

Working with substrate information in **opm**

Lea A.I. Vaas
Leibniz Institute DSMZ

Johannes Sikorski
Leibniz Institute DSMZ

Markus Göker
Leibniz Institute DSMZ

Abstract

The OmniLog® Phenotype Microarray system is able to monitor simultaneously, on a longitudinal time scale, the phenotypic reaction of single-celled organisms such as bacteria, fungi, and animal cell cultures to up to 2,000 environmental challenges spotted on sets of 96-well microtiter plates. The phenotypic reactions are recorded as respiration kinetics with a shape comparable to growth curves. Tools for storing the curve kinetics, aggregating the curve parameters, recording associated metadata of organisms and experimental settings as well as methods for analysing graphically and statistically these highly complex data sets are increasingly in demand.

The **opm** R package facilitates management, visualization and statistical analysis of Phenotype Microarray data. Raw measurements can be easily input into R, combined with relevant meta-information and accordingly analysed. The kinetics can be aggregated by estimating curve parameters using several methods. Containers of **opm** data can easily be queried for and subset by using the integrated metadata and other information. Using **KEGG**, a database resource for understanding high-level functions and utilities of the biological system, **opm** offers functionality to map and render PM data on relevant pathway graphs in customized manner. All methods are exemplified using real-world data sets that are part of the **opm** R package or are included in the accompanying data package **opmdata**.

This is the tutorial of **opm-substrates** in the version of September 28, 2013.

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

1. Scientific Introduction

A detailed description of the Phenotype MicroArray (PM) system, its measuring procedure and data characteristics can be found in the vignette “**opm**: An R Package for Analysing OmniLog® Phenotype MicroArray Data” (main tutorial). The analysis described here implies that the user is familiar with the functionality of **opm** and has read and understood the main tutorial.

Beside visual inspection of data or statistical comparative analyses of Phenotype Microarray data, users might be interested in mapping the data on pathway maps for biological interpretation of higher-level systemic functions.

This vignette focusses on this mapping and integrative data analysis methods.

In order to apply the functionality for integrative data analysis the user has to provide an OPMA or OPMS object containing the aggregated data or results from a multiple comparison using **opm_mcp**. The corresponding workflow to assemble those objects is described in detail in the main tutorial.

2. Preparation

Before starting, both the **opm** package and **opmdata**, containing example data introduced and explained in detail in the main tutorial, should be loaded into an R session as follows:

```
R> library("opm")
R> data(vaas_et_al, package = "opmdata")
```

here: prepare an OPMS-object containing also results from multiple comparisons using **opm_mcp**.

3. Available plate information

Currently substrate-layouts of 44 different plates are available within **opm** and listed in Table 1. An overview about currently available plates types can be viewed by

```
R> plate_type(full = TRUE)

[1] "AN2 (Anaerobe Identification)"
[2] "ECO (Microbial Community Analysis)"
[3] "FF (Fungi Identification)"
[4] "Gen III (Identification)"
[5] "PM-M TOX01 (Chemical Sensitivity)"
[6] "PM-M01 (Carbon and Energy Sources)"
[7] "PM-M02 (Carbon and Energy Sources / Nitrogen Sources)"
[8] "PM-M03 (Carbon and Energy Sources / Nitrogen Sources)"
[9] "PM-M04 (Carbon and Energy Sources / Nitrogen Sources)"
[10] "PM-M05 (Ions)"
[11] "PM-M06 (Hormones & Metabolic Effectors)"
[12] "PM-M07 (Hormones & Metabolic Effectors)"
[13] "PM-M08 (Hormones & Metabolic Effectors)"
[14] "PM-M11 (Anti-Cancer Agents)"
[15] "PM-M12 (Anti-Cancer Agents)"
[16] "PM-M13 (Anti-Cancer Agents)"
[17] "PM-M14 (Anti-Cancer Agents)"
[18] "PM01 (Carbon Sources)"
[19] "PM02 (Carbon Sources)"
[20] "PM03 (Nitrogen Sources)"
[21] "PM04 (Phosphorus and Sulfur Sources)"
[22] "PM05 (Nutrient Supplements)"
[23] "PM06 (Peptide Nitrogen Sources)"
[24] "PM07 (Peptide Nitrogen Sources)"
[25] "PM08 (Peptide Nitrogen Sources)"
[26] "PM09 (Osmolytes)"
[27] "PM10 (pH)"
[28] "PM11 (Chemical Sensitivity Bacteria)"
[29] "PM12 (Chemical Sensitivity Bacteria)"
[30] "PM13 (Chemical Sensitivity Bacteria)"
[31] "PM14 (Chemical Sensitivity Bacteria)"
[32] "PM15 (Chemical Sensitivity Bacteria)"
[33] "PM16 (Chemical Sensitivity Bacteria)"
[34] "PM17 (Chemical Sensitivity Bacteria)"
```

```
[35] "PM18 (Chemical Sensitivity Bacteria)"
[36] "PM19 (Chemical Sensitivity Bacteria)"
[37] "PM20 (Chemical Sensitivity Bacteria)"
[38] "PM21 (Chemical Sensitivity Fungi)"
[39] "PM22 (Chemical Sensitivity Fungi)"
[40] "PM23 (Chemical Sensitivity Fungi)"
[41] "PM24 (Chemical Sensitivity Fungi)"
[42] "PM25 (Chemical Sensitivity Fungi)"
[43] "SF-N2 (Sporulating, Filamentous, Gram- Identification)"
[44] "SF-P2 (Sporulating, Filamentous, Gram+ Identification)"
[45] "YT (Yeast Identification)"
```

explain available plates/substrate-layout, also ID plates can be run in PM-mode (Vaas, Sikorski, Michael, Göker, and Klenk 2012) accessing substrate information is explained in Section 4

4. Accessing substrate information

The **opm** package contains a number of functions suitable for accessing precomputed information on the substrates of wells and plates. In the manual and help pages these functions are contained in 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`.

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:

Plate	Group
PM01 (Carbon Sources)	C utilization
PM02 (Carbon Sources)	C utilization
PM04 (Phosphorus and Sulfur Sources)	P/S utilization
PM05 (Nutrient Supplements)	C / N utilization
PM03 (Nitrogen Sources)	N utilization
PM06 (Peptide Nitrogen Sources)	N utilization
PM07 (Peptide Nitrogen Sources)	N utilization
PM08 (Peptide Nitrogen Sources)	N utilization
PM09 (Osmolytes)	Osmotic sensitivity
PM10 (pH)	pH sensitivity
PM11 (Chemicals)	Chemical sensitivity
PM12 (Chemicals)	Chemical sensitivity
PM13 (Chemicals)	Chemical sensitivity
PM14 (Chemicals)	Chemical sensitivity
PM15 (Chemicals)	Chemical sensitivity
PM16 (Chemicals)	Chemical sensitivity
PM17 (Chemicals)	Chemical sensitivity
PM18 (Chemicals)	Chemical sensitivity
PM19 (Chemicals)	Chemical sensitivity
PM20 (Chemicals)	Chemical sensitivity
PM21 (Chemical Sensitivity)	Chemical sensitivity
PM22 (Chemical Sensitivity)	Chemical sensitivity
PM23 (Chemical Sensitivity)	Chemical sensitivity
PM24 (Chemical Sensitivity)	Chemical sensitivity
PM25 (Chemical Sensitivity)	Chemical sensitivity
Gen III (Identification)	Identification
ECO (Microbial Community Analysis)	Identification
PM-M TOX01 (Chemical Sensitivity)	C utilization mammals
PM-M01 (Carbon and Energy Sources)	C / N utilization mammals
PM-M02 (Carbon and Energy Sources / Nitrogen Sources)	C / N utilization mammals
PM-M03 (Carbon and Energy Sources / Nitrogen Sources)	C / N utilization mammals
PM-M04 (Carbon and Energy Sources / Nitrogen Sources)	C / N utilization mammals
PM-M05 (Ions)	Chemical sensitivity mammals
PM-M06 (Hormones & Metabolic Effectors)	Chemical sensitivity mammals
PM-M07 (Hormones & Metabolic Effectors)	Chemical sensitivity mammals
PM-M08 (Hormones & Metabolic Effectors)	Chemical sensitivity mammals
PM-M11 (Anti-Cancer Agents)	Chemical sensitivity mammals
PM-M12 (Anti-Cancer Agents)	Chemical sensitivity mammals
PM-M13 (Anti-Cancer Agents)	Chemical sensitivity mammals
PM-M14 (Anti-Cancer Agents)	Chemical sensitivity mammals
FF (Fungi Identification)	Identification fungi
SF-P2 (Sporulating and Filamentous P2)	Identification fungi
SF-N2 (Sporulating and Filamentous N2)	Identification fungi

Table 1: Currently available plate-layouts and substrate information in **opm**

```
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

explain pathway based data integration and visualization; map and render user data on relevant pathway graphs

KEGG is a database resource for understanding high-level functions and utilities of the biological system, such as the cell, the organism and the ecosystem, from molecular-level information, especially large-scale molecular datasets generated by genome sequencing and other high-throughput experimental technologies. Pathway Mapping KEGG PATHWAY mapping is the process to map molecular datasets, especially large-scale datasets in genomics, transcriptomics, proteomics, and metabolomics, to the KEGG pathway maps for biological interpretation of higher-level systemic functions.

package **pathview** (Luo and Brouwer 2013): All users need is to supply their gene or compound data and specify the target pathway. Pathview automatically downloads the pathway graph data, parses the data file, maps user data to the pathway, and renders pathway graph with the mapped data. Although built as a stand-alone program, pathview may seamlessly integrate

with pathway (and functional) analysis tools for a large-scale and fully automated analysis pipeline.

```
R> detach(package:opm)
R> suppressPackageStartupMessages(library("pathview"))
R> library("opm")
R> # here provide the annotated vector from vaas_1
R> xx <- annotated(vaas_1)
R> # here map the PM data stored in xx onto the pathway-map for galactose-metabolism
R>
R> vaas_1_KEGG_native <- pathview(gene.data = NULL,
                                cpd.data = xx , na.col="grey",
                                low = list(gene = "white", cpd = "white"),
                                mid = list(gene = "lightsteelblue1", cpd = "lightsteelblue1"),
                                high = list(gene = "green4", cpd = "magenta4"),
                                pathway.id = "00052", species = "ko", out.suffix = "KEGG-native",
                                keys.align = "x", kegg.native = T,
                                key.pos = "topright",
                                map.cpdname = TRUE,
                                is.signal = TRUE,
                                limit = list(gene=2, cpd=400),
                                bins = list(gene=0.5, cpd=20),
                                discrete = list(gene=FALSE, cpd=FALSE),
                                both.dirs = list(gene = FALSE, cpd = FALSE),
                                cex = 0.25, sign.pos = "bottomleft", cpd.lab.offset = 0,
                                same.layer = F,
                                rank.dir="TB")
R> file.rename("ko00052.KEGG-native.png", "ko00052_KEGG-native.png")
```

```
[1] TRUE
```

```
R> vaas_1 <- pathview(gene.data = NULL,
                      cpd.data = xx , na.col="grey",
                      low = list(gene = "white", cpd = "white"),
                      mid = list(gene = "lightsteelblue1", cpd = "lightsteelblue1"),
                      high = list(gene = "green4", cpd = "magenta4"),
                      pathway.id = "00052", species = "ko", out.suffix = "non-native",
                      keys.align = "x", kegg.native = F,
                      key.pos = "bottomright", map.cpdname = TRUE,
                      is.signal = TRUE, afactor = 1000,
                      limit = list(gene = 2, cpd = 400),
                      bins = list(gene = 0.5, cpd = 20),
                      discrete = list(gene=FALSE, cpd=FALSE),
                      both.dirs = list(gene = FALSE, cpd = FALSE),
                      cex = 0.7, sign.pos = "bottomleft", cpd.lab.offset = 0,
                      same.layer = F,
                      pdf.size = c(10,7),
                      rank.dir = "TB")
R> file.rename("ko00052.non-native.pdf", "ko00052_non-native.pdf")
```

```
[1] TRUE
```

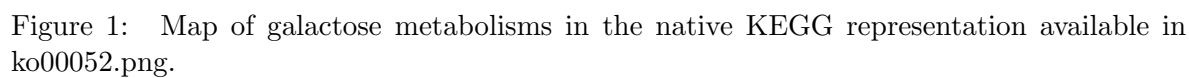


Figure 1: Map of galactose metabolisms in the native KEGG representation available in ko00052.png.

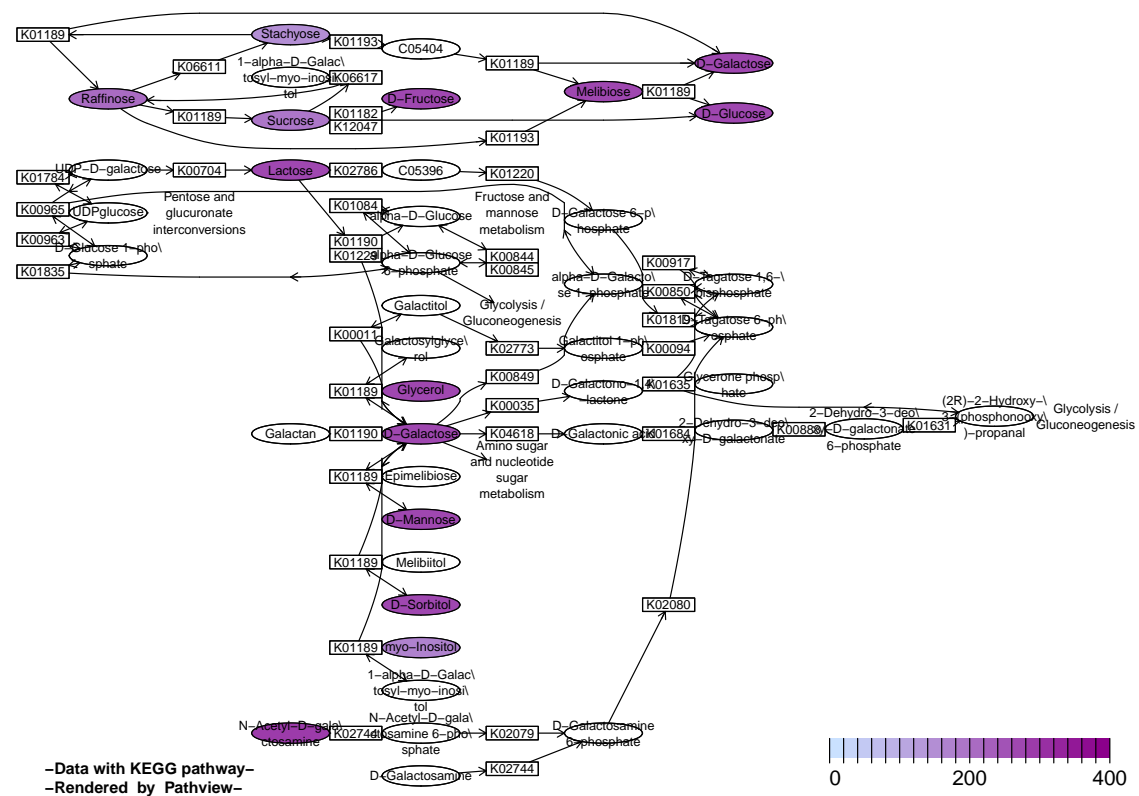


Figure 2: Map of galactose metabolisms in the non-native KEGG representation.

An enhancement of the **opm** package would be to include much more precomputed information about the substrates, thus greatly facilitating data arrangement and hypothesis testing. At the moment only the translation of well coordinates to substrate names is provided, as well as access to MeSH names and CAS, ChEBI, KEGG and Metacyc IDs (and the according web pages) for the majority of the substrates. More substrate information could be integrated into the package, particularly for arranging the substrate into groups, thus easing testing of phenotypic hypotheses.

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/Bioconductor 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>.
- Vaas LA, Sikorski J, Michael V, Göker M, Klenk H (2012). “Visualization and Curve Parameter Estimation Strategies for Efficient Exploration of Phenotype MicroArray Kinetics.” *PLoS ONE*, **7**, e34846. doi:10.1371/journal.pone.0034846.

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