Working with substrate information in opm

Lea A.I. Vaas
Leibniz Institute DSMZ

Johannes Sikorski
Leibniz Institute DSMZ

Markus Göker
Leibniz Institute DSMZ

Abstract

This is the substrate-information tutorial of **opm** in the version of October 2, 2013.

Keywords: Cell Lines, Metadata, Microbiology, Respiration Kinetics, Pathway, KEGG, **pathview**.

1. Introduction

A detailed description of the OmniLog® Phenotype MicroArray (PM) system, its measuring procedure and data characteristics are found in the vignette "**opm**: An R Package for Analysing OmniLog® Phenotype MicroArray Data" (called "main tutorial" in the following). The description of the methods below presupposes that the user is familiar with the usage of **opm** and has studied the main tutorial as well as the entries of the **opm** manual relevant to her or his research. Especially the concept and structure of the different object-classes should be made clear before starting with this tutorial.

In addition to visual inspection or statistical comparative analyses of Phenotype Microarray data, as described in the main manual, users might be interested in specific information on the substrates used in PM assays. The **opm** package contains a large variety of additional data on PM substrates. Beside methods for assessing these information directly, this tutorial introduces strategies for visualization of the measured PM results by mapping on pathway-maps and introduces analysis methods for modelling and identification of informative and less informative substrates.

2. Preparation

Before starting, install the package **pathview** from Bioconductor (http://bioconductor.org/packages/2.12/bioc/html/pathview.html) and load it together with the **opm** package, into an R session. Please note, that it is important to load **pathview** before **opm**, since otherwise some functions are not visible and the package does not work properly. In this vignette this is catched by the `if`-construct.

```
R> suppressPackageStartupMessages(library("pathview"))
R> if ("package:opm" %in% search())
     detach("package:opm")
R> library("opm")
R> data(vaas_et_al, package = "opmdata")
```

3. Available plate information

Currently substrate layouts of various plates are available within **opm**. An overview about the plate types available in the respective version of **opm** is obtained by entering

```
R> plate_type(full = TRUE)
```

The resulting vector of names does not only include OmniLog® plates; see the manual and the main tutorial for further details. Using other values for `full`, or additional arguments, distinct spelling variants of the plate names can be obtained.

4. Accessing substrate information

The **opm** package contains a number of functions suitable for accessing precomputed information on the substrates within certain wells and entire plates. In the manual and help pages these functions are explained within the family "naming-functions" with according cross-references. One usually would start a search by determining the exact spelling of an internally used name with `find_substrate()`:

```
R> substrates <- find_substrate(c("Glutamine", "Glutamic acid"))
R> substrates
```

The results is a list (of the S3 class "substrate_match") containing character vectors with the results for each query name as values. Surprisingly, nothing was found for "Glutamic acid" but several values for "Glutamine". The default `search` argument is "exact", which is exact (case-sensitive) matching of *substrings* of the names. One might want to use "glob" searching mode:

```
R> substrates <- find_substrate(c("L-Glutamine", "L-Glutamic acid"), "glob")
R> substrates
```

But with so-called wildcards, i.e. "*" for zero to many and "?" for a single arbitrary character the search is more flexible:

```
R> substrates <- find_substrate(c("*L-Glutamine", "*L-Glutamic acid"), "glob")
R> substrates
```

This fetches all terms that end in either query character string, and does so case-insensitively. Advanced users can apply the much more powerful "regex" and "approx" search modes; see the manual for details, entry `?find_substrate`.

Note that **opm** appends a concentration (or just repetition) indicator as a number after a hash sign ("#") to the substrate names wherever necessary. Thus a wildcard at the end of a name might often by the most useful search pattern.

Once the internally used names (which are not guaranteed to be stable between distinct **opm** releases) have been found, information on the substrates can be queried such as their occurrences and positions on plates:

```
R> positions <- find_positions(substrates)
R> positions
```

This yields a nested list containing two-column matrices with plate names in the first and well coordinates in the second column. References to external data resources for each substrate name can be obtained using `substrate_info()`:

```
R> subst.info <- substrate_info(substrates)
R> subst.info
```

By default this yields CAS numbers (http://www.cas.org/content/chemical-substances/faqs), but MeSH names (useful for conducting PubMed queries; see http://www.ncbi.nlm.nih.gov/mesh/) (Coletti and Bleich 2001), ChEBI IDs (Hastings, de Matos, Dekker, Ennis, Harsha, Kale, Muthukrishnan, Owen, Turner, Williams, and Steinbeck 2013), KEGG compound IDs, KEGG drug IDs (Kanehisa, Goto, Furumichi, Tanabe, and Hirakawa 2010) and MetaCyc IDs (Caspi, Altman, Dreher, Fulcher, Subhraveti, Keseler, Kothari, Krummenacker, Latendresse, Mueller, Ong, Paley, Pujar, Shearer, Travers, Weerasinghe, Zhang, and Karp 2012) IDs have also been collected for the majority of the substrates. Using the "browse" argument, full URLs can be created and optionally also directly opened in the default web browser. Using the "download" argument, if KEGG drug or compound IDs have been selected, these can be downloaded from the KEGG server if the **KEGGREST** is available and converted into customized objects. It is possible to nicely display all available information at once:

```
R> subst.info <- substrate_info(substrates, "all")
R> subst.info
```

Another use of `substrate_info()` is to convert substrate names to lower case but protecting name components such as abbreviations or chemical symbols. See the manual for further details, help page `?substrate_info`.

5. Visualisation by integration in pathway maps

In conjunction with other packages, it is possible to visualize PM results directly in pre-existing pathway maps as, for example, from the KEGG database. Those maps are essentially manually drawn pathway maps representing the currently available knowledge about genes, substrates and their connection in pathways. Depending on availability of genome and gene-annotation information within KEGG about a certain organism, individual maps are allocateable (Kanehisa *et al.* 2010).

The mapping itself is simply the introduction of PM-measurement data as colour-coding of nodes (here, representing the substrates) into those maps, as it can be done similarly with several other types of OMICS-data. For details, please see the description on the KEGG-homepage: (http://www.genome.jp/kegg/).

Here we will use the function `pathview` from the package of same name (Luo and Brouwer 2013). This function downloads the user-defined pathway map and subsequently maps and renders the given PM-data into it. Please note, that it has to be loaded *before* loading package **opm** into an R session.

The workflow starts with either an `OPMX` object containing the agregated values, or the result from an opm_mcp analysis. The first step in both cases is to bring the PM-results in a suitable format, which is a named vector created by `annotated`.

```
R> # here provide the annotated vector from vaas_1
R> xx <- annotated(vaas_1)
R> head(xx)
```

```
    <NA>   C00721   C00208   C01083   C00185   C08240
123.4558 248.1809 284.0994 269.7548 180.7536 287.7959
```

The resulting vector basically contains the numeric values (selected parameter estimates or `opm_mcp` results, as explained below) as well as an annotation of the according substrates. With the `what`-argument, passed as eponymous argument to `substrate_info`, the user can indicate which kind of information should be used for the annotation. `annotated` works directly on `OPMS`-objects containing aggregated data for single plates or bundles of plates. However, please note, that the output allows for only one value per substrate. When applying `annotated` to a set of plates, make sure, that only one experimental group is comprised, since the resulting values are averages per well over all plates. Using the `output`-argument, one is able to choose which parameter should be adressed, for example AUC instead of maximum height:

```
R> y <- annotated(vaas_1, output = param_names()[4])
R> head(y)
```

```
     <NA>    C00721    C00208    C01083    C00185    C08240
 8918.137 18391.590 21960.080 18531.180 11831.150 19254.160
```

Further options allow for indication of categorical (`different`) and for the `opm_glht` method, binary outcomes (`smaller`, `larger` or `equal`).

But visualization of the results of an opm_mcp analysis is also possible, which offers more (statistically interesting) opportunities for making sense of the PM data in the context of pathways. The results from an `opm_mcp` procedure are treated with `annotated` as shown before with an `OPMX` object:

```
R> x <- opm_mcp(vaas_4[, , 1:10], output = "mcp", model = ~ Well,
     linfct = c(`Dunnett.A05 (D-Cellobiose)` = 1))
R> (annogl1 <- annotated(x))
```

```
      <NA>     C00721     C00208     C01083     C08240     C00089     C19636     C01613
  1.577317  57.274661  26.661023  34.537328  25.078779  -1.606236  -8.458879  -4.975284
      <NA>
247.470724
attr(,"how")
[1] "numeric"
attr(,"cutoff")
[1] 0
```

Further options allow for indication of categorical (`different`) and for the `opm_glht` method, binary outcomes (`smaller`, `larger` or `equal`).

```
R> head(annogl1 <- annotated(x, output = "numeric"))

     <NA>    C00721    C00208    C01083    C08240    C00089
 1.577317 57.274661 26.661023 34.537328 25.078779 -1.606236

R> head(annogl4 <- annotated(x, output = "equal"))

  <NA> C00721 C00208 C01083 C08240 C00089
  TRUE   TRUE   TRUE   TRUE   TRUE   TRUE
```

This can be used to determine which pathways are actually of interest before proceeding with the visualization of the PM results within these pathways.

ko - KEGG orthology: manually defined ortholog groups (KO entries) for all proteins and functional RNAs that correspond to KEGG pathway nodes, BRITE hierarchy nodes, and KEGG module nodes.

6. Acknowledgements

The integration of missing OmniLog® substrates into ChEBI by the ChEBI staff is gratefully acknowledged.

References

Caspi R, Altman T, Dreher K, Fulcher CA, Subhraveti P, Keseler IM, Kothari A, Krummenacker M, Latendresse M, Mueller LA, Ong Q, Paley S, Pujar A, Shearer AG, Travers M, Weerasinghe D, Zhang P, Karp PD (2012). "The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases." *Nucleic Acids Research*, **40**(D1), D742–D753. doi:10.1093/nar/gkr1014.

Coletti MH, Bleich HL (2001). "Medical Subject Headings Used to Search the Biomedical Literature." *Journal of the American Medical Informatics Association*, **8**(4), 317–323. doi:10.1136/jamia.2001.0080317.

Hastings J, de Matos P, Dekker A, Ennis M, Harsha B, Kale N, Muthukrishnan V, Owen G, Turner S, Williams M, Steinbeck C (2013). "The ChEBI reference database and ontology for biologically relevant chemistry: enhancements for 2013." *Nucleic Acids Research*, **41**(D1), D456–D463. doi:10.1093/nar/gks1146.

Kanehisa M, Goto S, Furumichi M, Tanabe M, Hirakawa M (2010). "KEGG for representation and analysis of molecular networks involving diseases and drugs." *Nucleic Acids Research*, **38**(suppl 1), D355–D360. doi:10.1093/nar/gkp896.

Luo W, Brouwer C (2013). "Pathview: an R/Biocondutor package for pathway-based data integration and visualization." *Bioinformatics*. doi:10.1093/bioinformatics/btt285. URL http://bioinformatics.oxfordjournals.org/content/29/14/1830.full.pdf+html.

Affiliation:

Markus Göker
Leibniz Institute DSMZ – German Collection of Microorganisms and Cell Cultures
Braunschweig

Telephone: +49/531-2616-272
Fax: +49/531-2616-237
E-mail: markus.goeker@dsmz.de
URL: www.dsmz.de