

PIDC v. 2.0 beta

USER MANUAL

Ari Rantanen
University of Helsinki,
Department of Computer Science
E-mail: ari.rantanen@cs.helsinki.fi

Table of contents

INTRODUCTION	2
INSTALLATION	2
RUNNING PIDC V.2	2
SETTING UP FRAGMENT DATABASE.....	3
SETTING UP THE PARAMETERS	4
COMPUTATION OF POSITIONAL ISOTOPOMER DISTRIBUTION	5
COMBINING CONSTRAINTS OBTAINED FROM DIFFERENT RUNS.....	5

Introduction

This document gives an introduction to PIDC v.2 (Positional Isotopomer Distribution Calculator), a software package that computes a positional isotopomer distribution of the selected atom type of the compound when mass spectra of the compound and its fragments are given. PIDC can analyze mass spectrometric data originating from MS/MS measurements when daughter ion scanning mode is used in the fragmentation stage. At this stage PIDC v.2 does not support full scanning mode. If the full scanning is used an earlier version of the software can be applied.

A more verbose description to the computational methods applied in the PIDC v.2 is given in article “Isotopomer distribution computation from tandem mass spectrometric data with overlapping fragment spectra” [1]. PIDC-application runs in the MATLAB 6® - environment (<http://www.mathworks.com/products/matlab>). PIDC v.2 might also work with the previous versions of MATLAB, but this can not be guaranteed or supported. PIDC is tested in the Windows® and Linux operating systems. PIDC v.2 is distributed as a collection of MATLAB’s M-source files. These source files or part of them can freely be used in other freely distributed applications. However, it is prohibited to redistribute edited source files under name PIDC v.2 without permission. PIDC v.2 is a research prototype whose commercial use is prohibited without permission from the authors. It is distributed “as is” and authors do not take any responsibility of any harm or damage caused by PIDC. Notifications of the erroneous function of PIDC as well as suggestions for improvements can be sent to the e-mail address ari.rantanen@cs.helsinki.fi.

Installation

After downloading a zip-compressed installation package of PIDC v.2 (PIDC2.zip) extract files to a new directory. The extraction can be done using popular compression utilities such as Winzip or pkunzip in Windows or unzip in Linux. After extraction the M-files needed by PIDC can be found from the created directory together with one example input file, *input.txt* and example database file *fdb_example_30.db*. After the extraction of the installation package start MATLAB. There exists a drop down list in the main window of the MATLAB titled “*current directory*”. Input there a complete path to the directory where the M-files were extracted. Now PIDC is ready to run.

Running PIDC v.2

PIDC v.2 is started by typing “*PIDC2*” to a command window of MATLAB. After a short while the main window of PIDC v.2 shows up. From this window you can specify the name of the input and output files, select a MS device used in measurement, identify the peaks of input file, browse the identified peaks and the database containing fragment patterns of different molecules, start the computation of positional isotopomer distribution, browse the computed isotopomer distribution and combine the constraints

computed to isotopomer distribution of the same sample. In following subsections the process of computation is described with more details.

Setting up fragment database

Before PIDC v.2 can compute a positional isotopomer distribution of the carbon chain of a given compound, the fragmentation pattern of the compound must be defined in the database. Each different experimental setting (MS-device, energy used in the fragmentation) has its own database file. The name of the database file specifies the experimental settings. The format of the name of the database file is

$$fdb_{\textit{name of device}}_{\textit{energy level used}}.db$$

For example, a database file *fdb_A_15.db* corresponds to database of MS-device A with 15 eV of energy.

The database files are standard text files that can be edited with MATLAB's internal editor, for example. For each compound to be analyzed the database contains a block of rows, one row for mother ion and for each different fragment of an ion. Each row consists of six columns, separated by tabulator characters. First column contains an abbreviation of the name of the compound. Second column contains "M" for mother ion and "D" for daughter fragments. Third column has a chemical formula of the mother or fragment, fourth specifies which carbons of the mother are included in the fragment. Fifth column tells the mass of the mother ION or fragment. Sixth column specifies a relative abundance of the daughter ion fragment when the most abundant fragment gets the value of 100. Please note that for mother ions sixth column is left blank. Below we can see an example of the block in the database corresponding H⁺ ion of alanine and its three fragments.

ALA M C3H8NO2	C1C2C3	90			
ALA D C2H3	C3C2	27	6.0		
ALA D CH3N	C2	29	5.6		
ALA D C2H6N	C3C2	44	100		

In addition to database file also program file *initializeGlobals.m* have to be updated when new compound is added. For each compound there have to be an entry in *AminoAcids* data structure (never mind the name, compounds do not have to be amino acids!). Currently there exists 37 entries in data structure, last being "PyA". Now if the compound named "XXX" with molecular weight of 777 are to be added, data structure have to be updated as follows:

```
AminoAcids(36).name = 'ATP';  
AminoAcids(36).mass = 507;  
AminoAcids(37).name = 'PyA';  
AminoAcids(37).mass = 88;  
AminoAcids(38).name = 'XXX';  
AminoAcids(38).mass = 777;
```

Please note that every compound has its own index in the structure (37 for PyA, 38 for XXX).

If the new compound contains atoms not listed in the *Elements* data structure of *initializeGlobals.m* also this structure has to be updated. The updating process is similar to the update of *AminoAcids* structure. Each atom type has its own index in *Elements*. Field *name* contains the abbreviation of the name of the atom, *weight* the base mass of the atom and the vector *isotopes* the distribution of natural abundances of different isotopes of the atom, from lighter to heavier isotopes.

(Yes, we know that this updating process is cumbersome. We might do something to it but do not hold your breath ;))

Setting up parameters

The input files to PIDC v.2 contain MS/MS measurement data in the following format. For each peak there exists a row in the input file. Each line has three columns. First column gives the m/z value of the peak (+1 or -1 for z is assumed). Second column gives the intensity of the peak. Third column specifies if the peak corresponds to mother ion or daughter ion fragments. If the peak corresponds to mother ion, third column should contain *M* character. If the peak corresponds to daughter ion fragment, third column contains an integer specifying the mass difference between the fragmented mother ions and the base mass of the mother. For example, if the peak corresponds to daughter ions originating from mother ions of alanine having m/z value of 91 in positive scanning mode, the third column should contain number 1 = 91 – 90. Below is an example of an input file corresponding to fragmentation of alanine in positive scanning mode.

90	743899136	M
91	113999872	M
92	6172928	M
44	123844	1
45	174496	1
27	8794	1
28	13993	1
29	14401	1
44	50044	2
45	36667	2
46	21638	2
47	13342	2

The input file used can be selected from the user interface of PIDC v.2 by typing the name of the file to the box titled “*input file*” or by selecting the file from the directory tree that can be opened with “...” button. The input file can be opened in the text editor of MATLAB by pressing “*Edit*” button.

The output file written by PIDC v.2 contains the date of an analysis, name of an MS-device, energy level used, copy of an input data and found linear constraints to the positional isotopomer distribution of the carbon chain of the compound. In the constraints ‘0’ denotes a ¹²C carbon and ‘1’ ¹³C or labelled carbon. For example, [1 0 0] denotes an isotopomer of a three carbon compound having a label in its first carbon. The name of an output file can be specified in the user interface similarly to the input file.

The MS-instrument and energy level can be selected using pull down lists. The lists are populated with information extracted from the names of database files. Positive or negative ionization mode is selected with radio buttons.

Computation of positional isotopomer distribution

When the parameters are set the peaks of the input file are recognized with the help of appropriate database file by pushing the “*Recognize peaks*” button. The recognized peaks can be browsed by selecting the peak from “*Mass/Charge*” drop down list. Corresponding fragments are the showed in “*Source*” list. If some peaks are not identified a warning is showed in “*Messages*” field.

Computation of positional isotopomer distribution is started by pushing “*Analyze*” button. After a while “*Messages*” field shows text “*OK*” and the result is saved to output file. The result can be checked immediately by pressing “*Edit*” button next to field specifying the name of output file. The isotopomers having zero abundance are not showed in the result.

Combining constraints obtained from different runs

Sometimes the same sample is analyzed with different MS-device. Different settings can produce different constraints to the isotopomer distribution of the sample than can be combined to get more information about the distribution. In PIDC v.2 different constraints are combined by pressing the button “*Add to Stored Constraints*” after the computation of each individual constraint set. That is, after pressing the “*Analyze*” button, press “*Add to Stored Constraints*” if you want this constraint set to combined with other constraints corresponding the sample but different measurement settings. Constraints to isotopomer distribution defined by all combined constraints can be computed by pushing “*Compute Combined Constraints*” button. The combined results are also saved to the text file specified by “*Output file*” field.

[1] Juho Rousu, Ari Rantanen, Raimo Ketola, Juha Kokkonen: Isotopomer distribution computation from tandem mass spectrometric data with overlapping fragment spectra. *Spectroscopy* Vol. 19 (2005), 53 – 67.