Title

Supervised semi-automated data analysis software for gas chromatography / differential mobility spectrometry (GC/DMS) metabolomics applications

Permalink
https://escholarship.org/uc/item/3517p6pg

Journal
International Journal for Ion Mobility Spectrometry, 19(2-3)

ISSN
1435-6163

Authors
Peirano, Daniel J
Pasamontes, Alberto
Davis, Cristina E

Publication Date
2016-09-01

DOI
10.1007/s12127-016-0200-9

Peer reviewed

Published in final edited form as:
Int J Ion Mobil Spectrom. 2016 September ; 19(2): 155–166. doi:10.1007/s12127-016-0200-9.

Supervised Semi-Automated Data Analysis Software for Gas Chromatography / Differential Mobility Spectrometry (GC/DMS) Metabolomics Applications

Daniel J. Peirano, **Alberto Pasamontes**, and **Cristina E. Davis***
Mechanical and Aerospace Engineering, University of California, Davis, One Shields Avenue, Davis CA, 95616

Abstract

Modern differential mobility spectrometers (DMS) produce complex and multi-dimensional data streams that allow for near-real-time or post-hoc chemical detection for a variety of applications. An active area of interest for this technology is metabolite monitoring for biological applications, and these data sets regularly have unique technical and data analysis end user requirements. While there are initial publications on how investigators have individually processed and analyzed their DMS metabolomic data, there are no user-ready commercial or open source software packages that are easily used for this purpose. We have created custom software uniquely suited to analyze gas chromatograph / differential mobility spectrometry (GC/DMS) data from biological sources. Here we explain the implementation of the software, describe the user features that are available, and provide an example of how this software functions using a previously-published data set. The software is compatible with many commercial or home-made DMS systems. Because the software is versatile, it can also potentially be used for other similarly structured data sets, such as GC/GC and other IMS modalities.

Keywords

differential mobility spectrometry (DMS); field asymmetric ion mobility spectrometry (FAIMS); principal component analysis (PCA); partial least squares regression (PLS); data analysis; software

Introduction

Mobile chemical sensors have a large role to play in varying application areas that range from national defense, to human clinical diagnostics and precision agriculture. While a variety of high precision sensors exist for these tasks, the technology termed Field Asymmetric Ion Mobility Spectrometry (FAIMS) or Differential Mobility Spectrometry (DMS) has gained visibility over the last decade due to its robust performance (Cumeras et al. 2015a; Cumeras et al. 2015b). The tool has been commercially produced by several industries, and has operating principles that allow for mobile field use and point detection, or as a discovery tool in a research laboratory. While DMS hardware advances have been

*correspondence: cedavis@ucdavis.edu, TEL +1-530-754-9004, FAX +1-530-752-4158 .

widely explored, there have been fewer reports of data analysis breakthroughs that are applicable to diverse new data sets on multiple homemade or commercial instrument platforms (e.g. vendor agnostic data analysis approaches). Currently, there is no stand-alone software package that gives users the opportunity to easily visualize their data and pursue step-wise rigorous data analysis procedures to interpret their DMS results, especially as it relates to biological systems that are being monitored.

Many reports of DMS data analysis have focused on single chemical identification (Davis et al. 2010; Krylov et al. 2010; Manard et al. 2008) for specific applications, where users seek to determine if a target compound is present within a complex sample type, such as an air sample. Generally the resulting data is relatively easy to visualize and interpret, and only modest data processing methods may be needed. Noisy signals may be filtered to detect the target chemical peak, or additional hardware configurations may be used to couple DMS with orthogonal tools such as mass spectrometry (Manard et al. 2010), or gas chromatography (GC) to look for ion collision products that aid chemical identification (Kendler et al. 2007). But generally in these data sets, we seek to determine relatively simple information from the data stream. Occasionally, we also may be interested in detecting groups of several chemicals from a complex mixtures, but still only identifying a handful of major chemical features which is a small finite number (Camara et al. 2013; Lu and Harrington 2007; Rearden et al. 2007). In some examples, signal alignment has been used (Krebs et al. 2006a), along with feature selection procedures (Fong et al. 2011; Zhao et al. 2009) to assist in chemical identification. DMS data can also be augmented by mass spectrometry to confirm chemical identity (Lu et al. 2009), and in general these approaches are still numerically relatively straight forward to interpret. It is significantly more challenging to try and categorize complex DMS chemical data obtained from biological sources and mixtures, which themselves produce natural variation between samples.

Two major instrument configurations have been explored in connection with DMS for biological applications: pyrolysis (Py), and gas chromatography (GC). Pyrolysis helps to provide biomarker liberation from bacteria/endospores (Cheung et al. 2009; Eiceman et al. 2006; Krebs et al. 2006b; Prasad et al. 2008), which can then be separated and detected by DMS. Gas chromatography allows for a modest pre-separation of a very complex mixture before additional DMS separation and detection (Aksenov et al. 2014; Arasaradnam et al. 2014a; Arasaradnam et al. 2014b; Basanta et al. 2010; Covington et al. 2013; Rutolo et al. 2014; Schivo et al. 2013). The data sets that result are usually extremely complex and multi-dimensional, with some data features being very abundant but perhaps unimportant, and other data features having small abundance but high significance. It is often a combination of these low and high abundance chemical signals that help classify new incoming data, and both supervised and unsupervised learning algorithms have been used for untargeted biomarker discovery in biological applications. In addition, biological data can frequently include spurious chemicals that are not associated purely with one category of data over another. Thus, a common approach is to examine the entire signal space as a whole, thus determining a pattern of metabolic response.

A variety of algorithm approaches have been reported, each with advantages and limitations. Principal component analysis (PCA) is an unsupervised method to determine if data will cluster according to groups based on orthogonal transformations of the data. It is very useful as an initial look at metabolomic data (Covington et al. 2013; Prasad et al. 2008), and can be used towards outlier detection. Genetic algorithms have been one way to build and validate whole-spectrum models of GC/DMS responses (Eiceman et al. 2006; Krebs et al. 2006b; Shnayderman et al. 2005), which has the advantage of using an optimization method to determine biomarkers within data sets to achieve high classification rates to separate groups of data. However, they sometimes converge on local optima solutions, and can be computationally expensive when optimizing for a large number of features or samples. Frequently these local optima are not general enough to be applied to data not used in training the model. A fuzzy rule building expert system (Rearden et al. 2007) has also been demonstrated for synthetic complex mixtures, such as JP-8 jet fuel, and may also be appropriate for metabolomic data sets. Linear discriminant analysis (Covington et al. 2013; Rutolo et al. 2014) and Fisher discriminant analysis (Arasaradnam et al. 2014b) have been used as classification methods for biological data as well. Partial least squares (Aksenov et al. 2014; Cheung et al. 2009) is well established and a valuable method to classify groups of GC/DMS data, and our software system has initially included this analytical tool for model building. A description of our software applying this technique (PCA, PLS, background correction and smoothing) to data that have been previously published (Aksenov et al. 2014) is presented as an illustrative example throughout this paper. The software is designed so that additional capabilities and applications can be added in future versions.

Software Structure

Our software (Figure 1) is designed to implement the most common data analysis steps frequently applied to metabolomic data sets in the literature (Peirano DJ 2013). At present, there is no software package that can apply these steps seamlessly to DMS data. This software was not built to replace chemometrician input on data analysis, but to enable researchers to have preliminary observation of their sampled data, and to determine how their data would behave during preprocessing and model development. Briefly, we will outline the major stages in our software data analysis steps here, and we will then expand on specific procedure details in later sections.

Reflecting our overall data analysis procedure, the first step is to effectively visualize the data (Figure 1, **Step 1**). Devices actively being developed or samples collected using novel techniques or on new chemical sources require slow and comprehensive visual analysis of the samples that are initially collected. This can allow the user to refine experimental protocols, as well as instrument settings to optimize their method development for a new application. A user can easily observe if the signal is absent or saturating the detector by visualizing their data. Resolution and consistency of peaks can also be observed by deliberate interaction with the size of the visual field and the representation of the peaks. This leads to a greater understanding of the data, and can identify nuances or problems in the data before going further with an experiment.

The next step in the procedure is preprocessing the data (Figure 1, **Step 2**). Lab members not normally involved with the statistical analysis of data are able to implement baseline removal and smoothing on their data and observe these effects on their samples. By understanding the chemistry involved as well as the engineering specifications of their specific device, users may have different insights to apply to the settings they choose for the preprocessing. For example, if a system has a certain data acquisition frequency, this could help dictate specific smoothing parameters to avoid artificially aliasing the signal.

Outlier identification can occur at a variety of points during the procedure (Figure 1, **Step 3**). They can be identified through visual inspection, or by running an initial Principal Component Analysis (PCA) on the raw data, or running an additional PCA after the preprocessing occurs. Identifying outliers using any combination of these methods enables the user to observe the consistency of data collected using their device and techniques. These methods can be used to identify natural sources of biological signal variance or artificial instrument noise when used in conjunction with a well thought out experiment plan.

The steps up to this point are excellent for determining the general quality of the data collected. However, for certain applications, the user may aim to differentiate between groups of samples (such as healthy or infected). The collected data may be consistent and of good quality, but that alone may not be sufficient to identify biomarker metabolite features in the spectra that are statistically different between classes of samples. Our next step is to build and test a model targeting differences in the samples that correlate to different classifications or experimental groupings (Figure 1, **Step 4**). The model can either analyze pairwise comparisons between two groups, or we can model more than one group at a time. The models themselves can be produced using various common algorithms, and the model terms and variables should be saved for later reference. Finally, the last step of our metabolomic data analysis process uses our model training techniques to predict unknown samples (Figure 1, **Step 5**).

Description of Illustrative Data Set

To demonstrate each of the steps of our data analysis procedure implemented by the software, a previously analyzed set of samples was used. We previously published a GC/DMS analysis of volatile organic compounds (VOCs) emitted by a biological system (Aksenov et al. 2014). Briefly, this data set analyzed a fingerprint set of citrus volatiles that changed after a tree was infected with a specific pathogen. Trees were placed into three classifications: healthy, asymptomatic and mild. Asymptomatic samples were samples taken from an infected tree, but from a branch that showed no indications of infection. Both mild and asymptomatic samples could also be classified as infected samples as opposed to healthy samples. Though the initial paper used samples collected from throughout the year, the samples we provide here as an illustrative data set were all collected during October 2011 over 5 days of field sampling. During the collection of samples, we found the initial samples collected each day were less consistent as data collection started immediately after initializing the instrument. To demonstrate multiple aspects of outlier detection, we have retained this data as part of our example set.

1. Data Visualization

DMS data is collected as a large matrix of values as charged ions accumulate on the detector pads during a defined time interval, Δt. Each value indicates an intensity that corresponds with the abundance of a chemical (or more than one chemical) that is measured under very specific operating conditions. For instance, GC/DMS data includes chemical abundances that are a function of both a certain retention time (RT) as a chemical elutes off the GC column, and a certain compensation voltage (CV) as an ion mobility parameter. Typically, both negative and positively charged ions are recorded simultaneously, although the positive spectra are historically the most interesting in biological metabolite applications. Since the intensities at each point in the spectra indicate the measurement of a specific chemical(s), it is important to note the retention time is determined by that chemical's interaction with the GC column and the compensation voltage is based on the interaction with the RF electric fields within the drift tube region of the DMS instrument.

The digitization capability of DMS devices can vary. In our example data set, we digitized 100 electrode readings per second. This resolution of 100 digitized values also corresponded to sweeping the CV over the selected range of the compensation voltages, enabling one compensation voltage sweep per second. This can be modified for higher resolution, depending on the analog-to-digital capabilities of the instrument (which vary by manufacturer). In general, there is a tradeoff between increased data resolution and data acquisition time required. When DMS devices are connected to GC columns, researchers must try to match resolution with the elution times of chemicals from the column; otherwise, if the digitization is too slow, chemicals may be "missed" as they elute off the column faster than data is collected. For most of the resolutions selected for RT and CV in a DMS device, indications of the presence of a chemical will span multiple compensation voltage digitized points and multiple retention time increments. This will create a large area of higher amplitude intensities that span two dimensions around a central point which is usually the location of the highest intensity. This can also be called a "peak".

The presence of certain peaks can indicate specific chemical(s), and analyzing the position and intensity of the peaks in a signal space can be used to differentiate between different groups of collected samples. Therefore, data visualization needs to maximize the user's ability to observe peak locations and intensities. This visualization step also helps users to understand how data vary between certain samples in order to assess the performance of their device and get an idea of the ability of a statistical model to differentiate between samples of varying groups. Finally, effective data visualization can assure the user with some confidence that consistent sampling is occurring. This can be especially important for biological applications, and critical for other applications of low abundance volatile organic compound (VOC) sampling (e.g. ambient air monitoring).

Visualizing samples from current GC/DMS devices can be a relatively straight forward, although critical, first step in data interpretation. In our software, we map the CV to the x-axis and the RT to the y-axis, with color indicating peak amplitude. However, the human eye does not always recognize slight differences in color in pixels far from each other, making it challenging to compare peaks over an entire signal space. Therefore, the 3-dimensional (3D) plotting capabilities of MATLAB 2014a were employed to create a visualization of a true 3D signal space with the intensity of the signal plotted along the z-axis. This allows the signal to be observed as a landscape of multiple peaks. The software can also rotate the signal space, enabling the comparison of peak heights and locations with much greater clarity. Additionally, certain regions of the data can be targeted for visualization and model building by defining the CV and RT range of interest. The range of the z-axis can also be defined to assist in visualization without affecting subsequent data processing or model building steps. This allows the user to observe significantly smaller peaks that may be visually dominated by non-relevant peaks that do not impact the analysis of the sample, for example the reactant ion peak (RIP). We show a sample from our previously published data visualized in the software, both before and after reducing the plotted range (Figure 2). When the original raw data is plotted (Figure 2a), the dynamic range of the data makes it difficult to observe small features. By reducing the plotted signal space (Figure 2b), a clear picture of potential features takes shape. Finally, when the plotted retention time dimension and compensation voltage are reduced as well (Figure 2c), this removes the RIP from the signal. We are then able to discern small amplitude peaks emerging in the spectra. These features are further observed in the 3D visual plot (Figure 2d).

2. Pre-Processing

Often DMS devices have small but persistent voltage offsets on the detector pads that can be static, or can vary slightly over time. This is most likely attributed to minor operating issues with platform electronics, but it needs to be taken into account when data is analyzed in an automated or semi-automated fashion. Baseline voltage shifts are most noticeable at points when there are no chemical peaks present in the spectra, but the value of the signal is still above zero, representing a small offset potential on the electrode. As stated previously, this offset can be constant throughout the entire signal, or it can be affected over the time of the physical sampling. This fluctuation can potentially have a large effect on relative peak amplitudes within the sample, and if the offset varies between samples, then comparing peaks intensities of different samples becomes very challenging. Without baseline removal, it is still possible to determine the positions of peaks and attempt to identify biomarkers through peak location in the signal space, but analysis based on the quantification of the peaks would suffer.

Asymmetric least squares (ALS) is a one dimensional mathematical correction algorithm (Stevenson et al. 2013) that is performed on the samples in a direction parallel to retention time axis of the signal space. In other words, for each compensation voltage, all retention times that correspond to the compensation voltage are treated as a single time series, and the baseline is found for those values. For each time series, ALS iterates a least squares calculation multiple times. Each time, all points found below the calculated least squares line are given a heavier weight than those found above the line, and using this weighting a new least squares line is calculated, until a stable point is reached. The resulting equation is polynomial and maps closely to the bottom points of the time series, while remaining mostly indifferent to the chemical peaks above it in the spectra. This creates a continuous and differentiable solution that can be removed from the data and mimic a similar baseline that could have been observed visually. We show a sample from our previously published data before (Figure 3a) and after baseline removal (Figure 3b**)**. This step allows for a clear visualization of specific features that were previously obscured by signal fluctuations.

Besides the signal change that may occur due to an electrode offset potential, other factors can include signal noise as well. Each sample is made up of tens of thousands of individual digitized data value pixels, and each of those single measurements can include noise that originates from many potential sources. Examples include but are not limited to: Brownian noise in the electronics from ambient thermal shifts, physical vibrations on the instrument, and background trace chemicals that are detected near the noise floor of the instrument. All of the sources of noise are not necessarily apparent when visually observing the spectra at full working range of the instrument, but they can have a large impact on many types of mathematical models that are built based on the quantitative measurements of biomarker peaks versus the background. To remove this noise, we implement Savitzky-Golay smoothing (Bromba and Ziegler 1981; Madden 1978; Nevius and Pardue 1984). Savitzky-Golay takes an input of window size, and M Order (which defines the order of the polynomial that could occur in the signal within the window size) and generates a set of coefficients, equal in length to the window size, with values that represent a polynomial of M Order. We show a sample from our previously published data before (Figure 4a) and after noise removal (Figure 4b**)**. Note that the smoothing is applied before the baseline removal in the preprocessing, though the user can modify the order of preprocessing steps in the software.

3. Outlier Detection

Outlier detection is a critical step in data analysis, and involves moving aberrant samples from the data set prior to modeling and interpretation. Outliers can appear in a data set due to many reasons. One common origin is improper sample collection or a mistake by the researcher who collected the sample. Another common origin of outliers is the instrumentation itself, with issues ranging from not enough purge gas to clean the instrument between runs, lack of a warm up time for the device before sampling, or simply a brief electronics malfunction or interruption. Finally, most biological data is exceptionally heterogeneous, and outliers can simply be due to actual true variation within the system. What compounds this problem, however, is that a researcher may have a biological sample that they think represents one category, but it actually belongs to a different group (e.g. assuming a sample is healthy, but is in fact diseased). Data visualization can be used to identify sampling errors and equipment malfunction, and aberrant files can be manually removed. However, it is much harder to identify outliers that have a true biological origin or mistaken group membership.

Identifying when to denote a data sample as an outlier or a “good” sample can have a large impact on models later built to interpret the overall data set and forecast future unknown samples (to be discussed in subsequent sections). Additionally, aggressive outlier identification may result in a data set that does not represent the true variability of measured data, especially from biological systems. Perhaps most importantly, some researchers try to perform outlier detection “by hand”. This raises potential biases within the available data used to make a model, and is especially likely if outside information about the “true” classifications of samples is known or inferred (e.g. a separate “gold-standard” data stream that is independent of the DMS data set).

Multiway Principal Component Analysis (PCA) treats each sample as a single point in an *n*-dimensional field, where *n* is the number of variables contained within the sample (Wold et al. 1987). PCA then optimizes for the strongest covariances among these dimensions for the sample set, and aligns the data along these principal components. Outliers have a large impact on PCA as they will have multiple variables that vary with each other away from the “normal behavior” of the sample set. Observing the primary principal components can indicate samples that do not behave as the other samples in the set, providing a visualization to identify these samples. If many of the samples are clustered in the first few principal components while a few samples are dispersed away from this grouping, those dispersed samples are defining the principal components because the data within those samples behaves differently than most. If the full set of samples is evenly dispersed across the components, then the samples behave in a similar manner, and PCA is able to correctly identify the covariances that are defined by this shared behavior. Finally, it is important to note the number of samples in the set prior to removing outliers. When there are fewer samples, each sample has a much larger impact on defining the principal components, and therefore covariances coincide with impacts from individual samples appearing much stronger than they would if there were more samples collected. This can make samples appear as outliers when they are simply demonstrating natural variances.

We use our previously published data set to illustrate outlier removal (over multiple iterations of PCA) by showing the set of “Healthy” classification samples before outlier removal (Figure 5a), and the separation of Healthy samples after outlier removal based on the two days of sampling (Figure 5b). This can be compared with the score values of healthy samples in Figure S1 from our previous paper, which also demonstrated that the natural variation of the data separated along day of sampling in October 2011 (Aksenov et al. 2014). The loading of the first principal component when outliers are present contains much more noise, and large areas of similar behavior cannot be observed (Figure 5c). This means that the strongest observed behavior between the samples does not correlate to varying chemical intensity which is usually observed in peaks covering large areas, but rather to electrical noise generated from the device. However, after outliers are removed, the loading matrix of the first principal component contains much less noise while large areas are present demonstrating variation between the samples is based on the height of these peaks indicating varying chemical intensity (Figure 5d).

4. Model Building

Frequently, chemical analysis using DMS is performed with the intention of enabling forecasting unknown samples at a later time. Data is acquired under known conditions, allowing an algorithm to be trained to represent different group responses. After modeling a “training” data set, we can then use this to make predictions about specific unknown samples in the future (to be discussed in the next section). The first step in this process is creating an adequate mathematical model representation of the initial training data.

Multiway Partial Least Squares – Discriminant Analysis (nPLS-DA) (Geladi and Kowalski 1986) is a form of supervised Principal Component Analysis (PCA) where the principal components are identified based on their ability to differentiate between groups. Similar to PCA, nPLS-DA is able to incorporate a set of samples where the number of variables contained within each sample exceeds the number of samples that were collected. This works well for GC/DMS data where a sample can contain over 50,000 digitized points of data, and a variable is a single digital value within the spectra. The variables are analyzed based on their covariance with other variables within the set of samples, and groups of variables that behave similarly are grouped into principal components. Each principal component is analyzed based on how well it corresponds with the information on assigned sample groupings. They are then "rotated" within their multi-dimensional space to best align with the determined sample groupings. This allows for the construction of a linear regression model that can use the input of a single unknown sample and multiply through by the corresponding coefficients for each variable within the sample, returning a value corresponding to a range of zero to one which represents how much it co-aligns with other samples within that classification set.

The nPLS-DA algorithm is a supervised analysis, and like other supervised techniques, the training of the model is the most important aspect in creating a robust analysis method. The initial starting data needs to be randomly separated into training and testing sets, where samples of specific classification groups are proportionally represented. When building the nPLS-DA model, only the training set of data is used. The accuracy of the model is then subsequently assessed by applying it to the remaining testing set not initially used. This process creates models that are less impacted by "overtraining" and tend to perform better on additional future unknown sample forecasting.

There is a commonly used model building process termed "Leave-One-Out." In this case a model is built with all but one of the samples, and the last sample is treated as a testing set of size 1. This process is done repetitively, and an aggregate accuracy of many models can be calculated after this process completes a specific number of modeling cycles for all samples. This can be prohibitively slow, and an alternative is to create a training set of just 20-30% of the original data set samples. This may better reflect the overall forecasting accuracy of the model on true unknowns since less of the original data is used in the model training process. We should note that if a linear supervised analysis is used (such as nPLS-DA) then the Leave-One-Out approach should always result in the same classifications of the same samples. However, if the samples are randomly separated into training and testing sets, then the models will vary based on which samples were used to test them. Therefore many models should be built and tested to observe an expected performance of the model.

Unknown Sample Forecasting

For many experimentalists, a major goal for GC/DMS analysis of biological data is to enable prediction classifications of future unknown samples. Once a model is built, we go through a specific set of steps to perform this forecasting process. Using nPLS-DA, a model will return a single value between 0 and 1 for each new unknown sample applied to it that correlates to the classification of the model, and it will do this for each possible classification group in the system. For example, the nPLS-DA algorithm builds a separate model for each classification type within our illustrative data set (e.g. Healthy, HLB Asymptomatic, HLB Mild). Based on the method of model training (i.e. Leave-One-Out) each sample is evaluated as “unknown” to the trained model and values are returned that indicate the probability that sample is from each of the different groups. The ideal forecasting case results when a sample has a value of 1 calculated for a single classification, and value of 0 for all remaining groups. However, the model is applying a coefficient matrix to the unknown sample input, and the returned values will be a real number that ranges from 0 to 1. Models can return values outside of this range as the sum of the product of the model final coefficients and a specific sample may produce values slightly outside this range. Additionally, models that have few samples in certain classifications (as this one does in the mild classification) can often bias towards opposing classifications as it is being trained on much more data in those classifications, and therefore a “right answer” in the training is much more often to be away from the classification with a limited amount of samples. The most straightforward way of assigning forecasts to an unknown sample is to identify the classification that corresponds with the highest predicted value from the models. This reflects which model the sample most closely aligns with, although it can be hurt if a model based on a classification with few samples biases away in its predictions of those types of samples, and is still susceptible to a variety of factors including low sample numbers during the model building process. This type of classification approach does not assess if all of the classification models are returning values that are appropriate for a predicted group. In other words, for every sample there will always be a model that returns the largest corresponding value, but the other models may also be returning very high corresponding values for their classifications. Another problem that could surface is that other samples in the same classification may normally return much higher responses in that classification. These other aspects of the prediction of a sample should be incorporated into assessing if a valid prediction is being made.

A different approach is to define single or even multiple thresholds when assessing a sample prediction from the model. The initial step is to identify if thresholds can even be used on the predictions. For each classification model (i.e. HLB Mild), the mean and standard deviation are found for samples that correspond with this model (HLB Mild) and samples that do not correspond with this model (HLB Asymptomatic and Healthy). If the difference between the means of each set is less than two times the sum of the standard deviations from each set, then the model is not effective at separating the samples, at least in a manner that can make use of multiple thresholds. This is repeated for each type of classification. If all classification models have passed this initial test, then each unknown sample is assessed. If it is found to be within two standard deviations of the mean for a certain classification, and within two standard deviations of the means for samples that fail for all the rest of the classifications, then it is identified as being predicted to be a member of that classification. If it is found that it does not qualify this way, then it is identified as being unable to be predicted as any classification for this category. The model developed to classify samples based on healthy and infected was not able to differentiate between the groups strongly enough to make a definitive threshold, but there is clearly two different groups observed as shown by a box plot of the predictions of the model (Figure 6b).

Viewing the scores plot and loadings of the latent variables identified by nPLS can also show the behavior of the samples as well as identify where in the sample space the chemical peaks occur that most strongly differentiate between desired classifications. However, because the scores and loadings would need to correspond to a single model, a unique model is built using all of the data, demonstrating the groupings of the calculated scores (Figure 6a). The loadings for this unique model are also shown, and as can be seen in the scores plot, the first latent variable effectively maps the behavior that best correlates with differentiating healthy from sick, while the second latent variable correlates well with differentiating between asymptomatic and mild (Figure 6c).

Using the illustrative data set from our previous paper (Aksenov et al. 2014), we can demonstrate the development of an nPLS model using a Leave-One-Out training technique. For this model, the CV range selected was −23 V to 15 V and the RT range was 0 to 230 seconds. The order of pre-processing was Savitzky-Golay smoothing with a window size of 9 and an M Order of 3, then ALS baseline removal using a λ of 10^2 and a proportion of positive residuals of 0.01. Both preprocessing techniques were applied along constant CV values. The number of latent variables for the nPLS model was set at 2. Samples classified as asymptomatic and mild were combined into a classification of infected to be compared with samples classified as healthy. The confusion matrix for the results of the model is shown (Table 1).

There are some differences in the techniques used in this paper and the initial analysis of the illustrative data we previously published (Aksenov et al. 2014). In this paper we use our newly developed software, while in the initial paper the PLS_Toolbox developed by Eigenvector Research Inc. was used. The largest difference in the two implementations is during baseline removal, where instead of ALS as used in the current software, ALS Basis method is used. This introduces small differences that are reflected in slight discrepancies in outlier detection, resulting in different samples used to train and test the final models. Additionally, while this is a demonstration of an implementation of our procedure and for the final assessment and we used Leave-One-Out as a training method, for the previous paper, 300 models were built using a 70/30 breakdown of training and testing to ensure that the accuracy identified would appropriately reflect the final model built.

Despite these differences in analysis, the resulting accuracies are very similar. In the previous paper, the accuracy for classification of healthy samples was found to be 95.49 ± 4.11%, while the current software achieved an accuracy of 92.5%. For the accuracy of the classification of HLB infected samples, the previous paper achieved an accuracy of 88.48 ± 11.62% while this software achieved an accuracy of 83.3%.

Discussion and Conclusions

New data visualization tools such as our present software might aid additional advanced hardware and DMS system development (Zhao et al. 2008). For example, if certain regions of the GC/DMS or Py/GC/DMS data sets are particularly interesting and feature rich, then the hardware could be modified to enhance detection within that part of the data. Likewise operating protocols such as GC times or CV range adjustments may allow for higher accuracy once a pilot study indicates the portion of the signal with the most diagnostic power.

While we have presented one way to analyze GC/DMS metabolomic data, there are certainly more avenues to explore. One limitation of the current approach is the notion of discrete state classification, which assumes that new incoming samples must belong to one of the major classification groups, e.g. "healthy" or "sick". This paradigm can work well for many examples, such as our plant pathogen infection data presented as the example case in this paper. However, some metabolic diseases and disorders are more of a continuum of biological response that cannot always be articulated as discrete states. If we initially choose the two far ends of the spectrum as the training example, we cannot always predict how the model will classify new samples from the more moderate intermediate portion of the response. Likewise, even if researchers are exploring a true two-case system, an unknown sample may be taken during the transition of the biological organism from one state to the other (e.g. you measure it while moving from "healthy" to a "sick" state). If this is the case, the model may produce unreliable results that may be misinterpreted. Data taken during time-course experiments may be especially susceptible to this state-transition artifact without a firm understanding of the underlying biological pathophysiology being studied. Extremely well-controlled studies with appropriate replicate numbers are critical.

When we do consider a multi-state model, setting up the initial training study groups is very important to the success of this research. Frequently there are biological diseases or disorders with multiple etiologies, and there can be several different causes of the underlying pathology. If these are mixed into a single "diseased" state, then care must be taken to create a large enough training data set that well represents each of the sub-state conditions. It may be advantageous to include confounding factors, such as co-infections, at the time of training. This could lower the probability of false alarm in the model testing phase. Care should be taken not to systematically bias the data as well, e.g. all data from one state taken at a single time point, and data taken from another state taken at another time point. All of these factors may influence poor model building outcomes and performance.

Finally, it is important to note the effect experimental drift of operating conditions may have on model building. If the experiments do not seek to identify specific chemical profiles and are relying on the total pattern of GC/DMS response to match a class pattern, it is important to rigorously ensure that device operation does not change during the model building or model testing phases of the study. Examples might include: large changes to the GC column temperature profile which can lead to severe signal misalignment; altering the RF profile which would affect CV profile changes for metabolites; large humidity shifts during sampling that may affect the DMS reactant ion peak; and variation of sampling time could lead to non-uniformity of metabolite abundances between samples.

When used appropriately, we believe that our new data analysis software is likely to be a flexible platform that can be useful to many researchers in the community. In this present study, we outline one possible order of data processing steps, but the software code itself is flexible and can be adapted in the future for new modules developed by others. While we initially coded nPLS-DA for model building and testing, it is possible to add a model to the software that was built using other methods, such as genetic algorithms or neural networks. It is also possible that other researchers will later enhance the software to incorporate alternative model building strategies directly into our procedure as well.

Software

The software operates through MATLAB and was coded on MATLAB Version 8.3.0.532 (R2014a). It makes use of the Parallel Computing Toolbox, although this is not required for operation. The software currently reads input files through a tab-delimited file format which contains the compensation voltage and retention time alongside the matrix of recorded data. Additionally, data can be read directly from variables in the MATLAB workspace, with compensation voltage and retention time assigned by the user while importing the data.

All of the data visualizations within the figures of this manuscript were generated using our software, AnalyzeIMS (AIMS), Version 1.11. Examples of the current software interface are provided as supplemental figures (Supplemental Figures 1-8).

Software Dissemination

The AnalyzeIMS (AIMS) software is available on GitHub. Please refer to Professor Cristina Davis' webpage for more information. AnalyzeIMS is available as open source for research and personal use under a modified BSD license. Commercial licensing may be available, and a license fee may be required. The Regents of the University of California own the copyrights to the software. Future published scientific manuscripts or reports using this software must cite this original publication.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgements

Partial funding for this study was provided by: the National Science Foundation (NSF) grant #1255915 [CED], the California Citrus Research Board grant #5100-143 and #1500-159 [CED], The Hartwell Foundation [CED] and the United States Department of the Army grant W15P7T-12-C-A005 [CED], National Institutes of Health (NIH) grant number #1U01EB022003-01 and NIH grant #UL1 TR000002 [CED]. Student support was partially provided by the US Department of Veterans Affairs, Post-9/11 GI-Bill [DJP], and the National Science Foundation grant #1343479 Veteran's Research Supplement [DJP]. The contents of this manuscript are solely the responsibility of the authors and do not necessarily represent the official views of the funding agencies.

The authors would like to thank members of their research group for Beta testing the software and suggested improvements: Raquel Cumeras, Mitchell M. McCartney, Sierra Spitulski, and Yuriy Zrodnikov.

References

Aksenov AA, et al. Detection of Huanglongbing Disease Using Differential Mobility Spectrometry. Analytical Chemistry. 2014; 86:2481–2488. DOI: 10.1021/ac403469y [PubMed: 24484549]

Arasaradnam RP, et al. Detection of Colorectal Cancer (CRC) by Urinary Volatile Organic Compound Analysis. Plos One. 2014a; 9doi: 10.1371/journal.pone.0108750

Arasaradnam RP, et al. Differentiating Coeliac Disease from Irritable Bowel Syndrome by Urinary Volatile Organic Compound Analysis - A Pilot Study. Plos One. 2014b; 9doi: 10.1371/journal.pone.0107312

Basanta M, et al. Non-invasive metabolomic analysis of breath using differential mobility spectrometry in patients with chronic obstructive pulmonary disease and healthy smokers. Analyst. 2010; 135:315–320. DOI: 10.1039/b916374c [PubMed: 20098764]

Bromba MUA, Ziegler H. Application hints for Savitzky-Golay digital smoothing filters. Analytical Chemistry. 1981; 53:1583–1586. DOI: 10.1021/ac00234a011

Camara M, Gharbi N, Lenouvel A, Behr M, Guignard C, Orlewski P, Evers D. Detection and Quantification of Natural Contaminants of Wine by Gas Chromatography-Differential Ion Mobility Spectrometry (GC-DMS). Journal of Agricultural and Food Chemistry. 2013; 61:1036–1043. DOI: 10.1021/jf303418q [PubMed: 23356506]

Cheung W, Xu Y, Thomas CLP, Goodacre R. Discrimination of bacteria using pyrolysis-gas chromatography-differential mobility spectrometry (Py-GC-DMS) and chemometrics. Analyst. 2009; 134:557–563. DOI: 10.1039/b812666f [PubMed: 19238294]

Covington JA, et al. Application of a Novel Tool for Diagnosing Bile Acid Diarrhoea. Sensors. 2013; 13:11899–11912. DOI: 10.3390/S130911899 [PubMed: 24018955]

Cumeras R, Figueras E, Davis CE, Baumbach JI, Gracia I. Review on Ion Mobility Spectrometry. Part 1: Current instrumentation Analyst. 2015a; 140:1376–1390. DOI: 10.1039/c4an01100g [PubMed: 25465076]

Cumeras R, Figueras E, Davis CE, Baumbach JI, Gracia I. Review on Ion Mobility Spectrometry. Part 2: Hyphenated methods and effects of experimental parameters Analyst. 2015b; 140:1391–1410. DOI: 10.1039/c4an01101e [PubMed: 25465248]

Davis CE, et al. Analysis of Volatile and Non-Volatile Biomarkers in Human Breath Using Differential Mobility Spectrometry (DMS). IEEE Sensors Journal. 2010; 10:114–122. DOI: 10.1109/jsen.2009.2033562

Eiceman GA, Wang M, Prasad S, Schmidt H, Tadjimukhamedov FK, Lavine BK, Mirjankar N. Pattern recognition analysis of differential mobility spectra with classification by chemical family. Analytica Chimica Acta. 2006; 579:1–10. DOI: 10.1016/j.aca.2006.07.013 [PubMed: 17723720]

Fong SS, Rearden P, Kanchagar C, Sassetti C, Trevejo J, Brereton RG. Automated Peak Detection and Matching Algorithm for Gas Chromatography-Differential Mobility Spectrometry. Analytical Chemistry. 2011; 83:1537–1546. DOI: 10.1021/ac102110y [PubMed: 21204557]

Geladi P, Kowalski BR. Partial least-squares regression: a tutorial. Analytica Chimica Acta. 1986; 185:1–17. doi:http://dx.doi.org/10.1016/0003-2670(86)80028-9.

Kendler S, Lambertus GR, Dunietz BD, Coy SL, Nazarov EG, Miller RA, Sacks RD. Fragmentation pathways and mechanisms of aromatic compounds in atmospheric pressure studied by GC-DMS and DMS-MS. International Journal of Mass Spectrometry. 2007; 263:137–147. DOI: 10.1016/j.ijms.2007.01.011

Krebs MD, Kang JM, Cohen SJ, Lozow JB, Tingley RD, Davis CE. Two-dimensional alignment of differential mobility spectrometer data. Sensors and Actuators B-Chemical. 2006a; 119:475–482. DOI: 10.1016/j.snb.2005.12.058

Krebs MD, Mansfield B, Yip P, Cohen SJ, Sonenshein AL, Hitt BA, Davis CE. Novel technology for rapid species-specific detection of Bacillus spores. Biomolecular Engineering. 2006b; 23:119–127. DOI: 10.1016/j.bioeng.2005.12.003 [PubMed: 16542873]

Krylov EV, Coy SL, Vandermey J, Schneider BB, Covey TR, Nazarov EG. Selection and generation of waveforms for differential mobility spectrometry. Review of Scientific Instruments. 2010; 81doi: 10.1063/1.3284507

Lu Y, Chen P, Harrington PB. Comparison of differential mobility spectrometry and mass spectrometry for gas chromatographic detection of ignitable liquids from fire debris using projected difference resolution. Analytical and Bioanalytical Chemistry. 2009; 394:2061–2067. DOI: 10.1007/s00216-009-2786-9 [PubMed: 19396432]

Lu Y, Harrington PB. Forensic application of gas chromatography - Differential mobility spectrometry with two-way classification of ignitable liquids from fire debris. Analytical Chemistry. 2007; 79:6752–6759. DOI: 10.1021/ac0707028 [PubMed: 17683164]

Madden HH. Comments on Savitzky-Golay convolution method for least-squares fit smoothing and differentiation of digital data. Analytical Chemistry. 1978; 50:1383–1386. DOI: 10.1021/ac50031a048

Manard, M.; Weeks, S.; Kyle, K.; Ieee. Monitoring/verification using DMS: TATP example 2008 IEEE Conference on Technologies for Homeland Security; 2008. p. 226-230.

Manard MJ, Trainham R, Weeks S, Coy SL, Krylov EV, Nazarov EG. Differential mobility spectrometry/mass spectrometry: The design of a new mass spectrometer for real-time chemical analysis in the field. International Journal of Mass Spectrometry. 2010; 295:138–144. DOI: 10.1016/j.ijms.2010.03.011

Nevius TA, Pardue HL. Development and preliminary evaluation of modified Savitzky-Golay smoothing functions. Analytical Chemistry. 1984; 56:2249–2251. DOI: 10.1021/ac00276a061

Peirano, DJAA.; Pasamontes, A.; Davis, CE. Chapter 18: Approaches for Establishing Methodologies in Metabolomic Studies for Clinical Diagnostics. In: Agah, A., editor. Medical Applications of Artificial Intelligence. CRC Press, Taylor Francis Group; 2013. p. 279-304.

Prasad S, et al. Constituents with independence from growth temperature for bacteria using pyrolysis-gas chromatography/differential mobility spectrometry with analysis of variance and principal component analysis. Analyst. 2008; 133:760–767. DOI: 10.1039/b716371a [PubMed: 18493677]

Rearden P, Harrington PB, Karnes JJ, Bunker CE. Fuzzy rule-building expert system classification of fuel using solid-phase microextraction two-way gas chromatography differential mobility spectrometric data. Analytical Chemistry. 2007; 79:1485–1491. DOI: 10.1021/ac060527f [PubMed: 17297947]

Rutolo M, Covington JA, Clarkson J, Iliescu D. Detection of Potato Storage Disease via Gas Analysis: A Pilot Study Using Field Asymmetric Ion Mobility Spectrometry. Sensors. 2014; 14:15939–15952. DOI: 10.3390/s140915939 [PubMed: 25171118]

Schivo M, et al. A mobile instrumentation platform to distinguish airway disorders. Journal of Breath Research. 2013; 7doi: 10.1088/1752-7155/7/1/017113

Shnayderman M, et al. Species-specific bacteria identification using differential mobility spectrometry and bioinformatics pattern recognition. Anal Chem. 2005; 77:5930–5937. DOI: 10.1021/ac050348i [PubMed: 16159124]

Stevenson PG, Conlan XA, Barnett NW. Evaluation of the asymmetric least squares baseline algorithm through the accuracy of statistical peak moments. J Chromatogr A. 2013; 1284:107–111. DOI: 10.1016/j.chroma.2013.02.012 [PubMed: 23453461]

Wold S, Geladi P, Esbensen K, Öhman J. Multi-way principal components-and PLS-analysis. Journal of Chemometrics. 1987; 1:41–56. DOI: 10.1002/cem.1180010107

Zhao W, Bhushan A, Santamaria AD, Simon MG, Davis CE. Machine Learning: A Crucial Tool for Sensor Design. Algorithms. 2008; 1:130–152. DOI: 10.3390/a1020130 [PubMed: 20191110]

Zhao W, Sankaran S, Ibanez AM, Dandekar AM, Davis CE. Two-dimensional wavelet analysis based classification of gas chromatogram differential mobility spectrometry signals. Analytica Chimica Acta. 2009; 647:46–53. DOI: 10.1016/j.aca.2009.05.029 [PubMed: 19576384]

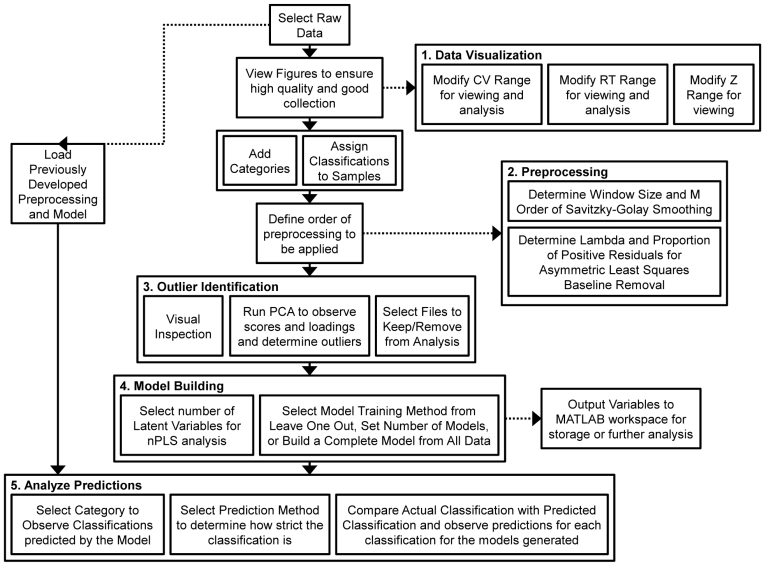
Select Raw Data
View Figures to ensure high quality and good collection
1. Data Visualization
Modify CV Range for viewing and analysis
Modify RT Range for viewing and analysis
Modify Z Range for viewing
Load Previously Developed Preprocessing and Model
Add Categories
Assign Classifications to Samples
2. Preprocessing
Determine Window Size and M Order of Savitzky-Golay Smoothing
Determine Lambda and Proportion of Positive Residuals for Asymmetric Least Squares Baseline Removal
Define order of preprocessing to be applied
3. Outlier Identification
Visual Inspection
Run PCA to observe scores and loadings and determine outliers
Select Files to Keep/Remove from Analysis
4. Model Building
Select number of Latent Variables for nPLS analysis
Select Model Training Method from Leave One Out, Set Number of Models, or Build a Complete Model from All Data
Output Variables to MATLAB workspace for storage or further analysis
5. Analyze Predictions
Select Category to Observe Classifications predicted by the Model
Select Prediction Method to determine how strict the classification is
Compare Actual Classification with Predicted Classification and observe predictions for each classification for the models generated

Fig. 1.

The procedure used to analyze data, implemented using our software. The vertical path downwards demonstrates the minimum required to develop a model from a set of raw data samples, while the dashed lines are optional steps that could improve the capabilities of the developed model or provide value to the researcher. The numbered boxes contain the primary steps of the analysis covered in this paper. The boxes inside these steps define specific algorithms that we have included in this software (i.e. Savitzky-Golay Smoothing, nPLS), but other algorithms can be used to complement or replace the existing ones.

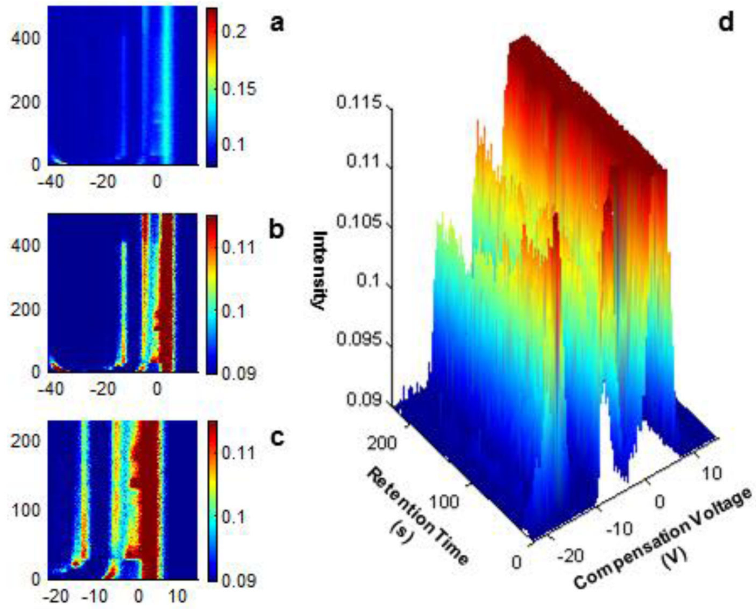

Fig. 2.

Interacting with a sample by modifying the scales of the axes and using 3D functionality. (a) The full raw data of a sample. (b) Modifying the range of the intensity to less than one fifth of its initial length is able to amplify visible features in the sample. (c) Modifying the CV Range and RT Range to remove the RIP and focus on the area of activity in the sample. (d) A rotation of (c) to demonstrate the ability to view the data in 3 dimensions. This allows users to clearly observe peak feature behavior in their samples, enabling the researcher to optimize their device and techniques used to collect the samples.

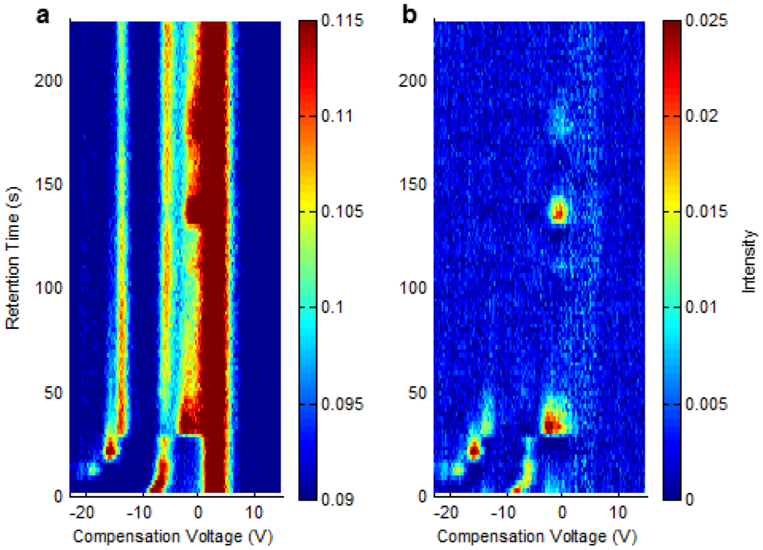

Fig. 3.

Baseline removal using Asymmetric Least Squares (ALS) with settings of λ = 100 (10^2), and PPR = 0.01. (a) A sample has been appropriately scaled to the targeted area of activity (shown previously, Fig. 2) but has long bleeds appearing after peaks measured by the device. (b) A sample after baseline removal, with only the initiating peaks shown and almost all evidence of bleeds removed. The intensity shift of the sample as shown by the colorbar also indicates the removal of unnecessary baseline so that the data is primarily located near the value of 0.

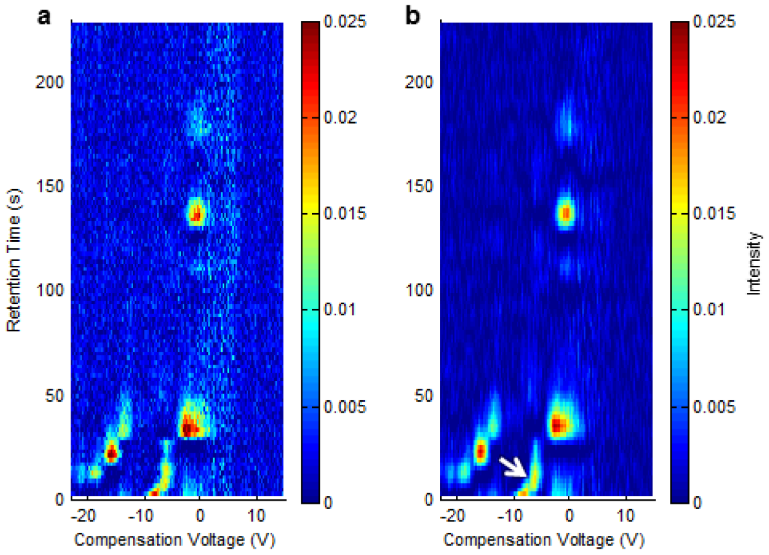

Fig. 4.

Smoothing using Savitzky-Golay with a window size of 9, and an M order of 3. (a) A sample has baseline removal applied to it (shown previously, Fig. 3), but still contains evidence of noise which can obstruct the construction of a model. (b) A sample after smoothing is applied removes the noise and the related confounding effects that can obscure model building. This can sometimes lead to combining previously separate peaks (as shown by white arrow) and care must be taken to minimize this occurrence. The order of preprocessing for the model used in this paper is smoothing first and then baseline removal.

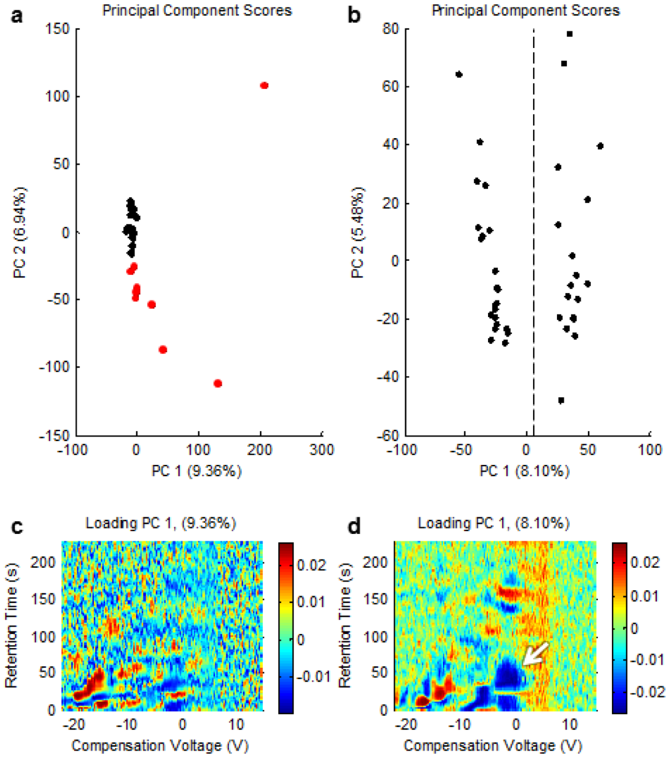

Fig. 5.

Outlier identification using Principal Component Analysis (PCA). (a) PCA scores plot of healthy samples after removal of samples identified through visual inspection. Samples shown in red were identified as outliers in this and subsequent runs of PCA before the final set of viable samples was identified. (b) PCA score-plot of the final set of samples remaining after outlier removal. Samples with a score less than zero on the first principal component were sampled on 05 October 2011, while samples with a score greater than zero on the first principal component were sampled on 03 October 2011. (c) Loading matrix of the first principal component for Fig. 5a. (d) Loading matrix for the first principal component for Fig. 5b. Outliers in the PCA that generated the loading in Fig. 5c create a component which identifies that the strongest correlation is primarily due to electronic noise shown by many small peaks and valleys. The removal of outliers enables PCA to generate the loading in Fig. 5d, where the unsupervised analysis identifies correlation between large chemical peaks and valleys (noted by white arrow) which we suspect correspond to chemical biomarkers that effectively discriminate between the data, resulting in separation based on specific days that the samples were collected.

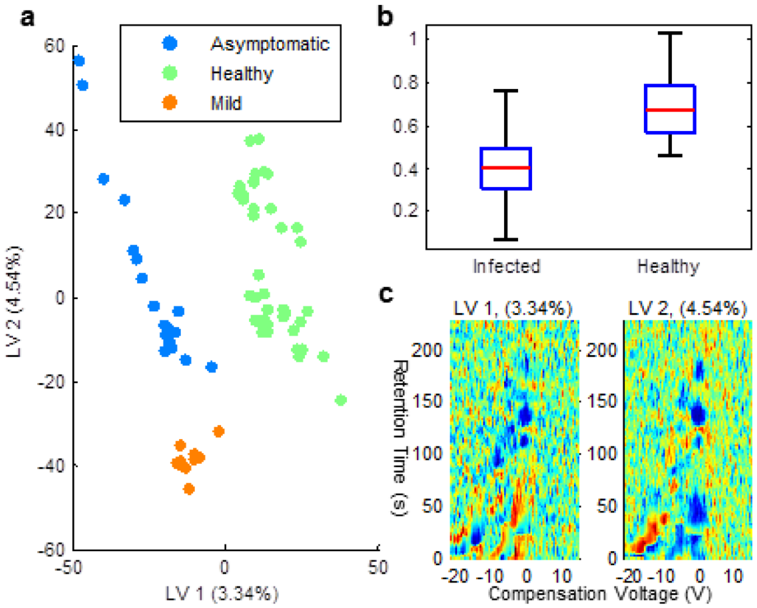

Fig. 6.

Observed separation of groups based on multiway Partial Least Squares – Discriminant Analysis (nPLS-DA). (a) nPLS-DA score plot based on separation of samples identified as healthy from other samples. The classifications of asymptomatic and mild are not included in the model, but are identified here to show additional groupings that are developed in the score plot. A full model with all samples after outlier removal included in the training set is used so that constant loadings can be applied to the data for development of a single score plot. The first latent variable effectively separates healthy from both classifications of infected. The second latent variable improves this separation and also identifies aspects that separate asymptomatic from mild without these separate classifications included in the building of the model. (b) Box plot of predicted nPLS-DA results using leave-one-out methodology to identify healthy from infected samples. (c) The loadings used by the full nPLS-DA model to separate healthy samples from other samples.

Table 1

Confusion matrix of the model developed using the current software on illustrative data previously collected and analyzed. TPR is the true positive rate, also known as sensitivity and TNR is the true negative rate, also known as specificity. PPV is the positive predictive value, also known as precision, and NPV is the negative predictive value. ACC is the total accuracy of the model. This table can be compared to the Fall classifications shown in Table 1 of Aksenov AA, Pasamontes A, Peirano DJ, Zhao W, Dandekar AM, Fiehn O, Eshani R, Davis CE. (2014) Detection of Huanglongbing disease using differential mobility spectrometry. *Analytical Chemistry* 86(5): 2481-2488.

	True Infected (30)	**True Healthy (40)**	
Predicted Infected (**28**)	**25**	**3**	PPV: **89.3%**
Predicted Healthy (**42**)	**5**	**37**	NPV: **88.1%**
	TPR: **83.3%**	TNR: **92.5%**	ACC: **88.6%**