MaSDA: A System for Analyzing Mass Spectrometry Data

Francesco Gullo [a], Giovanni Ponti [a], Andrea Tagarelli [a,*], Giuseppe Tradigo [b], Pierangelo Veltri [b]

[a] *Dept. of Electronics, Computer and System Sciences (DEIS), University of Calabria, Via P.Bucci 41c, Rende (CS) I87036, Italy*

[b] *Dept. of Experimental and Clinical Medicine, University Magna Græcia of Catanzaro, Viale Europa, Germaneto (CZ) I88100, Italy*

Abstract

Mass spectrometry (MS) approaches have been recently coupled with advanced data analysis techniques in order to enable clinicians to discover useful knowledge from MS data. However, effectively and efficiently handling and analyzing MS data requires to take into account a number of issues. In particular, the huge dimensionality and the variety of noisy factors present in MS data require careful preprocessing and modeling phases in order to make them amenable to the further analysis.

In this paper we present *MaSDA*, a system performing advanced analysis on MS data. MaSDA has the following main features: *i)* it implements an approach of MS data representation that exploits a model based on low-dimensional, dense *time series*; *ii)* it provides a wide set of *MS preprocessing* operations which are accomplished by means of a user-friendly graphical tool; *iii)* it embeds a number of tools implementing various tasks of *data mining and knowledge discovery*, in order to assist the user in taking critical clinical decisions. Our system has been experimentally tested on several publicly available datasets, showing effectiveness and efficiency in supporting advanced analysis of MS data.

Key words: mass spectra modeling, mass spectra preprocessing, time series, data mining

* Corresponding author. ph.: +39 0984 494751; fax: +39 0984 494713.
Email addresses: fgullo@deis.unical.it (Francesco Gullo), gponti@deis.unical.it (Giovanni Ponti), tagarelli@deis.unical.it (Andrea Tagarelli), gtradigo@si.deis.unical.it (Giuseppe Tradigo), veltri@unicz.it (Pierangelo Veltri).

1 Introduction

Mass Spectrometry (MS) is a powerful analytical technique aimed to extract interesting biological information from tissue or serum samples [1,2]. Due to its ability in separating ions of different masses from a sample, a mass spectrometer generates a vector of measurements representing the number of ions that hit the spectrometer detector during small, fixed intervals of time. A mass spectrum is hence represented as a plot of ion abundance (*intensity*) versus the mass-to-charge ratio (m/z). By analyzing mass spectra it is possible to identify macromolecules contained in the original compounds by associating (portions of) proteins to their peak expressions in a spectrum.

Recently, there has been a lot of research concerning advanced analysis on MS data in order to extract significant, previously unknown information or "knowledge" from such data. This usually involves various tasks aimed to identify biological patterns and organize them at different degrees of automation. Recently, several approaches to MS data management and mining have been developed. For instance, in [3], data mining techniques have been used to identify discriminants in a female population, distinguishing ovarian cancers from healthy conditions. Similarly, data mining techniques have been applied in [4] for surface-enhanced laser desorption/ionization mass spectrometry (SELDI MS) data to identify discriminants in rectal cancer diseases. In [5], machine learning algorithms have been used to identify biomarkers in SELDI MS data generated on tens of patients to figure out cerebral accident discriminants.

MS data preprocessing has been recognized as a mandatory phase in mass spectra data analysis. The need for preprocessing mass spectra arises since the data obtained from a mass spectrometer *i)* have very large dimensionality and *ii)* are naturally corrupted by various noisy factors. Several research studies have been proposed on the development of preprocessing steps for MS data (e.g., [6,7]), and in some cases they have focused on specific steps, such as baseline subtraction [8–10], peak identification [11,12], and peak alignment [13,14,9]. Also, there has been recently a growing interest for developing MS data preprocessing software systems that are able to fulfill certain requirements, such as filtering data and highlighting relevant spectra portions w.r.t. non-relevant ones (e.g., noise), and allowing the user to perform the various preprocessing stages iteratively and interactively.

1.1 Main contributions

In this paper we present *MaSDA – Mass Spectrometry Data Analysis*, a system for advanced analysis of MS data. The general objective of MaSDA is to

assist the user in discovering useful knowledge from MS data. The discovered patterns of knowledge might eventually support the user (e.g., the clinician) to take critical decisions; for instance, if interesting relationships on certain biological conditions referring to a given disease have been found out by analyzing MS data, then one might use this new information to design new therapies.

The key idea underlying our approach to MS data analysis, which is implemented in the MaSDA system, is to exploit the temporal information implicitly contained in mass spectra and model such data as compact *time series*. The proposed MS data representation model is aimed to take some advantages with respect to the traditional count-vector-based approaches to MS data representation, in particular:

- The problem of high dimensionality in MS data is addressed by identifying variable-length segments in the time series representing mass spectra. Each one of these segments is conceived to be comprised of locally tight points, and is finally mapped to a synthetic information. This enables to drastically reduce the number of noisy dimensions while preserving relevant features (i.e., trends in the series profile).
- The critical task of preprocessing MS data is relatively simplified by employing major existing techniques for similarity detection in time series, which allow for dealing with mass spectra in a way more robust to noise and suited to different profiles of the spectra.

Another important aspect of our MS data analysis system is that it offers a graphical tool for preprocessing the raw mass spectra, with the following main features:

- Wide set of supported preprocessing operations – it is designed to cover most of the MS data preprocessing steps that have been recognized as relevant in the literature;
- Efficiency – it guarantees high performance in MS preprocessing, by adopting fast algorithms for each step. This allows for efficiently dealing with high dimensional data;
- Support for user interaction – it enables the user to monitor and control the whole preprocessing task; in particular, the user can choose which preprocessing steps have to be performed and their execution order, and she/he can properly set the parameters involved into each step;
- Ease-to-use – it provides a user-friendly graphical interface and a simple wizard which guides the user in each preprocessing step;
- Web-based access – it makes use of the Java™ Web Start technology, [1] which allows for launching the tool directly from the Web.

Besides the functionalities of MS preprocessing and time series based MS

[1] http://java.sun.com/products/javawebstart/

modeling, our MaSDA system is designed to perform various tasks of MS data analysis, by employing *data mining and knowledge discovery* techniques, and to evaluate and visualize the patterns of knowledge discovered from the input MS data. As experimentally proved on publicly available datasets, our system has been shown to be a valid support for the user interested in effectively and efficiently analyzing MS data.

The rest of the paper is organized as follows. Section 2 introduces our system conceptually. Section 3 focuses on the MS data preprocessing module and describes how the involved preprocessing operations are implemented in our system. Section 4 describes a time series based modeling of MS spectra which is designed to represent such data into a form particularly convenient for the further analysis. Section 5 discusses the capabilities of the system for some tasks of MS data analysis. Finally, Section 6 concludes the paper.

2 Conceptual architecture of the MaSDA system

The MaSDA system consists of five main modules, which are sketched in Figure 1 and described below:

(1) MS Data Preprocessing: it performs one or more preprocessing steps on the raw spectra in order to make them amenable to the further analysis stage. In particular, this module includes at least the following preprocessing operations: range cut, peak smoothing, detection of valid peaks, baseline correction, quantization, and normalization.
(2) Time Series based MS Data Modeling: this module transforms the preprocessed MS data into time series, using a model conceived to maintain the significant trends (peak profiles) while reducing the data dimensions.
(3) MS Data Analysis: it includes a number of submodules each performing a certain task of knowledge discovery, such as cluster analysis, frequent pattern discovery, data summarization, and so on. The input for this module is the preprocessed spectra, which is of the form either original (output of module 1) or based on time series (output of module 2).
(4) Pattern Evaluation: it is in charge of assessing the validity of the discovered knowledge patterns.
(5) Knowledge Presentation: this module finally presents the discovered knowledge by using visualization tools.

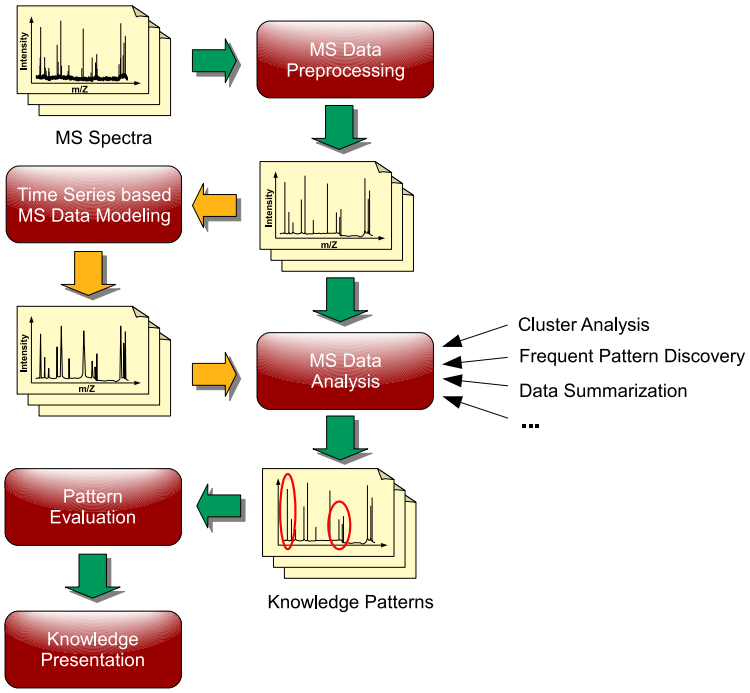


Fig. 1. The overall conceptual architecture of the MaSDA system

3 Pre-analysis processing of MS data

A raw spectrum generated by a mass spectrometer is substantially a combination of three components: the true signal, a baseline signal, and noise [6]; in particular, the true signal contains biological information, whereas the base intensity level (baseline) varies from point to point across the m/z axis, so that intensity values that are under the baseline represent ground noise and should be hence filtered out. Separating and reconstructing such individual components from a raw spectrum is a hard task, since their analytical forms are not known. Thus, spectra usually need to be subject to one or more preprocessing operations, in order to make them amenable to further analysis phases.

Since the variety of spectrometry platforms, experimental conditions and clinical studies, there exists a number of preprocessing operations (see, e.g., [6,7]). While there has not been shared agreement about a preprocessing scheme, a reasonable list of preprocessing steps on mass spectra can be given as follows:

- *calibration*, which is used to map the observed time of flight into the inferred mass-to-charge ratio;
- *filtering or denoising*, which aims to reduce random noise generated by electronic or chemical causes;

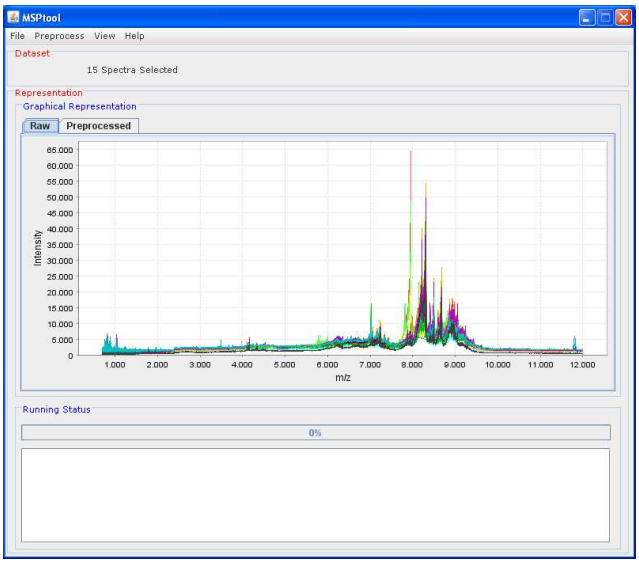


Fig. 2. A sample screenshot of the MaSDA tool for MS preprocessing

- *baseline correction*, which is in charge of recognizing and filtering out the baseline signal of mass spectra;
- *normalization*, which makes peak intensities understandable over a uniform range;
- *peak detection*, which is in charge of locating specific proteins or peptides on the identified locations on the m/z axis and typically involves an assessment of the spectra local maxima and their signal-to-noise ratio (S/N);
- *peak quantification*, which represents each detected peak by means of a concise information (e.g., peak heights or areas);
- *peak matching/alignment*, which aims to recognize the peaks in different samples that correspond to the same biological molecule.

In this section we describe the capabilities of the MaSDA module for MS preprocessing, we called *MSPtool*. MSPtool is a Java™ based tool that implements most of the MS preprocessing operations discussed above (Figure 2). This tool offers its features visually in order to assist the user in performing an MS preprocessing task, i.e., observing the raw spectra, selecting an appropriate sequence of preprocessing steps, and choosing the parameter setting for each of the selected preprocessing steps.

MSPtool is able to deal with various formats storing the raw spectrum/spectra to be preprocessed, including plain-text files, comma separated values files (CSV), and XML data. Also, the tool allows the user to graphically represent preprocessed spectra, which is useful to visually explore (and compare) the spectra profiles before and after the preprocessing step. Figure 3 shows a screenshot of the last step of the preprocessing wizard, which reports a summary of the preprocessing setting; in this step, it is also possible to change the order of the selected preprocessing operations.

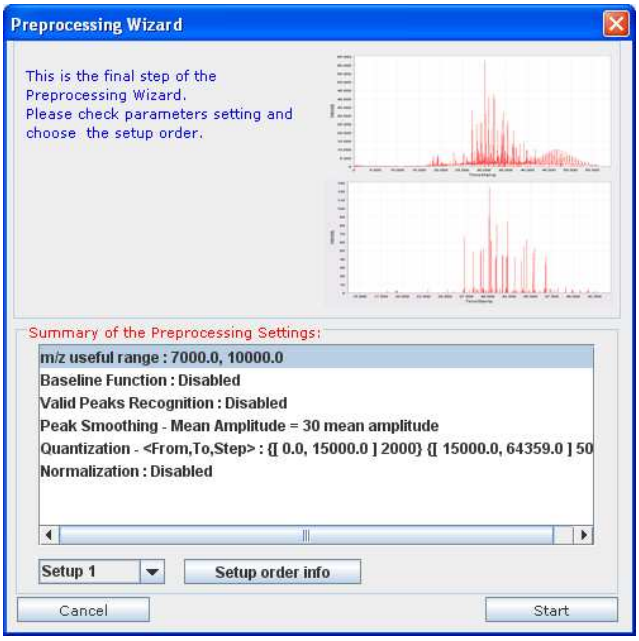


Fig. 3. Preprocessing wizard - summary of the preprocessing settings

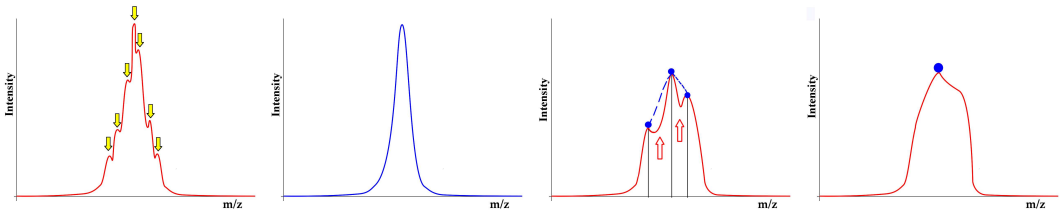
It should be noted that, although MSPtool has been originally designed as a standalone application, we have also provided a Web-based version using the Java™ Web Start technology. This feature of the tool is mainly motivated by our intention to make MSPtool publicly available and to simplify the processes of deployment and upgrade of the tool. [2] In the following, we describe the main steps of MS data preprocessing involved into MSPtool.

Range cut. This step performs a cut of the *m/z* range of the spectra, in order to filter out those portions of spectra that do not contain relevant biological information.

Peak smoothing. Peak smoothing falls into the category of peak detection/quantification operations. This step aims to smooth the peak profiles in the spectra and to reconstruct the theoretical Gaussian profile of the peaks. An ideal peak profile is comprised of two parts: a monotonic ascending side and a monotonic descending side. We call *M-peak* a spectrum peak having its intensity higher than both the previous and the next point, i.e., a local maximum in the spectrum (Figure 4 (a)–(b)).

The peak smoothing algorithm has a parameter w_p, *peak amplitude*, which is a function of the mass spectrometer resolution. This parameter can be initially set to the average width of peaks in the spectrum, or modified according to

[2] A beta version of the MS data preprocessing tool is available at the following Web address: http://polifemo.deis.unical.it/~gtradigo/jnlp/msptool/



(a) (b) (c) (d)

Fig. 4. Peak smoothing: (a) example M-peaks and (b) the corresponding ideal peak; (c) three local M-peaks and (d) the resulting profile after smoothing

the data features. Basically, the algorithm works as follows: first, it detects all the M-peaks in the spectrum; each M-peak (except the last one) is compared with the next M-peak. If the distance between these two M-peaks is lower than $w_p/2$ then either a descending phase or an ascending phase can occur, and the spectrum is modified such that the resulting peak has the expected pseudo-Gaussian shape for both the ascending and the descending sides (Figure 4 (c)–(d)).

Valid peaks recognition. Valid peaks recognition is a further step of peak detection/quantification. This step aims to recognize as valid peaks the local maxima into a mass spectrum that satisfy specific requirements. In particular, the algorithm for valid peaks recognition implemented into MSPtool takes into account the signal-to-noise ratio (S/N) and works as follows: for each spectrum, the S/N at each local maximum of the spectrum is computed as the ratio of the intensity at the maximum to the local noise estimate; then, only the local maxima having S/N greater than a user-defined threshold (*multiplicative factor*) are recognized as valid peaks. The non-valid peaks in a spectrum are discarded from the further analysis.

Baseline correction. This step aims to identify the baseline signal in the spectra and filter out all spectra intensity values below the baseline. The user can choose a function that approximates the baseline (i.e., the baseline function) and setting the parameters for each function. MSPtool offers the following baseline functions: *linear function*, *logarithmic function*, *exponential function* and *piecewise linear function*. The first three functions approximate the baseline as a linear, logarithmic and exponential function, respectively, whereas the definition of the piecewise linear function is as follows. The m/z range of each spectrum is divided into a user-defined number of equally-sized windows. The final piecewise linear function is composed by a number of linear functions, each of them properly defined according to the associated window. For each window, the corresponding linear function is computed by solving a line fitting problem to the local minima in the window.

Quantization. This step performs a quantization of the spectra, i.e, a discretization of the original intensity values according to specific quantization levels. A non-uniform quantization model is used in the MPStool in such a way that two or more ranges in the intensity axis are identified and subject to different fine-grained quantization.

Normalization. Spectra normalization changes spectra shapes by transforming original intensity values into new ones proportionally calculated according to a certain fixed range. MSPtool implements various normalization techniques, including z-normalization and min-max normalization. The former subtracts the mean over all the spectra intensities from each intensity value and then divides this difference by the standard deviation over all the spectra intensities; the latter scales the intensity values such that, for each m/z and over all the spectra, the smallest intensity value becomes zero and the largest intensity becomes one.

4 Time series based modeling of MS data

A (preprocessed) mass spectrum is a sequence of paired values $S = [((m/z)_1, I_1), \ldots, ((m/z)_n, I_n)]$, where each pair is comprised of a mass-to-charge-ratio value and the associated intensity value. A mass spectrum so defined can be trivially modeled as a time series $T = [(x_1, t_1), \ldots, (x_n, t_n)]$ whose x_i correspond to the spectrum intensity values I_i, and the time steps t_i correspond to the values $(m/z)_i$. Indeed, the notion of time implicitly lies in the sequence of mass-to-charge values.

Derivative time series Segment Approximation (DSA). Time series representing mass spectra are typically high dimensional data. Thus, it is desirable to model such time series into a compact representation that synthesizes the significant variations in the time series profile.

For this purpose, we exploit a representation scheme called *DSA (Derivative time series Segment Approximation)*, which has been proposed in [15,16]. Using the DSA model, a time series is transformed into a new, smaller sequence. The key idea is to approximate a series by producing a list of segments that follow the main trends in the series. Each segment summarizes a portion of the series that contains points having the same profile. Once the list of segments has been generated, each of these segments is finally represented by a pair of numerical values, namely the slope of the line containing the segment and the timestamp of the last point in the segment.

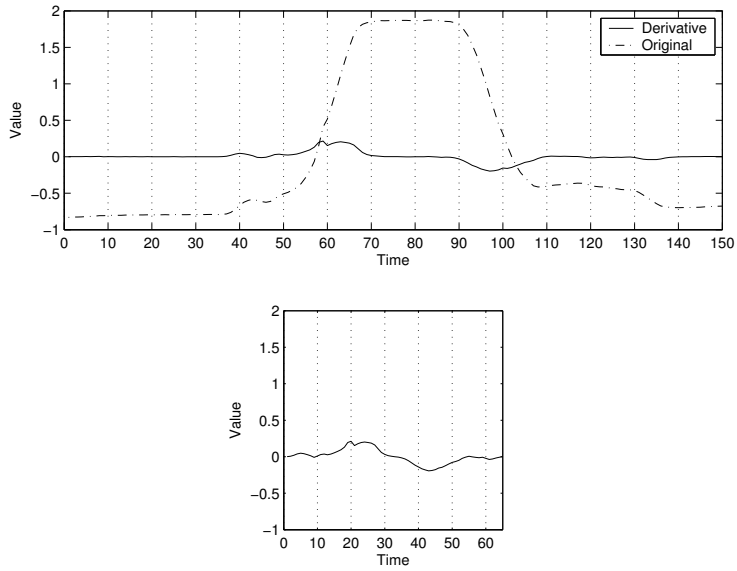


Fig. 5. Illustration of DSA. (top) A time series and its derivative version. (bottom) The final DSA sequence computed by segment approximation of the derivative time series

It should be emphasized that the number p of segments produced by DSA is usually much smaller than the number of original points in the series (i.e., $p \ll n$). This enables a significant increment of performance efficiency for the various data analysis algorithms that will be performed on DSA representations of time series.

Figure 5 shows the application of DSA on a sample time series. We can observe the ability of DSA in representing a typical time series with a new sequence reflecting the main trends of the original one. Also, it can be noted that the number of segments produced is very small and a good compression level has been reached. In the following we give a brief, informal description of DSA, whereas the interested reader can find further details in [15,16].

Given any time series corresponding to a mass spectrum, a DSA representation is accomplished in three main steps:

(1) *Derivation*, which computes the first derivatives of the points in the original series;
(2) *Segmentation*, which identifies a relatively small number of segments;
(3) *Segment approximation*, which finally obtains a lower dimensional still fine-grained representation of the original series.

The *derivation* step transforms an original series $T = x_1, \ldots, x_n$ into a new one $\dot{T} = [\dot{x}_1, \ldots, \dot{x}_n]$ containing the first derivative estimates of all the points in T. The objective of this step is to capture the main trends of the raw series, which enables an accurate segmentation of the series. To compute $\dot{T}$,

we use an estimation model that is sufficiently general (i.e., independent on the underlying data distribution model) and still enough robust to outliers [15,16].

The *segmentation* step applies to the derivative series $\dot{T}$ in order to identify contiguous subsequences (segments) in $\dot{T}$. Each of these segments is designed to aggregate subsequent data points having very close derivatives. To derive a compact, feature-rich representation of the original time series, we adopt a sliding-window scheme, which is able to identify variable-length segments and works as follows:

(1) It starts from the first point in $\dot{T}$ and proceeds by scanning all the points in the series.
(2) Given a sequence s_i of contiguous points identified at a certain iteration, the next point in the series is recognized as belonging to s_i if and only if the difference between its numerical value and the mean value of all the points within s_i does not exceed a certain threshold; otherwise, if such a difference is greater than the threshold, the subsequence s_i is identified as a segment, and the process repeats starting from the next point not yet considered.

The segmentation threshold required by the sliding-window algorithm is estimated globally with respect to a given time series, i.e., by considering the variance over the points in the derivative series.

In the final step of DSA, each of the detected segments is *approximated* with a synthetic information, which is a pair formed by the timestamp of the last point within the segment and an angle that explains the average slope of the portion of time series bounded by the segment.

5 Using the MaSDA system for organizing MS data

A major task of MS knowledge discovery consists in classifying spectra in order to discriminate them on the basis of their biological information (e.g., healthy or diseased individuals). To cope with huge dimensionality and frequently occurring noise in MS data, this task requires careful preprocessing and modeling of the data. However, the organization task is particularly difficult when a-priori knowledge on the predefined set of categories or a training set of positive/negative examples from data is poorly or not available at all. In this case, the goal is to infer an organization of a collection of MS data into meaningful groups (*clusters*), based on interesting relationships discovered in the data. *Clustering* of MS data finds natural application to many real MS scenarios, since the various pathologic states from clinical studies might require to be discovered in an unsupervised way. In the following we describe

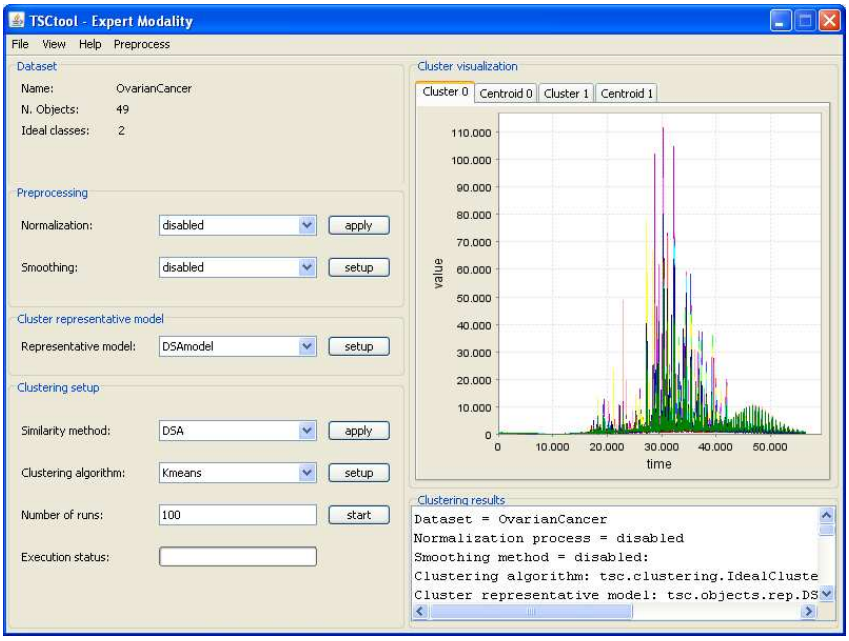
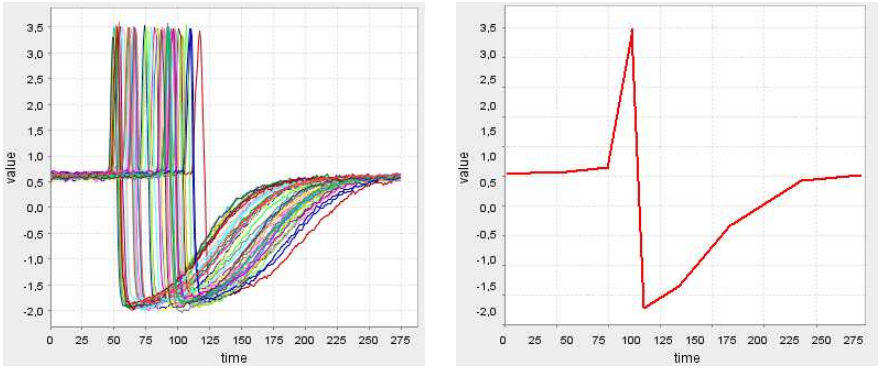


Fig. 6. A screenshot of the MaSDA tool for clustering MS data

how MaSDA can be used to organize MS data by a task of cluster analysis.

Figure 6 shows a screenshot of a Java™ based tool embedded in MaSDA for clustering MS data (available from the OvarianCancer dataset [3]). On the left of this figure, we can observe a number of component panels devoted to the configuration of a clustering experiment, which involves the choice of the preprocessing (smoothing) function, the model of cluster representative, the method of representation and similarity between series, and the algorithm of clustering. On the right of the figure, each of the output clusters and relating representative can be explored using different choices of visualization; the clustering results can be also saved into a file for further reloading. Also, we can observe in the menu bar the presence of a command for launching the preprocessing tool (MSPtool) previously described. The interested reader is referred to [17,15,16] for a detailed description of the smoothing functions, the similarity methods, the clustering algorithms and the evaluation criteria implemented in the clustering tool and used in experiments.

Another important task allowed by MaSDA is *data summarization*. Given a set of MS time series, the objective here is to generate a summary, or prototype sequence that is able to capture the most relevant features of the series in the given set. Since the input series may have different length and scales, the task is not trivial (i.e., we cannot directly resort to the computation of an "average" time series); rather, a concise representation is desirable to include the significant trends in the set as well as to filter out irrelevant information [15,16]. Figure 7 shows an example of summarization of a certain set of time series data.



(a) (b)

Fig. 7. An example of data summarization: (a) a set of time series and (b) the computed prototype

The MaSDA system has been tested on various real MALDI/SELDI-TOF MS data obtained using different clinical studies under different mass spectrometry platforms and experimental conditions; in particular, some of the used collections are publicly available from authoritative sources (e.g., the NCI's Center for Cancer Research, [3] other ones have been provided by the proteomics laboratory at the University of Catanzaro. [4]

A major goal of the experimentation conducted on MS data by using MaSDA was to identify groups of subjects that show similar characteristics according to the expected pathological states (e.g., in the Prostate dataset [18] different cancer or benign conditions at various levels of PSA). Moreover, in this context a challenge is represented by the discovery of the proteomic profiles that distinguish disease-related or cancer conditions from the healthy ones. For instance, some discriminatory patters might be found out around early m/z values, other ones might be detected according to sequences of peaks at a certain intensity level. Intuitively, this issue can be more easily addressed by exploiting our time-series-based modeling of MS data: indeed, the compact representation that is substantially comprised of relevant features in the data (while discarding various noisy factors) favors the identification of significant patterns in the spectra.

6 Conclusion

We presented the MaSDA system for advanced data analysis and knowledge discovery from MS data. This system features a number of graphical tools that enable the user to preprocess and model the mass spectra, to perform data

[3] http://home.ccr.cancer.gov/ncifdaproteomics/ppatterns.asp
[4] http://proteomics.unicz.it

mining tasks, and to evaluate and visualize the discovered knowledge patterns. In particular, MaSDA implements an approach to MS data representation that exploits a suitable model based on low-dimensional, dense time series. Using this MS data representation coupled with different available choices of preprocessing settings, MS data can be effectively and efficiently managed and analyzed by employing data mining and knowledge discovery methods, including cluster analysis and classification, frequent pattern extraction, data summarization. The usefulness of MaSDA has been experimentally demonstrated in clinical applications, such as the unsupervised learning (clustering) of disease-related conditions for early detection of cancers.

References

[1] G.L. Glish, and R.W. Vachet, The basics of mass spectrometry in the twenty-first century, Nature Reviews 2 (2003) 140–150.

[2] R. Aebersold, M. Mann, Mass spectrometry-based proteomics, Nature 422 (2003) 198–207.

[3] E.F. Petricoin 3rd, A.M. Ardekani, B.A. Hitt, P. Levine, V.A. Fusaro, and S. Steinberg, G.B. Mills, C. Simone, D.A. Fishman, E.C. Kohn, and L.A. Liotta, Use of proteomic patterns in serum to identify ovarian cancer, Lancet 359 (2002) 572–577.

[4] F.M. Smith, W.M. Gallagher, E. Fox, R.B. Stephens, E. Rexhepaj, E.F. Petricoin 3rd, L. Liotta, M.J. Kennedy, and J.V. Reynolds, Combination of SELDI-TOF-MS and data mining provides early-stage response prediction for rectal tumors undergoing multimodal neoadjuvant therapy, Annals of Surgery 2(245) (2007) 259–266.

[5] J. Prados, A. Kalousis, J.C. Sanchez, L. Allard, O. Carrette, and M. Hilario, Mining mass spectra for diagnosis and biomarker discovery of cerebral accidents, Proteomics 4(6) (2004) 2320–2332.

[6] K.R. Coombes, K.A. Baggerly, and J.S. Morris, Pre-Processing Mass Spectrometry Data, Fundamentals of Data Mining in Genomics and Proteomics, Kluwer, Boston, 2007.

[7] M. Wagner, D. Naik, and A. Pothen, Protocols for Disease Classification from Mass Spectrometry Data, Proteomics 3(9) (2003) 1692–1698.

[8] B. Williams, S. Cornett, B.M. Dawant, A. Crecelius, B. Bodenheimer, and R. Caprioli, An algorithm for baseline correction of MALDI mass spectra, in: Proc. ACM Southeast Regional Conf., 2005, pp. 137–142.

[9] A.C. Sauve and T.P. Speed, Normalization, Baseline Correction and Alignment of High-Throughput Mass Spectrometry Data, in: Proc. Genomic Signal Processing and Statistics Conference, 2004.

[10] A.F. Ruckstuhl, M.P. Jacobson, R.W. Field, and J.A. Dodd, Baseline subtraction using robust local regression estimation, Journal of Quantitative Spectroscopy and Radiative Transfer 68 (1999) 179–193.

[11] W.E. Wallace, A.J. Kearsley, and C.M. Guttman, An Automated Peak Identification/Calibration Procedure for High-Dimensional Protein Measures From Mass Spectrometers, Analytical Chemistry 76(9) (2004) 2446–2452.

[12] Y. Yasui, D. McLerran, B.L. Adam, M. Winget, M. Thornquist, and Z. Feng, An Operator-Independent Approach to Mass Spectral Peak Identification and Integration, Journal of Biomedicine and Biotechnology 4 (2003) 242–248.

[13] J.W.H. Wong, G. Cagney, and H.M. Cartwright, SpecAlign - processing and alignment of mass spectra datasets, Bioinformatics 21(9) (2005) 2088–2090.

[14] N.O. Jeffries, Algorithms for alignment of mass spectrometry proteomic data, Bioinformatics 21(14) (2005) 3066–3073.

[15] F. Gullo, G. Ponti, A. Tagarelli, and S. Greco, Accurate and Fast Similarity Detection in Time Series, in: Proc. 15th Italian Symposium on Advanced Database Systems (SEBD), 2007, pp. 172–183.

[16] S. Greco, M. Ruffolo, and A. Tagarelli, Effective and efficient similarity search in time series, in: Proc. ACM Int. Conf. on Information and Knowledge Management (CIKM), 2006, pp. 808–809.

[17] F. Gullo, G. Ponti, A. Tagarelli, G. Tradigo, P. Veltri, A Time Series Based Approach for Classifying Mass Spectrometry Data, in: Proc. Computer-Based Medical Systems (CBMS), 2007, pp. 412–420.

[18] E.F. Petricoin 3rd, D.K. Ornstein, C.P. Paweletz, A. Ardekani, P.S. Hackett, B.A. Hitt, A. Velassco, C. Trucco, L. Wiegand, K. Wood, C.B. Simone, P.J. Levine, W.M. Linehan, M.R. Emmert-Buck, S.M. Steinberg, E.C. Kohn, and L.A. Liotta, Serum Proteomic Patterns for Detection of Prostate Cancer, Journal of the National Cancer Institute 94(20) (2002) 1576–1578.