DEXUS – Identifying Differential Expression in RNA-Seq Studies with Unknown Conditions

Günter Klambauer and Thomas Unterthiner
Institute of Bioinformatics, Johannes Kepler University Linz
Altenberger Str. 69, 4040 Linz, Austria
klambauer@bioinf.jku.at

Version 1.14.0, October 17, 2016
Contents

1 Introduction 3

2 Getting Started and Quick Guide 3
   2.1 Unknown Conditions ........................................ 4
   2.2 Known Conditions .......................................... 4

3 Input Data: Read Count Matrices or CountDataSets 5
   3.1 Read Count Matrices or Count Tables as Input for DEXUS .......... 6
   3.2 CountDataSets as input for DEXUS .......................... 6

4 General Study Designs: No Replicates, Unknown Sample Groups or Conditions 6

5 Case-Control Like Study Designs: Replicates, Known Sample Groups or Conditions 9
   5.1 Two Known Groups or Conditions .............................. 9
   5.2 Multiple Known Groups or Conditions ........................ 10

6 Calling Differential Expression, Visualization and the Result Object 12
   6.1 Calling Differential Expression by the Informative/Non-Informative Call .... 12
   6.2 Visualization .................................................. 13
   6.3 The Structure of the Result Object ............................. 14

7 Parameter Settings of DEXUS 15

8 The Method 16

9 A MAP Estimate for the Size Parameter and the Overdispersion of a Negative Binomial 16
1 Introduction

DEXUS identifies differentially expressed transcripts in RNA-Seq data under all possible study designs such as studies without replicates, without sample groups, and with unknown conditions. DEXUS works also for known conditions, for example for RNA-Seq data with two or multiple conditions.

RNA-Seq read count data can be provided both by the S4 class CountDataSet and by read count matrices. Differentially expressed transcripts can be visualized by heatmaps, in which unknown conditions, replicates, and samples groups are also indicated. This software is fast as the core algorithm is written in C. For very large data sets, a parallel version of DEXUS is provided in this.

DEXUS is a statistical model that is selected in a Bayesian framework by an EM algorithm. DEXUS does not need replicates to detect differentially expressed transcript, since the replicates (or conditions) are estimated by the EM method for each transcript. This is an unsupervised machine learning approach that does not require labeled data. The method provides an informative/non-informative (I/NI) value to extract differentially expressed transcripts at a desired significance level or power.

Detection of differential expression in RNA-Seq data is currently limited to studies in which two or more sample conditions are known a priori. However, these biological conditions are typically unknown in cohort, cross-sectional, and non-randomized controlled studies such as the HapMap, the ENCODE, or the 1000 Genomes project. DEXUS models read counts as a finite mixture of negative binomial distributions.

See http://www.bioinf.jku.at/software/dexus for additional information, data sets, and R scripts.

2 Getting Started and Quick Guide

To load the package, enter the following in the R session:

```r
> library(dexus)
```

With the package dexus we provide the “Mice strains” (Bottomly et al., 2011), “Primate Liver” (Blekhman et al., 2010), “Maize leaves” (Li et al., 2010), “European HapMap” (Montgomery et al., 2010), and the “Nigerian HapMap” (Pickrell et al., 2010) data sets. The read counts are stored in the objects countsBottomly, countsGilad, countsLi, countsMontgomery, and countsPickrell, respectively.

```r
> data(dexus)
> ls()

[1] "countsBottomly" "countsGilad" "countsLi"
[4] "countsMontgomery" "countsPickrell" "dexusVersion"
```
2.1 Unknown Conditions

One can simply run DEXUS by applying the function `dexus` to the count matrices. This is the mode in which the conditions are unknown, i.e. no labels that indicate the replicate groups have to be provided.

```r
> result <- dexus(countsBottomly[1:1000, ])
> plot(result)
```

Figure 1: Result of DEXUS for data with unknown conditions. A heatmap of the log read counts of the top ranked transcripts of the “Mice strains” (Bottomly et al., 2011) data set is shown. Rows represent transcripts sorted by their I/NI values. The transcript on the top has the highest I/NI value. Columns represent different samples. The labels “D2” and “B6” represent the two different strains. Red crossed indicate the samples belonging to the second condition that DEXUS has identified.

2.2 Known Conditions

To test between two or more replicate groups, DEXUS needs to be provided with the group labels:

```r
> resultSupervised <- dexus(countsBottomly[1:1000, ],
+   labels=substr(colnames(countsBottomly),1,2))
> plot(resultSupervised)
```
3 Input Data: Read Count Matrices or CountDataSets

DEXUS expects a table of counts per transcript, transcript, exon, or any other region of interest as input in analogy to other RNA-Seq analysis methods (Anders and Huber, 2010; Robinson et al., 2010; Hardcastle and Kelly, 2010; Li et al., 2012; Wang et al., 2010; Li and Tibshirani, 2011; Tarazona et al., 2011; Wu et al., 2012). The table should have the transcripts as rows and samples as columns. An entry should correspond to the number of reads of the sample mapping to the transcript. Technical replicates of one sample should be summed up so that each column corresponds to one sample. There are various ways how to produce count matrices from BAM files:

- A full guide on processing RNA-Seq data including the calculation of read count matrices is provided at http://en.wikibooks.org/wiki/Next_Generation_Sequencing_(NGS)/RNA.


- Ready-made count tables for a lot of studies are available at http://bowtie-bio.sourceforge.net/recount/ (Frazee et al., 2011).

Figure 2: Result of DEXUS for data with known conditions. A heatmap of the log read counts of the top ranked transcripts of the “Mice strains” (Bottomly et al., 2011) data set is shown. Rows represent transcripts sorted by their p-values. The transcript on the top has the lowest p-value. Columns represent different samples. The labels “D2” and “B6” represent the two different strains. Red crossed indicate the samples belonging to the second condition, that was given by the labels
The function `countOverlaps` of the Bioconductor package `GenomicRanges` can also be utilized.

The function `getSegmentReadCountsFromBAM` of the Bioconductor package `cn.mops` (Klam-bauer et al., 2012) can also be utilized to extract read counts from BAM files efficiently.

### 3.1 Read Count Matrices or Count Tables as Input for DEXUS

A read count matrix or count table should look like the following:

```r
> data(dexus)
> countsBottomly[1:10,1:5]

<table>
<thead>
<tr>
<th></th>
<th>D2_1_4</th>
<th>D2_2_4</th>
<th>D2_3_4</th>
<th>D2_5_4</th>
<th>B6_6_4</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENSMUSG00000000001</td>
<td>290</td>
<td>440</td>
<td>292</td>
<td>358</td>
<td>453</td>
</tr>
<tr>
<td>ENSMUSG00000000003</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ENSMUSG000000000028</td>
<td>17</td>
<td>15</td>
<td>17</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>ENSMUSG000000000031</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>ENSMUSG000000000037</td>
<td>12</td>
<td>5</td>
<td>4</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>ENSMUSG000000000049</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ENSMUSG000000000056</td>
<td>263</td>
<td>303</td>
<td>221</td>
<td>236</td>
<td>323</td>
</tr>
<tr>
<td>ENSMUSG000000000058</td>
<td>116</td>
<td>184</td>
<td>122</td>
<td>157</td>
<td>132</td>
</tr>
<tr>
<td>ENSMUSG000000000078</td>
<td>300</td>
<td>388</td>
<td>304</td>
<td>407</td>
<td>357</td>
</tr>
<tr>
<td>ENSMUSG000000000085</td>
<td>747</td>
<td>928</td>
<td>608</td>
<td>700</td>
<td>899</td>
</tr>
</tbody>
</table>
```

A numeric matrix of read counts can directly be used with DEXUS.

```r
> result <- dexus(countsBottomly)
```

#### 3.2 CountDataSets as input for DEXUS

A `CountDataSet`, such as the ones used in the package `DESeq` (Anders and Huber, 2010), can also directly be used it with DEXUS:

```r
> library(DESeq)
> cds <- newCountDataSet(countData=countsBottomly,
+                        conditions=substr(colnames(countsBottomly),1,2) )
> result <- dexus(cds)
```

### 4 General Study Designs: No Replicates, Unknown Sample Groups or Conditions

Examples of studies in which the groups are unknown, are the studies of Montgomery et al. (2010) and Pickrell et al. (2010). They sequenced the RNA of HapMap individuals to investigate eQTLs.
DEXUS is able to identify differential expression in these data sets. The method estimates the conditions for each transcript individually.

To run the method simply apply the function dexus to the count table. In the following example we run the algorithm only one the first 1000 transcripts.

```r
> resultMontgomery <- dexus(countsMontgomery[1:1000, ])
```

To show a summary of the result object, simply type the following.

```r
> resultMontgomery
```

Displaying the 10 top ranked genes of the analysis:

<table>
<thead>
<tr>
<th>Index</th>
<th>Transcript</th>
<th>INIcall</th>
<th>INI Mean.Condition_1</th>
<th>Mean.Condition_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ENSG00000007038</td>
<td>TRUE</td>
<td>1.619</td>
<td>1.0</td>
</tr>
<tr>
<td>2</td>
<td>ENSG00000022556</td>
<td>TRUE</td>
<td>1.06</td>
<td>172.0</td>
</tr>
<tr>
<td>3</td>
<td>ENSG00000069011</td>
<td>TRUE</td>
<td>0.616</td>
<td>1.1</td>
</tr>
<tr>
<td>4</td>
<td>ENSG0000008196</td>
<td>TRUE</td>
<td>0.608</td>
<td>0.8</td>
</tr>
<tr>
<td>5</td>
<td>ENSG000000084710</td>
<td>TRUE</td>
<td>0.583</td>
<td>5.3</td>
</tr>
<tr>
<td>6</td>
<td>ENSG00000076716</td>
<td>TRUE</td>
<td>0.564</td>
<td>136.7</td>
</tr>
<tr>
<td>7</td>
<td>ENSG00000057294</td>
<td>TRUE</td>
<td>0.520</td>
<td>1.1</td>
</tr>
<tr>
<td>8</td>
<td>ENSG00000047648</td>
<td>TRUE</td>
<td>0.423</td>
<td>29.6</td>
</tr>
<tr>
<td>9</td>
<td>ENSG00000052723</td>
<td>TRUE</td>
<td>0.418</td>
<td>6.6</td>
</tr>
<tr>
<td>10</td>
<td>ENSG00000062370</td>
<td>TRUE</td>
<td>0.381</td>
<td>1.1</td>
</tr>
</tbody>
</table>

Total number of transcripts: 1000  
Number of differentially expressed transcripts: 105  
Percentage of differentially expressed transcripts: 10.5 %

The transcripts are in their original order; the displayed columns give the whether a transcript is differentially expressed (INIcall), the evidence for differential expression measured by the INI values (INIValues), and the means for each condition.

It is possible to sort the result object such that the transcripts with the highest INI values are ranked highest.

```r
> sort(resultMontgomery)
```

Transcripts with an INI value above 0.1 are classified as differentially expressed. The function INI filters the result object for informative transcripts and sorts them by their INI calls.

```r
> informativeTranscripts <- INI(resultMontgomery, threshold=0.2)
```

Total number of transcripts: 1000  
Number of differentially expressed transcripts: 32  
Percentage of differentially expressed transcripts: 3.2 %
The result can be visualized by a heatmap using the `plot` function.

```r
> plot(informativeTranscripts)
```

![Heatmap](image.png)

Figure 3: Result of DEXUS in unsupervised mode with unknown conditions. A heatmap of the log read counts of the top ranked transcripts of the “European HapMap” (Montgomery et al., 2010) data set is shown. Rows represent transcripts sorted by their I/NI values. The transcript on the top has the highest I/NI. Columns represent different samples. Red symbols indicate samples that belong to the minor condition that was identified by DEXUS.

Information about a specific transcript can also be accessed from the result object by subsetting it with the transcript name.

```r
> resultMontgomery["ENSG00000007038"]
```

Displaying the 10 top ranked genes of the analysis:

<table>
<thead>
<tr>
<th>Index</th>
<th>Transcript</th>
<th>INIcall</th>
<th>INI</th>
<th>Mean.Condition_1</th>
<th>Mean.Condition_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ENSG00000007038</td>
<td>TRUE</td>
<td>1.619</td>
<td>1</td>
<td>40.6</td>
</tr>
</tbody>
</table>

Total number of transcripts: 1
Number of differentially expressed transcripts: 1
Percentage of differentially expressed transcripts: 100 %

Even more information can be obtained by using the `as.data.frame` function.

```r
> as.data.frame(resultMontgomery["ENSG00000007038"])
```
### 5 Case-Control Like Study Designs: Replicates, Known Sample Groups or Conditions

#### 5.1 Two Known Groups or Conditions

In the study of Bottomly *et al.* (2011), two strains of mice, C57BL/6J (B6) and DBA/2J (D2), were compared using both RNA-Seq and microarrays. The data set consists of 21 lanes from male mice (10 of the B6 strain and 11 of D2 strain), produced using an Illumina GAIIx sequencing machine. The data set was provided by the ReCount repository (Frazee *et al.*, 2011) that is based on Ensembl 61 transcript definitions. In this case of two known conditions we provide DEXUS with the group labels, in order to detect transcripts that are differentially expressed between the two mice strains.

We apply the function `dexus` to the count table of the first 1000 transcripts and provide the labels of the samples, and set the normalization to “Upper Quartile” normalization.

```r
> resultSupervised <- dexus(countsBottomly[1:1000,],
+   labels=substr(colnames(countsBottomly),1,2),
+   normalization="upperquartile")
```

To show a list of differentially expressed transcripts, simply type the name of the result object.

```r
> resultSupervised
```

Displaying the 10 top ranked genes of the analysis:

<table>
<thead>
<tr>
<th>Index</th>
<th>Transcript</th>
<th>pvalues</th>
<th>Mean.Condition_1</th>
<th>Mean.Condition_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ENSMUSG00000000544</td>
<td>3.160461e-50</td>
<td>0.6</td>
<td>36.1</td>
</tr>
<tr>
<td>2</td>
<td>ENSMUSG000000003477</td>
<td>9.954497e-19</td>
<td>19.8</td>
<td>2.7</td>
</tr>
<tr>
<td>3</td>
<td>ENSMUSG000000002324</td>
<td>9.198666e-18</td>
<td>38.2</td>
<td>3.7</td>
</tr>
<tr>
<td>4</td>
<td>ENSMUSG000000002396</td>
<td>1.398955e-16</td>
<td>26.7</td>
<td>102.5</td>
</tr>
<tr>
<td>5</td>
<td>ENSMUSG000000004415</td>
<td>2.422523e-15</td>
<td>56.8</td>
<td>16.6</td>
</tr>
</tbody>
</table>
### 5.2 Multiple Known Groups or Conditions

Blekhman *et al.* (2010) investigated the differences in alternative splicing in liver tissue between humans, chimpanzees and rhesus macaques. For this purpose they performed RNA-Seq on three male and three female liver samples from each species. They focused on the expression values of exons that had reliably determined orthologs in all species. Read counts for exons were provided by Blekhman *et al.* (2010), who used transcript models from Ensemble (Release 50). In this case the three species are three distinct groups. The aim is to find transcripts, that show large differences between these groups.

We run DEXUS on this data set and provide the method with the group labels, i.e. the species.

```r
> resultMultipleGroups <- dexus(countsGilad[1:1000, ],
+    labels=substr(colnames(countsGilad),1,2))
```

To show a list of differentially expressed transcripts, simply type the name of the result object.

```r
> resultMultipleGroups
```
Displaying the 10 top ranked genes of the analysis:

<table>
<thead>
<tr>
<th>Index</th>
<th>Transcript</th>
<th>pvalues</th>
<th>Mean.Condition_1</th>
<th>Mean.Condition_2</th>
<th>Mean.Condition_3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ENSG00000002726</td>
<td>8.680312e-06</td>
<td>27.6</td>
<td>1.1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>ENSG00000010379</td>
<td>5.191181e-05</td>
<td>177.9</td>
<td>39.2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>ENSG0000007174</td>
<td>5.364664e-05</td>
<td>0.7</td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>ENSG00000025423</td>
<td>6.321030e-05</td>
<td>1187.6</td>
<td>41.2</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>ENSG00000037042</td>
<td>8.129194e-05</td>
<td>25.9</td>
<td>50.7</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>ENSG00000010626</td>
<td>8.867209e-05</td>
<td>9.1</td>
<td>15.2</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>ENSG00000066629</td>
<td>9.086187e-05</td>
<td>44.0</td>
<td>119.2</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>ENSG00000050628</td>
<td>9.132719e-05</td>
<td>2.3</td>
<td>1.4</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>ENSG00000010219</td>
<td>1.089026e-04</td>
<td>38.5</td>
<td>64.1</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>ENSG00000054277</td>
<td>1.141968e-04</td>
<td>127.3</td>
<td>90.3</td>
<td></td>
</tr>
</tbody>
</table>

Mean.Condition_3

| 1 | 17710.6 |
| 2 | 1.2    |
| 3 | 51.3   |
| 4 | 1850.5 |
| 5 | 1.9    |
| 6 | 0.5    |
| 7 | 498.2  |
| 8 | 91.5   |
| 9 | 1.3    |
| 10| 18.2   |

Total number of transcripts: 1000
Number of differentially expressed transcripts: 841
Percentage of differentially expressed transcripts: 84.1%

To sort the transcripts in the result object by p-values use the sort method.

```r
> resultMultipleGroups <- sort(resultMultipleGroups)
```

To obtain a heatmap of the top-ranked transcripts use the plot function.

```r
> plot(resultMultipleGroups)
```
6 Calling Differential Expression, Visualization and the Result Object

6.1 Calling Differential Expression by the Informative/Non-Informative Call

In a setting in which the conditions or sample groups are unknown, or in which there are no replicates, the I/NI value measures the evidence for differential expression. At different thresholds DEXUS has different detection powers (sensitivity) and significance levels (specificity). On 2,400 simulated data sets, I/NI value thresholds of 0.025, 0.05, and 0.1 yielded average specificities of 92%, 97%, and 99% at sensitivities of 76%, 61%, and 38% respectively. The threshold for the I/NI values is set by the function INIThreshold. The function I/NI filters out non-informative transcripts.
> informativeTranscripts2 <- INI(resultMontgomery, threshold=0.25)

Total number of transcripts:  1000  
Number of differentially expressed transcripts:  23  
Percentage of differentially expressed transcripts:  2.3 %

The object informativeTranscripts2 contains only the informative, i.e. the differentially expressed transcripts.

6.2 Visualization

There is a generic plotting function that can be applied to the result object of DEXUS. The log read counts are visualized as a heatmap, in which we also indicate the identified sample condition. We can select which transcripts we want to plot by using the parameter idx.

> # plots the top 8 transcripts
> plot(sort(informativeTranscripts2), idx=1:8)

Figure 5: Result of DEXUS in unsupervised mode with unknown conditions. A heatmap of the log read counts of the top ranked transcripts of the “European HapMap” (Montgomery et al., 2010) data set is shown. Rows represent transcripts sorted by their I/NI values. The transcript on the top has the highest I/NI. Columns represent different samples. Red symbols indicate samples that belong to the minor condition that was identified by DEXUS.
6.3 The Structure of the Result Object

DEXUS returns an instance of “DEXUSResult” that contains the following slots:

- **transcriptNames**: The names of the transcripts, genes, exons, or regions of interest.
- **sampleNames**: The sample names, as they were given in the input matrix.
- **inputData**: The original read count matrix.
- **normalizedData**: The normalized read count matrix.
- **sizeFactors**: The size factors that were calculated for the normalization. This is that factor that scales each column or sample.
- **INIValues**: An informative/non-informative (I/NI) value for each sample that measures the evidence for differential expression.
- **INIThreshold**: The threshold for the I/NI values. Transcript with I/NI values above the threshold will be considered as differentially expressed.
- **INICalls**: A binary value for each transcript indicating whether it is differentially expressed.
- **pvals**: In case of two known conditions or multiple known conditions it is possible to calculate a p-value for each transcript. This value is given in this slot.
- **responsibilites**: A matrix of the size of the input matrix. It indicates the condition for each sample and transcript. The condition named “1” is the major condition. All other conditions are minor conditions. In case of supervised (two known conditions or multiple known conditions) analyses this clustering matrix will be the same for all transcripts.
- **posteriorProbs**: An array of the dimension of transcripts times samples times conditions. It gives the probability that a certain read count \( x \) was generated under a condition.
- **logFC**: The log foldchanges between the conditions. The reference is always condition “1”.
- **conditionSizes**: The ratio of samples belonging to that condition. These are the \( \alpha_i \) values of the model.
- **sizeParameters**: The size parameter estimates for each condition. These are the \( r_i \) values of the model.
- **means**: The mean of each condition. The \( \mu_i \) values of the model.
- **dispersions**: The dispersion estimates for each condition. The inverse size parameters.
- **params**: The input parameters of the DEXUS algorithm.
Parameter Settings of DEXUS

The input parameters of the DEXUS algorithm are the following:

- \( X \): The read count matrix. If the reads are already normalized, then set \texttt{normalization} to "none".

- \( n\text{classes} \): The number of conditions that DEXUS should model. The number should be much smaller than the number of samples. In supervised mode (two known conditions or multiple known conditions) the algorithm uses the number of different labels as \( n\text{classes} \). Needs not be specified in supervised mode.

- \( \alpha\text{Init} \): The initialization of the \( \alpha_i \) values of the model. A vector with length of the number of conditions. The algorithm internally scales this vector to sum 1. Needs not be specified in supervised mode.

- \( G \): The weight of the Dirichlet prior. An important parameter that guides the EM algorithm. The higher this value the more transcripts will be explained by one condition and will therefore be classified as not differentially expressed. The lower the value of \( G \) the more transcripts will be found to be differentially expressed. Needs not be specified in supervised mode.

- \( \text{cyc} \): The number of cycles of the EM algorithm per transcript. Needs not be specified in supervised mode.

- \( \text{labels} \): If the conditions, groups, or classes are known, then they can be passed to the algorithm through this parameter.

- \( \text{normalization} \): The normalization method to be used. Choices are "RLE", "upperquartile", and "none".

- \( \text{kmeansIter} \): For the initialization of the algorithm a k-means clustering is run. This is the number of iterations of the clustering.

- \( \text{ignoreIfAllCountsSmaller} \): A transcript is considered as “not expressed”, if counts of all samples are below this value. The algorithm is not applied to these transcripts.

- \( \text{theta} \): The weight of the exponential prior on the size parameter of the negative binomial distributions. The higher this parameter, the lower the estimates of the size parameters, and consequently the higher the estimates of the overdispersions.

- \( \text{minMu} \): The minimal value for the mean parameter of the negative binomial distribution.

- \( \text{rmax} \): An upper bound for the size parameter and thereby a lower bound for the overdispersion. The value is set to the value that \texttt{DESeq} uses for this purpose.

- \( \text{initialization} \): How the initial estimates of the conditions are determined. Possible choices are “kmeans” and “quantiles”. Needs not be specified in supervised mode.

- \( \text{multiClassPhiPoolingFunction} \): In case of multiple known conditions it is possible to calculate one overdispersion value per transcript. This can be calculated over all conditions or as mean, maximum or minimum over the specified conditions. Usually the option “NULL” (calculation of the overdispersion across all conditions) performs best.
8 The Method

**DEXUS** models read counts as a finite mixture of negative binomial distributions in which each mixture component corresponds to a condition. **DEXUS** classifies a transcript as differentially expressed if modeling of its read counts requires more than one condition. To account for the high overdispersion observed in RNA-Seq data, **DEXUS** assumes that under each condition the read counts are drawn from a negative binomial distribution. Read count \( x \) is explained by a mixture of \( n \) negative binomial distributions:

\[
p(x) = \sum_{i=1}^{n} \alpha_i \text{NB}(x; \mu_i, r_i),
\]

where \( \alpha_i \) is the probability of being in condition \( i \) out of \( n \) possible conditions. In condition \( i \), read counts are drawn from a negative binomial distribution with mean \( \mu_i \) and size \( r_i \), where the size parameter \( r_i \) is the inverse of the overdispersion \( \phi_i \). An expectation maximization (EM) algorithm is used to estimate mean and overdispersion parameters of the negative binomials as well as the condition under which a particular read count was generated. **DEXUS** decomposes read count variation into variation due to noise and variation due to differential expression. The evidence for differential expression is measured by an informative/non-informative (I/NI) value. **DEXUS** applies a threshold to the I/NI value to extract differentially expressed transcripts with a desired specificity (significance level) or sensitivity (power).

**DEXUS** performs excellently in identifying differentially expressed transcripts on data with unknown conditions. **DEXUS** was tested on 2,400 simulated data sets. For I/NI value thresholds of 0.025, 0.05, and 0.1, it yielded average specificities of 92%, 97%, and 99% at sensitivities of 76%, 61%, and 38%, respectively. Subsequently, **DEXUS** was tested on real-world data sets, in which it identified differentially expressed transcripts between subgroups defined by sex, species, or tissue although information about these subgroups was withheld. On HapMap individuals, **DEXUS** detected several differentially expressed transcripts, the vast majority of which are related to sex, eQTLs, or copy number variable regions. However, we were unable to interpret the conditions for some differentially expressed transcripts which hints at the existence of another cause of differential expression.

9 A MAP Estimate for the Size Parameter and the Overdispersion of a Negative Binomial

We provide the function `getSizeNB` that gives an estimate for the size parameter of a negative binomial distribution from given data. In this function the maximum-likelihood estimate is used, if the argument `eta` is set to 0, and if `eta` is set to a value greater than 0, a maximum-a-posteriori estimator for the size parameter is calculated. In that case an exponential prior is used. The argument `eta` determines the weight of this prior.

The maximum-likelihood estimator overestimates the size parameter and, thus, underestimates the overdispersion parameter *Piegorsch* (1990). The maximum-a-posteriori estimator can correct for this bias and decreases the variance of the estimator, as we show in the following example. Another problem is that, if the mean of the given data exceeds the variance, the maximum-likelihood-
estimator tends to infinity Anscombe (1950). By setting the argument \( r_{\text{max}} \) to a positive value, one can infer an upper bound on the size parameter and, thereby, a lower bound on the overdispersion.

```r
> trueSizeParameter <- 2
> x <- rnbinom(n=5, size=trueSizeParameter, mu=40)
> (sizeML <- getSizeNB(x,eta=0))

[1] 59.26277

> (sizeMAP <- getSizeNB(x,eta=1))

[1] 2.32336

> (trueDispersion <- 1/trueSizeParameter)

[1] 0.5

> (dispersionML <- 1/getSizeNB(x,eta=0))

[1] 0.016874

> (dispersionMAP <- 1/getSizeNB(x,eta=1))

[1] 0.4304112
```

References


