A guide to Dynamic Transcriptome Analysis (DTA)

Bjoern Schwalb, Benedikt Zacher, Sebastian Duemcke, Achim Tresch

April 24, 2017

1 What is Dynamic Transcriptome Analysis?

Total RNA levels in a cell are the consequence of two opposing mechanisms, namely RNA synthesis and RNA degradation. DTA allows monitoring these contributions in a non-perturbing manner. It is provided with a kinetic modeling approach capable of the precise determination of synthesis- and decay rates, which is implemented in this package (see supplementary methods in [6]). DTA can be applied to reveal rate changes for all kinds of perturbations, e.g. in knock-out or point mutation strains, as responses to stress stimuli or in small molecule interfering assays like treatments through miRNA or siRNA inhibitors.
The experimental setup for DTA requires culturing cells in the presence of a labeling substrate (e.g. 4 thiouridine (4sU) or 4 thiouracil (4tU)) for a certain amount of time. Until the extraction of the RNA samples, the analogous (labeling) substrate will be incorporated into newly transcribed RNA. This setup yields three types of RNA fractions: total cellular RNA, newly transcribed labeled RNA and pre-existing unlabeled RNA. All three fractions can subsequently be quantified through gene expression profiling on microarrays or next generation sequencing (RNAseq).
The *DTA* package implements methods to process a given DTA experiment (total (T), labeled (L) and unlabeled (U) RNA measurements) and derive RNA synthesis and decay rate estimates. The package *DTA* is broadly applicable to virtually every organism. This manual is a hands-on tutorial and describes all functions and R commands that are required. The method is applied to real and synthetic data. Important quality control plots and error assessment to prove the reliability of the data and the estimation are discussed throughout this manual. For details on the theoretical background and a thorough description of the experimental design, we refer the reader to [6] (supplementary methods).

2 Estimation of synthesis and decay rates in steady state (`DTA.estimate`)

This section illustrates the function `DTA.estimate`, which calculates steady state mRNA synthesis and decay rates from measurements of individual labeling durations using a first order exponential decay model, which is described in [6]. Before starting, the package library must be loaded by typing:

```
> library(DTA)
```

2.1 `DTA.estimate` for real data in yeast

The package attachement contains two biological replicates of T, U and L for a labling time of 6 and 12 minutes of the yeast DTA data set published in [6]. Hereafter, the individual R objects, needed for the analysis are seperately loaded and described. The `*.RData` objects are loaded as follows:

```
> data(Miller2011)
```

The gene expression profiles are stored in `datamat`. This *matrix* contains the RNA intensity values for each gene across each RNA fraction and their replicate measurements. The column names of the matrix give the cel-file name and the row names the ORF identifier. DTA was performed for a labeling duration of 6 ("00-06") and 12 ("00-12") minutes in wild-type yeast cells.

```
> colnames(Sc.datamat)[1:6]

[1] "L 00-06 - WT 1 1 WT.CEL" "L 00-12 - WT 1 2 WT.CEL"
[3] "L 00-06 - WT 2 1 WT.CEL" "L 00-12 - WT 2 2 WT.CEL"
[5] "T 00-06 - WT 1 1 WT.CEL" "T 00-12 - WT 1 2 WT.CEL"

> rownames(Sc.datamat)[1:6]

[1] "YHR055C" "YPR161C" "YOL138C" "YDR395W" "YGR129W" "YPR165W"

> Sc.datamat[1:6,1:3]

        L 00-06 - WT 1 1 WT.CEL L 00-12 - WT 1 2 WT.CEL L 00-06 - WT 2 1 WT.CEL
YHR055C               47862.980               34254.234               32272.865
YPR161C                5922.690                2604.864                4093.175
YOL138C                4238.608                1805.858                3080.328
YDR395W                8762.665                6692.731                6325.877
YGR129W                1568.181                1081.841                1750.127
YPR165W               16844.130               12387.822               11078.113
```

The `phenomat` contains information about the experimental design. It is comprised of the file name, the type of RNA fraction measured (T, U or L), the labeling time and the replicate number. Rows in this *matrix* represent the individual experiments.

```
> head(Sc.phenomat)

                         name                        fraction time nr
L 00-06 - WT 1 1 WT.CEL "L 00-06 - WT 1 1 WT.CEL" "L"      "6"  "1"
L 00-12 - WT 1 2 WT.CEL "L 00-12 - WT 1 2 WT.CEL" "L"      "12" "2"
L 00-06 - WT 2 1 WT.CEL "L 00-06 - WT 2 1 WT.CEL" "L"      "6"  "3"
L 00-12 - WT 2 2 WT.CEL "L 00-12 - WT 2 2 WT.CEL" "L"      "12" "4"
T 00-06 - WT 1 1 WT.CEL "T 00-06 - WT 1 1 WT.CEL" "T"      "6"  "1"
T 00-12 - WT 1 2 WT.CEL "T 00-12 - WT 1 2 WT.CEL" "T"      "12" "2"
```

During the labeling procedure of the experiment about 1 out of approx. 200 uridines is replaced by 4sU and biotinylated. Thus not all newly synthesized RNAs are actually labeled. The proportion of newly transcribed RNAs that are 4sU labeled and captured in the L fraction thus increases with the uridine content of a gene [6]. For the correction of this labeling bias, `DTA.estimate` needs the number of uridines within each transcript. The *vector* `tnumber` contains the number of uridines for 5976 annotated genes. SGD [1] was used to extract the amount of thymines in the cDNA of each transcript.

```
> head(Sc.tnumber)

tA(AGC)K1     Q0144     snR63     snR61      snR4     snR82
       17       163       101        28        68        91
```

As there are cases in which there is no labeling bias, we strongly advise the user always to check and eventually correct for labeling bias if the L/T ratio shows any dependency on the number of uridines in the transcript. If there should be no labeling bias, the bias correction can be omitted via the option `bicor`. Before applying `DTA.estimate`, we excluded genes that are not reliable and could cause problems during the parameter estimation (see [6] for details). For the given yeast data set, the following genes from the SGD database are discarded:

- Genes annotated as dubious or silenced by the SGD are discarded (verified or uncharacterized ORFs are kept).
- Ribosomal protein genes. These genes are generally expressed at levels which are considerably higher than that of the other genes. Therefore, ribosomal protein genes probably do not lie in the linear measurement range of the experimental system and are likely to introduce a bias to our estimation procedure.
- The histogram of the (log-)expression distribution has a pronounced low intensity tail (see Figure 1). For the same reasons as for ribosomal protein genes, genes that have an expression value below a log-intensity of 5 (natural log basis) in at least 4 out of 15 total RNA measurements [6] are cut off.

```
> Totals = Sc.datamat[,which(Sc.phenomat[,"fraction"]=="T")]
> Total = apply(log(Totals),1,median)
> plotsfkt = function(){
+ par(mar = c(5,4,4,2)+0.1+1)
+ par(mai = c(1.1,1.1,1.3,0.7))
+ hist(Total,breaks = seq(0,ceiling(max(Total)),1/4),
+ cex.main=1.5,cex.lab=1.25,cex.axis=1.125,
+ main="Histogram of log(Total)",
+ xlab="gene-wise median of total samples")
+ hist(Total[Total >= 5],breaks = seq(0,ceiling(max(Total)),1/4),
+ col = "#08306B",add = TRUE)
+ }
> DTA.plot.it(filename = "histogram_cut_off",plotsfkt = plotsfkt,
+ saveit = TRUE,notinR = TRUE)
```

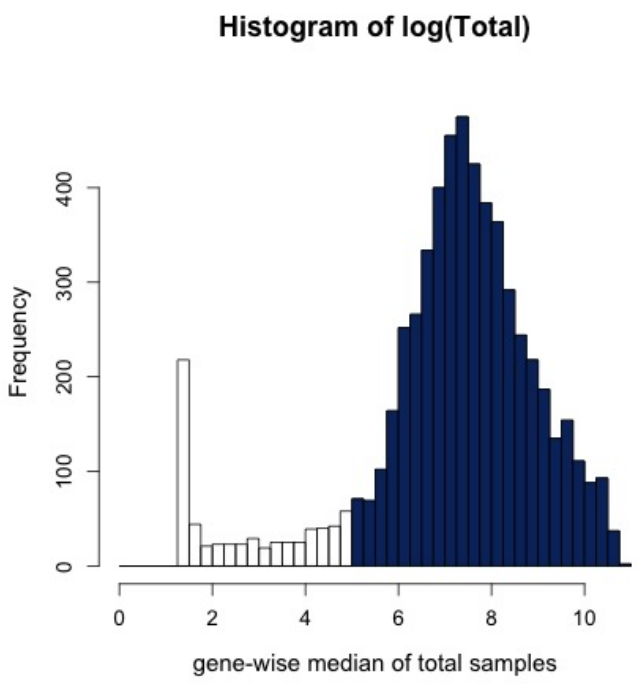

Figure 1: Genes that are below a log-intensity of 5 in total RNA measurements are discarded.

The 4490 genes, that passed the above criteria are included in the vector `reliable`.

```
> head(Sc.reliable)

[1] "YAL068C" "YAL063C" "YAL060W" "YAL059W" "YAL058W" "YAL056W"
```

Now we are ready to start the actual analysis. Besides the above data objects, some additional parameters have to be passed to `DTA.estimate`. `ccl` defines the cell cycle length of the cells and `mRNAs` (optional) is the estimated number of mRNAs in a typical yeast cell [9]. These parameters can be used to scale the synthesis rate to the number of transcript per cell per cell cycle. Furthermore, the following parameters `save.plots`, `notinR`, and `folder` can be used to save quality control and results plots in the current or specified working directory.

```
> res = DTA.estimate(Sc.phenomat,Sc.datamat,Sc.tnumber,
+ ccl = 150,mRNAs = 60000,reliable = Sc.reliable,
+ condition = "real_data",save.plots = TRUE,notinR = TRUE,folder = ".")
```

`DTA.estimate` automatically generates a series of quality control plots. Figure 2 shows the plane defined by the equation $T \sim U + L$ in the 3-dimensional Euclidean space. As all variables T, U and L have measurement errors, we perform a total least squares regression (`tls`) of T versus L and U, which accounts for a Gaussian error in the dependent variable (T) and, in contrast to ordinary linear regression, also in the independent variables (L and U). The total least squares regression minimizes the orthogonal distance of the datapoints to the inferred plane as opposed to a linear regression, which minimizes the distance of T to the inferred linear function of L and U. We use a robust version of total least squares regression. After the first run, data points with the 5% largest residues are removed to avoid the potentially detrimental influence of outlier values on the parameter estimation process. Standard linear regression can also be performed by setting the parameter `ratiomethod` to "lm" instead of "tls" though it is not considered suited as opposed to total least squares for this kind of data.

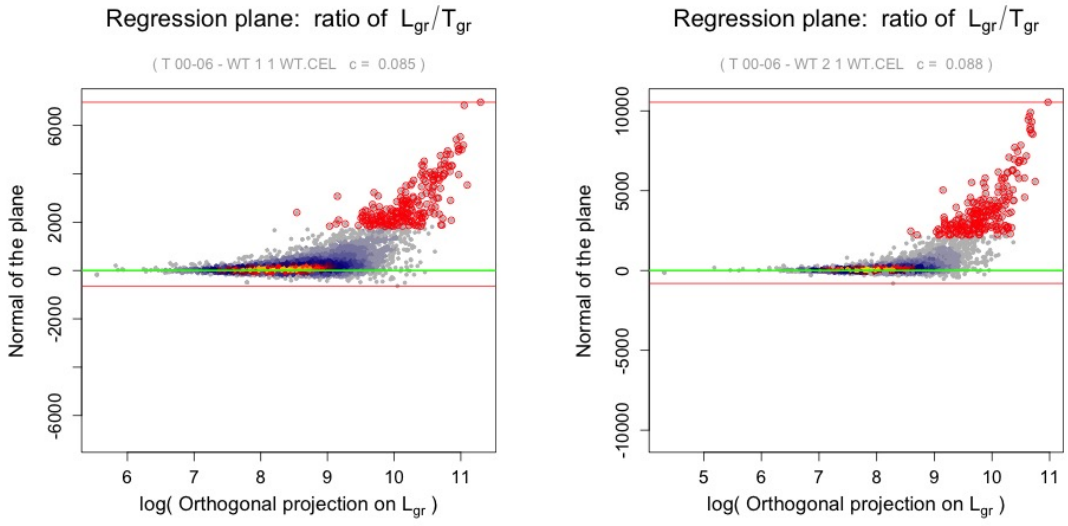

Figure 2: Two rounds of total least squares regression (`tls`). The resulting plane is shown exactly from the side and is colored green. The x-axis of those plots is chosen as the orthogonal projection on L in logarithmic scale. The y-axis is the normal of the plane. The second round of `tls` is performed without the 5% largest residues of the first round, depicted in red. Red lines indicate maximal residues.

Figure 3 shows the labeling bias, i.e. there is a dependency between the log-ratio of L and T and the number of uridines within a transcript as expected for a labeling efficiency below 100%. If labeled uridines were incorporated evenly among transcripts, all labeled expression values would be proportional to the total expression values by a common factor (assuming synthesis rates do not depend on transcript length/uridine content per se). Therefore, the points in Figure 3 should be scattered in parallel to the y-axis, which is clearly not the case. As we will see later, ignoring the labeling bias correction can lead to wrong estimates (see also section "Omission of labeling bias correction can lead to skewed estimates"). The labeling bias can also be used to estimate the ratio of L to T. This is achieved by setting the parameter `ratiomethod` to "bias". The ratio of L to T is then obtained by the fact that the same bias can be observed in the labeled fraction L and in the unlabeled fraction U, however characterized by different curvatures of the fitted line. The ratio of L to T can then be gained from the proportion of the curvatures.

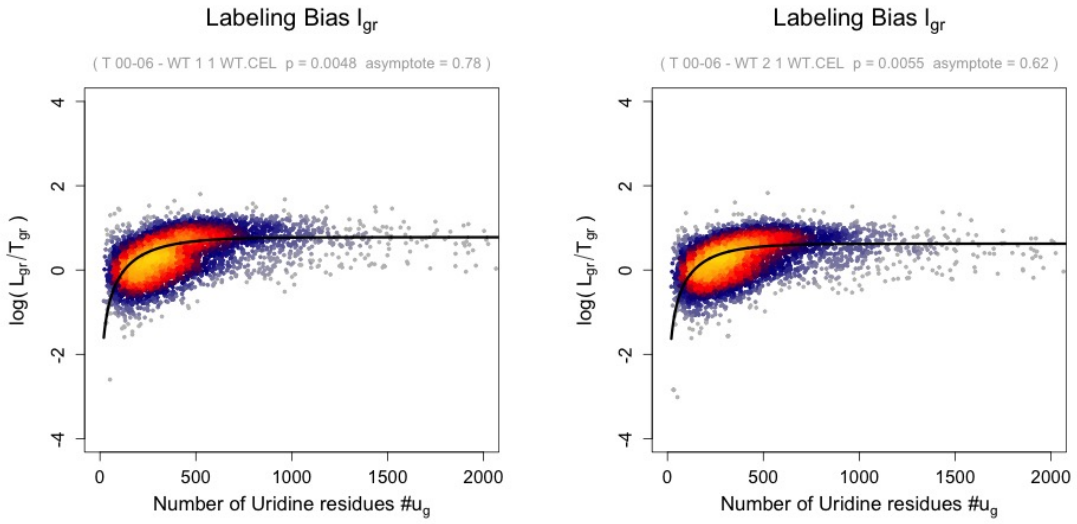

Figure 3: The number of Uridines is plotted versus the log-ratio of L and T for both replicates at labeling time 00-06. The points of the scatterplot are colored according to the (estimated) point density in that region [8]. The labeling bias parameter $p_{estimated} = 0.0048$ and $p_{estimated} = 0.0055$ imply that approximately every 208th resp. 182th Uridine residue is replaced by 4sU.

Figure 4 shows the range assessment of all 1-cL/T values. These are needed for the decay rate estimation. For details on the theoretical background and a thorough description of the formulae, we refer the reader to [6] (supplementary methods). Marginal values of the 1-cL/T distribution can lead to unreasonable (e.g. negative) decay rates. This is strongly dependent on the chosen ratio c of L/T.

Figure 5 shows the data of figure 3 after labeling bias correction. There is no dependency between the log-ratio of L and T and the number of uridines within a transcript any more.

If the labeling bias has been appropriately removed, there should be no correlation between the number of uridines (#U) and the synthesis rate (SR), decay rate (DR) or half-life (HL). Under steady-state condition the LE and TE should both be correlated to SR. The quality control plot shown in Figure 6 depicts, that the labeling bias correction has been successful in our case.

By default, `DTA.estimate` also generates pairwise scatterplots of 1-cL/T values of all replicates for each labeling duration (see Figure 7). The high correlation between the replicates confirms the robustness and reproducibility of the estimation.

By default, `DTA.estimate` also assesses the measurement errors of all available replicates (see Figure 8). The empirical standard deviation can be regularized if the number of replicate measurements is low. This regularized standard deviation can subsequently be used to calculate confidence

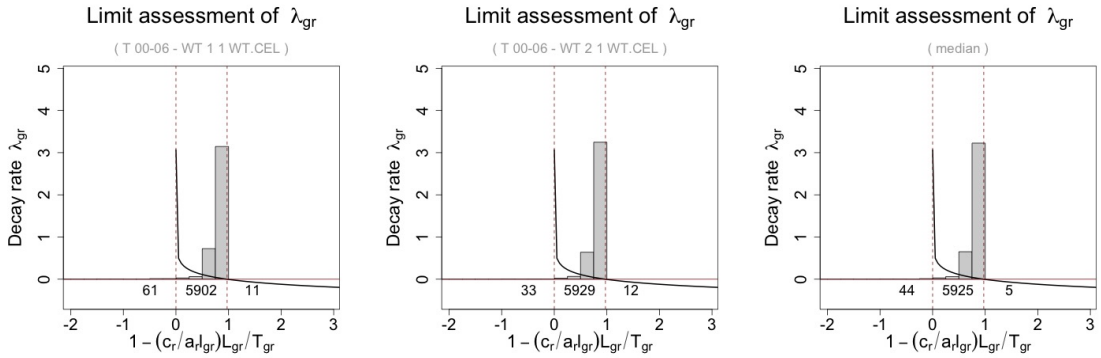

Figure 4: Dependency of 1-cL/T to the resulting decay rate. Reasonable decay rates can only be obtained for 1-cL/T values between the two dashed lines. All others will yield NAs.

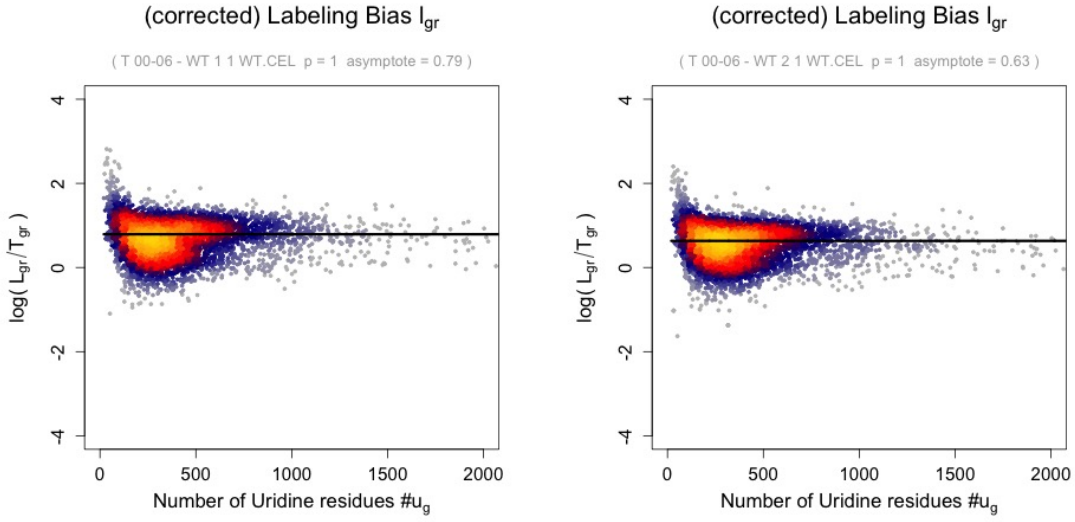

Figure 5: The number of Uridines is plotted versus the log-ratio of L and T for both replicates at labeling time 00-06 after labeling bias correction.

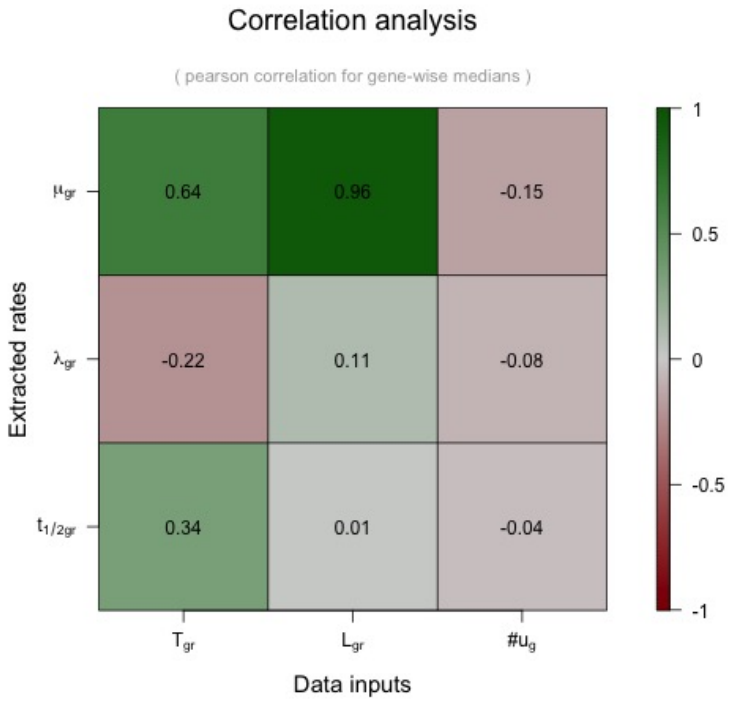

Figure 6: The pairwise correlations between the labeled (LE) expression values, total (TE) expression values, the number of uridines per transcript (#U) and the estimated synthesis rate (SR), decay rate (DR) and half life (HL) is shown after estimation with labeling bias correction.

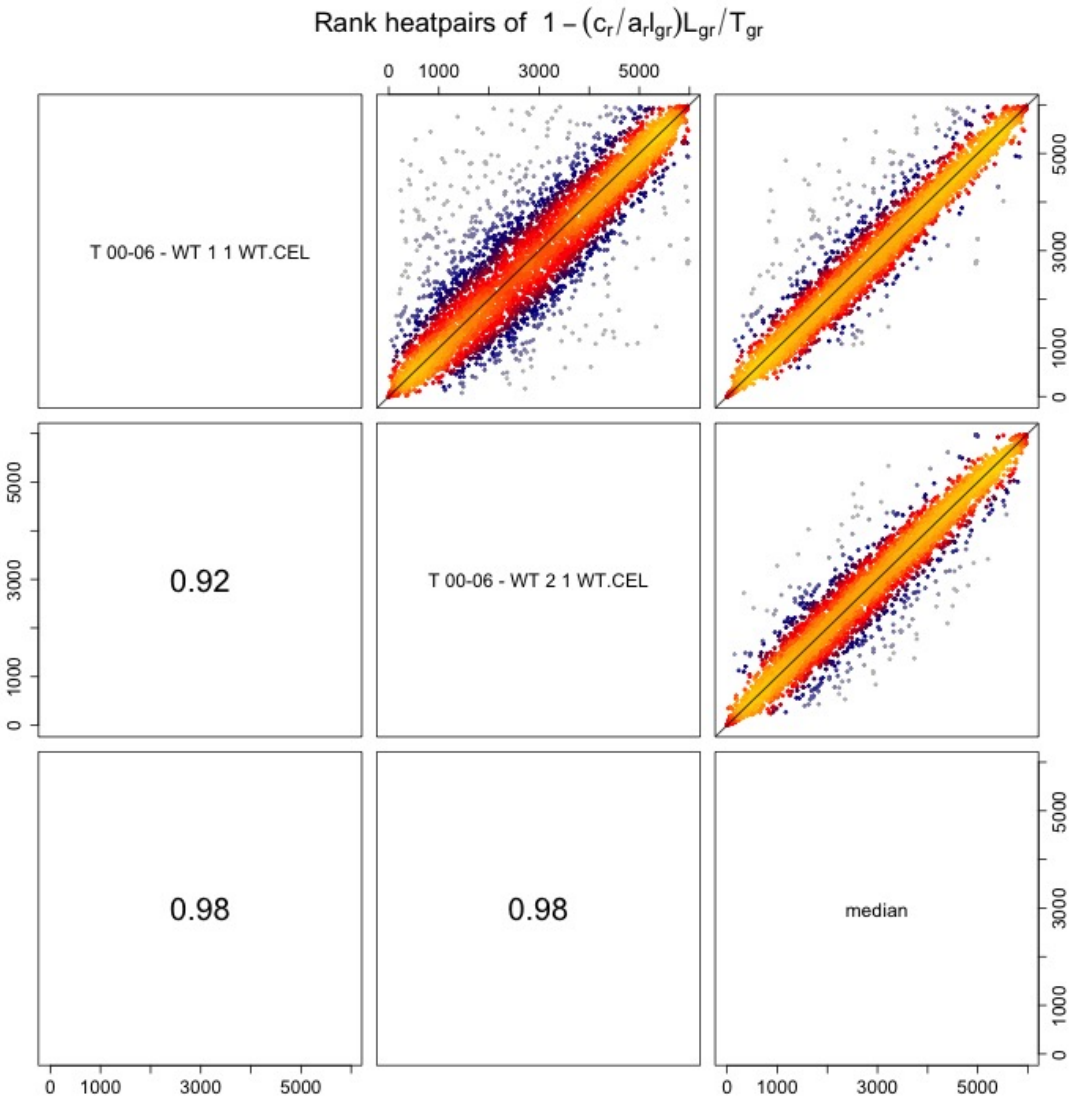

Figure 7: Pairwise heatscatterplots of the ranks of the resp. 1-cL/T value distributions are shown in the upper panel. The lower panel shows the respective spearman correlations.

regions of the extracted rates via resampling.

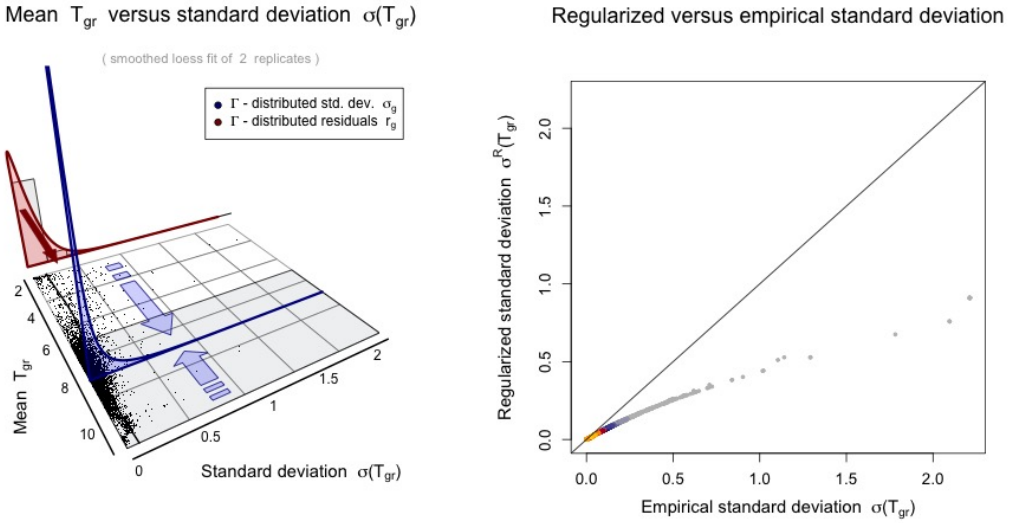

Figure 8: Left: Figure shows the gene-wise mean log(expression value) (x-axis) versus its standard deviation (y-axis), and the corresponding loess curve (black line). The red density (resp. histogram) shows the gamma distributed residuals (loess). Based on the variance of the red distribution the gene-wise standard deviation can be regularized if the number of replicate measurements is low. Right: The empirical versus the regularized standard deviation.

The object `res` - created by `DTA.estimate` - is a list, where each entry contains the estimation results for one of the labeling durations.

```
> names(res)
```

```
[1] "6"  "12"
```

For each labeling duration, median decay rates, synthesis rates and half-lives can be accessed via the entries "dr", "sr" and "hl". For a more detailed explanation of the function output, see the `DTA.estimate` help page.

```
> names(res[["6"]])
```

```
 [1] "triples"          "plabel"           "LtoTratio"        "UtoTratio"
 [5] "LtoUratio"        "correcteddatamat" "LtoTmat"          "LtoT"
 [9] "UtoTmat"          "UtoT"             "LTmat"            "LT"
[13] "drmat"            "hlmat"            "dr"               "hl"
[17] "TEmat"            "TE"               "LEmat"            "LE"
[21] "UEmat"            "UE"               "srmat"            "sr"
[25] "Rsrmat"           "Rsr"              "globaldrmat"      "globaldr"
[29] "TE.log.error"     "LE.log.error"     "UE.log.error"     "TE.confidence"
[33] "LE.confidence"    "LT.confidence"    "LtoT.confidence"  "dr.confidence"
[37] "hl.confidence"    "sr.confidence"
```

A short analysis of the inferred synthesis and decay rates will be given in the following. Figure 9 shows a scatterplot of mRNA half-lives and synthesis rates. We included functional annotations of the genes to highlight specific groups (transcription factors (TF) [5], ribosomal biogenesis genes (RiBi-genes) [4], ribosomal protein genes (Rp-genes) [7], stress response genes (Stress) [3]) in the

scatterplot. We observe that mRNAs with high synthesis rates encode ribosomal protein genes and genes involved in ribosome biogenesis, whereas mRNAs with low synthesis rates are mostly stemmed from genes that are silenced during normal growth, including most TFs.

```
> data(Sc.ribig.ensg)
> data(Sc.rpg.ensg)
> data(Sc.tf.ensg)
> data(Sc.stress.ensg)

> plotsfkt = function(){
+ par(mar = c(5,4,4,2)+0.1+1)
+ par(mai = c(1.1,1.1,1.3,0.7))
+ x=res[["6"]][["hl"]][c(Sc.reliable,Sc.rpg.ensg)]
+ y=res[["6"]][["Rsr"]][c(Sc.reliable,Sc.rpg.ensg)]
+ ellipsescatter(x,y,
+ groups = list("Stress"=Sc.stress.ensg,"RiBi-genes"=Sc.ribig.ensg,
+ "Rp-genes"=Sc.rpg.ensg,"TF"=Sc.tf.ensg),
+ colors = c("darkred","darkgreen","darkblue","grey20"),
+ xlim=c(0,150),ylim=c(0,600),xlab="half-lives",ylab="synthesis rates",
+ cex.main=1.5,cex.lab=1.25,cex.axis=1.125,main="Ellipsescatter")}
> DTA.plot.it(filename = "ellipse_scatter",plotsfkt = plotsfkt,
+ saveit = TRUE,notinR = TRUE)
```

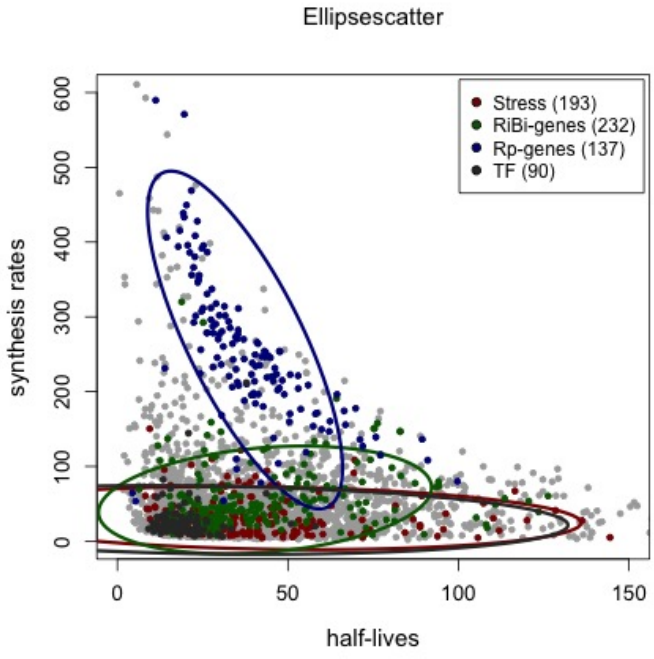

Figure 9: Scatterplot of mRNA half-lives and synthesis rates for exponentially growing yeast cells. Colored points belong to the indicated gene sets (green, ribosomal biogenesis genes; violet, ribosomal protein genes; red, stress genes; dark gray, transcription factors TFs). Assuming Gaussian distributions, ellipses show the 75% regions of highest density for the respective sets. Overall half-lives and synthesis rates are uncorrelated, however some gene groups behave differently.

2.1.1 Omission of labeling bias correction can lead to skewed estimates

To omit the labeling bias correction, the parameter `bicor` has to be set to `FALSE`. The results are displayed in Figure 10.

```
> res.nobias = DTA.estimate(Sc.phenomat,Sc.datamat,Sc.tnumber,
+ ccl = 150, mRNAs = 60000,reliable = Sc.reliable,save.plots = TRUE,
+ notinR = TRUE,folder = ".",bicor = FALSE,condition="no_bias_correction")
```

The obtained estimation results are not independent of the number of uridines. Half-lives show negative correlation to the number of uridines, whereas decay rates correlate positively to the amount of uridines in the mRNA sequence. Therefore we strongly advise the user always to check and eventually correct for labeling bias if the L/T ratio shows any dependency on the number of uridines in the transcript.

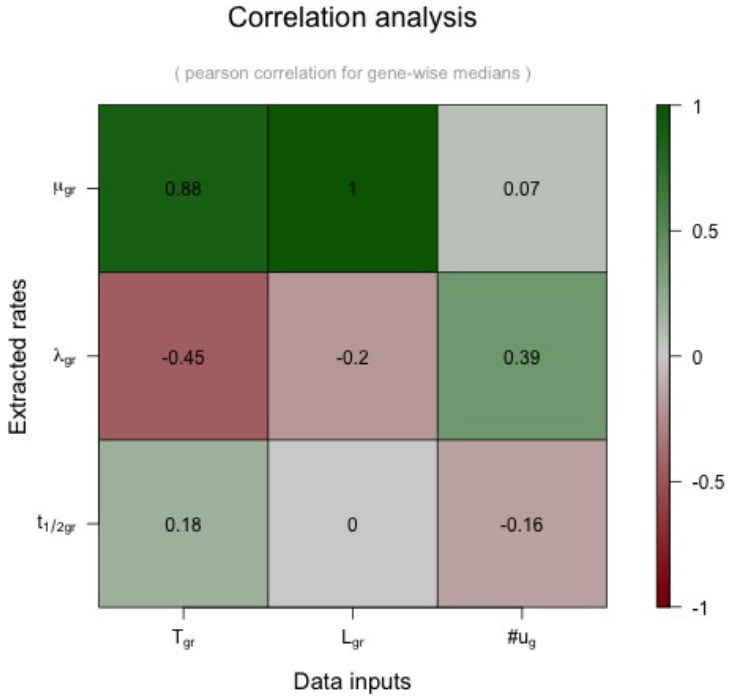

Figure 10: The pairwise correlations between the labeled (LE) expression values, total (TE) expression values, the number of uridines per transcript (#U) and the estimated synthesis rate (SR), decay rate (DR) and half life (HL) is shown after estimation without labeling bias correction.

2.2 Simulating data with `DTA.generate`

The function `DTA.generate` simulates an artificial dataset. This is useful to know the limits of the method, assess minimum data quality requirements, e.g. the optimal number of replicates and labeling durations. If not passed to `DTA.generate`, the decay rate and total expression are randomly generated F resp. normal distributions to match parameter specifications like the median half-life `mediantime` and the number of genes `nrgenes`. The complete set of artificially generated data can be provided with noise via `sdnoise`.
An additional feature of the function `DTA.generate` allows the user to simulate scenarios in which parts of the unlabeled fraction end up in the labeled fraction (e.g. unspecific binding of unlabeled

RNAs to the beads used in the experimental setup to separate the fractions L and U) and vice versa. For a more detailed explanation of the functionality of `DTA.generate`, see it's help page. For this example, two replicates for two labeling durations are generated. The median half-life is set to 12 minutes and the cell cycle length to 150 minutes (by default).

```
> sim.object = DTA.generate(timepoints=rep(c(6,12),2))
```

The output of the function is a list, containing a `phenomat` and `datamat` object. These are formatted as described for the real data in the previous chapter. Furthermore, all parameters used for the simulation are exported into the result *list*.

```
> names(sim.object)
```

```
 [1] "phenomat"       "datamat"        "tnumber"        "ccl"
 [5] "truecomplete"   "truelambdas"    "truemus"        "truehalflives"
 [9] "trueplabel"     "truear"         "truebr"         "truecr"
[13] "truecrbyar"     "truecrbybr"     "truebrbyar"     "trueLasymptote"
[17] "trueUasymptote"
```

`DTA.estimate` can then be used to infer the synthesis and decay rate estimates. Figure 11 shows a comparison between the "true" decay rates/half-lives/synthesis rates and the decay rates/half-lives/synthesis rates estimated by `DTA.estimate` in an absolute manner, on the basis of their ranks and a histogram of their log-quotients.

```
> res.sim = DTA.estimate(ratiomethod = "bias",save.plots = TRUE,
+ notinR = TRUE,folder = ".",simulation = TRUE,sim.object = sim.object,
+ condition = "simulation")
```

Once again we can omit the labeling bias correction. The disastrous results are shown in Figure 12.

```
> res.sim = DTA.estimate(ratiomethod = "bias",save.plots = TRUE,
+ notinR = TRUE,folder = ".",simulation = TRUE,sim.object = sim.object,
+ bicor = FALSE,condition = "simulation_no_bias_correction")
```

For further details on the simulation procedure, see [6] (supplementary methods).

3 Estimation of synthesis and decay rates under dynamic conditions (`DTA.dynamic.estimate`)

In order to use DTA for a time-resolved analysis of a set stimuli, our steady state approach has to be adapted. The main reason for this is that mRNA levels can no longer be assumed constant, i.e., synthesis and degradation are not necessarily in a dynamic equilibrium.

3.1 `DTA.dynamic.estimate` for real timecourse data in yeast

4sU was added at 0, 6, 12, 18, 24, and 30 min after the addition of sodium chloride to yeast cells to a concentration of 0.8 M. Cells were harvested after the labeling time of 6 min. Total and labeled mRNA was purified and analyzed to yield expression profiles for each time window ("0-6"; "6-12"; "12-18"; "18-24"; "24-30"; "30-36"). As for `DTA.estimate`, we need to provide the usual R objects for `DTA.dynamic.estimate`. Except for the `phenomat`, their underlying structure resembles that of the steady state case. In the dynamic case `phenomat` is provided with an additional column giving the sequence of the timecourse experiment.

```
> data(Miller2011dynamic)
```

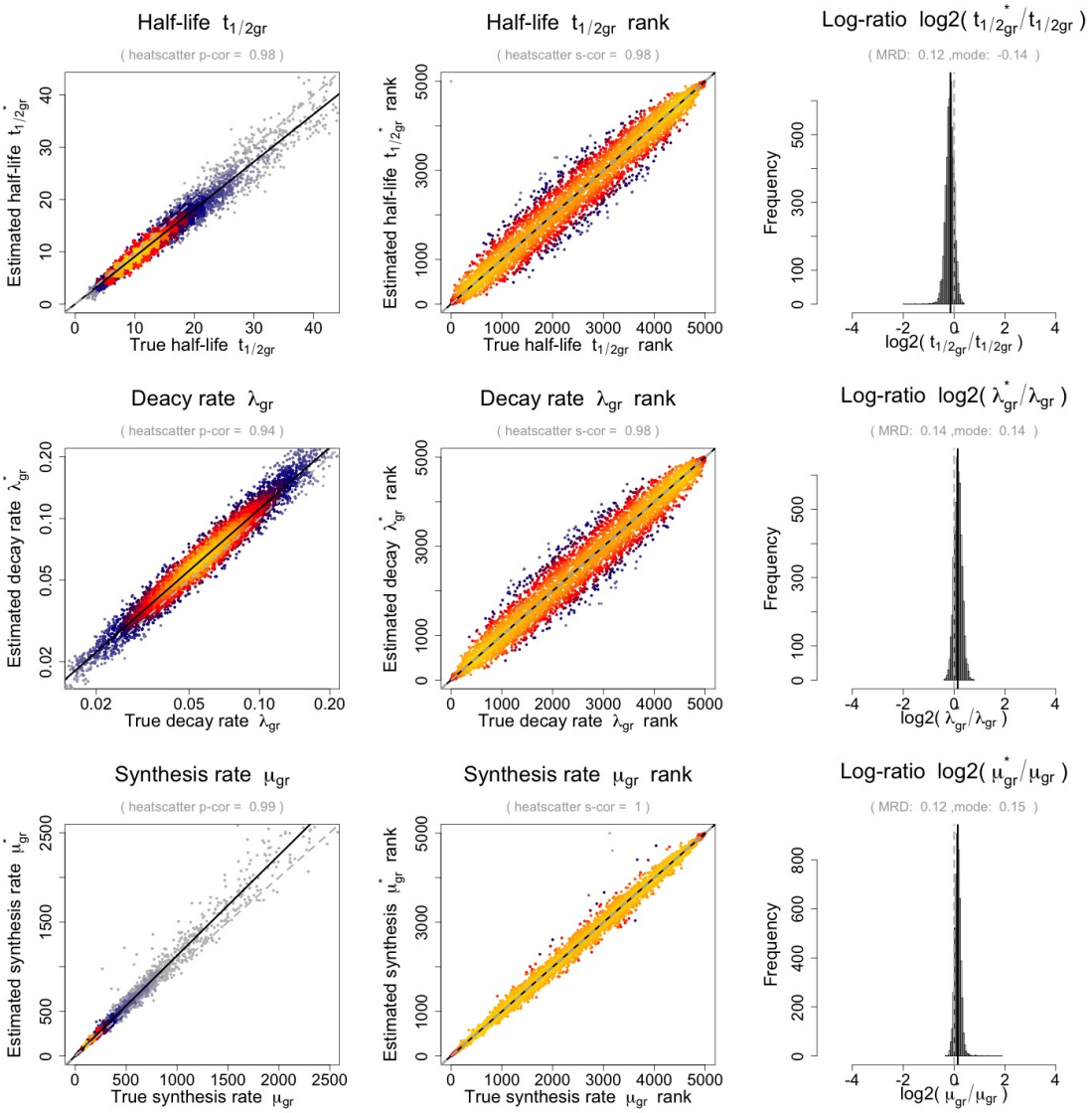

Figure 11: Comparison between "true" (= simulated) and estimated parameters in an absolute manner and on the basis of their ranks: synthesis rates, decay rates and half-lives. The rightmost plots shows the log-ratio of the estimated vs. the true parameters in a histogram. The mode is the maximum of the corresponding density indicated by the blue line. As a measure for a systematic deviation from zero (depicted in green) we calculated the MRD (mean relative deviation). The MRD is defined by mean(|estimated parameter - true parameter|/true parameter).

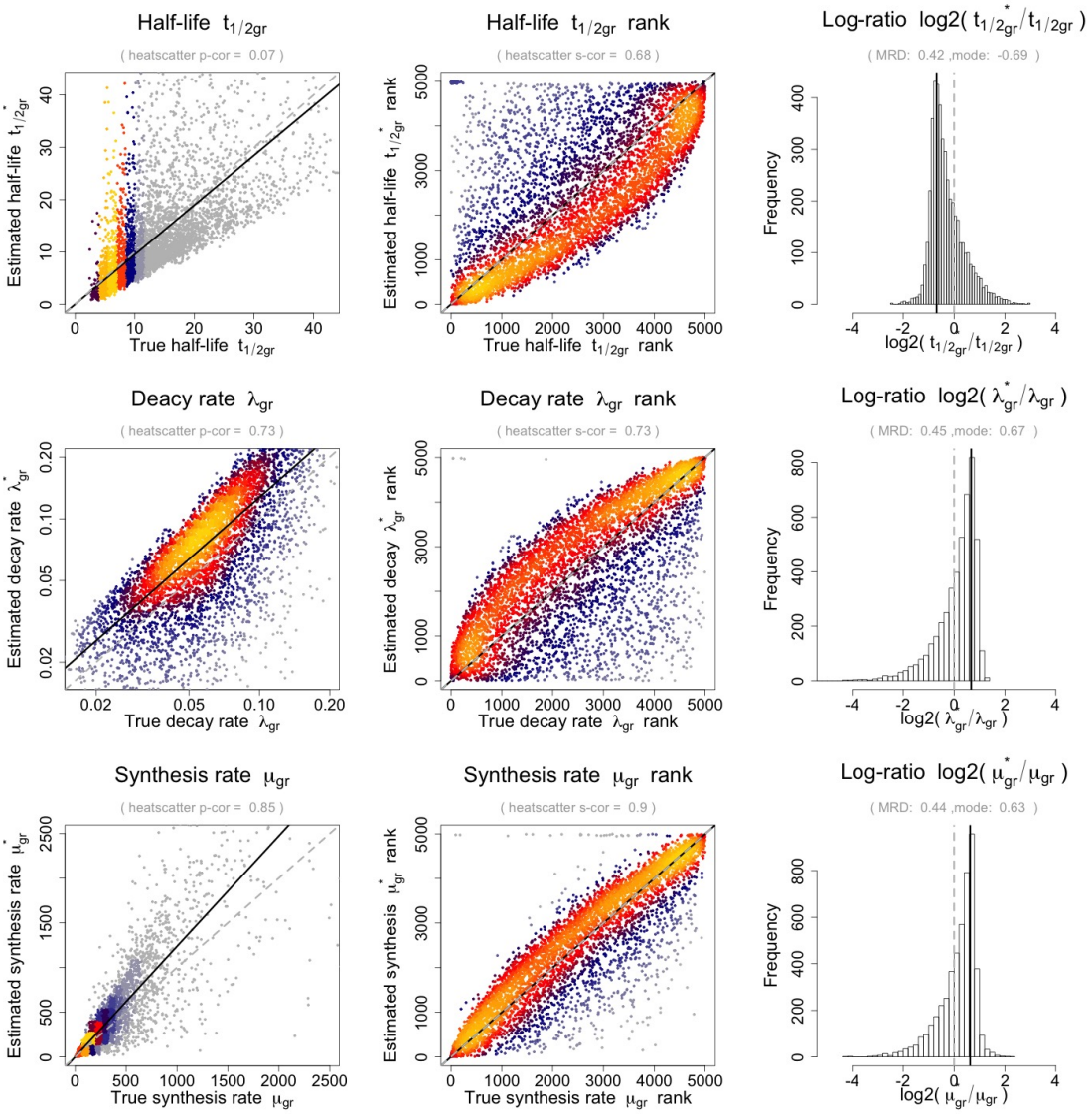

Figure 12: Comparison between "true" (= simulated) and estimated parameters where `DTA.estimate` was performed without labeling bias correction.

For the actual analysis use the following command. There are some additional parameters opposed to the steady state case. For a more detailed explanation of the function output, see the `DTA.dynamic.estimate` help page.

```
> timecourse.res = DTA.dynamic.estimate(Sc.phenomat.dynamic,Sc.datamat.dynamic,
+ Sc.tnumber,ccl = 150,mRNAs = 60000,reliable = Sc.reliable.dynamic,
+ LtoTratio = rep(0.1,7),save.plots = TRUE,notinR = TRUE,folder = ".",
+ condition = "timecourse")
```

Figure 13 shows the dynamics of synthesis and decay rates in the osmotic stress time series. Each point corresponds to one gene, which is colored according to its affiliation with one of 5 clusters chosen according to the rankgain given for the timepoint defined by `ranktime` (last timepoint by default) compared to the first timepoint.

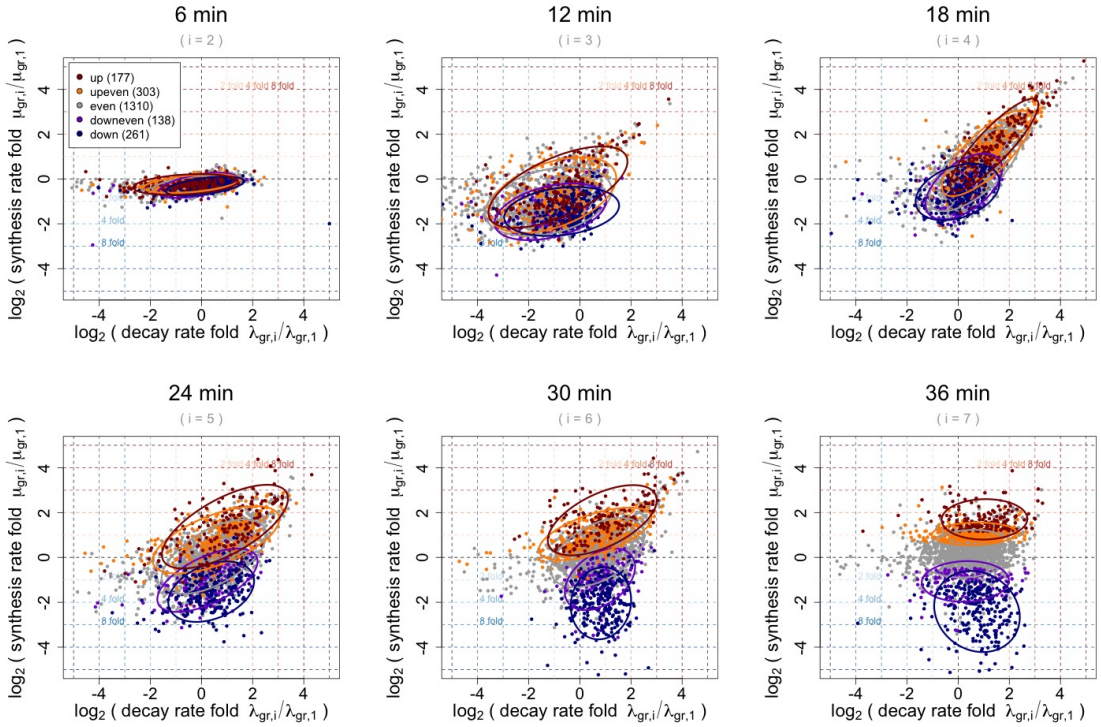

Figure 13: Each plot corresponds to one timepoint. Log decay rate fold versus log synthesis rate fold for the timepoint defined by `ranktime` (last timepoint by default) compared to the first timepoint. Each point corresponds to one gene, which is colored according to its affiliation with one of 5 clusters defined in a normalization-independent manner. Ellipses show the 75% regions of highest density within each cluster, assuming Gaussian distributions. The shape of the ellipses indicates the correlation structure within a cluster.

The object `timecourse.res` - created by `DTA.dynamic.estimate` - is a list, where each entry contains the estimation results for one of the timepoints.

```
> names(timecourse.res)
```

```
[1] "1"           "2"           "3"           "4"           "5"
[6] "6"           "7"           "genecluster"
```

For each timepoint, the output equals the steady state case. For a more detailed explanation of the function output, see the `DTA.dynamic.estimate` help page.

```
> names(timecourse.res[["1"]])
```

```
 [1] "triples"          "plabel"           "LtoTratio"        "UtoTratio"
 [5] "LtoUratio"        "correcteddatamat" "LtoTmat"          "LtoT"
 [9] "UtoTmat"          "UtoT"             "LTmat"            "LT"
[13] "drmat"            "hlmat"            "dr"               "hl"
[17] "TEmat"            "TE"               "LEmat"            "LE"
[21] "UEmat"            "UE"               "srmat"            "sr"
[25] "Rsrmat"           "Rsr"              "globaldrmat"      "globaldr"
[29] "TE.log.error"     "LE.log.error"     "UE.log.error"     "TE.confidence"
[33] "LE.confidence"    "LT.confidence"    "LtoT.confidence"  "dr.confidence"
[37] "hl.confidence"    "sr.confidence"
```

3.2 Simulating data with `DTA.dynamic.generate`

The function `DTA.dynamic.generate` simulates an artificial timecourse dataset. It needs to be provided with *matrices* `mu.values.mat`, `lambda.values.mat`, `mu.breaks.mat` and `lambda.breaks.mat`. The first two give the values of "true" synthesis and decay rates, the last two give their respective breaks. The complete set of artificially generated data can be provided with noise via `sdnoise`. For a more detailed explanation of the functionality of `DTA.dynamic.generate`, see it's help page. For this example, 6 timepoints of 6 min labeling (respectively) are generated.

```
> nrgenes = 5000
> truesynthesisrates = rf(nrgenes,5,5)*18
> steady = rep(1,nrgenes)
> shock = 1/pmax(rnorm(nrgenes,mean = 8,sd = 4),1)
> induction = pmax(rnorm(nrgenes,mean = 8,sd = 4),1)
> changes.mat = cbind(steady,shock,shock*induction)
> mu.values.mat = changes.mat*truesynthesisrates
> mu.breaks.mat = cbind(rep(12,nrgenes),rep(18,nrgenes))
> truehalflives = rf(nrgenes,15,15)*12
> truelambdas = log(2)/truehalflives
> changes.mat = cbind(steady,shock,shock*induction,steady)
> lambda.values.mat = changes.mat*truelambdas
> lambda.breaks.mat = cbind(rep(12,nrgenes),rep(18,nrgenes),rep(27,nrgenes))

> timecourse.sim.object = DTA.dynamic.generate(duration = 36,lab.duration = 6,
+ nrgenes = nrgenes,mu.values.mat = mu.values.mat,mu.breaks.mat = mu.breaks.mat,
+ lambda.values.mat = lambda.values.mat,lambda.breaks.mat = lambda.breaks.mat)
```

The output of the function is a list, containing a `phenomat` and `datamat` object. These are formatted as described in the previous chapter. Furthermore, all parameters used for the simulation are exported into the result *list*.

```
> names(timecourse.sim.object)
```

```
 [1] "phenomat"             "datamat"              "tnumber"
 [4] "truemus"              "truemusaveraged"      "truelambdas"
 [7] "truelambdasaveraged"  "truehalflives"        "truehalflivesaveraged"
[10] "trueplabel"           "truear"               "truebr"
[13] "truecr"               "truecrbyar"           "truecrbybr"
[16] "truebrbyar"           "trueLasymptote"       "trueUasymptote"
```

`DTA.dynamic.estimate` can then be used to infer the synthesis and decay rate estimates. Figure 14 shows a comparison between the "true" decay rates/half-lives/synthesis rates and the decay rates/half-lives/synthesis rates of the time window "12-18" estimated by `DTA.dynamic.estimate` in an absolute manner, on the basis of their ranks and a histogram of their log-quotients.

```
> timecourse.res.sim = DTA.dynamic.estimate(save.plots = TRUE,notinR = TRUE,
+ simulation = TRUE,sim.object = timecourse.sim.object,ratiomethod = "tls",
+ folder = ".",condition = "simulation_timecourse",check = FALSE)
```

4 Example estimations for Mouse and Human data

The package *DTA* is broadly applicable to virtually every organism. In particular, it provides vectors giving the amount of thymines in the cDNA (uridine residues in RNA) of each transcript for each Ensembl transcript ID of a selection of often used model organisms: Saccharomyces Cerevisiae (`Sc.tnumber`), Schizosaccharomyces Pombe (`Sp.tnumber`), Drosophila Melanogaster (`Dm.tnumber`), Mus Musculus (`Mm.tnumber`), and Homo Sapiens (`Hs.tnumber`). These vectors are needed for the assessment and correction of the labeling bias. Further we demonstrate the functionality of the package *DTA* in two example datasets from [2].

```
> data(Doelken2008)
```

The first dataset is a DTA experiment with NIH-3T3 cells (Mus Musculus [2]). The *vector* giving the number of uridine residues for each transcript of Mus musculus is given with Ensembl transcript IDs. As the data in `Mm.datamat` is given for each Ensembl gene ID, we need to map the `Mm.tnumber` *vector* to the same IDs with the function `DTA.map.it` provided in the package and the mapping *vector* `Mm.enst2ensg`:

```
> Mm.tnumber = DTA.map.it(Mm.tnumber,Mm.enst2ensg)
```

Now that we have given the `Mm.tnumber` *vector* with the same identifiers as the `Mm.datamat`, we can apply the `DTA.estimate` function:

```
> res = DTA.estimate(Mm.phenomat,Mm.datamat,Mm.tnumber,
+ ccl = 24*60,reliable = Mm.reliable,ratiomethod = "tls")
```

The second dataset is a DTA experiment with B-cells (BL41, Homo Sapiens [2]). Analogously to the processing steps above, we proceed as follows:

```
> Hs.tnumber = DTA.map.it(Hs.tnumber,Hs.enst2ensg)
```

Now that we have given the `Hs.tnumber` *vector* with the same identifiers as the `Hs.datamat`, we can apply the `DTA.estimate` function:

```
> res = DTA.estimate(Hs.phenomat,Hs.datamat,Hs.tnumber,
+ ccl = 50*60,reliable = Hs.reliable,ratiomethod = "tls",
+ bicor = FALSE)
```

5 Concluding Remarks

The package *DTA* implements methods for Dynamic Transcriptome Analysis (DTA). RNA synthesis and decay rate estimates can be inferred from pre-processed microarray or RNAseq experiments. The procedures generate a series of valuable quality control plots to assess the reliability and the error of the data and the quality and confidence of the resulting DR/SR estimates. Functions for data simulation enable the user to test the methods on synthetic data sets.
DTA builds upon the R package *LSD* [8] which provides the ability to depict it's outcome in a superior manner. Data can be illustrated in a plethora of variations to unveil it's underlying structure.

This vignette was generated using the following package versions:

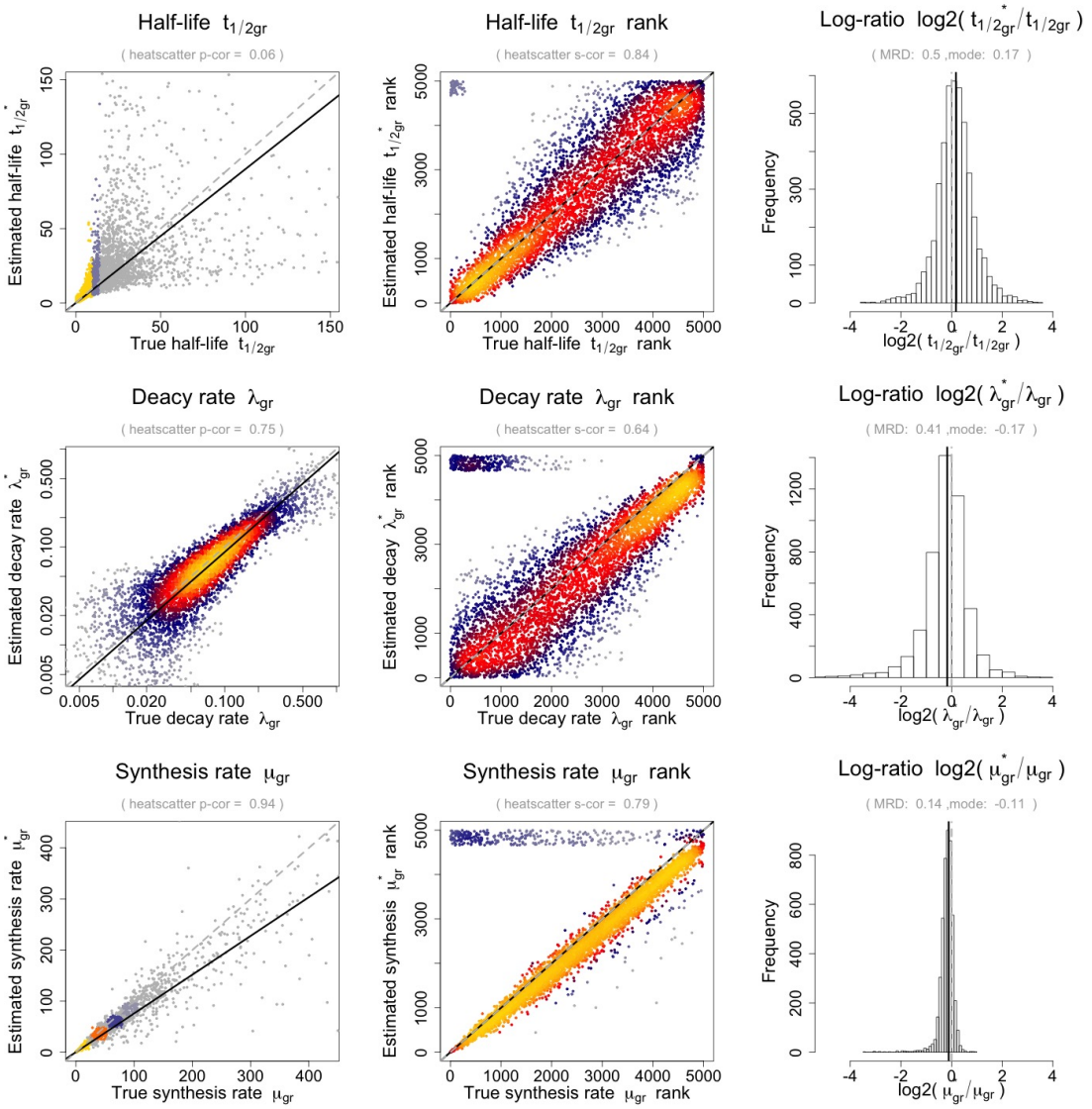

Figure 14: Comparison between "true" (= simulated) and estimated parameters in an absolute manner and on the basis of their ranks: synthesis rates, decay rates and half-lives of the time window "12-18". The rightmost plots shows the log-ratio of the estimated vs. the true parameters in a histogram. The mode is the maximum of the corresponding density indicated by the blue line. As a measure for a systematic deviation from zero (depicted in green) we calculated the MRD (mean relative deviation). The MRD is defined by mean(|estimated parameter - true parameter|/true parameter).

- R version 3.4.0 (2017-04-21), `x86_64-apple-darwin15.6.0`
- Locale: `C/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8`
- Running under: `OS X El Capitan 10.11.6`
- Matrix products: default
- BLAS: `/Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRblas.0.dylib`
- LAPACK: `/Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRlapack.dylib`
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Other packages: DTA 2.22.0, LSD 3.0
- Loaded via a namespace (and not attached): compiler 3.4.0, scatterplot3d 0.3-40, tools 3.4.0

Acknowledgments

We thank Patrick Cramer, Kerstin Maier, Mai Sun, Daniel Schulz and Dietmar Martin (Gene Center Munich) for stimulating discussions and great experimental work. AT, BS, BZ and SD was supported by the LMU excellent guest professorship "Computational Biochemistry", and by the SFB646 grant from the Deutsche Forschungsgemeinschaft.

References

[1] J. M. Cherry, C. Adler, C. Ball, S. A. Chervitz, S. S. Dwight, E. T. Hester, Y. Jia, G. Juvik, T. Roe, M. Schroeder, S. Weng, and D. Botstein. Sgd: Saccharomyces genome database. *Nucleic Acids Res*, 26(1):73–79, 1998.

[2] L. Doelken, Z. Ruzsics, B. Raedle, C. C. Friedel, R. Zimmer, J. Mages, R. Hoffmann, P. Dickinson, T. Forster, P. Ghazal, and U. H. Koszinowski. High-resolution gene expression profiling for simultaneous kinetic parameter analysis of rna synthesis and decay. *RNA*, 14(9):1959–1972, 2008.

[3] J. Ihmels, G. Friedlander, S. Bergmann, O. Sarig, Y. Ziv, and N. Barkai. Revealing modular organization in the yeast transcriptional network. *Nature genetics*, 31(4):370–377, August 2002.

[4] P. Jorgensen, I. RupeÅ¡, J. R. Sharom, L. Schneper, J. R. Broach, and M. Tyers. A dynamic transcriptional network communicates growth potential to ribosome synthesis and critical cell size. *Genes & Development*, 18(20):2491–2505, October 2004.

[5] K. D. MacIsaac, T. Wang, D. B. Gordon, D. K. Gifford, G. D. Stormo, and E. Fraenkel. An improved map of conserved regulatory sites for saccharomyces cerevisiae. *BMC Bioinformatics*, 7:113, 2006.

[6] C. Miller, B. Schwalb, K. Maier, D. Schulz, S. Duemcke, B. Zacher, A. Mayer, J. Sydow, L. Marcinowski, L. Dolken, D. E. Martin, A. Tresch, and P. Cramer. Dynamic transcriptome analysis measures rates of mrna synthesis and decay in yeast. *Mol Syst Biol*, 7:458, 2011.

[7] A. Nakao, M. Yoshihama, and N. Kenmochi. RPG: the Ribosomal Protein Gene database. *Nucleic acids research*, 32(Database issue), January 2004.

[8] B. Schwalb, A. Tresch, and R. Francois. LSD Lots of Superior Depictions. *The Comprehensive R Archive Network*, 2011.

[9] D. Zenklusen, D. R. Larson, and R. H. Singer. Single-rna counting reveals alternative modes of gene expression in yeast. *Nat Struct Mol Biol*, 15(12):1263–1271, 2008.