ICT - Isotope Correction Toolbox

Version 0.04

Christian Jungreuthmayer and Jürgen Zanghellini
Austrian Centre of Industrial Biotechnology (ACIB)
Metabolic Modeling Group
Muthgasse 11/DG
1190 Vienna, Austria
christian.jungreuthmayer@boku.ac.at

Teresa Mairinger, Stefan Neubauer and Stephan Hann
University of Natural Resources and Life Sciences Vienna (BOKU)
Department of Analytical Chemistry
Muthgasse 18
1190 Vienna, Austria
teresa.mairinger@boku.ac.at

August 4, 2015

Contents

1 Bug reports

Please send bug reports, comments, questions, and feedback about *ICT* to `christian.jungreuthmayer@boku.ac.at`.

2 Background and introduction

Many of the biggest health issues are related to metabolism, such as cancer, obesity, and diabetes. In order to overcome these problems, a thorough and in-depth understanding of metabolic processes is required. Isotope labeling experiments play a crucial role in exploring, studying and understanding metabolic pathways in living cells. However, these experiments suffer from the natural abundance of heavy isotopes which results in an interference of the measured mass spectra. In recent years, a number of scientific articles have been published that address the problem of correcting measured mass spectral data [1–4]. There exist several software tools which are able to compute the contribution from the natural abundant isotopes on the measured intensities, and hence are capable to correct the measured data.

Pynac [4,5] is a correction software written in the programming language Python. *Pynac* benefits from the multi-platform capabilities of Python, is highly flexible by using text files for the configuration and setup, and can account for any number of isotopes. However, it is not able to correct data from tandem mass spectrometry experiments.

PDIC [6,7] is an isotope correction tool written in *Matlab*. *PDIC* comes with a rich feature set including the ability to handle isotopomers of different fragments which overlap in the spectra. *PDIC* relies on the concept of conditional probabilities which are solved by utilizing Bayes' theorem. However, in order to run *PDIC* a valid license of the commercial software *Matlab* is required. Furthermore, *PDIC* is not able to consider the natural abundance and the (im)purity of the tracer.

LS-MIDA [8] is a tool to efficiently analyze isotopomer distributions in metabolic modelling. *LS-MIDA* is only free for academic users and is written in the Microsoft C# programming language. Hence, computers running a Microsoft operating systems are the primary target platform of *LS-MIDA*. However, C# applications also run on other operating systems, e.g. on Linux systems with the help of the open source .Net platform Mono [9]. *LS-MIDA* provides a graphical user interface (GUI) which might be an advantage for less proficient computer users. The main purpose of *LS-MIDA* is the analysis of full scan mass spectrometry and precursor unselective fragmentation experiments (e.g. GC-EI-MS), but *LS-MIDA* cannot handle precursor selective fragmentation.

Another prominent available program is *IsoCor* [10,11]. *IsoCor* is implemented in Python and is a fast and excellent tool to correct mass spectrometry data. It supports the correction of the natural abundance of the tracer at unlabeled positions and allows to take into account the isotopic purity of the labeling source. Furthermore, in *IsoCor* any isotope and not only ^{13}C can be used as tracer. Though, *IsoCor* is not able to deal with tandem mass isotopomer data.

We present a software tool named *ICT* (*Isotope Correction Toolbox*) which is able to deal with ion fragmentation and, hence, can correct mass spectra obtained by tandem mass spectrometry. The software program *ICT*, is written in the programming language Perl. Perl runs on all commonly available operating systems, such as Linux/Unix, BSD, Mac OS, and Windows. Hence, *ICT* can be used on virtually any computer platform. *ICT* is a command line tool which allows to perform easy corrections of batches of mass spectrum data. The input data (chemical composition of the fragments, the natural abundance, and the measured intensities) are simply provided in form of plain text files. After the successful correction of the measured data, the emended mass spectra are written to a text files, as well. *ICT* computes the probability of all relevant sets of isotope combinations for the given chemical composition of the precursor and the fragment. Based on these probabilities the correction matrix CM is generated, which relates the defective measured intensities with the corrected data: $CM \times I_{corr} = I_{meas}$. Solving this system of equations yields the desired corrected intensities I_{corr}. *ICT* is open source software and freely available under either the Artistic License [12] or the GNU General Public License [13].

3 Getting *ICT*

ICT is available as compressed software archive on request from the authors. It can also be downloaded from `http://github.com/jungreuc/XXX`. *ICT* is provided in two archive formats: (i) as gzipped tar archive, e.g. `ict-0.04.tar.gz` and (ii) as (win)zipped archive, e.g. `ict-0.04.zip`.

4 Installing *ICT*

4.1 Prerequisites

ICT requires that Perl is installed on the target platform. On Linux/Unix systems Perl is typically installed by default. If this is not the case, install Perl with your favorite package manager.

On Windows system we recommend to install and to use Strawberry Perl - which can be downloaded from `http://strawberryperl.com/` - as we tested *ICT* on Windows solely with Strawberry Perl.

4.2 Installing *ICT* under Linux/Unix

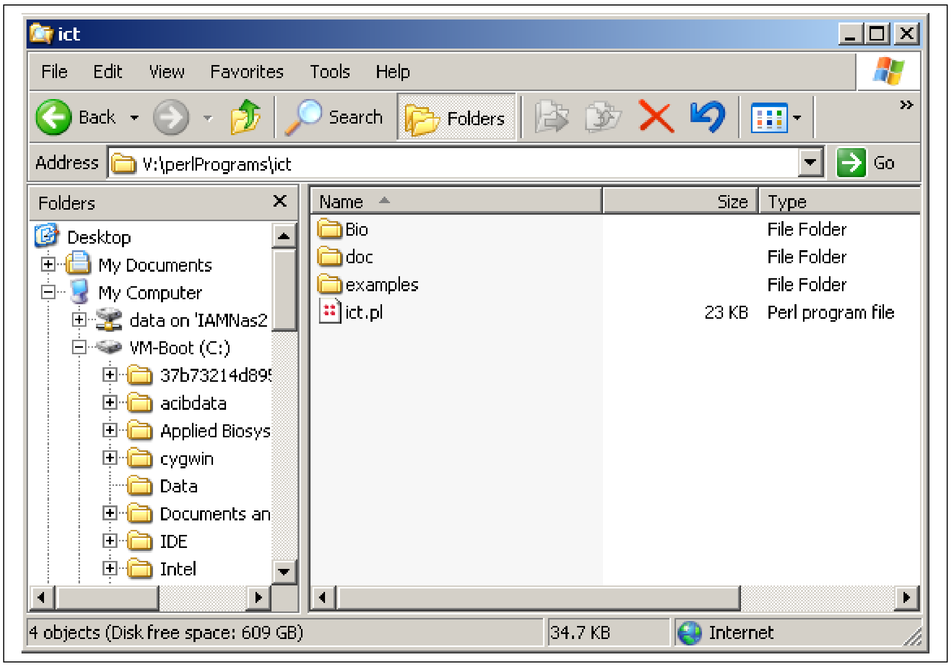
After downloading the gzipped tar archive, create the directory where *ICT* is to be installed (e.g. using the Unix command `mkdir`), and copy the archive file to this directory (e.g. using the Unix command `cp`). Change to this directory (e.g. using the Unix command `cd`) and, finally, extract the archive by executing the following command:

```
tar -xvzf ict-0.04.tar.gz
```

4.3 Installing *ICT* under Windows

After downloading the zipped archive, create the directory where *ICT* is to be installed (e.g. using the Windows Explorer) and copy the archive file to this directory. Finally, extract the archive with your favorite compression/zip tool, e.g. WinZip.

4.4 File and directory structure



Address: V:\perlPrograms\ict

Name	Size	Type
Bio		File Folder
doc		File Folder
examples		File Folder
ict.pl	23 KB	Perl program file

4 objects (Disk free space: 609 GB) | 34.7 KB | Internet

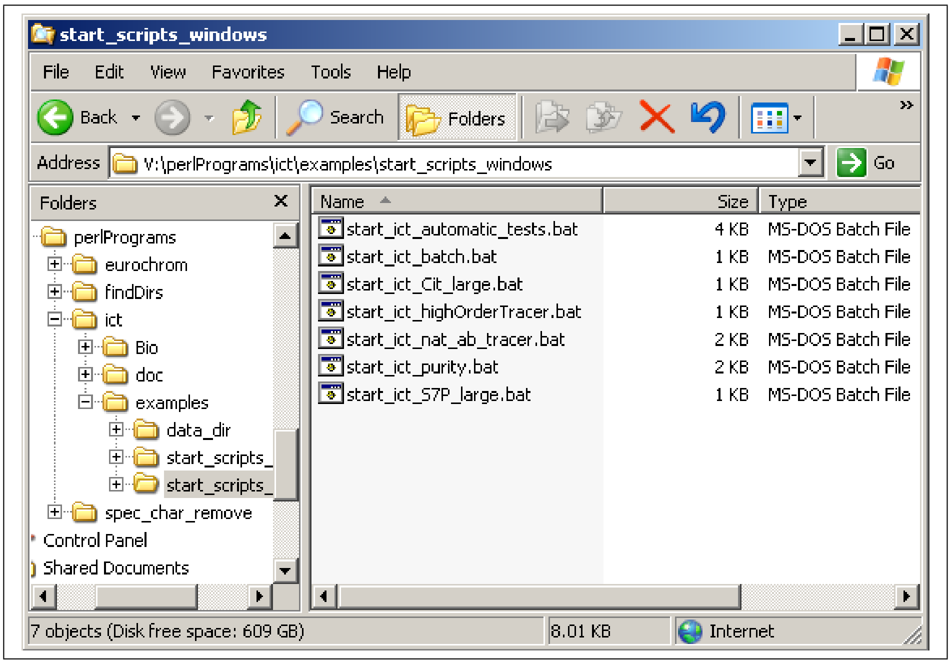
Figure 1: Using the Windows Explorer to display *ICT* 's root directory after the installation.

After the extraction of the archive, the following files and directories can be found in *ICT*'s root directory (see also figure 1):

```
Bio/
doc/
```

```
examples/
ict.pl
```

The file `ict.pl` is the Perl program that has to be executed to perform the correction of the mass spectrum. As mentioned above, Perl needs to be installed on the computer before `ict.pl` can be used. The program `ict.pl` uses two Perl modules (`Bio::IsotopeCorrection::Chemicals` and `Bio::IsotopeCorrection::NaturalIsotopes`) which are located in the directory `Bio/IsotopeCorrection/`. The *ICT* manual can be found in the `doc/` directory. *ICT* comes with numerous examples which can be found in the `examples/` directory. The input data of the examples are stored in `examples/data_dir/`. The examples contain manually created test data and real-world data of actual tandem mass spectrometry measurements. The directories `examples/start_scripts_linux/` and `examples/start_scripts_windows/` contain example start scripts for Linux/Unix and Windows systems, respectively (see figure 2).



Address: V:\perlPrograms\ict\examples\start_scripts_windows

Name	Size	Type
start_ict_automatic_tests.bat	4 KB	MS-DOS Batch File
start_ict_batch.bat	1 KB	MS-DOS Batch File
start_ict_Cit_large.bat	1 KB	MS-DOS Batch File
start_ict_highOrderTracer.bat	1 KB	MS-DOS Batch File
start_ict_nat_ab_tracer.bat	2 KB	MS-DOS Batch File
start_ict_purity.bat	2 KB	MS-DOS Batch File
start_ict_S7P_large.bat	1 KB	MS-DOS Batch File

7 objects (Disk free space: 609 GB) | 8.01 KB | Internet

Figure 2: Using the Windows Explorer to show the content of the directory containing Windows batch scripts for starting *ICT*.

5 Using *ICT*

ICT is a command line tool. Input data are provided as plain text files (see chapter 6 for details) and the corrected data are written to a text file, as well. *ICT* supports batch execution that allows to perform easy corrections of large sets of mass spectrometry data in a single run. Basic information about the usage of *ICT* can be obtained by starting the tool with the command line option `-h`: `ict.pl -h`. The following message will be displayed:

```
ict.pl -c chem_data -m measured_tof [-o output_file -i nat_isotope_data -e expected_tof -k -n -h -p purity_file]

Version:  0.04
Author:  Christian Jungreuthmayer
Year:  2014 - 2015

-c .....  name of input file containing chemical information about precursor and fragment
-m .....  name of input file containing measured mass spectrum values
-o .....  name of output file corrected mass spectral values are written to
-i .....  name of input file containing chemical/physical information about natural isotopes
-p .....  name of input file containing information about purity of tracer
-n .....  natural abundance of tracer (of not labeled fraction)
-k .....  keep (do not remove) temporary files
-e .....  name of input file containing expected corrected spectrum values (used for checking this program)
-h .....  print this message
```

ICT uses two mandatory command line parameters: (i) parameter `-c` to provide the name of the file containing the chemical composition of the *Precursor* and the *Product Ion* and (ii) parameter `-m` to provide the name of the file containing the measured mass spectrum data. Hence, a typical run of *ICT* might be started the following way if the file `chemBotfanin.txt` and `measMassSpectrum.txt` contain the chemical composition information and the measured spectral data, respectively:

```
ict.pl -c chemBotfanin.txt -m measMassSpectrum.txt
```

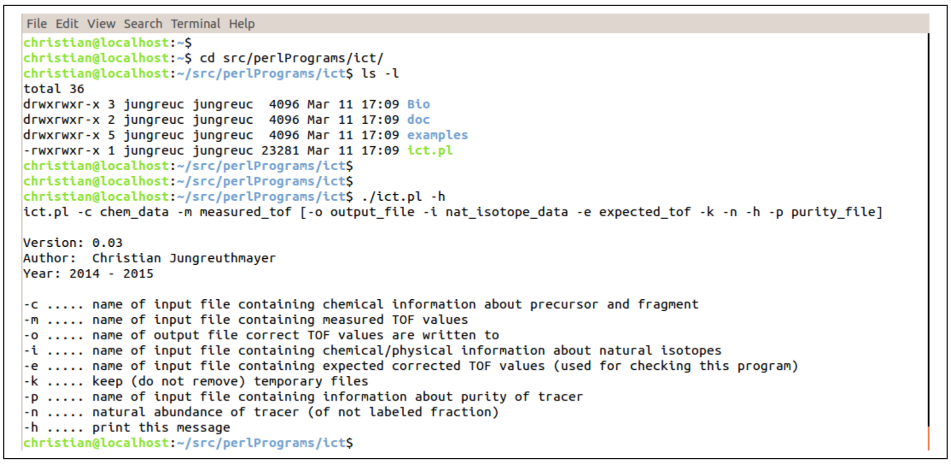
Optionally, the user can also provide the natural isotope abundance to be used by *ICT* during the correction procedure. This is done by using the parameter `-i` and providing the name of the file containing the natural abundance data. The name of the output file that *ICT* will create can be determined by using the parameter `-o`. If the natural abundance data are not provided by the user, an internal data set of abundance values is used by *ICT*. Furthermore, *ICT* supports the parameter `-p` which allows to compute the influence of the tracer purity on the mass spectrum (see chapter 7.3 for further information). Details about the file formats can be found in chapter 6 of this document. The parameter `-n` can be used to simulate the influence of the natural abundance of the tracer element at potentially labeled positions which are not occupied by tracer isotopes. See chapter 7.2 for further information about the natural abundance feature.

The parameter `-k` can be used to prevent *ICT* to delete the temporary files that are created during the correction of the measured mass spectrum values (see chapter 7.7). This is useful if one needs to debug *ICT*. Furthermore, there is the parameter `-e` which can be used to compare the results computed by *ICT* with already known (and expected) results. This might be useful if one needs to extend the functionality of *ICT* and wants to verify that existing parts have not been changed by the extension. Have a look at the examples in the directories `examples/data_dir/Comp17/` to `examples/data_dir/Comp25/` to learn more about the feature `-e` and the required file format.

Note that *ICT* comes with a decent set of examples, which can be found in the `examples/` directory. The example directory contains three subdirectory (i) `data_dir/` which contains all data required to run the examples, (ii) `start_scripts_linux/` which contains scripts to execute the examples in a Linux/Unix environment, and (iii) `start_scripts_windows/` which contains Windows batch scripts to run the examples in a Windows environment.

5.1 Running *ICT* under Linux/Unix

Running *ICT* under Linux/Unix is usually done by opening a terminal window, changing to the directory where *ICT* is installed, and running the required command (see figure 3).



```
christian@localhost:~$
christian@localhost:~$ cd src/perlPrograms/ict/
christian@localhost:~/src/perlPrograms/ict$ ls -l
total 36
drwxrwxr-x 3 jungreuc jungreuc  4096 Mar 11 17:09 Bio
drwxrwxr-x 2 jungreuc jungreuc  4096 Mar 11 17:09 doc
drwxrwxr-x 5 jungreuc jungreuc  4096 Mar 11 17:09 examples
-rwxrwxr-x 1 jungreuc jungreuc 23281 Mar 11 17:09 ict.pl
christian@localhost:~/src/perlPrograms/ict$
christian@localhost:~/src/perlPrograms/ict$
christian@localhost:~/src/perlPrograms/ict$ ./ict.pl -h
ict.pl -c chem_data -m measured_tof [-o output_file -i nat_isotope_data -e expected_tof -k -n -h -p purity_file]

Version: 0.03
Author:  Christian Jungreuthmayer
Year: 2014 - 2015

-c ..... name of input file containing chemical information about precursor and fragment
-m ..... name of input file containing measured TOF values
-o ..... name of output file correct TOF values are written to
-i ..... name of input file containing chemical/physical information about natural isotopes
-e ..... name of input file containing expected corrected TOF values (used for checking this program)
-k ..... keep (do not remove) temporary files
-p ..... name of input file containing information about purity of tracer
-n ..... natural abundance of tracer (of not labeled fraction)
-h ..... print this message
christian@localhost:~/src/perlPrograms/ict$
```

Figure 3: Using a Linux terminal to execute *ICT*

Of course, the usage of start scripts simplifies the repeated execution of *ICT* significantly. Typical start scripts can be found in the directory `examples/start_scripts_linux/`. The content of such a start script file might look as follows:

```
 ../../ict.pl -c ../data_dir/M8T_large/chem_data.txt
-m ../data_dir/M8T_large/measured_tof.txt
-i ../data_dir/M8T_large/nat_isotope.txt
-o ../data_dir/M8T_large/corrected_tof.txt
```

Note that the content of the above start scripts needs be written in a single line. If the command is spread over multiple lines, the character \ must be used to signal that the next line is still part of the command:

```
 ../../ict.pl -c ../data_dir/M8T_large/chem_data.txt \
-m ../data_dir/M8T_large/measured_tof.txt \
-i ../data_dir/M8T_large/nat_isotope.txt \
-o ../data_dir/M8T_large/corrected_tof.txt
```

5.2 Running *ICT* under Windows

Running *ICT* under Windows is usually done by creating Windows batch scripts files. Check out the directory `examples/start_scripts_windows/` which contains several example batch scripts. A typical batch file might look as follows:

```
..\..\ict.pl -c ..\data_dir\purity\chem_data6.txt
-i ..\data_dir\purity\nat_isotope6.txt
-m ..\data_dir\purity\measured_tof6.txt
-o ..\data_dir\purity\corrected_tof6.txt
-p ..\data_dir\purity\purity_data6.txt -n
```

Running a batch file can easily be done by either double-clicking the file or by opening the context menu with pressing the right button on the mouse and selecting the list item *Open*. Again, the entire command must be written in a single line. However, Windows operating systems use the caret symbol ^ to indicate that the command continues in the following line.

Of course, there is also the option to run *ICT* on the command line interface of Windows. To do so, first, open the command line interface window of Windows by pressing the *Start* button (see figure 4) and, next, run the command `cmd.exe` by using the *Run ...* item (see figure 5). Note that on Windows 8 systems the graphical user interface has changed dramatically and the command line interface can be launched by other ways, e. g. searching for the application `cmd.exe` using Window's central search dialog or opening the Explorer, navigating to `C:\Windows\System32` and double-clicking on `cmd.exe`.

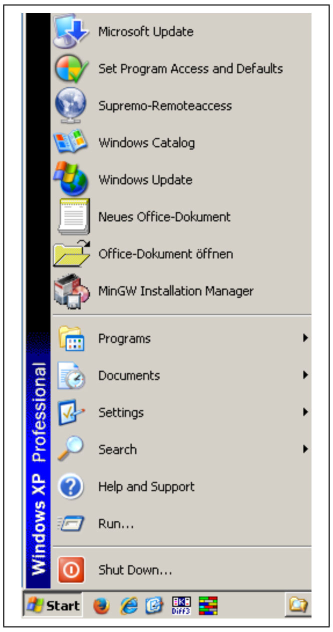


Figure 4: Pressing the Windows *Start* button opens a context menu that allows to run any command by clicking on *Run ...*.

Then, move to the partition (e.g. `V:`) and change to the directory (e.g. using the command line instruction `cd`) where *ICT* is installed (see figure 6).

In order to start *ICT* simply type the command `ict.pl -h` and press the RETURN button (see figure 7).

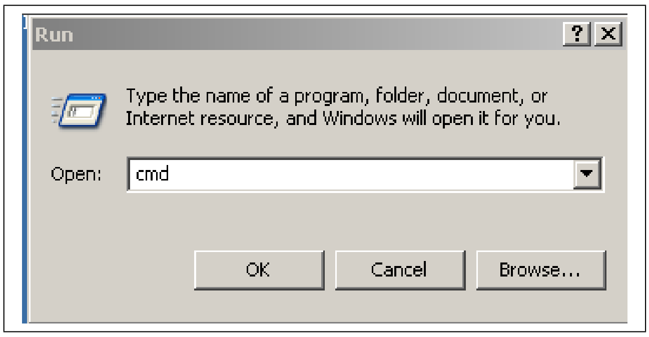
Figure 8 shows an example how the correction of the measured mass spectrum values is performed using the Windows command line interface.

6 File formats

6.1 General remarks

Note that all input files may contain empty lines which are simply ignored. Furthermore, *ICT* treats all lines starting with a hash character `#` as comments. Comment lines are not processed and are also always ignored by *ICT*. Hash characters only indicate a comment, if they are placed at the beginning of a line.

Some of the data fields of the input files need to be separated by commas '`,`' (see the following sub-chapters). If commas are required, the readability of the files can be increased by additionally using space or tabulator characters. *ICT* also tries to be tolerant when fields need to be separated by white

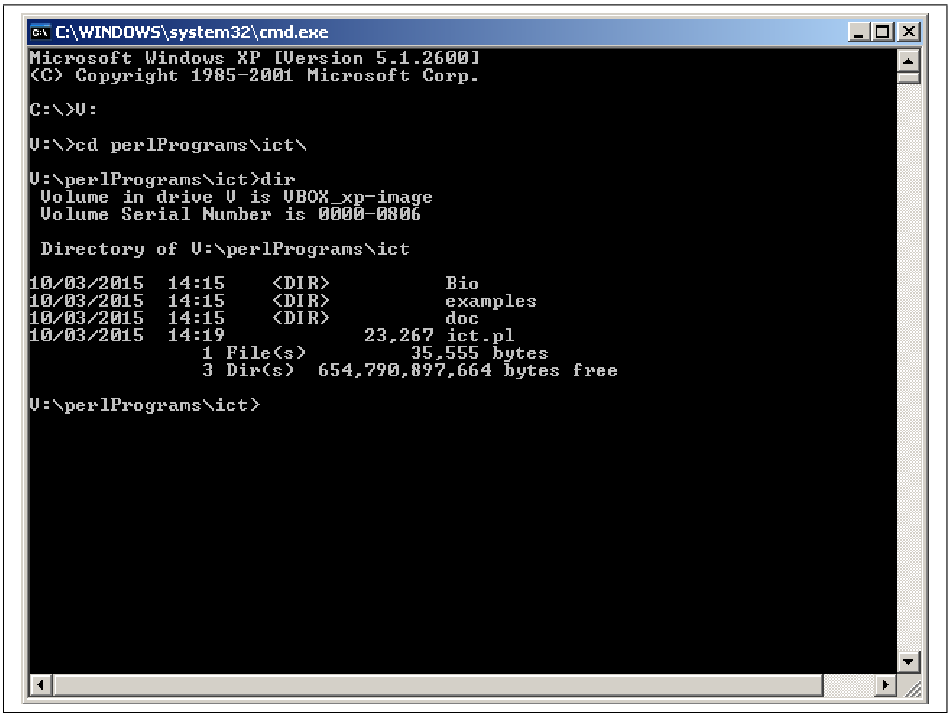


Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: cmd

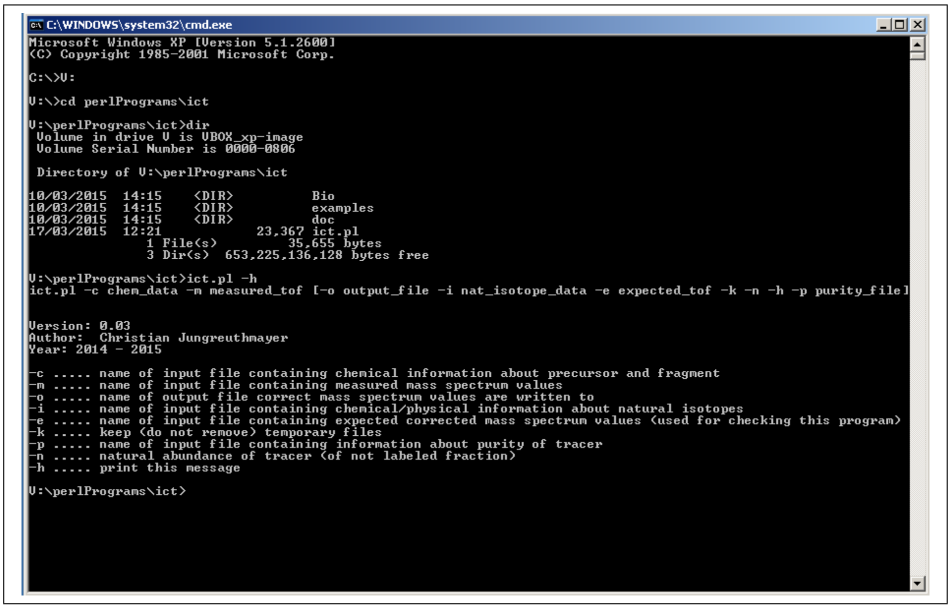
Figure 5: Executing the Windows command `cmd` or `cmd.exe` starts the Windows command line interface window.

.



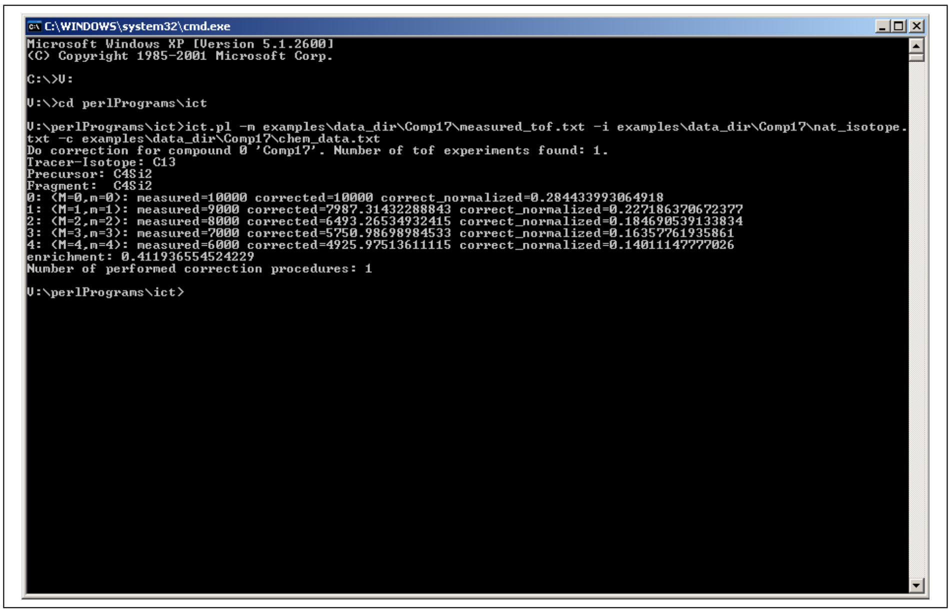
```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\>V:

V:\>cd perlPrograms\ict\

V:\perlPrograms\ict>dir
 Volume in drive V is VBOX_xp-image
 Volume Serial Number is 0000-0806

 Directory of V:\perlPrograms\ict

10/03/2015  14:15    <DIR>          Bio
10/03/2015  14:15    <DIR>          examples
10/03/2015  14:15    <DIR>          doc
10/03/2015  14:19            23,267 ict.pl
               1 File(s)         35,555 bytes
               3 Dir(s)  654,790,897,664 bytes free

V:\perlPrograms\ict>
```

Figure 6: Using the Windows command line interface to display *ICT* 's root directory.



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\>V:

V:\>cd perlPrograms\ict

V:\perlPrograms\ict>dir
 Volume in drive V is VBOX_xp-image
 Volume Serial Number is 0000-0806

 Directory of V:\perlPrograms\ict

10/03/2015  14:15    <DIR>          Bio
10/03/2015  14:15    <DIR>          examples
10/03/2015  14:15    <DIR>          doc
17/03/2015  12:21            23,367 ict.pl
               1 File(s)         35,655 bytes
               3 Dir(s)  653,225,136,128 bytes free

V:\perlPrograms\ict>ict.pl -h
ict.pl -c chem_data -m measured_tof [-o output_file -i nat_isotope_data -e expected_tof -k -n -h -p purity_file]


Version: 0.03
Author:  Christian Jungreuthmayer
Year: 2014 - 2015

-c ..... name of input file containing chemical information about precursor and fragment
-m ..... name of input file containing measured mass spectrum values
-o ..... name of output file correct mass spectrum values are written to
-i ..... name of input file containing chemical/physical information about natural isotopes
-e ..... name of input file containing expected corrected mass spectrum values (used for checking this program)
-k ..... keep (do not remove) temporary files
-p ..... name of input file containing information about purity of tracer
-n ..... natural abundance of tracer (of not labeled fraction)
-h ..... print this message

V:\perlPrograms\ict>
```

Figure 7: Executing *ICT* using the Windows command line interface.



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\>V:

V:\>cd perlPrograms\ict

V:\perlPrograms\ict>ict.pl -m examples\data_dir\Comp17\measured_tof.txt -i examples\data_dir\Comp17\nat_isotope.
txt -c examples\data_dir\Comp17\chem_data.txt
Do correction for compound 0 'Comp17'. Number of tof experiments found: 1.
Tracer-Isotope: C13
Precursor: C4Si2
Fragment:  C4Si2
0: (M=0,m=0): measured=10000 corrected=10000 correct_normalized=0.284433993064918
1: (M=1,m=1): measured=9000 corrected=7987.31432288843 correct_normalized=0.227186370672377
2: (M=2,m=2): measured=8000 corrected=6493.26534932415 correct_normalized=0.184690539133834
3: (M=3,m=3): measured=7000 corrected=5750.98698984533 correct_normalized=0.16357761935861
4: (M=4,m=4): measured=6000 corrected=4925.97513611115 correct_normalized=0.14011147777026
enrichment: 0.411936554524229
Number of performed correction procedures: 1

V:\perlPrograms\ict>
```

Figure 8: Executing *ICT* to compute the corrected mass spectrum values using the Windows command line window interface.

space characters (spaces or tabulators), as it allows to use any number of these characters to split data fields.

6.2 Chemical composition data (input)

A typical input file for the chemical composition of the analyzed molecules is shown below. As mentioned above the input file may have an unlimited number of comment lines. Otherwise the file consists of two lines for each analyzed compound. The compound name (e.g. *Max54S*) is given by the first element of the line. The second element indicates if it declares the chemical composition of the precursor (tag: *Precursor*) or of the fragment (tag: *Product Ion*). The third element identifies the tracer isotope (e.g. ^{13}C) and its maximum number of occurrences (e.g. seven times for the precursor and five times for the fragment). Then follows a list of elements and the maximum number of atoms which are to be considered in the correction procedure.

```
# tag:  precursor/product ion, tracer isotope, other elements
# Precursor:  C29H73N1O10P1Si7
# Product Ion:  C25H55N1O3P1Si4
Max54S:Precursor:  13C7, Si7, O10, N1, C22, H73, P1
Max54S:Product Ion:  13C5, Si4, O3, N1, C20, H55, P1
```

If an element completely disappears during the fragmentation, then the according element still needs to be listed in the input file. For instance, if there are ten atoms of oxygen in the *Precursor* and zero atoms in the *Product Ion*, the input file has to look as follows:

```
# tag:  precursor/product ion, tracer isotope, other elements
# Precursor:  C29H73N1O10P1Si7
# Product Ion:  C25H55N1P1Si4
Max54S:Precursor:  13C7, Si7, O10, N1, C22, H73, P1
Max54S:Product Ion:  13C5, Si4, O0, N1, C20, H55, P1
```

If the molecule is not fragmented, the chemical composition for the *Precursor* and the *Product Ion* must be identical, as shown below:

```
# tag:  precursor/product ion, tracer isotope, other elements
# Precursor:  C29H73N1O10P1Si7
# Product Ion:  C29H73N1O10P1Si7
Max54S:Precursor:  13C7, Si7, O10, N1, C22, H73, P1
Max54S:Product Ion:  13C7, Si7, O10, N1, C22, H73, P1
```

A single chemical composition input file may contain information for more than one compound. In order to provide information for several compounds, use a unique tag for each compound. A simple example for three compounds is shown below:

```
MIT:Precursor:13C6,Si3,C9
MIT:Product Ion:13C6,Si2,C6

Hippo:Precursor:13C6,Si3,C9
Hippo:Product Ion:13C6,Si1,C3

Ben3X:Precursor:13C3,Si2,C6
Ben3X:Product Ion:13C2,Si1,C3
```

6.3 Measured mass spectral data (input)

An example of a valid mass spectrum data set for a non-fragmentation experiment is shown below:

```
Max54S_M0, 40062
Max54S_M1, 27481
Max54S_M2, 19240
Max54S_M3, 8415
Max54S_M4, 3401
Max54S_M5, 1048
Max54S_M6, 292
Max54S_M7, 62
```

Note that for the above data file, a chemical composition file must be used that contains an entry with the tag *Max54S* and for which the chemical composition of the *Precursor* and of the *Product Ion* are identical. A valid chemical composition file for the above example might look as follows:

```
Max54S:Precursor:  13C7, Si7, O10, N1, C22, H73, P1
Max54S:Product Ion:  13C7, Si7, O10, N1, C22, H73, P1
```

If we want to perform a correction of a tandem mass spectrometry experiment, the mass spectrum file must provide the compound tag and the tandem mass isotopomer indices:

```
Botfanin_M0.0, 10103.5
Botfanin_M1.0, 3450.1
Botfanin_M1.1, 3256.2
Botfanin_M2.1, 5012.7
```

```
Botfanin_M2.2, 2580.0
Botfanin_M3.2, 898.4
```

The corresponding chemical composition file for the above measurement data might look as follows:

```
Botfanin:  Precursor:  13C3, Si2, C5
Botfanin:  Product Ion:  13C2, Si2, C5
```

Note that the syntax for specifying the mass spectral data is mandatory. After the compound tag (e.g. *Botfanin*), there must follow an underscore character _ and the mass specification indicated by an *M* and the mass index. The isotopologue mass index is simply given by a number (e.g. *_M3*). In the case of tandem experiments the tandem mass isotopomer indices are be provided by separating the index values by a dot (e.g. *_M2.1*). If one wants to run several correction runs for a single pair of *Precursor* and *Production Ion*, the measurement data can simply by given as comma-separated values (csv):

```
5HAL_M0,88262,105921,80589,80013,77274,58164
5HAL_M1,55618,64152,50914,50434,48973,37365
5HAL_M2,38849,44926,35857,34379,33280,25649
5HAL_M3,13625,15271,12190,12218,11735,8543
5HAL_M4,5547,6593,5178,5146,4817,3944
5HAL_M5,1755,2046,1749,1557,1587,1212
5HAL_M6,522,723,562,613,495,326
```

The above example file will result in six correction runs for the compound *5HAL* in a single execution of *ICT*. Furthermore, it is also possible to provide mass spectrometry data for more than one chemical compound, e.g. for *Hug* and *Fad*:

```
Hug_M0.0,518,1434,1036,1459,1717,1335
Hug_M1.0,115,241,210,255,299,310
Hug_M1.1,20,63,47,70,112,56
Hug_M2.0,38,102,113,95,105,128
Hug_M2.1,15,14,10,3,18,5
Hug_M2.2,15,45,26,36,40,30
Hug_M3.1,3,0,3,13,3,2
Hug_M3.2,0,0,0,0,12,7
Hug_M4.2,0,17,0,0,4,4
Fad_M0.0,4237,4476,3880,3834,3800,3168
Fad_M1.0,684,768,713,607,597,604
```

```
Fad_M1.1,715,941,858,856,829,510
Fad_M2.1,171,192,191,122,129,120
Fad_M2.2,311,279,242,258,287,215
Fad_M3.2,65,57,53,69,33,54
Fad_M3.3,35,88,47,51,55,42
Fad_M4.3,17,6,0,9,0,0
Fad_M4.4,0,0,0,0,0,0
Fad_M5.4,0,0,0,0,0,0
Fad_M5.5,0,0,0,0,0,0
Fad_M6.5,0,0,0,0,0,
```

6.4 Natural abundance data (optional input)

If the natural abundance of isotopes are provided by the user, they need to be given as absolute probabilities. As shown below, all isotope data for an element are described in a single line. Note that the sum of the probabilities must add up to 1.0.

```
C12 C13:  0.989 0.011

H1 H2:  0.9998 0.0002

N14 N15:  0.9964 0.0036

O16 O17 O18:  0.9975 0.0004 0.0021

Si28 Si29 Si30:  0.922 0.047 0.031

S32 S33 S34 S36:  0.9500 0.0075 0.0424 1e-4

P31:  1
```

The natural abundance is provided by the command line parameter `-i` which is an optional parameter. If the natural abundance data are not provided, an internal data set of abundance values is used by *ICT*. This internal data set is stored in the Perl module `Bio::IsotopeCorrection::NaturalIsotopes` which can be found in the file `Bio/IsotopeCorrection/NaturalIsotopes.pm`. If the user does not provide abundance values, the following default values are used by *ICT* [11, 14]:

```
C12 C13:  0.989299998329072 0.0107000016709277

H1 H2:  0.999885003225779 0.000114996774220997

N14 N15:  0.996359998019236 0.00364000198076368

O16 O17 O18:  0.997569999287336 0.000380004339828525 0.00204999637283548

Si28 Si29 Si30:  0.922229995288972 0.0468500031000761 0.0309200016109514

S32 S33 S34 S36:  0.949900003551676 0.00750000197104251 0.0424999985039075 9.99959733738849e-05

P31:  1.0
```

6.5 Corrected mass spectral data (output)

The corrected mass spectrometry data are written by *ICT* to a plain text file. The output of a simple correction for *4STA* might look as follows:

```
4STA_M0.0, 50000.0
4STA_M1.0, 6000.0
4STA_M1.1, 132000.0
4STA_M2.1, 13000.0
4STA_M2.2, 145784.496
4STA_M3.2, 41734.139
4STA_M3.3, 135151.069
4STA_M4.3, 55423.969
4STA_M4.4, 90540.278
4STA_M5.4, 53150.757
4STA_M5.5, 51011.446
4STA_M6.5, 41544.159
4STA_M6.6, 20988.154
4STA_M7.6, 19644.886
4STA_M7.7, 0.0
4STA_M8.7, 11159.1751039811
4STA_M8.8, 0.0
4STA_M9.8, 0.0
```

The corresponding chemical composition file for the above output file might have the following content:

```
4STA:Precursor:  13C9, S1
4STA:Product Ion:  13C8, S1
```

If a batch of correction procedures was performed, the corrected data are stored in form of comma-separated values:

```
X-MTB_M0.0, 355.0, 164.0, 230.0
X-MTB_M1.0, 56.3074946607193, 34.2716313362196, 37.9175317520153
X-MTB_M1.1, 230.410629139991, 177.623501912559, 130.801252682248
X-MTB_M2.0, 25.95363222081, 0.0, 0.0
X-MTB_M2.1, 91.7911134368254, 41.3868730133874, 87.2246115294892
X-MTB_M2.2, 293.232088992987, 98.0190232803821, 154.108282794041
X-MTB_M3.1, 0.0, 2.67637814637201, 0.0
X-MTB_M3.2, 67.0906056733473, 127.401733768079, 55.5837397052624
X-MTB_M3.3, 179.732214999231, 111.684286224647, 0.0
X-MTB_M4.2, 0.0, 6.97739206538934, 16.3268595168432
X-MTB_M4.3, 81.2516661232786, 6.79589570875958, 49.3873579717811
```

```
X-MTB_M4.4, 58.598619874574, 22.2423014900055, 68.0694739902165
X-MTB_M5.3, 0.0, 13.2036039240464, 0.0
X-MTB_M5.4, 0.0, 8.96563573108918, 0.0
X-MTB_M6.4, 8.71639255207118, 11.7945102171697, 0.0
```

6.6 Tracer purity (input)

ICT can compute the influence of the (im)purity of the tracer material on the corrected mass spectral data. If the tracer purity is not considered, it is assumed that 100% of the labeled atoms are of the desired isotope type, e.g. ^{13}C. However, if the tracer purity should be considered, the command line option `-p` needs to be used. A file name has to be provided with this option. The file needs to contain the purity values of the tracer element. A typical purity file for the tracer isotope ^{13}C might look as follows:

```
C12 C13:  0.1 0.9
```

The above file means that 10% of the tracer atoms are not ^{13}C isotopes, but ^{12}C isotopes. The effect of this deviation on the mass spectra is considered by *ICT*. If a tracer element with more than two isotopes is used, e.g. silicon, the purity file might look as shown below:

```
Si28 Si29 S30:  0.05 0.15 0.8
```

Note that the sum of purity values must add up to 1.0.

6.7 Further information

Check out the directory `examples/data_dir/` for further file format examples.

7 Technical details

7.1 Mean enrichment

The mean enrichment is computed if a mass spectrometry experiment without fragmentation is corrected. The mean enrichment is the average number of labeled atoms per molecule. For instance, if we obtained the following corrected spectrometry data:
0: (M=0): 0.296957592
1: (M=1): 0.221858505
2: (M=2): 0.18140636
3: (M=3): 0.161405358

4: (M=4): 0.138372185

Then, the enrichment would be given by the equation
$(1 * 0.221858505 + 2 * 0.18140636 + 3 * 0.161405358 + 4 * 0.138372185)/4 = 0.405594$

To our knowledge the calculation of the mean enrichment for tandem mass isotopomers has no practical relevance and is not implemented in the current version of *ICT*. Consequently, the enrichment is only computed and printed for corrections of mass spectrometry data without fragmentation.

7.2 Natural abundance at unlabeled positions

Most frequently glucose, labeled with ^{13}C on different positions, is applied for metabolic flux analysis experiments. In Figure 9 the structural formula of D-glucose is depicted.

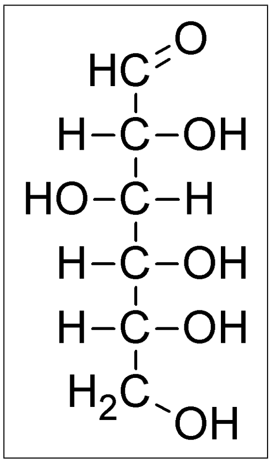


Figure 9: Chemical structure of glucose.

Fully labeled glucose, so that every carbon atom in the molecule is replaced by the heavy stable isotope ^{13}C (as depicted in figure 10a), is often employed in context of metabolic flux experiments. As in general the purity of this uniformly labeled ^{13}C glucose may not achieve 100%, molecules with a lower labeling degree, such as depicted in figure 10b, will occur. However, these unlabeled positions may be still be replaced by a ^{13}C because of the natural abundance of carbon (see figure 10c). Obviously, this natural abundance of the tracer element has an effect on the measured mass spectrometric data. This effect can be corrected by *ICT*. The probability that an unlabeled C-atom represents a heavy isotope is determined by the natural isotope abundance. In order to activate the correction of the natural abundance of the tracer molecule, the command line parameter `-n` needs to be used when ICT is started.

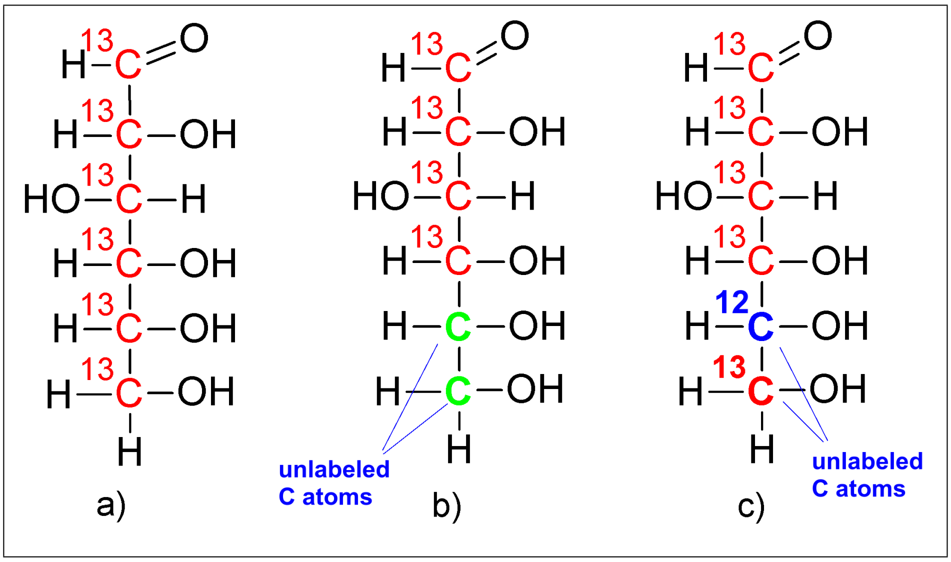


Figure 10: *a)* structural formula of uniformly labeled ^{13}C glucose (U^{13}C-Glc), *b)* impurity of U^{13}C-Glucose, where only four out of six carbon atoms were labeled with ^{13}C *c)* impurity of U^{13}C-glucose, where one of the unlabeled carbons of b) is now present as heavy isotope due to the natural isotope abundance of carbon.

7.3 Tracer (im)purity

Due to the aforementioned stable isotope tracer impurity, some metabolites of interest will carry a ^{12}C-atom instead of a ^{13}C-atom on certain positions. This means that labeling patterns might be somehow 'misrepresentative', as shown in Figure 11 for citrate as affected metabolite. The effect of this impurity can be corrected via *ICT*.

By default, *ICT* assumes that the tracer shows a purity of 100%, so that the respective tracer isotopes are present at all intentionally labeled positions. If the tracer impurity should be considered by *ICT*, the command line parameter -p needs to be used. ICT just requires the information of the tracer's purity for each isotope, e.g. if 90% of all labeled carbon atoms are ^{13}C isotopes and 10% are ^{12}C isotopes, the purity file (see chapter 6.6) must contain the following line:

```
C12 C13:  0.1 0.9
```

7.4 Possible labeling states

Figure 12 depicts all possible labeling states of valine and the respective isotopologues and tandem mass isotopomer fractions obtained in a ^{13}C tracer

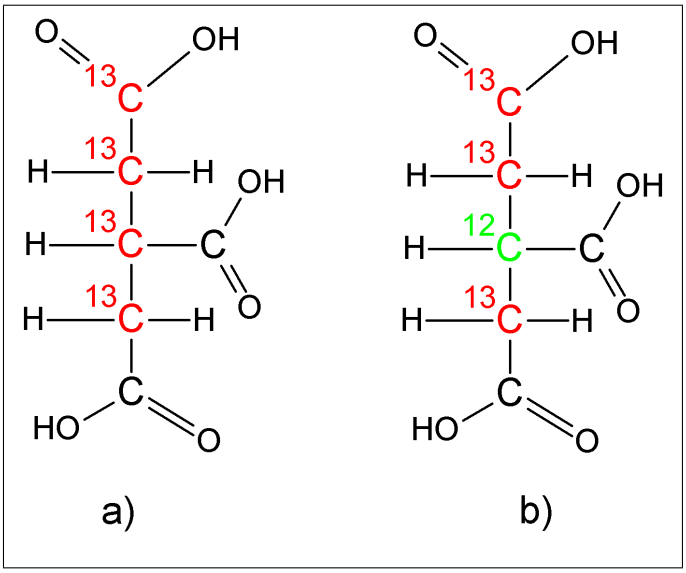


Figure 11: *a)* Citrate labeled on position C1-4 due to a stable isotope tracer experiment, *b)* citrate labeling state of a), though with a ^{12}C on position C3 due to tracer impurity.

experiment. In our approach, a GC-QTOFMS instrument employing chemical ionization as ionization method is used. Applying this soft ionization technique, the protonated molecules represent the base peak in the mass spectrum (Mairinger T et al., manuscript in preparation). Isotopologue selective fragmentation via collision induced dissociation yields tandem mass isotopomers delivering additional information on the position of the ^{13}C label. In order to determine the sum formula of the isotopologues as well as the structure and the exact mass of the fragment ion, dedicated software was utilized for *in silico* fragmentation (MassFrontier v7.0 (Thermo Fisher Scientific, Waltham,MA)).

Using a high resolution mass spectrometer as mass analyzer of the second MS stage (after fragmentation) is needed to confirm the sum formula of the fragments. A further advantage of using a QTOF or Orbitrap instrument is that all potentially interesting fragments and their respective labeling states are recorded. In comparison to a MS/MS (triple quad or tandem MS) this will decrease the number of transitions within one chromatographic peak and, therefore, the dwell time/acquisition time for each transition is increased, which is beneficial for both, the number of data points per chromatographic peak and the sensitivity. Additional advantages of high resolution mass spectrometry is the separation of interferences on tandem mass isotopomers as well as the possibility of retrospective data processing. Note that *ICT* can correct high resolution MS experiments, such as QTOF or Orbitrap, and low resolution mass spectrometry data obtained by single

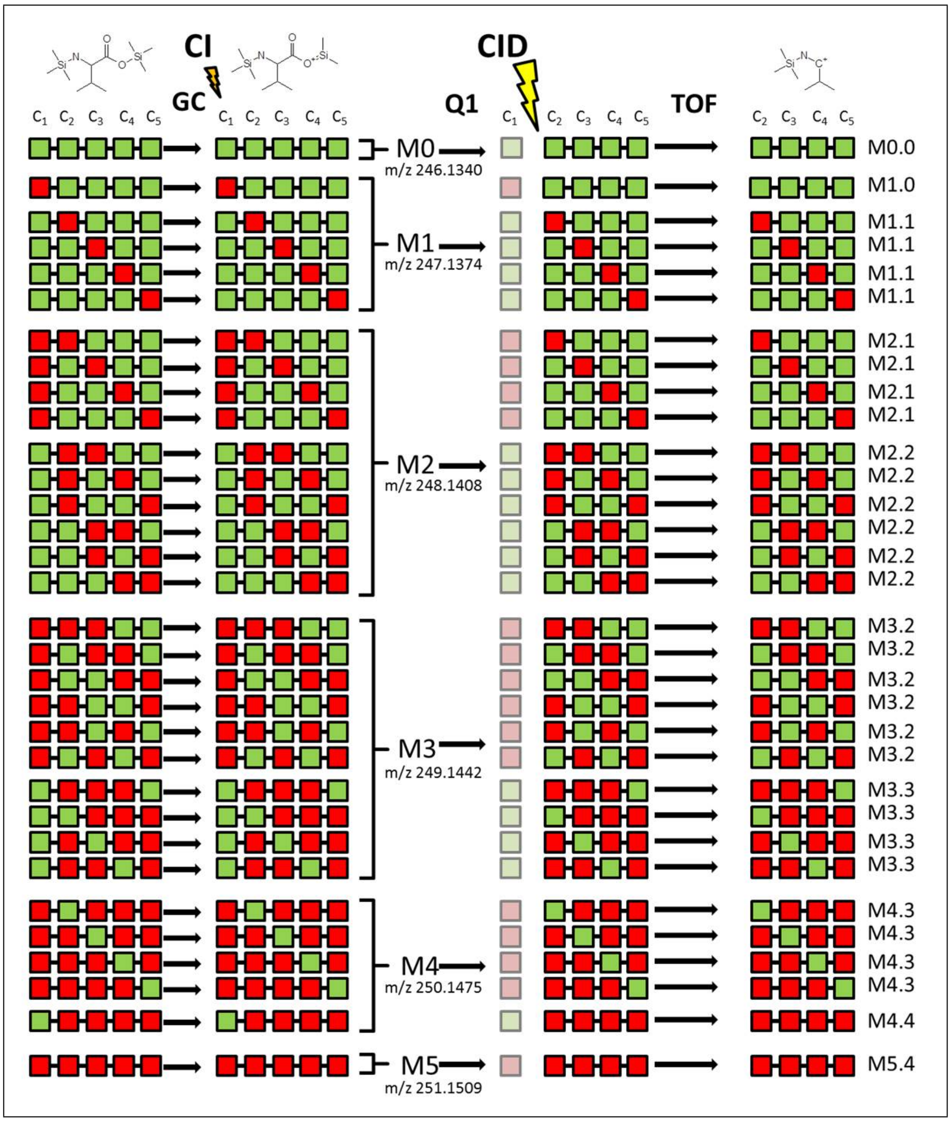


Figure 12: Analytical strategy for GC-CI-QTOFMS based measurement of all possible labeling states of valine and the respective isotopologues and tandem mass isotopomer fractions.

or triple quadrupole instruments.

An example of a correction procedure for valine, on the basis of flux experiment data, is given below. The chemical composition file for valine has the following content:

```
Val:Precursor:13C5,C5,H24,N1,O2,Si2
Val:Product Ion:13C4,C3,H18,N1,O0,Si1
```

The obtained peak areas for each possible tandem mass isotopomer fraction of this flux experiment are as follows:

```
Val_M0.0, 33.79, 34.28, 32.98,
Val_M1.0, 4.79, 4.72, 5.10,
Val_M1.1, 22.73, 22.13, 20.63,
Val_M2.1, 3.91, 3.98, 4.48,
Val_M2.2, 13.35, 13.17, 14.77,
Val_M3.2, 8.63, 9.30, 8.68,
Val_M3.3, 5.13, 4.83, 5.20,
Val_M4.3, 3.69, 3.66, 4.47,
Val_M4.4, 1.40, 1.22, 1.42,
Val_M5.4, 2.58, 2.72, 2.26,
```

which results in the following output file containing the corrected mass spectra data for valine:

```
Val_M0.0, 40.09, 40.65, 39.09
Val_M1.0, 2.73, 2.60, 3.16
Val_M1.1, 23.40, 22.62, 20.97
Val_M2.1, 2.41, 2.54, 3.23
Val_M2.2, 12.31, 12.14, 14.23
Val_M3.2, 8.75, 9.56, 8.60
Val_M3.3, 4.09, 3.77, 4.09
Val_M4.3, 3.05, 2.98, 3.95
Val_M4.4, 0.82, 0.64, 0.78
Val_M5.4, 2.35, 2.50, 1.89
```

The effect of the correction procedure (averaged over all three data sets) is illustrated in figure 13 (note that the uncertainty of the measured areas is in most cases significantly lower than the effect of the correction algorithm).

7.5 Verification of implementation

In order to verify the results of *ICT*, we performed numerous correction runs of non-fragmentation mass spectral data and compared them with results

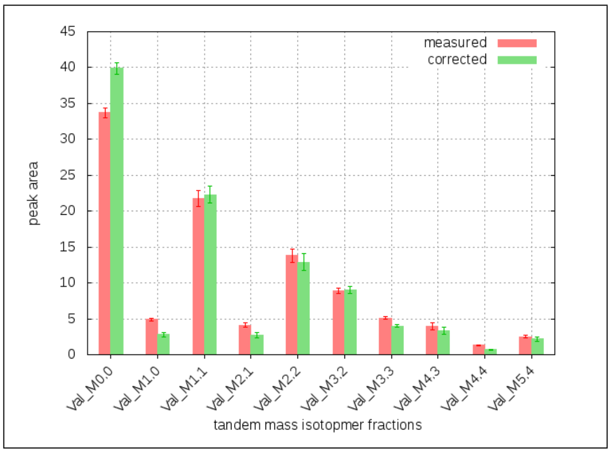


Figure 13: Measured and corrected values (average over three data sets) for a tandem mass spectrometry experiment using valine.

which we obtained by using *IsoCor*. We found an excellent agreement between the corrected data computed by *ICT* and *IsoCor* (see chapter 7.6.3 for details). We could not use *IsoCor* to verify the correctness of *ICT* for tandem mass spectrometry experiments, as *IsoCor* is not able to deal with the fragmentation. Hence, a verification of the correction procedure for tandem mass spectrometry data was performed by comparing the results of *ICT* with a large set of correction data which we obtained by a (slow and inefficient) Microsoft Excel based method.

7.6 Performance

Typically, the correction process for a smaller problem takes a fraction of a second. For instance, running a batch of 585 correction procedures taking into account the natural abundance of carbon and silicon for a variety of compounds, such as amino acids and low molecular weight organic acids, took approximately 20 seconds. Larger problems, such as the correction of the interferences caused by the natural abundance of carbon, hydrogen, nitrogen, oxygen, and silicon for ethoximated/trimethylsilylated sedoheptulose-7-phosphate (C29H73N1O10P1Si7), can usually be done within less than one second. See the following subsections for a detailed performance analysis and a comparison with *IsoCor*. The presented runs were performed on a Linux Ubuntu 14.04 system which was equipped with one Intel i5-2500S CPU (2.70GHz, four cores).

7.6.1 Runtime as a function of number of tracer atoms

In order to analyze the performance of *ICT* as a function of the number of tracer atoms, the following compound was analyzed:

```
Val:Precursor:13Cx,C5,H24,N1,O2,Si2
Val:Product Ion:13Cy,C3,H18,N1,O0,Si1
```

where x is the number of tracer atoms in the precursor and y is the number of tracer atoms in the product ion. x was varied from 1 to 15 and y was varied from 0 to x. Figure 14 depicts the runtime as a function of the number of tracer atoms in the precursor (x) and the product ion (y). Figure 14 clearly shows that the runtime increases with an increasing number of tracer atoms in the precursor. Note that the maximum number of tracer atoms in the product ion is limited by the number of tracer atoms in the precursor. Figure 14 shows that for a given x the runtime is low for small y values, then increases with increasing y, and finally decreases again for larger values of y. This effect is caused by the number of N/n-pairs which is highest for a given x for mid-sized y values, as a large number of N/n-pairs results in high computational costs and, hence, a slow program execution. Note that for the runs presented in this subsection (real-world) data acquired for a ^{13}C metabolic flux experiment were used as input for the correction procedures.

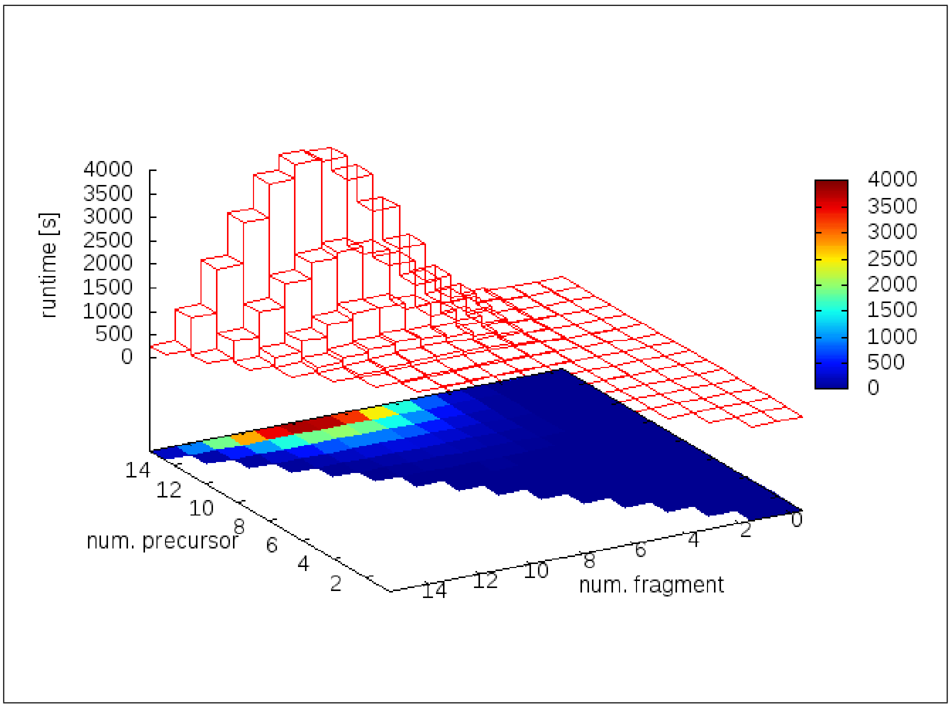


Figure 14: Runtime [s] as a function of the number of tracer atoms in the precursor and the product ion.

7.6.2 Runtime as a function of considered elements

In order to analyze the performance of *ICT* as a function of elements that are considered during the correction procedure, the following base compound was used:

```
Val:Precursor:13C5,C5,H24,N1,O2,Si2
Val:Product Ion:13C4,C3,H18,N1,O0,Si1
```

The contribution of the elements (C,H,N,O,Si) on the runtime was investigated by running simulations with all possible combinations of activated/deactivated elements. Figure 15 depicts the runtime as a function of considered elements. The runtime analysis was performed for four different cases: i) ordinary correction procedure, ii) correction including natural abundance of the tracer element, iii) correction including impurity of the tracer element (C12: 0.1, C13: 0.9), and iv) correction including natural abundance and impurity of the tracer element. Figure 15 clearly shows that the runtime of *ICT* increases with an increasing number of elements taken into account during the correction procedure. Note that for the runs presented in this subsection measured (real-world) intensities were used as input for the correction procedures.

7.6.3 Comparison with *IsoCor*

In this subsection the performance and the accuracy of *ICT* and *IsoCor* is compared. As *IsoCor* cannot correct tandem mass spectrometry experiments, the comparison was limited to cases without fragmentation. The comparison was done by using the following base system:

```
Val:Precursor:13Cx,C5,H24,N1,O2,Si2
Val:Product Ion:13Cx,C5,H24,N1,O2,Si2
```

where x was varied from 1 to 15. Note that the above system represents an experiments without fragmentation, as the chemical composition of the precursor and the product ion are identical. Again, the performance study was done for four different cases: i) ordinary correction procedure, ii) correction including natural abundance of the tracer element, iii) correction including impurity of the tracer element (C12: 0.1, C13: 0.9), and iv) correction including natural abundance and impurity of the tracer element. For the correction runs presented in this subsection manually created intensities were used.

Runtime

Figure 16 shows the runtime as a function of the number of tracer atoms.

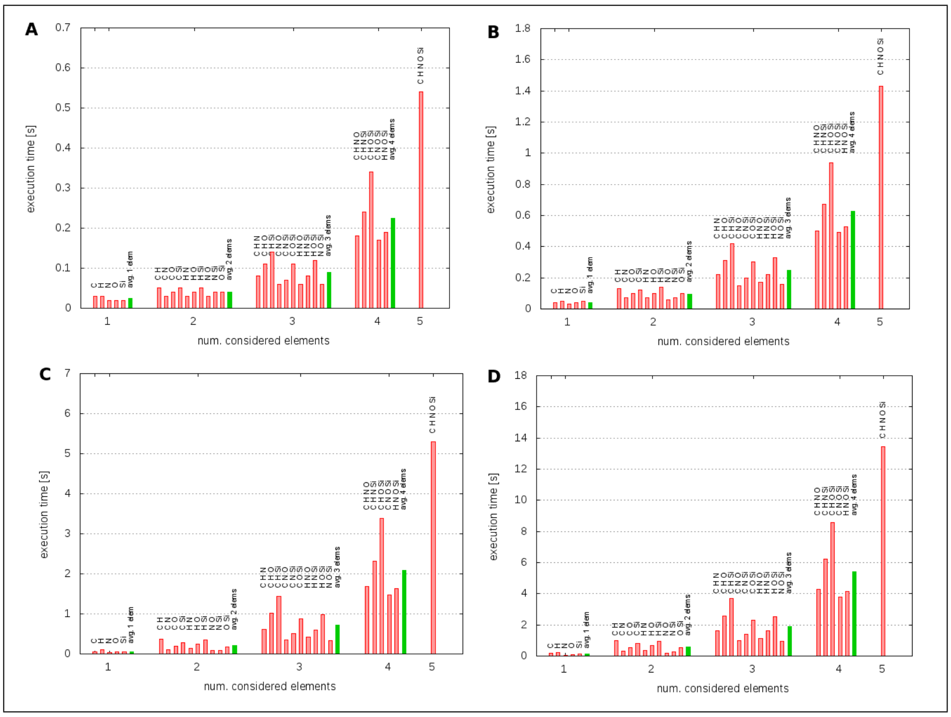


Figure 15: Runtime [s] as a function of elements considered during the correction procedure. A: ordinary correction procedure, B: correction including natural abundance of the tracer element, C: correction including impurity of the tracer element (C12: 0.1, C13: 0.9), and D: correction including natural abundance and impurity of the tracer element.

The number of tracer atoms in the precursor and in the product ion was equal. Figure 16 shows that the runtime increases with an increasing number of tracer atoms. It can be seen that for smaller systems *ICT* and *IsoCor* are fast and show approximately the same performance. However, for larger systems the execution time of *ICT* is significantly higher than the execution time of *IsoCor*. This is mainly caused by the fact that *ICT* also uses the general algorithm (which is able to handle fragmentation) for this special, degenerated case of non-fragmented compounds.

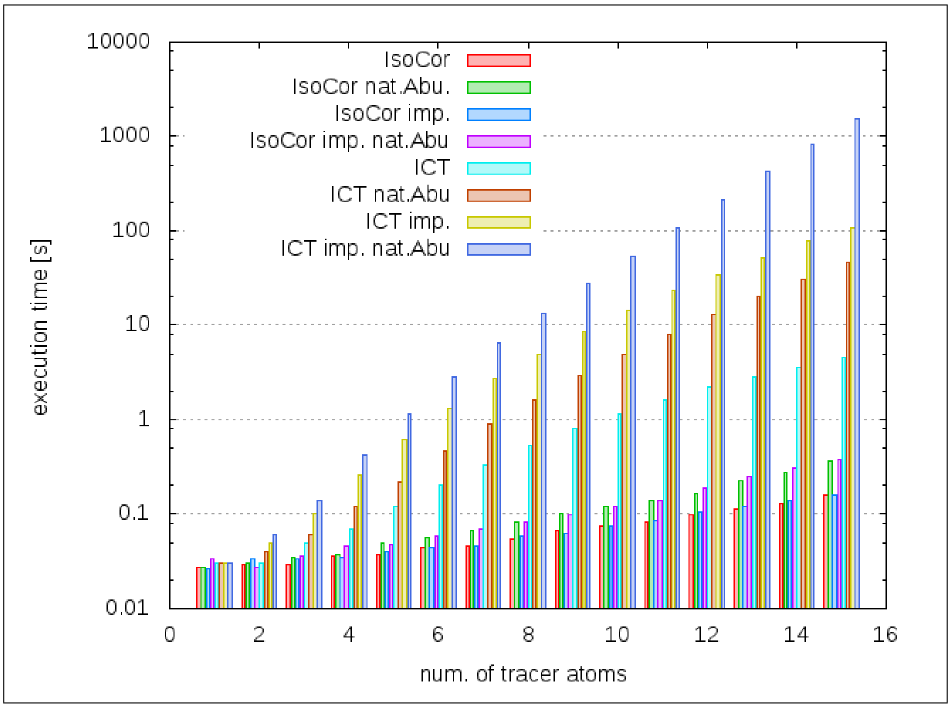


Figure 16: Execution times of *ICT* and *IsoCor* as a function of the number of tracer atoms. The labels 'nat.Abu' and 'imp.' stand for 'natural abundance of the tracer' and 'impurity of the tracer', respectively.

Accuracy

Figure 17 compares the corrected intensity values obtained by *ICT* and *IsoCor* for two situations: i) for eight tracer atoms ($x = 8$) and ii) for 15 tracer atoms ($x = 15$). Both cases considered the natural abundance and an impurity (C12: 0.1, C13: 0.9) of the tracer. Figure 17 clearly shows an excellent agreement of the corrected intensity values. Systems which do not lead to a corrected intensity of zero (e.g. figure 17A with $x = 8$) only differ in the range of the numerical accuracy of the computer system (see Table 1). Table 1 lists the average absolute error E over all corrected intensities for a varying number of tracer atoms, where E is given by:

$$E = \frac{\frac{\sum_{i=0}^{x} |I^i_{IsoCor} - I^i_{ICT}|}{I^0_{measured}}}{x+1}. \tag{1}$$

In the above equation x is again the number of tracer atoms.

Cases where one or more intensity values are corrected to zero show minor differences (e.g. see the intensity for a isotopologue mass index of 14 in figure 17B for the case $x = 15$). In Table 1 this case is depicted in the last row of the column 'avg. abs. err. imp. nat.Abu' which shows the average absolute error between *ICT* and *IsoCor* for a correction with 15 carbon atoms in the backbone that also takes into account an impurity and the natural abundance of the tracer. These deviations are caused by the way how the linear system of equations is solved. *IsoCor* uses an optimization method that tries to find a minimum of the residual intensities, whereas *ICT* uses a method based on the Gauss elimination algorithm.

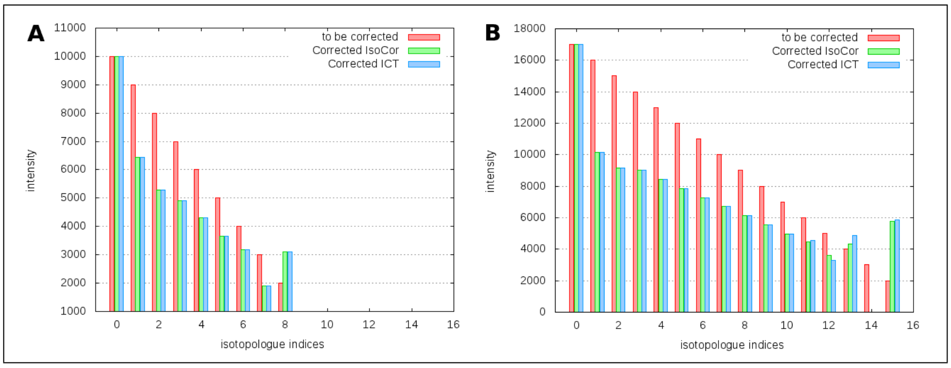


Figure 17: Comparison of the accuracy of the computed corrections. A: the number of carbon atoms in the backbone is eight ($x = 8$), B: the number of carbon atoms in the backbone is fifteen ($x = 15$).

7.7 Temporary files

During the execution of *ICT* temporary text files are created, for instance `tmp_chem_5HAL_9.txt` and `tmp_tof_5HAL_9.txt`, which are used by the Perl module `Bio::IsotopeCorrection::Chemicals` to perform the actual correction. After finishing the correction procedure these temporary files are automatically deleted by *ICT*. If *ICT* is pre-maturely terminated for some reason, these files cannot be deleted by *ICT* and, hence, might stay on your hard disk. Hence, if *ICT* is not running, these temporary files can safely be deleted.

num. tracer atoms in precursor	num. tracer atoms in product ion	avg. abs. error E ordinary	avg. abs. error E imp. & nat.Abu.
1	1	2.830507e-10	2.409597e-10
2	2	4.396483e-10	3.201829e-10
3	3	5.628536e-10	3.453293e-10
4	4	6.241349e-10	5.598037e-10
5	5	7.203269e-10	7.366913e-10
6	6	5.674503e-10	7.020442e-10
7	7	9.204511e-10	8.993089e-10
8	8	4.181206e-10	7.463616e-10
9	9	1.565045e-09	1.103979e-09
10	10	9.950871e-10	8.991184e-10
11	11	1.348375e-09	1.488016e-09
12	12	1.459897e-09	1.336208e-09
13	13	1.389632e-09	1.171479e-09
14	14	1.383621e-09	1.337102e-09
15	15	1.977444e-09	0.0039279286

Table 1: Average absolute error E for corrected intensities as a function of the number of tracer atoms. Two cases are listed: i) ordinary (impurity and natural abundance of the tracer are not considered), ii) including impurity and natural abundance of the tracer.

7.8 Perl modules used by *ICT*

The heart of *ICT* is built by the Perl modules `Bio::IsotopeCorrection::Chemicals` and `Bio::IsotopeCorrection::NaturalIsotope` which are used by the Perl program `ict.pl`. Both modules implement object oriented Perl classes which are instantiated by `ict.pl`. Furthermore, `ict.pl` uses the Perl modules `Getopt::Std` and `File::Basename` which are part of Perl's core. Additionally, only the standard Perl module `Carp` is used by *ICT*. As all required modules are either part of *ICT* or part of Perl's core, it is not required to install any extra Perl module to run *ICT*.

7.9 Computation of isotope combinations and probabilities

The principal workflow of *ICT* is illustrated in figure 18. Let's assume we are interested in the correction of a compound that has potentially five ^{13}C tracer isotopes in the precursor ($N = 5$) and after the fragmentation there are potentially three ^{13}C tracers isotopes left in the product ion ($n = 3$). This results in a limited number of mass index pairs we are interested in and which we want to correct (see table 2). Table 2 shows the number of unlabeled (^{12}C) and labeled (^{13}C) atoms before and after fragmentation. The columns M^{tracer} and m^{tracer} of table 2 denote the increase of the mass of the precursor and the product ion, respectively. For instance, in line 4 of table 2 we consider three ^{12}C and two ^{13}C atoms before the fragmentation. These two ^{13}C atoms increase the mass index of the precursor by two, hence, $M^{tracer} = 2$. Therefore, we read the measured mass spectrum values for these pairs form the input file and correct them according to the provided information about the chemical composition of the precursor and the product ion.

In order to perform the correction, *ICT* computes the correction matrix CM, [11] which relates the measured with the corrected spectral data:

$$CM \times I_{corr} = I_{meas}. \tag{2}$$

The correction matrix CM is generated by computing the probabilities of all relevant combinations of isotope sets and summing up the effect of these probabilities for each required mass pair shown in table 2. These combinations of isotope sets are computed by *ICT* by iterating through all considered elements. For each element the information about all isotopes is either read from the provided file or retrieved from *ICT* 's internal database. First, the N/n-pairs of each involved isotope are computed, where N and n indicate the number of occurrences of the isotope in the precursor and the product ion, respectively. If for instance silicon and nitrogen are considered with (i) potentially three and two silicon atoms in the precursor and the product ion, respectively and (ii) potentially two nitrogen atoms in the precursor and one nitrogen atom in the product ion, we get

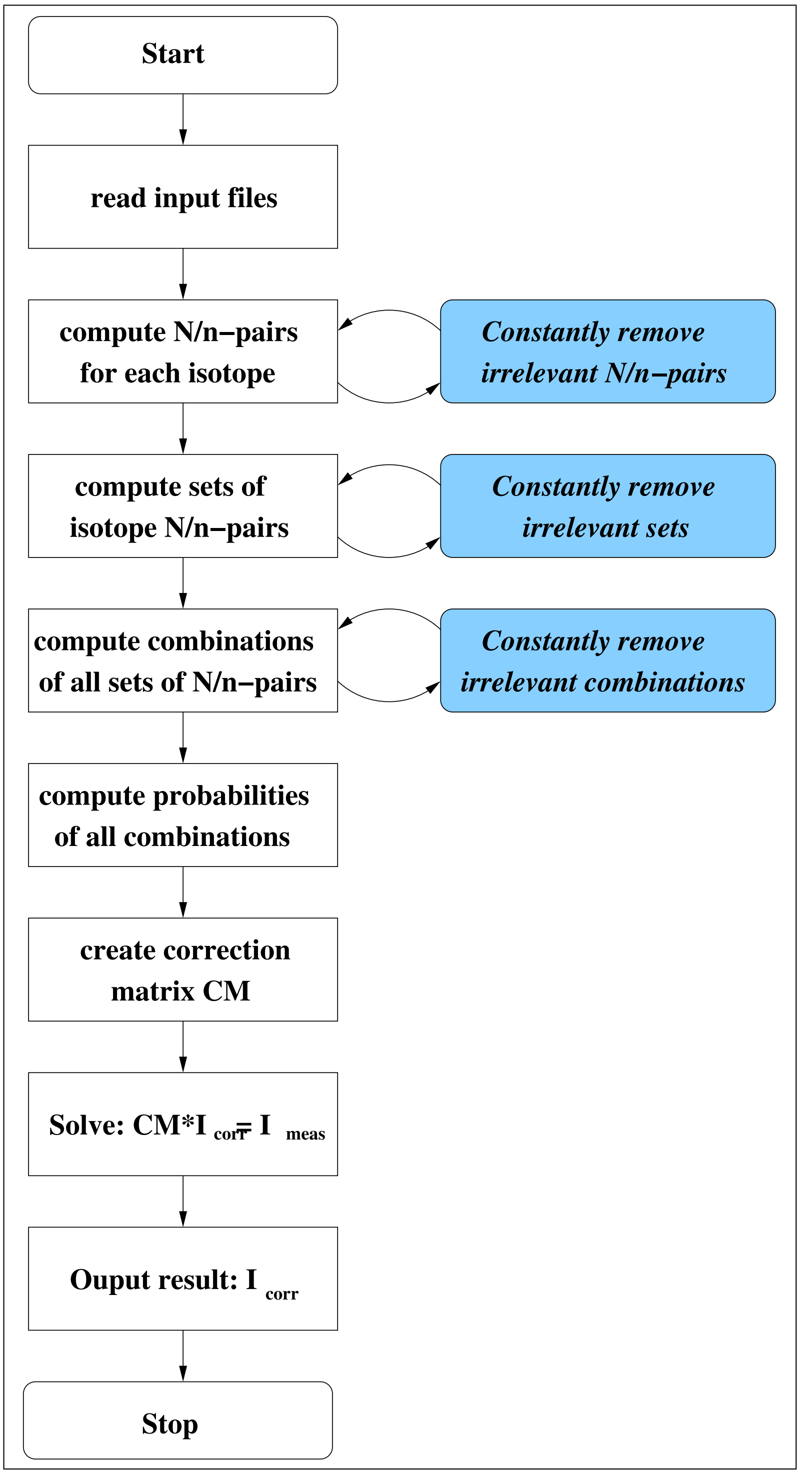

Start
read input files
compute N/n−pairs for each isotope
Constantly remove irrelevant N/n−pairs
compute sets of isotope N/n−pairs
Constantly remove irrelevant sets
compute combinations of all sets of N/n−pairs
Constantly remove irrelevant combinations
compute probabilities of all combinations
create correction matrix CM
Solve: CM*I corr = I meas
Ouput result: I corr
Stop

Figure 18: Illustration of the principal workflow of *ICT*.

Precursor		Product Ion		Precursor mass index	Product Ion mass index
^{12}C	^{13}C	^{12}C	^{13}C	M^{tracer}	m^{tracer}
5	0	3	0	0	0
4	1	3	0	1	0
4	1	3	1	1	1
3	2	3	0	2	0
3	2	2	1	2	1
3	2	1	2	2	2
2	3	2	1	3	1
2	3	1	2	3	2
2	3	0	3	3	3
1	4	1	2	4	2
1	4	0	3	4	3
0	5	0	3	5	3

Table 2: Relevant pairs of mass for a tandem mass spectrometry experiment with potentially five ^{13}C tracer isotopes in the precursor and potentially three ^{13}C tracer isotopes in the product ion. The columns M^{tracer} and m^{tracer} denote the increase of the mass before and after fragmentation caused by ^{13}C isotopes, respectively.

a list of N/n-pairs for the isotope ^{29}Si and the isotope ^{30}Si as shown in table 3 and table 4. In table 3 and table 4 the columns indicated with N and n denote the number of isotopes in the molecule before and after fragmentation, whereas the columns indicated with M and m indicate the increase of the mass index. For instance, the atomic mass of ^{30}Si is two larger than of ^{28}Si, hence, an N^{30Si} of three will result in an increase of the mass index, M^{30Si}, by six. Any N/n-pair that is not relevant for the subsequent computation is immediately removed. An N/n-pair is removed if the resulting tandem mass isotopomer index (M.m) is higher than of the fully labeled tandem mass isotopomer (every tracer atom is labeled), e.g. if either $M^{30Si} > M^{tracer}$ or $m^{30Si} > m^{tracer}$. This removal process is important in order to obtain fast execution times. In particular, if the compound contains elements with many atoms, e.g. hydrogen, carbon, and oxygen in ethoximated/trimethylsilylated sedoheptulose-7-phosphate which has the following chemical structure: C29H73N1O10P1Si7.

Next, all isotope N/n-pairs of a chemical element are combined by *ICT* to sets of N/n-pairs. This list of sets is computed by combining the N/n-pairs of all isotopes of a chemical element. Throughout the following paragraphs we will use an example with two elements: silicon (Si) and nitrogen (N). Table 5 and table 6 list these sets of pairs for silicon and for nitrogen, respectively. Again, any irrelevant set is immediately removed during the

Precursor N^{29Si}	Product Ion n^{29Si}	Precursor M^{29Si}	Product Ion m^{29Si}
0	0	0	0
1	0	1	0
1	1	1	1
2	0	2	0
2	1	2	1
2	2	2	2
3	2	3	2

Table 3: N/n pairs for the isotope ^{29}Si if the precursor and the product ion may contain three and two Si atoms, respectively. The columns N^{29Si} and n^{29Si} show the number of ^{29}Si isotopes in the molecule before and after fragmentation. Whereas m^{29Si} and m^{29Si} denote the increase of the mass caused by ^{29}Si isotopes.

Precursor N^{30Si}	Product Ion n^{30Si}	Precursor M^{30Si}	Product Ion m^{30Si}	
0	0	0	0	
1	0	2	0	
1	1	2	2	
2	0	4	0	
2	1	4	2	
2	2	4	4	(removed)
3	2	6	4	(removed)

Table 4: N/n pairs for the isotope ^{30}Si if the precursor and the product ion contain three and two Si atoms, respectively. The columns N^{30Si} and n^{30Si} show the number of ^{30}Si isotopes in the molecule before and after fragmentation. Whereas m^{30Si} and m^{30Si} denote the increase of the mass caused by ^{30}Si isotopes.

Precursor			Product Ion			Precursor	Product Ion
^{28}Si	^{29}Si	^{30}Si	^{28}Si	^{29}Si	^{30}Si	mass index M^{Si}	mass index m^{Si}
3	0	0	2	0	0	0	0
2	1	0	2	0	0	1	0
2	1	0	1	1	0	1	1
1	2	0	1	1	0	2	1
1	2	0	0	2	0	2	2
2	0	1	2	0	0	2	0
2	0	1	1	0	1	2	2
1	1	1	1	1	0	3	1
1	1	1	1	0	1	3	2
1	1	1	0	1	1	3	3
0	3	0	0	2	0	3	2
0	2	1	0	2	0	4	2
0	2	1	0	1	1	4	3

Table 5: Set of N/n-pairs for silicon if the precursor contains three Si atoms the product ion contains two Si atoms.

Precursor		Product Ion		Precursor	Product Ion
^{14}N	^{15}N	^{14}N	^{15}N	mass index $M^{Nitrogen}$	mass index $m^{Nitrogen}$
2	0	1	0	0	0
1	1	1	0	1	0
1	1	0	1	1	1
0	2	0	1	2	1

Table 6: Set of N/n-pairs for nitrogen if the precursor contains two N atoms and the product ion contains one N atom.

computation process. Such a removal process can be caused by a mass that is too high to be relevant for the correction process, e.g. $M^{Si} > M^{tracer}$ or $m^{Si} > m^{tracer}$.

After all relevant sets of N/n-pairs of all participating elements were calculated, each set is combined with all other sets in order to obtain all possible combinations that effect the measured mass spectrum. See table 7 for an example. For each combination the sum of the mass of the precursor and of the product ion is computed. Again, any combination that is irrelevant for the correction process is immediately removed in order to reduce the number of combinations and, thereby, reducing the amount of consumed memory (RAM) and speeding up the execution of *ICT*.

If all required set combinations and their masses are computed, the probability of each combination is calculated. In order to determine the

Precursor		Product Ion		Precursor		Product Ion		Precursor			Product Ion			Precursor	Product Ion	
Tracer		Tracer												total	total	
^{12}C	^{13}C	^{12}C	^{13}C	^{14}N	^{15}N	^{14}N	^{15}N	^{28}Si	^{29}Si	^{30}Si	^{28}Si	^{29}Si	^{30}Si	mass	mass	
5	0	3	0	2	0	1	0	3	0	0	2	0	0	0	0	
5	0	3	0	2	0	1	0	2	1	0	2	0	0	1	0	
5	0	3	0	2	0	1	0	2	1	0	1	1	0	1	1	
5	0	3	0	2	0	1	0	1	2	0	1	1	0	2	1	
5	0	3	0	2	0	1	0	1	2	0	0	2	0	2	2	
5	0	3	0	2	0	1	0	2	0	1	2	0	0	2	0	
5	0	3	0	2	0	1	0	2	0	1	1	0	1	2	2	
5	0	3	0	2	0	1	0	1	1	1	1	1	0	3	1	
5	0	3	0	2	0	1	0	1	1	1	1	0	1	3	2	
5	0	3	0	2	0	1	0	1	1	1	0	1	1	3	3	
5	0	3	0	2	0	1	0	0	3	0	0	2	0	3	2	
5	0	3	0	2	0	1	0	0	2	1	0	2	0	4	2	
5	0	3	0	2	0	1	0	0	2	1	0	1	1	4	3	
5	0	3	0	1	1	1	0	3	0	0	2	0	0	1	0	
5	0	3	0	1	1	1	0	2	1	0	2	0	0	2	0	
5	0	3	0	1	1	1	0	2	1	0	1	1	0	2	1	
5	0	3	0	1	1	1	0	1	2	0	1	1	0	3	1	
5	0	3	0	1	1	1	0	1	2	0	0	2	0	3	2	
5	0	3	0	1	1	1	0	2	0	1	2	0	0	3	0	
5	0	3	0	1	1	1	0	2	0	1	1	0	1	3	2	
5	0	3	0	1	1	1	0	1	1	1	1	1	0	4	1	
5	0	3	0	1	1	1	0	1	1	1	1	0	1	4	2	
5	0	3	0	1	1	1	0	1	1	1	0	1	1	4	3	
5	0	3	0	1	1	1	0	0	3	0	0	2	0	4	2	
5	0	3	0	1	1	1	0	0	2	1	0	2	0	5	2	
5	0	3	0	1	1	1	0	0	2	1	0	1	1	5	3	
5	0	3	0	1	1	0	1	3	0	0	2	0	0	1	1	
5	0	3	0	1	1	0	1	2	1	0	2	0	0	2	1	
5	0	3	0	1	1	0	1	2	1	0	1	1	0	2	2	
5	0	3	0	1	1	0	1	1	2	0	1	1	0	3	2	
5	0	3	0	1	1	0	1	1	2	0	0	2	0	3	3	
5	0	3	0	1	1	0	1	2	0	1	2	0	0	3	1	
5	0	3	0	1	1	0	1	2	0	1	1	0	1	3	3	
5	0	3	0	1	1	0	1	1	1	1	1	1	0	4	2	
5	0	3	0	1	1	0	1	1	1	1	1	0	1	4	3	
5	0	3	0	1	1	0	1	1	1	1	0	1	1	4	4	
5	0	3	0	1	1	0	1	0	3	0	0	2	0	4	3	
5	0	3	0	1	1	0	1	0	2	1	0	2	0	5	3	
5	0	3	0	1	1	0	1	0	2	1	0	1	1	5	4	(removed)
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	
0	5	0	3	0	2	0	1	3	0	0	2	0	0	7	4	(removed)
0	5	0	3	0	2	0	1	2	1	0	2	0	0	8	4	(removed)
0	5	0	3	0	2	0	1	2	1	0	1	1	0	8	5	(removed)
0	5	0	3	0	2	0	1	1	2	0	1	1	0	9	5	(removed)
0	5	0	3	0	2	0	1	1	2	0	0	2	0	9	6	(removed)
0	5	0	3	0	2	0	1	2	0	1	2	0	0	9	4	(removed)
0	5	0	3	0	2	0	1	2	0	1	1	0	1	9	6	(removed)
0	5	0	3	0	2	0	1	1	1	1	1	1	0	10	5	(removed)
0	5	0	3	0	2	0	1	1	1	1	1	0	1	10	6	(removed)
0	5	0	3	0	2	0	1	1	1	1	0	1	1	10	7	(removed)
0	5	0	3	0	2	0	1	0	3	0	0	2	0	10	6	(removed)
0	5	0	3	0	2	0	1	0	2	1	0	2	0	11	6	(removed)
0	5	0	3	0	2	0	1	0	2	1	0	1	1	11	7	(removed)

Table 7: A subset of all relevant isotope combinations.

probability, the natural abundance of each isotope is required. The natural abundance values are taken either from the file provided by the user (command line parameter `-i`) or from the internal dataset that is stored in the Perl module `Bio::IsotopeCorrection::NaturalIsotopes` [11, 14]. The probability value of a set combination is comprised of two parts. The first part, p_{pre}, is given by the contribution of the precursor and the second part, p_{frag}, is a result of the fragmentation. Hence, the total probability of a combination is given by:

$$p_{tot} = p_{pre} \cdot p_{frag}. \tag{3}$$

The precursor probability is the product of the element probabilities, p_{elem}, of all participating elements $elem$:

$$p_{pre} = \prod_{elem} p_{pre}^{elem}. \tag{4}$$

The probability, p_{elem}, is the product of the isotope probabilities, p_{iso}, of all isotopes iso of an element:

$$p_{pre}^{elem} = \prod_{iso} \binom{N^{elem} - \sum\limits_{iso} N^{iso}}{N^{iso}} p_{iso}^{N^{iso}}, \tag{5}$$

where N^{elem} is the number of atoms of an element in the precursor, e.g. for the above example $N^{Si} = 3$ and $N^{Nitrogen} = 2$ and p_{iso} is the natural abundance of an isotope, e.g. $p_{Si30} = 0.03092$. The number of atoms of an element in the fragment is represented by n^{elem}, which is $n^{Si} = 2$ and $n^{Nitrogen} = 1$ for the above example. The number of atoms of a specific isotope in the precursor and in the fragment is described by N^{iso} and n^{iso}, respectively. A possible valid set for the above example with $N^{Si} = 3$ is: $N^{Si28} = 0$, $N^{Si29} = 2$, and $N^{Si30} = 1$. For instance, the element probability for Si for the isotope set $N^{Si28} = 0$, $N^{Si29} = 2$, and $N^{Si30} = 1$ (see Figure 19) is computed as follows:

$$\begin{aligned} p_{pre}^{Si} &= \binom{3}{0} p_{Si28}^{0} \binom{3-0}{2} p_{Si29}^{2} \binom{3-2}{1} p_{Si30}^{1} \\ &= 1 \cdot 1 \cdot 3 \cdot 0.04685^2 \cdot 1 \cdot 0.03092^1 \\ &= 0.000203601 \end{aligned} \tag{6}$$

The fragment (or product ion) probability, p_{frag}, is again a product of element probabilities.

$$p_{frag} = \prod_{elem} p_{frag}^{elem}. \tag{7}$$

The fragment element probability, p_{frag}^{elem}, can be seen as a probability that is determined by drawing processes, where n^{elem} samples are drawn

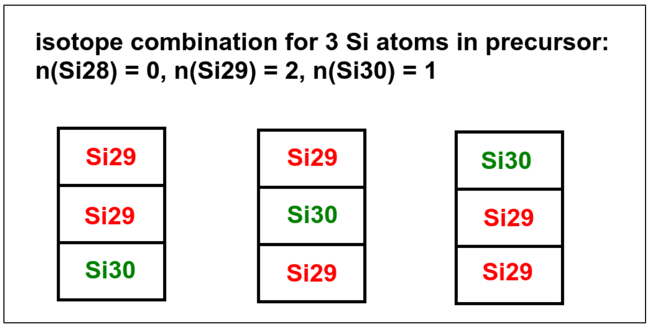

isotope combination for 3 Si atoms in precursor:
n(Si28) = 0, n(Si29) = 2, n(Si30) = 1
Si29
Si29
Si30
Si29
Si30
Si29
Si30
Si29
Si29

Figure 19: Illustration for the computation of the isotope probability for a specific isotope set ($N^{Si28} = 0$, $N^{Si29} = 2$, and $N^{Si30} = 1$).

from a pool of N^{elem} pieces (see figure 20). Hence, the fragment element probability is given by:

$$p^{elem}_{frag} = \prod_{iso} \binom{n^{elem} - \sum_{iso} n^{iso}}{n^{iso}} \prod_{i=0}^{n^{iso}-1} \frac{N^{iso} - i}{N^{elem} - \sum n^{iso} - i}. \tag{8}$$

For the example shown in figure 20 the fragment element probability is computed as follows:

$$\begin{aligned} p^{elem}_{frag} &= \binom{4}{1}\tfrac{2}{8}\binom{4-1}{2}\tfrac{4}{7}\tfrac{3}{6}\binom{4-3}{1}\tfrac{2}{5} \\ &= 4 \cdot \tfrac{1}{4} \cdot 3 \cdot \tfrac{2}{7} \cdot 1 \cdot \tfrac{2}{5} \\ &= \tfrac{12}{35} \\ &= 0.34286 \end{aligned} \tag{9}$$

The fragment probability for the example with $N^{Si28} = 0$, $n^{Si29} = 2$, and $N^{Si30} = 1$ (see figure 19) and a product ion isotope set $n^{Si28} = 0$, $n^{Si29} = 1$, and $n^{Si30} = 1$ is calculated by the following term:

$$\begin{aligned} p^{elem}_{frag} &= \binom{2}{0}\binom{2-0}{1}\tfrac{2}{3}\binom{2-1}{1}\tfrac{1}{2} \\ &= 1 \cdot 2 \cdot \tfrac{2}{3} \cdot 1 \cdot \tfrac{1}{2} \\ &= 0.66667 \end{aligned} \tag{10}$$

Multiplying the sub-probabilities yields the total probability, p_{tot}, of a combination of isotope sets. A subset of all combinations and their total probabilities for the above example are listed in table 8.

Finally, the correction matrix CM is computed by walking through the list of measured spectrum data which are labeled/tagged with their total precursor and total product ion tandem mass isotopomer index (e.g. M3.m1). This tag is used to lookup all entries in the complete list of isotope set combinations (see table 8) which have the same total mass pair. Next, for each found entry the isotope pair of the tracer ($N^{tracer(C13)}/n^{tracer(C13)}$) is

index	N^{N15}	n^{N15}	N^{Si29}	N^{Si30}	n^{Si29}	n^{Si30}	$N^{tracer(C13)}$	$n^{tracer(C13)}$	M	m	p_{tot}
0	0	0	0	0	0	0	0	0	0	0	0.77866
1	1	0	0	0	0	0	0	0	1	0	0.00284
2	1	1	0	0	0	0	0	0	1	1	0.00284
3	2	1	0	0	0	0	0	0	2	1	1.0393e-05
4	0	0	1	0	0	0	0	0	1	0	0.03956
5	1	0	1	0	0	0	0	0	2	0	0.00014
6	1	1	1	0	0	0	0	0	2	1	0.00014
7	2	1	1	0	0	0	0	0	3	1	5.2795e-07
8	0	0	1	1	0	0	0	0	1	1	0.07911
9	1	0	1	1	0	0	0	0	2	1	0.00029
10	1	1	1	1	0	0	0	0	2	2	0.00029

Table 8: Total probability, p_{tot}, for a subset of all relevant isotope combinations. The sum of the mass of the precursor and of the product ion are denoted by M and m, respectively. Note that in this table *ICT* 's internal data structure is used for displaying the isotope sets. This internal data structure does not store the lightest isotope (^{14}N, ^{28}Si, and tracer(^{12}C)), as the values of these isotopes are redundant and can be computed by subtracting the sum of the atoms of the heavy isotopes from the total number of atoms of an element.

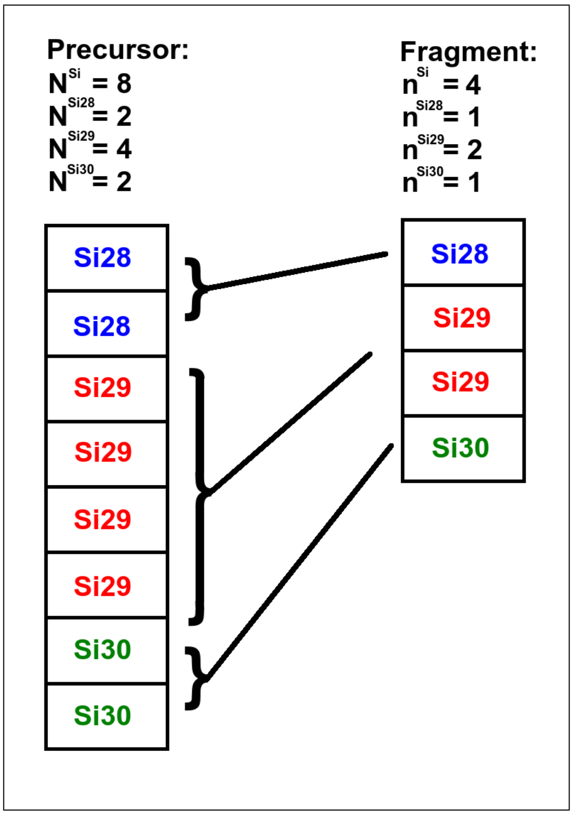


Figure 20: Illustration of the drawing process to compute the fragment probability of a participating element.

	M0.m0	M1.m0	M1.m1	M2.m0	M2.m1	M2.m2	M3.m1	M3.m2	M3.m3	M4.m2	M4.m3	M5.m3
M0.m0	0.7787	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
M1.m0	0.0424	0.7787	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
M1.m1	0.0820	0.0000	0.7787	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
M2.m0	0.0263	0.0424	0.0000	0.7787	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
M2.m1	0.0045	0.0820	0.0424	0.0000	0.7787	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
M2.m2	0.0545	0.0000	0.0820	0.0000	0.0000	0.7787	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
M3.m1	0.0028	0.0045	0.0263	0.0820	0.0424	0.0000	0.7787	0.0000	0.0000	0.0000	0.0000	0.0000
M3.m2	0.0030	0.0545	0.0045	0.0000	0.0820	0.0424	0.0000	0.7787	0.0000	0.0000	0.0000	0.0000
M3.m3	0.0029	0.0000	0.0545	0.0000	0.0000	0.0820	0.0000	0.0000	0.7787	0.0000	0.0000	0.0000
M4.m2	0.0018	0.0030	0.0028	0.0545	0.0045	0.0263	0.0820	0.0424	0.0000	0.7787	0.0000	0.0000
M4.m3	0.0002	0.0029	0.0030	0.0000	0.0545	0.0045	0.0000	0.0820	0.0424	0.0000	0.7787	0.0000
M5.m3	0.0001	0.0002	0.0018	0.0029	0.0030	0.0028	0.0545	0.0045	0.0263	0.0820	0.0424	0.7787

Table 9: Final correction matrix CM for the presented example. Note that correction matrix is a lower triangular matrix.

retrieved. The tuple of *total mass pair* and *tracer isotope pair* determines the location/index where the currently processed isotope combination (see table 8) contributes to the correction matrix with its total probability p_{tot}.

The lookups are realized by a hash table (associated array). Hash tables are a native variable type in Perl and allow fast access to data that are stored as key-value-pairs. The complete correction matrix for the example above is shown in table 9.

7.9.1 Natural abundance of tracer

The natural abundance of the tracer is considered by introducing an extra element to the workflow described above. For instance, if a correction of a ^{13}C labeling experiment without accounting for the natural abundance of the tracer considers silicon and nitrogen, the following elements would be present in the above workflow, lists and data structures: *Si*, *N*, and *tracer*. If now the natural abundance is to be considered, then the list would be extended by *nat_abu_tracer*. As it is assumed that it is a ^{13}C labeling experiment, the chemical element carbon would be used for the *nat_abu_tracer* element. The possible isotope combinations of this extra element *nat_abu_tracer* are computed very similar to the isotope combinations of the regular elements *Si*, *N*, and *tracer*. However, a special rule is used to filter any irrelevant combination as early as possible, as the sum of the atoms in the element *nat_abu_tracer* and *tracer* for the precursor and for the product ion must always be equal to the number of potentially labeled atoms.

7.9.2 Tracer (im)purity

In order to take into account an impurity of the tracer, a special step is introduced during the correction procedure. This step is performed after the complete list of isotope combinations has been computed and before the correction matrix is built. This extra step extends the list of isotope combinations and introduces an extra element *purity*. For each isotope combi-

index	$N^{tracer(C13)}$	$n^{tracer(C13)}$	$N^{purity_C 12}$	$n^{purity_C 13}$	$N^{purity_C 13}$	$n^{purity_C 13}$	change of M	change of m
0	3	2	3	2	0	0	-3	-2
1	3	2	2	2	1	0	-2	-2
2	3	2	2	1	1	1	-2	-1
3	3	2	1	1	2	1	-1	-1
4	3	2	1	0	2	2	-1	0
5	3	2	0	0	3	2	0	0

Table 10: The replacement of tracer isotopes (^{13}C) by other isotopes (^{12}C) results in the change of the total mass.

	M0.m0	M1.m0	M1.m1	M2.m0	M2.m1	M2.m2	M3.m1	M3.m2	M3.m3	M4.m2	M4.m3	M5.m3
M0.m0	0.7787	0.0779	0.0779	0.0078	0.0078	0.0078	0.0008	0.0008	0.0008	0.0001	0.0001	0.0000
M1.m0	0.0424	0.7050	0.0042	0.1406	0.0705	0.0004	0.0141	0.0071	0.0000	0.0014	0.0007	0.0001
M1.m1	0.0820	0.0082	0.7090	0.0008	0.0709	0.1410	0.0071	0.0141	0.0211	0.0014	0.0021	0.0002
M2.m0	0.0263	0.0408	0.0026	0.6386	0.0041	0.0003	0.0639	0.0004	0.0000	0.0064	0.0000	0.0006
M2.m1	0.0045	0.0742	0.0386	0.0148	0.6420	0.0077	0.1280	0.1276	0.0011	0.0255	0.0191	0.0038
M2.m2	0.0545	0.0055	0.0792	0.0005	0.0079	0.6460	0.0008	0.0646	0.1915	0.0065	0.0191	0.0019
M3.m1	0.0028	0.0043	0.0239	0.0672	0.0371	0.0048	0.5815	0.0074	0.0007	0.1156	0.0011	0.0173
M3.m2	0.0030	0.0494	0.0043	0.0098	0.0717	0.0352	0.0143	0.5849	0.0104	0.1166	0.1734	0.0346
M3.m3	0.0029	0.0003	0.0493	0.0000	0.0049	0.0762	0.0005	0.0076	0.5890	0.0008	0.0589	0.0059
M4.m2	0.0018	0.0029	0.0027	0.0447	0.0041	0.0218	0.0650	0.0338	0.0065	0.5298	0.0100	0.1570
M4.m3	0.0002	0.0026	0.0027	0.0005	0.0447	0.0042	0.0089	0.0690	0.0321	0.0138	0.5333	0.1063
M5.m3	0.0001	0.0001	0.0017	0.0023	0.0026	0.0026	0.0405	0.0040	0.0199	0.0625	0.0309	0.4831

Table 11: Final correction matrix CM for the presented example if impurity of the tracer material was considered. Note that correction matrix is not a lower triangular matrix if the tracer is not perfectly pure.

nation the tracer isotope pair ($N^{tracer(C13)}/n^{tracer(C13)}$) is analyzed and the combination is duplicated and adapted for each possible impurity situation. E.g. if $N^{tracer(C13)} = 3$, then either one, two or all three labeled atoms might be occupied by a ^{12}C isotope. The replacement of tracer isotopes (e.g. ^{13}C) by other isotopes (e.g. ^{12}C) also results in the change of the total mass of the isotopes combinations (see table 10).

The probability that such a 'replacement' occurs is determined by the purity values provided by the user. The total probability, p_{tot}, of each combination is modified accordingly. Assuming that 10% of the tracer isotopes are ^{12}C instead of ^{13}C, results - on the basis of the aforementioned example - in the correction matrix shown in table 11. Note that the correction matrix is not a lower triangular matrix anymore, as now corrected values of mass pairs with high masses (e.g. M4.m3) might be effected by a situation where the traced atoms were 'lost' by an impurity of the tracer material.

References

[1] Christoph Wittman and Elmar Heinzle. Mass Spectrometry for Metabolic Flux Analysis. *Biotechnol Bioeng.*, 62(6):739–750, 1999.

[2] Wouter A. van Winden, Christoph Wittman, Elmar Heinzle, and Joseph J. Heijnen. Correcting Mass Isotopomer Distributions for Naturally Occurring Isotopes. *Biotechnol Bioeng.*, 80(4):477–479, 2002.

[3] Hunter N. B. Moseley. Correcting for the effects of natural abundance in stable isotope resolved metabolomics experiments involving ultra-high resolution mass spectrometry. *BMC Bioinformatics*, 11:139, 2010.

[4] William J. Carreer, Robert M. Flight, and Hunter N. B. Moseley. A Computational Framework for High-Throughput Isotopic Natural Abundance Correction of Omics-Level Ultra-High Resolution FT-MS Datasets. *Metabolites*, 3:853–866, 2013.

[5] Hunter N. B. Moseley. Pynac: High-throughput natural abundance correction software for stable isotope resolved metabolomics (SIRM) experiments. `http://files.cesb.uky.edu/software/PYNAC/`, 2013. [Online; accessed 20-March-2015].

[6] Ari Rantanen, Juho Rousu, Juha T. Kokkonen, Virpi Tarkiainen, and Raimo A. Ketola. Computing positional isotopomer distributions from tandem mass spectrometric data. *Metabolic Engineering*, 4(4):285–294, 2002.

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

[8] Zeeshan Ahmed, Saman Zeeshan, Claudia Huber, Michael Hensel, Dietmar Schomburg, Richard Münch, Wolfgang Eisenreich, and Thomas Dandekar. Software LS-MIDA for efficient mass isotopomer distribution analysis in metabolic modelling. *BMC Bioinformatics*, 14:218, 2013.

[9] Mono - cross platform, open source .Net framework. `http://www.mono-project.com/`. [Online; accessed 22-July-2015].

[10] Pierre Millard. IsoCor: Correcting MS data in isotope labeling experiments. `http://metasys.insa-toulouse.fr/software/isocor/`, 2011. [Online; accessed 11-March-2015].

[11] Pierre Millard, Fabien Letisse, Serguei Sokol, and Jean-Charles Portais. IsoCor: Correcting MS data in isotope labeling experiments. *Bioinformatics*, 28(9):1294–1296, 2012.

[12] Artistic License. `http://opensource.org/licenses/artistic-license`. [Online; accessed 24-March-2015].

[13] GNU General Public License. `http://www.gnu.org/licenses/#GPL`. [Online; accessed 24-March-2015].

[14] K. J. R. Rosman and P. D. P. Taylor. Isotope Composition of the Elements 1997. *Pure & Appl. Chem.*, 1:217–235, 1998.