Conrad *et al. BMC Bioinformatics* (2017) 18:160
DOI 10.1186/s12859-017-1565-4

BMC Bioinformatics

RESEARCH ARTICLE Open Access

Sparse Proteomics Analysis – a compressed sensing-based approach for feature selection and classification of high-dimensional proteomics mass spectrometry data

Tim O. F. Conrad[1,5*], Martin Genzel[2], Nada Cvetkovic[1], Niklas Wulkow[1], Alexander Leichtle[3], Jan Vybiral[4], Gitta Kutyniok[2] and Christof Schütte[1,5]

Abstract

Background: High-throughput proteomics techniques, such as mass spectrometry (MS)-based approaches, produce very high-dimensional data-sets. In a clinical setting one is often interested in how mass spectra differ between patients of different classes, for example spectra from healthy patients vs. spectra from patients having a particular disease. Machine learning algorithms are needed to (a) identify these discriminating features and (b) classify unknown spectra based on this feature set. Since the acquired data is usually noisy, the algorithms should be robust against noise and outliers, while the identified feature set should be as small as possible.

Results: We present a new algorithm, *Sparse Proteomics Analysis* (*SPA*), based on the theory of compressed sensing that allows us to identify a minimal discriminating set of features from mass spectrometry data-sets. We show (1) how our method performs on artificial and real-world data-sets, (2) that its performance is competitive with standard (and widely used) algorithms for analyzing proteomics data, and (3) that it is robust against random and systematic noise. We further demonstrate the applicability of our algorithm to two previously published clinical data-sets.

Keywords: Machine learning, Feature selection, Classification, Compressed sensing, Sparsity, Proteomics, Mass spectrometry, Clinical data, Biomarker

Background

During the last decade, high-throughput assays systems[1] for measuring a variety of different biological sources have become standard in modern laboratories. This allows for the quick and cheap creation of very large data-sets which characterize for example the status of a cell by its billions of constituents, e.g. nucleotides, RNAs, contained proteins, or metabolites. Ideally, analyzing these massive data-sets leads to a better understanding of the underlying biological processes. Especially in the context of characterizing—and ultimately understanding—diseases, a first step is often to find significant differences in the data between samples from healthy and diseased individuals. There are many successful examples where this approach based on -omics data (e.g., genomics, proteomics, or metabolomics) led to the identification of biological markers, enabling a new type of molecular diagnostics. We call a collection of biological markers that represents the differences on the data level a *disease fingerprint*.

Many disease-relevant mechanisms are controlled by proteins (e.g. hormones), which can be detected in biological samples (blood, urine, etc.) using *mass spectrometry* (*MS*). This technique allows (potentially) for monitoring the entire set of proteins—the so-called proteome—in

*Correspondence: conrad@math.fu-berlin.de
[1]Department of Mathematics, Freie Universität Berlin, Arnimallee 6, Berlin, Germany
[5]Zuse Institute Berlin, Takustr. 7, Berlin, Germany
Full list of author information is available at the end of the article

a given sample. Due to its wide availability in hospitals, MS-based proteomics can bring the next wave of progress in diagnostics, since even subtle changes in the proteome can be detected and linked to disease onset and progression [1–4].

Disease fingerprints: The main idea of the identification of *disease fingerprints* using MS-based proteomics is sketched in Fig. 1:

(a) A mass spectrum is generated reflecting the constitution of a given (blood-)sample with respect to contained molecules. (b) Based on mass spectra from two sample groups (representing a healthy control group and a group having a particular disease) differences are detected. This set of differences precisely corresponds to a *disease fingerprint*, since it represents a trace caused by a particular disease in the proteome. Several studies have shown that this approach works well in practice and found differences do indeed reflect correlations between changes in the mass spectrum, the proteome, and phenotypic changes ([5–9]). Panels of proteomic markers (fingerprints) have been shown to be more sensitive and specific than conventionally biomarker approaches [2], for example when diagnosing cancer [10–12]. However, a single proteomics data-set can contain tens of millions of signals which is many orders of magnitudes larger than the number of available observations in a typical study.

Our ultimate goal is therefore to build a library of proteomics disease fingerprints which are extracted from high-throughput MS experiments. These would enable to diagnose diseases based on their proteomic fingerprints—just by analyzing an individual's proteome. Ideally, these fingerprints are of low-complexity allowing easy interpretation by experts, e.g. medical doctors, and the implementation of medical assays for routine diagnostics, e.g. in an hospital environment. Clearly, the less components an assay is composed of, the easier it is to implement and interpret.

Thus, a fingerprint should only consist of a minimal collection of proteins specific for a particular disease and should be robust against noisy measurements. On the other hand, the acquired data from the high-throughput experiments is very high-dimensional and contains large amounts of random and systematic noise which makes an automatic analysis of mass spectra a very challenging task. Hence, the discovery of biomarkers is still a widely open research topic and there are several analytic problems that hinder reproduction of results (see [13] for example).

Despite these challenges there is indeed hope that these disease specific, low-complexity fingerprints exist: It has been shown for several cancer types that a small numbers of genes and proteins can be identified that serve as biomarkers (e.g. for lung cancer [14], breast cancer [15] or pancreas cancer [16]). This means that only a few signals in a mass spectrum can be used to derive a sparse classifier.

MS1 data: In this work we consider mass spectrometry (MS) data acquired from a standard MALDI-TOF instrument because it is easy to obtain using comparatively cheap MS-instruments which are widely available, e.g. in hospitals. Opposed to other approaches such as tandem mass spectrometry (MS/MS), we directly work on the raw data acquired in *profile mode* and do not aim for identification. Thus, each mass spectrum (sample) always has the same number of d dimensions (number of entries).[2] Recall, that the entries in a mass spectrum are a weight-ordered list of ion-counts of the respective ion-masses. (See also Fig. 1.)

One of the reasons for this is that standard approaches for MS data analysis usually convert the MS data to peak lists as a first step and work on the converted data.

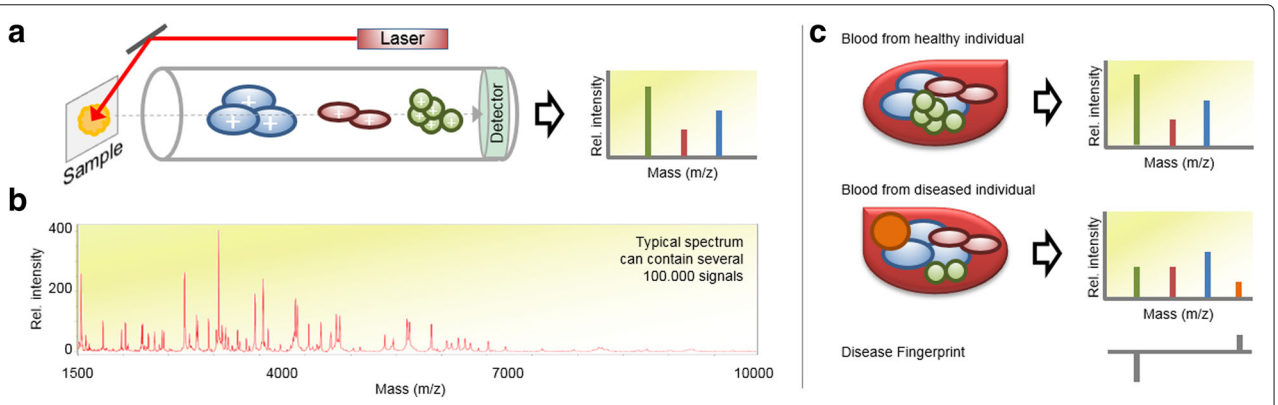

Fig. 1 a Schematic outline of a linear matrix-assisted laser desorption ionization (MALDI)–time-of-flight (TOF) mass spectrometer (MS). During the measurement process, the molecules of the examined sample are ionized, vaporized and finally analyzed by their respective time-of-flight through an electric field. This process generates a plot (mass spectrum) having mass-to-charge ratio (m/z) on the x-axis and intensity (ion count) on the y-axis. **b** Typical mass spectrum for a mass range of 1500–10.000 Dalton. **c** Example of a disease fingerprint, created by comparing mass spectra from a healthy and a diseased individual

However, signals can be missed by this conversion step due to noise or missing values in the raw data which hinders peak detection. Opposed to this, our approach does not rely on any peak identification but works on the raw data. This allows for a more robust analysis in presence of noise which is a typical challenge in MS data analysis.

Problem definition

In this article, we will focus on the following problem setting:

We assume that we are given data of n mass spectra derived from n biological samples (e.g. from blood of n individual patients) in form of n pairs $\{(x_i, y_i)\}_{i=1\dots n}$. Here, $x_i \in \mathbb{R}^d$ represents the mass spectrum of the i-th sample (e.g. the i-th patient) and $y_i \in \{-1, +1\}$ its respective class, e.g., healthy or diseased. Thus, each x_i (representing an individual mass spectrum) contains d entries.

The goal is to identify a (small) set of features, i.e. indices in the mass spectrum, separating these two classes. Thus, a feature represents a specific position (or mass) in a mass spectrum in which the two groups (e.g. healthy vs. diseased) differ. This corresponds to the well known problem of *feature selection*[3] and leads to a potential disease fingerprint for the given data.

Mathematically, this can be formulated as the identification of a *feature vector* $\omega_0 = (\omega_{0,1}, \dots, \omega_{0,d}) \in \mathbb{R}^d$ such that[4]

$$y_i = \operatorname{sign}\left(f_{\omega_0}(x_i)\right) \quad \text{for "many" samples} \quad i = 1, \dots, n, \tag{1}$$

with a *linear decision function* $f_{\omega_0}(x_i) := \langle \omega_0, x_i \rangle = \sum_{j=1}^{d} \omega_{0,j} x_{i,j}$.

From a geometric perspective, this means that the hyperplane with normal vector ω_0 appropriately separates the data-points of the respective classes.

This means that ω_0 can be used as a linear classifier where each entry of ω_0 corresponds to a specific position in a spectrum and the non-zero entries (which we call features) indicate their significance. Our goal is therefore to learn a sparse ω_0 for which Eqn. 1 holds. As a particular consequence, a classifier based on such ω_0 will yield good prediction accuracy.

In most realistic scenarios for feature selection, unfortunately, the number of features is much larger than available samples ($d \gg n$) and the data suffers from noisy measurements. For these reasons, the number of feasible classifiers ω_0 can become extremely large, so that the problem of *overfitting* can occur. In order to allow interpretability and generalization of the classifier, it is in fact inevitable to restrict the solution space for ω_0. In this paper, we focus on very *sparse*[5] vectors ω_0 satisfying (1), which precisely reflects our wish for a minimal disease fingerprint.

At this point, it should be emphasized that (1) does not need to hold for *all* samples but rather for most of them. Allowing for such a small "mismatch" in the model, we incorporate the crucial fact that a simple binary output model, such as (1), might describe the disease label only with high accuracy but not necessarily exactly. In turn, this asks for a certain robustness of the used method against wrong predictions with regard to (1).

We will approach this challenge by formulating the feature selection problem as a constrained (or regularized) optimization problem:

$$\min_{\omega \in \mathbb{R}^d} \sum_{i=1}^{n} L\left(y_i, f_\omega(x_i)\right) \quad \text{subject to} \quad R(\omega) \leq \lambda, \tag{2}$$

where $L\colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a *loss* (error) function, $R\colon \mathbb{R} \to \mathbb{R}$ is a *regularization* (cost) function that encourages a particular structure of ω (e.g., sparsity), and the parameter $\lambda \geq 0$ controls the degree of model complexity. Given any potential feature vector ω and the (true) output label y, the loss function $L(y, f_\omega(x))$ measures the discrepancy between the actual and the desired prediction.

As already pointed out, we are particularly interested in a method that produces *optimal and robust solutions* in the following situation:

- The input data (x, y) are noisy,
- the number of data dimensions d is large (typically: $d = 10^5 \dots 10^8$),
- the number of samples n is relatively small (typically: $n = 10^2 \dots 10^4$), and
- the set of highly-relevant features is small (i.e., a minimal disease fingerprint indeed exists), which corresponds to a small number of non-zero elements in ω_0 (typically: $\#\{i \mid \omega_{0,i} \neq 0\} \ll 100$).

On the contrary, we are not mainly interested in the methods' overall classification performance. Measures of classification performance such as accuracy are indicators whether a learned classifier accurately separates the data into classes. In our case, we assume that the data can be characterized well by a *sparse* classifier ω_0 whose non-zero entries are those used for classification and are therefore of medical relevance. That means, if ω_0 is sparse and leads to good classification accuracy then only a few entries contribute and medical interpretation becomes feasible. However, if there does not exists a sparse ω_0 such that Eq. 1 holds, there is strong evidence that no sparse (simple) characterization is possible. This indicates that the underlying biological mechanisms are too complex to be captured by a sparse (simple) model. If this is the case, every sparsity-encouraging method will fail, meaning that a sparse classifier will always give poor classification. As a consequence, an important assumption of this work is that a sparse ω_0 (ground-truth) exists.

As we will see later it is often possible to find *non*-sparse classifiers which achieve better classification accuracy. This might be favorable in some situations in which the main focus is indeed on overall classification accuracy. However, in these situations overfitting becomes an issue and the identification of interpretable, highly-discriminative features might be extremely difficult. In the context of MS-data analysis such a classifier would be especially hard to interpret because of the very high dimensionality of the data.

State of the art in sparse feature selection

There are numerous approaches for feature selection which mainly fall into three categories:

- **Filters**: Using some score or correlation function (e.g., based on Fisher's, t-test, information theoretic criteria) evaluating the importance of each feature in a *univariate* way and taking the top-rated features.
- **Wrappers**: Using machine-learning algorithms to evaluate and choose features using some search strategy (e.g. simulated annealing or genetic algorithms).
- **Embedded methods**: Selecting variables by directly optimizing an objective function (usually in a multivariate way) with respect to: goodness-of-fit and (optionally) number of features. This could be achieved with algorithms like least-square regression, support vector machines (SVM), or decision trees.

In this paper, we will mainly focus on *embedded methods*. Regarding this category, the literature contains several well-known options for choosing combinations of loss and regularization functions (cf. (2)), some of which are exemplarily listed in Table 1.

Table 1 Prominent options for choosing loss function and regularizer in feature extraction algorithms

Name	Loss function (L)	Regularizer (R)
AIC/BIC	$\|y - \langle\omega, x\rangle\|_2$	$\|\omega\|_0$
Lasso	$\|y - \langle\omega, x\rangle\|_2$	$\|\omega\|_1$
Elastic Net	$\|y - \langle\omega, x\rangle\|_2$	$\|\omega\|_2^2 + \|\omega\|_1$
Regularized Least Absolute Deviations Regression	$\|y - \langle\omega, x\rangle\|_1$	$\|\omega\|_1$
Classic SVM	$\max(0, 1 - y\langle\omega, x\rangle)$[a]	$\frac{1}{2}\|\omega\|_2^2$
ℓ_1-SVM	$\max(0, 1 - y\langle\omega, x\rangle)$[a]	$\frac{1}{2}\|\omega\|_1$
Logistic Regression	$\log(1 + \exp(-y\langle\omega, x\rangle))$	$\frac{1}{2}\|\omega\|_1$

[a]This is the so called *Hinge loss*

The ℓ_1- and ℓ_2-norm of a vector $z = (z_1, \ldots, z_d) \in \mathbb{R}^d$ are defined by $\|z\|_1 = \sum_{j=1}^d |z_i|$ and $\|z\|_2 = (\sum_{j=1}^d |z_i|^2)^{1/2}$, respectively. The "$\ell_0$-norm" $\|z\|_0$, simply counts the number of non-zero entries of z

Different combinations can influence the results dramatically: Fig. 2 demonstrates the effect of sparsity by comparing a ℓ_2- and ℓ_1-regularized version.

In this example, a proteomics data-set was created that contains three discriminant features between the two subgroups. It can be easily seen how the results differ: While the ℓ_1-based result is optimized for selecting only a few features, the ℓ_2-variant selects much more features which in turn results in a better fit of the observation model. In this paper, we are interested in developing a method that selects as few features as possible while achieving the best possible fit under this constraint. This is in contrast to methods that aim at only achieving the best possible fit. A low-complexity model is of particular interest in biological applications because each selected feature is usually analyzed in subsequent experiments, which creates additional costs.

Various approaches can be used to assess the outcome ω of a feature selection method, when appropriate training and test data are available. We will use the following three measures of quality: (i) correctness of the selected features, (ii) size of the selected feature set, (iii) performance of classifying an unknown test set (specificity, sensitivity, accuracy). Obviously, (i) can only be used if the correct features are known, which is the case in our benchmark data-sets (for more details see "Feature selection from simulated data-sets" section).

Contribution

As already mentioned above, the major challenge of sparse feature extraction is to robustly identify a *small* set of variables (non-zero components of ω) that can be used to accurately classify unknown proteomics data (e.g. healthy or diseased) according to (1). This paper introduces *Sparse Proteomics Analysis* (*SPA*), a novel framework for feature selection and classification. The key step of our method is based on *1-bit compressed sensing* (cf. "Compressed sensing-based data analysis" section) and solves the following optimization problem:[6]

$$\max_{\omega\in\mathbb{R}^d} \sum_{i=1}^n y_i\langle x_i, \omega\rangle \quad \text{subject to} \quad \|\omega\|_1 \le \sqrt{\lambda} \text{ and } \|\omega\|_2 \le 1, \tag{3}$$

where the regularization is now defined by two inequality constraints on the feature vector ω.[7] The above approach is motivated by the general theory of *compressed sensing*, which was originally introduced by Donoho as well as by Candès, Romberg, and Tao (cf. [17–19]) and provides a modern framework for efficiently acquiring and processing high-dimensional (nearly) sparse signals (for more details see "Compressed sensing-based data analysis" section).

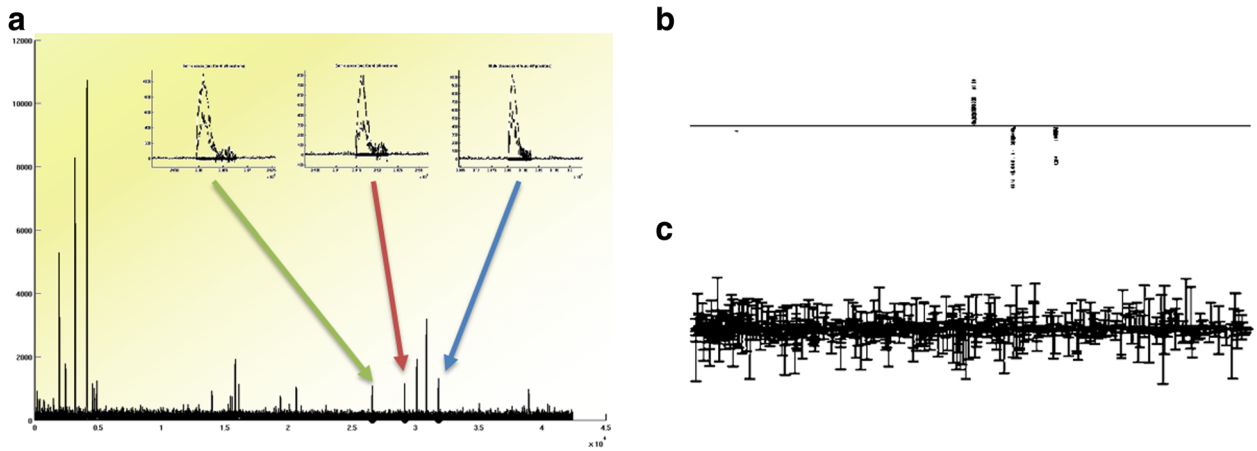

Fig. 2 a Overlaid spectra from two different groups. The three peaks marked by the *arrows* (magnified in the inlays) represent the underlying differences between the two groups. **b** Sparse ω found by a ℓ_1-regularized method (ℓ_1-SVM). **c** ω found by ℓ_2-regularized method (classical SVM)

We shall verify the competitiveness of our method by applying it to several synthetic and real-world data-sets and comparing the results to those of other widely-used algorithms in this field. Although the core of the algorithm (3) is surprisingly simple, we will observe that SPA (including pre- and postprocessing steps) finds optimal feature vectors which are extremely sparse, allow for highly accurate classification, and are robust against noise. In particular, for "very-sparse" situations, it even turns out that SPA outperforms the standard methods listed in Table 1.

Note that computational solutions to (2) or (3) are usually based on solving a convex program by standard optimization techniques, such as interior point methods. However, these methods sometimes scale poorly with increasing number of samples n and data dimension d, as it is typically the case for -omics data analysis. Several strategies have been proposed in the literature to speed up the calculations, e.g., using stochastic decent ([20–24]). In this article, we shall not focus on such computational issues but rather on providing a novel way of formalizing and solving the feature selection problem, namely in the context of compressed sensing.

Apart from the specific approach of (3), it is a general concern of this work to promote the benefit of *sparse* embedded methods. In contrast to classical (univariate) approaches, such as statistical tests, the process of variable selection takes place in an automatic fashion here. In this way, a costly preprocessing (e.g., peak detection) as well as subsequent feature assessments can be avoided as much as possible. Especially in a situation where only a very few samples are available, those additional steps may cause further instability and their success strongly relies on the specific data structure. In fact, it was already succinctly emphasized by Vapnik in ([25], p. 12) that *"If you possess a restricted amount of information for solving some problem, try to solve the problem directly and never solve the more general problem as an intermediate step. It is possible that the available information is sufficient for a direct solution but is insufficient for solving a more general intermediate problem."* This fundamental principle is precisely reflected by our viewpoint, which only makes a few (generic) assumptions on the underlying data model. Finally, we would like to mention that recently, rigorous theoretical guarantees for sparse feature selection from MS data were shown in [26]. Using the novel idea of *optimal problem representations*, the mathematical framework of [26] even goes beyond the binary output scheme of (1) and allows for a unified treatment of general observation and data models.

The next sections shortly review the background of *compressed sensing* and then describe our novel feature selection approach SPA in detail ("Methods" section). Finally, we present several benchmark results in "Feature selection from simulated data-sets" sections and "Results for real-world MALDI-TOF MS data" for simulated and real data-sets and compare them to current state-of-the-art algorithms.

Compressed sensing-based data analysis

In its most simple form, *compressed sensing* (*CS*) studies the recovery of an unknown vector $x \in \mathbb{R}^d$ from *linear measurements* $y = Ax$. Here, $A \in \mathbb{R}^{n\times d}$ is an $(n \times d)$-matrix and the entries of $y \in \mathbb{R}^n$ contain the measurements. The major challenge is now to design the measurement process A in such a way that the number of measurements n is as small as possible and, at the same time, x is still (uniquely) recoverable from y. Thus, we are asking for the maximal *compressibility* of x by linear measurements.

Obviously, when $n \ll d$, we require some additional information to obtain a unique solution of $y = Ax$. The prior information on x which is studied in compressed sensing is the assumption of *sparsity*, i.e., most coefficients of x are assumed to be zero, or at least very small. One naive approach to incorporate this additional property is to search for the sparsest solution of $Az = y$:[8]

$$\arg\min_{z\in\mathbb{R}^d} \|z\|_0 \quad \text{subject to} \quad Az = y. \tag{4}$$

Unfortunately, this problem is non-convex and cannot be efficiently solved in general. Therefore, one usually replaces (4) by its *convex relaxation*, which is also known as the *basis pursuit* ([27]):

$$\arg\min_{z\in\mathbb{R}^d} \|z\|_1 \quad \text{subject to} \quad Az = y. \tag{5}$$

One of the first key results in compressed sensing states that, if $A \in \mathbb{R}^{n\times d}$ is chosen *randomly*, e.g., with independent and identically distributed Gaussian entries, and $n = O(s \cdot \log(d/s))$, then (with "high probability") every s-sparse vector x (i.e., $\|x\|_0 \leq s$) can be uniquely recovered from (5). The most surprising fact is that the number of required measurements $n = O(s \cdot \log(d/s))$ only logarithmically depends on the (possibly large) dimension d of the ambient space. Hence, random measurement processes indeed allow for a very strong compression of sparse vectors (see also [17–19] for more details).

In order to consider more complicated situations, the stability and robustness of the basis pursuit algorithm was extensively studied. Various theoretical results and numerical experiments show that this algorithmic approach can also be applied for the stable recovery of vectors which are only nearly sparse, as well as to noisy measurements of the form $y = Ax + \eta$. To obtain a robust version of (5), one may replace its equality constraint by $\|Az - y\|_2 \leq \epsilon$ for some appropriate noise level $\epsilon > 0$. Not very surprisingly, this approach is also closely related to the Lasso introduced by Tibshirani in [28] (see also (2) and Table 1).

1-bit compressed sensing

In many practical scenarios, especially when working with computers, there is no way to represent real numbers exactly. Thus, it is reasonable to assume that the measurement vector Ax is acquired in a *quantized* (and therefore non-linear) fashion. The most extreme form directly leads to *1-bit measurements*, i.e., only the signs of Ax are known:

$$y_i = \operatorname{sign}(\langle a_i, x\rangle), \qquad i = 1, \dots, n, \tag{6}$$

where $a_1, \dots, a_n \in \mathbb{R}^d$ are the rows of the measurement matrix $A \in \mathbb{R}^{n\times d}$. As in classical compressed sensing, we are asking for an appropriate recovery of x from (6) using as few measurements as possible. This challenge was originally considered in [29] as *1-bit compressed sensing*, and has been extensively studied in [30, 31].

A surprisingly simple convex recovery approach was proposed by Plan and Vershynin in [31]:

$$\max_{z\in\mathbb{R}^d} \sum_{i=1}^{n} y_i \langle a_i, z\rangle \quad \text{subject to } \|z\|_1 \leq \sqrt{\lambda} \text{ and } \|z\|_2 \leq 1, \tag{7}$$

where $\lambda > 0$ denotes the sparsity-controlling parameter. To get some intuition, we first note that we have $y_i = \operatorname{sign}(\langle a_i, x\rangle)$ if and only if $y_i\langle a_i, x\rangle > 0$ holds. Hence, maximizing the sum in (7) will ensure the consistency of many measurements $i \in \{1, \dots, n\}$, according to (6). However, the total consistency is not enforced so that (7) indeed allows for noisy inputs y that do not satisfy (6). On the other hand, the constraint of (7) promotes sparsity of the final outcome. To see this, we may consider the set $S_{d,\lambda} := \{z \in \mathbb{R}^d : \|z\|_0 \leq \lambda, \|z\|_2 \leq 1\}$ and observe that (cf. [31] Sec. III)[9]

$$\operatorname{conv}(S_{d,\lambda}) \subset \{z \in \mathbb{R}^d : \|z\|_1 \leq \sqrt{\lambda}, \|z\|_2 \leq 1\} \subset 2\operatorname{conv}(S_{d,\lambda}).$$

This means that (7) optimizes over a convex relaxation of the set $S_{d,\lambda}$ which contains all λ-sparse vectors in the unit ball. For more details, see also [31]. The main statement of [31] proves that the robust 1-bit compressed sensing algorithm (7) indeed allows for an appropriate recovery of sparse vectors, using only $n = O(\lambda \cdot \log(d/\lambda))$ measurements. Moreover, it is surprisingly robust against several types of noise, including (random) bit-flips of the labels.

Remark *The minimized functional of* (7) *is closely related to the hinge loss which is used for SVMs (cf. Table* 1*). Indeed, without rejecting the negative part of the hinge loss, we would precisely end up with the objective functional in* (7)*.*

The constraint of (7)*, on the other hand, can be regarded as a combined ℓ_1-ℓ_2-condition, where the tuning parameter λ controls the desired level of sparsity of the minimizer. This type of regularization strongly resembles the idea of elastic nets, originally proposed by Zou and Hastie in* [32]*.*

Why compressed sensing?

At a first sight, the main challenges of compressed sensing and machine learning (ML) seem to be very different. In compressed sensing, we intend to design a measurement process A in order to *compress* a vector x, whereas in machine learning, the training data is already contained in the rows of A and we are rather willing to *explain* the observations y by some appropriate vector x. However, in both areas we are asking for a (sparse) recovery from a certain type of measurement. Indeed, a *linear regression* in

ML exactly corresponds to classical CS model (see "Compressed sensing-based data analysis" subsection), and a *classification* problem is actually equivalent to 1-bit CS (see "1-bit compressed sensing" subsection).

Therefore, it is not very surprising that the applied algorithms for compressed sensing and machine learning resemble each other, and that theoretical results in both fields rely on the same mathematical foundations (concentration of measure, convex geometry, etc.). Unfortunately, both communities only rarely interacted with each other. In this paper, we would like to emphasize the viewpoint of compressed sensing, in particular, because it is still not very common for the classification tasks that we deal with.

With the recent progress in compressed sensing and related areas as low-rank matrix recovery or quantized CS, also new algorithms like nuclear norm minimization or 1-bit CS have been proposed. Although these methods are typically motivated by theoretical studies, they perform also very well for real-world data. In general, we believe that these alternative perspectives allow for deeper theoretical insights, finally leading to the improvement of the classical (ℓ_1-based) tools from machine learning.

For an extensive introduction to compressed sensing, we refer to [33, 34]. As we already mentioned above, comparing this text to literature from statistical learning theory (see [35] for example), the reader will quickly notice many interesting connections between both fields.

Methods

In this section, we present the details of our novel framework which we call *Sparse Proteomics Analysis (SPA)*. It is based on the ideas of 1-bit compressed sensing presented in the previous section. The first part provides a mathematical formulation of the feature selection problem as well as a brief overview of the steps that are performed in SPA. The rest of this section is then devoted to a detailed description and discussion of the single steps.

Setting and overview

As already mentioned in the introduction, we assume that our learning process is *supervised*, i.e., we know which spectrum belongs to the class of healthy ($y_i = +1$) and diseased ($y_i = -1$) samples in advance. If the data vectors $x_i \in \mathbb{R}^d$, $i = 1, \dots, n$ are mass spectra, the indices $j = 1, \dots, d$ of $x_i = (x_{i,1}, \dots, x_{i,d})$ correspond to the m/z-values[10] and its entries represent the intensities. The nonzero entries of the feature vector $\omega_0 = (\omega_{0,1}, \dots, \omega_{0,d}) \in \mathbb{R}^d$ describe the location of the disease fingerprints and its respective values the significance of these features.

In the setting of classical learning theory, we are asking for a hyperplane $\{\omega_0\}^\perp$ which correctly separates most of the data points x_i labeled by y_i. More precisely, this means[11]

$$y_i = \operatorname{sign}\left(\langle x_i, \omega_0 \rangle\right) \quad \text{for "many" samples } \; i = 1, \dots, n. \tag{8}$$

Equivalently, we can view (8) as a problem from 1-bit compressed sensing (cf. "Why compressed sensing?" section), i.e., we have acquired noisy 1-bit measurements and are now looking for a sparse recovery.

In the development of SPA, we have primarily focused on the latter perspective, and therefore, the 1-bit recovery program (7) forms the key step of our algorithm:

Algorithm 1 (SPA at a glance)**.**
Input: Raw data samples $\{(x_i, y_i)\}_{i=1,\dots,n}$
Output: Sparse feature vector $\tilde{\omega} \in \mathbb{R}^d$ **Preprocessing:**

1: Normalize data to make the spectra comparable.
2: Perform smoothing by a convolution with Gaussian density.
3: Standardize data.

Sparse Feature Selection:

4: Perform 1-bit CS optimization (7) to find feature vector $\hat{\omega}$.

Postprocessing:

5: Detect the connected components of $\hat{\omega}$ to obtain a sparsified version $\tilde{\omega}$.
6: (Optional) Reduce dimension by projecting data onto the feature space.

Algorithmic details

In the following, we are going to specify and discuss the single steps of Algorithm 1.

Step 1: normalization of the data

This step heavily depends on the underlying acquisition method of the data. Every spectrum $x_i \in \mathbb{R}^d$ is normalized by a certain scaling factor $\mu_i > 0$, i.e., $x_i \mapsto \mu_i x_i$ for $i = 1, \dots, n$. The individual scalars μ_i should be chosen such that the resulting data vectors are "comparable."

For example, when we assume that the data are acquired by MALDI-TOF-MS as described in Fig. 1, it seems to be quite natural to normalize them by the total ion count. Mathematically, this means that we would divide every spectrum by its ℓ_1-norm, i.e., we choose $\mu_i = 1/\|x_i\|_1$.

Step 2: smoothing by gaussian density

We already pointed out that one major challenge is the strong noise within the raw data. Therefore, it is crucial to perform some noise reduction before trying to extract features. For this purpose, we suggest a simple smoothing strategy by a Gaussian density:

Let G_σ denote the (centered) *Gaussian density function* with fixed standard deviation $\sigma > 0$, that is,

$$G_\sigma(t) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{t^2}{2\sigma^2}\right), \quad t \in \mathbb{R}.$$

The smoothed spectra $\tilde{x}_i = (\tilde{x}_{i,1}, \ldots, \tilde{x}_{i,d}) \in \mathbb{R}^d$ are then obtained by a discrete convolution

$$\tilde{x}_{i,k} := (x_i * G_\sigma)_k = \sum_{l=1}^{d} x_{i,l} \cdot G_\sigma(k-l), \qquad k = 1, \ldots, d, \quad i = 1, \ldots, n. \tag{9}$$

Using the fast Fourier transform (FFT), this computation can be performed quickly with $O(nd \log(d))$ operations. In a very simplified scenario, a spectrum can be written as the sum of Gaussian-shaped peaks plus some baseline noise in each mass channel. Since the convolution of two Gaussian densities is again Gaussian, the original (local) structure of the spectra is essentially preserved in $\tilde{x}_i$, whereas the noise of x_i is significantly reduced. Note that the deviation $\sigma > 0$ serves as a tuning parameter of the algorithm. A good choice of σ clearly depends on the nature of the data; usually it depends on the noise level as well as on the (average) width of the peaks.

Finally, we would like to emphasize another interesting interpretation of the above smoothing approach: The convolution in (9) can be written as a scalar product of x_i with the shifted Gaussian density $G_\sigma(\cdot - k)$ (note that G_σ is symmetric), that is, $\tilde{x}_{i,k} = \langle x_i, G_\sigma(\cdot - k)\rangle$. Thus, the entries of $\tilde{x}_i$ are actually the *analysis coefficients* of the *Gaussian dictionary* $\{G_\sigma(\cdot - k) \mid k = 1, \ldots, d\}$. The perspective of analyzing data by a *dictionary* offers several opportunities for generalization. For instance, one could also consider (redundant) dictionaries with more than one standard deviation or more sophisticated functions than G_σ.

Step 3: standardizing the data

The 1-bit optimization of (7) does not incorporate a bias term. Hence, it is necessary to center the data first. For this, we compute the *mean spectrum*[12]

$$\bar{x} := \frac{1}{n} \sum_{i=1}^{n} x_i \in \mathbb{R}^d,$$

i.e., $\bar{x}_k$ contains the average of the k-th entry of all spectra.

The spectra are further scaled by dividing the non-constant features by their *standard deviation*

$$\sigma_j := \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left(x_{i,j} - \bar{x}_j\right)^2}, \quad j = 1, \ldots, d.$$

The *standardized spectra* $\check{x}_i = (\check{x}_{i,1}, \ldots, \check{x}_{i,d}) \in \mathbb{R}^d$ are then obtained by

$$\check{x}_{i,j} := \frac{x_{i,j} - \bar{x}_j}{\sigma_j}, \quad i = 1, \ldots, n, \quad j = 1, \ldots, d.$$

In this way, all feature variables are centered and have an empirical standard deviation equal to 1, so that they get equally weighted in the selection process.

Step 4: sparse feature selection

We are now ready to perform the actual feature extraction step, using the 1-bit recovery method presented in "1-bit compressed sensing" subsection:

Algorithm 2 (1-Bit Compressed Sensing)**.**
Input: Samples $\{(x_i, y_i)\}_{i=1,\ldots,n}$, sparsity parameter $\lambda > 0$, threshold $\epsilon > 0$
Output: Estimated feature vector $\hat{\omega} = (\hat{\omega}_1, \ldots, \hat{\omega}_d) \in \mathbb{R}^d$
Compute:

$$1: \hat{\omega}' = \arg\max_{\omega \in \mathbb{R}^d} \sum_{i=1}^{n} y_i \langle x_i, \omega \rangle \quad \text{subject to } \|\omega\|_1 \le \sqrt{\lambda} \text{ and } \|\omega\|_2 \le 1. \tag{10}$$

$$2: \hat{\omega}_k = \begin{cases} \hat{\omega}'_k, & \text{if } |\hat{\omega}'_k| > \epsilon, \\ 0, & \text{otherwise,} \end{cases} \qquad k = 1, \ldots, d. \tag{11}$$

The second part (in (11)) is a simple hard thresholding that tries to eliminate computational inaccuracies by setting almost zero entries of $\hat{\omega}'$ to 0 (ϵ is usually very small, e.g., $\sim 10^{-3}$).

The actual feature selection takes place in (10). Recalling the observation model from (8), we conclude that the i-th sample is correctly classified by a vector ω if and only if $y_i \langle x_i, \omega \rangle > 0$. Hence, the objective functional of (10) will be particularly large if sufficiently many samples are correctly classified by ω. However, a consistent prediction of *all* measurements (i.e., $y_i = \text{sign}(\langle x_i, \omega \rangle)$ for all $i = 1, \ldots, n$) is not strictly enforced, and therefore, our strategy enjoys a certain robustness against (random) perturbation of the model (8). This could occur in practice, for example, when a training sample was wrongly classified from the very beginning. On the other hand, the constraint of (10) guarantees that the maximizer will be "effectively" sparse (depending on the choice of the sparsity parameter $\lambda > 0$). This intuition indicates that the estimator $\hat{\omega}$ will be indeed a sparse vector allowing for an appropriate separation of the two classes.

Step 5: detecting the connected components

One advantage of Algorithm 2 is that it does not make any assumptions on the structure of the data vectors x_i. Hence, it might be even suited for much more general types of data. However, its "universality" comes with the drawback that the characteristic peak structure of MS data is not captured at all. In fact, a spectrum does not

consist of sharp spikes but rather wide-spread Gaussian shaped peaks. Hence, if the algorithm finds a significant feature position, say at the maximum of some peak, it usually tends to select also those features which are close to this position. Such a behavior is not very surprising, because nearby features are highly correlated to the maximum of the peak, and therefore, they may allow for a good separation as well.

Empirical results have shown that this process of selection "evolves" in a continuous fashion when changing the sparsity level λ. As a consequence, the support of a feature vector $\hat{\omega}$ from Algorithm 2 typically consists of a few connected "intervals" (consecutive sequences of indices) which are centered around the selected peaks (see also Fig. 3). The actual sparsity of $\hat{\omega}$ should be therefore measured by means of its connected intervals and not by simply counting its non-zero entries.

For this reason, we may easily improve the sparsity of $\hat{\omega}$ by reducing every interval to its most significant entry:[13]

Algorithm 3 (Sparsification of $\hat{\omega}$)**.**
Input: (Sparse) feature vector $\hat{\omega} = (\hat{\omega}_1, \ldots, \hat{\omega}_d) \in \mathbb{R}^d$
Output: Sparsified version $\tilde{\omega} = (\tilde{\omega}_1, \ldots, \tilde{\omega}_d) \in \mathbb{R}^d$
Compute:

1: Find the connected components $A_1, \ldots, A_N \subset \mathrm{supp}(\hat{\omega})$ of $\hat{\omega}$.
2: For every $l = 1, \ldots, N$ do the following: Set all entries of $\hat{\omega}$ in A_l to 0, except from $\arg\max_{k \in A_l} |\hat{\omega}_k|$.
3: The resulting vector is $\tilde{\omega}$.

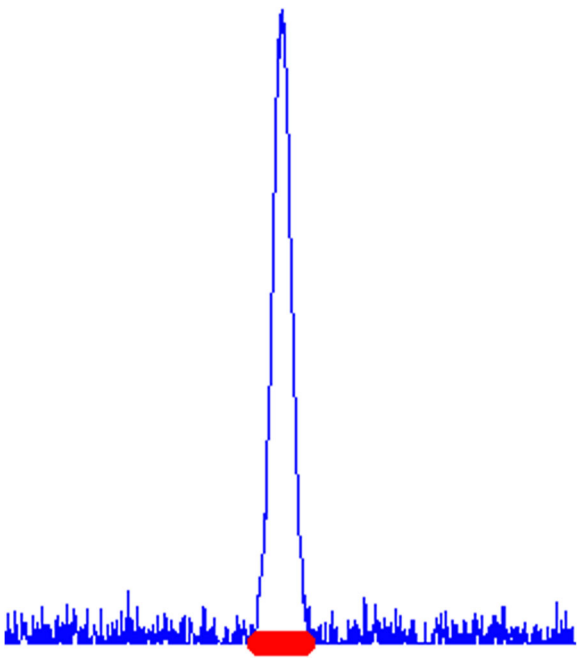

Fig. 3 The *red* stripe indicates the support of $\hat{\omega}$. Relevant features usually occur as intervals and not as isolated points

Step 6: dimension reduction

This final (optional) step does not involve any further computations but shows how to proceed with our result $\tilde{\omega}$. As mentioned before, the main purpose of SPA is not just to classify (unknown) samples, but rather to reduce the data to its significant entries (dimensions). Indeed, we may use $\tilde{\omega}$ for a *dimension reduction*: Let $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ be some (possibly unknown) data vector. Then, we can project x onto the selected feature positions of $\mathrm{supp}(\tilde{\omega})$. More precisely, all entries that do not belong to $\mathrm{supp}(\tilde{\omega})$ are set to 0:

$$\hat{x}_k := \begin{cases} x_k, & k \in \mathrm{supp}(\tilde{\omega}), \\ 0, & \text{otherwise}, \end{cases} \qquad k = 1, \ldots, d. \tag{12}$$

The resulting data vector $\hat{x} = (\hat{x}_1, \ldots, \hat{x}_d) \in \mathbb{R}^d$ is now trivially embedded into a low-dimensional space of dimension $\#\mathrm{supp}(\tilde{\omega})$.[14] But it still contains the most important information which has been found by the above algorithm. Note that we have not made any use of the actual values of $\tilde{\omega}$ but merely of its support.

By this projection, we may reduce the danger of overfitting. In particular, by working in a low-dimensional space, a large tool set from *machine learning* is now available for classification and clustering. But how to explicitly proceed with the data heavily depends on the specific application and is therefore not part of SPA.

Results and discussion

Feature selection from simulated data-sets

In this section, we assess our framework of SPA with regard to a typical situation in mass-spectrometry analysis: We would like to extract discriminating features from MS data with respect to two groups (e.g., healthy and diseased patients). A major difficulty is usually that only a small number of measurements (observations) is available. Building on this, we ask for the following: Given a simulated data-set for which the position and number of discriminating peaks are known (this will be called ω_0 below), how many samples are needed to identify these features with high accuracy?

We shall compare our results to the widely used state-of-the-art algorithms LIBLINEAR (ℓ_1-regularized SVM) and the standard MATLAB implementation of Lasso.

Creating a simulated data-set

We assume that our sample set $\{(x_i, y_i)\}_{i=1,\ldots,n} \subset \mathbb{R}^d \times \{-1, +1\}$ follows a certain joint random distribution (X, Y), where each sample is independently drawn. In order to make the problem tractable, let us make two model assumptions on X and Y. First, the mass spectra X are generated as follows:

$$x_i = \sum_{m=1}^{M} s_i^m a^m + n_i, \quad i = 1, \ldots, n,$$

where $s_i^m \in \mathbb{R}^d$ determines the (random) amplitude of the m-th peak, $a^m \in \mathbb{R}^d$ specifies its position and shape, and $n_i \in \mathbb{R}$ represents the low-amplitude baseline noise. We shall assume that the amplitudes and the noise are Gaussian, that is, $s_i := (s_i^1, ..., s_i^M) \sim \mathcal{N}(0, \Sigma)$ with $\Sigma \in \mathbb{R}^{M \times M}$ positive definite and $n_i \sim \mathcal{N}(0, \sigma^2 I)$ with $\sigma > 0$. Note that the generated data might have negative components. This does not mimic the structure of real-world mass spectra which is always non-negative. However, since centering is part of our preprocessing anyway (cf. Step 3 in "Algorithmic details" subsection), the assumption of mean-zero amplitudes is quite natural. The (disease) labels Y are then simply modeled as 1-bit observations (see also (8))

$$y_i = \text{sign}(\langle x_i, \omega_0 \rangle), \quad i = 1, \ldots, n, \tag{13}$$

where $\omega_0 \in \mathbb{R}^d$ is the sparse ground-truth feature vector, which we intend to estimate. In the following, each non-zero entry of ω_0 is located at the center of a specific peak (see Fig. 4(d)–(f)), so that $\text{supp}(\omega_0)$ actually determines all biologically relevant peaks (molecular structures). Since Σ is invertible (i.e., the features are linearly independent), this collection of peaks is an optimal fingerprint in the sense that removing or adding any feature variable would decrease the prediction accuracy (with respect to the "perfect" model of (13)).

In our experiments, we create data-sets $x_1, \ldots, x_n \in \mathbb{R}^{8192}$, each one consisting of 200 equidistant peaks (atoms a^m) shaped like Gaussian density function of width 10. The vector $\omega_0 \in \mathbb{R}^{8192}$ is chosen to have five non-zero components, which means that only five prechosen peaks were used to generate the labels $y_1, \ldots, y_n$. Hereafter, we will refer to these as *condition positive peaks*. Figure 4 shows three different data instances magnifying only the first seven peaks, generated in the described way. In order to verify our method, we will use two types of data-sets DS1 and DS2 which only differ in their correlation matrix Σ. For DS1, Σ is chosen to be the identity matrix. This implies that the heights of all of the 200 peaks are standard Gaussian random variables. For DS2, we have chosen three pairs of negative peaks to be positively correlated and in addition, one condition positive peak was chosen to be positively correlated with one of the negative peaks. Thus, there are a few entries of value 0.8 off the main diagonal in Σ. To test the algorithm's performance increasing amount of Gaussian noise $n_i \sim \mathcal{N}(0, \sigma^2)$ with $\sigma = \{0.1, 0.3\}$ was added to DS1 and DS2. These corresponds to signal-to-noise (SNR) ratio of 10, 3.33 repectively [15]. The values of SNR are chosen to represent the behaviour of the algorithm up to the levels of noise that are normally found in MS data.

Setup and evaluation criteria

Let us recall the essential question of our experiments: Can we recover the support of ω_0, and if so, how many

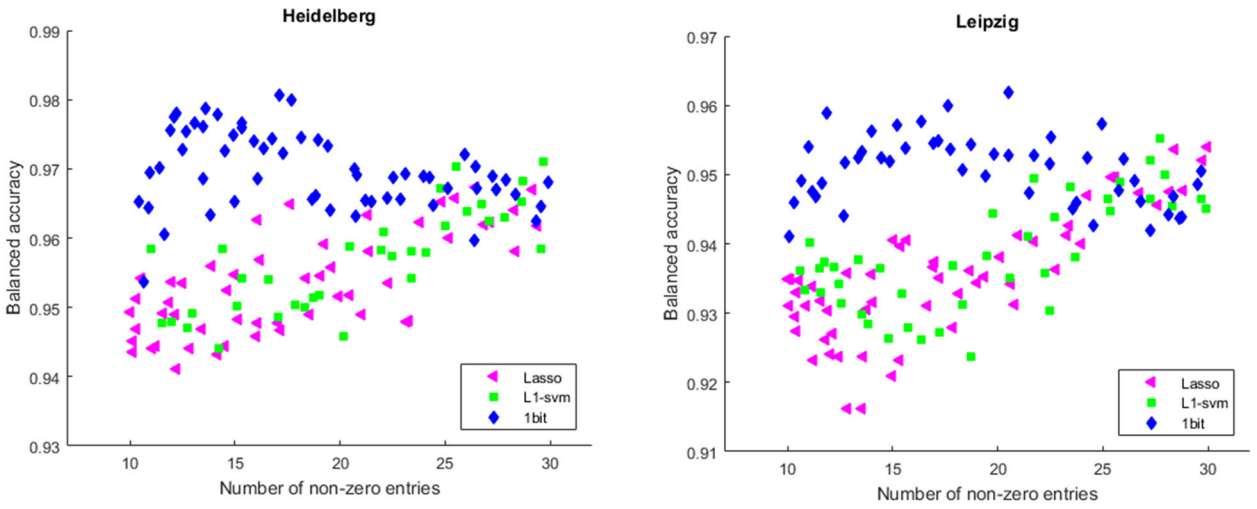

Fig. 4 Illustration of the generated data instances. **a**–**c**: First seven equidistant Gaussian peaks that are located in fixed positions in each of the three data instances; **d**–**f**: Visualization of the data instances from (**a**)–(**c**) with additive noise with standard deviation $\sigma = 0.1$, where the positions of the five condition positive peaks are highlighted by *black dots*. The *blue* and *red colors* indicate the different classes which are determined by the observation process of (13)

samples do we need for that? For this purpose, we shall successively increase the number of available samples in the (training) data-set and examine whether SPA (or Lasso, or ℓ_1-SVM) succeeds in recovering $\mathrm{supp}(\omega_0)$. Since each of the considered algorithms involves a variable parameter, we have decided to perform an adaptive tuning for each problem instance. In fact, the sparsity parameter was chosen such that the resulting classifier $\tilde{\omega}$ matches the sparsity level of ω_0. But of course, this does not automatically imply that the supports of $\tilde{\omega}$ and ω_0 completely coincide.[16] For each problem instance, the smallest sparsity parameter which resulted in a classifier with five non-zero entries was chosen in the following way: The initial value of the sparsity parameter for SPA and ℓ_1-SVM (Lasso) was set to the value which corresponds to the classifier with less than (more than) five non-zero values[17]. For SPA and ℓ_1-SVM (Lasso), the sparsity parameter was increased for a preset step size until the outcome had five or more (five or fewer) non-zero entries. If the previous step provided a sparse classifier with strictly more than (strictly less than) five non-zero entries, the bisection method was used on the interval between the two last sparsity parameter values. The bisection method was used until the optimal sparsity parameter was found or the difference between the two consecutive parameters became smaller than a preset tolerance.

We will use a measure based on *sensitivity*. Sensitivity, defined as[18]

$$\mathrm{sens} := \frac{TP}{TP+FN}$$

is an appropriate measure for our objectives because it represents an algorithm's ability to detect the relevant features. Note that ideally, the number of condition positives ($TP + FN$) is equal to predicted condition positives ($TP + FP$). In such a situation, the *precision*, given by $p := TP/(TP + FP)$ is equal to the sensitivity. However, in the presence of noise it is possible that the final selection encompasses several features which are associated with a single peak. This could lead to a precision value equal to 1 if all of the selected values are declared as true positives, though some other true features remain undetected. Since for us, it is equally important to penalize both false positives and false negatives, we have chosen the sensitivity to be the main point of reference. A measure of similar importance is the *specificity*, which is defined by

$$\mathrm{spec} := \frac{TN}{FP + TN}.$$

Finally, due to the possibly imbalanced number of relevant features, we shall also take into account the so-called *balanced accuracy*

$$\mathrm{bacc} := \frac{\mathrm{sens} + \mathrm{spec}}{2}.$$

Results for the simulated data-sets

Data-sets of sample sizes between 50 and 350 were generated as described above and each of the methods was performed for standardized input data. Note that the hard thresholding step described in (11) was also applied to the classifiers obtained from Lasso or ℓ_1-SVM. Otherwise, any computational inaccuracy would completely destroy the sparsity structure of the results.

For the sake of statistical stability, each experiment was repeated 10-times. The averaged results are presented in the Fig. 5. We can see that SPA (= 1-bit CS) performs better than the ℓ_1-SVM or Lasso with regard to the capability of recognizing the true positive features (sensitivity in Fig. 5). In our setting, if one method fails to select 5 condition positive peaks because one of them was selected twice, and the other method selects exactly the same 4 peaks and one false positive in addition, the specificity penalizes only the latter one. But effectively, both cases are suboptimal, since only the 5 positive peaks together can predict the class correctly. This effect is reflected by a smaller value of specificity of the 1-bit approach comparing to the specificity of other two methods for data-sets with less than 300 spectra (column 2 in Fig. 5). However, this also implies that SPA performs sligtly worse in rejecting true negatives than the other two approaches. The average results for balanced accuracy are visualized in the third column of Fig. 5. We observe that SPA outperforms the other two methods and even achieves 100% accuracy with relatively few observations. With further decreasing SNR the performance of the three algorithms becomes more similar. Figure 6 shows the numerical outcomes for the data-set DS2. The non-trivial correlation structure of DS2 eventually leads to a slight drop of sensitivity and accuracy for SPA (compared to DS1), whereas the performance of the other two methods essentially remains unaffected. As before with further decreasing SNR the performance of the three algorithms becomes more similar in terms of sensitivity and balanced accuracy.

Results for real-world MALDI-TOF MS data

In this section, we present results of SPA, Lasso, and ℓ_1-SVM for analyzing real-world mass-spectrometry data and compare them to the MALDIquant proteomics analysis workflow [36]. All data was acquired in our earlier studies [10, 37]. It was approved by the local ethics committees and fulfils the requirements of the Helsinki declaration. All subjects gave informed consent to participate in the study. We will demonstrate the performance of our method on two data-sets:

- *Spiked data*: The spiked data-set is a labelled ground-truth data-set containing *control* (e.g. healthy) and *case* (e.g. diseased) mass spectra where the true labels

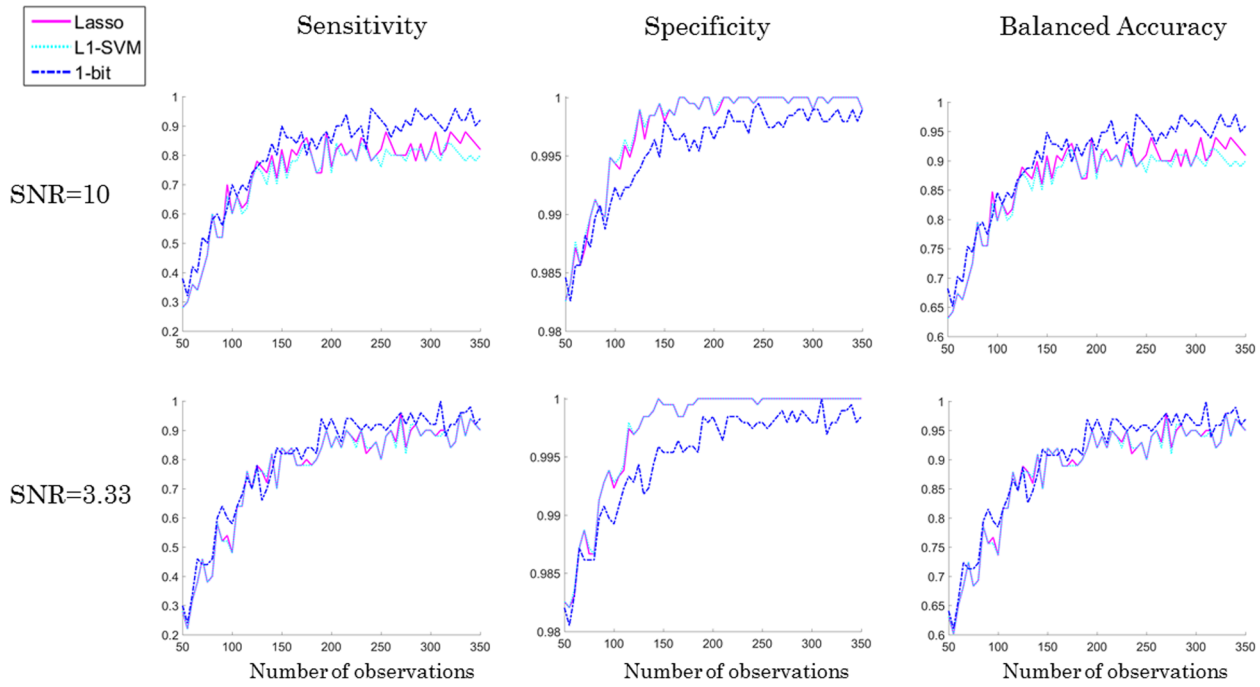

Fig. 5 Comparison of numerical results for SPA (= 1-bit CS), Lasso, and ℓ_1-SVM on the data-set DS1 with SNR = 10, and 3.33, showed in the respective row. Note that the data consist of 5 condition positive and 195 condition negative peaks which are equidistantly located in the spectra

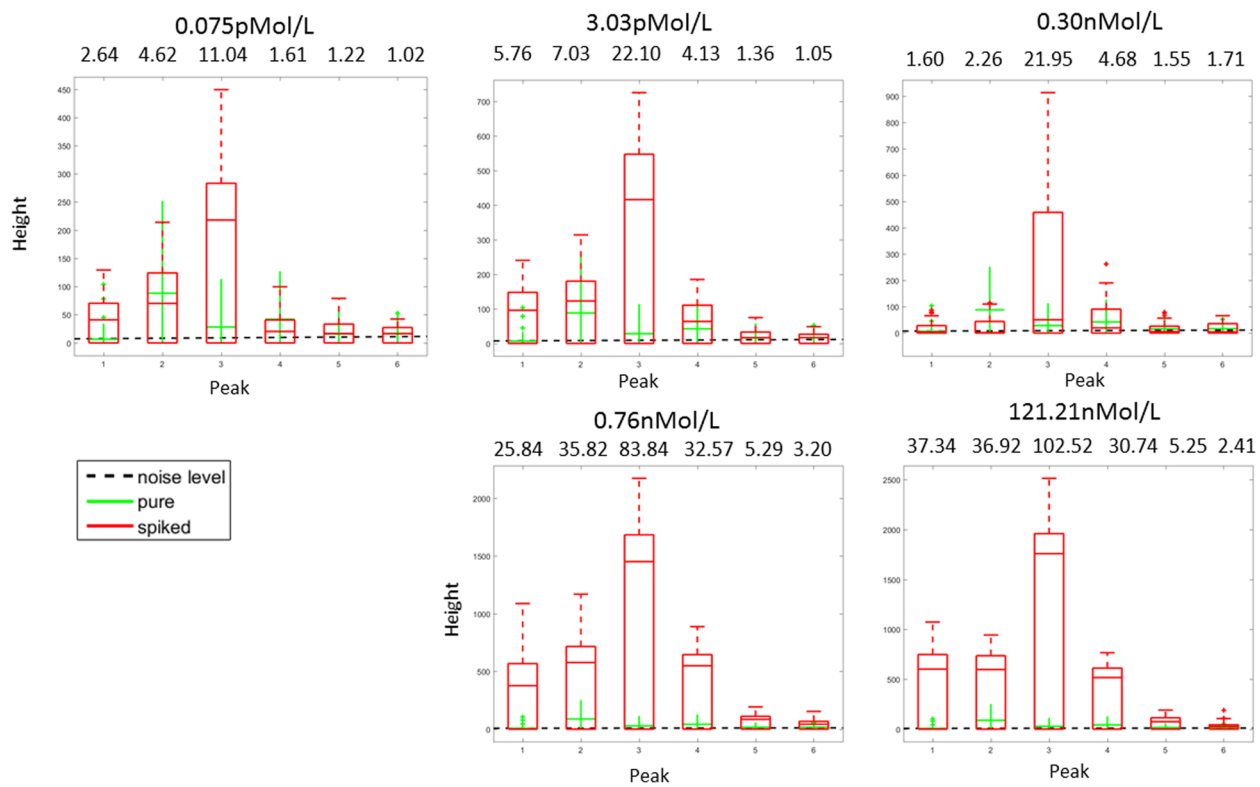

Fig. 6 Comparison of numerical results for SPA (= 1-bit CS), Lasso, and ℓ_1-SVM on the data-set DS2 with SNR = 10 and 3.33 showed in the respective row

are known. It is created from human blood samples[19] which were either unchanged (control group) or in which a protein-mix has been mixed (spiked) into (case group). In order to simulate different strength of an effect caused e.g. by a disease, we further sub-divided the *case* group into five sub-groups where the amount of spiked-in proteins is increasing. The five volumes in the case sub-groups were spiked with the following concentrations of the protein mix[20]: 0.075pMol/L, 3.03pMol/L, 0.30nMol/L, 0.76nMol/L and 121.21nMol/L. This mix contains the hormones Angiotensin, ACTH, clip 18-39, Substance P and the cell protein Ubiquitin. The peptide mix was added before sample pre-treatment and 64 spectra were measured due to 4-fold spotting (technical replicates). Mass spectra were acquired using the protocol described in the Additional file 1. Each volume corresponds to a data-set. What differentiates the data-sets are the amplitudes of the 6 spikes resulting from the added substances. The signal-to-noise ratio of the spiked-in peaks is shown in the Fig. 7[21].

- *Pancreas Cancer Data (P. CA)*: A total of 120 patients with pancreatic cancer and controls were recruited for this study [10]. For the discovery study sera were obtained from two different clinical centres (University Hospital Leipzig (UHL, set L) and Heidelberg (UHH, set H)) as described in the supplementary material (S1). Note that each acquired spectrum has been assigned a class-label, i.e., healthy or diseased. So, the health status of the training samples is known in advance (supervised learning).

Baseline removal was performed on the raw MS data using TopHat filtering ([38]). In particular, no additional calibration or noise reduction steps have been applied. More information on the data and sample preparation can be found in the supplementary material (S1).

The missing-data problem

When dealing with data coming from measurements of, say, a Mass Spectrometer instrument, the so called *missing-data problem* usually occurs. This means that the instrument failed to give measurements for some of the measured masses, usually due to the stochastic nature of the process happening inside the device. Due to the smoothing step in our algorithm and the arguments of e.g. Rubin et al. ([39]) this problem can be mainly ignored in our case for identifying the relevant features. However, this does not necessarily hold for the classification step, i.e. applying the identified sparse classifier to an unknown data-set. In this scenario, where data is missing in an unknown sample, there are basically two options: (1) applying a method for inferring the missing data or

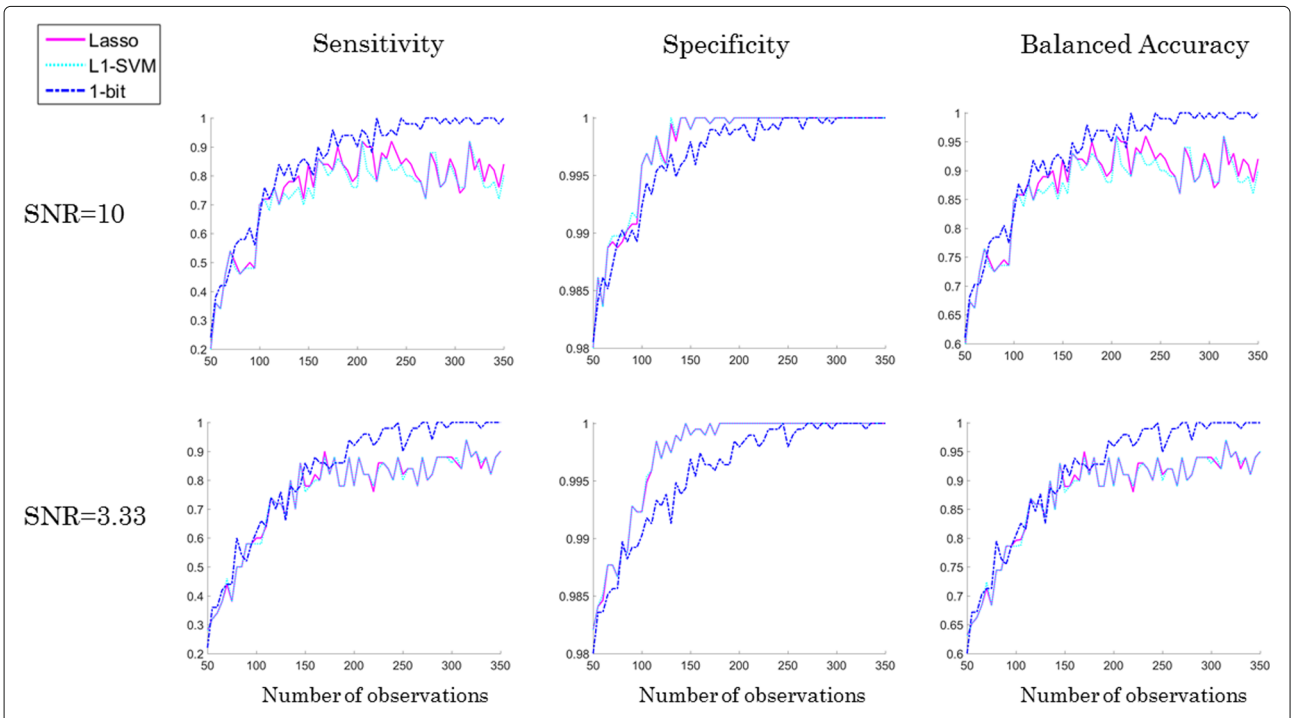

Fig. 7 The height of true signals (6 spiked in peaks) comparing to the height of noise and height of the corresponding values in the pure data-set. Signal-to-noise ratio, which was calculated as the ratio of median of spiked-in signals and the estimated level of noise is shown above the corresponding peaks

(2) stopping the classification and return an error message to the user. In this work we decided to follow the latter approach, since inferring missing data is not in the scope of this paper[22] but is an unarguable crucial point in any data analysis pipeline and should depend on the actual use-case.

Accuracy vs. number of features

We performed the *real world* experiments with respect to the same evaluation categories as in the case of simulated data. Note that the normalization and standardization as described in "Algorithmic details" section were applied as preprocessing steps in each of the methods. Similarly, a hard thresholding as described in (11) was applied to all classifiers estimated by the examined algorithms.

For the each of the algorithms, we are testing the performance of the obtained classifiers learned on the pure data-set which corresponds to the condition negative class and one spiked data-set at a time corresponding to the condition positive class.

The results of the classifier with 6 non-zeros on the spiked data-set are shown in Table 2. The main question in these experiments is how successful each of the algorithms is in detecting the 6 peaks that were initially spiked (see the data-set description above). We can see that the values of sensitivity for SPA are at least as high as those of the other methods, which implies that the approach of 1-bit CS is very competitive in this situation and mostly achieves the best detection rate. However, the relatively poor performance of all the algorithms on the spiked data-set can be explained by the nature of the data. Since the peptide mix was added to the blood samples before acquiring the mass spectra, the spiked peaks are not always present in all the resulting mass spectra in the positions where we expect to find them. There exist data-sets for which all the mass spectra failed to exhibit certain spiked peaks at their expected locations. as can be seen in the Fig. 7. Thus, we cannot expect any of the algorithms to find these missing peaks. Nonetheless, there is still a chance to build a reliable fingerprint out of the remaining spikes while there is no chance to detect the missing spikes because the data-set is not rich enough to represent it. On the other hand, this spiked data-set combines the advantages of both simulated and clinical data, since the positions of the desired biomarkers are known in advance while their representative behavior in the spectra is quite realistic.

In contrast to that in the case of pancreas cancer data-sets, we do not know the true-positive feature positions. Consequently, we can only rely on the classification performance of the obtained sparse classifiers by each of the algorithms. To evaluate the reliability of our results, for each of the methods, we have employed the cross-validation scheme as described in the Algorithm 4 with the number of folds K set to 5. In order to ensure statistical stability, each experiment was repeated 10-times. Figure 8 shows the average results over 10 repetitions.

Algorithm 4 (Cross-Validation of Classification Performance)**.**
Input: Raw data $(x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^d \times \{-1, +1\}$; Number of CV-folds K;
Output: Classification accuracy $Acc \in [0, 1]$; Average sparsity $\#F$ (number of selected features)
Compute:

1: Split the sample set $\{1, \ldots, n\}$ randomly into K disjoint folds $P_1, \ldots, P_K$ of (almost) equal size.
For each fold $k \in 1, \ldots, K$ *perform the following steps 2-4:*
2: Compute the feature vector ω^k employing the desired method on the samples of $\bigcup_{k' \in \{1,\ldots,K\} \setminus \{k\}} P_{k'}$
3: Dimension reduction as described in Algorithmic Details (step 6). Project all spectra onto $\text{supp}(\omega^k)$ and put $\#F^k := ||\omega^k||_0$.
4: Classification of P_k: Use the projected samples of $\bigcup_{k' \in \{1,\ldots,K\} \setminus \{k'\}} P_{k'}$ to predict the labels of the spectra

Table 2 This table shows the main results comparing the feature selection benchmarks of our approach with Lasso and ℓ_1-SVM on the spiked data-set. Given results correspond to the average results over 10 repetitions of the classifier with 6 non-zero values

	SPA				Lasso				ℓ_1-SVM			
Concentration	TP[a]	Sens[b]	Specs[c]	B. Acc[d]	TP	Sens	Spec	B. Acc	TP	Sens	Spec	B. Acc
0.075pMol/L	2	0.333	1.000	0.667	1	0.167	1.000	0.583	1	0.167	1.000	0.583
3.03pMol/L	4	0.667	1.000	0.833	2	0.333	1.000	0.667	1	0.167	1.000	0.583
0.30nMol/L	2	0.333	1.000	0.667	1	0.167	1.000	0.583	1	0.167	1.000	0.583
0.76nMol/L	2	0.333	1.000	0.667	2	0.333	1.000	0.667	2	0.333	1.000	0.667
121.21nMol/L	3	0.500	1.000	0.750	2	0.333	1.000	0.667	2	0.333	1.000	0.667

[a] TP: Number of spiked peaks that are correctly detected
[b] Sens: Sensitivity in detecting spiked peaks ($TP/(TP + FN)$)
[c] Spec: Specificity in detecting spiked peaks ($TN/(FP + TN)$)
[d] B. Acc: Balanced Accuracy ($\frac{\text{sens.}+\text{spec.}}{2}$)

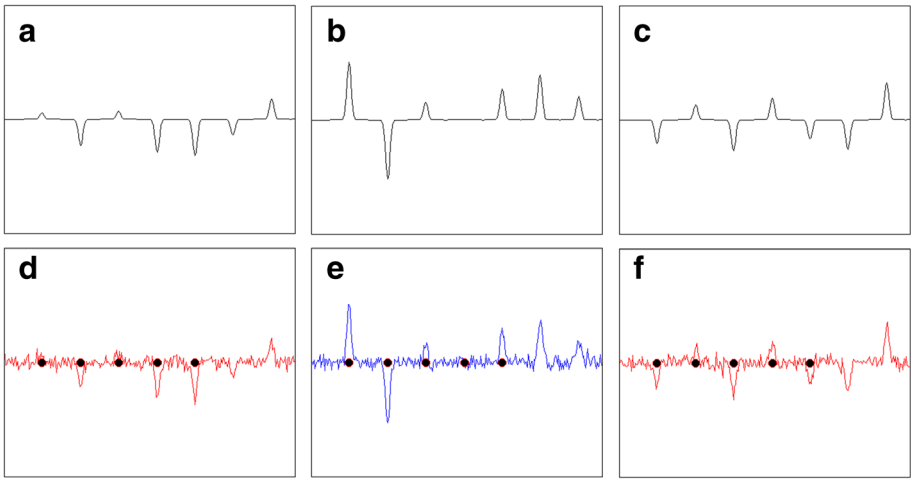

Fig. 8 Accuracies of sparse classifiers from SPA, Lasso, and ℓ_1-SVM on the real pancreatic cancer data-sets. While Lasso and ℓ_1-SVM achieve better classification accuracy with increasing number of features, SPA is particularly well suited for the "very-sparse regime" where only few features (< 20) are used for classification

P_k by an ordinary SVM[1]. Denote the prediction accuracy by Acc^k.

5: Compute the average accuracy $\text{Acc} := \frac{1}{K}\sum_{k=1}^{K} Acc^k$ and average sparsity $\#F := \frac{1}{K}\sum_{k=1}^{K} \#F^k$.

In order to ensure statistical stability, each experiment was repeated 10-times. Figure 8 shows the results. Note that our results show that accurate predictions are already possible with a very few features, so that the assumption of small disease fingerprint seems to hold for this data-set. Furthermore, it can be seen that SPA is especially well suited for situations where a sparse classifier (containing only very few features) is preferred. This is appealing because fewer features enable an easier interpretation of the actual components of a potential disease fingerprint. Moreover, follow-up experiments that often involve an individual treatment of each component (e.g., potential biomarkers) would become much less costly. Note that in the non-sparse region with more than 30 features selected, it is not meaningful to relate the achieved accuracy to the quality of the learned feature vector due to the small sample size. The considered algorithms assume the underlying fingerprint to be sparse. This assumption usually does not fully hold in practice. Therefore, we cannot expect that a learned feature vector achieves perfect classification. The classification accuracy should be therefore considered as an indicator of how well our model assumption of the sparse fingerprint fits to the unknown ground-truth. If we let the algorithms operate out of the region for which they have been designed for, we may achieve indeed a higher accuracy, but this is probably a consequence of overfitting. And even more importantly, the learned feature vector (model) is not reliable anymore.[23]

Best classifier

Apart from that, we are interested in the performance of the best sparse classifier (i.e. small number of features) found by each of the algorithms (SPA, Lasso, ℓ_1-SVM). For all learned classifiers with 10 to 30 non-zero components, Table 3 presents those with the best classification accuracy. Furthermore, we also considered a typical analysis pipeline (MALDIQuant) to see how the "purely-data-based" approaches (SPA, Lasso, ℓ_1-SVM) compare to a model-based approach[24]. In Table 3, it can be seen that SPA provides the sparsest solutions while achieving competitive results with respect to sensitivity and specificity at the same time. Lasso and ℓ_1-SVM select almost the same features and therefore perform similarly. On the other hand, MALDIQuant selects the features based on a prior model-based peak detection followed by a feature selection based on shrinkage diagonal discriminant analysis ([40]). But however, it still performs worst on the UHL data-set.

Medical interpretation of results

Pancreatic cancer is not only a common and increasingly frequent [41], but also still a fatal disease, with a survival rate of 3-5% five years after diagnosis [42]. The conventional tumor marker, Carbohydrate Antigen 19-9 (CA19-9), as a blood group antigen not present in a significant proportion of the patients [43], shows insufficient diagnostic sensitivity and specificity (AUC 0.71), even in combination with the second-line tumor marker Carcinoembryonic Antigen (CEA, combined

Table 3 This table shows the main results comparing the feature selection benchmarks of our approach with Lasso, ℓ_1-SVM, and Maldi-Quant. These are averages over 10 repetitions of a 5-fold cross-validation. Note that these results have been calculated based on the highest accuracy criterion for all classifiers with between 10 and 30 selected features. This particularly means that better accuracy values might be achieved for the individual methods if less sparse feature vectors would be allowed. For more details see text

	SPA				Lasso				ℓ_1-SVM				Maldi-Quant			
Dataset	Feat.[e]	Sens[f]	Spec[g]	B. Acc[h]	Feat.	Sens	Spec	B. Acc	Feat.	Sens	Spec	B. Acc	Feat	Sens	Spec	B. Acc
P. CA - UHL	20.48	0.975	0.949	0.962	29.94	0.969	0.939	0.954	27.72	0.964	0.947	0.955	21	0.888	0.888	0.888
P. CA - UHh	17.1	0.986	0.975	0.981	26.46	0.966	0.969	0.967	29.68	0.966	0.976	0.971	17	0.975	0.963	0.969

[e] Feat.: Number of features
[f] Sens: Sensitivity ($TP/(TP+FN)$)
[g] Spec: Specificity ($TN/(FP+TN)$)
[h] B. Acc: Balanced Accuracy ($\frac{\text{sens.+spec.}}{2}$)

AUC 0.75) [44]. The need for better markers for screening and differential diagnosis is evident, as panceratic carcinoma would be principally curable if detected and identified very early in the course of the disease. Along with the emerging "-omics"-technologies great hope was rised to find tumor-specific peptides or metabolic alterations to increase sensitivity and specificity of early and differential diagnostics, and several combinatory marker models could be identified by proteomics [10] and metabolomics [45]. Pancreatic carcinoma is a complex disease - it affects the metabolism as a whole (e.g. the so-called Warburg effect) [46], but also alters proteolytic activity [47]. Therefore, it might be naïve to expect a single marker capable to indicate presence, progression and exact type of the malignancy at once [48] 9– it might even be overly reductionistic to attribute these capabilities to a single model, even if it consists of several entities measured by different "- omics" technologies [43]. As Raftery states "basing inferences on a single "best" model as if the single selected model were true ignores model uncertainty, which can result in underestimating uncertainty about quantities of interest" [49], and the larger the "-omics" data-sets grow, the larger is the 'probability, that there is not one "single best" predictive marker model, but instead several with comparable selectivity [48]. And it is very reasonable to assume that, even on the same data-set, different algorithms might favor different models consisting of different feature sets and bring forth completely different results, when only the best differentiating models are regarded. For an in-depth comparison of the validity of the results of different algorithms, the underlying peak features should also be taken into account, and similarities in the selected features corroborate the algorithms superimposed on them. In the case of our study, we have the great advantage, that the same data-set was evaluated in three different studies: the principal one by Fiedler et al. [10], a subsequent BinDA-algorithm-based manuscript by Gibb and Strimmer published recently [50], and the present one. Fiedler et al. [10] identified one discriminating peptide, Platelet Factor 4 (m/z *3884*, identified in italics, double hits in bold) within four discriminating peaks (m/z 3194, *3884*, 4055, and **5959**). The 30 most differential peaks in Gibb et al. [50] were m/z 4495, 8868, 8989, 1855, 4468, 8937, 2023, 1866, 5864, 5946, 1780, 2093, **5906**, **5960**, 8131, 1207, 4236, 2953, 9181, 1021, ***1466***, 4092, 4251, 5005, 8184, 1897, 3264, 2756, 6051, and 1264, with m/z 8937 identified as pancreatic progenitor cell differentiation and proliferation factor-like protein. m/z 3884 could not be identified as discriminating marker (while it might play a role in pancreatic carcinoma nonetheless [51]), whereas m/z ***1466*** can be attributed to a fragment of fibrinopeptide A (DSGEGDFLAEGGGVR), as previously described in tumor samples [52]. In the present study, the peaks m/z ***1464***, 1546, 1944, **5904**, 1619, 4209, and 2662 could be identified as discriminating features. The slight mass shift of about 2 Da for m/z ***1464*** / ***1466*** and **5904** / **5906** is probably arising from different peak preprocessing procedures, peaks are wide enough to tolerate this deviation. Further investigations and the application of further algorithms on the same data-set are highly likely to yield a similar, partially overlapping set of features, each with a comparable discriminating power (Fiedler et al. [53] $AUC_{[3884/(CA19-9*CEA)]}$ 1.0; Gibb et al. [50] in a 5-feature model: accuracy of 0.96, sensitivity of 0.96, specificity of 0.97, positive predictive value of 0.97 and negative predictive value of 0.95; the present study $accuracy_{[UHL]}$ 0.96, $sensitivity_{[UHL]}$ 0.97, $specificity_{[UHL]}$ 0.95 and $accuracy_{[UHH]}$ 0.98, $sensitivity_{[UHH]}$ 0.99, $specificity_{[UHH]}$ 0.97. This also corresponds to a recently published comparable study investigating a glycoprotein marker panel (AUC 0.95) [54]. Biomarkers for clinical diagnostics comprise a wide field of applications (e.g. population-wide screening, early diagnostics, characterization, treatment guidance, efficacy and toxicity monitoring, prognosis, susceptibility estimation and many more) [43], each with special requirements for sensitivity and specificity, that are only partially condensed in the AUC as an overall selectivity measure [48]. Especially for screening purposes, sensitivity is extremely important [45], and clinically applied tests e.g. for newborn screening frequently surpass the 0.99 hallmark [53]. Compared with the conventional, "not-for-screening" marker CA19-9, the SPA-based model shows considerable improvement, however there is still a big gap to screening suitability, which in the next years might be bridged by improved sensitivity of new instrumentation, refined algorithms (as the SPA), and combination with other "markers" from the "big data" field, enabling a more holistic view – not only of the disease, but also of the affected patient [43].

Conclusions

Workflows for analyzing high-dimensional (bio-medical) data often contain a step where discriminating features between two groups need to be identified. This is important for applications such as classification and clustering but is also essential for understanding biological differences, e.g. between two phenotypes. In this paper we have presented a new algorithm based on the theory of *Compressed Sensing* that identifies the *minimal* set of such features. This is of particular importance for modern, very high-dimensional data-sets such as proteomics mass-spectrometry data to allow interpretation of the results. Our experiments and comparisons to state-of-the-art algorithms show that our method finds smaller features sets resulting in

similar or better results when used for a classification task.

Endnotes

[1] Assays, e.g. immunoassays, are used in molecular diagnostics to detect concentrations of specific molecules even in low concentrations from a biological sample, such as blood [55].

[2] The data-sets used in this paper contain $d = 42.381$ dimensions in each MS1 spectrum but our approach is not limited by that.

[3] In feature selection, one is interested in identifying relevant dimensions of the data (features) which can be used to distinguish between two (or more) classes within a data-set.

[4] Here, $\mathrm{sign}(\cdot)$ denotes the sign function, i.e., $\mathrm{sign}(t) = 1$ if $t \geq 0$ and $\mathrm{sign}(t) = -1$ if $t < 0$.

[5] We call a vector sparse if the number of non-zero entries is small.

[6] Here, $\langle \cdot, \cdot \rangle$ again denotes the Euclidean scalar product.

[7] For the sake of convenience, we formulate our algorithm as in (3), but with some slight modifications, it could be equivalently stated in the form of (2).

[8] Here, $\|z\|_0 := \#\{i \mid z_i \neq 0\}$ simply counts the number of non-zero elements of $z = (z_1, \ldots, z_d) \in \mathbb{R}^d$.

[9] Here, $\mathrm{conv}(S)$ denotes the convex hull of the set $S \subset \mathbb{R}^d$.

[10] m/z is the unit for the mass-to-charge ratio.

[11] Compared to "Compressed sensing-based data analysis" section, we are now using the standard notations from learning theory. In particular, the measurement vectors are denoted by x_i (instead of a_i) and the feature vector is ω_0 (instead of x).

[12] Actually, we use the smoothed data vectors $\tilde{x}_i$ from Step 2 as input for this computation. But in order to keep the notation simple, we still write x_i. This convention holds also for all forthcoming steps.

[13] Here $\mathrm{supp}(\hat{\omega}) = \{k \mid \hat{\omega}_k \neq 0\}$ denotes the support of $\hat{\omega}$, i.e., the set of indices corresponding to its non-zero entries.

[14] In practice, one would simply reject all indices that are not contained in $\mathrm{supp}(\tilde{\omega})$.

[15] Signal-to-noise ratio was calculated as $SNR = \frac{\text{power of signal}}{\text{power of noise}}$.

[16] Due to the redundancy of the peak-associated feature variables (cf. Step 5 in "Algorithmic details" subsection), an estimated feature vector is considered to be equal to the ground-truth vector with some tolerance, which particularly depends on the width of the peaks.

[17] This difference arises from the implementation of Lasso.

[18] TP - true positives, i.e. correctly identified peaks
FP - false positives, i.e. incorrectly identified peaks
TN - true negatives, i.e. correctly rejected peaks
FN - false negatives, i.e. incorrectly rejected peaks

[19] Blood serum of 16 apparently healthy individuals from a clinical study ([37]) was used.

[20] Protein calibration standard mix Part No.: 206355 & 206196) from Bruker Daltronics (Leipzig, Germany)

[21] The power of noise for each of the 5 analyzed data-sets is estimated as an average of intensity of noise of the observations using median absolute deviation.

[22] The interested reader might find a good starting point about this topic in these two reviews [56, 57]

[23] Here, the standard MATLAB implementation of SVM was used.

[24] By "model-based" we mean that specific model assumptions on the data are made and exploited, such as noise-structure for denoising or Gaussian-shaped structures for peak detection.

Additional file

Additional file 1: Supporting Information(PDF 56 kb)

Abbreviations

AIC: Akaike information criterion; BIC: Bayesian information criterion; CS: Compressed sensing; MALDI-TOF: Matrix-assisted laser desorption ionization time-of-flight; ML: Machine learning; MS: Mass spectrometry; SPA: Sparse proteomics analysis; SVM: Support vector machine; TP: True positive; TN: True negative; FP: False positive; FN: False negative

Acknowledgements

The authors are thankful to Irena Bojarovska for fruitful discussions and help conducting the experiments.

Funding

JV was supported by the ERC CZ grant LL1203 of the Czech Ministry of Education. TC, MG, NC, JV, GK and CS were supported by the Einstein Center for Mathematics Berlin (ECMath), project grant CH2, and by the DFG Research Center Matheon *Mathematics for key technologies*, Berlin. TC and CS are supported by the German Ministry of Research and Education (BMBF) project Grant 3FO18501 (Forschungscampus MODAL). GK acknowledges support by the Einstein Foundation Berlin, by the Deutsche Forschungsgemeinschaft (DFG), and by the DFG Collaborative Research Center TRR 109 *Discretization in Geometry and Dynamics*.

Availability of data and materials

The method source-code can be downloaded from our homepage: http://software.medicalbioinformatics.de. The used data will be made available through request from the authors.

Authors' contributions

Conceived and designed the experiments: TC, MG, JV, GK, and CS. Performed the experiments: NC, TC, and NW. All authors contributed to writing the paper. All authors read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Consent for publication
Not applicable.

Ethics approval and consent to participate
The ethics committees of the Medical Faculties of the Universities of Leipzig and Heidelberg approved the use of the samples. All subjects gave written informed consent at both centers to participate in the study.

Author details
[1]Department of Mathematics, Freie Universität Berlin, Arnimallee 6, Berlin, Germany. [2]Department of Mathematics, Technische Universität Berlin, Düsternbrooker Weg 20, Berlin, Germany. [3]Center of Laboratory Medicine, Inselspital - Bern University Hospital, Düsternbrooker Weg 20, 24105 Bern, Switzerland. [4]Department of Mathematical Analysis, Charles University, Düsternbrooker Weg 20, Prague, Czech Republic. [5]Zuse Institute Berlin, Takustr. 7, Berlin, Germany.

Received: 27 August 2016 Accepted: 24 February 2017
Published online: 09 March 2017

References

1. Aebersold R, Mann M. Mass spectrometry-based proteomics. Nature. 2003;422(6928):198–207.
2. Petricoin EF, Belluco C, Araujo RP, Liotta LA. The blood peptidome: a higher dimension of information content for cancer biomarker discovery. Nat Rev Cancer. 2006;6(12):961–7.
3. Rai AJ, Chan DW. Cancer proteomics: serum diagnostics for tumor marker discovery. Ann N Y Acad Sci. 2004;1022:286–94.
4. Coombes KR, Morris JS, Hu J, Edmonson SR, Baggerly KA. Serum proteomics profiling–a young technology begins to mature. Nat Biotechnol. 2005;23(3):291–2.
5. Liotta LA. Clinical proteomics: written in blood. Nature. 2003;425:905.
6. Phizicky E, Bastiaens PIH, Zhu H, Snyder M, Fields S. Protein analysis on a proteomic scale. Nature. 2003;422:26–30.
7. Issaq HJ, Xiao Z, Veenstra TD. Serum and plasma proteomics. Chem Rev. 2007;107(8):3601–20.
8. Stühler K, Meyer HE. MALDI: more than peptide mass fingerprints. Curr Opin Mol Ther. 2004;6(3):239–48.
9. Sitek B, Waldera-Lupa DM, Poschmann G, Meyer HE, Stühler K. Application of label-free proteomics for differential analysis of lung carcinoma cell line A549. Methods Mol Biol. 2012;893:241–8.
10. Fiedler GM, Leichtle A, Kase J, Baumann S, Ceglarek U, Felix K, et al. Serum peptidome profiling revealed platelet factor 4 as a potential discriminating peptide associated with pancreatic cancer. Clin Cancer Res. 2009;15(11):3812–9.
11. Strenziok R, Hinz S, Wolf C, Conrad TOF, Krause H, Miller K, et al. Surface-enhanced laser desorption/ionization time-of-flight mass spectrometry: serum protein profiling in seminoma patients. World J of Urology. 2009;28(2):193–7.
12. Leichtle A, Nuoffer JM, Ceglarek U, Kase J, Conrad TOF, Witzigmann H, et al., Vol. 8. Serum amino acid profiles and their alterations in colorectal cancer; 2011, pp. 643–653.
13. Diao L, Clarke CH, Coombes KR, Hamilton SR, Roth J, Mao L, et al. Reproducibility of SELDI Spectra Across Time and Laboratories. Cancer Inform. 2011;10:45–64.
14. Marrugal A, Ojeda L, Paz-Ares L, Molina-Pinelo S, Ferrer I, Vol. 2016. Proteomic-Based Approaches for the Study of Cytokines in Lung Cancer; 2016, pp. 1–12.
15. Tang S, Zhou F, Sun Y, Wei L, Zhu S, Yang R, et al. CEA in breast ductal secretions as a promising biomarker for the diagnosis of breast cancer: a systematic review and meta-analysis. Breast Cancer. 20161–7.
16. Le N, Sund M, Vinci A, Beyer G, Javed MA, Krug S, et al. Prognostic and predictive markers in pancreatic adenocarcinoma. Dig Liver Dis. 2016;48(3):223–30.
17. Donoho DL. Compressed sensing. IEEE Trans Inform Theory. 2006;52:1289–306.
18. Candés EJ, Tao T. Decoding by linear programming. IEEE Trans Inform Theory. 2005;51:4203–15.
19. Candés EJ, Romberg J, Tao T. Stable signal recovery from incomplete and inaccurate measurements. Comm Pure Appl Math. 2006;59:1207–23.
20. Genkin A, Lewis D, Madigan D. Largescale Bayesian logistic regression for text categorization. Technometrics. 2007;49:291–304.
21. Friedman J, Hastie T, Tibshirani R. Regularized paths for generalized linear models via coordinate descent: Department of Statistics, Stanford University; 2008.
22. Efron B, Hastie T, Johnstone I, Tibshirani R. Least angle regression. Ann Statist. 2004;32:407–99.
23. Koh K, Kim S, Boyd S. An interior-point method for large-scale l1-regularized least squares. Selected Topics Signal Process. 2007;1(4):606–17.
24. Wu TT, Lange K. Coordinate descent algorithms for lasso penalized regression. Ann Appl Stat. 2008;2:224–44.
25. Vapnik VN. Statistical Learning Theory. New York: John Wiley & Sons; 1998.
26. Genzel M, Kutyniok G. Towards a Mathematical Theory of Feature Selection from Real-World Data with Non-Linear Observations. preprint; 2016.
27. Chen SS, Donoho DL, Saunders MA. Atomic decomposition by basis pursuit. SIAM J Sci Comput. 1998;20:33–61.
28. Tibshirani R. Regression shrinkage and selection via the Lasso. J R Stat Soc B. 1996;58:267–88.
29. Boufounos PT, Baraniuk RG. 1-Bit compressive sensing. In: Proceedings of the 42nd Annual Conference on Information Sciences and Systems. Princeton, NJ: IEEE, Princeton, NJ; 2008. p. 16–21.
30. Plan Y, Vershynin R. One-bit compressed sensing by linear programming. Comm Pure Appl Math. 2013;66:1275–97.
31. Plan Y, Vershynin R. Robust 1-bit compressed sensing and sparse logistic regression: a convex programming approach. IEEE Trans Inf Theory. 2013;59(1):482–94.
32. Zou H, Hastie T. Regularization and variable selection via the elastic net. J R Stat Soc B. 2005;67(2):301–20.
33. Davenport MA, Duarte MF, Eldar YC, Kutyniok G. Introduction to compressed sensing. Cambridge: Cambridge Univ. Press; 2012.
34. Foucart S, Rauhut H. A mathematical introduction to compressive sensing. New York: Birkhäuser; 2013.
35. Bühlmann P, Van De Geer S. Statistics for high-dimensional data: methods, theory and applications. Berlin/Heidelberg: Springer; 2011.
36. Gibb S, Strimmer K. MALDIquant: a versatile R package for the analysis of mass spectrometry data. Bioinformatics. 2012;28(17):2270–1.
37. Kratzsch J, Fiedler GM, Leichtle A, Brügel M, Buchbinder S, Otto L, et al. New reference intervals for thyrotropin and thyroid hormones based on National Academy of Clinical Biochemistry criteria and regular ultrasonography of the thyroid. Clin Chem. 2005;51(8):1480–6.
38. Sauve AC, Speed TP. Normalization, baseline correction and alignment of high-throughput mass spectrometry data. In: Proceedings of the Data Procedings Gensips. Baltimore: IEEE; 2004.
39. Rubin DB. Inference and missing data. Biometrika. 1976;63(3):581–92.
40. Ahdesmäki A, Strimmer K. Feature selection in omics prediction problems using cat scores and false non-discovery rate control. Ann Appl Stat. 2010;4:503–519.
41. Yeo TP, Lowenfels AB. Demographics and epidemiology of pancreatic cancer. Cancer J. 2012;18(6):477–84.
42. Michl P, Pauls S, Gress TM. Evidence-based diagnosis and staging of pancreatic cancer. Best Pract Res Clin Gastroenterol. 2006;20(2):227–51.
43. Leichtle A. Biomarker – vom Sein und Wesen. J Lab Med. 2015;39:97–101.
44. Reitz D, Gerger A, Seidel J, Kornprat P, Samonigg H, Stotz M, et al, Vol. 68. Combination of tumour markers CEA and CA19-9 improves the prognostic prediction in patients with pancreatic cancer; 2015, pp. 427–33.
45. Leichtle A, Ceglarek U, Weinert P, Nakas CT, Nuoffer JM, Kase J, et al. Pancreatic carcinoma, pancreatitis, and healthy controls - metabolite models in a three-class diagnostic dilemma. Metabolomics. 2013;9(3):677–87.
46. Zhou W, Capello M, Fredolini C, Racanicchi L, Piemonti L, Liotta LA, et al. Proteomic analysis reveals Warburg effect and anomalous metabolism of glutamine in pancreatic cancer cells. J Proteome Res. 2012;11(2):554–63.
47. Brand RE, Nolen BM, Zeh HJ, Allen PJ, Eloubeidi MA, Goldberg M, et al. Serum biomarker panels for the detection of pancreatic cancer. Clin Cancer Res. 2011;17(4):805–16.
48. Leichtle AB, Dufour JF, Fiedler GM. Potentials and pitfalls of clinical peptidomics and metabolomics. Swiss Med Wkly. 2013;w13801:143.

49. Raftery AE, Madigan D, Hoeting JA. Bayesian model averaging for linear regression models. JASA. 1997;92(437):179–91.
50. Gibb S, Strimmer K. Differential protein expression and peak selection in mass spectrometry data by binary discriminant analysis. Bioinformatics. 2015;31(19):3156–62.
51. Poruk KE, Firpo MA, Huerter LM, Scaife CL, Emerson LL, Boucher KM, et al. Serum platelet factor 4 is an independent predictor of survival and venous thromboembolism in patients with pancreatic adenocarcinoma. Cancer Epidemiol Biomarkers Prev. 2010;19(10):2605–10.
52. Villanueva J, Shaffer DR, Philip J, Chaparro CA, Erdjument-Bromage H, Olshen AB, et al. Differential exoprotease activities confer tumor-specific serum peptidome patterns. J Clin Invest. 2006;116(1):271–84.
53. Ceglarek U, Leichtle A, Brügel M, Kortz L, Brauer R, Bresler K, et al. Challenges and developments in tandem mass spectrometry based clinical metabolomics. Mol Cell Endocrinol. 2009;301(1-2):266–71.
54. Nie S, Lo A, Wu J, Zhu J, Tan Z, Simeone DM, et al. Glycoprotein biomarker panel for pancreatic cancer discovered by quantitative proteomics analysis. J Proteome Res. 2014;13(4):1873–84.
55. Rissin D, Kan C, Campbell T, Howes S, Fournier D, Song L, et al. Single-molecule enzyme-linked immunosorbent assay detects serum proteins at subfemtomolar concentrations. Nat Biotechnol. 2010;28:595–9.
56. Pigott TD. A review of methods for missing data. Educ Res Eval. 2001;7(4): 353–83.
57. Schafer JL, Olsen MK. Multiple imputation for multivariate missing-data problems: a data analyst's perspective. Multivar Behav Res. 1998;33(4): 545–71.