Smith C, Smyth G, Tierney L, Yang JYH, Zhang J (2004). **Bioconductor: open software development for computational biology and bioinformatics**. *Genome Biology*, 5(10):R80.

Gross J, Ligges U (2015). nortest: Tests for Normality. R package version 1.0-3. http://CRAN.R-project.org/package=nortest

Hardcastle TJ, Kelly KA (2010). **baySeq: Empirical Bayesian methods for identifying differential expression in sequence count data**. *BMC Bioinformatics.* 11:422.

Hayden EC (2009). **Genome sequencing: the third generation**. *Nature.* 457: 769.

Hollander M, Wolfe A (1973). *Nonparametric Statistical Methods*. John Wiley & Sons, New York-London-Sydney-Toronto. ISBN: 0-471-40635-X.

Kadota K, Nishiyama T, Shimizu K (2012). **A normalization strategy for comparing tag count data**. *Algorithms for Molecular Biology*, 7:5.

Kvam VM, Liu P, Si Y (2012). **A comparison of statistical methods for detecting differentially expressed genes from RNA-Seq data**. *American Journal of Botany*. 99(2): 248–256.

Law CW, Chen Y, Shi W, Smyth GK (2014). **voom: precision weights unlock linear model analysis tools for RNA-seq read counts**. *Genome Biology.* 15:R29.

Levene H (1960). **Robust tests for equality of variances**. *Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling*. Stanford University Press. pp. 278–292.

Lilliefors HW (1967). **On the Kolmogorov-Smirnov Test for Normality with Mean and Variance Unknown**. *Journal of the American Statistical Association* . Vol. 62, No. 318, pp. 399-402.

Mendiburu F (2014). agricolae: Statistical Procedures for Agricultural Research. R package version 1.2-1. http://CRAN.R-project.org/package=agricolae.

Nunes MA (2014). *Análise de Dados Genéticos*. Minicurso apresentado no XIII Encontro Mineiro de Estatística (MGEST). Disponível em http://marcusnunes.me/mgest-2014-minicurso/. Acesso em 20/12/2014.

Oshlack A, Robinson MD, Young MD (2010). **From RNA-Seq reads to differential expression results**. *Genome Biology* 11: 220.

Prati RC, Batista G, Monard MC (2008). **Curvas ROC para avaliação de classificadores**. *Revista IEEE América Latina* 6 (2), 215-222. Disponível em https://www.icmc.usp.br/~gbatista/files/ieee_la2008.pdf. Acesso em 14/04/2014.

Prati RC, Batista GE, Monard MC (2011). **A survey on graphical methods for classification predictive performance evaluation.** *IEEE Transactions on Knowledge and Data Engineering.* Vol. 23, No. 11.

R Core Team (2014). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL: http://www.R-project.org/.

Ribeiro Júnior JI (2012). *Métodos estatísticos aplicados à melhoria da qualidade*. Viçosa-MG: Editora UFV. ISBN: 9788572694346.

Riboldi J, Barbian MH, Kolowski ABS, Selau LPR, Torman VBL (2014). **Precisão e poder de testes de homocedasticidade paramétricos e não-paramétricos avaliados**

por simulação. *Revista Brasileira de Biometria*, v. 32, p. 334-344.

Robinson MD, McCarthy DJ, Smyth GK (2010). **edgeR: a Bioconductor package for differential expression analysis of digital gene expression data**. *Bioinformatics.* Vol. 26 no. 1 2010, pages 139–140.

Robinson MD, Oshlack A (2010). **A scaling normalization method for differential expression analysis of RNA-seq data**. Genome Biology, 11:R25.

Sing T, Sander O, Beerenwinkel N, Lengauer T (2005). **ROCR: visualizing classifier performance in R**. *Bioinformatics*. 21(20), pp. 7881.

Soneson C, Delorenzi M (2013). **A comparison of methods for differential expression analysis of RNA-seq data**. *BMC Bioinformatics*. 14:91.

Sun J, Nishiyama T, Shimizu K, Kadota K (2013). **TCC: an R package for comparing tag count data with robust normalization strategies**. *BMC Bioinformatics*. 14:2019.

Sun J, Nishiyama T, Shimizu K, Kadota K (2015). **TCC: Differential expression analysis for tag count data with robust normalization strategies**. Disponível em http://www.bioconductor.org/packages/release/bioc/vignettes/TCC/inst/doc/TCC.pdf. Acesso em 20/05/2015.

Triola MF (2013). *Introdução à Estatística: Atualização da Tecnologia*. 11ª edição. Rio de Janeiro: LTC.

Zaha A, Ferreira HB, Passaglia LMP (2003). *Biologia Molecular Básica*. 3ª Edição. Porto Alegre: Mercado Aberto.

Wang L, Li P, Brutnel TP (2010). **Exploring plant transcriptomes using ultra high-throughput sequencing**. *Breifings in Functional Genomics.* 9: 118 – 128.

Wang Z, Gerstein M, Snyder M (2009). **RNA-Seq: a revolutionary tool for transcriptomics**. *Nat Rev Genet*. 2009 January ; 10(1): 57–63.