Accurate Mass and Time Databases for Increased Resolution in LC-MS/MS Experiments

6 June 2009

1 Abstract

We develop an algorithm for generating accurate mass and time (AMT) databases from high-resolution LC-MS/MS datasets.

2 Introduction

The task of joining a set of LC-MS/MS datasets into a representative accurate mass and time (AMT) database has typically been done by mapping retention time of each peptide identification to a normalized consensus axis. Although such an approach has been shown to be successful, an important consequence of such run-by-run normalization is that each dataset is internally calibrated; it is plausible that, in addition to this internal calibration, an additional step of inter-dataset alignment would increase the accuracy of retention time locations on the consensus axis.

The question of aligning multiple such datasets along a single dimension is analogous to the ubiquitous bioinformatics challenge of multiple sequence alignment. Here, we provide a multiple sequence alignment-based approach to cross-calibrating LC-MS/MS datasets prior to collection into an AMT database. We apply a basic neighbor joining algorithm to perform heirarchical agglomerative clustering and generate a calibration ”tree”. Starting at the leaves of the tree, we perform a pairwise alignment of retention time. This pairwise alignment is generated based on chosen ”anchor points” within the two runs - peptides which have been confidently identified via MS/MS (PeptideProphet probability > 0.9) in both experiments, thus providing a sense of the retention time drift between them. After each pairwise alignment, in keeping with complete-linkage clustering, the two aligned datasets are merged, and the distance matrix for the tree is recalculated. This agglomerative approach yields a final dataset that represents the location of all observed peptides in accurately calibrated retention time.

The purpose of such a database is generally to increase the resolution of LC-MS/MS experiments by searching the mass and time coordinates of an MS1 feature for a nearby peptide identification in the database. We tested the accuracy of our method in this regard by matching known MS2 identifications to the database independently of sequence; i.e., we used the mass and time coordinates of a set of MS2 identifications to query a consensus database. In this way, we quantified true and false positives and negatives in order to measure the sensitivity and specificity of our method. At a false positive rate of X, we achieve a true positive rate of Y.

Using this false positive rate as a constraint for match tolerance, we tested the ability of this method to increase the resolution of a single LC-MS/MS run. We generated a database from fractionated and pooled human serum samples. In total, our database contained $\sim X$ high-confidence peptide identifications, which yielded X unique high-confidence proteins when searched as a whole (ProteinProphet score > 0.9). In contrast, the query experiment contained $\sim X$ high-confidence peptide identifications, which yielded X unique high-confidence proteins alone. We obtained significant gains in protein identification with this approach, both in the sense of increasing the amino acid coverage and confidence of a subset of the originally identified proteins, as well as decreasing the confidence and score of another subset. These results show the ability of our method to filter false protein identifications in an experiment as well as to increase the confidence in true protein identifications. These increases in accuracy will prove useful as high-throughput proteomic profiling continues to gain popularity as a tool for comparative analysis of disease cases.

3 Methods

3.1 Overview of algorithm

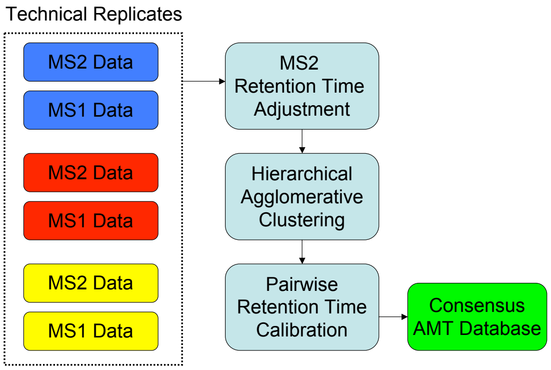
Technical Replicates
MS2 Data
MS1 Data
MS2 Data
MS1 Data
MS2 Data
MS1 Data
MS2 Retention Time Adjustment
Hierarchical Agglomerative Clustering
Pairwise Retention Time Calibration
Consensus AMT Database

Figure 1: Workflow of AMT database generation.

3.2 MS1 feature detection

Blah blah blah darren. Figure: feature detection workflow

3.3 Adjustment of sequencing event retention time to accurate location

Our consensus database is to be constructed based on the mass and time coordinates of MS2 identifications. Although the mass estimate given with an MS2 identification is fairly accurate (calibration in this area is another topic), the retention time coordinates given are not necessarily representative of the elution of a given peptide, as a peptide can be sequenced multiple times during its elution. To correct for this, we first preprocess these identifications to obtain the most accurate retention time value possible. Given a set of MS2 and MS1 data for a single LC-MS/MS experiment, we proceed to adjust retention time of the MS2 identifications to match the peak of the elution profile of the nearest MS1 feature in the dataset. Here, we define ”nearest” using the usual Euclidean distance, weighted to reflect that we expect mass per charge accuracy to be on the order of 10^{-3} whereas we expect retention time deviation to be on the order of 10^{2}. Given an MS2 identification and MS1 feature with mass per charge deviation of x and retention time deviation of y, we score the distance between the two as:

$$d(A,B) = \sqrt{\frac{x^2}{.001} + \frac{y^2}{100}} \tag{1}$$

3.4 Dataset clustering

As a preprocessing step to AMT database generation, the eharmony algorithm clusters runs based on a run-to-run distance attribute, or vector of such attributes. The eharmony source code provides a flexible interface for the definition of new distance attributes, allowing users to tailor the clustering step to an individual set of experiments. The final run-to-run distance function must define a pseudosemimetric on the set of all runs. That is, for any two runs A and B, along with distance function d, we require that:

$$\begin{aligned} d(A,B) &\geq 0 && (2)\\ d(A,A) &= 0 && (3)\\ d(A,B) &= d(B,A) && (4) \end{aligned}$$

These conditions arise intuitively as follows. We require (2) in order to restrict search for a minimum distance to the positive real numbers. (3) ensures that for every run, no other run is ”closer” to it than itself. We do not require that $d(A,B) = 0$ if and only if $A = B$, as it is certainly conceivable that two distinct runs share enough similarity in peptide content to have effectively no

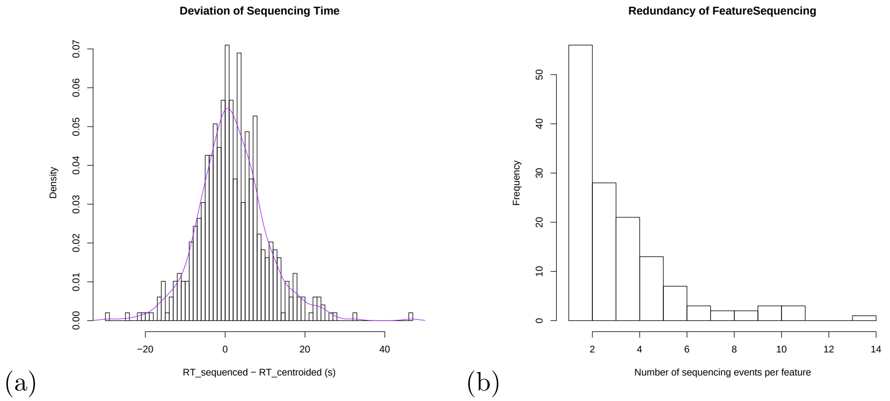

Figure 2: Adjustment of MS2 sequencing event retention time using elution profile. (a) Deviation of sequencing time from peak of elution profile. (b) Redundancy of sequencing events per peptide.

distance between them. Our implementation of retention time calibration is symmetric; therefore we require (4), as it is counterintuitive to include a preprocessing step that determines which two runs to calibrate based on a distance function that is asymmetric. Such a step would suggest additional gains from a direction-specific calibration, which will not be achieved with our algorithm.

We evaluated several such distance attributes; those containing multiple values are represented as vectors and distance is calculated as the unweighted dot product of these vectors between runs.

3.4.1 Mean and standard deviation of delta retention time between matching MS2 identifications

Let A and B be two runs, and let $P = \{P_1, \ldots, P_n\}$ be the set of peptides identified in both A and B. We denote the retention time of, e.g., the ith peptide in run A as $RT_{P_i,A}$. The mean and standard deviation of delta retention time between matching MS2 identifications are then calculated as :

$$\bar{\Delta}_{RT}(A,B) = \frac{1}{n}\sum_{i=1}^{n} |RT_{P_i,A} - RT_{P_i,B}| \tag{5}$$

$$\sigma_{\Delta_{RT}}(A,B) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(|RT_{P_i,A} - RT_{P_i,B}| - \bar{\Delta}_{RT}(A,B))^2} \tag{6}$$

The use of this attribute is motivated by the fact that we expect runs with high variability in retention time between matched MS2 identifications to be

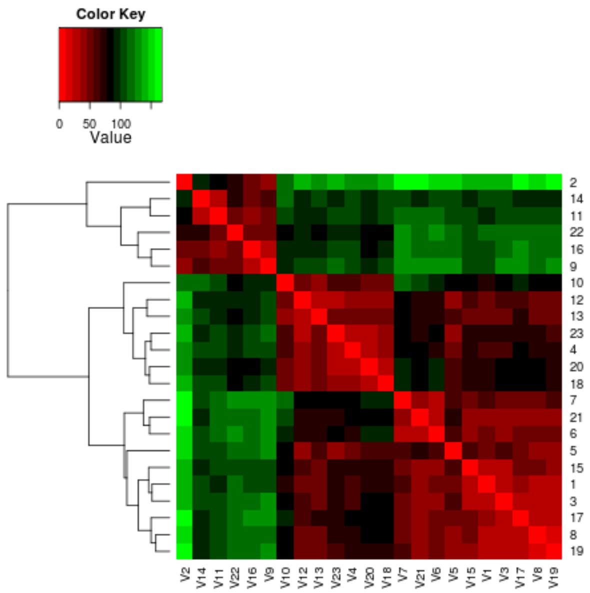

Figure 3: An example of hierarchical clustering using the retention time difference distribution distance attribute. The clearpatterning shows that this particular distance attribute captures an essential distinction between experiments.

intrinsically harder to calibrate; thus the quality of our retention time calibration should improve by preferentially calibrating those pairs of datasets with the smallest mean and standard deviation of delta retention time.

3.4.2 Hamming distance between vectors of peptide identifications

Consider the set $S = \{S_1, \ldots, S_n\}$ of all unique peptides identified in the set of runs that are to comprise the final AMT database. Let $X = \{X_1, \ldots, X_n\}$, $X_i \in \{0, 1\}$, be a binary vector whose entries represent the presence of peptide i in run A; similarly, let Y be an n-element vector representing the same for run B. The Hamming distance is then calculated as :

$$d_H(A, B) = \sum_{i=1}^{n} |(X_i - Y_i)| \tag{7}$$

This attribute essentially enumerates the number of MS2 identifications that differ between two runs. A limitation of this attribute is that, while it minimizes the number of non-matching peptides between two runs, it does not distinguish between a pair of runs with many matching peptides and a pair of runs with few matching peptides, nor does it correct for the total number of unique peptides identified in a run. In order to overcome these limitations, we introduce a modified Hamming distance calculation, as described below.

3.4.3 Weighted Hamming distance between vectors of peptide identifications

Let $S_A \subset S$ be the set of unique peptides identified in run A; define S_B analogously. The significance of the Hamming distance between A and B is related to the cardinality of these two sets; since we are measuring and minimizing $S_A \setminus S_B$ and $S_B \setminus S_A$, a weighting of the Hamming distance related to $|S_A|$ and $|S_B|$ will encapsulate the two phenomena above. That is, if we weight according to the total number of unique peptides identified in each run, we have accounted for not only the number of identifications but also for the size of the overlap. (This is much clearer with a picture. I promise it's true.)

We define a set of weights $W = \{W_{ij}\}$ such that each pairwise calculation of the Hamming distance is weighted by the sum of the number of unique peptides identified in experiment i and the number in experiment j (in the above notation, $W_{ij} = |S_i| + |S_j|$). The weighted Hamming distance for two runs A and B is then :

$$d'_H(A,B) = \frac{W_{AB}}{\Gamma} * d_H \tag{8}$$

where d_H is defined as above and, given N total experiments, Γ is a normalization factor calculated by summing the W_{ij} over all the $\frac{N(N-1)}{2}$ distinct candidate pairs for calibration :

$$\Gamma = \sum_{i=2}^{N} \sum_{j=1}^{i} (W_{ij}) \tag{9}$$

3.4.4 Damerau-Levenshtein distance between vectors of peptide identifications

The Damerau-Levenshtein distance allows us to heavily weight against calibration of two runs whose set of matching peptides includes a transposition event, i.e., peptide X elutes before peptide Y in run A, but after peptide Y in run B.

3.5 Pairwise retention time calibration

The goal of retention time calibration is to define a monotonically increasing function F that maps from the retention time space of a first experiment (RT_A) onto that of a second experiment (RT_B). Our basic approach is to restrict F to a certain functional form and then solve for the parameters of F that minimize some cost function C. As a particular example, suppose we restrict F to be a linear function:

$$F(x \in RT_A) = k_0 + k_1 x \tag{10}$$

Defining C as a sum-of-squared-residuals cost function, the coefficients k_0 and k_1 are found by a standard least-squares linear regression on the anchor points.

Our algorithm supports linear and piecewise linear calibration functions. The source code is written to allow a developer to easily implement new functions; it is up to the developer to determine that these functions satisfy the relevant constraints of the problem.

The complete retention time calibration process consists of two substeps: anchor selection and anchor interpolation. Anchor interpolation is done as detailed above; anchor selection is done in a relatively naive manner. The candidates for anchors are the retention time coordinates of matching peptide identifications between experiment A and B. The user defines two parameters, an integer n and a threshold t. The matching peptide identifications are sorted according to RT_A; every nth candidate is selected as an anchor, provided that $|RT_A - RT_B| < t$. This approach is clearly sub-optimal, and we hope to improve upon it in the future. The existing strategy was motivated by an attempt to avoid overfitting the data and to exclude outliers; data fitting and outlier exclusion can be handled by a variety of more sophisticated statistical techniques, and we intend to try a variety of methods to optimize this step.

3.6 Database postprocessing

We provide for an optional postprocessing step as part of our algorithm.

As the consensus database has been developed thus far, each peptide identification has been considered as an individual entity. Before querying the database, we include a post-processing step to merge peptides that have the same identified sequence into a single entity for matching. We call these entities ”islands”.

An island has an associated probability distribution that is calculated as follows. For each peptide identification in the island, the parameters of a two-dimensional (mass, retention time) normal distribution are determined. The mean of this distribution is the observed mass and retention time for the peptide entry. The standard deviation is determined based on the confidence of the original identification, i.e., Peptide Prophet score, as follows. We allow for a baseline mass standard deviation of .005 Daltons and a baseline retention time deviation of 100 seconds; these values are then weighted by the Peptide Prophet score. The motivation behind this weighting is that we would like to restrict matching such that it is harder for a feature to map to a low-confidence peptide than it is for it to match a high-confidence peptide, thus allowing the opportunity for a query feature to map to a low-confidence peptide, but with more stringent scoring.

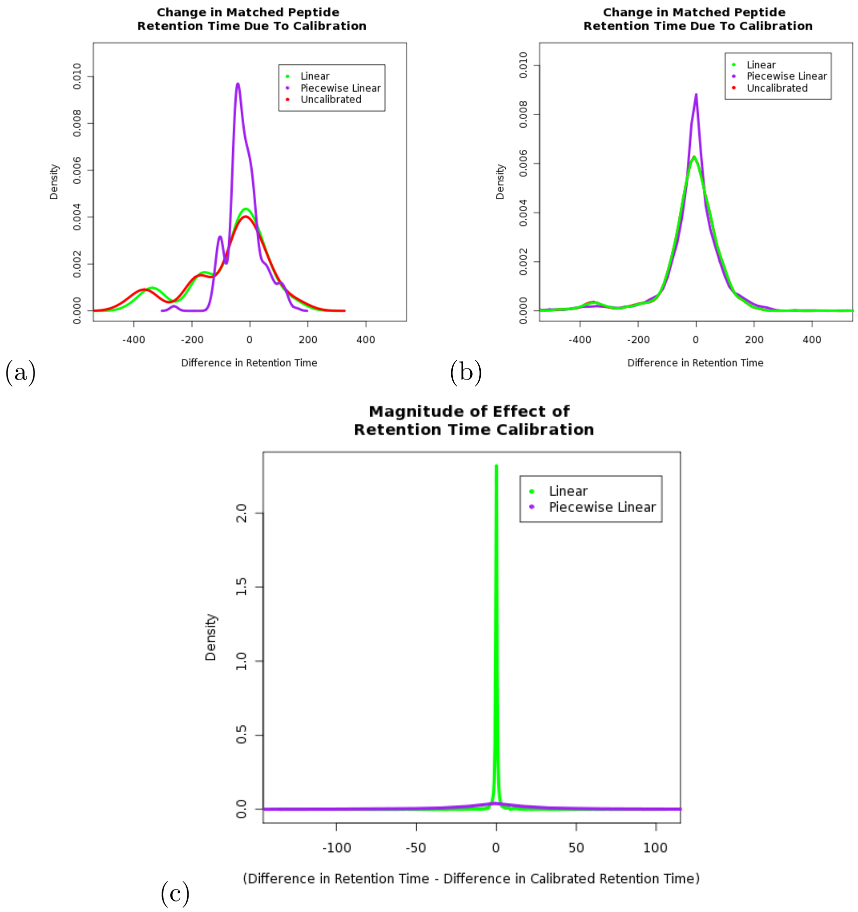

Figure 4: Effect of retention time calibration on the difference in retention time within the set of confidently identified peptides shared between a query dataset and a consensus database. (a) An example of the effect of calibration on a single run. (b) Effect of calibration, aggregated over 17 runs. (c) Magnitude of the effect of calibration.

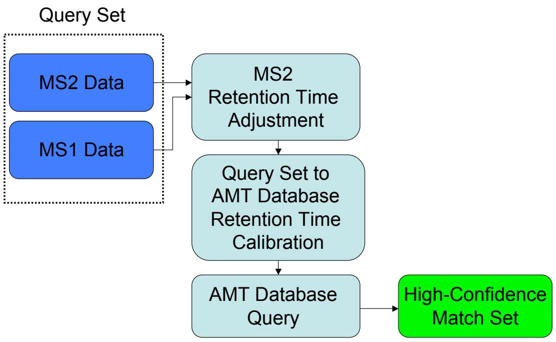
Query Set
MS2 Data
MS1 Data
MS2 Retention Time Adjustment
Query Set to AMT Database Retention Time Calibration
AMT Database Query
High-Confidence Match Set

Figure 5: Workflow of AMT database query.

Once this distribution has been determined for each identification in the island, the probability distribution corresponding to the island itself is taken to be the sum of these individual distributions. Then the probability density function for the island is the sum of the individual, normal densities, normalized such that it is a valid probability density function. This density allows us to calculate a match probability for any given mass and retention time; note that while it is defined over the dynamic range of the mass spectrometer, it is zero almost everywhere, except for in the island itself.

This "islandization" is done for every distinct sequence in the database. It is to these islands that a feature will be matched in the query process.

3.7 Database query

Having constructed a consensus database as detailed above, our algorithm effects queries to the database as follows. Given a set of MS2 and MS1 data for a single LC-MS/MS experiment, we proceed by first adjusting retention time of the MS2 identifications to match the peak of the elution profile of the nearest MS1 feature in the dataset. Here, we define ”nearest” using the usual Euclidean distance, weighted to reflect that we expect mass per charge accuracy to be on the order of 10^{-3} whereas we expect retention time deviation to be on the order of 10^{2}. Given an MS2 identification and MS1 feature with mass per charge deviation of x and retention time deviation of y, we score the distance between the two as:

$$d(A, B) = \sqrt{\frac{x^2}{.001} + \frac{y^2}{100}} \tag{11}$$

Following this correction of retention time in the MS2 data, we match the set of confidently identified peptides in the query to the peptides in the database and generate a warp function anchored by a selected subset of these peptides, as described above. After the query dataset has been calibrated to the database, MS1 features are searched against the database for confident matches.

3.7.1 Match search and scoring function

As detailed above, we attempt to match each MS1 feature to an island in the database. Consider a query feature F. A database lookup returns a set of islands X with a non-zero probability density at the mass (m_F) and retention time (rt_f) coordinates of F. We calculate the probability that F is a match to island X_i by calculating first the probability $P_{X_i}(m_f, rt_f)$ that the sequence corresponding to island X_i would be observed at m_f and rt_f. We then weight this probability to account for any overlapping islands that could also be candidates, using the following weighting factor:

$$W_i = \frac{P_{X_i}(m_f, rt_f) * P(X_i)}{\sum_{j=1}^{|X|}(P_{X_j}(m_f, rt_f) * P(X_j))} \tag{12}$$

where $P(X_i)$ is a prior on matching to island i, calculated from the area in (mass x retention time) space covered by the island relative to that covered by the dynamic range of the instrument.

This weighted probability is calculated for each candidate island, and the island with the highest probability is identified as the match.

4 Results

4.1 Strategy for validation of algorithm using MS2 identifications

We validated our algorithm by using MS2 identifications to label true and false positives and negatives in a query to a consensus database. Specifically, if a query feature with a corresponding MS2 identification as peptide A is given a database match of peptide A with a score that is greater than a user-defined threshold t, the match is a true positive. If the score is less than t, the match is a false negative. Alternatively, if the feature is given a database match of peptide B with a score greater than t, the match is a false positive; with a score less than t, the match is a true negative.

We make one additional distinction with regards to matches. Noting that a peptide with a modification causing a significant mass change should not be expected to match to the database, unless it were present in that exact modified state in one of the experiments used to construct the database. As the question of modification-based searches is beyond the scope of this project, we exclude any negative match with a deviation of observed mass from calculated mass of greater than .01 Da. This tolerance is more than enough to account for the

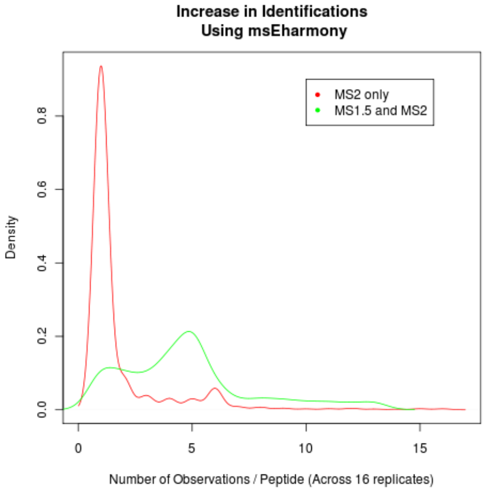

Figure 6: Reproducibility of peptide observations is significantly improved by AMT mapping. Initially, 95% of all peptides were observed only once across 16 technical replicates. Using our algorithm, we reduce this number to roughly 10%.

majority of the deviations that one would expect to see due to inaccuracy of mass estimation in the machine. (Cite IPAM work, show figures)

4.2 Validation procedure

Our testing database was constructed from 14 technical replicates of human serum samples (”BOB”). The database was constructed using piecewise linear retention time calibration, which consistently offered the best calibration performance in our experiments (see Retention time calibration, add delta delta t figure). In order to test the various heirarchical preprocessing methods, a different database was generated for each distance attribute, and the query was done for every one. (For the random distance attribute, the database was generated five times and query result statistics (will be) averaged somehow to create a representative result).

In order to generate a query dataset rich in both true positives and true negatives, we combined two experimental datasets into one. The first dataset was another technical replicate of the ”BOB” sample, whereas the second was a dataset of A431 cell lysate. Prior to querying the database, we determined that the only peptides shared by these two datasets were those corresponding to trypsin, the proteolytic enzyme used for digestion in our pipeline.

4.3 Evaluation of performance on samples of unknown composition

Having supported the validity of our approach through the above experiment, we generated LC-MS/MS datasets for a fractionated human serum sample.

Figure: before and after distribution of number of peptides/protein, score per protein,

5 Discussion

6 Supplementary

6.1 Computational resources

6.2 LC-MS/MS protocol