SIMPEL: Stable Isotope assisted Metabolomics for Pathway ELucidation

Shrikaar Kambhampati, Allen Hubbard and Doug Allen

3/1/2022

1. Introduction

SIMPEL is an R package for streamlining data analysis of non-stationary stable isotopic pulse or pulse chase labeling experiments and can be used with both targeted and untargeted metabolomics data sets. SIMPEL is capable of calculating and identifying single and dual labeled isotopologues from XCMS feature lists for chemical formulae from the user. SIMPEL can generate Mass Isotopologue Distribution (MID) matrices and calculate average labeling for compounds. The data can serve as an important input for metabolic flux calculations in modeling software. SIMPEL includes post-processing data analysis features to generate MID plots, average labeling diagrams, perform *k*-means clustering analysis, Principle Component Analysis (PCA) and to visualize label enrichment as a heat map.

2. Getting Started

To use SIMPEL, first install the package. In R, type

```
##install devtools if you don't already have it
install.packages("devtools")
##load the library devtools
library("devtools")
##install SIMPEL using the install_github function within devtools
install_github("victorfrnak/SIMPEL_final/SIMPEL")
```

This is a one-time process. For each new R session, first load the SIMPEL package using the `library` function.

```
library("SIMPEL")
```

3. Importing Data

A Lipidomics experiment and a small molecule metabolomics experiment, were pre-processed using XCMS and are provided as .csv files within "extdata" ("xcms_data_Lipidomics_SIMPEL.csv" and "xcms_data_DualLabel_SIMPEL.csv", respectively). The lipidomics dataset corresponds to an isotopically non-stationary pulse (0-32 hours) labeling experiment using developing seeds of *Camelina sativa* performed with U-^{13}C Glucose, and the second is a pulse (0-8 hours) labeling experiment using roots of *Arabidopsis thaliana* with $^{13}C_5{}^{15}N_2$ Glutamine. Users can also choose to download this data from metabolomics workbench, convert it into mzML files and perform independent XCMS based data pre-processing.

In addition to the pre-processed xcms_data, SIMPEL also requires “Compound_data”, which is an annotation file that contains chemical formulas and retention times of compounds for which the user wishes to identify isotopologues. Two annotation files, “Compound_data_Lipidomics_SIMPEL.txt” and “Compound_data_DualLabel_SIMPEL.txt” are provided within ‘extdata’, for Lipidomics and small molecule metabolomics experiment, respectively, generated via MS2 analysis using Compound Discoverer 4.0. In this example, we will work with the Dual Label dataset. First load the “extdata” corresponding to the dual label data set.

```
Compound_data <- read.table(system.file("extdata",
                "Compound_data_DualLabel_SIMPEL.txt", package = "SIMPEL"),
                sep = "\t", header = TRUE)

xcms_data <- read.table(system.file("extdata",
                "xcms_data_DualLabel_SIMPEL.csv", package = "SIMPEL"),
                sep = ",", header = TRUE)
```

The annotation data table `Compound_data` contains five columns

1. “prefix” - corresponds to a bin number to be used as a unique identifier for each of the compounds.
2. “polarity” – whether the compound is to be searched against “pos” ionization mode data or a “neg” ionization mode data within the `xcms_data`. Note: Currently SIMPEL can only calculate m/z for +H and –H adducts.
3. “rt” – the retention time at which the compound is to be expected
4. “formula” – the chemical formula for which exact mass is to be calculated
5. “Compound” – name of the compound, which can also be used as a unique identifier.

To view the first few rows/structure of `Compound_data` use the following command

```
head(Compound_data)
```

The xcms data table `xcms_data` contains three essential columns that serve to categorize data by “m/z”, “rt” and “polarity”. The remaining columns contain mass spectral data. Sample names should be specified as “Time_Category_Replicate” where “time” denotes the time point of the isotopic labeling experiment, “Category” represents the treatment/tissue type/genotype that is specific to the experiment (i.e. in the “extdata” included examples, the Lipidomics dataset has two categories that are the types of tissue, “Cotyledon” and “EA” (Embryo Axis), and the Dual Label dataset also has two categories (genotypes), a “WT” and a mutant “GAT”. To view the structure of the first few rows of the `xcms_data` set run the following command,

```
head(xcms_data)
```

The remainder of the tutorial uses the Dual Label dataset, you can also try out the Lipidomics dataset to test the application using single labeled sources.

4. Identifying isotopologues and creating data objects

Based on the list of chemical formulae provided within `Compound_data` SIMPEL first extracts elemental information using the function `get_element_count()` and calculates the m/z for a +H

ion or a –H ion depending on the polarity specified using the fuction `get_comp_mass()`. Try the example compound, adenine;

```
get_element_count"C5H5N5")

get_comp_mass("C5H5N5", "pos")
```

Using the element list and the calculated m/z, a "compound_lookup_table" is created with all the possible C and N isotopologues for the given compound (or the entire list of compounds within `Compound_data`) using the function `get_comp_mz_lookup()`. As an example, use the function to create a lookup table for adenine.

```
Ade_lookup_table <- get_comp_mz_lookup(compound_data = Compound_data,
                        comp_formula = "C5H5N5", r_time = 2.53, ppm = 5,
                        polarity = "pos")

##to view the look up table entirely
view(Ade_lookup_table)

##to view the first isotopologue within the lookup table
Ade_lookup_table[(1)]
```

The number of possible isotopologues for a compound with C and N labels can be calculated using, [no. of C's +1] x [no. of N's +1]. In the above example with adenine (C5H5N5), the total number of combined carbon and nitrogen isotopologues is 36, and hence a list of the 36 possible isotopologues is created. The lookup table also contains the calculated m/z for each isotopologue, and an error margin (specified as tolerance by the user in ppm) with upper and lower bounds of allowed m/z for a match in the XCMS feature list. This information is then used by `get_comp_stage()` function to append additional columns to the `xcms_data`, including the isotopologue ID, number of labeled carbons and nitrogens within that isotopologue as well as the total number of carbon and nitrogen atoms within the formula. This information is used by subsequent steps to create MIDs and average labeling tables.

When the two input files `xcms_data` and `Compound_data` are provided, the function `get_table_objects()` can be used to extract the isotopologue information for all the compounds listed within the `Compound_data`. For more information on the usage of the function;

```
?get_table_objects
```

For the example dual label dataset, use the following command to run the function.

```
DualLabel_extdata <- get_table_objects(xcms_data, Compound_data,
                        ppm = 5, rt_tolerance = 0.1, output = "DualLabel")
```

In dual label data sets with ^{13}C and ^{15}N labels, the m/z difference between the two labeled elements is 0.006 amu, using high resolution mass spectrometry will successfully resolve the two isotopologues and setting the `ppm` error to ≤ 5 will enable correct assignment of ^{13}C and ^{15}N specific isotopic features and reduce false positives. Similarly, for short chromatographic methods, where major retention deviations are not expected between runs, tighter `rt_tolerance` will also help reduce false positive ID of labeled features.

In addition to creating `MIDs` and `average_labeling` objects, the function `get_table_objects()` also creates `scaled_MIDs` and `mol_equivalent_labeling` objects. A direct comparison of label enrichment between compounds of different pool sizes or the same compound between two different categories, may not always be appropriate (for details see, Buescher et al. 2015, COPBIO). Since obtaining pool size measurements for metabolomics data is not always feasible,

a proxy table for pool sizes of compounds that were identified within the metabolomics data set is created. To create a proxy pool table, the `xcms_data` is first normalized by the median of all the signals within each sample, and all the identified isotopologues for each of the listed compounds are summed. This proxy for pool size of each compound is used to scale (multiply) the MIDs to create a `scaled_MIDs` object where the abundance of each of the isotopologues is now a "mol equivalent" amount and is no longer a proportion of that isotopologue within the compound (as it is for `MIDs`). A table with the sum of all the labeled isotopologues (M1-Mn) for each compound, that represents the mol equivalent of labeled product formed, in the non-stationary isotopic labeling experiment, is then created to get the `mol_equivalent_labeling` object. The isotopologues of different compounds within the `scaled_MIDs` object and the `mol_equivalent_labeling` for compounds within and between categories are now directly comparable. See below for calculation of `average labeling` and `mol equivalent labeling`

$$\text{Average Labeling (\% label Enrichment)} = \frac{\sum_{i=0}^{n} i \,.\, Si}{l}$$

Where, *i* = mass isotopologue, *n* = total number of possible labeled isotopologues ([no. of C's +1] x [no. of N's +1]), *S* = relative % of the respective isotopologues (0-n), *l* = total number of possible labeled atoms in the chemical formula (no. of C's + no, of N's)

$$Mol\ equivalents\ of\ labeled\ product\ = \sum_{i=1}^{n} \frac{x}{100} . Si$$

Where, *i* = mass isotopologue, *n* = total number of possible labeled isotopologues ([no. of C's +1] x [no. of N's +1]), *S* = relative % of the respective isotopologue (0-n), *x* = proxy for pool size

The function `get_table_objects()` now contains four objects that are exported as data tables into a sub-directory labeled as the `output` argument within the SIMPEL directory of R libraries in the users local environment. The output files have the `output` argument appended to the file names. The table objects can can be viewed using the following command (note: replace MIDs with the data object you wish to view)

```
DualLabel_extdata$MIDs
```

Update: SIMPEL now includes natural abundance (NA) correction for isotope enriched data from both single and dual labels. High percentage of naturally occurring ^{13}C and ^{15}N especially for large molecules potentially skews interpretation of label enriched data. To alleviate this concern, the NA correction R package IsoCorrectoR (Heinrich et al. 2018, Sci. Rep.) is incorporated as `NACorrectionfxnII()` function within `get_table_objects()` and a new function `get_table_objects_NA_corrected()` is added to SIMPEL. To execute the NA correction using the updated function, try the following command

```
DualLabel_extdata <- get_table_objects_NA_corrected(xcms_data,
                        Compound_data, ppm = 5, rt_tolerance = 0.1,
                        output = "DualLabel")
```

This creates all four of the above described data objects along with four additional objects that represent the NA corrected versions of these objects. All eight objects are exported, with the directory structure described in Appendix I below, along with the output sub-directory created by IsoCorrectoR.

5. Mass Isotopologue Distributions (MIDs) plot to evaluate label enrichment within compounds

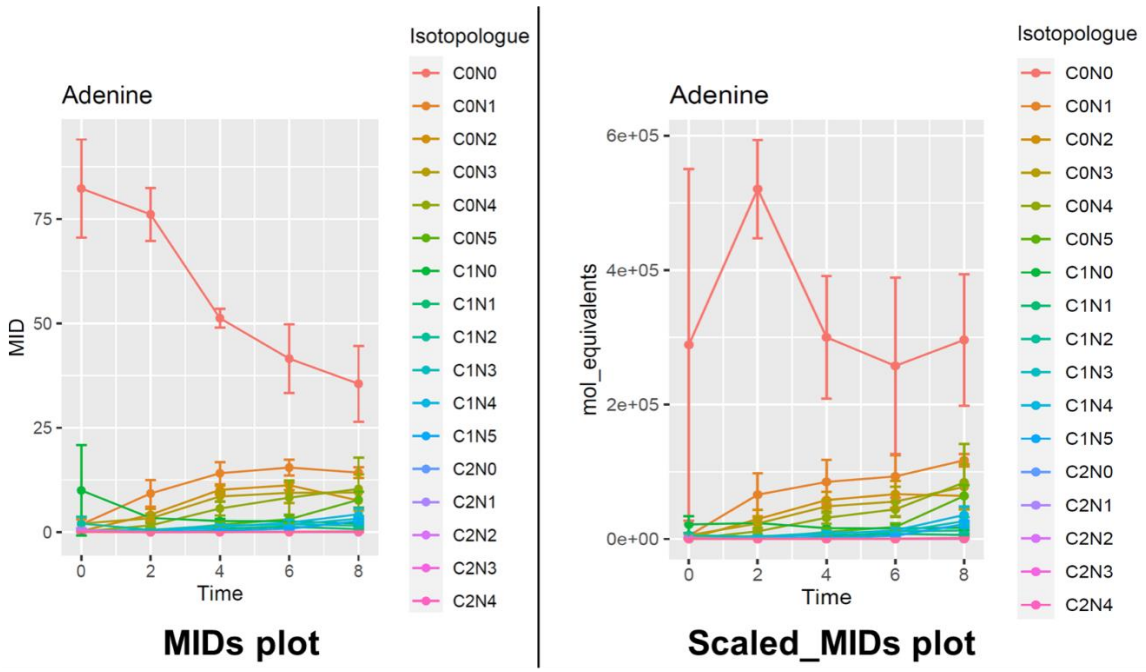
For visualization of MIDs, the `MIDplots()` function can be used to generate a pdf file containing MID plots for all the compounds identified using the MIDs object. This function generates two pdf files, one with all the MIDs plotted together and the other with an option to split the MIDs of each compound into two plots. The split version may be useful to visualize labeling in compounds that vary extensively in label enrichment, to show several trends, over the course of the experiment. Use this command to view details on `MIDplots()`

```
?MIDplots
```

To plot the MIDs object and the scaled_MIDs object of the current example with Dual labels, try the following command

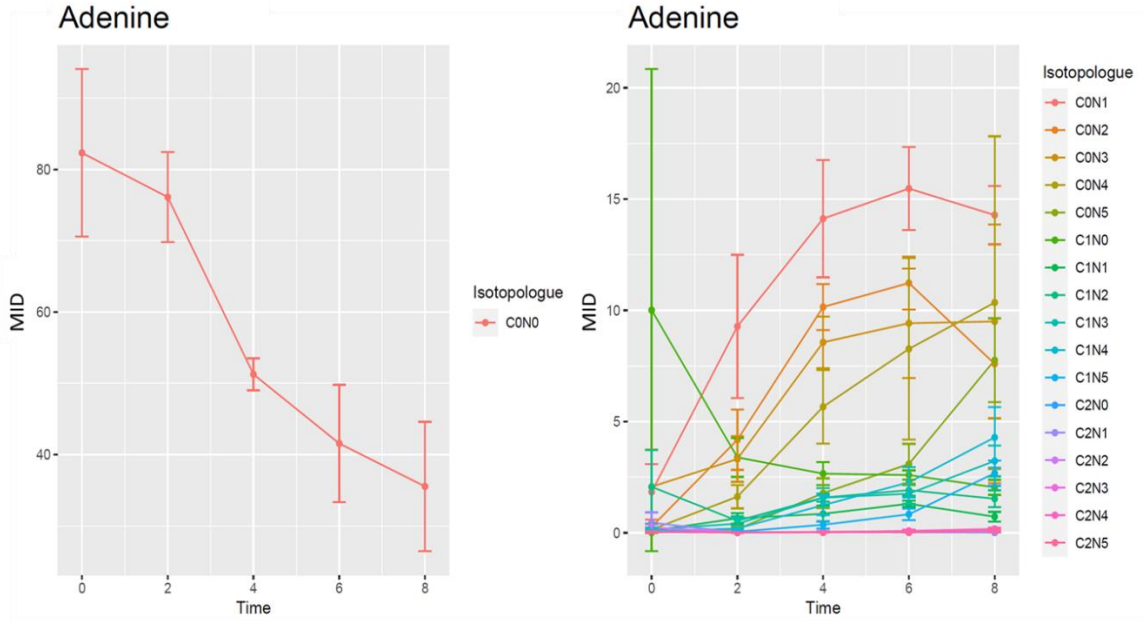
```
MIDs <- MIDplot(DualLabel_extdata$MIDs, Category = "WT", axisTitle = "MID",
        plotTitle ="Compound")

##To append additional title to the exported pdf the argument, outputName
is used
scaled_MIDs <- MIDplot(DualLabel_extdata$scaled_MIDs, Category = "WT",
                axisTitle = "mol_equivalents", plotTitle = "Compound",
                outputName = "scaled")
```

Open the PDF files created within your working directory and verify the MIDs plot for adenine



All the detected isotopologues for the compound adenine within `xcms_data` should be shown as MID plots. These MID patterns can be evaluated to infer pathway functionality. MID plots for the mutant “GAT” can be generated by using the argument `Category = "GAT"` instead of WT and the split MIDs provide better visualization of labeled isotopologues. The labeling in the isotopologues of adenine follow an expected trend with the M0 (unlabeled/C0N0) decreasing

over time as the labeled isotopologues increase in abundance over the course of this pulse labeling experiment.



Split MIDs plot

6. Label Enrichment plots to visualize and compare average labeling description for compounds

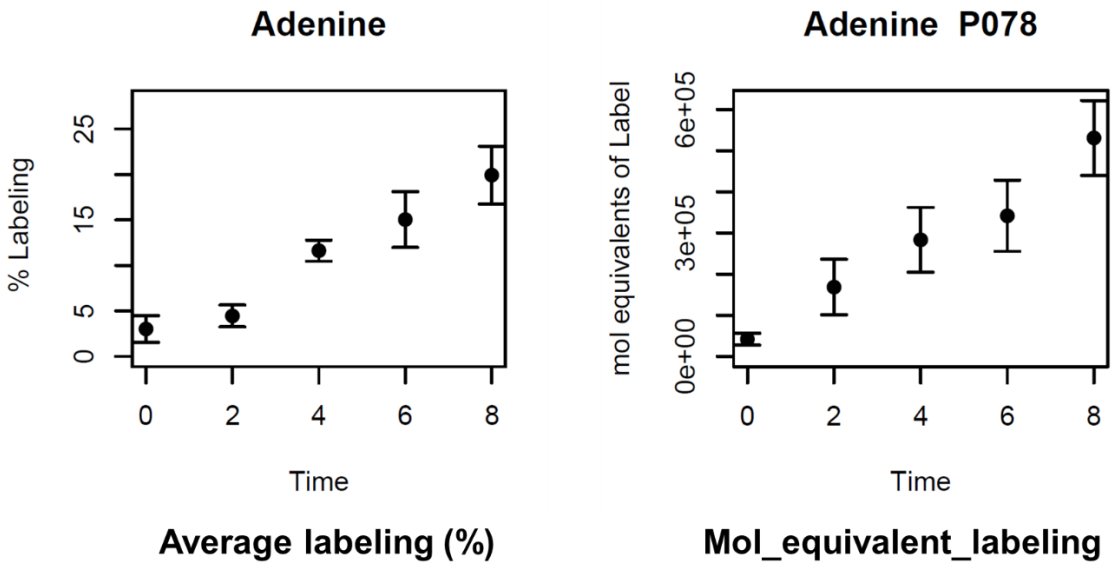
For visualization of average labeling, the `label_enrichment_plot()` function can be used to generate a pdf file containing average label enrichment (%) plots for all the compounds identified using the average_labeling object. This plot identifies all compounds that are actively changing in mass due to label incorporatin. `label_enrichment_plot()`

```
?label_enrichment_plot
```

To plot the average_labeling object and the mol_equivalent_labeling object of the current example with Dual labels, try the following command

```
Average_Labeling <-
               label_enrichment_plot(DualLabel_extdata$average_labeling,
               Category = "WT", axisTitle=" % Labeling",
               plotTitle="Compound", outputName = "Average")

## To add additional plot title, use plotTitle2 with the column of interest
such as "Bin" or "Formula" or "mz"
mol_equivalents <-
       label_enrichment_plot(DualLabel_extdata$mol_equivalent_labeling,
       Category = "WT", axisTitle= "mol equivalents of Label", plotTitle=
       "Compound", plotTitle2= "Bin", outputName = "mol Equi")
```

Open the PDF files created within your working directory and verify the plots for adenine



Average labeling (%) **Mol_equivalent_labeling**

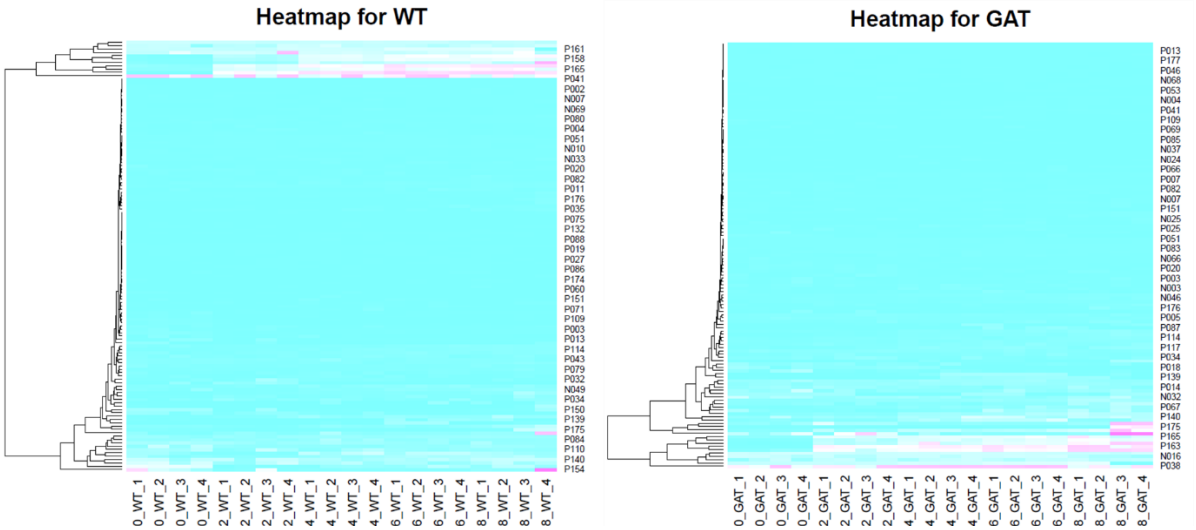
Notice the difference in label enrichment trend between the two plots. The average labeling represents proportion of labeled atoms on a 'per atom' basis out of the total adenine pool. The inactive pool (if any) or spatially separated (amongst different subcellular compartments) pool of adenine significantly contributes to the labeling trend in average labeling description. While there is valuable information that can be obtained here, it is often not suitable when comparative studies are performed, especially when comparing two categories (in this case WT and mutant) with different pool sizes. mol equivalent labeling is independent of the unlabeled pool since only the labeled isotopologues are summed and can be used to directly compare trends between categories. The two objects, average_labeling and mol_equivalent_labeling, with `Category = "GAT"` show the differences in label enrichment between WT and the mutant.

7. PCA plot and heatmaps to visualize global labeling patterns

A global comparative view of the label enrichment data can be obtained using the function `PCA_and_heatmap()`. This function performs principle component analysis using label enrichment data, and both average_labeling and mol_equivalent_labeling can be used to perform this analysis. By default the function plots three principle components, however, the user has the option of specifying more if required (i.e. if not all the variation is explained by the three PCs). In addition, the function also produces a heatmap for each category specified highlighting label enrichment on a time scale. This information can be used to determine the proportion of identified metabolites that exhibited a labeling pattern (active) vs the ones with no labeling (inactive). Users can infer relationships between the two categories using this function, and perform more detailed analysis using the next function `getClustersAndPlots()`. To use this function, execute the following commands and ensure that the images below (for average_labeling) are replicated.

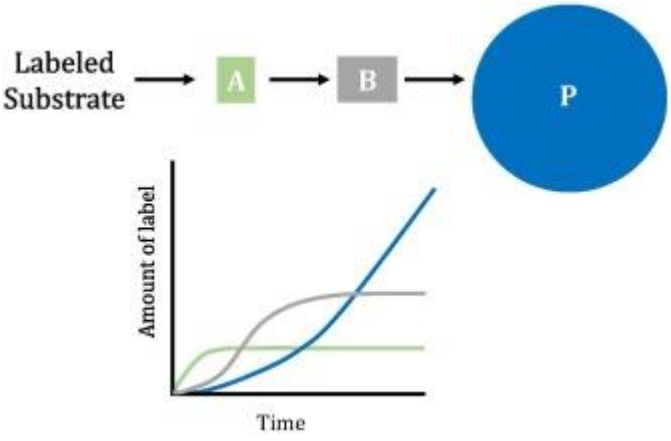
```
PCA_Heatmap = PCA_and_heatmap(DualLabel_extdata$average_labeling,
                PCMax = 3, heatMapCategories = c("WT", "GAT"),
                labels="Bin", outputName = "Average")

## the labels in the above command can be substituted with "Compound" if
desired to replace the labeling in the heatmaps. Use the line below for
plotting the mol_equivalent_labeling object
PCA_Heatmap = PCA_and_heatmap(DualLabel_extdata$mol_equivalent_labeling,
                PCMax = 3, heatMapCategories = c("WT", "GAT"),
```

```
                labels="Bin", outputName = "mol equi")
```



8. *k*-means clustering analysis to group compounds based on pathway activity

A major post-processing data analysis function that SIMPEL performs is the *k*-means clustering analysis. Compounds that belong to same pathway tend to show similar labeling trends, at metabolic steady state. *k*-means clustering analysis groups compounds based on their similarities in labeling patterns. In a pulse labeling experiment with multiple time points that eventually reach isotopic steady state, the labeling kinetics of compounds that are closest to the labeled substrate (source being provide) exhibit a hyperbolic trend that quickly plateaus, while the compounds that are farther away from the labeled substrate show a lag in the initiation of labeling followed by an even greater delay in the subsequent compounds (see the model figure below for a series of compounds A, B and P. For a detailed review, see Allen et al. 2015, Prog Lipid Res).



Using the function `getClustersAndPlots()`, you can perform k-means clustering analysis with either the average_labeling or mol_equivalent_labeling objects. Since mol_equivalent_labeling generally results in an uneven y-axis for different compounds (depending on the size of their pools) as opposed to average labeling (that is represented in %), the data is first transformed with the maximum value for each compound set to 1 and all the other data points normalized to

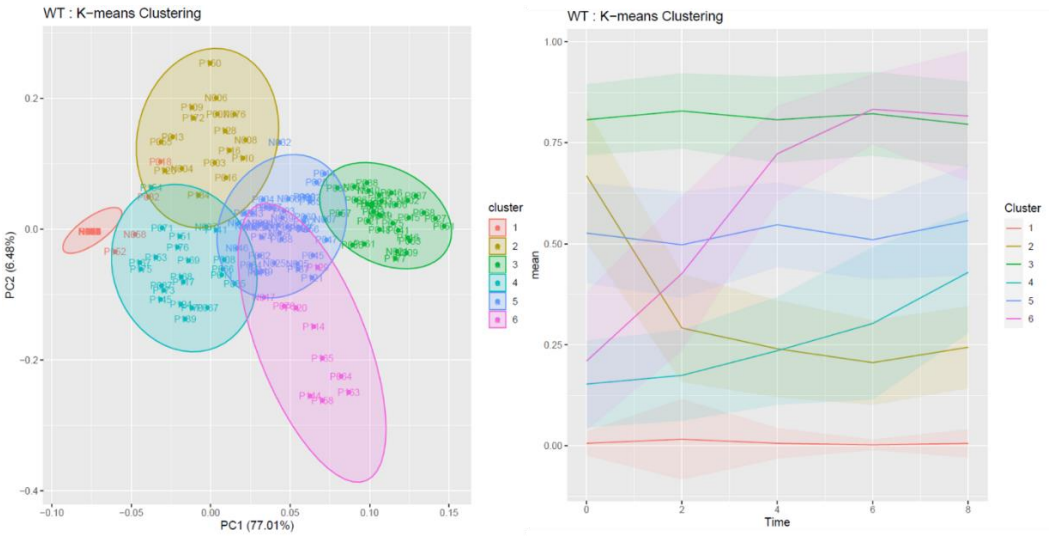
that value. This enable an “apples to apples” comparison of trends in labeling. While clustering analysis groups compounds based on their label enrichment over time, there is useful information within the MIDs of individual compounds. The compounds within each cluster can be further clustered using their MIDs. Clustered MIDs provide information on the specific MIDs per compound along each step in the pathway. Try the following command

```
?getClustersAndPlots()
```

Try the following command to perform *k*-means clustering analysis using “WT” as Category. By default (`nClust = 0`), the number of clusters is calculated by the formula *k* = sqrt(n/2). However, the user also has the option of using a predetermined number, in the example below we use 6.

```
clusters_AvgLabeing <-
                getClusterAndPlots(DualLabel_extdata$average_labeling,
                DualLabel_extdata$MIDs, Category = "WT", nClust = 6,
                labels="Bin", outputName = "Average")

## note that in the above line when we analyze average_labeling, we use the
corresponding MIDs object
## similarly if we use mol_equivalent_labeling, the object scaled_MIDs need
to be used
## by default, clustering at the MIDs level is not performed, if this is
required set “doMIDs = TRUE”
clusters_moleEquivalents <-
        getClusterAndPlots(DualLabel_extdata$mol_equivalent_labeling,
        DualLabel_extdata$scaled_MIDs, Category = "WT", nClust=6,
        labels="Bin", doMIDs=TRUE, outputName = "mol Equi")
```

Notice that in the above lines we use `labels=”Bin”` due to a high number of compounds and long chemical names that can clutter the plot. If the compound name is desired, users have the option of using `labels=”Compound”`. This will produce a single pdf with two plots, one of which is k-means clustering represented on a principle component space as a 2D plot and additionally a time course representation. If you set doMIDs =TRUE, the MIDs of compounds within each of the clusters (called “Global clusters” or “compound clusters”) can be used to perform a second *k*-means analysis, the clusters of which are now called “MID clusters”. Network relationships between compounds can be established by carefully verifying the trends of the clusters and compounds within each cluster. Open the file “WT_Average_kmeans_plots.pdf” that is created in the working directory. Two plots are generated, one with the PC separation on a 2D space and the other plotted on a time scale. The same analysis can be performed for comparison on the second category by substituting “WT” with “GAT”.



Out of the 255 compounds searched against the xcms_data, 202 compounds and their isotopologues were identified and grouped into six clusters. Four out of the six clusters (1, 2, 3 and 5) did not show an increase in label enrichment that is the signature of a pulse labeling experiment. The clusters that represent a flat line (1, 3 and 5) are likely not involved in active metabolism in the context under examination. Since a decrease in label enrichment is unexpected in pulse labeling experiments under steady state conditions, cluster 2 likely represents baseline or noise level measurements. Clusters 6 and 4 that contained 10 and 23 compounds, respectively, are involved in active metabolism with cluster 6 showing a hyperbolic trend (and hence are closer to labeled substrate within the network) and cluster 4 showing a delay in label incorporation followed by the climb in slope. Further analysis of compounds within these two clusters and their MIDs using the MIDs clustering analysis, would be the next step in network reconstruction. While the subsequent part of data analysis (reconstructing metabolic network) is out of the scope of SIMPEL, and is heavily dependent on the user knowledge, future updates to SIMPEL will include an overlay on KEGG pathways based on clustering analysis highlighting the fastest, medium and slowest reaction in the context specific metabolic network under study.

9. Appendix I

Output directory structure for data tables and images exported via SIMPEL