

# Introduction to management and analysis of ion fragmentation data

Mingshu Cao  
Mingshu.Cao@agresearch.co.nz  
AgResearch Limited, Grasslands Research Center  
Palmerston North 4442, New Zealand

October 13, 2014

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Data source</b>	<b>1</b>
<b>3</b>	<b>Construction of ion tree</b>	<b>2</b>
<b>4</b>	<b>Database manipulation</b>	<b>4</b>
<b>5</b>	<b>References</b>	<b>8</b>
<b>6</b>	<b>Acknowledgements</b>	<b>8</b>

## 1 Introduction

A few examples are given on how the 'iontree' package can be used for ion fragmentation data analysis and management. Ion fragmentation data also refer to ion tree or spectral tree data. MS2 and MS3 raw scans (DIMS data, see Cao et al. 2008) were provided in the 'iontreeData' package along with a demo SQLite database built from the same experimental data.

## 2 Data source

MS2RAW and MS3RAW are a list of MS2 and MS3 scans, respectively. They were produced from mzXML data files using a function 'saveMSnRaw' from this package.

```
> rm(list=ls())  
> library(iontree)  
> library(iontreeData)  
> data(MS2RAW)  
> data(MS3RAW)  
> ls()
```

```
[1] "MS2RAW" "MS3RAW"

> length(MS2RAW)

[1] 4

> attributes(MS2RAW[[1]])

$names
[1] "premz" "rt" "msn.sp"

$filename
[1] "dims6.mzXML"
```

Use 'getMetaInfo' function to get all the necessary meta-information.

```
> mzxml=system.file("mzxml/DIMS1.mzXML", package="iontreeData")
> getMetaInfo(mzxml)
```

```
--- General information on the experiment:
```

```
*****
```

```
RT range <min>: 0 10
```

```
mz range: 150 1000
```

```
Total scans: 1360
```

```
MSn scans:
```

```
mslevels
```

```
  1  2  3
```

```
729 262 369
```

```
Polarity: +
```

```
Collision energy: 35
```

```
-----
Manufacturer: Thermo Scientific
```

```
MS Model: LTQ
```

```
Ionisation: ESI
```

```
MS Analyser: ITMS
```

```
MS Detecotr: unknown
```

```
Software: Xcalibur 1.0 SP1
```

```
Centroid: Yes
```

```
*****
```

### 3 Construction of ion tree

An ion tree could be built for a given range of m/z and retention time, the parameters could be obtained from MS1 peak detection of LCMS data or m/z binning of DIMS data. The following example shows an ion tree built from DIMS data in one sample. A full time range is specified for direct infusion mass spectrometry.

```
> mz=554.5
> mzDelta=0.5
```

```

> mzRange=c(mz-mzDelta, mz+mzDelta)
> rtRange=c(0,600)
> idx=hasMS2(MS2RAW, mzRange, rtRange)

3 dims16.mzXML
4 dims17.mzXML

> idx.ms2=idx[1]
> ms2=MS2RAW[[idx.ms2]]
> ms3=MS3RAW[[idx.ms2]]
> tree1=buildIonTree(mzRange, rtRange=c(0, 600), ms2, ms3)

2 ms2 spectra found
Raw MS2 rtRange: [ 235.471 257.535 ]
the number of sibling ms3 spectra found: 1

> tree1

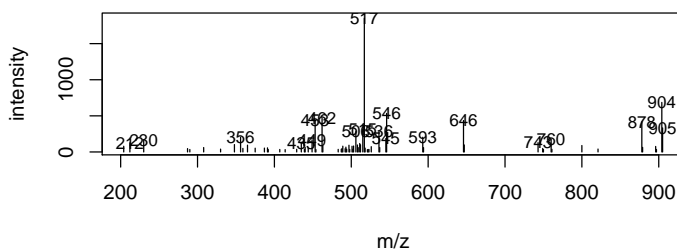
--- iontree object ---

mz: 554 555
rt: 235 258 sec
dim of ms2 matrix: 72 2
num of ms3 spectra: 1

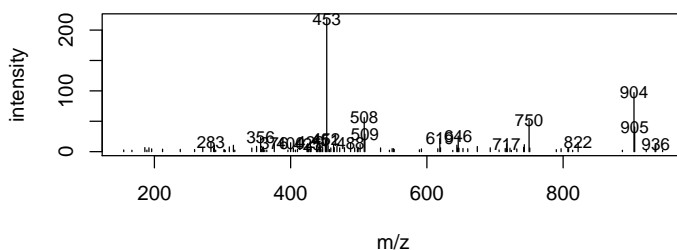
> plot(tree1)

```

**MS2 of premz 554~555 @ time 235~258s**



**MS3 of 517**



## 4 Database manipulation

With assumed nominal mass resolution, batch construction and loading of ion trees into the database can be conducted for DIMS data. Because of the noisy data in the raw MSn scans and binning errors quality control is advised. QC will be discussed in a publication.

A SQLite database 'mzDB.db' was provided in iontreeData package. The following example is to query and plot an iontree of precursor m/z 596 from the database. m/z 596 was annotated as thesinine-rhammoside-hexoside from forage ryegrass (Koulman et al. 2009).

```
> dbname=system.file("db/mzDB.db", package = "iontreeData")
> db=dbConnect(dbDriver("SQLite"), dbname)
> sql="SELECT mz, ms2, ms3 FROM mz WHERE mz=596"
> q1=dbSendQuery(db, sql)
> rs=fetch(q1, n=-1)
> dim(rs)

[1] 1 3

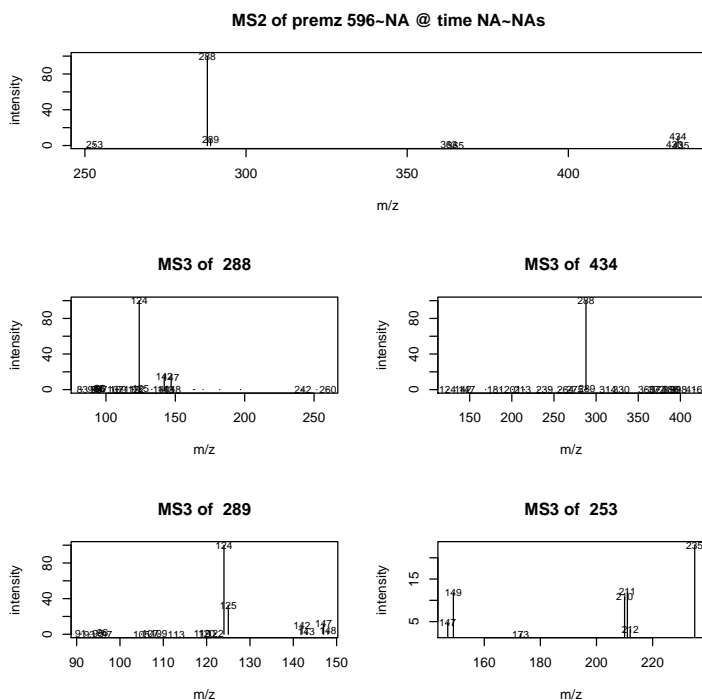
> t1=rs2iontree(rs)
> dbClearResult(q1)

[1] TRUE

> dbDisconnect(db)

[1] TRUE

> plot(t1[[1]])
```



Then we use a MS3 spectrum, derived from m/z 596 with the precursor m/z 434 at the MS2 level, as a query spectrum to search the database. The comparison of MS3 spectrum with MS2 spectra may provide additional structural information on the MS1 ions. A comparison of 3 metrics for MS2 spectral similarity is demonstrated by the output of 'searchMS2' function.

```
> query=t1[[1]]@MS3[[2]]$sp3
> premz=t1[[1]]@MS3[[2]]$prems
> premz
```

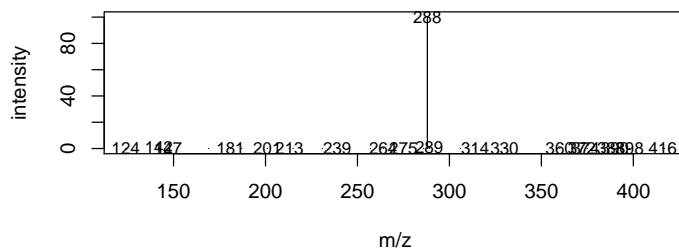
```
ms3.premz
"434"
```

```
> searchMS2(query, premz, dbname, scoreFun="d")
```

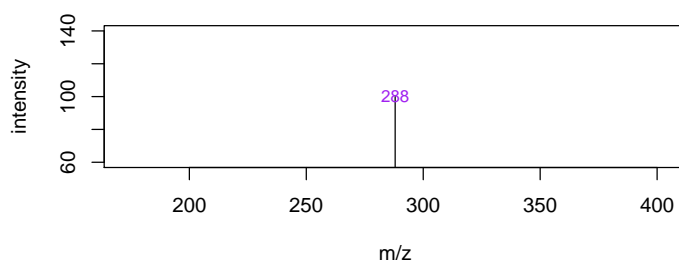
```
most similar spectra by metric: distMS2
```

	exp_id	mz	score
126	1	433	0.00
127	1	434	0.04
202	1	596	0.32
128	1	435	0.45
203	1	597	0.48

**query spectrum from m/z 434**



**top db spectrum from m/z 433**



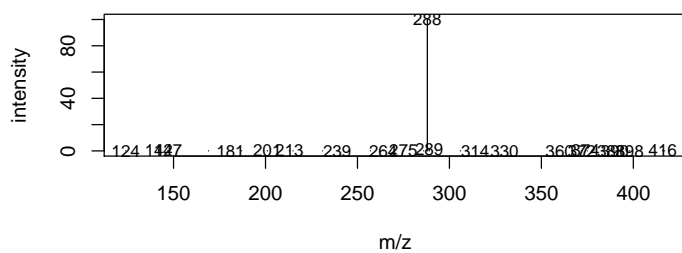
```
> searchMS2(query, premz, dbname, scoreFun="t")
```

```
most similar spectra by metric: tanimoto
```

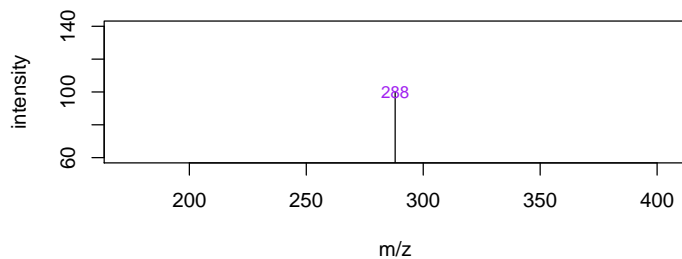
	exp_id	mz	score
--	--------	----	-------

126	1	433	1.00
127	1	434	1.00
128	1	435	1.00
129	1	436	0.33
69	1	289	0.18

**query spectrum from m/z 434**



**top db spectrum from m/z 433**

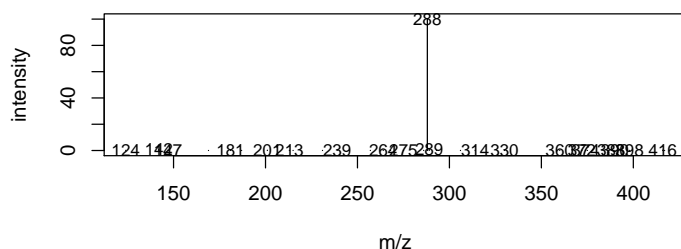


```
> searchMS2(query, premz, dbname, scoreFun="c")
```

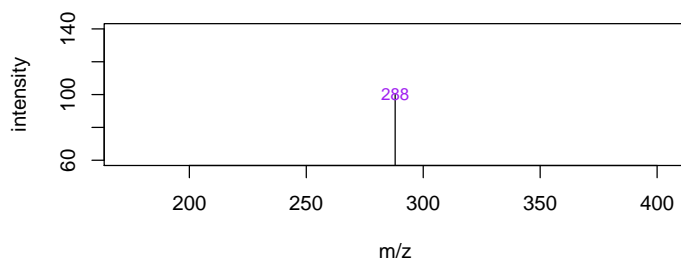
```
most similar spectra by metric: cos
```

	exp_id	mz	score
126	1	433	1.00
127	1	434	1.00
49	1	244	1.00
195	1	585	1.00
172	1	544	0.99

query spectrum from m/z 434



top db spectrum from m/z 433



In summary, although the top ion was correctly retrieved by 3 metrics, correlation-based similarity measurement could fail for sparse MS<sup>n</sup> spectra. For example, the 3rd top-ranked based on cosine was a sparse MS<sup>2</sup> spectrum from m/z 244. The following example shows correlation without direct match of product ions could be misleading. Metric 'distMS<sup>2</sup>' was based on city-block distance (Cao et al 2009), and unnormalised distance score was shown.

```
> dbname=system.file("db/mzDB.db", package = "iontreeData")
> db=dbConnect(dbDriver("SQLite"), dbname)
> sql="SELECT mz, ms2, ms3 FROM mz WHERE mz=244"
> q1=dbSendQuery(db, sql)
> rs=fetch(q1, n=-1)
> dim(rs)

[1] 1 3

> t1=rs2iontree(rs)
> dbClearResult(q1)

[1] TRUE

> dbDisconnect(db)

[1] TRUE

> x=t1[[1]]@MS2
> dim(x)

[1] 5 2
```

```

> y=topIons(query[,1], query[,2], top=nrow(x))
> x

      mz  inten
[1,]  86   3.35
[2,] 169   6.72
[3,] 198 100.00
[4,] 216   2.88
[5,] 226   4.19

> y

      [,1] [,2]
[1,]  142  0.80
[2,]  147  0.40
[3,]  288 100.00
[4,]  289  1.35
[5,]  416  0.29

> round(cor(x[,2], y[,2]),2)

[1] 1

```

## 5 References

- Cao M, Koulman A, Johnson LJ, Lane GA and Rasmussen S. 2008. Plant Physiology. 146:4
- Koulman A, Cao M, Faville M, Lane GA, Mace W and Rasmussen S. 2009. Rapid Communication in Mass Spectrometry. 23.

## 6 Acknowledgements

I would like to thank Karl Fraser, AgResearch for the discussion on the chemical aspects of selected spectra when preparing this document.

```

> sessionInfo()

R version 3.1.1 Patched (2014-09-25 r66681)
Platform: x86_64-unknown-linux-gnu (64-bit)

locale:
 [1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C
 [3] LC_TIME=en_US.UTF-8      LC_COLLATE=C
 [5] LC_MONETARY=en_US.UTF-8  LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8     LC_NAME=C
 [9] LC_ADDRESS=C             LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:

```



```
[1] stats      graphics  grDevices  utils      datasets  methods   base

other attached packages:
[1] iontreeData_1.1.0 iontree_1.12.0  XML_3.98-1.1   RSQLite_0.11.4
[5] DBI_0.3.1         rJava_0.9-6

loaded via a namespace (and not attached):
[1] tools_3.1.1
```