

JRAP Extended

Table of contents

1 Overview.....	2
2 Documentation.....	2
3 First Extension - Get 'Precursor Intensity' information.....	2
4 Second Extension - Get 'Activation Method' information.....	2
5 Third Extension - 64 bits Precision and Compressed Spectrum Ready.....	3
6 Fourth Extension - Empty Peaks Characters [# jrap-ext-1.1].....	3
7 Fifth Extension - Empty Peaks Characters (part 2) [# jrap-ext-1.2].....	3

1. Overview

Note:

Package `jpl-io` depends on library `jrap-ext.jar`.

JRAP is a mzXML parser developed at the [Seattle Proteome Center](#). It is a port of RAP (Random Access Parser C-parser) to java. Here is the [project page](#). It is an open source project.

We have extended JRAP for many reasons: we found bugs, we needed to retrieve more informations from mzXML data and we add new capabilities to the parser (for example, the parser is now able to read compressed spectra).

Here the last version of [jrap-ext-1.2.jar](#).

2. Documentation

The [Javadoc API](#) documents are available.

3. First Extension - Get 'Precursor Intensity' information

JRAP has been extended because of numerous caveats. We have created a new project named JRapExt, an extension of jrap, in a new package named `org.systemsbiology.jrap.extension` :

- `ExtMSXMLParser` extends `MSXMLParser`
- `ExtSAX2ScanHandler`
- `ExtSAX2ScanHeaderHandler`
- `ExtScan` extends `Scan`
- `ExtScanHeader` extends `ScanHeader`

We've modified SAX2 handlers to fetch the `precursor_intensity` of level 2 scan. `ExtScan` now implements 2 new methods `setPrecursorIntensity(float newValue)` and `float getPrecursorIntensity()`.

4. Second Extension - Get 'Activation Method' information

Once again, we had to get another tag from mzXML file added in the 3rd version of this format. In level 2 precursorMz, 'activationMethod' (CID or HCD) is now available from `getPrecursorActivationMethod()` and `setPrecursorActivationMethod()` methods.

5. Third Extension - 64 bits Precision and Compressed Spectrum Ready

The parser is now ready to read compressed spectra and to handle 32/64 bits precision peak values (mz and intensity) as defined in mzXML **peaks** tag: `<peaks precision="32" byteOrder="network" pairOrder="m/z-int" compressionType="zlib" compressedLen="315">`

- Fixing bug (the last intensity value of the peak list was missing !!).
- ExtSAX2ScanHandler now extends SAX2ScanHandler.
- Isolates code specific to bytes management from SAX2ScanHandler into Bytes class.
- Now handles zlib compression via Bytes class (in peak xml element there are parameters `compressionType('zlib'|'none')` and `compressedLen`)
- New methods in Bytes `quadletToFloat(byte[] bytes)` and `octletToDouble(byte[] bytes)` called depending on precision (32/64 bits).

6. Fourth Extension - Empty Peaks Characters [# jrap-ext-1.1]

Bug fixed: the parser crashed when no data found between xml `<peaks>` tags.

7. Fifth Extension - Empty Peaks Characters (part 2) [# jrap-ext-1.2]

Bug fixed: the parser crashed when no data found between xml `<peaks>` tags and nor "byteorder" or "network" parameters in peaks tag was defined.