Supplemental Documentation

Phenotyping the Carbon-Partitioning Kinetics of Excess and Limited Inorganic Carbon Supply in Cells of the Wild Type and Photorespiratory Mutants of the Cyanobacterium *Synechocystis* sp. Strain PCC 6803

Jan Huege[1], Jan Goetze[2], Doreen Schwarz[3], Hermann Bauwe[3], Martin Hagemann[3] and Joachim Kopka[1]

[1] *Max-Planck-Institut für Molekulare Pflanzenphysiologie, Am Mühlenberg 1, 14476 Potsdam-Golm, Germany (J.H., J.K.)*
[2] *Universität Potsdam, Institut für Chemie, Theoretische Chemie, Karl-Liebknecht-Str. 24-25, 14476 Potsdam-Golm, Germany (J.G.)*
[3] *Universität Rostock, Institut für Biowissenschaften, Pflanzenphysiologie, Albert-Einstein-Str. 3, 18051 Rostock, Germany (D.S., H.B., M.H.)*

CORRECTOR **Documentation - A high throughput tool for mass isotope correction of mass spectra from isotope labelling experiments**

TABLE OF CONTENTS

1. Introduction

Gas chromatography-mass spectrometry (GC-MS) based metabolite profiling experiments are an integral part of biological and medical research. A variety of tools have been made available which facilitate the high-throughput experimentation and data processing approach necessary for modern metabolome studies. As is commonly agreed, metabolome studies need to be complemented by flux investigations (e.g. Brunengraber et al. 1997; Sauer 2004; Roessner-Tunali et al. 2004; Fernie et al. 2005; Ratcliffe and Shachar-Hill 2006) in order to enhance our current knowledge of the metabolism and physiology of biological systems and to enable evidence-driven mathematical modelling of metabolic pathways. In contrast to the achievements made for high-throughput processing and screening of relative changes in metabolite pool sizes, studies which aim to reveal and quantify in vivo fluxes in metabolic networks are still hard to perform, low in throughput and lack a repertory of pre-processing software tools which are equally powerful than the tool box available for metabolome investigations.

The GC-MS based metabolite profiling technology represents a well established routine method used to monitor central metabolism. The same technological setup can be used for flux profiling, provided the investigated biological systems are challenged with stable isotope labelled nutrients or metabolites. Using GC-MS detection the metabolic fate of the labelled precursors and the time course of the isotope dilution within the metabolic networks are monitored. For this purpose GC-MS technology provides metabolite specific mass isotopomer distributions of the complete metabolite structure (molecular ions) or of electron impact induced mass fragments which represent defined moieties or substructures of the original metabolite.

The required pre-processing steps for multi parallel flux studies are (1) multiple alignment of up to hundreds of GC-MS chromatograms constituting a profiling experiment, (2) comprehensive identification of mass spectral tags (MSTs) which represent each metabolite (e.g. Kopka et al. 2005), (3) targeted retrieval of mass isotopomer distributions of respective molecular ions and mass fragments, (4) computational correction of the observed mass isotopomer distributions for the bias caused by the occurrence of naturally occurring isotopes (NOISs) and (5) calculation of the experimentally induced fractional enrichment of the fed stable isotope within each targeted mass spectral item.

Steps 1-3 are identical to conventional metabolite profiling experiments and can be performed by previously reported software applications for the non-targeted, and within the limits of technology, comprehensive data mining of GC-MS profiling experiments, such as TagFinder (Luedemann et al. 2008) or MetAlign (http://www.pri.wur.nl/UK/products/MetAlign/; e.g. DeVos et al. 2007; Lommen, 2009). The *CORRECTOR* tool described here performs steps 4-5, which are the bottle neck for the processing of large flux profiling experiments (e.g. Huege et al. 2007).

As naturally occurring isotopes will influence the composition of mass isotopomers in the typical spectra of GC-MS based MSTs, refined theory and mathematical tools have been developed which primarily target 13C-flux analysis (e.g. Fernandez et al. 1996; Wittmann and Heinzle, 1999; van Winden et al. 2002; Wahl et al. 2003). The currently available software (e.g. Wahl et al. 2003) enables the correction of mass isotopomer distributions representing a single MST vector but do not support batch processing for the required high throughput.

The *CORRECTOR* tool is a simple command line application which is configured once and then, triggered by a single command, automatically corrects large data sets of multiple MSTs which can be provided by the TagFinder software. The contribution of NOISs is subtracted from measured signals using the previously reported enhanced matrix calculation algorithm (Wittmann and Heinzle, 1999; van Winden et al. 2002; Wahl et al. 2003). *CORRECTOR* can be configured beyond 13C-labelling studies to accommodate any other biologically relevant elemental isotopes, for example 2H or 15N, and provides two types of ready to use output files which report the corrected mass isotopomer distributions and respective fractional enrichments. In the following, we will describe the implementation, the software operation and the chosen input- and output-data formats.

2. Workflow

2.1. Software requirements and source code compilation

The *CORRECTOR* C++ tool has no requirements of additional supporting software. *CORRECTOR* operates within an UNIX environment. Data input conventions and returned data-output files are delimited spread sheets in .txt-file format. As these files are used to transfer data to and from the *CORRECTOR* program package, other software, such as OpenOffice or commercial equivalents, are recommended to view and edit uploaded and returned data. The data output of the TargetFinder plug-in, which is part of the TagFinder software (Luedemann et al. 2008), can be directly used as

CORRECTOR input after deletion of surplus column and row information (see below). When attempting to compile the *CORRECTOR* source code, it should be noted that *CORRECTOR* has not been tested on other systems than UNIX environments. As we consider the programming style of the *CORRECTOR* source code to be simple and self-explaining, no problems should be expected when compiling *CORRECTOR* for other systems. Standard C++ libraries should suffice to establish an operative *CORRECTOR* version on any type of computer.

2.2. Configuration

Before application of the *CORRECTOR* tool, two configuration steps need to be performed. For this purpose two .cfg-files are included within the download package. The isoconfig.cfg file contains the isotopic composition of each element considered by the *CORRECTOR* program. The isoconfig.cfg format has separate rows for each element which are organized as follows,

type / massmin / N / F0 / F1 / ... / FN

where "type" is the element code, e.g. C for carbon, "massmin" represents the nominal mass of the lightest stable isotope of the element given as an integer value, e.g. 12 for carbon, followed by "N" indicating the number of considered isotopes, e.g. 2 namely ^{12}C and ^{13}C, and "F" defines the representative isotopic composition scaled to 1, e.g 0.9893 (^{12}C) and 0.0107 (^{13}C). Note that "N" is not the number of all existing stable isotopes, but the difference between the nominal mass of the heaviest and the nominal mass of the lightest isotope plus 1. As a consequence, all possible isotopes which have a mass lower than the heaviest stable isotope but which may not naturally occur need to be listed and assigned an abundance, F = 0. This procedure applies, for example, to the element sulphur which has the stable isotopes ^{34}S and ^{36}S (Rosman et al. 1998). Isotope ^{35}S is not stable and receives F = 0 as it can be neglected for the correction of natural isotope abundances. The default isoconfig.cfg file contains the elements H, C, N, O, Si, P and S. Additional elements can be included as they may become necessary. Published natural abundances can be taken from respective references (cf. Rosman et al. 1998). In addition, experimental conditions are conceivable, which require sub-setting of single elements into atom groups of different origin. For example, GC-MS utilizes chemical derivatization reagents, such as N-methyl-N-(trimethylsilyl)-trifluoroacetamide (MSTFA), to chemically modify metabolites and thus increase the volatility of these compounds for gas chromatographic analysis (Fiehn et al. 2000). This chemical modification introduces carbon atoms which originate not from natural sources but form chemical synthesis and may therefore bear an isotope composition different from the natural abundances. The present *CORRECTOR* distribution is capable of covering such effects, even though the included configuration files do not make use of this functionality. This is also useful for the consideration of the natural isotope discrimination effects known to occur in biological organisms or to accommodate biases caused by different mass spectral technologies, e.g. time of flight, quadrupole or ion tap detection systems.

The second and main configuration information is given by the file corrconfig.cfg. This configuration file defines the identity and properties required for mass isotopomer correction of all annotated derivatized metabolites and respective molecular ions or mass fragments. Corrconfig.cfg represents an adaptation of the supplementary file 1 published by Huege and co-authors (2007). The first row of corrconfig.cfg file represents a comment line and descibes the molecular properties of the listed items where each item (MST) is characterized by a single row. An exemplary corrconfig.cfg file is provided within the program package download. The required row format is as follows:

Identifier / masslow / masshigh / masssurplus / NA / type0 / N0 … $type_i$ / N_i / $type_{label}$ / N_{label}

An MST entry of a specific analyte is defined by the analyte identifier and the lowest and highest possible nominal mass introduced by chosen stable isotope labelling, namely "Identifier / masslow / masshigh" given in columns 1-3 (cf. Huege et al. 2007). For example, the identifier given in corrconfig.cfg "M000029_A132003-101_METB_1298_Amino_acid_Proline_(2TMS)_MP" represents the main silylation product of the metabolite proline. Two rows are defined for this identifier, i.e. the mass fragment 142-146 ("masslow"-"masshigh") and 216-220. The mass range indicates that both mass isotopomer distributions may contain up to 4 carbon atoms which can be labelled by ^{13}C-feeding. Because chemical derivatization, e.g. by MSTFA, introduces trimethylsilyl-moieties, which generate higher mass isotopomers than the expected full labelling of the metabolite, the "masssurplus" option was introduced. This criterion can be used to enter additional heavy mass isotopomers into the correction algorithm for enhanced correction, especially of samples with high fractional isotope enrichments. The default setting of "masssurplus" is zero. The user is enabled to comfortably experiment with different "masssurplus" settings without changing the experimentally fixed

"masshigh" value. The properties define the elemental composition of each mass fragment and are derived from the respective sum formula. For example, the m/z 142 fragment of the proline (2TMS) analyte has "NA" = 25 atoms and the elemental composition "type0 / N0 … $type_i$ / N_i": C 3 H 16 O 0 N 1 S 0 Si 1 and $type_{label}$ / N_{label}: C 4, where each element $type_i$ has N_i atoms. Note that in this example carbon is labelled and accordingly the carbon atoms are split into those which are introduced by chemical derivatization, C 3, and those which can be labelled, C 4. This maximum labelling information needs to be defined at the end of each row. The sum of N_i atoms should add up to "NA", whereas N_{label} should be equal to the difference "masshigh" – "masslow". The *CORRECTOR* program will create an error message if discrepancies are detected for the number of atoms. Each entry is separated by standard delimiter, either spaces, tabulators or line breaks. Identifiers or element type definitions cannot contain spaces. The type variables are compared to the isoconfig.cfg type entries and must match. Customized element type definitions for enhanced isotope correction schemes must be carefully matched between both configuration files.

2.3. Input data

The standard input format of *CORRECTOR* can be created or edited with conventional spread sheet processing programs. An example is given in Fig. 1. The input convention requires samples to be defined as columns which are described by rows that contain the information of the measured intensities of the chromatographically aligned nominal masses observed in all co-processed samples. The use of data matrices, which are aligned by co-elution of mass fragments that represent the same compound and by sequence of observed nominal masses, enhances processing time and facilitates co-processing of hundreds of samples and MST annotations. Within the data matrix each MST identifier needs to be assigned only once to the respective entries within the aligned data matrix. *CORRECTOR* parses the annotated data matrix for MST identifiers, processes all respective sample entries linked to this identifier and then moves on to the subsequent identifier within the annotated data matrix. Note that the MST identifiers used to annotate the data input matrix must match exactly to the identifiers defined in the corrconfig.cfg file. Also note that *CORRECTOR* expects rows within an identifier spectrum to be sorted ascending by nominal masses of the observed mass fragments. The order of the identifier sets themselves is of no concern.

Input

Name_Analyte	Tag_Mass	chromatogram_x+1	chromatogram_x+2	chromatogram_x+i
identifier_a	min	intensity_value	intensity_value	intensity_value
…	…	…	…	…
identifier_a	max	intensity_value	intensity_value	intensity_value
identifier_b	min	intensity_value	intensity_value	intensity_value
…	…	…	…	…
identifier_b	max	intensity_value	intensity_value	intensity_value
identifier_c	min	intensity_value	intensity_value	intensity_value
…	…	…	…	…
identifier_c	max	intensity_value	intensity_value	intensity_value
…	…	…	…	…
identifier_x	min	intensity_value	intensity_value	intensity_value
…	…	…	…	…
identifier_x	max	intensity_value	intensity_value	intensity_value

Fig. 1. ***The data import convention of CORRECTOR.*** **The first row of the data input file of *CORRECTOR* contains the column headers and sample/chromatogram identifiers; spaces are not allowed. Columns are indicated by standard delimiters. Missing mass abundance information must be substituted by 0. The first two columns are fixed to contain the MST identifier (Name_Analyte) and nominal mass (Tag_Mass). Note that *CORRECTOR* expects the matrix to be sorted within one MST identifier ascending by Tag_Mass.**

2.4. Data processing and output data

To ensure system independency, the *CORRECTOR* program was written in C++. Given the appropriate mass spectral vector for an analyte, ranging from minimum to maximum mass, the program creates element specific correction matrices. For each chemical element (respectively atom type), a separate correction matrix is created. Thus multiple correction steps may be required, which are handled by a loop structure. Beforehand, another loop structure applies the correction matrix of a specific atom type to the same mass vector in every sample in the data array, thereby eliminating the

need to recalculate the matrix for each sample. The matrices are created according to principles given elsewhere (Wittmann and Heinzle 1999, van Winden et al. 2002, Wahl et al. 2003).

The construction of each isotopologue for the correction matrix elements occurs by setting up a counting scheme: The number of possible isotopes of one atom type represents the maximum of available digits. The numerical base of the counting scheme is set to the total number of atoms of the respective type plus one. The base is then used by the program to count every possible isotope combination representing one isotopologue up to the point when all isotopologues are found, which is the number whose highest place contains the total number of atoms. In this scheme, all numbers with the sum of digits equal to the base represent a valid isotopologue, which only needs to be weighted according to its mass and then put into place in the correction matrix. This way, no isotopologue occurs twice and the algorithm can finish the process immediately after all isotopologues are found. This method does not require error checking, as it counts all isotopologues once and only once. To improve speed, the program does not count every number in the counting scheme but adds the total number of atoms in every step, as all valid combinations are multiples of the total atom number. This is a general property because every number z represented in a base n can be checked on its divisibility by n-1 through the sum of digits.

If masses within the specified mass range of a vector are not provided by the input, they are considered zero. The required values for the correction matrix such as the number of total atoms of a given atom type (carbon, oxygen, hydrogen etc.) are taken from one of two reference files (corrconfig.cfg), the properties of each atom type in terms of isotopic composition from a second reference file (isoconfig.cfg).

The program expects a data matrix as input in which samples are defined as columns, which are described by rows that contain the information of the measured intensities of the aligned masses observed in all co-processed samples (Luedemann et al. 2008).

Data processing is performed as follows (cf. Figure 2): First, the number of total parallel samples is read by parsing the first line of the input file, which is forwarded to both output files. Next, the program sequentially processes the provided mass spectral data according to identifiers. The identifier characterises a specific mass spectral vector and connects the data in the input file to a specific entry in the corrconfig.cfg. This enables the program to find the mass intensities of the vector elements with which to perform the correction. When the starting mass of a vector is found, an array of the dimensions (masses of vector x number of samples) is created and filled by parsing the values found in the input. Once all required masses of a vector are read, the parser continues until facing the starting mass for the next vector or a different identifier. The input parser waits for the correction of the already parsed vectors to be finished.

After dividing the mass vectors of each sample by the correction matrices, the corrected mass vector explicitly requested by the corrconfig.cfg is exported to Output 01 (cf. Figure 3). Output 02 (cf. Figure 3) receives the results for the calculated fractional isotopic enrichment of the respective vector (atomic percentage). Figure 4 exemplifies the application of *CORRECTOR* to a single empirical mass isotopomer distribution.

Error messages are produced in case no matching identifier in the corrconfig.cfg is found or if double or unsorted masses are encountered.

As a result of utilizing the above mentioned matrix calculation concept (Wittmann and Heinzle 1999, van Winden et al. 2002, Wahl et al. 2003), the corrected intensity values of low intensity vector elements or vector elements with intensities of 0 can drop below zero after correction. Since values below zero are unphysical, the algorithm in *CORRECTOR* sets those negative values to zero before using them in the next correction step.

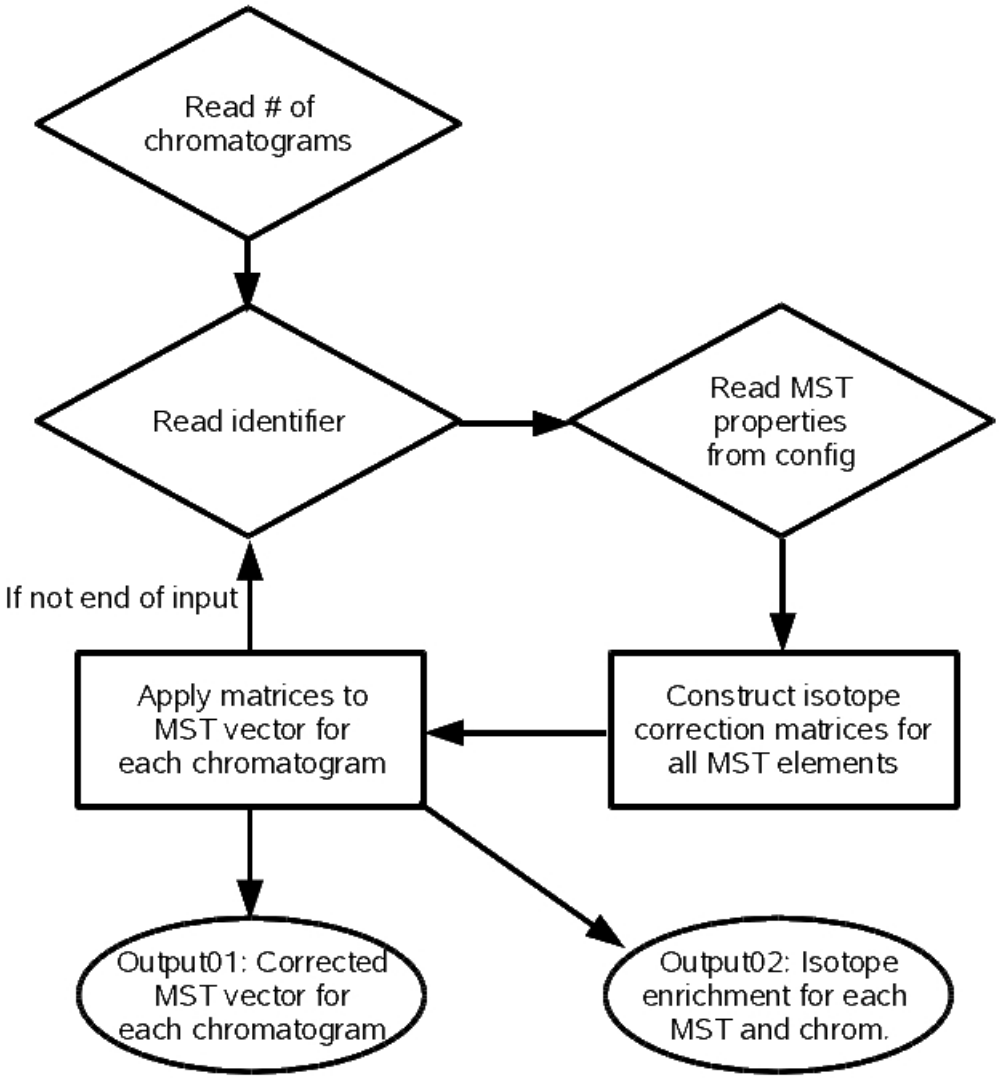

Fig. 2. ***The CORRECTOR work flow chart.***

Output 01

Name_Analyte	Tag_Mass	chromatogram_x+1	chromatogram_x+2	chromatogram_x+i
identifier_a	min_fragment_a	intensity_value	intensity_value	intensity_value
...	...	...	...	...
identifier_a	max_fragment_a	intensity_value	intensity_value	intensity_value
identifier_a	min_fragment_b	intensity_value	intensity_value	intensity_value
...	...	...	...	...
identifier_a	max_fragment_b	intensity_value	intensity_value	intensity_value
...	...	...	...	...
identifier_x	min_fragment_a	intensity_value	intensity_value	intensity_value
...	...	...	...	...
identifier_x	max_fragment_a	intensity_value	intensity_value	intensity_value

Output 02

Name_Analyte	Tag_Mass	chromatogram_x+1	chromatogram_x+2	chromatogram_x+i
identifier_a	min_fragment_a	isotopic_enrichment	isotopic_enrichment	isotopic_enrichment
identifier_a	min_fragment_b	isotopic_enrichment	isotopic_enrichment	isotopic_enrichment
identifier_a	min_fragment_c	isotopic_enrichment	isotopic_enrichment	isotopic_enrichment
identifier_b	min_fragment_a	isotopic_enrichment	isotopic_enrichment	isotopic_enrichment
identifier_b	min_fragment_b	isotopic_enrichment	isotopic_enrichment	isotopic_enrichment
...	...	...	...	...
identifier_x	min_fragment_a	isotopic_enrichment	isotopic_enrichment	isotopic_enrichment

Fig. 3. ***The data output format of CORRECTOR.*** **Two data output files are generated. Output 01 contains the NOIS corrected mass isotopomer distribution vectors in input format. Output 02 lists the respective isotopic fractional enrichments.**

2.5. Installation and Basic Usage

Installation:

1 create a folder 'CORRECTOR' (recommended)
2 unpack the file '**CORRECTOR_package.zip**' into this folder

Basic Usage:

1. copy input file containing experimental mass spectral data into the *CORRECTOR* directory
2. start command line tool (e.g. terminal)
3. change directory to *CORRECTOR* directory
4. start correction procedure by running *CORRECTOR*:
   CORRECTOR.exe name_input_file name_output01 name_output02.

Options: -igz [integer]; will ignore up to [integer] empty or negative mass peaks for the enrichment calculation (atom percentages) after the first positive mass peak. Useful for high enrichment experiments with a leading peak followed by a series of empty peaks. Default is 0.

File Names:
The definition of outputA and outputB file names is required since there are no respective default settings.

Error Log:
An error log is provided for cases of encountered processing problems. This file will be automatically overwritten by each application run. We recommend archiving of error log files at a different designated location.

Test Run: `./corrector.exe [Optional: -igz [integer]] inputfile outputfile1(raw data) outputfile2(atom percentages)`

After a test run with the provided test files ('input_test.txt') two output files should have been returned by the *CORRECTOR* program. Output files and error log should provide the same content as given by the files in the example folder, since 'input_test.txt' file contains the same mass spectral information as 'input_exemplar.txt' file in this folder.
The 'Input_test.txt' file contains mass spectral data from a ^{13}C-delabelling experiment (cf. Huege et al. 2007). Plants had been grown in a $^{13}CO_2$ atmosphere and were subsequently exposed to an ambient atmosphere ($^{12}CO_2$). A kinetic series of samples was taken. The sample or chromatogram identifier in the second row was substituted by the time point of the sample. These data enable a direct plot of the calculated ^{13}C-enrichment over time.
To exemplify the function of the error log the 'input_exemplar_with_errors.txt' file containing two deliberate mistakes is provided. Firstly, the identifier 'Malic_acid_(3TMS)' (rows 789-895) does not match to any identifier in 'corrconfig.cfg' file; secondly, in row 898 mass m/z=245 for 'M000067_A137001-101_METB_1346.23_Acid_(Dicarboxylic)_Fumaric_acid_(2TMS)' occurs twice. In both cases the correction procedure is not applied due to a flawed input. By substitution of 'Malic_acid_(3TMS)' identifier with identifier 'M000065_A149001-101_METB_1477.3_Acid_(Dicarboxylic,_Hydroxy-)_Malic_acid_(3TMS)' contained in the 'corrconfig.cfg' and removal of the duplicated mass information within the fumaric acid data, the correction procedure should be performed without reported errors.

Should the compilation of an executable windows version of *CORRECTOR* with the provided source code be difficult or if an UNIX system is not available ("Windows-Only-User"), the following description may help to utilize the provided *CORRECTOR* executables.

Installation:

1. create a folder 'CORRECTOR' (recommended) on an **external drive** such as a flash drive or external hard drive
2 unpack the file '**CORRECTOR_package.zip**' into this folder

Basic Usage:

1. copy the input file containing your mass spectral data into the *CORRECTOR* directory

2.) put a 'LiveCD' or an installation disc containing a 'Live' option of a UNIX operating system into your optical drive, reboot your computer, allow it to boot from the respective UNIX disc and use the 'Live' option.

3.) proceed with step 2. of the `Basic Usage´ as was mentioned above

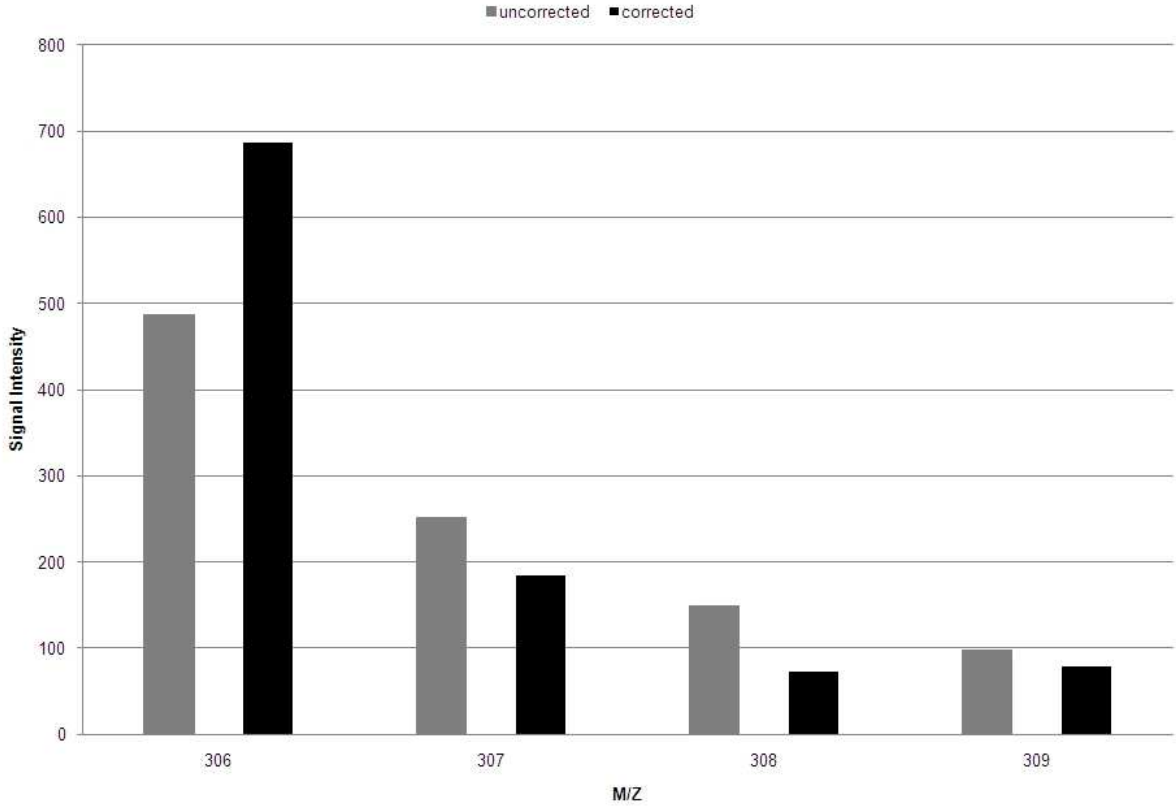

Fig.4. ***Visualization of a single corrected mass isotopomer distribution vector compared to the uncorrected initial recording.*** **The mass isotopomer range represents the M-15*$^{+}$ fragment of Serine (3TMS) and represents the full carbon backbone of the metabolite. The apparent isotopic enrichment of the uncorrected isotopomer distribution is 28.46 atom%, whereas the corrected ^{13}C-isotopic enrichment is 18.47 atom%.**

3. References

Brunengraber,H., Kelleher,J.K., Des Rosiers,C. (1997) Applications of mass isotopomer analysis to nutrition research. *Annu. Rev. Nutr.*, **17**, 559-596.

De Vos,R.C.H., Moco,S., Lommen,A., Keurentjes,J.J.B., Bino,R.J., Hall,R.D. (2007) Untargeted large-scale plant metabolomics using liquid chromatography coupled to mass spectrometry. *Nature Protocols,* **2**, 778 - 791

Fernandez,C.A., Des Rosier,C., Previs,S.F., David,F., Brunnengraber,H. (1996) Correction of ^{13}C mass isotopomer distributions for natural stable isotope abundance. *J. Mass Spectrometry*, **31**, 255-262.

Fernie,A.R., Geigenberger,P., Stitt,M. (2005). Flux an important, but neglected, component of functional genomics. *Curr. Opin. Plant Sci.*, **8**, 174-182.

Fiehn,O., Kopka,J., Dörmann,P., Altmann,T., Trethewey,R.N., Willmitzer,L. (2000) Metabolite profiling for plant functional genomics. *Nat. Biotechnol.*, **18**, 1157-1161.

Huege,J., Sulpice,R., Gibon,Y., Lisec,J., Koehl,K., Kopka,J. (2007) GC-EI-TOF-MS analysis of in vivo carbon-partitioning into soluble metabolite pools of higher plants by monitoring isotope dilution after ($^{13}CO_2$)-labelling. *Phytochemistry*, **68**, 2258-2272.

Kopka,J., Schauer,N., Krueger,S., Birkemeyer,C., Usadel,B., Bergmüller,E., Dörmann,P., Gibon,Y., Stitt,M., Willmitzer,L., Fernie,A.R., Steinhauser,D. (2005) GMD@CSBDB: The Golm metabolome database. *Bioinformatics*, **21**, 1635-1638.

Lommen,A. (2009) MetAlign: Interface-Driven, Versatile Metabolomics Tool for Hyphenated Full-Scan Mass Spectrometry Data Preprocessing. *Anal. Chem.,* **81**, 3079-86.

Luedemann,A., Strassburg,K., Erban,A., Kopka,J. (2008) TagFinder for the quantitative analysis of gas chromatography - mass spectrometry (GC-MS) based metabolite profiling experiments. *Bioinformatics,* **24**, 732 -737.

Ratcliffe,R.G., Shachar-Hill,Y. (2006) Measuring multiple fluxes through plant metabolic networks. *Plant J.* , **45**, 490-511.

Roessner-Tunali,U., Liu,J., Leisse,A., Balbo,I., Perez-Melis,A., Willmitzer,L., Fernie, A.R. (2004) Kinetics of labelling of organic and amino acids in potato tubers by gas chromatography-mass spectrometry following incubation in ^{13}C labelled isotopes. *Plant J.*, **39**, 668–679.

Rosman,K.J.R., Taylor,P.D.P. (1998) Isotopic compositions of the elements 1997. *Pure Appl. Chem.*, **70**, 217–235.

Sauer, U. (2004) High-throughput phenomics: experimental methods for mapping fluxomes. *Curr. Opin. Biotechnol.*, **15**, 58-63.

Selivanov,V.A., Marin,S., Lee,P.W.N., Cascante,M. (2006) Software for dynamic analysis of tracer-based metabolomic data: estimation of metabolic fluxes and their statistical analysis. *Bioinformatics*, **22**, 2806-2812.

van Winden,W.A., Wittmann,C, Heinzle,E, Heijnen,J. (2002) Correcting Mass Isotopomer Distributions for Naturally Occurring Isotopes. *Biotechnology and Bioengineering*, **80**, 477-479.

Wahl,A.S., Dauner,M., Wiechert,W. (2003) New Tools for Mass Isotopomer Data Evaluation in ^{13}C Flux Analysis: Mass Isotope Correction, Data Consistency Checking, and Precursor Relationships. *Biotechnology and Bioengineering*, **85**, 259-268.

Wittmann,C., Heinzle,E. (1999) Mass Spectrometry for Metabolic Flux Analysis. *Biotechnology and Bioengineering*, **62**, 739-750.

Wittmann,C. (2007) Fluxome analysis using GC-MS. *Microbial Cell Factories*, **6**, 6.

Yoo,H., Antoniewicz,M.R., Stephanopoulos,G., Kelleher,J.K. (2008) Quantifying Reductive Carboxylation Flux of Glutamine to Lipid in a Brown Adipocyte Cell Line. *J Biol Chem*, **283,** 20621–20627.

4. Availability

CORRECTOR is made freely available from http://www-en.mpimp-golm.mpg.de/03-research/researchGroups/01-dept1/Root_Metabolism/smp/CORRECTOR/index.html.
Supplementary documentation, exemplary data and operating instructions are made available with the *CORRECTOR* download from the above URL.

5. Contact

jgoetze@uni-potsdam.de, huege@mpimp-golm.mpg.de, kopka@mpimp-golm.mpg.de

6. Authors' contributions

JG programmed *CORRECTOR* and drafted the documentation. JH evaluated and tested *CORRECTOR*, provided feedback and drafted the documentation.

7. Acknowledgments

J.G. acknowledges the C++ programming advice of PD Dr. Tillmann Klamroth, University of Potsdam, Institute of Chemistry.

8. License

This work (*CORRECTOR*) is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-sa/3.0/ or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

9. Disclaimer

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)

ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.