Software Manual

Institute of Bioinformatics, Johannes Kepler University Linz

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 1.8.0, April 16, 2015

Contents

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:

```
> 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.

```
> 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.

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

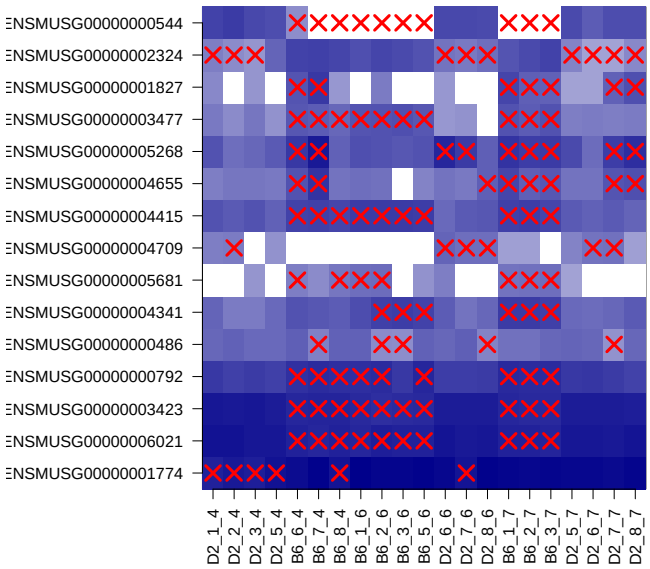

Figure 1: Result of DEXUS for data with *unknown* conditions. A heatmap of the log read counts of the top ranked transcripts of the "Mice strains" (Bottomly *et al.*, 2011) data set is shown. Rows represent transcripts sorted by their I/NI values. The trancript on the top has the highest I/NI value. 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:

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

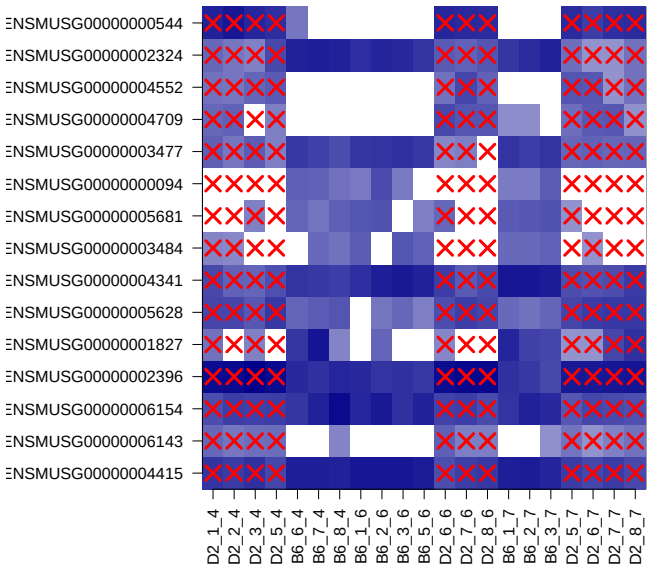

Figure 2: Result of DEXUS for data with *known* conditions. A heatmap of the log read counts of the top ranked transcripts of the "Mice strains" (Bottomly *et al.*, 2011) data set is shown. Rows represent transcripts sorted by their p-values. The trancript on the top has the lowest p-value. Columns represent 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

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`.
- The function `HTSeq-count` of `HTSeq` Python package `http://www-huber.embl.de/users/anders/HTSeq/doc/count.html`.
- Ready-made count tables for a lot of studies are available at `http://bowtie-bio.sourceforge.net/recount/` (Frazee *et al.*, 2011).

- The function `countOverlaps` of the Bioconductor package `GenomicRanges` can also be utilized.
- The function `getSegmentReadCountsFromBAM` of the Bioconductor package `cn.mops` (Klambauer *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:

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

	D2_1_4	D2_2_4	D2_3_4	D2_5_4	B6_6_4
ENSMUSG00000000001	290	440	292	358	453
ENSMUSG00000000003	0	0	0	0	0
ENSMUSG00000000028	17	15	17	10	20
ENSMUSG00000000031	0	0	1	0	1
ENSMUSG00000000037	12	5	4	6	2
ENSMUSG00000000049	0	2	0	0	0
ENSMUSG00000000056	263	303	221	236	323
ENSMUSG00000000058	116	184	122	157	132
ENSMUSG00000000078	300	388	304	407	357
ENSMUSG00000000085	747	928	608	700	899

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

```
> 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:

```
> 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.

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

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

```
> resultMontgomery
```

```
Displaying the 10 top ranked genes of the analyis:
   Index      Transcript INIcall   INI Mean.Condition_1 Mean.Condition_2
1     88 ENSG00000007038    TRUE 1.619             1.0            40.6
2    224 ENSG00000022556    TRUE 1.106           172.0             0.8
3    569 ENSG00000069011    TRUE 0.616             1.1             9.8
4    114 ENSG00000008196    TRUE 0.608             0.8            12.3
5    866 ENSG00000084710    TRUE 0.583             5.3            39.8
6    732 ENSG00000076716    TRUE 0.564           136.7             1.1
7    417 ENSG00000057294    TRUE 0.520             1.1            14.1
8    337 ENSG00000047648    TRUE 0.423            29.6             2.8
9    379 ENSG00000052723    TRUE 0.418             6.6             0.8
10   450 ENSG00000062370    TRUE 0.381             1.1             7.8

Total number of transcripts:  1000
Number of differentially expressed transcripts:  104
Percentage of differentially expressed transcripts:  10.4 %
```

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 I/NI values (`INIValues`), and the means for each condition.

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

```
> sort(resultMontgomery)
```

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

```
> 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.

```
> plot(informativeTranscripts)
```

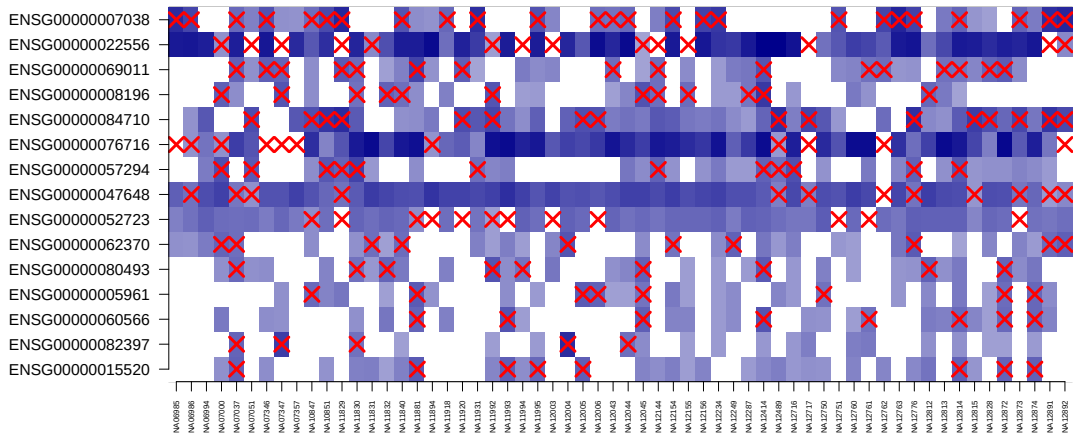

Figure 3: Result of DEXUS in unsupervised mode with *unknown conditions*. A heatmap of the log read counts of the top ranked transcripts of the "European HapMap" (Montgomery *et al.*, 2010) data set is shown. Rows represent transcripts sorted by their I/NI values. The trancript on the top has the highest 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.

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

```
Displaying the 10 top ranked genes of the analyis:
  Index      Transcript INIcall   INI Mean.Condition_1 Mean.Condition_2
1     1 ENSG00000007038    TRUE 1.619                1             40.6

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.

```
> as.data.frame(resultMontgomery["ENSG00000007038"])
```

```
       Transcript INIcall      INI pval Mean.Condition_1 Mean.Condition_2
1 ENSG00000007038    TRUE 1.619496   NA        0.9659015          40.5991
  logFC.Condition_1 logFC.Condition_2 conditionSize.Condition_1
1                 0          3.738338                 0.5667871
  conditionSize.Condition_2 dispersion.Condition_1 dispersion.Condition_2
1                 0.4332129             0.07692308               1.082242
```

To convert the full result object to a data frame that can be exported the function `as.data.frame` can be used.

```
> as.data.frame(sort(resultMontgomery))
```

For more information on the result object, see Section 6.

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.

```
> 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.

```
> resultSupervised
```

```
Displaying the 10 top ranked genes of the analyis:
   Index         Transcript    pvalues Mean.Condition_1 Mean.Condition_2
1    691 ENSMUSG00000004267 0.01075699          11029.5          12489.9
2    112 ENSMUSG00000000632 0.01181238           5748.2           6903.4
3    597 ENSMUSG00000003469 0.01264576           7556.2           8032.5
4    276 ENSMUSG00000001525 0.01326458           4922.6           5479.5
5    973 ENSMUSG00000006273 0.01349587           8199.7           8431.8
```

```
6    683 ENSMUSG00000004187 0.01403671          4448.6          5012.4
7    402 ENSMUSG00000002265 0.01406697          8999.1          9080.6
8    118 ENSMUSG00000000701 0.01438271             0.0             0.0
9    849 ENSMUSG00000005442 0.01484236          3061.5          3636.0
10    92 ENSMUSG00000000538 0.01485909          4481.5          4924.3

Total number of transcripts:  1000
Number of differentially expressed transcripts:  252
Percentage of differentially expressed transcripts:  25.2 %
```

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

```
> resultSupervised <- sort(resultSupervised)
```

To obtain a heatmap of the differentially expressed transcripts, type `plot`.

```
> plot(resultSupervised)
```

To get the full list of transcripts together with additional information, such as the I/NI values, conditions, dispersions, and means the function `as.data.frame` can be used.

```
> as.data.frame(resultSupervised)
```

For more information on the result object, see Section 6.

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.

```
> 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.

```
> resultMultipleGroups
```

```
Displaying the 10 top ranked genes of the analyis:
   Index      Transcript      pvalues Mean.Condition_1 Mean.Condition_2
1     26 ENSG00000002726 8.680312e-06             27.6              1.1
2    253 ENSG00000010379 5.191181e-05            177.9             39.2
3    168 ENSG00000007174 5.364664e-05              0.7              0.7
4    432 ENSG00000025423 6.321030e-05           1187.6             41.2
5    521 ENSG00000037042 8.129194e-05             25.9             50.7
6    257 ENSG00000010626 8.867209e-05              9.1             15.2
7    951 ENSG00000066629 9.086187e-05             44.0            119.2
8    667 ENSG00000050628 9.132719e-05              2.3              1.4
9    240 ENSG00000010219 1.089026e-04             38.5             64.1
10   707 ENSG00000054277 1.141968e-04            127.3             90.3
   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.

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

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

```
> plot(resultMultipleGroups)
```

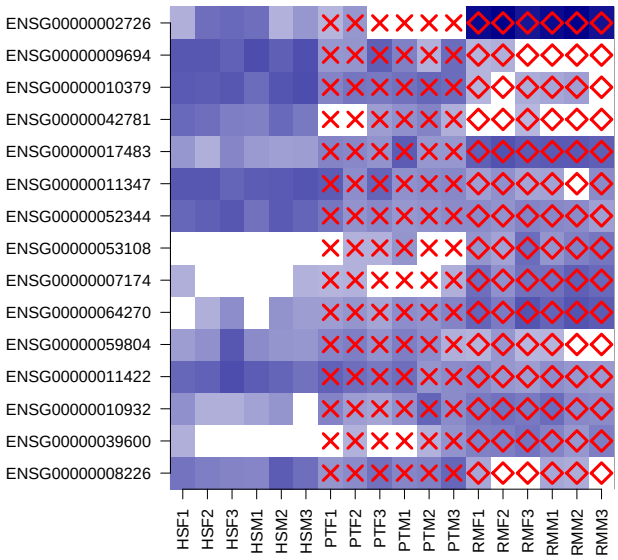

Figure 4: Result of DEXUS in supervised mode with *multiple known groups*. A heatmap of the log read counts of the top ranked transcripts of the "Primate Liver" (Blekhman *et al.*, 2010) data set is shown. Rows represent transcripts sorted by their p-values. The trancript on the top has the lowest p-value. Columns represent different samples. The labels "HS","PT" and "MM" represent the three different species. Red symbols indicate the different species.

To get the full list of transcripts together with their p-values the function `getResult` can be used.

```
> as.data.frame(resultMultipleGroups)
```

For more information on the result object, see Section 6.

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 `INI` 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)
```

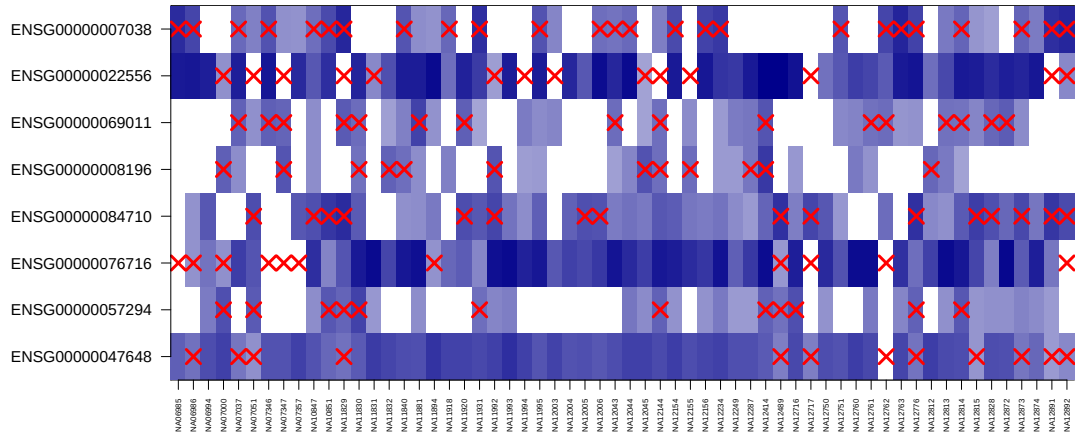

Figure 5: Result of DEXUS in unsupervised mode with *unknown conditions*. A heatmap of the log read counts of the top ranked transcripts of the "European HapMap" (Montgomery *et al.*, 2010) data set is shown. Rows represent transcripts sorted by their I/NI values. The trancript on the top has the highest 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 α_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 μ_i values of the model.
- `dispersions`: The dispersion estimates for each condition. The inverse size parameters.
- `params`: The input parameters of the DEXUS algorithm.

7 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 `normalization` to "`none`".
- `nclasses`: 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 `nclasses`. Needs not be specified in supervised mode.
- `alphaInit`: The initialization of the α_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.
- `cyc`: The number of cycles of the EM algorithm per transcript. Needs not be specified in supervised mode.
- `labels`: If the conditions, groups, or classes are known, then they can be passed to the algorithm through this parameter.
- `normalization`: The normalization method to be used. Choices are "RLE", "upperquartile", and "none".
- `kmeansIter`: For the initialization of the algorithm a k-means clustering is run. This is the number of iterations of the clustering.
- `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.
- `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.
- `minMu`: The minimal value for the mean parameter of the negative binomial distribution.
- `rmax`: An upper bound for the size parameter and thereby a lower bound for the overdispersion. The value is set to the value that DESeq uses for this purpose.
- `initializiation`: How the initial estimates of the conditions are determined. Possible choices are "kmeans" and "quantiles". Needs not be specified in supervised mode.
- `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:

$$p(x) \;=\; \sum_{i=1}^{n} \alpha_i \,\mathrm{NB}(x \;;\; \mu_i, r_i) \;, \tag{1}$$

where α_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 μ_i and size r_i, where the size parameter r_i is the inverse of the overdispersion ϕ_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 `rmax` to a positive value, one can infer an upper bound on the size parameter and, thereby, a lower bound on the overdispersion.

```
> 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

Anders, S. and Huber, W. (2010). Differential expression analysis for sequence count data. *Genome Biology*, **11**(10), R106.

Anscombe, F. J. (1950). Sampling theory of the negative binomial and logarithmic series distributions. *Biometrika*, **37**(3/4), 358–382.

Blekhman, R., Marioni, J. C., Zumbo, P., Stephens, M., and Gilad, Y. (2010). Sex-specific and lineage-specific alternative splicing in primates. *Genome Research*, **20**(2), 180–189.

Bottomly, D., Walter, N. A. R., Hunter, J. E., Darakjian, P., Kawane, S., Buck, K. J., Searles, R. P., Mooney, M., McWeeney, S. K., and Hitzemann, R. (2011). Evaluating gene expression in C57BL/6J and DBA/2J mouse striatum using RNA-Seq and microarrays. *PLoS One*, **6**(3), e17820.

Frazee, A. C., Langmead, B., and Leek, J. T. (2011). ReCount: a multi-experiment resource of analysis-ready RNA-seq gene count datasets. *BMC Bioinformatics*, **12**, 449.

Hardcastle, T. J. and Kelly, K. A. (2010). baySeq: empirical Bayesian methods for identifying differential expression in sequence count data. *BMC Bioinformatics*, **11**, 422.

Klambauer, G., Schwarzbauer, K., Mayr, A., Clevert, D.-A., Mitterecker, A., Bodenhofer, U., and Hochreiter, S. (2012). cn.MOPS: mixture of Poissons for discovering copy number variations in next-generation sequencing data with a low false discovery rate. *Nucleic Acids Research*, **40**(9), e69.

Li, J. and Tibshirani, R. (2011). Finding consistent patterns: A nonparametric approach for identifying differential expression in RNA-seq data. *Statistical Methods in Medical Research*, **Published online**.

Li, J., Witten, D. M., Johnstone, I. M., and Tibshirani, R. (2012). Normalization, testing, and false discovery rate estimation for RNA-sequencing data. *Biostatistics*, **13**(3), 523–538.

Li, P., Ponnala, L., Gandotra, N., Wang, L., Si, Y., Tausta, S. L., Kebrom, T. H., Provart, N., Patel, R., Myers, C. R., Reidel, E. J., Turgeon, R., Liu, P., Sun, Q., Nelson, T., and Brutnell, T. P. (2010). The developmental dynamics of the maize leaf transcriptome. *Nature Genetics*, **42**(12), 1060–1067.

Montgomery, S. B., Sammeth, M., Gutierrez-Arcelus, M., Lach, R. P., Ingle, C., Nisbett, J., Guigo, R., and Dermitzakis, E. T. (2010). Transcriptome genetics using second generation sequencing in a caucasian population. *Nature*, **464**(7289), 773–777.

Pickrell, J. K., Marioni, J. C., Pai, A. A., Degner, J. F., Engelhardt, B. E., Nkadori, E., Veyrieras, J.-B., Stephens, M., Gilad, Y., and Pritchard, J. K. (2010). Understanding mechanisms underlying human gene expression variation with rna sequencing. *Nature*, **464**(7289), 768–772.

Piegorsch, W. W. (1990). Maximum likelihood estimation for the negative binomial dispersion parameter. *Biometrics*, **46**(3), 863–867.

Robinson, M. D., McCarthy, D. J., and Smyth, G. K. (2010). edgeR: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, **26**(1), 139–140.

Tarazona, S., García-Alcalde, F., Dopazo, J., Ferrer, A., and Conesa, A. (2011). Differential expression in RNA-seq: A matter of depth. *Genome Research*, **8**.

Wang, L., Feng, Z., Wang, X., Wang, X., and Zhang, X. (2010). DEGseq: an R package for identifying differentially expressed genes from RNA-seq data. *Bioinformatics*, **26**(1), 136–138.

Wu, H., Wang, C., and Wu, Z. (2012). A new shrinkage estimator for dispersion improves differential expression detection in rna-seq data. *Biostatistics*.