CompMS2miner WorkFlow

WMB Edmands

Sunday, June 21, 2015

Matches MS1 features to MS2 spectra (.mzXML) files based on a mass-to-charge and retention time tolerance. Composite spectra and other data can subsequently be visualized during any stage of the CompMS2miner processing workflow. Composite spectra can be denoised, ion signals grouped and summed, substructure groups identified, common Phase II metabolites predicted and features matched to data bases monoisotopic mass data and insilico MS2 fragmentation data. The resulting data can then be readily curated by sending to a local or online couchDB database.

Ensure all necessary R packages are installed, rCharts must be installed from GitHub using the install_github function of devtools. The rCharts packages is used for reactive graphical visualizations in the shiny application. The following will install this additional package resource:

```
install.packages("devtools")
library(devtools)
## install rCharts directly from github using devtools
install_github("ramnathv/rCharts")
```

At any stage during the compMS2 workflow a compMS2 class object can be visualized with a Shiny application. Full usage of the compMS2shiny application functionality requires an internet connection. The end result of following the workflow within this document using the example data provided can be visualized using the function:

```
library(CompMS2miner)
compMS2shiny(compMS2example)
```

The following example illustrates the CompMS2miner workflow:

1. construct compMS2 class object.

From MS2 data (in the .mzXML file format) and an MS1feature table. The compMS2 object can also be constructed in as a parallel computation in the case of large peak tables and/ or larger numbers of MS2 data files.

```
# file path example MS1features in comma delimited csv file
# (see ?example_mzXML_MS1features for details).
MS1features_example <- system.file("extdata", "MS1features_example.csv",
                                   package = "CompMS2miner")
# mzXml file examples directory
mzXmlDir_example <- dirname(MS1features_example)
# use parallel package to detect number of cores
nSlaves <- parallel::detectCores()
# create compMS2 object
compMS2demo <- compMS2(MS1featuresCsvName = MS1features_example,
                       mzXMLdir = mzXmlDir_example, nSlaves=nSlaves,
                       mode = "pos", precursorPpm = 10, ret = 10,
                       TICfilter = 10000)
## creating compMS2 object in positive ionisation mode
```

```
## Reading MS1 feature table...
## ...Done
## 2 MS2 (.mzXML) files were detected within the directory...
## Starting SNOW cluster with 8 local sockets...
## matching MS1 peak table features to the following MS2 files:
## DDA_ACN_80.mzXML
## DDA_MeOH_80.mzXML
# View summary of compMS2 class object at any time
compMS2demo
## A "CompMS2" class object derived from 2 MS2 files
##
## 163 MS1 features were matched to 1784 MS2 precursor scans
## containing 220161 ion features
##
## Average ppm match accuracy: 1.77
## with a ppm mass accuracy tolerance of (+/-) 10
##
## Average retention time match accuracy: 3.93 seconds
## with a retention time tolerance of (+/-) 10  seconds
##
## Memory usage: 5.18 MB
```

2. Dynamic noise filtration.

filter variable noise from the data using a dynamic noise filter.

```
# dynamic noise filter
compMS2demo <- deconvNoise(compMS2demo, "DNF")
## Applying dynamic noise filter to 257 composite spectra...
## Starting SNOW cluster with 8 local sockets...
## ...done
## 256 composite spectra contained more than or equal to 5 peaks following dynamic noise filtration
# View summary of compMS2 class object at any time
compMS2demo
## A "CompMS2" class object derived from 2 MS2 files
##
## 163 MS1 features were matched to 1782 MS2 precursor scans
## containing 5030 ion features
##
## Average ppm match accuracy: 1.775
## with a ppm mass accuracy tolerance of (+/-) 10
##
## Average retention time match accuracy: 3.94 seconds
## with a retention time tolerance of (+/-) 10  seconds
##
## Memory usage: 1.94 MB
```

3. Intra-spectrum ion grouping and inter-MS2 file spectra grouping with signal summing.

group and sum ions from different scans and then combine summed ion composite spectra across multiple files. This create a single composite spectra for each MS1 EIC matched to MS2 precursor scans.

```
# intra-spectrum ion grouping and signal summing
compMS2demo <- combineMS2(compMS2demo, "Ions")
## Grouping ions in 256 composite spectra...
## Starting SNOW cluster with 8 local sockets...
## ...done
## 175 composite spectra contained more than or equal to 5 peaks following ion grouping
# View summary of compMS2 class object at any time
compMS2demo
## A "CompMS2" class object derived from 2 MS2 files
##
## 128 MS1 features were matched to 1073 MS2 precursor scans
## containing 1504 ion features
##
## Average ppm match accuracy: 1.808
## with a ppm mass accuracy tolerance of (+/-) 10
##
## Average retention time match accuracy: 3.81 seconds
## with a retention time tolerance of (+/-) 10  seconds
##
## Memory usage: 1.8 MB
#inter spectrum ion grouping and signal summing
compMS2demo <- combineMS2(compMS2demo, "Spectra")
## Combining 175 composite spectra by MS1 feature number...
## Starting SNOW cluster with 8 local sockets...
## ...done
## 128 composite spectra contained more than or equal to 5 peaks following ion grouping
# View summary of compMS2 class object at any time
compMS2demo
## A "CompMS2" class object derived from 2 MS2 files
##
## 128 MS1 features were matched to 1073 MS2 precursor scans
## containing 1198 ion features
##
## Average ppm match accuracy: 1.864
## with a ppm mass accuracy tolerance of (+/-) 10
##
## Average retention time match accuracy: 3.74 seconds
## with a retention time tolerance of (+/-) 10  seconds
##
## Memory usage: 1.75 MB
```

4. Possible substructure identification.

Characteristic neutral losses/ fragments of electrospray adducts and metabolites from literature sources (?Substructure_masses for details).

```
# annotate substructures
compMS2demo <- subStructure(compMS2demo, "Annotate")
## matching Precursor to fragment and interfragment neutral losses and fragments in 128 composite spect
# identify most probable substructure annotation based on total relative intensity
# explained
compMS2demo <- subStructure(compMS2demo, "prob")
```

```
## Identifying likely substructure type...
# summary of most probable substructure annotation
mostProbSubStr <- subStructure(compMS2demo, "probSummary")
## Substructure annotation summary :
## 128 composite spectra
## 115 substructures identified above minimum sum relative intensity of 30
##
## SubStr.table
##             phosphatidylcholine           methionine side chain
##                              52                              29
##                    polysiloxane                       phthalate
##                              27                               3
## nitroaromatics, hydroxyaldehydes                   nitrotoluenes
##                               2                               1
##                       triazines
##                               1
```

5. Metabolite identification methods.

A variety of metabolite identification methods are implemented through the function metID (see ?metID) for further details.

```
# annotate composite MS2 matched MS1 features to metabolomic databases (default
#is HMDB, also DrugBank, T3DB and ReSpect databases can also be queried).
#Warning: this may take 2-3 mins as large number of query masses
 compMS2demo <- metID(compMS2demo, "dbAnnotate")

# select most probable annotations based on substructures detected
 compMS2demo <- metID(compMS2demo, "dbProb")

# predict Phase II metabolites from SMILES codes
 compMS2demo <- metID(compMS2demo, "predSMILES")

# metFrag insilico fragmentation
#compMS2demo <- metID(compMS2demo, "metFrag")
```

6. Optional curate data in CouchDB/ publish results as a shiny App.

CompMS2 class objects can be sent to either a local or online couchDB database. Furthermore, the results of the CompMS2miner workflow can be published on the web as a shiny application to share metabolite identification information.