`easyDifferentialGeneCoexpression`, a handy bioinformatics tool to easily perform differential gene coexpression

13 February 2022

Summary

Gene expression is a way to measure the activity of genes in a structured experiment: the more intense the gene expression is, the more active the gene is in that sample (Chowdhury, Bhattacharyya, and Kalita 2019). Differential gene coexpression is a technique that indicates the pairs of genes that have different expression trends in samples having two different conditions, which then result in different correlation coefficients (Zheng et al. 2014).

If two genes have very different gene expression trends in data samples of patients with breast cancer and in data samples of healthy controls, it is possible that both those genes (or at least one of them) might have a significant role in breast cancer development and/or prognosis (Gov and Arga 2017; Choi et al. 2005).

In this short paper, we present `easyDifferentialGeneExpression`, an R package which handily computes the differential gene expression between genes in a specific dataset, and returns the list of significant different pairs of genes, if found. Our software is available as a R library, as a Perl module that can be used in any standard terminal shell, and as a repository on GitHub.

Statement of need

Several R package for differential gene expression already exist (`diffcoexp` (Wei, Amberkar, and Hide 2021; Yang et al. 2013), `decode` (Lui et al. 2015) and `dcanr` (Bhuva et al. 2019)) on Bioconductor (Huber et al. 2015), but they all have limitations. First, they provide results measured with multiple coefficients, which can be an asset for experienced researchers, but can also be confusing for beginners and unexperienced users.

The `diffcoexp()` function of the `diffcoexp` (Wei, Amberkar, and Hide 2021) library, for example, returns the pairs of differentially coexpressed genes ranked by difference between correlation coefficients under the second condition and the first condition (`cor.diff`), p-value under null hypothesis that difference between two correlation coefficients under two conditions equals to zero using Fisher's r-to-Z transformation (`p.diffcor`), and adjusted p-value under null hypothesis that difference between two correlation coefficients under two conditions equals to zero using Fisher's r-to-Z transformation (`q.diffcor`). These three coefficients have different meanings and can generate three different rankings. Instead, our `easyDifferentialGeneExpression` package, that we built right on `diffcoexp`, generates a final ranking of pairs of significantly expressed genes only through the p-value difference ranking, which we believe being the most informative coefficent and ranking.

Additionally, our `easyDifferentialGeneExpression` package returns a list of significantly coexpressed gene pairs only if their p-values are stricly lower than the 0.005 significance threshold, as suggested by Benjamin et al. (2018). To avoid p-hacking (Head et al. 2015), the users cannot choose their prefered significance threshold. By using this 5×10^{-3} threshold, in fact, users can rest assured that any pair of coexpressed genes is significant enough to be reliable, avoiding insignificant results that could lead to unimportant discoveries (J. P. Ioannidis 2005).

Example

To install `easyDifferentialGeneExpression` from CRAN, in an R environment:

```
install.packages("easyDifferentialGeneExpression")
```

To install `easyDifferentialGeneExpression` from GitHub:

```
git clone https://github.com/davidechicco/easyDifferentialGeneCoexpression.git
```

To install `easyDifferentialGeneExpression` from CPAN, on a Linux operating system:

```
cpanm App::easyDifferentialGeneExpression
```

Please notice that in a Linux Ubuntu system the user might have to run the last command in the `sudo` mode.

To use `easyDifferentialGeneExpression` in an R environment:

```
## Load the library
library("easyDifferentialGeneExpression")

## List of probesets of the genes for which
```

```
## to compute the differential gene expression
probesetList <- c("200738_s_at", "217356_s_at",
"206686_at", "226452_at", "223172_s_at", "223193_x_at",
"224314_s_at", "230630_at", "202022_at")

## Function parameters
verboseFlag <- TRUE
datasetGEOcode <- "GSE16237"
conditionFeatureName <- "outcome of the patient:ch1"
firstConditionName <-  "Died of disease"
secondConditionName <- "Alive"

## Function call
easyDifferentialGeneCoexpression(probesetList, datasetGEOcode, conditionFeatureName,
firstConditionName, secondConditionName, verboseFlag)
```

The output of the call is the following result:

```
Significant top coexpresseed pairs of genes based
on p-value difference (p.diffcor < 0.005):

                        geneSymbolLeft geneSymbolRight    p.diffcor
206686_at,223172_s_at             PDK1           MTFP1 3.111242e-06
206686_at,223193_x_at             PDK1         FAM162A 1.022005e-05
223193_x_at,223172_s_at        FAM162A           MTFP1 1.132584e-05
217356_s_at,223172_s_at           PGK1           MTFP1 3.917640e-05
206686_at,226452_at               PDK1            PDK1 1.956924e-04
217356_s_at,223193_x_at           PGK1         FAM162A 7.650626e-04
217356_s_at,226452_at             PGK1            PDK1 1.132578e-03
223172_s_at,226452_at            MTFP1            PDK1 2.243446e-03
200738_s_at,226452_at             PGK1            PDK1 2.506663e-03
206686_at,217356_s_at             PDK1            PGK1 4.742024e-03
```

In this example, we computed the differential gene coexpression of the genes related to the probesets saved in the `probesetList` variable in the GSE16237 gene expression dataset (Ohtaki et al. 2010) available on Gene Expression Omnibus (GEO). Other prognostic gene expression datasets for this scope can be found through the recently released Perl package `geoCancerPrognosticDatasetsRetriever` (Alameer and Chicco 2022).

`easyDifferentialGeneCoexpression` accepts probesets or gene symbols as input; in the latter case, it associates the input gene symbols to the corresponding microarray platform probesets through the `annotate` (Gentleman 2021) and the `geneExpressionFromGEO` packages (Chicco 2022).

The GSE16237 dataset contains prognostic gene expression samples of 51 patients diagnosed with neuroblastoma. In this cohort, 39 patients died of this childhood cancer and 12 patients survived. This condition is encoded in the `"outcome of`

`the patient:ch1"` variable of the dataset: the `"Died of disease"` label indicates the deceased patients and the `"Alive"` label indicates the survived individuals, of course. In the reported R code example, we specified all these pieces of information in the `datasetGEOcode`, `conditionFeatureName`, `firstConditionName`, and `secondConditionName` variables.

Our package computes the differential gene coexpression through the `easyDifferentialGeneCoexpression()` function, that eventually generates a list of significantly coexpressed pairs of genes, whose p-value is lower than 0.005, as suggested by Benjamin et al. (2018). To avoid p-hacking (Head et al. 2015), this threshold cannot be changed by the user.

In the results, the PDK1-MTFP1, PDK1-FAM162A, and FAM162A-MTFP1 gene pairs result being the most significantly coexpressed gene pairs, suggesting an active role of these three genes (FAM162A, MTFP1, and PDK1) in neuroblastoma. Researchers can use this information to carry on new experiments and scientific analyses investigating the role of these three genes in neuroblastoma.

Acknowledgements

The authors thank the Perl, CPAN, and CRAN community members for their help.

References

Alameer, Abbas, and Davide Chicco. 2022. "geoCancerPrognosticDatasetsRetriever, a bioinformatics tool to easily identify cancer prognostic datasets on Gene Expression Omnibus (GEO)." *Bioinformatics* Online ahead of print: btab852. doi:10.1093/bioinformatics/btab852.

Benjamin, Daniel J, James O Berger, Magnus Johannesson, Brian A Nosek, E-J Wagenmakers, Richard Berk, Kenneth A Bollen, et al. 2018. "Redefine statistical significance." *Nature Human Behaviour* 2 (1). Nature Publishing Group: 6–10. doi:10.1038/s41562-017-0189-z.

Bhuva, Dharmesh D, Joseph Cursons, Gordon K Smyth, and Melissa J Davis. 2019. "Differential co-expression-based detection of conditional relationships in transcriptional data: comparative analysis and application to breast cancer." *Genome Biology* 20 (1). BioMed Central: 1–21. doi:10.1186/s13059-019-1851-8.

Chicco, Davide. 2022. "geneExpressionFromGEO: an R package to facilitate data reading from Gene Expression Omnibus (GEO)." In *Microarray Data Analysis*, 2401:187–94. Methods in Molecular Biology. Springer Protocols.

doi:10.1007/978-1-0716-1839-4_12.

Choi, Jung Kyoon, Ungsik Yu, Ook Joon Yoo, and Sangsoo Kim. 2005. "Differential coexpression analysis using microarray data and its application to human cancer." *Bioinformatics* 21 (24). Oxford University Press: 4348–55. doi:10.1093/bioinformatics/bti722.

Chowdhury, Hussain Ahmed, Dhruba Kumar Bhattacharyya, and Jugal Kumar Kalita. 2019. "Differential co-expression analysis of gene expression: a survey of best practices." *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 17 (4). IEEE: 1154–73. doi:10.1109/TCBB.2019.2893170.

Gentleman, Robert. 2021. "annotate: annotation for microarrays." https://doi.org/doi:10.18129/B9.bioc.annotate URL visited on 2nd January 2022. doi:10.18129/B9.bioc.annotate.

Gov, Esra, and Kazim Yalcin Arga. 2017. "Differential co-expression analysis reveals a novel prognostic gene module in ovarian cancer." *Scientific Reports* 7 (1). Nature Publishing Group: 1–10. doi:10.1038/s41598-017-05298-w.

Head, Megan L, Luke Holman, Rob Lanfear, Andrew T Kahn, and Michael D Jennions. 2015. "The extent and consequences of p-hacking in science." *PLoS Biology* 13 (3). San Francisco, Californiam USA: Public Library of Science: e1002106. doi:10.1371/journal.pbio.1002106.

Huber, Wolfgang, Vincent J Carey, Robert Gentleman, Simon Anders, Marc Carlson, Benilton S Carvalho, Hector Corrada Bravo, et al. 2015. "Orchestrating high-throughput genomic analysis with Bioconductor." *Nature Methods* 12 (2). Springer Science; Business Media LLC: 115–21. doi:10.1038/nmeth.3252.

Ioannidis, John PA. 2005. "Why most published research findings are false." *PLoS Medicine* 2 (8). Public Library of Science: e124. doi:10.1371/journal.pmed.0020124.

Lui, Thomas WH, Nancy BY Tsui, Lawrence WC Chan, Cesar SC Wong, Parco MF Siu, and Benjamin YM Yung. 2015. "DECODE: an integrated differential co-expression and differential expression analysis of gene expression data." *BMC Bioinformatics* 16 (1). BioMed Central: 1–15. doi:10.1186/s12859-015-0582-4.

Ohtaki, Megu, Keiko Otani, Keiko Hiyama, Naomi Kamei, Kenichi Satoh, and Eiso Hiyama. 2010. "A robust method for estimating gene expression states using Affymetrix microarray probe level data." *BMC Bioinformatics* 11 (1). Springer: 1–14. doi:10.1186/1471-2105-11-183.

Wei, Wenbin, Sandeep Amberkar, and Winston Hide. 2021. "`diffcoexp`: Differential Co-expression Analysis. R package version 1.14.0." https://doi.org/doi:10.18129/B9.bioc.diffcoexp URL visited on 5th December 2021. doi:10.18129/B9.bioc.diffcoexp.

Yang, Jing, Hui Yu, Bao-Hong Liu, Zhongming Zhao, Lei Liu, Liang-Xiao Ma, Yi-Xue Li, and Yuan-Yuan Li. 2013. "DCGL v2.0: an R package for

unveiling differential regulation from differential co-expression." *PLoS One* 8 (11). San Francisco, Californiam USA: Public Library of Science: e79729. doi:10.1371/journal.pone.0079729.

Zheng, Chun-Hou, Lin Yuan, Wen Sha, and Zhan-Li Sun. 2014. "Gene differential coexpression analysis based on biweight correlation and maximum clique." In *BMC Bioinformatics*, 15:1–7. 15. BioMed Central. doi:10.1186/1471-2105-15-S15-S3.