An Algorithm for Feature Finding in LC/MS Raw Data

Clemens Gröpl

Algorithmische Bioinformatik
Fachbereich Mathematik und Informatik
Freie Universität Berlin
Takustr. 9
14195 Berlin, Germany
groepl@inf.fu-berlin.de

Abstract. Liquid chromatography coupled with mass spectrometry is an established method in shotgun proteomics. A key step in the data processing pipeline is to transform the raw data acquired by the mass spectrometer into a list of features. In this context, a *feature* is defined as the two-dimensional integration with respect to retention time (RT) and mass-over-charge (m/z) of the eluting signal belonging to a single charge variant of a measurand (e. g., a peptide). Features are described by attributes like average mass-to-charge ratio, centroid retention time, intensity, and quality. We present a new algorithm for feature finding which has been developed as a part of a combined experimental and algorithmic approach to absolutely quantify proteins from complex samples with unprecedented precision. The method was applied to the analysis of myoglobin in human blood serum, which is an important diagnostic marker for myocardial infarction. Our approach was able to determine the absolute amount of myoglobin in a serum sample through a series of standard addition experiments with a relative error of 2.5%. It compares favorably to a manual analysis of the same data set since we could improve the precision and conduct the whole analysis pipeline in a small fraction of the time. We anticipate that our automatic quantitation method will facilitate further absolute or relative quantitation of even more complex peptide samples. The algorithm was implemented in the publicly available software framework OpenMS (www.OpenMS.de)

Keywords. Computational Proteomics, Quantitative Analysis, Liquid Chromatography, Mass Spectrometry, Algorithm, Software.

1 Introduction

Liquid chromatography in combination with mass spectrometry is increasingly being used for accurate and reliable identification and quantification of proteins and peptides in complex biological samples. Currently, the huge amount of data being produced and difficulties with absolute quantification of individual peptides are still a major problems with this method. In this work, we propose an HPLC-ESI-MS based approach for the absolute quantification of myoglobin in human blood serum and demonstrate the viability of this approach using reference material developed by the European Commission Joint Research Centre.

Myoglobin is a low-molecular weight (17 kDa) protein present in the cytosol of cardiac and skeletal muscle. Due to these characteristics, myoglobin appears in blood after tissue injury earlier than other biomarkers. It is thus of pivotal importance in clinical diagnosis as an early biomarker of myocardial necrosis. Currently, the National Academy of Clinical Biochemistry [1], the International Federation of Clinical Chemistry and Laboratory Medicine (IFCC) [2], and the American College of Emergency Physicians [3] have recommended the use of myoglobin as an early marker of myocardial necrosis.

Serum myoglobin has been used in routine practice since the development of automated non-isotopic immunoassays [4]. Unfortunately, results from different analytical procedures for myoglobin determination have shown significant biases as a result of a lack of assay standardization. Results from National External Quality Assurance Schemes showed a bias of over 100% for serum myoglobin [5,6]. Standardization of any measurand requires a reference measurement system, including a reference measurement procedure and (primary and secondary) reference materials (RM) [7]. A joint HPLC-MS/bioinformatics approach has been used to develop a reference method that can be used to standardize myoglobin assays[8,9] and subsequently to reduce the bias observed between commercial myoglobin assays, to standardize and harmonize measurement results, and to improve quality of diagnostic services.

Dagstuhl Seminar Proceedings 05471
Computational Proteomics
http://drops.dagstuhl.de/opus/volltexte/2006/534

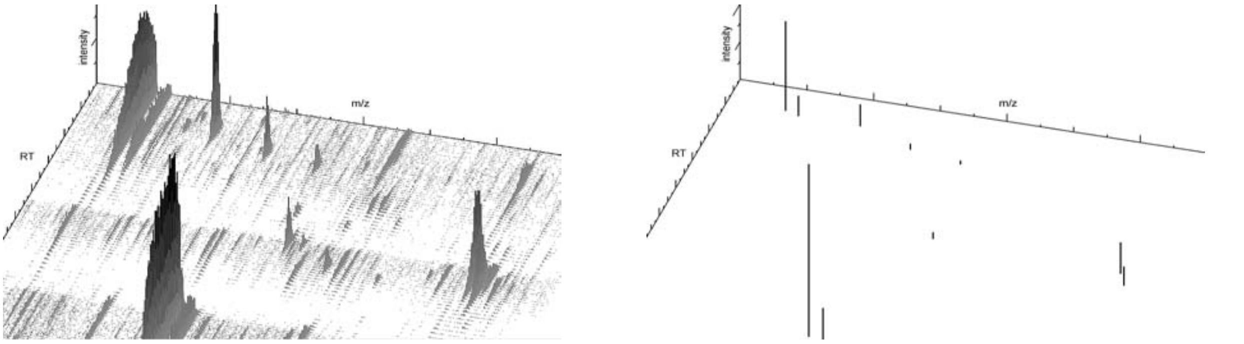

Fig. 1. Feature finding from a global perspective. A section of a LC/MS raw data map (left) and the features extracted from it by the feature finding algorithm (right).

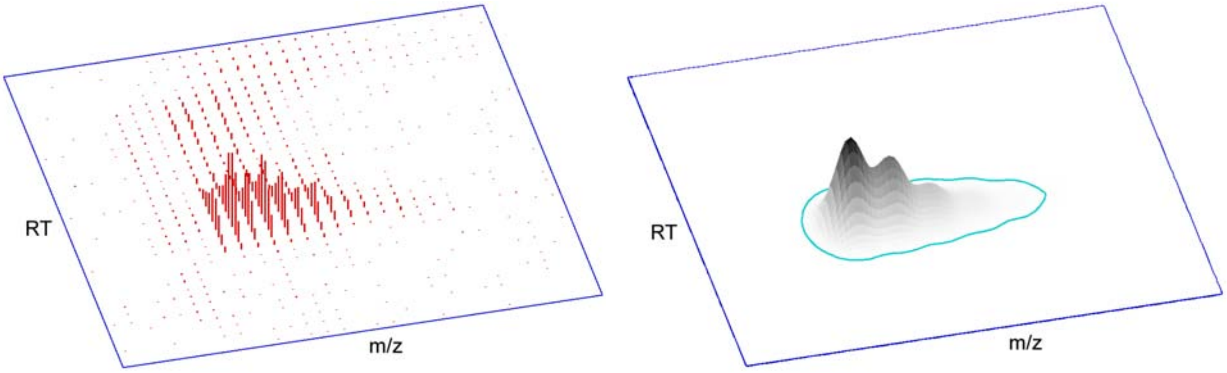

Fig. 2. A small part of the raw data (left) and a model adjusted to it (right).

HPLC-MS experiments produce a flood of data that is difficult to handle and analyze. It is necessary to reduce the raw instrument data to the essential *features* therein: the retention time, mass-to-charge ratio, and intensity of each peptide (or any other component) eluting from the column. The transformation of raw instrument data to so-called *feature maps* reduces the data volume and improves the running time of further analysis steps. Besides, it yields valuable secondary information which is not immediately evident from the raw data, such as charge estimates of peptides. The idea of two-dimensional raw data maps and the concept of peptide features is not novel [10,11] but has just started to emerge as a basis of quantitation and visualization of MS data.

Our new algorithm for feature finding identifies the raw data points belonging to a feature and fits a two-dimensional model to the extracted region of the input data. The model is based on a Gaussian elution profile along the retention time axis (or any other appropriate function) and a theoretical isotope pattern along the m/z axis. The output of the feature finder is a map of features, each identified by its RT and m/z coordinates and its intensity. Fig. 1 shows a part of the raw data file and the features found in it. Fig. 2 shows an example of how a feature model is adjusted to a (small) segment of the input data. Features can have annotations like the charge state and the quality of the model fit to the data.

The feature finding algorithm has been implemented in OpenMS [12], an object-oriented software platform for shotgun proteomics. OpenMS makes extensive use of generic programming techniques in C++ and thereby provides fast execution of programs and portable code. It is tested on different Linux platforms (e.g. Fedora Core 4, Scientific Linux 4 and Suse Linux 9 and 10) using 32 bit and 64 bit architectures. OpenMS itself is based on several other open-source libraries such as QT (TrollTech Inc.) and the GNU Scientific Library. OpenMS provides efficient data structures and algorithms for the analysis of multi-dimensional HPLC-MS data. It is available as open-source software under the lesser GNU public license (LGPL). Source code is available from the project web site at `www.OpenMS.de`.

In the experimental part of this work, myoglobin was separated from the highly abundant serum proteins by means of strong anion-exchange chromatography. Subsequently, the myoglobin-fraction was trypsinized and the peptides were analyzed by capillary reversed-phase high-performance liquid chromatography-electrospray ionization mass spectrometry (RP-HPLC-ESI-MS) using an ion-trap mass spectrometer operated in full-scan mode. In order to avoid quantification errors by artifacts in the sample preparation we added a constant amount of horse myoglobin to each sample in the additive series. We chose horse myoglobin as internal standard, since the tryptic horse peptides corresponding to their human counterparts elute roughly at the same time and are sufficiently different from the human peptides, such that corresponding peptides have different mass. To achieve an absolute quantification, known amounts of human myoglobin were added to aliquots of the sample. Each of the samples was measured in four replicates. The details of the experimental conditions have been described elsewhere [13,14].

The eluting peptides were detected in a quadrupole ion trap mass spectrometer (Esquire HCT from Bruker, Bremen, Germany) equipped with an electrospray ion source in full scan mode (m/z 500-1500). Each measurement consisted of ca. 1830 scans. The scans were roughly evenly spaced over the whole retention time window with an average of 0.9 scans per second. The sampling accuracy in mass-to-charge dimension was 0.2 Th. The instrument software was configured to store the measurement data in its most unprocessed form available (described below). The raw data was converted to flat files of size ca. 300 MB each using Bruker's CDAL library. (MzData was not an option at the time the analysis was performed. In the meantime, support for mzData has been added to OpenMS.) Quantitation of the myoglobin peptides in the serum sample was then conducted as described in the next section.

2 Feature finding

By the term *feature finding* we refer to the process of transforming a file of raw data as acquired by the mass spectrometer into a list of features. Here a *feature* is defined as the two-dimensional integration with respect to retention time (RT) and mass-over-charge (m/z) of the eluting signal belonging to a single charge variant of a peptide. Its main attributes are average *mass-to-charge ratio*, centroid *retention time*, *intensity*, and a *quality* value.

In our study, the raw data set exported from the instrument consisted of *profile spectra*, but no baseline removal or noise filtering had been performed. In particular, no *peak picking* had taken place (where peak picking denotes the process of transforming a profile spectrum to a stick spectrum by grouping the raw data points into one-dimensional "peaks", which have a list of attributes similar to those of features). Features are commonly generated from raw data by forming groups with respect to one dimension after the other, thereby reducing the dimensionality one by one. However better results can be achieved using a genuinely two-dimensional approach.

Theoretical model of features Each of the chemical elements contributing to the sum formula of a peptide has a number of different isotopes occurring in nature with certain abundance[15]. The mass differences between these isotopes can be approximated by multiples of 1.000495 Da up to the imprecision of the instrument. Given these parameters, and the empirical formula of a peptide, one can then compute its the theoretical stick spectrum. In our study, such an *isotope pattern* has 3-6 detectable masses. Since the lightest isotopes are by far most abundant for the elements C, H, N, O, and S, it is common to use the corresponding stick as a reference point, called *monoisotopic peak.*

If the peaks for consecutive isotope variants are clearly separated in the profile spectra, they can be picked individually and combined to isotope patterns afterward. However, in our raw data set, having a sampling accuracy of 0.2 Th, this is the case only for charge 1. Already for charge 2 the profiles of peaks overlap to such an extent that such a two-step approach is not feasible. Moreover, as the mass and charge increases, the whole isotope pattern at a given fixed value of m/z becomes more and more bell-shaped and eventually converges to a normal distribution. In our case, neither extreme is a good approximation. Instead, we model the m/z profile of the raw data points belonging to a single isotope pattern by a mixture of normal distributions, as shown in Fig. 4 (left).

One of our design goals was that the algorithm should not rely on information about specific peptides given in advance. Therefore the empirical formula of a peptide of a given mass is approximated using so-called *averagines*, that is, average atomic compositions taken from large protein databases. For example, an averagine of mass 1350 contains "59.827" C atoms, "4.997" N atoms etc. We calculated the isotopic distributions of the tryptic myoglobin peptides and found that they are well approximated by averagines (see Fig. 3 in the Appendix). If necessary, an even better approximation could be used that takes into account that the peptides are digested by a specific protease (in our case trypsin), which results in a bias of the amino acids at the end of a peptide. The theoretical m/z distribution is then obtained by convoluting the sticks of the theoretical isotope pattern with a normal distribution to simulate the measurement inaccuracy.

The left part of Fig. 4 shows the effect of the smoothing width on an averagine isotope distribution at mass 1350 Da.

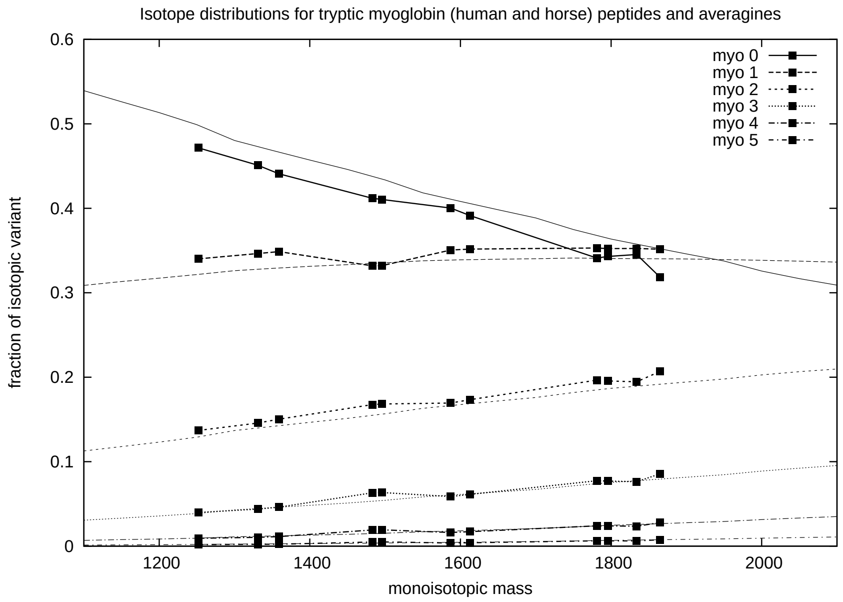

Fig. 3. The comparison shows that the isotope distributions for tryptic myoglobin (human and horse) peptides are well approximated by averagines.

The signal of a single charge variant of a peptide extends over a certain interval of retention time. As a model for the retention profile, we currently use a normal distribution with variable width. More sophisticated models that incorporate fronting and tailing effects that are observed especially for high intensity peaks are known (see e.g. [16,17]). These shall be investigated in subsequent work.

It is natural to assume that isotope pattern and elution profile are independent from each other. Consequently, our theoretical model for features is a product of a model for the m/z domain and a model for the retention time domain. An example of a two-dimensional feature model is shown in the right part of Fig. 4.

Algorithm The algorithm for feature finding consists of four main phases:

1. *Seeding.* Data points with high signal intensity are chosen as starting points of the feature detection.
2. *Extension.* The region around each seed is conservatively extended to include all potential data points belonging to the feature.
3. *Modeling.* A two-dimensional statistical model of the feature is calculated.
4. *Adjusting.* The tentative region is then adjusted to contain only those data points that are compatible with the model.

The modeling and adjusting phases can potentially have a large effect on the statistical model of the feature. Therefore we re-calculate the statistical model and apply the adjusting phase for a second time. That is, we repeat phases 3 and 4. A feature is reported only if its quality value is above a user-specified value. Input and output of the algorithm is illustrated in Fig. 2 and 1. We will now go through the four stages in more detail.

Seeding After the relevant portion of the input file (a retention time window) has been read into main memory, it is (effectively) sorted according to the intensity of the raw data points. In a greedy fashion we consider the most intense data point as a so-called *seed* for the formation of a feature. This is motivated by the fact that the most intense data points are very likely to belong to a feature. A seed is considered for the next phase (extension) only if it is not already contained in a feature. We stop when the seed intensity falls below a threshold. (The actual implementation does not sort the raw data physically, but uses a priority

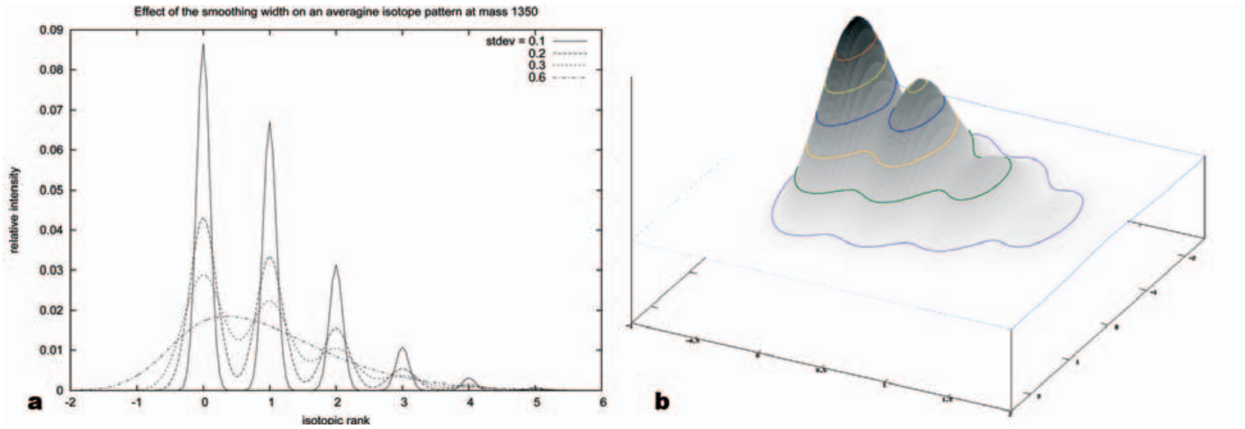

Fig. 4. (a) Effect of the smoothing width on the theoretical isotope distribution of a peptide of mass 1350 Da. Increasing the smoothing width can emulate the effect of low instrument resolution. **(b)** A two-dimensional model for a feature of charge two and mass 1350 Da.

queue instead, from which the seeds are extracted in order of intensity. This way the low-intensity data points need not be sorted.)

Extension Given a seed, we conservatively determine a *region* around it that very likely contains all data points of the feature. The region grows in all directions simultaneously, preferring the strongest raw data points near the *boundary*.

Initially, the region is empty and the boundary set only contains the seed. In each step, a data point in the boundary is selected and moved into the region. Then the boundary is updated by exploring the neighborhood of the selected data point. The selected data point is chosen based on a *priority* value, and the boundary set is implemented as a priority queue (This should not to be confused with the priority queue used for seeding). The priority of a data point is never decreased by an update of the boundary. If the updated priority of a neighboring data point exceeds a certain threshold, it is moved into the boundary. The seed extension stops when the intensity of all data points in the boundary falls below a certain threshold.

The priority values of raw data points are not identical to their intensities. Their purpose is to control the growth of the feature, such that a number of constraints are met: The boundary should be a relatively 'thin' layer around the region. It should be resistant to noise in the data and allow for 'missing' raw data points. Data points close to the region should be preferred. We compute the priority values as follows: When a data point is extracted from the priority queue, we explore a cross-like neighborhood around it in four directions ("m/z up", "m/z down", "RT up", "RT down"). The priority is calculated by multiplying the intensity of the data point with a certain function of the distance from the extracted point. Currently we use triangular shapes that go to zero at distance 2.0 s in RT and 0.5 Th in m/z.

The criteria controlling the growth of the boundary and the stopping of the seed extension are adapted during the seed extension process based on the information gathered so far. This is done as follows:

1. We compute an intensity threshold for stopping the extension phase. The threshold is a fixed percentage of the fifth-largest intensity (we do not choose the largest for robustness reasons).

2. We maintain a running average of the data point positions, weighted by their intensities. The neighborhood of a boundary point is not further explored if it is too distant from the centroid of the feature. This is important to avoid collecting low intensity data points (baseline) when the seed has a relatively low intensity.

Modeling Given a region, we fit a two-dimensional statistical *model* to it. The point intensity of the two-dimensional model is the product of two one-dimensional models, one explaining the isotope pattern and one explaining the elution profile. The raw data points are considered empirical samples from this distribution.

The fit in m/z dimension examines different distributions implied by charge states in a range provided by the user (currently 1 to 3). For each charge, we try a number of smoothing widths of the averagine isotope pattern (currently 0.15, 0.2, 0.25, 0.3, and 0.35 Th). The correct charge state is likely to provide the best fit to the data points. In addition we also fit a normal distribution using maximum likelihood estimators. As a measure of confidence in the charge prediction we report on the distance to the fit with the second best charge hypothesis. The fit in retention time dimension uses a maximum likelihood normal approximation.

The quality of fit of the data against a model is measured using the squared correlation

$$\frac{(\sum_x f(x)g(x))^2}{\sum_x f(x)^2 \sum_x g(x)^2},$$

where f = observed, g = model, x = data point position. Other methods like the χ^2-test have already been implemented in OpenMS and can be used if desired.

Adjusting At this stage of the algorithm, we have a region of data points and a statistical model for it. But the region is very likely to contain data points not belonging to the feature. To discard those, and keep only those data points which are consistent with the statistical model, we re-assemble the data points contained in the feature similar to the extension phase using a modified priority that takes the model into account. Using a model is the main difference of this phase compared to the extension phase.

To combine the theoretical and observed intensities, we use the geometric mean of the observed intensity of a data point and its prediction by the model as the priority for extension. This is based on the following considerations: Since the normal distribution decays exponentially at its tails, data points not explained by the model are effectively cut off. Moreover the geometric mean compensates for inaccuracies when the intensity of the data points decays faster than predicted. Of course many other strategies for adjusting can be considered and should be tested in the future.

3 Results

We performed a series of 32 RP-HPLC-ESI-MS measurements as described above (four replicates of eight different spiked concentrations). The quantification was performed using the eleventh tryptic peptide of human myoglobin, HGATVLTALGGILK, here denoted *T11hu*, with and without the tenth tryptic peptide of horse myoglobin, HGTVVLTALGGILK, denoted *T10ho*, as an internal standard. These two peptides are sufficiently similar to behave similarly in terms if ionization and still can be separated easily in both RT and m/z dimension.

To assess the quality of the automated analysis, we also report the results of a manual expert analysis of the same data set that was performed earlier by Bettina Mayr [18]. Manual quantification was performed using the Bruker instrument software and Microsoft Excel. The peak areas were calculated from extracted ion chromatograms with an isolation width of ±0.5 Da after smoothing with a Gauss filter.

Automated analysis was performed using the features found by the algorithm described in Section 2 without further manual intervention. We provided approximate masses and approximate retention times of the peptides used for quantification and restricted the feature finding to a large window of the raw data (RT = 900–1600 sec, m/z = 600–1000 Th) to speed up the process. The algorithm then identified features in the 32 data sets, integrated the feature areas and performed the statistical analysis detailed in the following table:

Method	OpenMS	Manual
Computed concentration [ng/μl]	**0.474**	**0.382**
Lower bound of 95% interval [ng/μl]	0.408	0.315
Upper bound of 95% interval [ng/μl]	0.545	0.454
True value [ng/μl]	*0.463*	*0.463*
Relative deviation from true value [%]	**+2.46**	**−17.42**
Lower bound of 95% interval [%]	−11.82	−32.04
Upper bound of 95% interval [%]	+17.62	−1.84

Both manual and automated analysis were able to estimate the true concentration of myoglobin in the serum sample with very high precision. However the manual analysis of these large data sets amounts to half a day of work, whereas the automated analysis of the data sets could be performed in less than 2 hours on a 2.6 GHz Pentium IV machine with 1 GB of RAM running Linux. The regression results are shown in Fig. 5.

The results of several additional independent studies for myoglobin quantification all yielded relative quantification errors below 8% (data not shown). Automated analysis of the data sets yielded comparable or better results in these experiments.

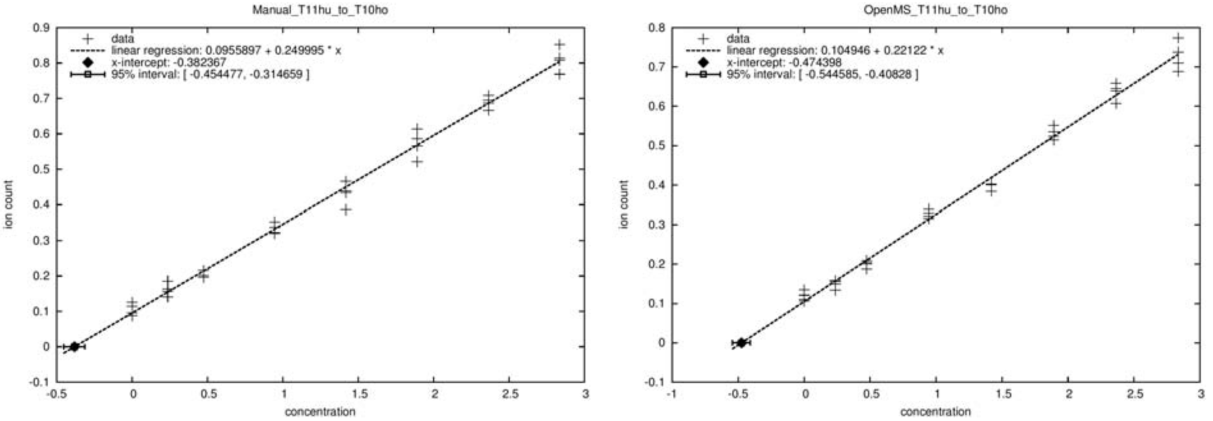

Fig. 5. Regression results for manual (left) and automated (right) analysis of myoglobin in serum samples. The automated analysis yields smaller standard deviations between replicates of the same sample and tighter error bounds on the absolute concentration computed.

4 Conclusion and Outlook

We have presented a new algorithm for feature finding in LC/MS raw data. Using it, we were able to determine the absolute myoglobin content of human serum plasma with very high precision.

Since the presentation of the talk, the implementation has been fully redesigned. For example, the four key steps have become individual classes, and the data structure for LCMS maps has been factored out. See Fig. 6. We are currently investigating alternatives for the individual stages of the algorithm (seeding, extending, modeling, adjusting) and improving the running time.

References

1. Wu, A., Apple, F., Gibler, W., Jesse, R., Warshaw, M., Valdes, J.R.: National academy of clinical biochemistry standards of laboratory practice: recommendations for use of cardiac markers in coronary artery diseases. Clinical Chemistry **45** (1999) 110–121
2. M., P., Apple, F., Christenson, R., Dati, F., Mair, J., Wu, A.: Use of biochemical markers in acute coronary syndromes. Clin. Chem. Lab. Med. **37** (1999) 687–693
3. Fesmire, F., Campbell, M., Decker, W., Howell, J., Kline, J.: Clinical policy: critical issues in the evaluation and management of adult patients presenting with suspected acute myocardial infarction or unstable angina. Ann. Emerg. Med. **35** (2000) 521–544
4. Wu, A.H., Laios, I., Green, S., Gornet, T.G., Wong, S.S., Parmley, L., Tonnesen, A.S., Plaisier, B., Orlando, R.: Immunoassays for serum and urine myoglobin: myoglobin clearance assessed as a risk factor for acute renal failure. Clin Chem **40** (1994) 796–802
5. College of American Pathologists: Cardiac markers survey (2003) Northfield, IL.
6. Panteghini, M.: Recent approaches to the standardization of cardiac markers. Scand. J. Clin. Lab. Invest. **61** (2001) 95–102
7. Panteghini, M. In: Standardization of cardiac markers. Totowa (2003) 213–229
8. Dati, F., Linsinger, T., Apple, F., Christenson, R., Mair, J., Ravkilde, J., et al.: IFCC project for standardization of myoglobin immunoassays. Clin. Chem. Lab. Med. **40** (2002) S311
9. Dati, F., Panteghini, M., Apple, F., Christenson, R., Mair, J., Wu, A.: Proposals from the IFCC committee on standardization of markers of cardiac damage (C-SMCD): strategies and concepts on standardization of cardiac marker assays. Scand. J. Clin. Lab. Invest. **230** (1999) 113–123
10. Leptos, K.C., Sarracino, D.A., Jaffe, J.D., Krastins, B., Church, G.M.: MapQuant: Open-source software for large-scale protein quantification. Proteomics **6** (2006) 1770–1782
11. Radulovic, D., Jelveh, S., Ryu, S., Hamilton, T.G., Foss, E., Mao, Y., Emili, A.: Informatics platform for global proteomic profiling and biomarker discovery using liquid chromatography-tandem mass spectrometry. Molecular and Cellular Proteomics **3** (2004) 984–997
12. Kohlbacher, O., Reinert, K., Gröpl, C., Lange, E., Pfeifer, N., Schulz-Trieglaff, O., Sturm, M.: TOPP – The OpenMS Proteomics Pipeline. submitted (2006)

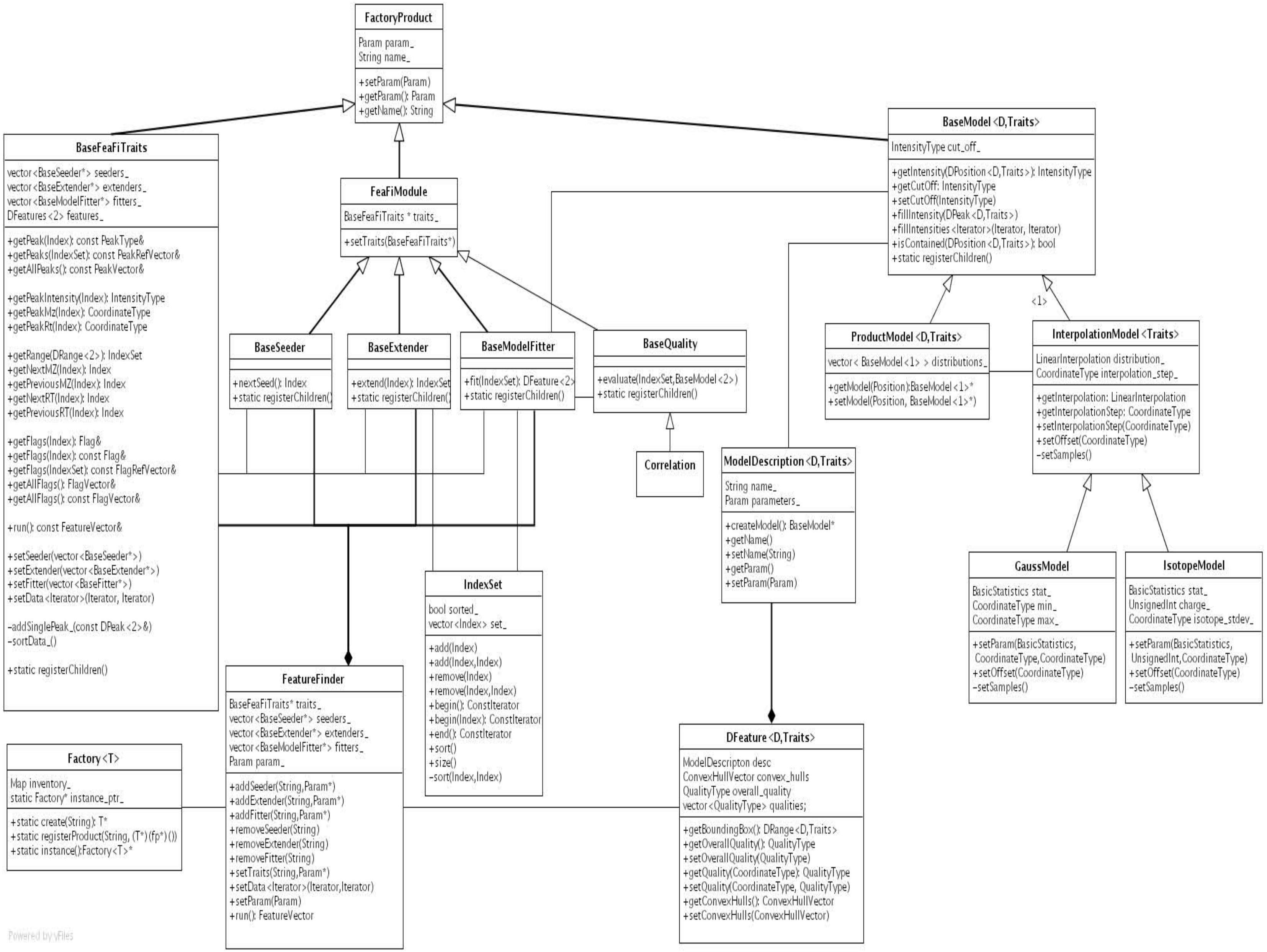
FactoryProduct
Param param_
String name_
+setParam(Param)
+getParam(): Param
+getName(): String
BaseFeaFiTraits
vector<BaseSeeder*> seeders_
vector<BaseExtender*> extenders_
vector<BaseModelFitter*> fitters_
DFeatures<2> features_
+getPeak(Index): const PeakType&
+getPeaks(IndexSet): const PeakRefVector&
+getAllPeaks(): const PeakVector&
+getPeakIntensity(Index): IntensityType
+getPeakMz(Index): CoordinateType
+getPeakRt(Index): CoordinateType
+getRange(DRange<2>): IndexSet
+getNextMZ(Index): Index
+getPreviousMZ(Index): Index
+getNextRT(Index): Index
+getPreviousRT(Index): Index
+getFlags(Index): Flag&
+getFlags(Index): const Flag&
+getFlags(IndexSet): const FlagRefVector&
+getAllFlags(): FlagVector&
+getAllFlags(): const FlagVector&
+run(): const FeatureVector&
+setSeeder(vector<BaseSeeder*>)
+setExtender(vector<BaseExtender*>)
+setFitter(vector<BaseFitter*>)
+setData<Iterator>(Iterator, Iterator)
-addSinglePeak_(const DPeak<2>&)
-sortData_()
+static registerChildren()
FeaFiModule
BaseFeaFiTraits * traits_
+setTraits(BaseFeaFiTraits*)
BaseSeeder
+nextSeed(): Index
+static registerChildren()
BaseExtender
+extend(Index): IndexSet
+static registerChildren()
BaseModelFitter
+fit(IndexSet): DFeature<2>
+static registerChildren()
BaseQuality
+evaluate(IndexSet,BaseModel<2>)
+static registerChildren()
Correlation
BaseModel<D,Traits>
IntensityType cut_off_
+getIntensity(DPosition<D,Traits>): IntensityType
+getCutOff: IntensityType
+setCutOff(IntensityType)
+fillIntensity(DPeak<D,Traits>)
+fillIntensities<Iterator>(Iterator, Iterator)
+isContained(DPosition<D,Traits>): bool
+static registerChildren()
<1>
ProductModel<D,Traits>
vector< BaseModel<1> > distributions_
+getModel(Position):BaseModel<1>*
+setModel(Position, BaseModel<1>*)
InterpolationModel<Traits>
LinearInterpolation distribution_
CoordinateType interpolation_step_
+getInterpolation: LinearInterpolation
+getInterpolationStep: CoordinateType
+setInterpolationStep(CoordinateType)
+setOffset(CoordinateType)
-setSamples()
ModelDescription<D,Traits>
String name_
Param parameters_
+createModel(): BaseModel*
+getName()
+setName(String)
+getParam()
+setParam(Param)
GaussModel
BasicStatistics stat_
CoordinateType min_
CoordinateType max_
+setParam(BasicStatistics,
CoordinateType,CoordinateType)
+setOffset(CoordinateType)
-setSamples()
IsotopeModel
BasicStatistics stat_
UnsignedInt charge_
CoordinateType isotope_stdev_
+setParam(BasicStatistics,
UnsignedInt,CoordinateType)
+setOffset(CoordinateType)
-setSamples()
IndexSet
bool sorted_
vector<Index> set_
+add(Index)
+add(Index,Index)
+remove(Index)
+remove(Index,Index)
+begin(): ConstIterator
+begin(Index): ConstIterator
+end(): ConstIterator
+sort()
+size()
-sort(Index,Index)
Factory<T>
Map inventory_
static Factory* instance_ptr_
+static create(String): T*
+static registerProduct(String, (T*)(fp*)())
+static instance():Factory<T>*
FeatureFinder
BaseFeaFiTraits* traits_
vector<BaseSeeder*> seeders_
vector<BaseExtender*> extenders_
vector<BaseModelFitter*> fitters_
Param param_
+addSeeder(String,Param*)
+addExtender(String,Param*)
+addFitter(String,Param*)
+removeSeeder(String)
+removeExtender(String)
+removeFitter(String)
+setTraits(String,Param*)
+setData<Iterator>(Iterator,Iterator)
+setParam(Param)
+run(): FeatureVector
DFeature<D,Traits>
ModelDescripton desc
ConvexHullVector convex_hulls
QualityType overall_quality
vector<QualityType> qualities;
+getBoundingBox(): DRange<D,Traits>
+getOverallQuality(): QualityType
+setOverallQuality(QualityType)
+getQuality(CoordinateType): QualityType
+setQuality(CoordinateType, QualityType)
+getConvexHulls(): ConvexHullVector
+setConvexHulls(ConvexHullVector)
Powered by yFiles

Fig. 6. UML class diagram of the redesigned FeatureFinder application in OpenMS.

13. Gröpl, C., Lange, E., Reinert, K., Kohlbacher, O., Sturm, M., Huber, C., Mayr, B., Klein, C.: Algorithms for the automated absolute quantification of diagnostic markers in complex proteomics samples. In Berthold, M., Glen, R., Diederichs, K., Kohlbacher, O., Fischer, I., eds.: Proceedings of the 1st Symposium on Computational Life Sciences (CLS 2005). Volume 3695 of Lecture Notes in Bioinformatics (LNBI)., Springer (2005) 151–161
14. Mayr, B.M., Kohlbacher, O., Reinert, K., Sturm, M., Gröpl, C., Lange, E., Klein, C., Huber, C.G.: Absolute myoglobin quantitation in serum by combining two-dimensional liquid chromatography-electrospray ionization mass spectrometry and novel data analysis algorithms. J. Proteome Res. **5** (2006) 414–421
15. de Hoffmann, E., Charette, J., Stroobant, V.: Mass Spectrometry. 2nd edn. John Wiley and Sons (2001)
16. Marco, V.B.D., Bombi, G.G.: Mathematical functions for the representation of chromatographic peaks. Journal of Chromatography A **931** (2001) 1–30
17. Pai, S.C.: Temporally convoluted gaussian equations for chromatographic peaks. Journal of Chromatography A **1028** (2004) 89–103
18. Mayr, B.M.: Die Kopplung der Flssigchromatographie mit der Elektrospray-Ionisations- Massenspektrometrie als Werkzeug fr die Genomanalyse und die Quantitative Proteomforschung. PhD thesis, Universitt des Saarlandes (2005)