Published in final edited form as:

Proteomics. 2020 November ; 20(21-22): e2000002. doi:10.1002/pmic.202000002.

A fast and memory-efficient spectral library search algorithm using locality-sensitive hashing

Lei Wang, Kaiyuan Liu, Sujun Li, Haixu Tang

Indiana University, Bloomington, IN 47405, USA

Abstract

With the accumulation of MS/MS spectra collected in spectral libraries, the spectral library searching approach emerges as an important approach for peptide identification in proteomics, complementary to the commonly used protein database searching approach, in particular for the proteomic analyses of well-studied model organisms, such as human. Existing spectral library searching algorithms compare a query MS/MS spectrum with each spectrum in the library with matched precursor mass and charge state, which may become computationally intensive with the rapidly growing library size. Here, we present the software msSLASH, which implements a fast spectral library searching algorithm based on the Locality-Sensitive Hashing (LSH) technique. The algorithm first converts the library and query spectra into bit-strings using LSH functions, and then computes the similarity between the spectra with highly similar bit-string. Using the spectral library searching of large real-world MS/MS spectra datasets, we demonstrated our algorithm significantly reduced the number of spectral comparisons, and as a result, achieved 2–9X speedup in comparison with existing spectral library searching algorithm SpectraST. The spectral searching algorithm is implemented in C/C++, and is ready to be used in proteomic data analyses.

1 Introduction

The mass spectrometry (MS) technology, in particular, the liquid chromatography coupled tandem mass spectrometry (LC-MS/MS), has evolved rapidly in the past decade, with drastically improved throughput and sensitivity [25, 43, 2]. The state-of-the-art LC-MS/MS instruments can acquire up to 10^6 tandem mass (MS/MS) spectra from a complex proteomic sample within a few hours in a single run. With the help of additional biochemical techniques that reduce the sample complexity, LC-MS/MS analyses can achieve even higher sensitivity on complex samples [22, 33]. Consequently, LC-MS/MS is adopted to address a broad range of biomedical questions, e.g., for the discovery of protein biomarkers for diagnosis and proper treatments for complex diseases [38], including cardiovascular diseases [3, 1], diabetes [35, 36, 9] and cancer [23]. These proteomics studies often involved hundreds to thousands of clinical samples, generating massive MS/MS datasets, similar to the data from other sequencing-based ‘omics’ fields such as genomics and transcriptomics. Conventionally, the identification of the MS/MS spectra are achieved through database searching [42], in which the experimental MS/MS spectra are searched against the peptides with expected precursor mass that are generated from the proteins in a target database (e.g.,

the human proteins) through *in silico* digestion. The evaluation of peptide-spectrum matches (PSMs) is performed using an empirical scoring function, and the peptide receiving the highest score is reported as the identified peptide of each experimental mass spectrum. Several popular database searching engines such as Sequest [19], Mascot [34], and MSGF+ [26], have been extensively used in this field, while open search algorithms were developed more recently for improving the identification of MS/MS spectra containing unknown mutations or post-translational modifications (PTMs) [28, 11]. Despite the success of data searching algorithms in proteomics, the empirical scoring schemes implemented in these algorithms may give some false PSMs high scores that are indistinguishable from true PSMs. To control the false discovery rates (FDRs) of peptide identification, a target-decoy search approach is often adopted to estimate a score threshold for a desirable FDR of PSMs or peptides (e.g., 0.01) [17]. The PSMs below the threshold are discarded even though a fraction of them are true. With the accumulation of MS/MS data, several spectral libraries have been constructed [13, 20] (e.g. the NIST spectrum library [53, 54]), which collect reliable experimental MS/MS spectra (i.e., manually annotated spectra or the spectra from synthetic peptides) of peptides with various charge states. The latest spectral libraries consists of thousands to millions of peptide spectra, in particular achieving a high coverage of tryptic peptides in human proteins. To exploit the utility of such large spectral library in proteomics, the spectral library searching approach emerges as an alternative approach to peptide identification. Instead of searching against peptides in a target protein database, spectral library searching methods query the experimental spectra against a large collection of mass spectra in a spectral library, and the annotated peptides of the library spectra with sufficiently high similarity are reported as the identified peptides. The success of spectral library search approach depends on the coverage of the target spectral library, because a query spectrum cannot be identified if a spectrum from the same peptide is not collected in the library. On the other hand, the similarity measure used here for comparing two MS/MS spectra is straightforward (e.g., the cosine distance [16] or Pearson correlation coefficient [6]), and high confidence can be reached for the identification when the similarity is high, because the MS/MS data are known to be highly reproducible at the same experimental setting [30, 8]. In fact, spectral library searching algorithm was shown superior sensitivity on peptide identification than the database search algorithms due to the employment of real library spectra for matching [29, 55], when the spectral library has a high coverage, because these algorithms considers the peak intensities in their scoring function, as opposed to the conventional database searching algorithms. Therefore, spectral library search represents a complementary approach to peptide identification in proteomics, and is expected to exhibit broader application with the increasing accumulations of reliable MS/MS spectra.

Existing mass spectral searching algorithms MS/MS spectral libraries experimentally curated [51, 57] or predicted [44, 31], such as SpectraST [29] and M-SPLIT [47], all adopted a same strategy to compare the query spectrum against each candidate spectrum in the spectral library with matched precursor mass and charge state. ANN-SoLo[7], a tool optimized for open modification spectral library searching first utilizes the approximate nearest neighbor (ANN) index to identify a small subset of library spectra similar with each query spectrum and then computes the explicit similarity between them. ANN-SoLo runs faster than SpectraST in open modification search; however, it runs 4.8 times slower in the

standard library search while identifying fewer PSMs in comparison with SpectraST. In this paper, we present a fast and memory-efficient spectral library searching algorithm using the Locality-Sensitive Hashing (LSH) technique. It first preprocesses the spectral library to generate an index for all spectra using LSH consisting of L SimHash functions [4]. As a result, each library spectrum is mapped to a bucket, represented by an L -bit long hash string (each bit corresponding to the outcome of the random projection using one SimHash function). The spectrum search is then conducted in two steps. In the first step, for a query spectrum, we map it into a bucket, represented by a L -bit string (i.e., the query string) using the same compound hash function, and attempt to look up all bit strings (buckets) that are within a Hamming distance from the query string. Because of the property of LSH, the smaller the cosine distance between two spectra, the more probable the Hamming distance between their bit strings is smaller than the threshold. In the second step, the cosine distance is computed between each library spectra with a matched LSH string with the query spectrum, and the library spectrum with the highest similarity is reported. We implemented the LSH-based spectral library searching algorithm in a software tool called *msSLASH* in C++. To enable the efficient memory usage for loading millions of spectra, We used the run-length encoding (RLE) [37] to compress both the query and library spectra and compute their similarity accordingly. We compared the performance of *msSLASH* with SpectraST and dot-product based searching algorithm on searching three proteomics datasets against two target spectral libraries, respectively. The results showed *msSLASH* achieved up to 8.9 times speedup over SpectraST (M-SPLIT is significantly slow to complete so we only compare with SpectraST), which identified 5% more peptides than SpectraST overall.

2 Methods

The spectral library searching algorithm implemented in *msSLASH* is depicted in Figure 1. Instead of comparing a query spectrum with each other spectrum in a spectral library with matched precursor mass and charge state, which is time consuming and resource wasting, only a subset of potentially similar library spectra are retrieved for explicit similarity computation in *msSLASH*. Similar spectra are more likely to be mapped through LSH algorithms to same buckets than dissimilar spectra. The more similar two MS/MS spectra, the more likely they are fragmented from the same peptide. The target-decoy search strategy [17] has been applied to precisely estimate incorrect peptide identifications at a low False Discovery Rate (FDR) level, i.e. 1%.

2.1 Spectrum Similarity

In this paper, the spectrum searching problem is considered as the problem of searching high dimensional vectors (referred to as the *spectrum vectors*), where each input spectrum is represented as a *sparse* vector, usually consisting of several hundreds of peaks. Each dimension of a spectrum corresponds to a bin of mass-to-charge (m/z) ratio, and the *normalized* peak intensity is assigned to each bin accordingly. Using high resolution mass spectrometers (e.g., the Orbitrap MS), the fragment ions in MS/MS spectra can be measured in high mass accuracy, e.g. 0.02 Da, and wide mass range, e.g. from 0 Da to 2000 Da. Various methods can be used to measure the similarity of two MS/MS spectra, such as

cosine similarity [16], statistics-based similarity [21]. Among these measures, we adopt the cosine similarity, which is a variant of angle-based cosine distance.

2.2 Locality Sensitive Hashing

Spectral library searching essentially can be formulated as a similarity searching problem in proteomics. There have been various methods proposed to address the similarity searching issue in different fields, such as k-d tree for nearest neighbor search and range search in multidimensional space [56]; KNN is a simple yet powerful algorithm that has extensive application in classification and regression [52]. Locality Sensitive Hashing (LSH) algorithms have also been adopted in proteomics research, e.g., to speed up the database searching [16], and to cluster large-scale mass spectra [49], etc.

Hashing functions in general can be expressed in the form $y = h(x)$, where y is the hash value (or hash code) of the vector x , to map vectors into entries (or buckets) in a hash table. Conventional hashing algorithms separate apart two vectors unless they are exactly the same, decreasing the collision possibility. Locality sensitive hashing algorithms, in contrast, map similar vectors into the same *buckets* with higher probability than dissimilar vectors: the more similar two vectors, the higher probability of collision. Spectra of the same peptide usually share high similarity than the spectra of different peptides, and therefore they are more likely to be hashed into the same buckets.

LSH family[4] $\mathcal{H}$ is a family of hash functions which can place similar vectors (in our case, the spectrum vectors) to the same buckets with greater probability than dissimilar vectors. Formally, The family $\mathcal{H}$ is called $(R, cR, P1, P2)$ -sensitive, with the constraint of $c > 1$; $P1 > P2$, if for any two objects p, q ,

- if $d(p, q) \leq R$, then $Pr[h(p) = h(q)] \geq P1$,
- if $d(p, q) \geq cR$, then $Pr[h(p) = h(q)] \leq P2$,

where $d(p, q)$ is the distance between objects p and q (in our case, the spectrum vector), and $h(p) = h(q)$ means p and q gets assigned to the same bucket.

2.2.1 Random projection

Different families of LSH algorithms have been proposed for different distance measure, such as the LSH with p-stable distributions for the L_p distance [15], MinHash for the set intersection estimation [12], and the Leech lattice LSH for the Euclidean distance [5]. For the angle-based distance, i.e. the cosine similarity in our case, several LSH functions were available, including the random projection, the super-bit LSH and the kernel LSH[48]. Here, we adopt the random projection for approximating cosine distance in spectra similarity searching, which was successfully applied to mass spectra clustering previously by us [49].

Random projection assigns two high dimensional vectors with a small angle to same buckets (i.e., *colliding*) with high probability, where the *colliding probability* is linearly proportional to their cosine similarity, as depicted in Figure 2. The angle between two vectors s_i and s_j is

$$\theta(s_i, s_j) = \arccos \frac{s_i^T s_j}{\|s_i\| \|s_j\|}, \text{ and the random projection hash function is defined by } h(x) =$$

$\text{sign}(w^T x)$, where w is a vector chosen randomly from the multi-dimensional Gaussian distribution and x is an input vector. If the input vector lies on the positive side of the hyperplane, then $h(x) = 1$; otherwise, $h(x) = -1$. The collision probability for any s_i and s_j is therefore $Pr[h(s_i) = h(s_j)] = 1 - \frac{\theta(s_i, s_j)}{\pi}$. Overall, the collision probability of two input vectors is high if θ is small, i.e. the two vectors are sufficiently similar.

2.2.2 Augmented LSH—In practice, many LSH hash functions from the same LSH family are often concatenated to increase (amplify) the gap of collision probability between similar objects and the objects that are dissimilar. A compound hash function g is formed by concatenating multiple (single) hash functions in the form of $g(x) = (h_1(x), \dots, h_K(x))$, where each hashing function $h_k(x)$ is chosen randomly from the same family $\mathcal{H}$ [48]. Two input vectors collide if and only if their compound hash codes are exactly the same. The collision probability of two vectors under a single hash function $h(x)$ is proportional to their pairwise cosine similarity p , and therefore, the probability of two vectors to collide under the compound hash function with K single hash functions is p^K . The gap of collision probability between similar spectra and dissimilar spectra is enlarged subsequently, for instance, the collision probability of two similar spectra with cosine similarity 0.7 is $0.7^{10} \approx 0.3$, whereas the probability is $0.2^{10} \approx 10^{-8}$ for two dissimilar spectra with cosine similarity 0.2.

The combination of several single hash functions will increase the gap of collision probability between similar spectra and those dissimilar ones; however, at the same time, it will decrease the collision probability for similar spectra, as depicted in Figure 2. To address this issue, multiple hash tables each containing a compound hash function shall be adopted. Two vectors are considered as colliding as long as they share the same compound hash code in any hash table. Therefore, the collision probability of two spectra with probability p under L hash tables containing K single hash functions each, will be $1 - (1 - p^K)^L$. For example, for $K = 10$ and $L = 100$, the collision probability for two similar spectra with probability 0.7 is ≈ 0.94 , whereas the collision probability is $\approx 10^{-5}$ for two dissimilar spectra with cosine similarity 0.2, as shown in Figure 2. In short, the gap of collision probability between similar spectra and dissimilar spectra can be enlarged with the concatenation of hash functions in a single hash table; while the probability of two similar spectra can retain sufficiently high to collide (≈ 0.94) compared to dissimilar spectra ($\approx 10^{-5}$) when multiple hash tables are considered.

2.3 Run-length encoding (RLE) of MS/MS spectra

To exploit the sparsity of the spectrum vectors, we adopt the run-length encoding (RLE) [39] to compress them so that a large number of spectrum vectors can be simultaneously loaded into the memory. In RLE, a sequence (run) of consecutive data elements in the same value are represented by a single data value and the sequence length. As the spectrum vector is sparse and contains runs of zeros, we used a sequence of two values (bytes) to store the vector, the first representing the length of consecutive zeros between the peaks, and the second representing the peak intensities. When the length of consecutive zeros is above 255, we insert a pseudo-peak with intensity of zero in between. For examples, the spectrum vector of (0, 0, 0, 0, 124, 0, 0, 0, 0, 212, 0, 0) will be encoded as (4, 124, 5, 212). Hence,

we need $\approx 2n$ bytes to store a spectrum of n peaks. As a typical MS/MS spectrum contains ≈ 100 peaks, we can reduce the memory usage from 2KB to 0.2KB (i.e., $\approx 10\times$ reduction). Note that, we can compute the cosine similarity between two RLE vectors faster than using spectrum vectors directly: we can increment two counts of runs along with the RLE vectors, respectively, and when the two counts are equal, the product of corresponding peak values is added to the similarity. The LSH can be computed similarly on the RLE.

2.4 Spectra Library

Three spectral libraries of human peptides were used in this study for the evaluation purpose. The first library is curated by NIST¹ and contains 340,356 unique collision-induced dissociation (CID) spectra. They were assembled from multiple data sources, and the spectra of the same changes and identified as the same peptides with the same (or without any) post-translational modification (PTM) were merged into a single representative spectrum by SpectraST. Note that a decoy spectral library of roughly the same size was constructed by applying the precursor swap [10] method described in the following section to each spectrum. The distribution of spectra of various charge states are shown in Supplementary Figure S7.

The second library is curated by the MassIVE repository [51] from more than 30 TB human MS/MS data, including ProteomeTools human-derived synthetic peptides [57], and contains 2,154,269 higher energy collisional dissociation (HCD) spectra. We removed 12; 286 spectra in the MassIVE HCD spectral library, whose precursor mass does not match our manually calculated mass based on the peptide sequence, the documented modifications, and the respective charge state. A decoy spectral library of roughly the same size was constructed by applying the precursor swap method on the spectrum level. The distribution of spectra of various charge states are shown in Supplementary Figure S7.

The third library is derived from the predicted MS/MS spectra made by MS2PIP [44] for the tryptic peptides of the entire human proteome. For MS2PIP prediction, we specified the minimum peptide sequence length of 7 and the maximum peptide mass of 5,000 Da, with up to 2 missed cleavages, the fixed Carbamidomethyl C and the variable Oxidation M. The MS2PIP library contains a total of 6,090,280 higher energy collisional dissociation (HCD) spectra, with the same number of doubly charged and triply charged spectra. We removed 1,051,720 duplicated peptide ions, and thus retained 5,038,560 spectra for the spectral library search. The same number of decoy spectra was generated using the precursor swap method on the spectrum level, which was used for the FDR estimation (see below).

2.5 FDR Estimation using the Target-decoy Search Approach

A spectral library (i.e., the target library) is meticulously generated from a large corpus of previously observed and identified with high confidence, or from synthetic peptides. The spectra within a spectral library are usually unique, i.e., each peptide has at most one spectrum for each charge state. The target-decoy search approach [17] is often adopted to

¹<https://chemdata.nist.gov/dokuwiki/doku.php?id=peptidew:cdownload>

estimate the fraction of false identifications at a low false discovery rate (FDR) level, e.g., 1%.

The same strategy as SpectraST [29] was to construct the decoy spectral libraries using the precursor swap and peak shift [10] methods at the spectrum level. Briefly, given a swap distance d , two spectra of the same charge but with the difference of precursor mass-to-charge-ratios greater than d are randomly selected in the input target library, and subsequently, their precursor mass-to-charge-ratios are swapped (i.e., *precursor swap*), and each peak in one of the two spectra is shifted by $+$, whereas each peak in the other spectrum is shifted by $-$ (i.e., *peak shifting*). The resulting decoy spectral library usually contains the same size as the target library, as two decoy spectra are generated for every two spectra in the target library. The spectra search is finally conducted against the concatenated spectral library including both the target and the decoy spectra, as suggested by SpectraST.

Instead of performing the one-against-all approach, msSLASH first adopted LSH to select a small subset of spectra in the library that are potentially similar to a query spectrum, and then only conducted the similarity computation between the query and the subset of spectra in the library. This approach is in general analog to the commonly used “seed-and-extend” approach to sequence similarity search, e.g., used in BLAST [24], where sequence-based hash functions (e.g., based on k-mers) are utilized to filter the sequences in a target database before rigorous sequence comparison algorithms are performed.

In msSLASH, each query spectrum is searched against a small subset of possibly similar spectra from the concatenated spectral library indexed by the same K-bit-long hash string, and the spectrum receiving the highest cosine similarity against the query spectrum is retained for the calculation of FDR. With the equal size of target and decoy libraries, it is expected that a same ratio of target and decoy spectra among the false identifications [18]. As demonstrated in Figure 3, the 1:1 target-decoy ratio is still preserved in each LSH bucket, and therefore the assumption for target-decoy method remains solid and the FDR estimation is accurate. The same results were observed regardless of the number of hash functions, the number of hash tables, or the target spectral library. FDR is thus calculated as the ratio between the number of identifications from the decoy library (i.e., the decoy hits) and the number of identifications from the target library (i.e., the target hits). The spectrum identifications are sorted in a non-ascending order by their cosine similarities according to the respective query spectra, and those above a given FDR threshold (e.g., 1%) are finally reported as searching results.

2.6 Spectrum Preprocessing

The same spectrum preprocessing step have been applied to each input MS/MS spectrum, following the common practices [49, 16] to remove noise peaks and enhance the signal to noise ratio thereafter. Specifically, we split the peaks into 100-Da wide bins, and retained the five strongest peaks within each bin. We consider all peaks with m/z between 0 and 2000. To alleviate the influence of dominant peaks when computing cosine similarities, each peak's intensity is logarithmized. Finally, the intensities of all peaks in each MS/MS spectrum are normalized such that the strongest peak in each spectrum receives the same intensity (1000 in our case).

2.7 MS/MS Datasets

In this paper, we used three benchmark datasets of MS/MS spectra to evaluate the performance of msSLASH in comparison with SpectraST (release 5.2.0), a commonly used spectral library search tool. The dataset A (ProteomeXchange[46] ID:PXD001197) contains 1,161,304 CID spectra in total with charges from 1+ to 4+, acquired using the high resolution LTQ Orbitrap Elite from the human cell line HEK293 [40]. The dataset B (ProteomeXchange ID: PXD000561) contains 23,644,033 CID spectra in total with charges from 1+ to 4+, acquired in a comprehensive human proteomic study by using the high resolution Fourier-transform mass spectrometry from 30 histologically normal human samples, including 17 adult tissues, 7 fetal tissues and 6 purified primary haematopoietic cells[25]. The Dataset C (ProteomeXchange ID: PXD004452) contains 22,275,341 HCD spectra in total, in which 13,466,922 spectra have no assigned charges, 4,914,471 are doubly charged, and 2,649,176 are triply charged. All raw MS data were converted into MGF files using the Proteomics Tools Suite². Detailed of spectra with each charge states are summarized in Table 1.

2.8 Database searching

In order to provide a reference for evaluating the spectral library searching results, we used the MSGF+[26] search engine for peptide identification. The parameters for database searching is set as the following to match the experimental conditions[40, 25]: 1) Instrument type: Orbitrap/FTICR; 2) Precursor mass tolerance: 50ppm; 3) Isotope error range: -1, 2; 4) Modification: oxidation as variable and carbamidomethylation as fixed; 5) Maximum charge: 7; 6) Minimum charge: 1; 7) Number of tolerable termini: 2. The false discovery rate (FDR) is estimated by using a target-decoy search (TDA) approach, in which the decoy proteins are generated by reversing the protein sequences in the target Uniprot human database (UP000005640 9606.fasta file containing 19,962 Swiss-Prot and 1,044 TrEMBL sequences was downloaded on April 25, 2017).

2.9 Software availability

We implemented our algorithm including the pre-processing step and the spectral library searching in C++, in the software package msSLASH, which is portable across different platforms, e.g., Windows, Linux, Unix and Macintosh. The functionality of multi-threading was implemented in msSLASH using OpenMP[14] to further speed up the spectral searching. msSLASH can be accessed as open-source software at github: <https://github.com/COL-IU/msSLASH>.

3 Results

3.1 Parameter Selection

Different combinations of parameters such as number of hash tables (iterations in our case) and number of single hash functions in one hash table, play vital role in searching spectra with satisfying specificity and sensitivity. More single hash functions in a composite

²<https://github.com/shengqh/RCPA.Tools>

function ensures sufficient gap in collision probability of spectra from same peptide and spectra from different peptide, however, it will also decrease the probability of similar spectra significantly. More iterations of searching can increase the number of identified PSMs at the cost of taking substantial time. Finding an equilibrium between number of hash functions and number of iterations thus is important. We evaluated the performance of msSLASH with iterations from 25 to 150 with step of 25 and hash functions from 6 to 13. As shown in Supplementary Figure S1 to S6, it appears that different parameter combinations should be selected for different datasets. Considering the existence of PSMs with low similarity to the annotated peptide, a small number of hash functions would usually suffice. We analyzed the frequency distribution of the cluster sizes as the number of concatenations and LSH functions increases on the MassIVE HCD spectral library. As illustrated in Figure 4, we selected 8 and 12 hash functions, 25 and 150 concatenations, respectively. We noticed that the distribution roughly follows a Gaussian distribution regardless of number of hash functions and concatenations selected. More hash functions will lead to fewer spectra in each bucket, as shown in each column, while more concatenations (i.e. iterations) will lead to proportionally more spectra in each bucket, as shown in each row. Note that, 1:1 target-decoy ratio is still preserved in each bucket after LSH, as shown in Figure 3. We observed similar a trend in the results of searching against NIST CID spectral library (data not shown). With the overall consideration of speed and specificity, we selected 75 hash tables (i.e. iterations of searching) each with 7 hash functions when running msSLASH on dataset A and B, against NIST CID spectral library; 10 hash functions and 100 iterations when searching dataset C against MassIVE HCD spectral library. Optimal choice of number of hash functions and iterations depend on datasets.

3.2 Evaluation

We evaluated the performance of msSLASH on three publicly available datasets in comparison with a naive spectral library searching algorithm, and SpectraST, the state-of-the-art spectral library searching algorithm that has been widely used in shotgun proteomics. The naive searching method was implemented to use the same scoring function as msSLASH, but directly computes the cosine similarities without using the LSH.

For comparison purpose, we considered those identified spectra that were also identified by MSGF+ when searching against the Uniprot human proteome at 1% FDR on unique peptide level for each of the three benchmarking datasets. As shown in Figure 5, the naive spectral library searching method identified slightly ($\approx 2\text{--}3\%$) more spectra than msSLASH because it does not employ the LSH approximation which will introduce false negatives to speed up the spectrum searching; on the other hand, it runs about 1.13–4.37 slower than msSLASH in practice, as shown in Supplementary Table S1 to S6. The larger spectral library size, the slower the naive spectral library searching algorithm. Importantly, Table 1 shows that because the curated experimental spectral libraries do not contain the spectra of all human peptides, the numbers of peptide identification from spectral library searching are smaller than those by MSGF+. For instance, when searching the dataset A-2+ against the NIST CID spectral library, the naive spectral library searching method (that attempts to match all spectra with similarity greater than the threshold) covers about 91% (284,530 vs 312,714) of

the MSGF identifications; in contrast, the coverage is only about 53.7% when searching the B-3+ dataset against the NIST CID spectral library.

In addition to searching against the curated experimental datasets, we also tested on the HCD spectra of human peptides predicted by MS2PIP. Note that for MSGF settings, we specified the fixed Carbamidomethyl C and the variable Oxidation M as well, without the requirements on the peptide length and peptide mass. The searching results are summarized in Table 2. We observed that the msSLASH spectral library searching method can identify much more PSMs at the same 1% FDR threshold; for instance, on dataset C-2+, it identified 4,888,045 PSMs compared to 2,350,961 from MSGF. Spectral library searching method identified 86.3% PSMs on dataset C-2+ (83.0% on PSMs on dataset C-3+) that were identified by MSGF with respective peptides also exist in the spectral library. Similar results were obtained by the naive spectral searching method (See Supplementary Table S1). Note that the MS2PIP library contains the predicted spectra with the minimum peptide length 7 and the maximum pepmass of 5000, and may cover fewer human tryptic peptides than the target database considered by MSGF+.

The comparison of msSLASH and SpectraST searching algorithms are summarized in Table 1. Overall, msSLASH runs 1.6 to 8.9 times faster than SpectraST, and the acceleration rate is higher for spectral libraries of larger size, as shown in Figure 6 and Table 1. Specifically, msSLASH runs 1.6 to 3.9 times faster when tested searching dataset A and B against the NIST CID library, which consists of 679,912 spectra, including both the target and decoy spectra that were constructed using the precursor swap and peak shift method. On a larger spectral library, i.e. the MassIVE HCD library that contains 4,215,721 target and decoy spectra in total, msSLASH completed in 1,419 seconds (≈ 24 minutes) over 100 iterations, 6.7 times faster than SpectraST, which finished in 9,517 seconds (≈ 2 hours and 38 minutes), on searching dataset C-3+ of 2.6 million HCD spectra; whereas msSLASH completed in 184 seconds over 75 iterations, 8.9 times faster than SpectraST, which finished in 1638 seconds, on searching dataset A-3+ of 461 thousands CID spectra. Note that the acceleration rate is correlated with the size of spectral library, not the size of the input query spectra dataset, as depicted in Figure 5. This is because the spectra from the spectral library are assigned to different buckets through augmented locality sensitive hashing, and a hit is then reported as the match with the highest cosine similarity in the *subgroup* of library spectra that collide with the query spectrum. Dataset B-2+ contains 14.4 million doubly charged spectra, on which msSLASH runs 1.6 times faster than SpectraST against the NIST CID spectral library, in comparison with the 6.7 times speedup when searching the dataset C-3+ that contains 2.6 million triply charged spectra against the MassIVE HCD spectral library. Here, we specified the SpectraST search settings as closely as possible to the msSLASH settings to ensure a fair comparison. Specifically, we set precursor m/z tolerance to 0.05 Da and ion bin width of 1 Th. We specified SpectraST to employ the rank-based similarity scoring function when computing similarity between two spectra.

For the datasets A and B, msSLASH outperforms SpectraST on the numbers of matched spectra that were also identified by MSGF+. The identified PSMs within 1% FDR by the target-decoy method is summarized in Figure 5. Significantly more matched spectra was identified for B-3+ dataset, in which msSLASH matched 1,210,049 spectra, 30% more than

those (923,106) identified by SpectraST, both at the same 1% FDR level. Similar results were obtained in the B-2+ dataset. It is worth noting that on dataset C-2+ (similarly on C-3+), msSLASH matched fewer spectra than SpectraST, covering 96.8% of the identifications from naive library searching method, and 97.3% of the identifications from SpectraST. The missing PSMs by msSLASH may be due to the LSH approximation (i.e., the spectrum of the true peptide is not in the same bucket as the query spectrum) and the different scoring functions used by msSLASH and SpectraST.

Figure 7 shows that on each of the dataset A-2+, A-3+, C-2+, and C-3+, the matched spectra by the naive searching method significantly overlap with those identified by SpectraST and msSLASH, which demonstrates that the identified PSMs by msSLASH are trustworthy. On dataset B-2+ and B-3+, msSLASH shares more matched spectra with the naive method than SpectraST.

We further investigated the difference among the identified peptides by these three methods, in which the peptide of the top matched spectrum in the target library was assigned to each input spectrum. We considered only the matched spectra that were also identified by MSGF+. As shown in Figure 8, on the dataset A and B, msSLASH identified more peptides than SpectraST when searching against the NIST CID spectral library. For instance, msSLASH identified 24% more (53,126 vs. 42,958) peptides than SpectraST on dataset B-3+ when searching against NIST CID spectral library. On all datasets, the peptides matched by the naive method significantly overlaps with the peptides matched by SpectraST and msSLASH. Note that on the dataset C, msSLASH is still competitive though SpectraST identified around 1% more peptides for +2 and +3 charged spectra. The small fraction of missing hits may be due to the fact that msSLASH employed the LSH techniques to accelerate the search that may introduce false negatives. These results again showed that msSLASH can significantly speed up spectral library searching without loss of sensitivity.

4 Discussion

In this paper, we demonstrated locality-sensitive hashing (LSH) can significantly improve the efficiency for spectral library searching while retaining the competitive sensitivity and implemented the algorithm in the open source msSLASH package. As shown in Table 1, msSLASH runs 1.6–8.9 times faster than SpectraST, while retaining 97% matched spectra by the naive spectral library searching algorithm. In Figure 7, we demonstrated that the matched spectra by msSLASH is largely overlapped with the searching results by naive searching algorithm and SpectraST, respectively. In summary, msSLASH reduces the running time for spectral library searching, while preserving the sensitivity of spectral library search.

To significantly reduce the unnecessary similarity computations between dissimilar tandem mass spectra, we made use of the LSH technique, which assigns similar spectra more likely into the same hash bucket. To alleviate the approximation introduced by LSH, which may miss some similar spectra, an iterative spectral searching strategy is adopted: with more iterations, similar spectra are more likely to be grouped together while keeping spectra from different peptides away. As shown in Supplementary Table S1 to S6, fewer iterations of

searching can lead to further speedup but may reduce the sensitivity of peptide identification. As shown in Figure 5, the acceleration rate increases with the size of spectral libraries: msSLASH runs 1.62 to 2.51 times faster when searching dataset B than SpectraST against the NIST CID spectral library containing 679,912 (both target and decoy) spectra. Specifically, the peptide identifications by msSLASH covers 94.6% matched spectra by the naive method on dataset B-2+, in comparison with SpectraST covering only 84.4%. msSLASH runs 6 to 9 times faster than SpectraST when searching the dataset C against the Massive HCD spectral library containing 4,215,721 spectra.

We expect the algorithm presented here can enhance the usage of spectral library searching, because the reference MS/MS spectra from millions of synthetic human peptides [32, 58] are fastly accumulated. Although the algorithm is evaluated here for searching peptide spectral library, it can be directly applied to the spectral library searching for other molecules, such as metabolites and natural products [41, 50], lipids [27, 45] and glycoconjugates. Furthermore, our approach can be generalized to searching for experimental MS/MS spectra from a molecule of specific interest (e.g., as a potential disease marker) in massive omic datasets that are made available by mass spectra data repositories.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgement.

This research was partially supported by the National Institute of Health grants (1R01GM130091, 1U01CA225753 and 1R01AI108888) and Indiana University (IU) Precision Health Initiative (PHI).

References

- [1]. Addona TA, Shi X, Keshishian H, Mani D, Burgess M, Gillette MA, Clauser KR, Shen D, Lewis GD, Farrell LA, et al., *Nature biotechnology* 2011, 29, 635.
- [2]. Aebersold R, Mann M, *Nature* 2016, 537, 347. [PubMed: 27629641]
- [3]. Anderson L, *The Journal of physiology* 2005, 563, 23. [PubMed: 15611012]
- [4]. Andoni A, Indyk P, *Foundations of Computer Science*, 2006. FOCS'06. 47th Annual IEEE Symposium on, IEEE 2006 459–468.
- [5]. Andoni A, Indyk P, *Foundations of Computer Science*, 2006. FOCS'06. 47th Annual IEEE Symposium on, IEEE 2006 459–468.
- [6]. J. Benesty, J. Chen, Y. Huang, I. Cohen 2009, 1–4.
- [7]. Bittremieux W, Meysman P, Noble WS, Laukens K, *Journal of proteome research* 2018, 17, 3463. [PubMed: 30184435]
- [8]. Boja ES, Rodriguez H, *Proteomics* 2012, 12, 1093. [PubMed: 22577011]
- [9]. Brinkley T, Craft S, *Type 2 Diabetes and Dementia*, 67–86, Elsevier 2018.
- [10]. Cheng C-Y, Tsai C-F, Chen Y-J, Sung T-Y, Hsu W-L, *Journal of proteome research* 2013, 12, 2305. [PubMed: 23560440]
- [11]. Chi H, Liu C, Yang H, Zeng W-F, Wu L, Zhou W-J, Niu X-N, Ding Y-H, Zhang Y, Wang R-M, et al., *bioRxiv* 2018, 285395.
- [12]. Chum O, Philbin J, Zisserman A, et al., *BMVC*, volume 810 2008 812–815.
- [13]. Craig R, Cortens J, Fenyo D, Beavis RC, *Journal of proteome research* 2006, 5, 1843. [PubMed: 16889405]
- [14]. Dagum L, Menon R, *IEEE computational science and engineering* 1998, 5, 46.

- [15]. Datar M, Immorlica N, Indyk P, Mirrokni VS, Proceedings of the twentieth annual symposium on Computational geometry, ACM 2004 253–262.
- [16]. Dutta D, Chen T, Bioinformatics 2007, 23, 612. [PubMed: 17237061]
- [17]. Elias JE, Gygi SP, Nature methods 2007, 4, 207. [PubMed: 17327847]
- [18]. Elias JE, Gygi SP, Proteome bioinformatics, 55–71, Springer 2010.
- [19]. Eng JK, McCormack AL, Yates JR, Journal of the American Society for Mass Spectrometry 1994, 5, 976. [PubMed: 24226387]
- [20]. Frewen BE, Merrihew GE, Wu CC, Noble WS, MacCoss MJ, Analytical chemistry 2006, 78, 5678. [PubMed: 16906711]
- [21]. Griss J, Perez-Riverol Y, Lewis S, Tabb DL, Dianas JA, del Toro N, Rurik M, Walzer M, Kohlbacher O, Hermjakob H, Nature methods 2016, 13, 651. [PubMed: 27493588]
- [22]. Hein MY, Hubner NC, Poser I, Cox J, Nagaraj N, Toyoda Y, Gak IA, Weisswange I, Mansfeld J, Buchholz F, et al., Cell 2015, 163, 712. [PubMed: 26496610]
- [23]. Intasqui P, Bertolla RP, Sadi MV, Expert review of proteomics 2018, 15, 65. [PubMed: 29251021]
- [24]. Kent WJ, Genome research 2002, 12, 656. [PubMed: 11932250]
- [25]. Kim M-S, Pinto SM, Getnet D, Nirujogi RS, Manda SS, Chaerkady R, Madugundu AK, Kelkar DS, Isserlin R, Jain S, Nature 2014, 509, 575. [PubMed: 24870542]
- [26]. Kim S, Pevzner P, Ms-gf makes progress towards a universal database search tool for proteomics. nat commun 5: 5277 2014. [PubMed: 25358478]
- [27]. Kind T, Tsugawa H, Cajka T, Ma Y, Lai Z, Mehta SS, Wohlgemuth G, Barupal DK, Showalter MR, Arita M, et al., Mass spectrometry reviews 2017.
- [28]. Kong AT, Leprevost FV, Avtonomov DM, Mellacheruvu D, Nesvizhskii AI, Nature methods 2017, 14, 513. [PubMed: 28394336]
- [29]. Lam H, Deutsch EW, Eddes JS, Eng JK, King N, Stein SE, Aebersold R, Proteomics 2007, 7, 655. [PubMed: 17295354]
- [30]. Li S, Arnold RJ, Tang H, Radivojac P, Analytical chemistry 2010, 83, 790. [PubMed: 21175207]
- [31]. Liu K, Li S, Wang L, Ye Y, Tang H, Analytical Chemistry 2020, 92, 4275. [PubMed: 32053352]
- [32]. Marx H, Lemeer S, Schliep JE, Matheron L, Mohammed S, Cox J, Mann M, Heck AJ, Kuster B, Nature biotechnology 2013, 31, 557.
- [33]. Nesvizhskii AI, Nature methods 2014, 11, 1114. [PubMed: 25357241]
- [34]. Perkins DN, Pappin DJ, Creasy DM, Cottrell JS, Electrophoresis 1999, 20, 3551. [PubMed: 10612281]
- [35]. Rao PV, Lu X, Standley M, Pattee P, Neelima G, Girishesh G, Dakshinamurthy K, Roberts CT, Nagalla SR, Diabetes care 2007, 30, 629. [PubMed: 17327332]
- [36]. Rao PV, Reddy AP, Lu X, Dasari S, Krishnaprasad A, Biggs E, Roberts CT Jr, Nagalla SR, Journal of proteome research 2009, 8, 239. [PubMed: 19118452]
- [37]. Reiss SP, Renieris M, Proceedings of the 23rd International Conference on Software Engineering. ICSE 2001, IEEE 2001 221–230.
- [38]. Rifai N, Gillette MA, Carr SA, Nature biotechnology 2006, 24, 971.
- [39]. Robinson A, Cherry C, Proceedings of the IEEE 1967, 55, 356.
- [40]. Roos A, Kollipara L, Buchkremer S, Labisch T, Brauers E, Gatz C, Lentz C, Gerardo-Nava J, Weis J, Zahedi RP, Molecular neurobiology 2016, 53, 5527. [PubMed: 26468156]
- [41]. Shahaf N, Rogachev I, Heinig U, Meir S, Malitsky S, Battat M, Wyner H, Zheng S, Wehrens R, Aharoni A, Nature communications 2016, 7, 12423.
- [42]. P. Sinitcyn, J. D. Rudolph, J. Cox 2018.
- [43]. Uhlén M, Fagerberg L, Hallstrom BM, Lindskog C, Oksvold P, Mardinoglu A, Sivertsson Å, Kampf C, Sjostedt E, Asplund A, et al., Science 2015, 347, 1260419.
- [44]. Van Puyvelde B, Willems S, Gabriels R, Daled S, De Clerck L, Vande Casteele S, Staes A, Impens F, Deforce D, Martens L, et al., Proteomics 2020, 20, 1900306.
- [45]. Vinaixa M, Schymanski EL, Neumann S, Navarro M, Salek RM, Yanes O, TrAC Trends in Analytical Chemistry 2016, 78, 23.

- [46]. Vizcaíno JA, Deutsch EW, Wang R, Csordas A, Reisinger F, Rios D, Dianes JA, Sun Z, Farrah T, Bandeira N, Nature biotechnology 2014, 32, 223.
- [47]. Wang J, Perez-Santiago J, Katz JE, Mallick P, Bandeira N, Molecular & Cellular Proteomics 2010, mcp-M000136.
- [48]. Wang J, Shen HT, Song J, Ji J, arXiv preprint arXiv:14082927 2014.
- [49]. Wang L, Li S, Tang H, Journal of proteome research 2018, 18, 147. [PubMed: 30511858]
- [50]. Wang M, Carver JJ, Phelan VV, Sanchez LM, Garg N, Peng Y, Nguyen DD, Watrous J, Kapon CA, Luzzatto-Knaan T, et al., Nature biotechnology 2016, 34, 828.
- [51]. Wang M, Wang J, Carver J, Pullman BS, Cha SW, Bandeira N, Cell systems 2018, 7, 412. [PubMed: 30172843]
- [52]. Weinberger KQ, Blitzer J, Saul LK, Advances in neural information processing systems 2006 1473–1480.
- [53]. Yang X, Neta P, Stein SE, Journal of The American Society for Mass Spectrometry 2017, 28, 2280. [PubMed: 28721670]
- [54]. Yen C-Y, Houel S, Ahn NG, Old WM, Molecular & Cellular Proteomics 2011, mcp-M111.
- [55]. Zhang X, Li Y, Shao W, Lam H, Proteomics 2011, 11, 1075. [PubMed: 21298786]
- [56]. Zhou K, Hou Q, Wang R, Guo B, ACM Transactions on Graphics (TOG) 2008, 27, 126.
- [57]. Zolg DP, Wilhelm M, Schnatbaum K, Zerweck J, Knaute T, Delanghe B, Bailey DJ, Gessulat S, Ehrlich H-C, Weininger M, et al., Nature methods 2017, 14, 259. [PubMed: 28135259]
- [58]. Zolg DP, Wilhelm M, Schnatbaum K, Zerweck J, Knaute T, Delanghe B, Bailey DJ, Gessulat S, Ehrlich H-C, Weininger M, et al., Nature methods 2017, 14, 259. [PubMed: 28135259]

Statement of Significance

Due to the rapid accumulation of MS/MS spectra in the proteomics community, the spectral library searching emerges as an alternative approach to the protein database searching for peptide identification. Various spectral searching algorithms have been developed to identify the best annotated spectrum in a target spectral library for an input experimental spectrum, the key problem in spectral library searching. Despite their success in many applications, these algorithms suffers from computational inefficiency for large input datasets. In this paper, we demonstrated the *locality-sensitive hashing* (LSH) can significantly improve the *efficiency* for spectral library searching while retaining *sensitivity*. We implemented the algorithm in the *open source* msSLASH package, and showed it runs 1.6–8.9 times faster than SpectraST, while retaining more than 97% matched spectra by a naive (and slow) *spectral library searching* algorithm. We also demonstrated that the matched spectra by msSLASH largely overlap with the searching results of the naive searching algorithm and SpectraST. Therefore, msSLASH can enhance the usage of spectral library searching, for the identification of not only the peptides but for other molecules such as metabolites and natural products.

Figure 1:

Work ow of msSLASH algorithm. Decoy spectral library was constructed based on the target spectral library using the precursor swap method at spectrum level. The input query spectra together with the target and decoy spectra are assigned to buckets within one hash table, such that similar spectra are assigned into the same bucket with high probability. Multiple hash tables are then concatenated to increase the specificity for spectra identifications. The cosine similarity are computed between an input spectrum and a small subset of similar spectra in the library, and the input spectrum is identified based on the the spectrum in the target or decoy library with the highest similarity. The target-decoy search approach is employed to estimate the false discovery rate (FDR) and the identifications within 1% FDR are reported and compared with the results from the conventional database searching using MSGF+.

Figure 2:

The collision probability between two spectra under various compound LSH functions and hash tables. (a) The collision probability decreases as more hash functions (denoted by K) are concatenated in a compound hash function; (b) Augmented LSH functions (10 hash functions are concatenated per hash table) incorporating multiple hash tables (denoted by L) can increase the collision probability for similar spectra, while retaining the collision probability between dissimilar spectra relatively low.

Figure 3:

Target-decoy ratio in each LSH bucket. We converted the 8-bit string hash keys to integers. LSH with 8 hash functions was applied to the MassIVE concatenated spectral library. The LSH scheme randomly assigns a spectrum to one side of a hyperplane defined by a concatenated hash function, regardless of it being target or decoy. With the target-decoy ratio being 1:1 in each bucket, the FDR estimation is accurate.

Figure 4:
Frequency distribution of the cluster sizes (i.e. the number of spectra in each LSH bucket) vs. the number of concatenations (denoted by L) and LSH functions (denoted by K) on MassIVE spectral library.

Figure 5:
msSLASH identifies more PSMs than SpectraST on those also identified by MSGF for dataset A and B. Specifically, msSLASH identified 31.1% more PSMs within 1% FDR on dataset B-3+ than SpectraST. Overall, msSLASH identified 8.6% more PSMs than SpectraST across all datasets.

Figure 6:

1.6–8.9X speedup was achieved using msSLASH in comparison with SpectraST for performing spectral library searching. All three datasets A, B and C were searched against the NIST CID library (left) with ≈ 680 thousand spectra and the the MassIVE library with ≈ 4.2 million spectra (right). Higher acceleration rates are achieved for the large spectral library while the rates across the input data with different sizes (A, B vs. C) are similar.

Figure 7:

Venn diagrams demonstrating the overlap of matched spectra by SpectraST, msSLASH, and the naive library searching method on three testing datasets, respectively. Dataset A and B were searched against the NIST CID spectral library, while dataset C was searched against the MassIVE HCD spectral library. Spectra that were identified by both MSGF+ and spectral library searching method are retained.

The top row corresponds to doubly charged spectra across multiple datasets, and the bottom row corresponds to triply charged spectra. Each column corresponds to spectra from a dataset.

Figure 8:

Venn diagrams demonstrating the overlap of identified unique peptides by SpectraST, msSLASH, and the naive library searching method on three testing datasets, respectively. Spectra that were identified by both MSGF+ and spectral library searching method are retained.

The top row corresponds to doubly charged spectra across multiple datasets, and the bottom row corresponds to triply charged spectra. Each column corresponds to spectra from a dataset.

Table 1:

Comparison of msSLASH and SpectraST algorithms. Spectra are counted in thousand(k) or million(m). Each software was executed in a single thread and running time was measured in seconds. The datasets A and B were searched against the NIST CID spectral library, while the dataset C was searched against the MassIVE HCD spectral library. The doubly and triply charged spectra in each dataset were evaluated separately, as indicated by 2+ and 3+, respectively. All searching results were obtained at the 1% FDR. $\| \|$ represents the counts of the respective spectra. Specifically, $\| MSGF \|$ represents the number of PSMs within 1% FDR reported by MSGF.

dataset	$\ dataset \ $	library	type	$\ library \ $	$\ MSGF \ $	$Time_{SpectraST}$	$Time_{msSLASH}$
A-2+	533k	NIST	CID	679912	312714	728	368
A-3+	461k	NIST	CID	679912	194672	543	141
B-2+	14.4m	NIST	CID	679912	6434159	11390	7024
B-3+	6.5m	NIST	CID	679912	2467976	4230	1686
C-2+	4.9m	MassIVE	HCD	4215721	2350961	20810	3465
C-3+	2.6m	MassIVE	HCD	4215721	1346691	9517	1419

Table 2:

Comparison of PSM identification results within 1% FDR on searching dataset C against human theoretical proteome with MSGF, and against predicted human proteome (MS2PIP) with msSLASH. $\| \quad \|$ represents the counts of PSMs within 1% FDR. $\| \text{MSGF in Library} \|$ represents the number of PSMs identified by MSGF, whose peptides also exist in the spectral library derived from the MS2PIP predicted human proteome. $\| \text{MSGF \& msSLASH} \|$ represents the number of PSMs identified by MSGF and msSLASH as well. 7 hash functions and 75 iterations were used to search dataset C against MS2PIP with msSLASH.

dataset	$\ \text{MSGF} \ $	$\ \text{msSLASH} \ $	$\ \text{MSGF in Library} \ $	$\ \text{MSGF \& msSLASH} \ $
C-2+	2350961	4888045	2268708	1957316
C-3+	1346691	2616717	1264608	1050131