BIOINFORMATICS APPLICATIONS NOTE Vol. 27 no. 16 2011, pages 2316–2318
doi:10.1093/bioinformatics/btr379

Systems biology

Advance Access publication June 22, 2011

Metab: an R package for high-throughput analysis of metabolomics data generated by GC-MS

Raphael Aggio[1], Silas Granato Villas-Bôas[1] and Katya Ruggiero[1,2,*]

[1]Centre for Microbial Innovation, School of Biological Sciences and [2]Department of Statistics, The University of Auckland, 3A Symonds Street, Private Bag 92019, Auckland 1142, New Zealand

Associate Editor: Trey Ideker

ABSTRACT

Motivation: The Automated Mass Spectral Deconvolution and Identification System (AMDIS) is freeware extensively applied in metabolomics. However, datasets processed by AMDIS require extensive data correction, filtering and reshaping to create reliable datasets for further downstream analysis. Performed manually, these processes are laborious and extremely time consuming. Furthermore, manual corrections increase the chance of human error and can introduce additional technical variability to the data. Thus, an automated pipeline for curating GC-MS data is urgently needed.

Results: We present the *Metab* R package designed to automate the pipeline for analysis of metabolomics GC-MS datasets processed by AMDIS.

Availability: The Metab package, the AMDIS library and the reference ion library are available at www.metabolomics.auckland.ac.nz/index.php/downloads.

Contact: k.ruggiero@auckland.ac.nz

Received March 17, 2011; revised on June 1, 2011; accepted on June 19, 2011

1 INTRODUCTION

Metabolomics has grown extraordinarily quickly in the last decade and its laboratorial and analytical aspects are now considerably well-advanced (Dunn, 2008). However, the same cannot be said for the data mining and analysis techniques needed to process the raw data generated by Gas Chromatograpy–Mass Spectrometry (GC-MS) in preparation for downstream analyses. While freely available web-based software tools (e.g. MeltDB and metaP) are available, they generally perform only a subset of the necessary data preprocessing steps. These partially processed data must, therefore, be further analyzed using other software(s). Most of these tools do not, however, allow users to create their own pipeline flux.

Automated Mass Spectral Deconvolution and Identification System (AMDIS) is the most commonly used freeware for deconvoluting chromatograms generated by GC-MS and for identifying and quantifying metabolites in biological samples. However, it has three major limitations: selection of different mass fragments for quantification of the same compound, assignment of multiple compounds to the same retention time and results, generated in a layout that does not make the data amenable to further preprocessing and analysis (Smart *et al.*, 2010). Consequently, data processed by AMDIS often require extensive correction, filtering and reorganization. In addition, metabolomics datasets generated by GC-MS need normalization; for example, by internal standard, biomass and/or medium content. Thus, processing datasets generated by freeware such as AMDIS is enormously time consuming and increases the chance of additional technical variability being introduced to the data through human error. Hence, an automated pipeline is urgently needed to perform these processes in a flexible and high-throughput manner.

We present a new R (R Development Core Team, 2008) package, `Metab`, which automates the data preprocessing steps described above. Based on functions provided by the XCMS (Smith *et al.* 2006), an R package specifically developed for processing LC/MS-based data, Metab also reports and processes mass fragment profiles of biological samples; an approach widely applied in non-targeted metabolomics analyses. Thus, `Metab` considerably speeds up the data mining process, produces considerably more consistent results and allows users to customize their own pipeline fluxes for routine analysis. Furthermore, since it was developed in the R environment, it offers substantial flexibility for creating pipelines that best fit the user's requirements. `Metab` was also developed to enable the use of interactive features, facilitating ease of use of its functionalities by R novices. To the best of our knowledge we know of no other freely available software that comprises all of these features.

1.1 Requirements

`Metab` is available for two operating systems: MacOS (R version 2.11.1) and Windows (R version 2.13.0). `Metab` for MacOS depends on `XCMS` (Smith *et al.* 2006), `tcltk` (Dalgaard, P. 2001) and `R.utils` (Bengtsson, 2010), while for Windows it depends only on `XCMS`. These required packages can be installed from the Bioconductor database (http://www.bioconductor.org/) by typing the following code in the R console:

```
>source("http://bioconductor.org/biocLite.R")
>biocLite(c("xcms","tcltk","R.utils"))
```

1.2 Description

We now describe the usage of the seven functions in `Metab`.

Any argument with default value `"popup"` generates a popup dialog box enabling the user to click-and-point to select desired directories and/or open files. The user can, however, elect to enter a character string containing the path name to a file or directory of his or her choice. Additional details for each function can be accessed

*To whom correspondence should be addressed.

by typing ? immediately followed by the function's name at the command prompt.

- `clean.fix(main.folder = "popup", amdis.report = "popup", ion.lib = "popup", save = TRUE, output = "metab_data")`

`clean.fix` is used to correct, filter and rearrange reports generated by AMDIS. It must be coupled to our in-house AMDIS and reference ion libraries. The reference ion library assigns a specific mass fragment to each metabolite present in the AMDIS library. The AMDIS library and the reference ion library are accessed through the website www.metabolomics.auckland.ac.nz/index.php/downloads. Both the AMDIS and ion libraries can be modified by the user and their folder locations are determined by the values given to the arguments `amdis.report` and `ion.lib`, respectively.

GC-MS analysis generates one data file per sample. To avoid the user having to define which file corresponds to which experimental condition, `clean.fix` requires that files, in AIA format, corresponding to the same condition are stored within their own folder and this is (usually) named after the condition. Thus, there must be one folder per condition. All such folders must then be stored within another (main) folder whose path is defined by a character string supplied to the `main.folder` argument. `clean.fix` loads the GC-MS files into the R session and uses the ion library to recalculate the intensity of each metabolite present in the AMDIS report (see, e.g. `data(amdis.report)`). Finally, it generates a data frame containing the metabolites identified in each sample and their respective intensities (see, e.g. `data(clean.fix.example)`). When `save = TRUE` (default) this data frame is saved to a CSV file in `main.folder` and is named according to the character string supplied to the `output` argument.

- `del.false(data = "popup", true = 0.65, medium.tag = "none", true.medium = 0.68, save = FALSE, folder = "popup", output = "no_false")`

`del.false` is used to exclude compounds that are detected in less than a user-specified proportion of samples since such compounds are considered to be false positives. The argument `true` takes a value from 0.0 to 1.0 indicating the smallest proportion of samples in which each compound must be identified in order for it to be considered a true compound. For example, consider an experiment with six replicates per condition and `true = 0.50`. For each condition, compounds detected in fewer than three replicates will have their intensities replaced by `NA`.

The `medium.tag` argument is required when studying extracellular metabolites (e.g. footprinting) and is used to specify the columns of the input data that correspond to samples from the uncultured medium see data(`clean.fix.example`). Due to the high reproducibility of compound profiles and intensities of samples from uncultured media, often fewer replicates are needed for these than are required for the experimental conditions. Therefore, a different proportion of false positives to `true` may be appropriate and can be specified through `true.medium`.

`del.false` generates a data frame containing only those metabolites present in at least the proportion of samples specified by `true` and `true.medium`. When `save = TRUE` this data frame is saved to a CSV file with filename defined by `output` (e.g. `"AerobicIntra"`) and stored in a folder defined by `folder`.

The arguments `data`, `save`, `folder` and `output` in the following four functions, i.e. `norm.internal`, `norm.medium`, `norm.biomass` and `htest`, follow the same behavior as described in `del.false`.

- `norm.internal(data = "popup", internal.std = "ask", save = FALSE, folder = "popup", output = "norm_int")`

`norm.internal` generates a data frame containing metabolite intensities normalized by a user-nominated internal standard. The `internal.std` argument is used to specify this metabolite. If `internal.std = "ask"` (default) a list of metabolites will be presented to the user.

- `norm.medium(data = "popup", medium = "popup", log.transform = TRUE, save = FALSE, folder = "popup", output = "norm_medium")`

`norm.medium` is applied when performing footprint analysis. It subtracts the average log-intensity of each metabolite identified in the uncultured medium from the log-intensity of the same metabolite identified in each biological sample. `medium` takes a character string indicating which columns of the input data frame correspond to the uncultured medium. The data are log-transformed (default) because intensities often differ by at least one order of magnitude across conditions. Compounds having negative log-intensities after normalization by internal standard indicate cell consumption, while positive intensities indicate cell secretion, of metabolites.

- `norm.biomass(data = "popup", biomass = "popup", save = FALSE, folder = "popup", output = "norm_bio")`

Biological samples may have different biomass content due to the different experimental conditions and/or technical variability. Thus, `norm.biomass` produces a data frame where the intensities of the compounds within a sample are scaled by that sample's biomass. `biomass` defines the biomass of each biological sample (see `data(biomass.example)`).

- `htest(data = "popup", signif.level = 0.05, log.transform = TRUE, save = FALSE, folder = "popup", output = "htest")`

When more than two conditions are under investigation ANOVA is used to test differential metabolite intensities between conditions, otherwise a *t*-test is used. A column of p-values resulting from these is added to the data frame specified through `data.` `htest` generates a data frame comprising only compounds statistically significantly different at the specified `signif.level`. If `log.transform = TRUE` (default), the *t*-test or ANOVA is calculated using the log-transformed input data. Care should be taken since, if `htest` is applied to the data frame generated by norm.medium, the input data might already be on the log-scale. If so, set `log.transform = FALSE`.

An additional analysis commonly performed in metabolomics is called non-targeted analysis. In this case, the researcher works with the profile of mass fragments present in each biological sample instead of the identified metabolites. In order to extract the profile of

mass fragments from the raw data we have developed the following function:

- `raw.peaks(main.folder = "popup", correct.RT = TRUE, method = "loess", save = TRUE, output = "mass_fragments")`

`raw.peaks` produces a data frame containing all the mass fragments detected by the GC-MS, and their intensities, in each biological sample. It requires the same folder structure as per `clean.fix`. The arguments `main.folder` and `save` also behave as per `clean.fix`. When the argument `correct.RT = TRUE` (default) the retention time of all samples is corrected using the method specified by `method` (see the function `rector` in the `XCMS` package for more details). Further analyses can then be performed using any of the above functions. See `data(raw.peaks.example)` for an example of the results obtained from `raw.peaks`.

2 CONCLUSION

We have introduced a new R package, `Metab`, which automates the tasks commonly required for the analysis of metabolomics GC-MS datasets. `Metab` substantially increases throughput by substantially reducing the time normally required to perform these tasks manually and provides protection against human error. Furthermore, our interactive functions allow its usage by users with little knowledge of the R language.

ACKNOWLEDGEMENTS

We thank the contributions of Xavier Duportet, Sonia Carneiro and Farhana Pinu.

Funding: Grants from the Health Research Council of New Zealand (HRC); Foundation for Research, Science and Technology (FRST).

Conflict of Interest: none declared.

REFERENCES

Dalgaard,P. (2001) The R-Tcl/Tk interface. In Hornik,K. and Leisch,F. (eds) *Proceedings of the 2nd International Workshop on Distributed Statistical Computing*, March 15–17, 2001. Technische Universit¨at Wien, Vienna, Austria.

Dunn,W.B. (2008) Current trends and future requirements for the mass spectrometric investigation of microbial, mammalian and plant metabolomes. *Phys. Biol.*, **5**, 11001.

Bengtsson,H. (2010) R.utils: various programming utilities. R package version 1.4.3.

R Development Core Team (2008) *R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing.* Vienna, Austria.

Smart,K.F. *et al.* (2010) Analytical platform for metabolome analysis of microbial cells using methyl chloroformate derivatization followed by gas chromatography-mass spectrometry. *Nat. Protoc.*, **5**, 1709–1729.

Smith,C.A. *et al.* (2006) XCMS: processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification. *Anal. Chem.*, **78**, 779–787.