Peak Gather Example

bjp

7/14/2021

Example of gathering data for DB input

Include all necessary packages:

```
source('src/compliance.R')
```

Step 1) Load Method JSON data and mzML data

```
jsonfile <- 'example/PFAC30PAR_PFCA2_mzML.JSON'
methodjson <- parse_methodjson(jsonfile)
mzmlfile <- paste(dirname(jsonfile), "/", methodjson$sample$filename, sep = "")
mzml <- mzMLtoR(mzmlfile)
```

JSON Data

Sample Info:

filename	PFAC30PAR_PFCA2.mzML
description	Reference Standard for PFAS
class	analytical standard
submitter	bjp@nist.gov

Chromatography Info:

ctype	Liquid Chromatography
cvendor	ThermoFisher Scientific
cmodel	UltiMate 3000
ssolvent	water
mp1solvent	water
mp1add	ammonium acetate
m2solvent	methanol
mp2add	ammonium acetate
mp3solvent	none
mp3add	none
mp4solvent	none
mp4add	none
colvendor	Agilent Technologies

colname	Poroshell C18
colchemistry	C18
colid	2.1
collen	50
coldp	2.7
gcolvendor	none
gcolname	
gcolchemistry	none
gcolid	
gcollen	
gcoldp	

Mass Spectrometry Info:

msvendor	ThermoFisher Scientific
msmodel	Q-Exactive
imode	electrospray ionization
polarity	negative
vvalue	2500
vunits	V
massanalyzer1	quadrupole
massanalyzer2	orbitrap
fragmode	HCD
cevalue	30
cetype	fixed
ceunits	normalized
ms2exp	DDA
isowidth	0.7
msaccuracy	5
ms1resolution	70000
ms2resolution	17500

QC Method Info:

qcused	TRUE
qctype	list(name = “Mass Analyzer Calibration”, value = TRUE)
qctype	list(name = “External Standard Verification”, value = TRUE)
qctype	list(name = “Internal Standard Verification”, value = FALSE)
qctype	list(name = “Matrix Standard Verification”, value = FALSE)

Peaklist:

count	name	identifier	ionstate	mz	rt	peak_starttime	peak_endtime	verified
1	Perfluoropentanoic acid	2646	[M-H]-	262.9760	8.60	8.4	9.1	TRUE
2	Perfluorohexanoic acid	2643	[M-H]-	312.9730	10.80	10.5	11.1	TRUE
3	Perfluoroheptanoic acid	2640	[M-H]-	362.9699	12.09	11.9	12.4	TRUE
4	Perfluorooctanoic acid	2637	[M-H]-	412.9665	13.05	12.8	13.4	TRUE
5	Perfluorononanoic acid	2635	[M-H]-	462.9635	13.80	13.6	14.1	TRUE
6	Perfluorodecanoic acid	2632	[M-H]-	512.9602	14.50	14.3	14.8	TRUE
7	Perfluoroundecanoic acid	2630	[M-H]-	562.9573	15.10	14.9	15.4	TRUE

count	name	identifier	ionstate	mz	rt	peak_starttime	peak_endtime	verified
8	Perfluorododecanoic acid	2629	[M-H]-	612.9540	15.60	15.4	15.8	TRUE
9	Perfluorotridecanoic acid	2628	[M-H]-	662.9516	16.00	15.9	16.2	TRUE
10	Perfluorotetradecanoic acid	2627	[M-H]-	712.9487	16.30	16.2	16.6	TRUE

Step 2) For each peak in the peak list, gather method data, peak data, and peak-specific MS data.

The data is also matched against the compound list to pair **COMPOUND ID** values. A surrogate will be the 'src/gather/pfas_cmpds.csv' file.

```
compoundtable <- read.csv('src/gather/pfas_cmpds.csv', header = TRUE, row.names = NULL)
dat <- peak_gather_json(methodjson, mzml, compoundtable)
```

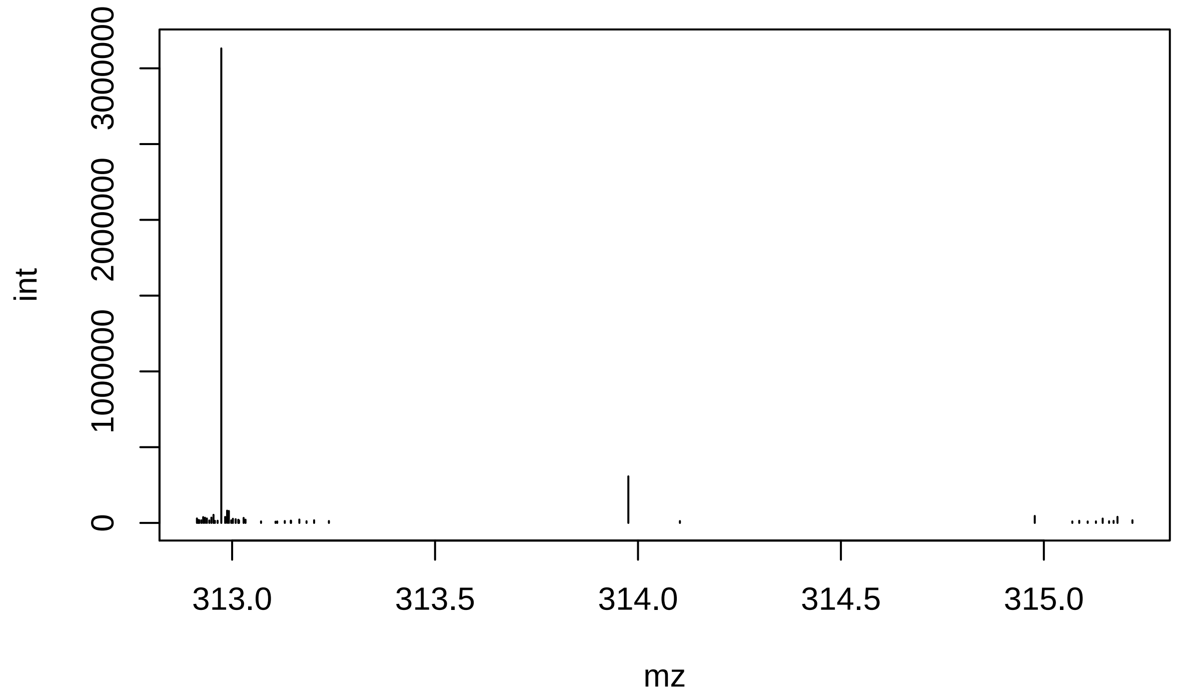
Example dataset:

```
i <- 2 ## because there were no annotations for peak 1
dat[[i]]$peak
```

```
##   count                     name identifier ionstate      mz   rt peak_starttime
## 1     2 Perfluorohexanoic acid       2643    [M-H]- 312.973 10.8           10.5
##   peak_endtime verified
## 1         11.1     TRUE
```

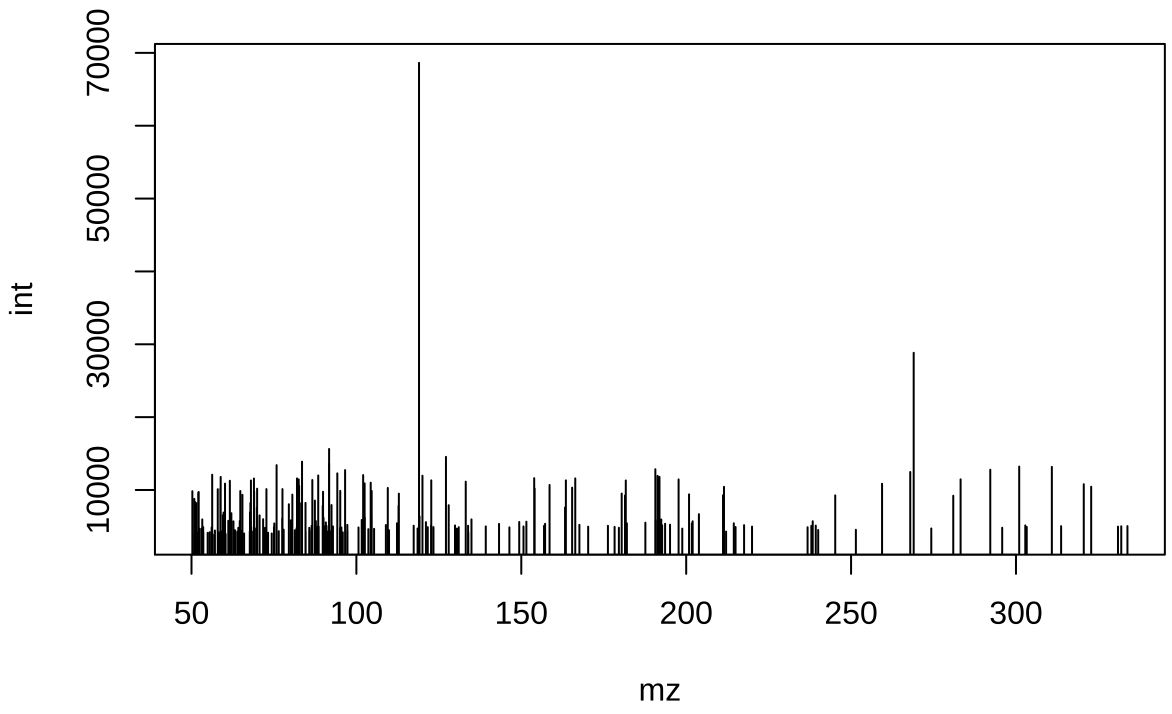
MS1

```
ms1range = c(0.5, 3)
ms1list <- lapply(dat[[i]]$msdata, function(x) {if (x$msn == 1) {matrix(unzip(x$msdata), ncol = 2, byro
  ms1list <- lapply(ms1list, function(x) x[which(x[,1] >= as.numeric(dat[[i]]$peak$mz) - ms1range[1] & 
  ms1list <- ms1list[-which(sapply(lapply(ms1list, nrow), is.null))]
  ms1list <- lapply(ms1list, zipms)
  ms1empirical <- peaktable(ms1list, masserror = as.numeric(dat[[i]]$massspectrometry$msaccuracy))
  ms1empirical <- data.frame(mz = rowMeans(ms1empirical$mass, na.rm = TRUE), int = rowMeans(ms1empirica
  plot(ms1empirical, type = "h")
```



MS2

```
ms2list <- lapply(dat[[i]]$msdata, function(x) {if (x$msn == 2) {x$msdata}})
   ms2list <- ms2list[-which(sapply(ms2list, function(x) length(nchar(x)) == 0))]
   ms2empirical <- peaktable(ms2list, masserror = as.numeric(dat[[i]]$massspectrometry$msaccuracy))
   ms2empirical <- data.frame(mz = rowMeans(ms2empirical$mass, na.rm = TRUE), int = rowMeans(ms2empiri
 plot(ms2empirical, type = "h")
```



Annotations:

```
knitr::kable(dat[[i]]$annotation)
```

fragment_mz	fragment_formula	fragment_SMILES	fragment_radical	fragment_citation
118.9912	C2F5	FC-C(F)(F)F	FALSE	DOI: 10.1002/rcm.3274
268.9828	C5F11	FC(F)(C(F)(F)C(F)(F)F)C(F)(F)C-F	FALSE	DOI: 10.1002/rcm.3274

Step 3) Data Quality Check

To perform a automated check on quality, for each peak

```
qc <- gather_qc(dat[[i]], exactmasses)
```

parameter	reportedmz	compoundmz	value	limit	result
measurederror	312.973	312.9728	0.6026083	5	TRUE

parameter	value	limit	result
ms1_isotopepattern	0.9887309	0.5	TRUE

parameter	reportedmz	measuredmz	msaccuracy	value	result
ms1precursor_detected	312.973	312.9733	5	TRUE	TRUE

parameter	reportedmz	measuredmz	msaccuracy	value	result
annfragments_detected	118.9912	118.9911	5	TRUE	TRUE
annfragments_detected	268.9828	268.9824	5	TRUE	TRUE

parameter	measuredmz	calculatedmz	msaccuracy	minmzerror	mzdiff	error	result
annfragments_accuracy	118.9912	118.9926	5	0.002	-0.0013634	-11.4579902	TRUE
annfragments_accuracy	268.9828	268.9830	5	0.002	-0.0001814	-0.6743926	TRUE

parameter	reportedformula	parentformula	result
annfragments_subset	C2F5	C6HF11O2	TRUE
annfragments_subset	C5F11	C6HF11O2	TRUE

parameter	reportedformula	reported_smiles	calculatedformula	result
annfragments_elementalmatch	C2F5	FC-C(F)(F)F	C2F5	TRUE
annfragments_elementalmatch	C5F11	FC(F)(C(F)(F)C(F)(F)F)C(F)(F)C-F	C5F11	TRUE

write file for future reference

```
saveRDS(dat, 'example/PFAC30PAR_PFCA2_output.RDS')
```