

**Конвертация форматов хромато-масс-спектров:** *msconvert.exe* (из пакета ProteoWizard, установщик: */INSTALL/pwiz.exe*)

Утилита позволяет конвертировать набор файлов агилентовского формата (.d) в более универсальные форматы - лучше в mzXML.

Можно использовать гуй MSConvertGUI. Можно ее спокойно программно вызывать из кода, типа *ShellExecute("msconvert.exe", 0, "--format mzxml", ...)*.

Кода, в целом, немного. Некоторые пояснения:

## Парсинг mzxml файлов (ChromatogramForm.cs)

Использовались следующие сторонние компоненты:

**1) MSDataFileReader.dll** - для парсинга mzxml-файлов (но, судя по всему, не может делать zlib-декомпрессию, предполагаемую новым форматом mzxml 3.0. Поэтому декомпрессию делаем сами).

В коде:

```
// просто грузим кусок XML, в котором хранится информация по масс-спектру с номером idx
mzXMLParser.GetSourceXMLByScanNumber(idx, ref xmlString);

System.Xml.XmlDocument doc = new System.Xml.XmlDocument();
doc.LoadXml(xmlString);

// find the encoded and compressed binary data in xml file: the <peaks> tag
string peaks = doc.SelectSingleNode(@"//peaks").InnerText;

// Сначала декодируем массив байт из base64 строки
byte[] byteArray = System.Convert.FromBase64String(peaks);

// ... и потом еще переворачиваем байты (little endian)

// а потом декомпресс

// функция получает масс-спектр с номером скана idx
private MZSpectrum GetMZSpectrumByIndex( int idx )

// класс масс-спектра. По сути просто обернул два массива вещественных чисел: m/z и интенсивности пиков

class MZSpectrum
{
    public double[] MZList;
```

```

public double[] intensityList;

public MZSpectrum() { }

public int PeakCount() { return MZList.Count(); }
}

```

## 2) ICSharpCode.SharpZipLib - для декомпрессии данных из zlib

в коде:

```
public static byte[] DecompressZlib(Stream source)
```

## DIMS (MainForm.cs)

ДИМСы считываются из csv-файлов и хранятся здесь:

```
private List<List<int>>> spectra = new List<List<int>>>();
```

Если в csv-файле есть еще второй столбец с x-координатами, они хранятся здесь

```
private List<List<float>>> spectral_points = new List<List<float>>>();
```

Метаинформация идет первым элементом в файле и парсится с помощью регулярки

```

// parse measure_params using RegExp
var col = System.Text.RegularExpressions.Regex.Matches(measure_params[nSerie],
@"(?<key>\s*\w+[\s,]*\w+\s*)=(?<val>\s*\d*[\s,]?\d+\s*)");

```