Supplementary

1.1 3DSpectra workflow

For the convenience of the reader we report here the sequence of the main steps executed by the 3DSpectra algorithm:

```
for each peptide in the peptide library
       collect sequence info
       collect charge info

       if the peptide wasn't analyzed before
                ▪ retrieve peptide metadata
                ▪ check in library whether also the isotopic partner was identified
                  otherwise estimate isotopic partner position

                ▪ both for peptide and partner:
                ▪ access data by mzRTree optimized access
                ▪ compute isotopic distribution features
                ▪ detect main peak of the isotopic distribution
                      • refine window of interest to look for elution time using only
                        3 most intense isotopes
                      • for each chromatogram identify putative elution time
                      • refine the max peak position, and center the data matrix
                        accordingly
                ▪ fit the GMM by ML implemented via EM
                      • model fitted starting with prior info as for mean, variance
                        and probability for each component
                      • best model choice by AIC
                      • pdf and isopdf curves computation
                      • clustering data according to the GMM
                      • computing clusters features
                      • selection of putative noise component of the GMM
                      • application of the signal mask according to the isopdf curves
                        and the noise component
                ▪ execution of the quantification module
                      • signal processing: background removal and Savitzky-Golay
                        smoothing
                      • fit of the isotopic distribution model on each spectrum using
                        WLLS (each spectrum component is weighted by its probability)
                      • calculation of the max of the resulting chromatograms
                      • fit of the isotopic distribution model on the resulting unique
                        spectrum using WLLS (each spectrum component is weighted by
                        its probability)
                ▪ add current peptide to the list of already analyzed peptides
       else
              go to next peptide in the peptide library
       end

       save current progress
end
```

1.2 3D isotopic distribution model

Here, the step involving the fit of a 3D isotopic distribution model by using a Finite Mixture Modeling (FMM) approach [1] is described in detail.

1.2.1 The Gaussian mixture model

We assume that data vectors $X = \left\{x_1, x_2, \dots, x_n | \, x_i = \left(\frac{m}{z}_i, rt_i\right), \forall \, i = 1, \dots, n\right\}$ are independent and identically distributed with distribution p, parameters of which are represented by Θ. Thus, recalling the maximum likelihood estimation principle:

$$p(X|\Theta) = p(x_1, x_2, \dots, x_n|\Theta) = \prod_{i=1}^{n} p(x_i|\Theta) = L(\Theta|X) \tag{1}$$

where $p(X|\Theta)$ is equal to the likelihood function $L(\Theta|X)$ of the parameters Θ given the data X. $L(\Theta|X)$ is a function of the parameters Θ where the data X are fixed. The ML parameters estimate $\hat{\Theta}$ is given by the maximization of the likelihood function $L(\Theta|X)$, or of its log:

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmax}} \log\left(L(\Theta|X)\right) \tag{2}$$

Data on m/z dimension can be described by a sum of Gaussians distribution and its shape factors are defined by the theoretical isotopic distribution of the peptide [2]. Therefore the peptide distribution can be modeled as a probabilistic bivariate Gaussian Mixture Model:

$$p(X|\Theta) = \sum_{k=1}^{N} \alpha_k p_k(X|\theta_k) \tag{3}$$

where the parameters are $\Theta = (\alpha_k, \theta_k), \; k = 1, \dots, N$. Mixing proportions α_k are such that $\sum_{k=1}^{N} \alpha_k = 1$. Each p_k is a bivariate Gaussian PDF parameterized by $\theta_k = (\mu_k, \Sigma_k), \; k = 1, \dots, N$, where μ_k is the mean vector and Σ_k is the covariance matrix of the k^{th} Gaussian component. The GMM consists of as many Gaussian density components as is the number N of peaks considered for the theoretical isotopic distribution of the peptide. The $log(L(\Theta|X))$ to be maximized to estimate Θ is:

$$log\big(L(\Theta|X)\big) = log \prod_{i=1}^{n} p(x_i|\Theta) = \sum_{i=1}^{n} \log\left(\sum_{k=1}^{N} \alpha_k p_k(x_i|\theta_k)\right) \tag{4}$$

1.2.2 Expectation maximization for the GMM

The $log(L(\Theta|X))$ for the GMM is difficult to optimize because it contains the log of the sum. Here, the EM algorithm [3,4] was used, which is widely used in the computational pattern recognition community.

The hypothesis is that the observed data X is an incomplete set of data drawn from the distribution of which we want to estimate the parameters. The EM defines a complete dataset $Z = (X, Y)$ where $Y = \left\{ y_i \in \{1, \dots, N\} \mid x_i = \left(\frac{m}{z}_i, rt_i\right) \in G_k \leftrightarrow y_i = k, \forall\, i = 1, \dots, n \right\}$ is unknown and G_k is the k^{th} Gaussian component of the GMM. Therefore the log-likelihood function $\log\left(L(\Theta|X)\right)$ is substituted with:

$$\log\big(L(\Theta|Z)\big) = \log\big(L(\Theta|X,Y)\big) = \log\big(p(X,Y|\Theta)\big) \tag{5}$$

The EM algorithm is an iterative procedure consisting of 2 steps. The first step of EM algorithm, called Expectation step (E-step), estimates the expected value of the $\log\big(L(\Theta|Z)\big)$ with respect to the observed data X, the unknown data Y and the current parameter estimates $\hat{\Theta} = \Theta^{(i-1)}$. At the beginning, parameter estimates $\hat{\Theta} = \Theta^0$ can be extracted from some "a priori" information (e.g., search engines metadata and/or chemical properties) or randomly (increasing though the risk of estimate errors). The expectation $Q\big(\Theta, \Theta^{(i-1)}\big)$ is:

$$Q\big(\Theta, \Theta^{(i-1)}\big) = E\big[\log\big(L(\Theta|Z)\big)\big] = E\big[\log\left(p(X,Y|\Theta)\right)|X, \Theta^{(i-1)}\big] \tag{6}$$

where Θ is a normal variable we are adjusting, X and $\Theta^{(i-1)}$ are known, Y is a random variable related to the unobserved data and its distribution is the posterior probability of each GMM component with respect to each observation (i.e., ion):

$$p\big(Y\big|X, \Theta^{(i-1)}\big) \tag{7}$$

Notice that $\Theta^{(i-1)}$ are the parameters used to evaluate the expectation, whereas Θ are the parameters we are going to optimize in order to maximize the likelihood $L(\Theta|Z)$.

Indeed, the EM algorithm in a second step, called Maximization step (M-step), maximizes the expectation computed in the former step:

$$\Theta^i = \underset{\Theta}{\operatorname{argmax}}\ Q(\Theta, \Theta^{(i-1)}) \tag{8}$$

The E-step and M-step are iteratively repeated until a local maximum of the likelihood function is reached. For a GMM, the new estimates $\hat{\Theta}$ of the parameters based on the old estimates are as follows:

$$\hat{\alpha}_k = \frac{1}{n}\sum_{i=1}^{n} p(k|x_i, \Theta) \tag{9}$$

$$\hat{\mu}_k = \frac{\sum_{i=1}^{n} x_i p(k|x_i, \Theta)}{\sum_{i=1}^{n} p(k|x_i, \Theta)} \tag{10}$$

$$\widehat{\Sigma}_k = \frac{\sum_{i=1}^{n} p(k|x_i, \Theta)(x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T}{\sum_{i=1}^{n} p(k|x_i, \Theta)} \tag{11}$$

where k =1,…,N indicates the k^{th} Gaussian component and $p(k|x_i, \Theta)$ is the posterior probability of the k^{th} Gaussian component with respect to each ion. These update equations perform both E-step and M-step.

Since the local optimum of the likelihood function is strongly dependent on starting values, it is quite important to supply suitable EM starting parameters, which are Gaussians' centers (i.e., $\hat{\mu}_k$) and shapes (i.e., $\hat{\alpha}_k$ and $\widehat{\Sigma}_k$). These values are extracted by: the metadata stored in the peptide library, the theoretical isotopic distribution associated to the peptide under analysis and the peaks' position estimated in the main isotopic peak detection step.

1.2.3 Number of components/isotopes for the GMM

An important step for creating a good model is to choose a suitable number of components: too few would fail to model the data accurately; too many would lead to an over-fit model with singular covariance matrices. The user can set this number according to the expected number of isotopes and taking into account an additional component for the noise. Otherwise, 3DSpectra determines the most appropriate N by means of the Akaike Information Criterion (AIC) that minimizes the following term:

$$AIC = 2 \cdot nlogL + 2 \cdot m \tag{12}$$

where m is a penalty term, dependent to N, given by the number of estimated parameters, and $nlogL$ is the optimum negative log-likelihood for the data given the estimated parameters.

1.3 Implementation

In this section, some details about 3DSpectra's implementation are given.

1.3.1 Metadata retrieval for local peptide analysis

The peptide library is automatically generated by 3DSpectra starting from the metadata file path by means of the *library*(filePath) function. It works properly only if the metadata file follows a strictly defined schema, which is provided with the software itself. The peptide library variable is saved in a *.mat* file, which is loaded at the beginning of 3DSpectra execution.

1.3.2 Optimized data access via mzRTree

In order to allow efficient and flexible data accesses, 3DSpectra is provided together with a data access toolbox enabling to retrieve data by range queries.

By default, data access is performed by the mzRTree default range query method. mzRTree can be automatically created by the 3DSpectra built-in function *mzRTreeCreation*(path_mzXML, path_mzRTree), starting from the mzXML (or, mzML) file path. The mzRTree data structure is then stored in the path_mzRTree folder.

1.3.3 Main isotopic peak detection

The fit of the sum of Gaussians model on every elution profile along the temporal dimension is implemented by means of the *fit*(retTimes,ionCounts,libname) function from the Curve Fitting Toolbox. It fits the data in the column vectors retTimes and ionCounts using the library model specified by libname.

1.3.4 3D isotopic distribution model

The isotopic distribution shaped by the GMM is fitted to peptide data using the *gmdistribution.fit*(X,k) function from the Statistics Toolbox, which implements the Expectation Maximization (EM) algorithm. It outputs an object of the *gmdistribution* class containing maximum likelihood estimates of the parameters

of the Gaussian mixture model with k components for data in the n-by-d matrix X, where n is the number of observations and d is the dimension of the data.

In particular, the gmdistribution.fit assumes that a collection of samples from the mixture is observed rather than an aggregate representation of the samples, such as the histogram. Since the observed mixture is the LC-MS signal, it gives an aggregate representation of samples. Thus, we need to compute the collection of samples that generated it. Such operation is computationally very demanding under MATLAB and in order to optimize it, a C source file has been compiled and linked into a shared library called a binary MATLAB Executable (MEX) file.

The theoretical isotopic distribution parameters are computed making use of some MASPECTRAS built-in methods which have been embedded in the implementation in a Java executable library [5].

1.3.5 Recognition of the isotopic distribution borders

To recognize the GMM PDF iso-density curves we used the *pdf*(gmm,X) function of the *gmdistribution* class. It returns a vector y of length n containing the values of the PDF for the *gmdistribution* object gmm, evaluated at the n-by-d data matrix X, where n is the number of observations and d is the dimension of the data.

Data clustering was implemented by the *cluster*(gmm, X) function from the *gmdistribution* class: the method assigns a cluster to each observation in the n-by-d data matrix X, where n is the number of observations and d is the dimension of the data. Clusters are determined by the k components of the Gaussian mixture distribution defined by gmm. It returns a n-by-1 vector of indexes, idx, where idx(I) is the cluster index of observation I referring to the component of the GMM with the largest posterior probability, weighted by the component probability.

The probability of each ion count of belonging to the noise component is estimated employing the *posterior*(gmm,X) function from the *gmdistribution*. It returns P, the posterior probabilities of each of the k components in the Gaussian mixture distribution defined by gmm for each observation in the data matrix X. P is a n-by-k matrix, with P(I,J) being the probability of component J given observation I. X has n-by-d size, where n is the number of observations and d is the dimension of the data.

1.3.6 Processing And Ratio Computation

To implement the smoothing of spectra and chromatograms using the Savitzky and Golay method, we used the *mssgolay*(x, ionCounts) MATLAB function from the Bioinformatics Toolbox. It smoothes raw noisy signal data with peaks using least-squares polynomial. The x vector consists of separation-unit values. The ionCounts parameter is a vector of intensity values.

The theoretical isotopic distribution model is fitted on data by means of Weighted Linear Least Squares (WLLS), implemented in the *lscov*(A,b,w) MATLAB function. It computes a weighted least-squares (WLS) fit when provided with a vector of relative observation weights, w. It returns x, the weighted least squares solution to the linear system A*x = b, that is, x minimizes (b - A*x)'*diag(w)*(b - A*x) and here is a scalar. Matrix A is a vector made of the theoretical relative intensities in the isotopic distribution. The weights w are the probabilities of each isotopic peak.

The correlation reliability score, or weight, associated to each ratio is computed by the *corr2*(A, B). It computes the 2-D correlation coefficient between A and B, where A and B are the data matrices of the same size associated respectively to the peptide and its labeled partner.

Outlier removal is performed by a MASPECTRAS built-in method which have been embedded in the proposed implementation as a Java executable library.

In order to allow visual inspection a function, named *visualInspection.m*, for the automatic visualization of every pair (peptide, partner) has also been implemented.

Results are stored both in a MATLAB workspace variable and in an Excel file; regression lines of light to heavy abundances are printed to a postscript file.

3DSpectra can be compared to any other software, the results of which are stored in an Excel file compliant to the schema of MASPECTRAS results; the compared regression lines are also printed to a postscript file automatically.

Moreover MATLAB allows inspecting the *results* variable and their values through the built-in visual editor. The variable has a field for every relevant information related to the analyzed peptide: the peptide sequence, its charge, its index to retrieve additional metadata from the peptide library (e.g., its labeling status, elution time, etc.), the estimated quantification ratio, the abundance of both the peptide and its partner, the experimental replicate where the peptide was found, the value for the correlation to its labeling partner. If the quantification ratio has been computed starting from multiple peptide occurrences with different charges, all of them are reported in the charge field, and the corresponding library indexes appear in the index field.

1.4 Experimental Results

The graphs reproducing all the regression lines on common peptides before and after outlier removal are provided as supporting material in 2 separate files named respectively *regressions_commons_with_outliers.tiff* and *regressions_commons_without_outliers.tiff*. Please notice that before outlier removal ASAPRatio showed no linearity on the 1l:10h ratio (i.e., R^2 not significant at 5% level) and poor linearity in other ratios (i.e., R^2 smaller than 0.7): in such cases the regression line doesn't model properly the broad cloud of underlying data and regression parameter estimates are not reliable, therefore it was not fitted. In addition, the graphs reproducing the regression lines on all peptide abundance estimates before and after outlier removal are provided as well in other 2 files named respectively *regressions_all_with_outliers.tiff* and *regressions_all_without_outliers.tiff*. As pointed out in the main text, regression lines are statistically comparable only for the set of commonly quantified peptides.

Table 3 reports the performance of both methods before and after outlier removal and its values are summarized in Table 2 in the main text.

Table 1 3DSpectra and ASAPRatio columns are respectively 3D and 2D labeled, while l and h stand for light and heavy, respectively. "Estimated pep ratios" is the total number of quantified peptide ratios across experimental replicates. "Efficiency gain" is the gain provided by 3DSpectra. It also reports quantification accuracy and precision in terms of mean, standard deviation (SD) and coefficient of variation (CV) of the ratios. All parameters are estimated before outlier removal in the upper panel (Outliers included) and after outlier removal in the lower panel (Outliers removed).

Efficiency	**1l:2h**		**2l:1h**		**1l:5h**		**5l:1h**		**1l:10h**		**10l:1h**		**1l:1h**	
	3D	**2D**	**3D**	**2D**	**3D**	**2D**	**3D**	**2D**	**3D**	**2D**	**3D**	**2D**	**3D**	**2D**
Outliers included														
Estimated pep ratios	**58**	**54**	**59**	**54**	**53**	**53**	**52**	**44**	**52**	**38**	**36**	**30**	**121**	**121**
Efficiency Gain	**7%**		**9%**		**0%**		**18%**		**37%**		**20%**		**0%**	
Accuracy - Mean ratio	0.62	0.62	2.11	2.73	0.31	1.25	4.77	4.55	0.29	1.86	9.84	9.13	1.29	1.59
Precision - SD	0.29	0.36	1.85	5.03	0.17	4.88	1.58	3.72	0.49	4.48	4.90	4.53	1.03	3.05
Precision - CV	46%	59%	88%	184%	55%	392%	33%	82%	171%	241%	50%	50%	79%	191%
Outliers removed														
Estimated pep ratios	**48**	**39**	**47**	**38**	**48**	**38**	**51**	**34**	**40**	**23**	**32**	**23**	**94**	**85**
Efficiency gain	**23%**		**24%**		**26%**		**50%**		**74%**		**39%**		**11%**	
Accuracy - Mean ratio	0.57	0.54	1.88	1.99	0.26	0.27	4.82	4.23	0.15	0.13	9.88	9.08	1.07	1.06
Precision - SD	0.14	0.14	0.57	0.66	0.09	0.09	1.55	1.18	0.06	0.05	3.52	3.37	0.32	0.27
Precision - CV	25%	26%	30%	33%	35%	34%	32%	28%	39%	37%	36%	37%	30%	26%

References

[1] McLachlan, G., Peel, D., *Finite Mixture Models*, vol. 44, Wiley-Interscience, 2000.

[2] Jaffe, J.D., Mani, D.R., Leptos, K.C., Church, G.M., et al., PEPPeR, a platform for experimental proteomic pattern recognition. *Mol. Cell. Proteomics* 2006, 5, 1927–41.

[3] Bilmes, J.A., A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. *Int. Comput. Sci. Inst.* 1998, 4, 15.

[4] Dempster, A.P., Laird, N.M., Rubin, D.B., Maximum Likelihood from Incomplete Data Via Em Algorithm. *J. R. Stat. Soc. Ser. BMethodological* 1977, 39, 1–38.

[5] Hartler, J., Trötzmüller, M., Chitraju, C., Spener, F., et al., Lipid Data Analyzer: unattended identification and quantitation of lipids in LC-MS data. *Bioinformatics* 2011, 27, 572–7.