Massively Parallel Open Modification Spectral Library Searching with Hyperdimensional Computing

Jaeyoung Kang, Weihong Xu, Wout Bittremieux, Tajana Rosing
University of California, San Diego
{j5kang,wexu,wbittremieux,tajana}@ucsd.edu

ABSTRACT

Mass spectrometry, for protein identification, generates a massive number of spectra that need to be matched against a large database. In reality, most spectra remain mismatched due to unexpected post-translational modifications. Open modification search (OMS) improves the identification rate by considering every possible change in spectra, but it expands the search space exponentially. We propose HyperOMS, which redesigns OMS based on hyperdimensional computing to cope with such challenges. HyperOMS encodes floating-point spectral data with *high-dimensional binary vectors*, enabling the massive parallelism in OMS. Experimental results show that HyperOMS on GPU is up to 17× faster and 6.4× more energy efficient than the state-of-the-art GPU-based OMS tool [2] while providing comparable search quality.

1 INTRODUCTION

Tandem mass spectrometry (MS/MS) is one of the most popular and reliable methods for identifying proteins and peptides in complex biological samples in proteomics. It gathers spectrum charge, precursor m/z, and spectrum, which is a *unique fingerprint of the measured peptide*. Peptide sequences are assigned to experimental MS/MS spectra by matching them against a spectral library of known peptides. A significant portion of spectra acquired during the experiment remains unidentified due to the post-translational modifications (PTMs), which change spectra patterns. However, spectral libraries mainly contain reference spectra for unmodified peptides, so PTMs make experimental spectra challenging to identify as they no longer exactly match the reference.

Open modification searching (OMS) addresses these issues by (i) using a wide precursor m/z tolerance that exceeds mass shifts induced by modifications, and (ii) using alternative spectrum similarity measures that take peak shifts due to modifications into account [1]. Using a wide precursor m/z tolerance enables finding matches between unmodified reference spectra and their modified variants. However, OMS faces low searching speed and efficiency due to the increased search space as it considers all possible PTMs. This problem is exacerbated by the increasing spectral data due to the cost reduction of the experiment (2× in the recent two years).

Several tools have been introduced to perform OMS efficiently. The state-of-the-art OMS tool ANN-SoLos [2] performs nearest neighbor searching using GPU and computes shifted cosine similarities on candidates. However, existing solutions involve a complex execution pipeline and exhibit low data parallelism requiring high-precision floating-point (FP32) arithmetic for good search quality. As such, we redesign an OMS algorithm that only involves hardware-friendly Boolean operations with a simple execution pipeline.

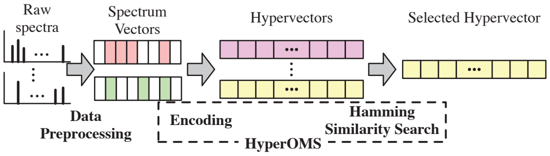


Figure 1: Overview of OMS process using HyperOMS.

In this work, we propose novel hyperdimensional computing (HDC)-inspired OMS algorithm called HyperOMS. Our algorithm is based on the efficient computing paradigm, HDC, which has shown high efficiency for pattern-matching tasks. HDC improves the data separability and robustness by mapping data into high-dimensional (HD) space. We leverage HDC's robustness to minimize the effects of PTMs. It reflects the spatial and value locality of peaks in the spectrum, making the encoded data resilient to peak shifts and intensity changes. Spectra can be identified with a single similarity computation, simplifying the execution pipeline and enhancing the data parallelism. Furthermore, HyperOMS replaces FP32 operations with Boolean arithmetic. HyperOMS on GPU achieves up to 17× speedup and 6.4× energy efficiency over the state-of-the-art GPU-based OMS solution, ANN-SoLo [2] while offering comparable search quality to other tools [1, 2, 4].

2 HYPEROMS ALGORITHM

HyperOMS encodes raw spectral data to a binary HD vector called *hypervector* (HV) to capture the position and intensities of peaks while considering the spatial and value locality. Although peaks shift due to PTMs, the similarity between a query spectrum and a matching reference spectrum does not change dramatically. Furthermore, HyperOMS replaces complex similarity metrics in existing OMS tools with a simple Hamming similarity operation.

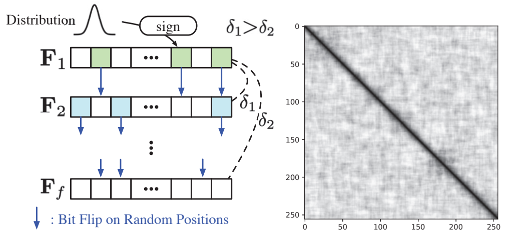
Fig. 1 shows a flow of HyperOMS. It starts with preprocessing, a common step in OMS. It refines and vectorizes raw spectra, resulting in spectrum vectors. In the **encoding** step, HyperOMS encodes the data into a *binary vector*, which can enhance the computation efficiency. There have been efforts to represent raw data in an HD binary vector, e.g., locality-sensitive hashing. However, they do not reflect the characteristics of OMS, i.e., peak shifts and intensity changes. They treat each feature position as orthogonal and peak shifts can lead to significant similarity changes. Conversely, the proposed encoding considers both *spatial locality* (for peak shift) and *value locality* (for peak intensity change) of each feature.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

PACT '22, October 10–12, 2022, Chicago, IL, USA

© 2022 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-9868-8/22/10.
https://doi.org/10.1145/3559009.3569672



(a) (b)

Figure 2: Position HV generation. (a) Strategy overview. (b) The pairwise similarity between position HVs.

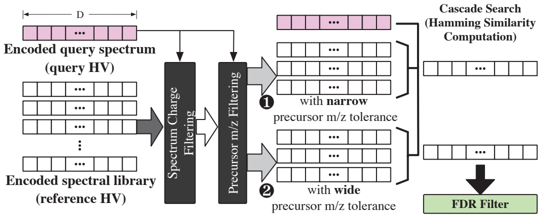

D
Encoded query spectrum
(query HV)
Encoded spectral library
(reference HV)
Spectrum Charge Filtering
Precursor m/z Filtering
❶ with narrow precursor m/z tolerance
❷ with wide precursor m/z tolerance
Cascade Search (Hamming Similarity Computation)
FDR Filter

Figure 3: Hamming similarity searching of HyperOMS.

Unique *position HVs* F are assigned for each index in a spectrum vector, i.e., $\mathbf{F}_i$ corresponds to index i, and $\mathbf{F} \in \{\mathbf{F}_1, \ldots, \mathbf{F}_f\}$ where f is the dimensionality of spectrum vector. We use *level HVs* L to capture intensities; we quantize intensity range into Q levels, and $\mathbf{L}_i$ is assigned to each level i where $i \in [0, Q]$. Given F and L, a spectrum vector is encoded into a HV I as follows: $\mathbf{I} = \sum_{(i,j)\in\mathbb{P}} \mathbf{F}_i \odot \mathbf{L}_j$ where $\odot$ indicates Hadamard product. $\mathbb{P}$ is the set of peaks in the spectrum vector, consisting of tuples (i, j), with i the peak index and j the step value of its intensity. Finally, we binarize the I.

We devise a new position HV generation method to reflect the *spatial locality* (see Fig. 2(a)). We randomly generate $\mathbf{F}_1 = \{+1, -1\}^D$ and flip α components in random positions. The similarity between the original vector and the flipped vector decreases as more flips occur. Neighboring positions have spatial locality to deal with peak shifts, while distant positions have adequate orthogonality. Position HVs do not change significantly even if peak shifts occur; the resulting representation is tolerable to PTMs. As shown in Fig. 2(b), for $\mathbf{F}_i$ and $\mathbf{F}_j$, the pairwise similarity has a high value when $i \approx j$ and is maximum when $i = j$. Note that we scaled down f and HV dimensionality D for visibility. For the *value locality*, we generate level HVs similar to [3], which reflects the closeness of the intensity.

Hamming similarity searching step finds the matched reference HV that is most similar to the query HV using *Hamming similarity*. Reference spectra primarily need to satisfy spectrum charge and precursor m/z condition. It gathers reference spectra that (1) have the same spectrum charge and (2) satisfy the precursor m/z tolerance (precursor m/z difference) condition given the query. OMS uses a wide precursor m/z tolerance to consider PTMs, but we may miss the case of a reference with a similar precursor m/z that can pass through the FDR filter with high similarity. We adopt *cascade search*. A narrow precursor m/z tolerance is used for standard search (Fig. 3-❶). The remaining unidentified spectra are processed with a wide precursor m/z tolerance (Fig. 3-❷).

3 EVALUATION

We evaluate HyperOMS on a system with Intel i7-8700K and NVIDIA Geforce GTX 1080Ti. We use small-scale and large-scale dataset used in [1, 2]. Also, we preprocess spectra in a same fashion to [1, 2, 4]. We compare the search quality of HyperOMS to existing search tools, including (1) SpectraST [4] and (2) the state-of-the-art OMS tool, ANN-SoLo [1, 2]. We count the number of identifications to compare the search quality, which is evaluated at a fixed 1% FDR threshold. Through the hyperparameter search, we set D to 8192, α to $D/2$, and Q to 16, which gives the best search quality.

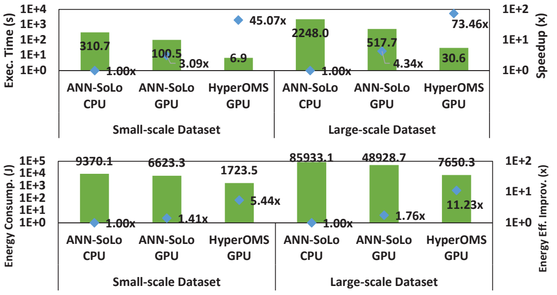


Figure 4: Performance and energy efficiency comparison.

In both dataset configurations, HyperOMS offers a higher search quality than SpectraST, i.e., HyperOMS identifies more spectra. ANN-SoLo managed to identify more spectra than our HyperOMS. HyperOMS approximates spectra in a way robust to PTMs and uses Hamming similarity. Besides, ANN-SoLo uses shifted cosine similarity metric, which is accurate when finding the original spectra. Nevertheless, the HyperOMS identification rate is within the range of the state-of-the-art in MS identification. For example, we typically expect an identification rate of 33–66% currently for human samples that we have used, and HyperOMS satisfies the expected range criterion and is practically usable in terms of search quality.

Fig. 4 compares the runtime and the energy consumption of HyperOMS on GPU to the state-of-the-art OMS tool ANN-SoLo [1, 2]. ANN-SoLo builds the index on the CPU while the encoding of HyperOMS is done on the GPU. The HyperOMS encoding is parallelized over HV dimensions and datapoints. The encoding stage of HyperOMS, which corresponds to the index build of ANN-SoLo, is up to 8.6× faster than ANN-SoLo. HyperOMS uses binary vector and easily parallelizable Hamming similarity, while ANN-SoLo uses FP32 vector. The search process of HyperOMS GPU achieves on average 82× speedup over ANN-SoLo on CPU and 11.2× speedup ANN-SoLo on GPU. Overall, HyperOMS GPU gains an average speedup of 15.7× over ANN-SoLo GPU. Besides, HyperOMS yields 7.8× and 5× energy efficiency improvement over ANN-SoLo CPU and GPU on average, respectively, as shown in Fig. 4.

ACKNOWLEDGMENTS

This work was supported in part by CRISP, one of six centers in JUMP (an SRC program sponsored by DARPA), SRC Global Research Collaboration (GRC) grant, and NSF grants #1826967, #1911095, #2052809, #2112665, #2112167, and #2100237.

REFERENCES

[1] W. Bittremieux et al. 2018. Fast Open Modification Spectral Library Searching through Approximate Nearest Neighbor Indexing. *J. Proteome Res.* (Sept. 2018).
[2] W. Bittremieux et al. 2019. Extremely Fast and Accurate Open Modification Spectral Library Searching of High-Resolution Mass Spectra Using Feature Hashing and Graphics Processing Units. *J. Proteome Res.* (Aug. 2019).
[3] Y. Kim et al. 2018. Efficient human activity recognition using hyperdimensional computing. In *IoT*.
[4] H. Lam et al. 2007. Development and validation of a spectral library searching method for peptide identification from MS/MS. *PROTEOMICS* (March 2007).