

# A geometric approach for the alignment of liquid chromatography–mass spectrometry data

Eva Lange<sup>1,\*</sup>, Clemens Gröpl<sup>1</sup>, Ole Schulz-Trieglaff<sup>1,2</sup>, Andreas Leinenbach<sup>3</sup>, Christian Huber<sup>3</sup> and Knut Reinert<sup>1</sup>

<sup>1</sup>Free University Berlin, Department of Mathematics and Computer Science, <sup>2</sup>Max Planck Research School for Computational Biology and Scientific Computing, Berlin, Germany and <sup>3</sup>Saarland University, Department of Chemistry, Instrumental Analysis and Bioanalysis, Saarbrücken

## ABSTRACT

**Motivation:** Liquid chromatography coupled to mass spectrometry (LC-MS) and combined with tandem mass spectrometry (LC-MS/MS) have become a prominent tool for the analysis of complex proteomic samples. An important step in a typical workflow is the combination of results from multiple LC-MS experiments to improve confidence in the obtained measurements or to compare results from different samples. To do so, a suitable mapping or *alignment* between the data sets needs to be estimated. The alignment has to correct for variations in mass and elution time which are present in all mass spectrometry experiments.

**Results:** We propose a novel algorithm to align LC-MS samples and to match corresponding ion species across samples. Our algorithm matches landmark signals between two data sets using a geometric technique based on pose clustering. Variations in mass and retention time are corrected by an affine dewarping function estimated from matched landmarks. We use the pairwise dewarping in an algorithm for aligning multiple samples. We show that our pose clustering approach is fast and reliable as compared to previous approaches. It is robust in the presence of noise and able to accurately align samples with only few common ion species. In addition, we can easily handle different kinds of LC-MS data and adopt our algorithm to new mass spectrometry technologies.

**Availability:** This algorithm is implemented as part of the OpenMS software library for shotgun proteomics and available under the Lesser GNU Public License (LGPL) at [www.openms.de](http://www.openms.de)

**Contact:** [lange@inf.fu-berlin.de](mailto:lange@inf.fu-berlin.de)

## 1 INTRODUCTION

Liquid chromatography-mass spectrometry (LC-MS) is among the dominant technologies for high-throughput proteomics experiments. It can provide quantitative and qualitative information about the compounds in a biological sample acquired on a large scale (Mann and Aebersold, 2003). The sample is usually subjected to a proteolytic digestion which yields a mixture of peptides and other compounds. This mixture is injected into a chromatographic column for a first separation. Due to their interaction with the stationary phase of the column, peptides elute at different retention times (RT). The resulting sample fractions are continuously injected into a mass spectrometer, ionized and separated by their

mass/charge ( $m/z$ ) ratio. Thus LC-MS experiments yield a set of 3D points, described by  $m/z$ , RT and ion count, usually called an *LC-MS map* (see also Fig. 1).

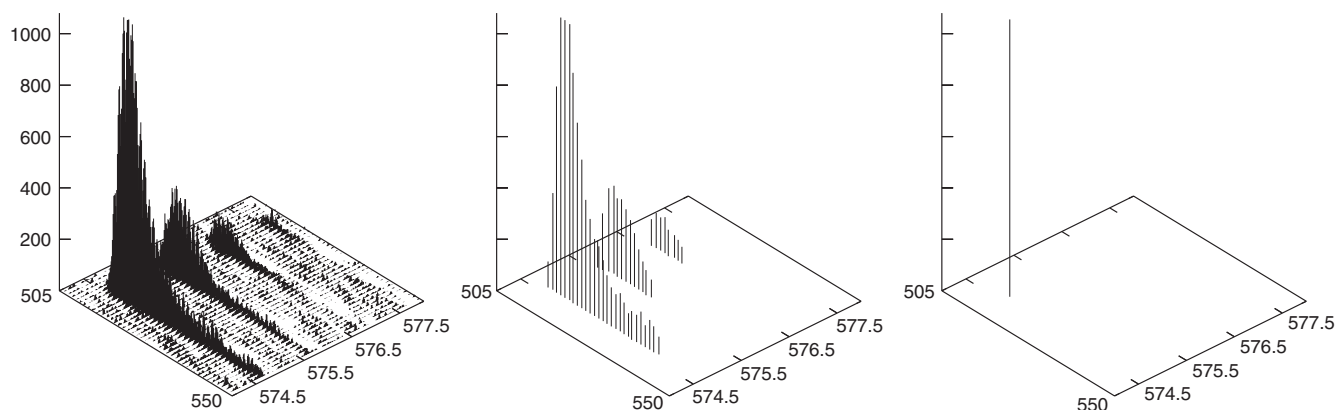
To quantify all peptides in a proteomic sample based on information in a LC-MS map, one has to summarize the signals caused by all charge and isotopic variants of the peptides and extract the masses and elution times from their joint distribution in subsequent LC-MS spectra. Several algorithms exist for this task, such as Li *et al.* (2003), Schulz-Trieglaff *et al.* (2007) and Wang *et al.* (2003). This is also a significant data reduction step and usually yields a manageable list of *compounds* or *features*, each characterized by mass, RT and abundance. Figure 1 shows a single charge variant of a peptide at different stages of this data reduction process.

The quantitative information in a LC-MS map can be used in numerous applications. The spectrum ranges from additive measurements in analytical chemistry (Gröpl *et al.*, 2005), over analysis of time series in expression experiments, to applications in clinical diagnostics, in which we want to find statistical significant markers for detecting certain disease states. All these applications have in common that we need to relate the same peptides in different measurements to each other. This is usually done under the assumption that the measured  $m/z$  and RT of a peptide stay roughly constant. As with each laboratory experiment, this only holds true to a certain extent.

In particular, the RT often shows large shifts and possibly distortions when different runs are compared, but also the  $m/z$  dimension might show (relatively smaller) distortions. This fact makes the assignment of similar peptides difficult since the relative shift of two maps to each other is not known in advance. But it is crucial to correct for those warps. Otherwise, it is hard or even impossible to find for a peptide in the first map and the corresponding partner in the second map.

The advent of high-throughput quantitative proteomics made a solution to this problem a key task. Any computational approach to this problem should compute a transformation which shifts one map relative to a second such that (a) the correct peptide pairs can be found, and (b) the transformation can be computed quickly as to allow for the pairwise comparison of hundreds to thousands of maps in reasonable time. The first requirement addresses the ‘correctness’ of the computed transform, the second requirement comes from the fact that most applications require the (multiple) alignment of a

\*To whom correspondence should be addressed.



**Fig. 1.** Part of an LC-MS map at different stages of data reduction. Axes depict RT,  $m/z$  and intensity. From left to right raw data points, peak picked data points and a feature are shown.

large number of maps (e.g. to achieve statistical significance in clinical studies).

Since current LC-MS instruments are still undergoing rapid development, it is crucial that any LC-MS alignment algorithm is as independent as possible of format and type of the LC-MS data.

To summarize, the correction of systematic shifts in  $m/z$  and RT between corresponding ion species in different MS samples along with the arrangement of them is called alignment. We present a new computational approach to this problem. Our method is inspired by pose clustering and we will demonstrate that it is fast, reliable and robust in the presence of noise.

## 1.1 Previous work

The computational challenges in LC-MS map alignment have recently moved into the focus of the bioinformatics community and several alignment algorithms have already been developed. We will review some of the key concepts and previous publications and contrast them to our approach.

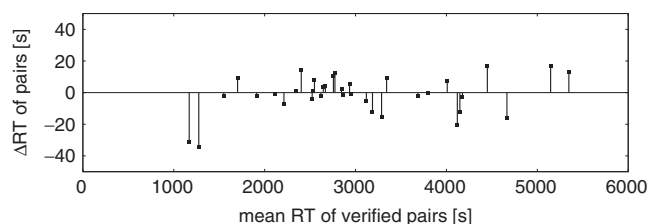
Many algorithms are based on two standard non-parametric approaches which are dynamic time warping (DTW) (Sakoe and Chiba, 1978) and correlation optimized warping (COW) (Nielsen *et al.*, 1998). Both approaches align time series by stretching or shrinking the time axis. DTW has its origin in speech processing and computes a non-linear mapping of one signal onto another by minimizing the distances between time series. COW is comparable to DTW, but it computes a piecewise linear transformation by dividing the time series into segments and then performing a linear warp within each segment to optimize overlap while constraining segment boundaries. The parameters for the best linear transformation are determined by maximizing the sum of correlation coefficients or covariance between data segments in pairs of samples. Both techniques appeared first in the alignment of chromatograms (Tomasi *et al.*, 2004) and were afterwards extended for the case of 2D LC-MS data (Bylund *et al.*, 2002; Prakash *et al.*, 2006; Prince and Marcotte, 2006). Prakash *et al.* (2006) and Prince and Marcotte (2006) describe an extension of DTW and differ mainly in the similarity function they maximize. Prakash *et al.* (2006) introduces a score based

on a normalized dot product of the mass spectra, which lowers the influence of noise peaks. Prince and Marcotte (2006) verify the applicability of DTW for the alignment of LC-MS raw data and show in a comprehensive study that the best scoring function for the similarity of MS spectra is the Pearson correlation coefficient coupled with a global gap penalty function.

Listgarten *et al.* (2007) propose a continuous profile model (CPM) which aligns multiple raw data sets and normalizes the peak intensities at the same time. The disadvantage of this HMM-based alignment is the reduction of the LC-MS map to its total ion chromatogram and its computational costs.

In general, raw map alignment methods tend to produce more accurate warping functions, but they are computationally expensive and therefore often not applicable for the multiple alignment of many samples. Moreover, algorithms that compute an alignment using time warping cannot accommodate for reversals in the RT of peptides. If in one measurement peptides  $A$ ,  $B$  and  $C$  appear in the order  $A-B-C$  and in the second measurement in order  $C-B-A$  those approaches can inherently assign only one correct pair. This scenario is not unlikely if the RT of  $A$ ,  $B$  and  $C$  are within a short interval. Since time warping algorithms preserve the temporal order of the peptides, these methods are only suitable for the determination of the warping function, but not for the mapping of corresponding elements.

In contrast to raw map alignment methods, there exist also approaches for aligning processed LC-MS data sets (America *et al.*, 2006; Jaitly *et al.*, 2006). America *et al.* (2006) perform a multiple alignment of peak data. All maps are aligned to a reference map in an iterative manner. Jaitly *et al.* (2006) propose an algorithm for the multiple alignment of feature maps. They compute a piecewise linear warping function for each pair of feature maps. Using a separation of the time dimension into segments, matching scores for the alignment of all segment pairs can be calculated and the alignment score is maximized via dynamic programming. To compute a multiple alignment, a complete or single-linkage clustering approach is performed to search for feature cluster across multiple maps.



**Fig. 2.** The plot shows the remaining differences in RT after a suitable affine dewarping function has been applied to the time standard (37 verified common identifications) of 6-28-03 and 7-11-03 (both protein profilings of *Mycobacterium smegmatis* in middle exponential growth-phase). For each pair of RT ( $s_i, m_i$ ), we plot  $s_i - m_i$  (vertically) against  $(s_i + m_i)/2$  (horizontally). The figure shows that the remaining error after affine dewarping is almost independent from the RT. The affine transformation used for dewarping was calculated by a linear regression of all but the lowest two RT pairs.

An alignment algorithm that improves upon previous work should be able to cope with order changes in elution time. This poses a problem for approaches based on DTW. Furthermore, it should be as independent as possible of the data format and processing state of the data.

## 1.2 LC-MS map alignment using pose clustering

We present a novel algorithm for the alignment of LC-MS maps that emphasizes the ‘geometric’ rather than the ‘sequential’ aspects of time warping. Our algorithm makes the assumption that the dominant part of the optimal ‘dewarping function’ is an affine transformation of the form  $t(x) = ax + b$ . Note that our approach does not depend on this assumption and can be extended to accommodate more sophisticated dewarping function.

However, we observed that an affine transformation is frequently sufficient. Figure 2 shows the results of an experiment which supports this observation. We selected a set of high-confidence peptides in two LC-MS samples of *M. smegmatis*. We give details of data acquisition and sample processing further below. Corresponding peptides in both samples were matched and manually validated. An affine correction was applied to the RT coordinates. For each pair, we plot difference versus mean RT. As can be seen in Figure 2, the error in RT remaining after correction is scattered around zero. Although we could compute transformations using higher-order functions, it is doubtful whether they are necessary or even practical since there is the potential of overfitting.

Our algorithm is applicable to LC-MS maps at any stage of data reduction (see Fig. 1), robust to large shifts as well as random noise and order changes in elution time of the peptides as we will show further below. It is implemented in a modular fashion and can easily be extended. The algorithm is part of OpenMS (Kohlbacher *et al.* (2006)), our software framework for shotgun proteomics.

The remainder of this article is organized as follows. In Section 2, we describe our algorithm for the alignment of LC-MS maps and give details of data acquisition and sample

## MULTIPLE ALIGNMENT

**Input:** List of element maps  $map\_list = m_1, \dots, m_n$

**Output:** Consensus map  $consensus = (c_1, \dots, c_l)$  with combined elements  $c_i$

```
// multiple alignment
```

```
// choose map with highest number of features as model map
```

```
ref = indexOfModelMap( $m_1, \dots, m_n$ )
```

```
consensus =  $m_{ref}$ 
```

```
for all maps  $scene_i$  in  $map\_list$  (with  $i \neq ref$ ) do
```

```
   $t_i = \text{superpositionPhase}(model, scene_i)$ 
```

```
   $scene_{t_i} = \text{dewarp}(scene_i, t_i)$ 
```

```
  consensus = consensusPhase( $scene_{t_i}, consensus$ )
```

**Fig. 3.** Pseudocode of the multiple map alignment.

preparation. We evaluate sensitivity and robustness of our algorithm using real data in Section 3.

## 2 MATERIALS AND METHODS

An alignment technique for LC-MS maps has to find common elements in several LC-MS maps and combine them to a *consensus map*. Each element in such a consensus map represents a collection of ion species, one from each LC-MS map. Depending on the processing stage of the data, the *elements* of a map can be raw data points, 1D peaks or 2D features. Taking only the RT and the  $m/z$  position as well as the ion count of all elements into account, our algorithm is independent of the element type and even the experiment type itself. Our approach is inspired by point pattern matching algorithms from computational geometry. Consequently, our method can be used to align all possible combinations of raw or processed LC-MS or LC-MS/MS maps.

### 2.1 Algorithm for map alignment

The pseudocode of the algorithm for multiple alignment is shown in Figure 3. We are given a set of element maps. First, we select the map with the highest number of elements. It is used to initialize the reference or *model map*. The other maps are called *scene maps*.<sup>1</sup> They are successively aligned to the model map. Thus, we perform a star-like progressive multiple alignment based upon pairwise alignments. Each pairwise alignment step has a superposition phase and a consensus phase.

In the *superposition phase*, we estimate a suitable affine transformation of the scene map onto the model map. The pseudocode is shown in Figure 4. After some preprocessing, we compute an initial affine transformation that maps as many elements of the scene map as possible close to elements of the model map. This initial transformation is found using pose clustering and does not rely on any particular assignment of matching partners. After that, we apply the initial transformation to the scene map and find a list of reliable correspondences between the scene map and the model map. Once the list of matching partners has been found, we obtain the final affine transformation from it using linear regression.

In the *consensus phase*, we combine components of the superimposed maps. The pseudocode is shown in Figure 5. We find groups of matching elements and maintain growing consensus elements which represent them. Elements of the scene having a matching partner in the model are added to their corresponding consensus elements.

<sup>1</sup>The terms ‘model’ and ‘scene’ have been adopted from the parlance of object recognition. In fact, the model map might be chosen as an annotated list of theoretical masses with predicted or experimentally determined average RT.

## SUPERPOSITION PHASE

```

Input: Model map  $model = m_1, \dots, m_k$ , scene map  $scene = s_1, \dots, s_l$ 
Output: Affine transformation  $t_{final}$  such that  $t_{final}(model) \approx scene$ .

// preprocessing
normalizeIntensities(model)
normalizeIntensities(scene)
for all elements  $m$  in  $model$  do
   $partner\_list_m = searchForPartners(scene, m)$ 

// find initial transformation by pose clustering
for all elements  $m_1$  in  $model$  do
  for all elements  $m_2 \neq m_1$  in  $model$  do
    for all partners  $p_1$  of  $m_1$  in  $partner\_list_{m_1}$  do
      for all partners  $p_2 \neq p_1$  of  $m_2$  in  $partner\_list_{m_2}$  do
         $t_{local} = computeTransformation((m_1, p_1), (m_2, p_2))$ 
        if isAdmissible( $t_{local}$ ) then
          hashTransformation( $t_{local}$ )
 $t_{initial} = estimateInitialTransformation()$ 

// find corresponding element pairs
 $scene_{dewarped} = dewarp(scene, t_{initial})$ 
computeDelaunayTriangulation( $model$ )
 $pair\_list = findElementPairs(model, scene_{dewarped})$ 

// compute final transformation
 $t_{final} = linearRegression(pair\_list)$ 

```

Fig. 4. Pseudocode of the superposition phase.

## CONSENSUS PHASE

```

Input: Scene map  $scene = (s_1, \dots, s_l)$  and consensus map  $consensus = (c_1, \dots, c_n)$ 
Output: Updated consensus map  $consensus = (c'_1, \dots, c'_n)$  with combined elements  $c'_i$ , ( $l \leq n' \leq l + n$ ).

computeDelaunayTriangulation( $consensus$ )
for all elements  $s_i$  in  $scene$  do
  if  $c_i = searchCorrespondingElement(consensus)$  then
    insert( $s_i, c_i$ )
  else // no corresponding element  $c_i$  found
    push( $consensus, s_i$ )

```

Fig. 5. Pseudocode of the consensus phase.

Components without partners are pushed into the consensus map as singleton elements.

Both the superposition phase and the consensus phase rely on a method to find reliable matching partners. In the next two sections, we will first explain the superposition phase in detail and then describe the method we devised to find reliable pairs of matching elements efficiently.

**2.1.1 Superposition** In the superposition phase, we determine an affine transformation  $t(x) = ax + b$  that maps the points  $(s_1, s_2, \dots)$  of the *scene* onto (or nearby) the points  $(m_1, m_2, \dots)$  of the *model*, respectively. We refer to this task as the *superposition problem*. We call  $a$  the *scale* and  $b$  the *shift* of the mapping. The parameters  $a$  and  $b$  can be recovered from the data using a general paradigm for geometric superposition algorithms called *pose clustering* (Stockman, 1987). Note that the result of the superposition is a *mapping*  $t$ . We will use a list of *matched* element pairs  $((s_1, m_1), (s_2, m_2), \dots)$  as an intermediate step, though.

To apply pose clustering in this situation, one has to solve the superposition problem *locally*, i.e. for a limited number of data points.

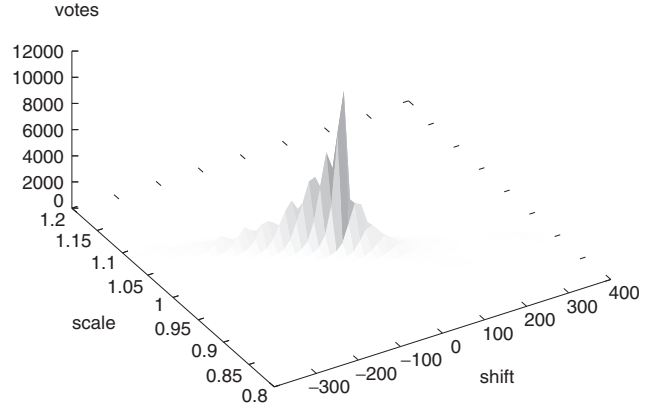


Fig. 6. Histogram of the transformation hash table used for aligning the *M.smegmatis* samples. The accumulation point stands out clearly. The minor ‘ripples’ are artifacts due to the discretization of positions during the resampling. For details see Section 3.

Note that the optimal affine transformation  $t$  is uniquely determined when there are only two data points in each map. In this case, the parameters  $a$  and  $b$  can be determined easily from a linear equation system: if  $t(s_1) = m_1$ ,  $t(s_2) = m_2$  and  $s_1 \neq s_2$ , then  $a = \frac{m_1 - m_2}{s_1 - s_2}$  and  $b = \frac{s_1 m_2 - s_2 m_1}{s_1 - s_2}$ .

For larger sets of points  $(s_1, s_2, \dots, s_k)$ ,  $(m_1, m_2, \dots, m_l)$ , the corresponding system of linear equations is generally overdetermined and one has to find an approximate solution that maps most points ‘close’ to their partners in the image. Moreover, we have to deal with the problem that the assignment of matching partners is not known to the algorithm.

Following the paradigm of pose clustering, we resolve both of these issues by a *voting scheme*. Consider the set of solutions of the local superposition problem for all pairs of pairs of data points  $(s_1, m_1), (s_2, m_2)$ . In the space of all affine transformations (which is spanned by the parameters  $a$  and  $b$ ) the ‘right’ transformation shows up as an accumulation point (or cluster), whereas the local solutions for non-matching pairs of pairs are more or less randomly distributed over the  $(a, b)$  plane. An example is shown in Figure 6. We use the centroid of the accumulation point as a guess for the optimal transformation. These initial parameters are optimized afterwards.

The algorithm records the candidate transformations in a hash table. The hash table itself is implemented as a sparse matrix (actually, a `std::map`), and the vote of a candidate transformation is distributed among its four neighboring discretized positions in the hash table in such a way that by taking their weighted average we will retrieve the original parameters.

In its simplest form, the voting scheme could iterate over all pairs of pairs of features and then search for the accumulation point using the hash table. However, this leads to an  $\Omega(k^2 l^2)$  algorithm, which is potentially very slow or even infeasible for  $k, l \geq 1000$ , as is often the case in real applications.

Fortunately, the set of candidate transformations is highly restricted for LC-MS maps. For example, experimentalists can guarantee that the scale of the warping function is in the range  $[0.5, 2]$  and in many cases, the shift can be bounded at least to some extent as well. The error in  $m/z$  will hardly ever exceed  $3 m/z$  over a typical mass-charge range like 400–3000  $m/z$  (we allowed 0.5  $m/z$  in our tests); this further reduces the number of candidates drastically. Besides the restriction of matching candidates among the maps, we also reduce the number of pairs considered in the *model* map during the superposition phase. We observed that it is sufficient to consider only

pairs of points in the *model* map that lie close together in  $m/z$ . This is a reasonable assumption since local distortions are frequently dominating. These restrictions on the set of candidate transformations can be enforced during the enumeration using appropriate geometric data structures for range queries.

In our algorithm, we multiply each vote by a weight indicating a level of confidence in the mapping before adding it to the hash table. We give matching points with similar intensity a higher vote than the matching of points with varying ion counts. This is a sensible assumption if the majority of peptides are not differentially expressed, which is usually the case. To make the ion count of elements in different maps comparable, we normalize by the total ion count of the map. We can easily incorporate other restrictions such as equal charge state of matched peptides, to prevent the hashing of unrealistic candidate transformations.

While the experimental side conditions cannot improve the theoretical worst-case running time, the number of candidate transformations is reduced to  $O(k^2 + l^2)$  under realistic assumptions. Besides speeding up the algorithm, the filtering of candidate transformations does also reduce the amount of background noise present in the hash table used by the voting scheme. This means that the right transformation will stand out more clearly. Another way to speed up the pose clustering is to consider only a random subset of pairs  $(m_1, m_2)$  in the model.

Once we have found the accumulation point in the hash map, we estimate the parameters of the transformations using a weighted average over a small neighborhood of it, to compensate for discretization errors and random fluctuations present in the data.

Finally, we apply the initial transformation to the model. Matching pairs are found using a Delaunay triangulation of the model map, as explained below.

We obtain the final transformation by linear regression using the list of matching pairs. While the final transformation will typically not differ much from the initial transformation, it is guaranteed to be at least locally optimal. Moreover, it renders our algorithm independent from small changes in the parameter settings applied for pose clustering, such as bin size of the hash table,  $m/z$  tolerance, etc.

Experimental side conditions can be incorporated in the subroutines `searchForPartners()`, `isAdmissible()`, `estimateInitialTransformation()` and `findElementPairs()`.

**2.1.2 Matching** The simplest way to find matching elements between dewarped maps would be to enumerate all pairs and check a distance criterion for them. The method we devised improves upon this in two ways: (1) two elements can be matched only if, for each of them, the other one is the nearest neighbor in the other map, and (2) the distance to the second-nearest neighbor is significantly larger than the distance to the nearest one.

Nearest and second-nearest neighbors of 2D points with respect to the Euclidean distance measure can be found using a geometric data structure called Delaunay triangulation (we use the implementation of CGAL, the Computational Geometry Algorithms Library [www.cgal.org](http://www.cgal.org)). In our case, differences in  $m/z$  are much less tolerable, and should be weighted more heavily, than differences in RT. We achieve the same effect by scaling the  $m/z$  dimension relative to the RT dimension proportionally to its weight.

## 2.2 Implementation

The multiple alignment algorithm is implemented within OpenMS, our software framework for mass spectrometry-based proteomics. The OpenMS tools allow us to deal with the most common data format for mass spectrometry data such as the open community format `mzXML` and `mzData` but also some proprietary formats. Each module of the algorithm also exists as an individual module of the OpenMS proteomics pipeline (TOPP) and can easily be integrated

into versatile computational workflows as described in Kohlbacher *et al.* (2006).

An instance of the alignment algorithm is constructed using the factory design pattern from a specific XML description. It is therefore very easy to add extensions or refinements to our setup. The implementation and OpenMS itself are obtainable free of charge from [www.openms.de](http://www.openms.de).

## 2.3 Sample preparation

**Data set A:** LC-LC-MS/MS runs of the tryptic digested *Mycobacterium smegmatis* proteome in different growth phases measured on an ESI ion trap mass spectrometer (for more details see Wang *et al.*, 2005). Raw `mzXML` data and corresponding SEQUEST (Eng *et al.*, 1994) identification results of 14 protein profiles of the bacterium were downloaded from the Open Proteomics Database (OPD).

**Data set B:** A tryptic digested protein mix of 10 known proteins (beta-Casein, conalbumin, myelin, hemoglobin, albumin, leptin, creatine, alpha1-Acid-Glycoprotein and bovine serum albumin). HPLC separation was performed on a capillary column (monolithic polystyrene-divinylbenzene phase, 60 mm × 0.3 mm) with 0.05% trifluoroacetic acid (TFA) in water (eluent A) and 0.05% TFA in acetonitrile (eluent B). Separation was achieved at a flow of 2.0 µl/min at 50°C with an isocratic gradient of 0–25% eluent B over 7.5 min. Eluting peptides were detected in a TOF mass spectrometer (microTOF from Bruker, Bremen, Germany) equipped with an electrospray ion source. Two replicate measurements were obtained from this sample.

## 3 RESULTS AND DISCUSSION

### 3.1 Multiple alignment and assessment of dewarping functions

Protein profiles of different biological states typically differ in many proteins and only share a small set of common proteins. An alignment algorithm would be used to align successive LC-MS runs to some reference map to facilitate meaningful comparisons between the different protein profiles. The amount of perturbations and inherent complexity of biological samples makes this a challenging task for every alignment algorithm.

Using the *M. smegmatis* maps which is the above-mentioned data set A, we aim to demonstrate that the concept of pose clustering is applicable for LC-MS sample alignment of real-world data.

We aligned moderately processed raw maps of the 14 LC-MS runs in data set A, but our method would also allow the application of further processing steps (such as peak picking or feature detection) before the alignment. Raw data was transformed into a uniformly spaced matrix by bilinear resampling. The spacing of the transformed matrix was 1  $m/z$  and 17 s. To remove the baseline, we applied a morphological top-hat filter to the spectra, which preserves structures like isotopic patterns that are narrower than 5  $m/z$ . After that, data points with an ion count below background noise level (in our case  $2 \cdot 10^6$ ) were discarded. The parameters for preprocessing were chosen after some initial inspection of the data, but our algorithm does not depend on them.

The LC-MS maps were aligned in a starwise manner. That is, we chose one map as a reference point, aligned and dewarped all other maps with respect to this map. The accuracy of the resulting warping functions is evaluated using time

**Table 1.** Multiple alignment of *M.smegmatis* LC-MS maps

Alignment	AAD (s)	
	None	OpenMS
6-17-03::4-03-03	259.59	35.89
6-17-03::6-18-03	74.003	39.1
6-17-03::6-28-03	43.96	32
6-17-03::7-11-03	52.67	28.12
6-17-03::7-13-03	42.22	13.46
6-17-03::7-17-03	67.96	9.74
6-17-03::7-19-03	73.9	14.16
6-17-03::7-20-03	74.68	18.9
6-17-03::7-21-03	40.95	12.32
6-17-03::7-22-03	78.54	26.9
6-17-03::7-23-03	157.57	15.53
6-17-03::7-24-03	114.03	41.3
6-17-03::7-25-03	101.49	17.3

Error measure as the average AAD in seconds of warped peptide standards.

standards for each pairwise alignment. MS/MS identification results of all runs were downloaded from the OPD and RT of high-confidence peptide identifications were extracted (Wang *et al.*, 2005). These peptide served as reference points and the alignment quality was evaluated based on their dewarped RT coordinates.

In Table 1, we give the error of each pairwise alignment measured as the average absolute difference (AAD) between the RT of the reference peptides after applying the estimated warping function. This quality measure for an alignment was introduced by Prince and Marcotte (2006). An optimal warping function should position the time standards precisely along the diagonal. Our algorithm dewarps the selected reference peptides accurately and compares favorably with Prince and Marcotte (2006). They measure the performance of their alignment algorithm by using the transitive AAD in a progressive alignment. Our algorithm aligns multiple maps in a starwise manner. Therefore, we can only measure the direct AAD for each pairwise alignment and our results are not directly comparable to Prince and Marcotte (2006).

Moreover, time warping algorithms tend to produce locally step-like functions, bearing the risk of overfitting. This may explain to some extent the slightly larger AADs values for the affine warping functions computed by our algorithm. We believe that the AAD is probably not a good quality measure for a multiple alignment of LC-MS samples. Besides finding the optimal warping function, it is even more important to match corresponding elements in multiple maps. Thus a more meaningful measure represents the number of correctly matched elements, but this requires hand-curated reference data which are not available yet.

Each of the *M.smegmatis* data sets consists of about 2350 elements. The alignment of all 14 maps takes only 4 s using our pose clustering algorithm. OBI-Warp needs roughly 13 s (personal communication). This experiment shows that the application of our method results in a quick and precise alignment of these complex data.

### 3.2 Aligning noisy LC-MS maps

After illustrating the performance of our approach on real-world data, we will now assess its robustness in a more specific setting. We systematically introduced noise into the peptide lists extracted from data set B and assessed the ability of our algorithm to match corresponding peptides in the presence of noise and changes in elution order.

In the previous section, we demonstrated that our pose clustering algorithm is able to accurately align complex samples with high precision and that it compares favorably to other approaches. But a precise and fast correction for transformations in RT is only one important criterion for the performance and applicability of an alignment algorithm.

Another important criterion is the ability to detect as many corresponding peptide pairs as possible in the presence of noise. Furthermore, differences in RT might cause changes in the order in which the peptides elute and consequently appear in the LC-MS map. Therefore, we designed a second set of experiments in which we particularly addressed these two issues.

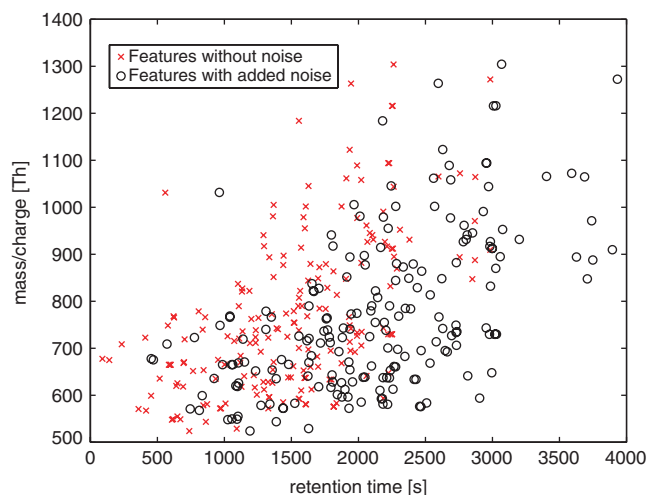
Our aim is to test the performance of our algorithm on noisy data with variations in mass and RT of the contained ion species of varying strengths. Furthermore, we want to assess the ability of our algorithm to deal with order-changes in the RT of ion species in the data.

The data sets resulting from the experimental procedure described above are of high resolution, i.e. single isotopic peaks for charges up to 4 can easily be distinguished and the LC-MS maps take up to 1 GB disk space per run. If dealing with data of this size, one usually reduces its complexity by summarizing isotopic peak cluster as proposed, e.g. by Horn *et al.* (2000) and Zhang and Marshall (1998). We followed this approach and summarized groups of data points to single peaks using a wavelet-based pattern matching algorithm (Lange *et al.*, 2006). In short, we apply the continuous wavelet transform using the Marr wavelet to detect single peaks in each MS spectrum. The corresponding raw data points are collected, refined by fitting a peak-shape function, and we estimate peak parameters such as area and FWHM from the data.

We create lists of peptide charge variants (or features) for each data set by grouping cluster of isotopic peaks that appear in consecutive scans. The charge of each feature is determined by fitting a theoretical isotope model based on the average composition of a peptide for a given mass as proposed earlier (Schulz-Trieglaff *et al.*, 2007).

We aligned these features and observed a high correlation between mass and RT of aligned features in both replicates (Pearson correlation  $R^2 = 0.99$  and Spearman rank correlation  $R_S = 0.98$ ). This indicates an alignment of very high quality which is of course due to the low complexity of the data set: the 10 proteins give rise to about 200 features in total. Mass and RT were measured with very high precision. We use this simple data set as a *ground truth* and systematically add noise to the coordinates ( $m/z$  and RT) of each detected peptide in order to assess the performance of our approach on noisy data. Our noise model for mass and RT is given by:

$$f(x) = x * a_{(m/z, RT)} + b_{(m/z, RT)} + \epsilon_{(m/z, RT)}$$



**Fig. 7.** The peptide features used for the robustness analysis and their distorted counterparts (with  $\sigma_{RT} = 30.0$  s).

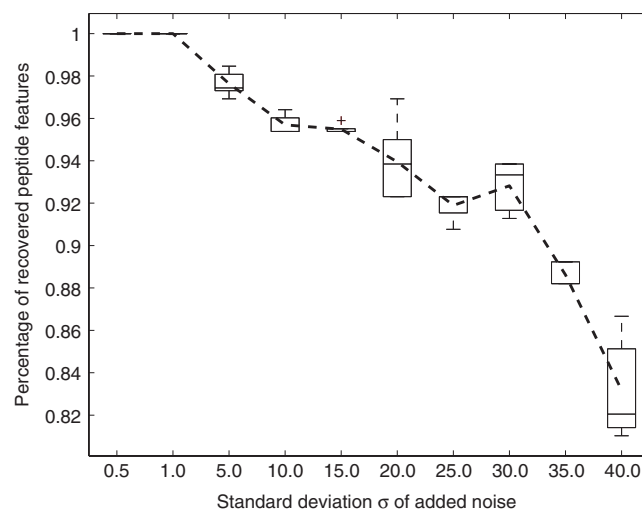
where  $x$  is the original coordinate (either RT or  $m/z$ ) and  $\epsilon_{(m/z, RT)} \sim N(0, \sigma_{(m/z, RT)}^2)$ .  $a_{(m/z, RT)}$ ,  $b_{(m/z, RT)}$  and  $\sigma_{(m/z, RT)}^2$  need to be chosen in advance. Figure 7 shows the feature set extracted from data set B and an exemplary set with added noise ( $b_{RT} = 300$ ,  $a_{RT} = 1.2$ ,  $b_{m/z} = 0.3$ ,  $a_{m/z} = 1.0$ ,  $\sigma_{m/z} = 0.1$   $m/z$  and  $\sigma_{RT} = 30.0$  s).

Note that we kept the noise added to the  $m/z$  feature coordinate relatively low, which is realistic since the mass is usually significantly stabler than the retention across different experiments. We also left the abundance of each feature unchanged. This allows us to verify that we aligned the true peptide feature pairs and not just pairs that had similar coordinates after transformation and addition of noise.

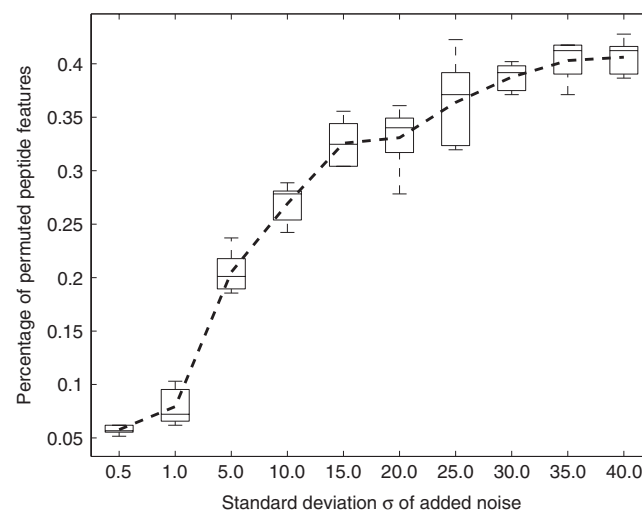
We kept the above-mentioned parameters fixed and changed only the SD of the noise added to the RT coordinate of each peptide feature. We computed alignments of the undistorted LC-MS map and its noisy copy for different values of  $\sigma_{RT}$  and counted the number of recovered feature pairs. Figure 8 gives an overview of the results. As expected, the number of correctly aligned features decreases with noise added. However, this decrease is relatively slow: with a deviation of 40 s in RT, 83% of the features can still be aligned correctly and matched to their counterparts in the original data set.

Figure 9 gives the percentage of RT order changes in elution time for each peptide feature for varying levels of added noise. The number of these permutations increases with the noise SD. This is due to certain characteristics of the data. Even if the 10 protein mixture is not too complex, it is relatively dense and consequently, the extracted peptide features lie closely together. Small disturbances in RT will already result in features moving even closer together, larger ones will result in peptides changing their elution order. Therefore, this data set is particularly well suited to assess the performance of our algorithm in these situations.

We are aware that the evaluation of algorithms on simulated data has its caveats. Nonetheless, these experiments allow us to assess the performance of our method on data with specific characteristics. It is also not clear if our model for the distortion



**Fig. 8.** Results of the robustness analysis. The SD  $\sigma_{RT}$  of the error in RT was systematically increased, ranging from 0.1 to 40. For each value of  $\sigma_{RT}$ , the results were averaged over 5 runs and box-and-whisker diagrams are given.



**Fig. 9.** Percentage of elution order changes (permutations) for changing SD  $\sigma_{RT}$  of added noise.

of  $m/z$  and RT coordinates comes close to perturbations in real experiments. But affine warps as introduced in these experiments are frequently observed in practice. Note that we sampled Gaussian distributed noise for each feature independently. This results in distortions that are more severe than one would expect in real-world data. In a real large-scale experiment, one would expect locally correlated perturbations in RT and systematic shifts in subsets of the LC-MS maps. Since we introduce noise into a real sample, and not an entirely artificial one, our data already incorporates this phenomenon to a certain extent. We further aggravate these drifts by applying our noise model to  $m/z$  and RT and by doing so, we can estimate the robustness of our algorithm and its ability to handle changes in the elution order of peptides, something which is impossible for algorithms based on DTW.

**Table 2.** Aligning LC-MS maps with little overlap: features were removed from the map and replaced by random features

Random features (%)	True features recovered (%)	$ a_{m/z}-a'_{m/z} $	$ b_{m/z}-b'_{m/z} $	$ a_{RT}-a'_{RT} $	$ b_{RT}-b'_{RT} $
20.0	94.23	0.00	0.10	0.19	6.46
40.0	95.79	0.01	0.10	0.20	8.18
60.0	96.01	0.10	0.05	0.21	10.71
80.0	92.02	0.08	0.10	0.22	10.91
90.0	90.04	0.11	0.09	0.23	11.24

We counted the percentage of remaining true features recovered as well as the absolute error for the estimated coefficients in the affine transformation. Note that an absolute error of 11.24 for  $b_{RT}$  (the shift for RT in the error model) is equivalent to a relative deviation of 0.037.

### 3.3 Aligning maps with little overlap

A third important issue in the performance evaluation of an alignment algorithm is the ability to align LC-MS maps with little overlap such as maps obtained from different sample fractions in a MuDPit (multidimensional protein identification technology) experiment. In these experiments, complex peptide mixtures are separated using 2D liquid chromatography. That is, several chromatographic columns are coupled and the separation proceeds in several steps.

The LC-MS data acquired in these experiments results in several sample fractions that are mostly distinct regarding the contained peptide but also share a set of common peptides. The size of this common peptide set depends on the column technology. Another application of alignment algorithms is to create the superset of the peptides contained in the sample fractions for further processing. To achieve this, peptides occurring in several fractions need to be found and used to compute an accurate alignment.

To assess the performance of our approach in a MudPit experiment, we replaced some of the peptide features from data set B by features at random positions. The remaining ones were transformed using the error model described in the previous section and the following parameter settings:  $b_{RT} = 300$ ,  $a_{RT} = 1.2$ ,  $b_{m/z} = 0.3$ ,  $a_{m/z} = 1.0$ ,  $\sigma_{m/z} = 0.1$   $m/z$  and  $\sigma_{RT} = 30.0$  s. The random features were inserted within the bounding box of the remaining true features.

We computed alignments for changing numbers of random features and counted the number of recovered true feature pairs. Again, we can validate these pairs since we left the abundances unchanged. We also tested to which extent the parameters of the transform could be recovered from the true feature set and give the absolute error (distance) between scale and shift estimated by our pose clustering algorithm ( $a'_{m/z,RT}, b'_{m/z,RT}$ ) and the true ones we used to distort the feature coordinates ( $a_{m/z,RT}, b_{m/z,RT}$ ).

Table 2 shows that our method is able to recover the true shift and scale with good precision. The true features left in the sample can be matched to a large extent even with a very high number of random features placed in the same area of the LC-MS map. The quality is not deteriorated for up to 60% random features and even for 90% we obtain a usable result.

Note that this map contains only 195 features. That means, that only 20 features were enough to estimate a good

dewarping function and to align the two maps (last row of Table 2).

## 4 CONCLUSIONS AND FUTURE WORK

The automatic alignment of LC-MS data sets is an important step in every high-throughput proteomics experiment. Algorithms that can perform this task efficiently and accurately have a huge potential for basic research in biology but also for more applied questions such as biomarker discovery and drug research in general.

To our best knowledge, the algorithm we presented in this work is the first one which is based on a geometric rather than a time warping approach. Consequently, we can deal with order changes of peptide elution times in a simple manner. Our approach scales well on real-world data. It is faster than previous approaches and computes alignments with a good precision. Moreover, we demonstrated that our pose clustering algorithm performs also well on noisy data with severe distortions in RT. We will include further computational experiments in the journal version of this article.

The proposed method is also independent of the processing stage of the LC-MS data it is applied to. This makes it flexible and applicable to any kind of data from upcoming LC-MS technologies and processing algorithms.

Finally, our experiments reveal that an affine transformation suffices to align and dewarp typical LC-MS samples. At the same time, it is easy to incorporate more sophisticated regressions and mapping functions due to the modular architecture of our algorithm and OpenMS in general.

We also plan to apply this algorithm on a set of biological studies in collaboration with experimental labs. An obvious extension would be the implementation of a progressive approach for multiple LC-MS map alignment and further evaluations in large-scale experiments.

## ACKNOWLEDGEMENTS

We thank Andreas Leinenbach (Huber Research Group, Saarland University Saarbruecken, Germany) for the preparation of the protein mix used in Section 3. C.G., E.L., and K.R. acknowledge funding by the Berlin Center for Genome Based Bioinformatics (BCB) and the German Federal Ministry for Education and Research. O.S.-T. was supported by the

Max Planck Research School for Computational Biology and Scientific Computing.

*Conflict of Interest:* none declared.

## REFERENCES

- America, A. *et al.* (2006) Alignment and statistical difference analysis of complex peptide data sets generated by multidimensional LC-MS. *Proteomics*, **6**, 641–653.
- Bylund, D. *et al.* (2002) Chromatographic alignment by warping and dynamic programming as a pre-processing tool for PARAFAC modelling of liquid chromatography mass spectrometry data. *J. Chromatogr. A*, **961**, 237–244.
- Eng, J.K. *et al.* (1994) An approach to correlate tandem mass-spectral data of peptides with amino-acid-sequences in a protein database. *J. Am. Soc. Mass Spectrom.*, **11**, 976–989.
- Gröpl, C. *et al.* (2005) Algorithms for the automated absolute quantification of diagnostic markers in complex proteomics samples. In Berthold, M. (ed.), *Proceedings of Computational Life Science (CompLife) 2005, Lecture Notes in Bioinformatics*, Springer, Heidelberg, pp. 151–163.
- Horn, D.M. *et al.* (2000) Automated reduction and interpretation of high resolution electrospray mass spectra of large molecules. *J. Am. Soc. Mass Spectrom.*, **11**, 320–332.
- Jaitly, N. *et al.* (2006) Robust algorithm for alignment of liquid chromatography-mass spectrometry analyses in an accurate mass and time tag data analysis pipeline. *Anal. Chem.*, **78**, 7397–7409.
- Kohlbacher, O. *et al.* (2006) TOPP – the OpenMS proteomics pipeline. In *Proceedings of the 5th European Conference on Computational Biology*.
- Lange, E. *et al.* (2006) High accuracy peak-picking of proteomics data using wavelet techniques. In *Proceedings of the Pacific Symposium on Biocomputing (PSB) 2006*, *Bioinformatics* 2007, **23**(2):e191–e197; doi:10.1093/bioinformatics/btl299.
- Li, X.-J. *et al.* (2003) Automated statistical analysis of protein abundance ratios from data generated by stable-isotope dilution and tandem mass spectrometry. *Anal. Chem.*, **75**, 6648–6657.
- Listgarten, J. *et al.* (2007) Difference detection in LC-MS data for protein biomarker discovery. *Bioinformatics*, **23**, e198–e204.
- Mann, M. and Aebersold, R. (2003) Mass spectrometry-based proteomics. *Nature*, **422**, 198–207.
- Nielsen, N.-P.V. *et al.* (1998) Aligning of single and multiple wavelength chromatographic profiles for chemometric data analysis using correlation optimised warping. *J. Chromatogr.*, **805**, 17–35.
- Prakash, A. *et al.* (2006) Signal maps for mass spectrometry-based comparative proteomics. *Mol. Cell Proteomics*, **5**, 423–432.
- Prince, J. and Marcotte, E. (2006) Chromatographic alignment of ESI-LC-MS proteomics data sets by ordered bijective interpolated warping. *Anal. Chem.*, **78**, 6140–6152.
- Sakoe, H. and Chiba, S. (1978) Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoust.*, **26**, 43–39.
- Schulz-Trieglaff, O. *et al.* (2007) A fast and accurate algorithm for the quantification of peptides from LC-MS data. In *Proceedings of Eleventh Annual International Conference on Research in Computational Molecular Biology, RECOMB 2007*, pp. 473–487.
- Stockman, G. (1987) Object recognition and localization via pose clustering. *Comput. Vision Graph. Image Process.*, **40**, 361–387.
- Tomasi, G. *et al.* (2004) Correlation optimized warping and dynamic time warping as preprocessing methods for chromatographic data. *J. Chemom.*, **18**, 231–241.
- Wang, R. *et al.* (2005) Mass spectrometry of the *M. smegmatis* proteome: protein expression levels correlate with function, operons, and codon bias. *Genome Res.*, **15**, 1118–1126.
- Wang, W. *et al.* (2003) Quantification of proteins and metabolites by mass spectrometry without isotopic labeling or spiked standards. *Anal. Chem.*, **75**, 4818–4826.
- Zhang, Z. and Marshall, A.G. (1998) A universal algorithm for fast and automated charge state deconvolution of electrospray mass-to-charge ratio spectra. *J. Am. Soc. Mass Spectrom.*, **9**, 225–233.