Locality-sensitive hashing enables signal classification in high-throughput mass spectrometry raw data at scale

Konstantin Bob¹, David Teschner¹, Thomas Kemmer¹, David Gomez-Zepeda², Stefan Tenzer², Bertil Schmidt¹, and Andreas Hildebrandt^{1,3}

¹Institute of Computer Science, Johannes Gutenberg University, Mainz

²Institute for Immunology, University Medical Center of the Johannes Gutenberg University, Mainz

³Corresponding author: andreas.hildebrandt@uni-mainz.de

July 1, 2021

Mass spectrometry is an important experimental technique in the field of proteomics. However, analysis of certain mass spectrometry data faces a combination of two challenges: First, even a single experiment produces a large amount of multi-dimensional raw data and, second, signals of interest are not single peaks but patterns of peaks that span along the different dimensions. The rapidly growing amount of mass spectrometry data increases the demand for scalable solutions. Existing approaches for signal detection are usually not well suited for processing large amounts of data in parallel or rely on strong assumptions concerning the signals properties. In this study, it is shown that locality-sensitive hashing enables signal classification in mass spectrometry raw data at scale. Through appropriate choice of algorithm parameters it is possible to balance false-positive and false-negative rates. On synthetic data, a superior performance compared to an intensity thresholding approach was achieved. The implementation scaled out up to 88 threads on real data. Locality-sensitive hashing is a desirable approach for signal classification in mass spectrometry raw data. Generated data and code are available at <https://github.com/hildebrandtlab/mzBucket>. Raw data is available at <https://zenodo.org/record/5036526>.

29 Background

30 Mass spectrometry in proteomics

31 Valuable information for medicine and design of new drugs for several severe diseases [1] are
32 expected to be gained by new discoveries in proteomics [2][3][4], the field that studies proteins
33 experimentally on a large scale. An experimental technique commonly used in proteomics is
34 mass spectrometry (MS) [5], which allows to separate ionized molecules by their mass-to-
35 charge ratio (m/z) and which can be combined with measurement of other physical and
36 chemical properties.

37 The overall goal in an untargeted MS-based proteomics experiment is to identify and
38 quantify as many proteins as possible in a given sample with a high quantitative performance
39 in terms of precision and reproducibility. In particular, in bottom-up proteomics proteins are
40 digested into peptides first and then those peptides are measured.

41 If only the mass-to-charge ratio of a large, complex sample of peptides was measured,
42 the resulting signal would be highly convoluted as many peptides have the same or a very
43 similar m/z . In order to minimize overlapping signals and to get further information on the
44 molecules measured, mass spectrometers are coupled with a previous separation device based
45 on orthogonal (ideally) physical and chemical properties. Typically, the peptides are first
46 separated using liquid chromatography (LC), where molecules are separated and gradually
47 eluted at a certain retention time range, e.g., based on their polarity in the commonly used
48 reversed phase (RP) chromatography. More recently, ion mobility (IMS) has become widely
49 accessible in commercial mass spectrometers as an extra dimension of separation, where
50 ionized molecules are continually separated based on their shape and size before being analyzed
51 in the MS. For instance, in LC-IMS-MS [6][7][8] the retention time and mobility dimensions
52 are recorded in addition to m/z . Figure 1 shows the experimental workflow in the left column.

53 Finally, in many experimental setups two types of mass spectra are recorded: So-called MS1
54 spectra of all the ions, typically for localization of signals of interest denominated precursors
55 and MS2 spectra, where selected (or unselected) precursors are fragmented and the obtained
56 ion patterns (fragmentation spectra) are typically used for identification.

57 The first step in the analysis of MS1 spectra is to identify regions of interest. These signals
58 form characteristic patterns in the raw data. More precisely, molecules produce so-called
59 isotopic patterns [9] that are caused by the occurrence of different isotopes in the chemical
60 elements. Thus, one expects a pattern of evenly spaced peaks, where the distance between
61 consecutive peaks varies inversely according to the charge state of the molecule (i.e., a spacing
62 of $\approx 1m/z$ for $z = 1$, $\approx 0.5m/z$ for $z = 2$, etc.).

63 The distribution of peak intensity within an isotopic pattern depends on the chemical
64 elements present in a molecule and their respective distribution of isotopes. Although it is
65 possible to deal with the resulting combinatorial complexity [10], often simpler approaches
66 that assume a typical chemical composition are used to model the distribution of the peak
67 intensity. A common example is the so-called *averagine* model [11].

68 Due to the coupling with the separation devices, the signals of interest are expected to
69 occur repeatedly over time with the same pattern along the mass axis. Figure 1 shows an
70 example of a resulting signal in the center and right column.

71 Problem statement and challenges

72 The problem to be solved can now be formulated as follows: Given MS1 spectra of a high-
73 throughput setup with additional dimensions of separation, classify whether the signals belong
74 to a region of interest or not.

75 On top of the signal processing challenge, another technical problems arises: By introducing
76 additional dimensions of separation (such as LC and IMS), the sizes of single data sets
77 increase. Combined with a growth in the number of data sets measured, the storage used
78 for mass spectrometry data has thus grown tremendously over the past years (cf. Figure 2)
79 and is expected to grow further. Consequently, dealing with mass spectrometry raw data
80 will eventually make the usage of Big Data technologies necessary. This means that data
81 will be stored and processed in a distributed manner, which in turn restricts the algorithms
82 applicable.

83 Locality-sensitive hashing

84 An often used Big Data method for the comparison of high-dimensional data is locality-
85 sensitive hashing (LSH) [12][13]. In particular, it is a generic algorithm for finding similar
86 pairs of data points (by some measure) in linear runtime. However, this reduction of runtime
87 comes at the prices that the algorithm is probabilistic in nature.

88 Owing to its wide applicability, the technique is widely used in different fields, including
89 image retrieval [14], pattern recognition [15], and genome analysis [16][17].

90 Related work

91 In the particular context of mass spectrometry, LSH has been used for looking up peptide
92 sequences in databases [18][19], to cluster different spectra for MS1 spectra on LC-MS data
93 [20], and for fast database lookup on MS2 spectra [21].

94 Previous approaches to signal detection in mass spectrometry raw data either rely on
95 assumptions concerning the isotopic distribution [22] or are based on deep learning [23] and
96 thus lack interpretability.

97 Concerning the processing of larger data sets, established tools like MaxQuant [24] partially
98 bypass large data sizes by using only parts of the data, i.e., by only looking at every 4th
99 spectrum by default [25].

100 To the best of our knowledge, signal classification by means of LSH for mass spectrometry
101 raw data has not been treated publicly.

102 Results

103 Approach

104 Our approach exploits the fact that all signals from a given region of interest are expected
105 to be similar to each other [25], while noise is assumed to be much more random. Thus,
106 classification of the signal is achieved by deciding whether there are similar signals present.

107 Finding similar objects is achieved by locality-sensitive hashing, as it allows to leverage parallel
108 computation.

109 The overall scheme of the approach, see Figure 3 for illustration, is the following: A
110 mass spectrometry raw data set is considered to be a set of mass axes for each possible
111 replicated measurement, that is, each retention time for LC-MS data or each retention time
112 and mobility measurement for LC-IMS-MS data, respectively. These mass axes are then cut
113 into small intervals, called windows. For each window several hash values are computed. If
114 two windows have the same hash value they are said to collide. The classification into “true”
115 signal and noise used the following criterion: If a peak lies within a window that collided with
116 any other window it is considered “true” signal, otherwise noise. To facilitate the lookup of
117 collisions, a second map structure is used that maps hash values to their respective number
118 of occurrences.

119 Advantages of the approach

120 As the hash function of each window can be computed and checked independently of the
121 other windows, the algorithm is embarrassingly parallel in nature. By using an augmented
122 LSH with n AND connectives and m OR connectives, the algorithm allows for tuning of
123 false-positive and false-negative rates.

124 In particular, our approach assumes no model or distributions for the signal shapes, only
125 similarity. Thus, we are able to distinguish different isotopic patterns as true signal, regardless
126 of the composition, size and charge of the ionized molecule.

127 Signal classification capability

128 Figure 4 shows the receiver operating characteristic (ROC) of a classification task on synthetic
129 data. By varying the amplification parameters m and n^1 of the LSH, different types of
130 discrimination can be achieved, depending on the goal of the user².

131 The achieved performance is in accordance with expected behaviour when comparing the
132 implied similarity thresholds in Figure 5. When setting a low threshold, e.g., $(m, n) = (30, 22)$
133 (corresponding to the blue line in Figure 5), almost all windows are considered signal, resulting
134 in a true-positive and false-positive rate of almost one.

135 A stricter discrimination with $(m, n) = (30, 32)$ improves the performance significantly.
136 This in turn indicates that the typical similarity measure of the data is in the area where the
137 orange line in Figure 5 has a high slope.

138 Using a very high threshold, e.g., $(m, n) = (30, 64)$ (corresponding to the green line in
139 Figure 5), some windows are lost, resulting in a lower true-positive rate. As the false-positive
140 rate shrinks as well, this setting may be employed as a strong filter to reduce large data sets.

141 The performance of our approach was compared to the common approach of filtering signal
142 by an intensity threshold. While the overall behavior of its ROC curve is similar to the one
143 of our approach, the performance is considerably better.

¹A plot with more pairs of m and n can be found in the Supplementary Information. The best parameters still lie on the implied ROC curve.

²Note that the number of windows in the data set influences the collision probabilities as well.

144 Several synthetic data sets with different noise characteristics were created and all main
145 findings persisted, see Supplementary Information.

146 Scalability

147 Figure 6 shows the scalability of the approach for different choices of m and n on parts of a
148 measured data set. In this double-logarithmic plot, the wall-clock time of the implementation
149 roughly follows a linearly decreasing function of the number of threads used. This shows that
150 the implementation scales out well.

151 Discussion

152 While the evaluation on synthetic data showed that locality-sensitive hashing could be used in
153 a promising way to detect signals in mass spectrometry raw data, two challenges are expected
154 when applying the method to real world data.

155 First, in some mass spectrometry setups so-called chemical noise is present. These are sig-
156 nals that originate in chemical impurities in the measurement workflow and are characterized
157 by repeated occurrences. Thus, the chemical noise is self-similar and would be classified as a
158 “true” signal accordingly by this approach.

159 Finally, signals with a low relative intensity to noise peaks could be lost, as the similarity
160 measure on which the LSH is based drops.

161 The inability to remove noise peaks from within an isotopic pattern is typically mitigated by
162 the fact that subsequent processing steps like feature finding can take the multidimensional
163 signal shape into account which facilitates the removal of remaining noise peaks.

164 Testing the approach on real world data is desirable, but we could not find a gold-standard
165 ground-truth peak-level annotation of data.

166 Conclusions

167 Due to the rapidly growing amount of mass spectrometry data, the analysis of mass spectrom-
168 etry raw data could greatly benefit from Big Data methods, most notably implying distributed
169 data storage and highly scalable algorithms.

170 In this study we showed that locality-sensitive hashing is a desirable approach for signal
171 classification in mass spectrometry raw data. It allows for scalability and provides an approach
172 to signal classification that has a strong focus on self-similarity rather than model assumptions
173 as an intrinsic property of the data.

174 We propose an implementation using a Big Data framework, such as Apache Spark [26],
175 to facilitate testing on many large data sets from different types of mass spectrometry mea-
176 surements.

177 Methods

178 Mass axis windows

179 For the types of mass spectrometry data considered here, the setup of the mass spectrom-
 180 eter gives rise to a hierarchical structure of the data. In the case of LC-IMS-MS, data is
 181 continuously acquired as individual scans across the three dimensions. For each retention
 182 time several mobility bins are recorded and in turn for each mobility bin the full spectrum
 183 along the mass axis is acquired. In order to allow detection of smaller regions of interest, the
 184 algorithm works on short compact subsets (intervals) of the mass axis, henceforth referred to
 185 as *windows*. One such window, represented by a list of tuples $(m/z, i)$, is the single datum
 186 considered by the algorithm for similarity search.

187 Window generation

188 The set of windows is created by dividing all recorded mass axes into windows. In order to
 189 avoid missing an isotopic pattern by distributing it into two windows, a second set of windows
 190 that is offset by half a window length is created, such that the mass axes are covered in an
 191 overlapping fashion.

192 The length of the windows should be wide enough to capture a whole isotopic pattern, if
 193 the windows are applied in an overlapping fashion and small enough such that the chance of
 194 having several isotopic patterns in a window is small. For an assumed pattern length of up
 195 to 5Da a window length of 10Da is considered useful.

196 Binning

197 The calculation of a window's hash value requires the mass axis in a binned form with equally
 198 spaced bins. Finding an optimal binning scheme is by no means trivial: A very fine binning
 199 resolution makes the algorithm less robust and increases the computational cost, while a
 200 very coarse binning resolution yields an increased loss of information. A binning resolution of
 201 $0.1m/z$ was considered suitable as, on the one hand, it still resolves isotopic pattern with up
 202 to charge state five (corresponding to a spacing of $0.2m/z$) and, on the other hand, keeps
 203 the computational load feasible.

204 Locality-sensitive hashing

205 Used similarity measure

Two windows W_i and W_j shall be considered similar when their mass spectra have the same
 shape but not necessarily the same overall scale. Therefore, the similarity function $s(W_i, W_j)$
 used is the cosine similarity

$$s(W_i, W_j) = \frac{\langle \mathbf{I}_i, \mathbf{I}_j \rangle}{\|\mathbf{I}_i\| \|\mathbf{I}_j\|}, \quad (1)$$

where $\mathbf{I}$ denotes the intensity array of a binned window, $\langle \cdot, \cdot \rangle$ the standard scalar product and
 $\|\cdot\|$ the Euclidean norm. As a direct consequence from the linearity of the scalar product

and norm, the cosine similarity is scale invariant:

$$s(\alpha W_i, \beta W_j) = \frac{\langle \alpha \mathbf{I}_i, \beta \mathbf{I}_j \rangle}{\|\alpha \mathbf{I}_i\| \|\beta \mathbf{I}_j\|} = \frac{\langle \mathbf{I}_i, \mathbf{I}_j \rangle}{\|\mathbf{I}_i\| \|\mathbf{I}_j\|} = s(W_i, W_j), \quad (2)$$

for $\alpha, \beta > 0$. Thus, the findings of the algorithm are independent of the absolute intensity values.

Hash function and amplification

The appropriate family of hash functions for the cosine similarity is the random projection hashing [27]. The hash function is given by $h(\cdot) = \text{sign}(\langle \cdot, r \rangle)$ with the components of the vector r being random samples from a standard normal distribution.

To control for false positives or false negatives, an augmented LSH with n AND connectives and m OR connectives is used. This means that instead of computing a single hash function $m \times n$ hash functions are calculated and divided into m arrays of length n . Two objects x_i and x_j are now considered collided if all n hash functions of a block yield the same value. By choosing m and n appropriately, a sharp sigmoid shape of the collision probability $P_{m,n}([x_i, x_j])$ can be achieved, which effectively translates into a similarity threshold. Figure 5 shows $P_{m,n}([x_i, x_j])$ for different values of m and n .

Intensity thresholding

A very simple and general approach to signal classification is by means of thresholding with respect to a signal-to-noise ratio [28]. If one assumes a global noise estimate, this reduces to a thresholding with respect to the intensity of each peak.

Data generation

The synthetic data sets consist of windows of length of $10m/z$, with a binning resolution of $0.1m/z$.

In each set, two types of data were created: Firstly, windows containing no isotopic patterns and noise only. Secondly, windows containing both isotopic patterns and noise. The label of each peak ("signal", if it is part of an isotopic pattern, "noise_2", if it is a noise peak in a window that contains an isotopic pattern, and "noise_1", if it is a noise peak inside a window without an isotopic pattern being present) was stored for later evaluation. In total 18445 windows per data set were created, of which 14107 contained noise only and 4338 contained both signal and noise.

In the different data sets the intensities of the "true" signals were scaled differently, such that the maximum signal peak has an intensity of 1000, 500, 250, 125, 64, or 32, respectively.

Modeling noise

The noise signal is assumed to consist of independently sampled peaks that share the following properties: The number of peaks per window, k , is given as

$$k \sim \text{Pois}(\lambda_p) + 1,$$

where Pois denotes a Poisson distribution with mean λ_p . Our data was generated using $\lambda_p = 4$.

The location of each peak was sampled uniformly within the window and the intensity value i of each peak was sampled from

$$i \sim \text{Exp}(\lambda_e),$$

where Exp denotes a exponential distribution with mean λ_e^{-1} . Our data was generated using $\lambda_e = 15$.

Modeling isotopic patterns

For the “true” signal the average model [11] was used. The monoisotopic peak was placed in the middle of the window and the contribution of the next five peaks was considered. In order to model the repeated occurrences of patterns, two copies with all peaks scaled to half intensities were added. Finally, a noise signal, individually sampled according to description above, was added to every true pattern window.

The monoisotopic masses were ranged from 150u to 5000u in steps of 10u. Charge states from 1e up to 5e were included if the resulting m/z was in the interval $[150m/z, 2000m/z]$.

Classification by collision

In order to classify windows into “noise” or “signal”, for each window in the data set (several hash values are computed. Then the number of occurrences of each hash value is counted and all hash values that occurred more than one time are stored in a table, the so-called collision table. A window in question is called “signal” if at least one of its hash values can be found in the collision table, otherwise it is called “noise”.

A single peak in the data set is classified by whether there was a collided window that contains the peak. This is especially important as peaks can be part of several windows due to the overlapping window approach.

Evaluation

Although the synthetic data provides ground truth labels, a meaningful computation of true-positive and false-negative rates is not straightforward.

Since in our approach all peaks in a window will be assigned the same label, the following problem is caused: When signal and noise is present in a window, peaks with label “noise_2” will be considered “true” as well, which would result in a high false classification rate. For work on real world data however, the inability to remove noise peaks among isotope patterns is typically cured by the fact that subsequent processing steps like feature finding can take the multidimensional signal shape into account, which facilitates the removal of remaining noise peaks.

Thus, in order to learn about the false classification rate of actual interest, peaks with label “noise_2” were not considered for the computation of classification rates both for the intensity thresholding and our approach.

270 The presence of peaks with label “noise_2” is nevertheless important to test whether whole
271 patterns are missed due to noise in the same window.

272 Scalability study

273 For parallel computing, the algorithm was implemented in C++ with multithreading enabled
274 by usage of openMP [29]. The scalability study was performed on a single Ubuntu 20.04 LTS
275 machine with two Intel Xeon Gold 6238 CPUs, each featuring 22 physical cores @2.1 GHz.
276 Overall, enabled hyper-threading allows for the parallel execution of 88 threads. The machine
277 is further equipped with 192 GiB (12x 16 GiB) of DIMM DDR4 @2933 MHz main memory.
278 Our C++ package was compiled using GCC 9.3 and optimization level -O3.

279 As test data a single MS1 frame of a nanoLC-TIMS-MS/MS (DDA-PASEF [8]) analysis
280 of HeLa whole proteome digest was used and raw data access was enabled by OpenTIMS
281 [30]. HeLa cells were lysed in a urea-based lysis buffer (7 M urea, 2 M thiourea, 5 mM
282 dithiothreitol (DTT), 2% (w/v) CHAPS) assisted by sonication for 15 min at 4°C in high
283 potency using a Bioruptor instrument (Diagenode). Proteins were digested with Trypsin using
284 a filter-aided sample preparation (FASP) [31] as previously detailed [32]. 200 ng of peptide
285 digest were analyzed using a nanoElute UPLC coupled to a TimsTOF PRO MS (Bruker).
286 Peptides injected directly in an Aurora 25 cm x 75 µm ID, 1.6 µm C18 column (Ionopticks)
287 and separated using a 120 min. gradient method at 400 nL/min. Phase A consisted on
288 water with 0.1% formic acid and phase B on acetonitrile with 0.1% formic acid. Sample was
289 injected at 2% B, lineally increasing to 20% B at 90 min., 35% B at 105 min., 95% at 115
290 min. and hold at 95% until 120 min. before re-equilibrating the column at 2%B. The MS
291 was operated in DDA-PASEF mode [8], scanning from 100 to 1700 m/z at the MS dimension
292 and 0.60 to 1.60 1/k0 at the IMS dimension with a 100 ms TIMS ramp. Each 1.17 sec MS
293 cycle comprised one MS1 and 10 MS2 PASEF ramps (frames). The source was operated
294 at 1600 V, with dry gas at 3 L/min and 200°C, without nanoBooster gas. The instrument
295 was operated using Compass Hystar version 5.1 and timsControl version 1.1.15 (Bruker). All
296 reagents and solvents used were MS-grade.

297 Appendix

298 Acknowledgements

299 The authors would like to thank Mateusz Łacki for fruitful discussions and help in development
300 of ideas and Patrick Raaf for his input on Big Data technology.

301 Funding

302 This work was supported by Deutsche Forschungsgemeinschaft (DFG)[329350978] and Bun-
303 desministerium für Bildung und Forschung (BMBF)[031L0217A/B].

304 Availability of data and materials

305 Data and code is available at <https://github.com/hildebrandtlab/mzBucket>

306 Ethics approval and consent to participate

307 Not applicable.

308 Competing interests

309 The authors declare that they have no competing interests.

310 Consent for publication

311 Not applicable.

312 Authors' contributions

313 BS and AH conceptualized the work, KB and DT designed the work, DGZ acquired data, KB
314 and DT analyzed data, KB, DT and TK created new software used in this work, ST, BS and
315 AH revised the work. All authors read and approved the final manuscript.

316 References

- 317 [1] Andrea D. Weston and Leroy Hood. "Systems Biology, Proteomics, and the Future of
318 Health Care: Toward Predictive, Preventative, and Personalized Medicine". In: *Journal*
319 *of Proteome Research* 3.2 (Apr. 2004), pp. 179–196.
- 320 [2] N. Leigh Anderson and Norman G. Anderson. "Proteome and proteomics: New technolo-
321 gies, new concepts, and new words". In: *Electrophoresis* 19.11 (Aug. 1998), pp. 1853–
322 1861.
- 323 [3] Walter P Blackstock and Malcolm P Weir. "Proteomics: quantitative and physical
324 mapping of cellular proteins". In: *Trends in Biotechnology* 17.3 (Mar. 1999), pp. 121–
325 127.
- 326 [4] Lindsay K. Pino et al. "Emerging mass spectrometry-based proteomics methodologies
327 for novel biomedical applications". In: *Biochemical Society Transactions* 48 (5 Oct.
328 2020), pp. 1953–1966.
- 329 [5] Ruedi Aebersold and Matthias Mann. "Mass spectrometry-based proteomics". In: *Nature*
330 422.6928 (Mar. 2003), pp. 198–207.
- 331 [6] Erin Shammel Baker et al. "An LC-IMS-MS Platform Providing Increased Dynamic
332 Range for High-Throughput Proteomic Studies". In: *Journal of Proteome Research* 9.2
333 (Feb. 2010), pp. 997–1006.

- 334 [7] Ute Distler et al. "Drift time-specific collision energies enable deep-coverage data-
335 independent acquisition proteomics". In: *Nature Methods* 11.2 (Feb. 2014), pp. 167–
336 170.
- 337 [8] Florian Meier et al. "Online parallel accumulation–serial fragmentation (PASEF) with a
338 novel trapped ion mobility mass spectrometer". In: *Molecular and Cellular Proteomics*
339 17.12 (2018), pp. 2534–2545.
- 340 [9] Dirk Valkenburg et al. "The isotopic distribution conundrum". In: *Mass Spectrometry*
341 *Reviews* 31.1 (2012), pp. 96–109.
- 342 [10] Mateusz K. Łącki, Dirk Valkenburg, and Michał P. Startek. "IsoSpec2: Ultrafast Fine
343 Structure Calculator". In: *Analytical Chemistry* 92.14 (2020), pp. 9472–9475.
- 344 [11] Michael W. Senko, Steven C. Beu, and Fred W. McLafferty. "Determination of monoiso-
345 topic masses and ion populations for large biomolecules from resolved isotopic distri-
346 butions". In: *Journal of the American Society for Mass Spectrometry* 6.4 (Apr. 1995),
347 pp. 229–233.
- 348 [12] Piotr Indyk and Rajeev Motwani. "Approximate nearest neighbors". In: *Proceedings of*
349 *the thirtieth annual ACM symposium on Theory of computing - STOC '98*. New York,
350 New York, USA: ACM Press, 1998, pp. 604–613.
- 351 [13] Aristides Gionis, Piotr Indyk, and Rajeev Motwani. "Similarity Search in High Dimen-
352 sions via Hashing". In: *Proceedings of the 25th International Conference on Very Large*
353 *Data Bases* (1999), pp. 518–529.
- 354 [14] Zhihua Xia et al. "A Privacy-Preserving and Copy-Deterrence Content-Based Image
355 Retrieval Scheme in Cloud Computing". In: *IEEE Transactions on Information Forensics*
356 *and Security* 11.11 (Nov. 2016), pp. 2594–2608.
- 357 [15] Ying Zhang et al. "Video anomaly detection based on locality sensitive hashing filters".
358 In: *Pattern Recognition* 59 (Nov. 2016), pp. 302–311.
- 359 [16] Konstantin Berlin et al. "Assembling large genomes with single-molecule sequencing
360 and locality-sensitive hashing". In: *Nature Biotechnology* 33.6 (June 2015), pp. 623–
361 630.
- 362 [17] André Müller et al. "MetaCache: context-aware classification of metagenomic reads
363 using minhashing". In: *Bioinformatics* 33.23 (Dec. 2017). Ed. by Inanc Birol, pp. 3740–
364 3748.
- 365 [18] D. Dutta and T. Chen. "Speeding up tandem mass spectrometry database search:
366 metric embeddings and fast near neighbor search". In: *Bioinformatics* 23.5 (Mar. 2007),
367 pp. 612–618.
- 368 [19] Chuang Li et al. "MCtandem: an efficient tool for large-scale peptide identification on
369 many integrated core (MIC) architecture". In: *BMC Bioinformatics* 20.1 (Dec. 2019),
370 p. 397.
- 371 [20] Lei Wang, Sujun Li, and Haixu Tang. "msCRUSH: Fast Tandem Mass Spectral Cluster-
372 ing Using Locality Sensitive Hashing". In: *Journal of Proteome Research* (Dec. 2018),
373 acs.jproteome.8b00448.

- 374 [21] Lei Wang et al. "A Fast and Memory-Efficient Spectral Library Search Algorithm Using
375 Locality-Sensitive Hashing". In: *Proteomics* 20 (21-22 Nov. 2020).
- 376 [22] Martin Slawski et al. "Isotope pattern deconvolution for peptide mass spectrometry
377 by non-negative least squares/least absolute deviation template matching". In: *BMC*
378 *Bioinformatics* 13.1 (Dec. 2012), p. 291.
- 379 [23] Fatema Tuz Zohora et al. "DeepIso: A Deep Learning Model for Peptide Feature De-
380 tection from LC-MS map". In: *Scientific Reports* 9.1 (Dec. 2019), pp. 1–13.
- 381 [24] Jürgen Cox and Matthias Mann. "MaxQuant enables high peptide identification rates,
382 individualized p.p.b.-range mass accuracies and proteome-wide protein quantification".
383 In: *Nature Biotechnology* 26.12 (Dec. 2008), pp. 1367–1372.
- 384 [25] Nikita Prianichnikov et al. "Maxquant software for ion mobility enhanced shotgun pro-
385 teomics". In: *Molecular and Cellular Proteomics* 19 (6 Mar. 2020), pp. 1058–1069.
- 386 [26] Matei Zaharia et al. "Spark : Cluster Computing with Working Sets". In: *HotCloud'10*
387 *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing* (2010).
- 388 [27] Moses S. Charikar and Moses S. "Similarity estimation techniques from rounding al-
389 gorithms". In: *Proceedings of the thirty-fourth annual ACM symposium on Theory of*
390 *computing - STOC '02*. New York, New York, USA: ACM Press, 2002, p. 380.
- 391 [28] Chris Bauer, Rainer Cramer, and Johannes Schuchhardt. "Evaluation of Peak-Picking
392 Algorithms for Protein Mass Spectrometry". In: *Data Mining in Proteomics: From Stan-*
393 *dards to Applications*. Ed. by Michael Hamacher, Martin Eisenacher, and Christian
394 Stephan. Totowa, NJ: Humana Press, 2011, pp. 341–352.
- 395 [29] Leonardo Dagum and Ramesh Menon. "OpenMP: an industry standard API for shared-
396 memory programming". In: *Computational Science & Engineering, IEEE* 5.1 (1998),
397 pp. 46–55.
- 398 [30] Mateusz K. Łacki et al. "OpenTIMS, TimsPy, and TimsR: Open and Easy Access to
399 timsTOF Raw Data". In: *Journal of Proteome Research* 20.4 (2021), pp. 2122–2129.
- 400 [31] J R Wisniewski et al. "Universal sample preparation method for proteome analysis". In:
401 *Nat Methods* 6.5 (2009), pp. 359–362.
- 402 [32] Ute Distler et al. "Label-free quantification in ion mobility-enhanced data-independent
403 acquisition proteomics". In: *Nature Protocols* 11.4 (2016), pp. 795–812.
- 404 [33] Charles E Cook et al. "The European Bioinformatics Institute in 2018: tools, infras-
405 tructure and training". In: *Nucleic Acids Research* 47.D1 (Jan. 2019), pp. D15–D22.

406 Figures

Figure 1: Experimental workflow in an LC-IMS-MS setup and resulting signal of interest. **a)** Chemical separation by high-performance liquid chromatography (HPLC) and resulting intensity distribution along retention time. **b)** Subsequent ion-mobility separation (IMS) and resulting intensity distribution along the mobility dimension. **c)** Time-of-flight mass spectrometry (TOF-MS) and resulting isotopic pattern, i.e., evenly spaced peaks with a certain distribution of the peaks envelope. The column on the right shows the signal as a function of two variables, drift time and retention time on the top and mass-over-charge ratio and drift time on the bottom. Note the repeated occurrence of these isotopic patterns in the 3D plot on the bottom right.

Figure 2: Size of recorded data at the European Bioinformatics Institute (EMBL-EBI) over time for different platforms in life sciences. Mass spectrometry and other data has shown an exponential growth. Plot recreated after [33].

Figure 3: Schematic overview of the approach: Short intervals (windows) of the several mass axes are considered as smallest building blocks of the data set and generated from the raw data. Then for each window the (several) hash functions h map into the hash buckets. Finally, if more than one window is mapped in a given hash box, all windows inside the box are considered “true” signal.

Figure 4: Receiver operating characteristic. Single points mark the results of our approach and the tuples denote the number of AND and OR amplifications used. The solid line shows the performance of the intensity-threshold approach and the dashed line the results of random guessing.

Figure 5: Controlling typical similarity of pairs: Probability $P_{m,n}([x_i, x_j])$ to retrieve a pair for several combined hashes as a function of similarity $s(x_i, x_j)$. By appropriate choice of m and n , found pairs have a high probability of having at least a certain similarity. Note that the x-axis starts at $s(x_i, x_j) = 0.55$, for values smaller than that all curves are almost zero.

Figure 6: Scalability for different m and n : Wall-clock time of the implementation in seconds as a function of the number of threads used. Note the logarithmic scales on both axes. The approximately linear trend shows that the implementation scales out well. See text for the setup used.