Assessing the Significance of Peptide Spectrum Match Scores*

Anastasiia Abramova[1] and Anton Korobeynikov[2]

1 **Department of Statistical Modeling, Saint Petersburg State University, Saint Petersburg, Russia; and Center for Algorithmic Biotechnology, Saint Petersburg State University, Saint Petersburg, Russia**

2 **Department of Statistical Modeling, Saint Petersburg State University, Saint Petersburg, Russia; and Center for Algorithmic Biotechnology, Saint Petersburg State University, Saint Petersburg, Russia**

Abstract

Peptidic Natural Products (PNPs) are highly sought after bioactive compounds that include many antibiotic, antiviral and antitumor agents, immunosuppressors and toxins. Even though recent advancements in mass-spectrometry have led to the development of accurate sequencing methods for nonlinear (cyclic and branch-cyclic) peptides, requiring only picograms of input material, the identification of PNPs via a database search of mass spectra remains problematic. This holds particularly true when trying to evaluate the statistical significance of Peptide Spectrum Matches (PSM) especially when working with non-linear peptides that often contain non-standard amino acids, modifications and have an overall complex structure.

In this paper we describe a new way of estimating the statistical significance of a PSM, defined by any peptide (including linear and non-linear), by using state-of-the-art Markov Chain Monte Carlo methods. In addition to the estimate itself our method also provides an uncertainty estimate in the form of confidence bounds, as well as an automatic simulation stopping rule that ensures that the sample size is sufficient to achieve the desired level of result accuracy.

1998 ACM Subject Classification G.3 [Probability and Statistics] Statistical Software, J.3 [Computer Applications] Biology and Genetics

Keywords and phrases mass spectrometry, natural products, peptide spectrum matches, statistical significance

Digital Object Identifier 10.4230/LIPIcs.WABI.2017.14

1 Introduction

Tandem mass-spectrometry (MS/MS) is an attractive alternative to nuclear magnetic resonance (NMR) spectroscopy that can be used to sequence non-linear (cyclic and branch-cyclic) peptides. Usually MS/MS is coupled with a database search algorithm capable of locating candidate peptides within the database of protein sequences, computing the peptide-spectrum match scores and estimating the statistical significance of the PSMs found.

A number of recent studies have been focusing on trying to compute the statistical significance of the PSMs. Since this particular problem is very similar to the thoroughly researched issue of having to compute the statistical significance of sequence match scores, many different approaches were proposed. For example, in [2] it was proposed to approximate

* This work was supported by the Saint Petersburg State University (grant number 15.61.951.2015).

the statistical significance of PSMs by first modeling the distribution of the PSM scores (e.g., by Gumbel distribution) and further using this distribution to calculate the probability of interest. Unfortunately, while useful in many other applications, this approximation approach, often fails when one has to estimate extremely small PSM probabilities typical for mass spectrometry (e.g., values as small as 10^{-10} are often required to achieve 1% FDR [12]).

Linear peptides and additive scoring functions use a polynomial-time algorithm [11] to compute the PSM p-values. It would seem, however, that the same approach cannot be applied to non-linear peptides. A groundbreaking breakthrough [15] gave rise to MS-DPR, an algorithm capable of computing the p-values of the PSM using the Markov Chain Direct Probability Redistribution approach. Unfortunately, while the algorithm has great appeal and appears to be quite universal, MS-DPR does not give any indication as to the accuracy of the calculated estimates and its overall performance greatly depends on the size of the sample, i.e. the number of simulations used to compute the p-value. The algorithm, however, does not provide any guidelines as to how one should go about selecting the correct size for the initial sample so as to assure quality end results.

Fortunately, the *rare probability estimation* problem itself is not new and has been very well studied within the framework of such fields as particle physics, stochastic simulation, financial mathematics, chemistry and telecommunication theory among the others.

We are using several state-of-art methods of the Monte Carlo sampling theory, including the Markov Chain Monte Carlo, importance sampling, the Wang-Landau algorithm and the efficient variance estimates for Markov Chains to derive a novel method, capable not only of estimating the statistical significance of the PSMs, but also of constructing confidence bounds for the p-value of interest and provide a way to predict the size of the sample that would be required to achieve the desired level of result accuracy.

2 Methods

2.1 Probabilistic model of a spectrum of an arbitrary peptide

We use the same probabilistic model to compute the statistical significance of PSM as presented in [15, 14]. For the sake of completeness we will describe it below.

A *PNP graph* G of a peptide P is defined as a graph with nodes $V(G)$ corresponding to amino acids in P and edges $E(G)$ corresponding to *generalized* peptide bonds [14][1]. The mass $Mass(G)$ of a PNP graph is defined as the total mass of its amino acids, i.e. $Mass(G) = \sum_{v \in V(G)} m(v)$.

A peptide bond is called a *bridge* if its removal disconnects the graph. A pair of bonds is called a *2-cut* if neither of them are bridges but removing both of them simultaneously disconnects the graph. Let $\mathcal{C}_b$ be the set of bridges of G and $\mathcal{C}_2$ be the set of pairs of 2-cut edges and we define the *set of cuts* of G as $\mathcal{C}(G) = \mathcal{C}_b(G) \cup \mathcal{C}_2(G)$.

Any cut $C' \in \mathcal{C}$ induces two masses (theoretical peaks) $m_b(C')$ and $m_y(C')$ of the connected components of G resulting from the cut C'. Note that these two peaks are *complementary* with a total mass equal to the molecular mass of the compound, $Mass(G)$. This means that for the PNP graph G and its set of cuts $\mathcal{C}$ there exist two vectors of masses $\vec{m}_b = (m_b^{(1)}, \ldots, m_b^{(|\mathcal{C}|)})$ and $\vec{m}_y = (m_y^{(1)}, \ldots, m_y^{(|\mathcal{C}|)})$. The vector $\vec{m}_b$ is called the *theoretical spectrum* of P and further be referred as $TheoreticalSpectrum(P)$.

[1] Generalized peptide bonds include N-C-O linkage amide bonds as well as C-C-O linkage bonds between thiazoles/oxazoles and dehydroalanines/dehydrobutyrines and other amino-acids. The notion of generalized peptide bonds is useful as illustrated by identification of the thiazole/oxazole containing PNP plantazolicin from *B. amyloliquefaciens*, lanthipeptide SapB from *S. coelicolor*, and complex PNPs such as two-rings containing actinomycin from *Streptomyces sp. CNS654* [14].

The $TheoreticalSpectrum(P)$ can also be represented via a *fragmentation matrix* $H = \{h_{ij}\}$ of size $|\mathcal{C}| \times |V(G)|$ with the elements $h_{ij} = 1$ if $j \in V(G_1(C^{(i)}))$ and 0 otherwise. Here $C^{(i)} \in \mathcal{C}$ and $G \setminus C^{(i)} = G_1(C^{(i)}) \cup G_2(C^{(i)})$. Rows of the fragmentation matrix correspond to different, potentially observable fragmentations. Each row specifies which amino acids are to be found on one of the connected component of graph G after the removal of some nodes. This means that $TheoreticalSpectrum(P) = H\vec{\mu}$, where $\vec{\mu}$ is a vector of the masses of the amino acids.

$SPCScore(P, Spectrum)$ is defined as the *Shared Peak Count*, the number of peaks shared between $TheoreticalSpectrum(P)$ and the filtered MS/MS spectrum $Spectrum$ [5]. Two peaks are considered as shared if their masses are within a pre-defined threshold (typically 0.02 Da for high-resolution spectra). From here on we will consider $Spectrum$ to be fixed and we will denote $Score(\vec{\mu}) = SPCScore(Spectrum, H\vec{\mu})$.

We will use $\mathcal{M}$ to denote a set of vectors that satisfy the following condition:

$$\mathcal{M} = \{\vec{\mu} = (\mu_1, \dots, \mu_{|V(G)|}), \mid \mu_i > 0, \sum_{i=1}^{|V(G)|} \mu_i = Mass(G)\}. \tag{1}$$

This set represents a variety of amino acid mass-vectors (with possible non-standard amino-acids that are typical for non-ribosomal peptides, modifications, etc. mixed in). Our goal is to calculate the probability

$$p = \mathbb{P}(SPCScore(Spectrum, H\vec{\mu}) \geq S^*) = \mathbb{P}(Score(\vec{\mu}) \geq S^*), \tag{2}$$

where $\vec{\mu}$ is a random variable uniformly distributed on set $\mathcal{M}$ and S^* is a fixed threshold (usually $S^* = SPCScore(Spectrum, P)$). Note that the probability (2) defined above depends on the particular choice of the set $\mathcal{M}$. We could obtain different models of PSM significance via changing the scoring function $SPCScore$ and/or the set $\mathcal{M}$. For example, if we consider an integer simplex $\mathcal{M}'$ (so all the μ_i would be integers), additive scoring functions and linear peptides, then we will end with the PSM statistical significance model as used by MS-GF+ [11]. The estimates presented below could easily be adopted to a different model.

2.2 Monte Carlo and the Importance Sampling Approach

The probability (2) could be estimated by using the Monte Carlo sampling approach. Consider the set

$$\mathcal{S} = \{\vec{\mu} \in \mathcal{M} : Score(\vec{\mu}) \geq S^*\}.$$

Denote by $\mathbb{1}_{\mathcal{S}}$ an indicator function of the set $\mathcal{S}$, i.e. $\mathbb{1}_{\mathcal{S}}(\vec{\mu})$ equals 1 if $\vec{\mu} \in \mathcal{S}$ (equivalently, $Score(\vec{\mu}) \geq S^*$) and 0 otherwise. Let $\vec{\mu}_1, \dots, \vec{\mu}_N$ be N iid random variables with a uniform distribution on the set $\mathcal{M}$. Then

$$\widehat{p}_{MC} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{\mathcal{S}}(\vec{\mu}_i)$$

is an unbiased and consistent estimate of p.

Variance $D(\widehat{p}_{MC})$ of $\widehat{p}_{MC}$ equals to $\frac{p(1-p)}{N}$ and tends to 0 as $N \to \infty$. However, its relative error

$$RE(\widehat{p}_{MC}) = \frac{D(\widehat{p}_{MC})}{p^2} = \frac{p(1-p)}{Np^2} = \frac{1}{Np} - \frac{1}{N} \longrightarrow \infty, \; p \to 0, \tag{3}$$

is unbounded indicating that the performance deteriorates when the event is rare. For example, if a relative error at 1% is desired and the probability is of order 10^{-6} then we

need to take N such that $\sqrt{(10^6-1)/N} \leq 0.01$. This implies that $N \approx 10^{10}$ which is an unfeasible task for most computer systems. Therefore we need to look for an estimate that would have its variance smaller than $D(\widehat{p}_{MC})$, ideally having the relative error bounded (or at least growing much slower).

Importance sampling is a general Monte-Carlo approach to reduce the variance in the estimation of quantities that can be written as an expectations. Importance sampling generates the "interesting" events more often by sampling from a different distribution and correcting for this bias afterward, which results in a more accurate estimate with a reasonable number of samples.

Formally, let $\mathcal{P}$ be the distribution of $\vec{\mu}_i$ and f the corresponding density. Consider another distribution $\mathcal{Q}$ with density q. Let $\vec{\nu}_1, \ldots, \vec{\nu}_N$ be a sequence of iid random variables having the distribution $\mathcal{Q}$. Then *importance sampling estimator* of p is defined as

$$\widehat{p}_{IS} = \frac{1}{N} \sum_{i=1}^{N} \frac{f(\vec{\nu}_i)}{q(\vec{\nu}_i)} \mathbb{1}_{\mathcal{S}}(\vec{\nu}_i). \tag{4}$$

Note that this estimator is also consistent and unbiased.

Suppose that the density q has the following form:

$$q(x) = cw(x)f(x), \tag{5}$$

where $c > 0$ is a normalizing constant and $w(x)$ is some biasing factor. Then $\widehat{p}_{IS}$ would not depend on f and could be written as

$$\widehat{p}_{IS} = \frac{\sum_{i=1}^{N} \mathbb{1}_{\mathcal{S}}(\vec{\nu}_i)/w(\vec{\nu}_i)}{\sum_{i=1}^{N} 1/w(\vec{\nu}_i)}, \tag{6}$$

where $\vec{\nu}_1, \ldots, \vec{\nu}_N$ is a sequence of iid random variables with density function q.

2.3 Metropolis-Hastings Algorithm with Wang-Landau weighting

In order to calculate $\widehat{p}_{IS}$ we need to sample from the distribution $\mathcal{Q}$ with density (5). This might be a non-trivial task, since the normalizing constant c of $q(x)$ is unknown and weights $w(x)$ could be arbitrary.

Usually this task is approached by using the Metropolis-Hastings [7] algorithm that allows for the usage of unnormalized densities. Our goal is to construct a Markov Chain having $\mathcal{Q}$ as an equilibrium distribution and obtain the desired sequence $\{\vec{\nu}_n\}$ of random variables via sampling from this distribution. Then the ergodicity of the Markov chain will ensure that as $N \to \infty$ still $\widehat{p}_{IS}$ will converge to the target probability p almost surely [18, 17].

Algorithm 1: Metropolis-Hastings algorithm

Input : Transition kernel $\gamma(x|y)$, current state of Markov Chain $\vec{\nu}_i$
Output : Next state of Markov-Chain $\vec{\nu}_{i+1}$

Sample random variable $\vec{\nu}$ from conditional probability distribution $\gamma(\cdot|\vec{\nu}_i)$
Sample uniform random variable r on the interval $[0; 1]$
Calculate *the acceptance ratio* $\alpha = \min\left(\frac{q(\vec{\nu}_i)\gamma(\vec{\nu}|\vec{\nu}_i)}{q(\vec{\nu})\gamma(\vec{\nu}_i|\vec{\nu})}, 1\right)$
if $r < \alpha$ **then**
    $\vec{\nu}_{i+1} \leftarrow \vec{\nu}$
else
    $\vec{\nu}_{i+1} \leftarrow \vec{\nu}_i$
end

The density q here is defined by (5) and sampling from proposal density $\gamma(\cdot|\vec{\nu}_i)$ is performed as follows:

Algorithm 2: Simulation from conditional density $\gamma(\cdot|\vec{\nu}_i)$

Input : Current state of Markov Chain $\vec{\nu} = (\nu_1, \ldots, \nu_{|V(G)|}) \in \mathcal{M}$
Output : Proposed state $\tilde{\nu}$

Sample index i uniformly on $\{1, \ldots, |E(G)|\}$
Consider $e_i \in E(G)$. Denote by v_1 a starting vertex of e_i and by v_2 an ending vertex
Sample δ uniformly on $[-m(v_1); m(v_2)]$
Set $\tilde{\nu}_{v_1} \leftarrow \nu_{v_1} + \delta$, $\tilde{\nu}_{v_2} \leftarrow \nu_{v_2} - \delta$, and $\tilde{\nu}_j \leftarrow \nu_j$ for the rest j

The density f in our case is a uniform density on the set $\mathcal{M}$ and it is natural to assume that the weights $w(\vec{\nu})$ should be score-invariant. Therefore to calculate $\widehat{p}_{IS}$ we will consider a proposal distribution $\mathcal{Q}$ with the density $q(\vec{\nu}) = cw(Score(\vec{\nu}))f(\vec{\nu})$. In order to decrease the relative error $RE(\widehat{p}_{IS})$ we aim to choose $w(S) \approx 1/\mathbb{P}(Score(\vec{\nu}) = S)$. This way sampling from $\mathcal{Q}$ will yield a flat score distribution reducing the variance of $\widehat{p}_{IS}$.

For this purpose we adapt a variant of Wang-Landau algorithm [13, 19]. This algorithm is an adaptive modification of the Metropolis-Hastings algorithm, which can simultaneously construct the Markov Chain and estimates weights. We use this algorithm, however, only to estimate weights because the resulting random walk is not even Markovian and therefore one could not guarantee the consistency of the estimates and lack of bias.

Algorithm 3: Wang-Landau algorithm

Input : Minimum and maximum values of log-weight increments C_{min}, C_{max}
Output : Set of weights $w(S)$

Set $w[i] \leftarrow 0$ for $i \in S_{min}, \ldots, S_{max}$, where S_{min} and S_{max} are minimum and maximum scores correspondingly
$C \leftarrow C_{max}$
while $C > C_{min}$ **do**
  Set $Hist[i] \leftarrow 0$ for $i \in S_{min}, \ldots, S_{max}$
  Simulate $\vec{\nu}$ uniformly on $\mathcal{M}$
  while *Hist is not sufficiently flat* **do**
    Run a step of Metropolis-Hastings algorithm with density $q(\vec{\nu}) = cw(Score(\vec{\nu}))f(\vec{\nu})$. Denote obtained new state by $\tilde{\nu}$
    $w[Score(\tilde{\nu})] \leftarrow w[Score(\tilde{\nu})]/C$
    $Hist[Score(\tilde{\nu})] \leftarrow Hist[Score(\tilde{\nu})] + 1$
  end
  $C \leftarrow \sqrt{C}$
end
return $w(S)$

Typically we use $C_{min} = \exp(0.6)$, $C_{max} = \exp(0.0000367)$. The criterion for a "sufficiently flat" histogram is that counts in every bin of the histogram are larger than 70% and smaller than 130% of the value expected in a perfectly flat histogram.

Use of Metropolis-Hastings algorithm coupled with Wang-Landau sampling is a common technique used recently for rare event sampling (see [9] for an extensible review). In [21] it was used to calculate the probabilities of sequence local alignment scores (however, neither accuracy estimates in the form of variance nor the sample sizes required to achieve the desired accuracy of estimates were given).

2.4 Variance Estimation

The estimate alone is useless without knowing how accurate it is. Regardless of the length of the simulation, there will be an unknown Monte Carlo error, $\widehat{p}_{IS} - p$. While it is impossible to assess this error directly, we can obtain its approximate sampling distribution through a Markov Chain central limit theorem (CLT) [18]. That is, if

$$\sqrt{N}\,(\widehat{p}_{IS} - p) \to N(0, \sigma_p^2),$$

as $N \to \infty$ with some $\sigma_p^2 > 0$. Denote by λ_p^2 the posterior variance associated with p. Then it is important to note that due to the correlation present in a Markov chain $\sigma_p^2 \neq \lambda_p^2$.

For now, suppose we have an estimator $\widehat{\sigma}_N^2$ such that $\widehat{\sigma}_N^2 \to \sigma_p^2$ almost surely as $N \to \infty$. This allows construction of a $(1-\delta)100\%$ confidence interval C_N for p by

$$C_N = (\widehat{p}_{IS} - z_{\delta/2}\widehat{\sigma}_N/\sqrt{N}; \widehat{p}_{IS} + z_{\delta/2}\widehat{\sigma}_N/\sqrt{N}), \tag{7}$$

where $z_{\delta/2}$ is a quantile of a standard Normal distribution. The width w_δ of C_N is given by

$$w_\delta = 2z_{\delta/2}\widehat{\sigma}_N/\sqrt{N}$$

and allows reporting the uncertainty of estimate $\widehat{p}_{IS}$.

There are many strongly consistent variance estimation techniques applicable for $\widehat{p}_{IS}$ including batch means [4, 10], spectral variance estimators[4] and regenerative simulation [8, 16].

Unfortunately, all these methods require storing the entire trajectory of the Markov chain to allow for the recalculations as the batch size increases with N. This might quickly become a problem if the fixed accuracy criterion is used as a stopping rule for the simulation process. Indeed, while storage capabilities overall are gradually becoming less and less of an issue, still, in order to obtain proper estimates in this case one would need to recalculate them over the length of the entire chain over and over again, which would make the process prohibitively computationally expensive.

Most likely the first recursive approach to update a σ_p^2 estimate when new observations come with $O(1)$ memory and computational complexity was proposed in [22]. The challenge here is to figure out a way to determine the batch sizes recursively to preserve consistency of the estimates and have a small mean square error, while simultaneously keeping the computational and computer memory requirements low. We are using a novel recursive estimator for σ_p^2 proposed in [23] that in the most situations works better than the estimator from [22] while preserving the $O(1)$ storage requirements.

2.5 Stopping Rule

In order to be able to process big MS/MS databases we need to carry out PSM significance estimation *en masse* in a fully automated manner. It follows that in this case performing chain diagnostics by hand or using a fixed time Markov chain stopping rule is out of the question. Recently in [3] an automated sequential stopping procedure was proposed that terminates the simulation when the computation uncertainty is small relative to the posterior uncertainty. In [6] it was shown that this stopping rule is equivalent to stopping when the effective sample size is sufficiently large.

Let $\widehat{\lambda}_N$ be an estimator of λ_p and consider a relative standard deviation fixed-width stopping rule, i.e.

$$N_\epsilon = \inf\left\{N > 0 : 2z_{\delta/2}\widehat{\sigma}_N/\sqrt{N} \leq \epsilon\widehat{\lambda}_N\right\}. \tag{8}$$

From [3] it follows that if $\widehat{\lambda}_N \to \lambda_p$ a.s. and $\widehat{\sigma}_N \to \sigma_p$ a.s. as $N \to \infty$, then as $\epsilon \to 0$ the simulations will terminate with probability 1 and $\mathbb{P}(p \in C_{N_\epsilon}) \to 1 - \delta$. In practice we are using $\epsilon = 0.02$ and the $\widehat{\sigma}_N$ estimate from [23].

A useful modification of this stopping criterion comes from the specific MS/MS database search problem statement. In certain situations the aim is not to estimate the probability of interest (2), but to decide whether p satisfies $p < p_0$ with p_0 being some fixed threshold. Usually p_0 is much larger compared to p (e.g. $p_0 = 10^{-7}$ and $p < 10^{-10}$). Therefore in addition to checking a condition (8) for a particular N we could also check if $p_0 \notin C_N$. If this is indeed so, then it automatically implies that either $p < p_0$ or $p > p_0$ with probability $1 - \delta$ as $N \to \infty$. This addition to the stopping rule might result in a significant reduction of the amount of simulations required, since it would depend on a much larger p_0 and not p.

2.6 Outline of the Algorithm

Gathering all the parts of the proposed method together we end with the following algorithm to compute $\widehat{p}_{IS}$.

Algorithm 4: Importance Sampling estimator for p

Input : A peptide P and spectrum *Spectrum*
Output : An estimate $\widehat{p}_{IS}$ of statistical significance p and confidence interval C_N

```
Construct PNP graph G of a peptide P and determine the set of cuts C
Construct fragmentation matrix H and let Score(μ) = SPCScore(Spectrum, Hμ)
Determine the set of weights w(S) using Wang-Landau algorithm (see algorithm 3)
while Stopping criterion (8) is not satisfied do
    Simulate next state ν⃗_N using Metropolis-Hastings algorithm (see algorithm 1)
    Update estimates σ̂_N and λ̂_N
end
Calculate p̂_IS using (6) and confidence interval C_N via (7).
```

3 Results

To confirm the validity of our approach, we have made a point to verify the accuracy of our calculations in a number of different ways. First, we have chosen a number of linear, cyclic and branch-cyclic peptides and selected several PSMs that had not extremely small probability of interest (say, within the $10^{-8} - 10^{-6}$ range). This allowed us to calculate them via direct Monte Carlo sampling, construct confidence intervals and compare the variances. For cyclic peptides we have chosen five examples from [15, Table 1], namely cyclic peptides $(10, 20, 40)$, $(10, 20, 40, 80)$, $(10, 20, 40, 80, 160)$, $(10, 20, 40, 80, 160, 320)$, and $(10, 20, 40, 80, 160, 320, 640)$. The branch-cyclic example is *Surfactin* test dataset for the Dereplicator algorithm described in [14].

We denote $\widehat{p}_{MC}$ as the probability estimate calculated via Monte Carlo sampling, $\widehat{p}_{IS}$ as the probability estimate calculated via the proposed algorithm (importance sampling via MCMC), and $\widehat{p}_{DPR}$ as the probability calculated by the MS-DPR algorithm from [14]. Note that the latter does not provide any accuracy estimate and therefore we were unable to construct confidence interval for $\widehat{p}_{DPR}$. $\widehat{p}_{MC}$ were calculated via $N = 50 \cdot 10^6$ simulations, $\widehat{p}_{IS}$ were calculated using the stopping rule (8) with $\epsilon = 0.02$, and $\widehat{p}_{DPR}$ were calculated by Dereplicator, using default settings.

Table 1 Comparison of Monte Carlo, MCMC and MS-DPR approaches: estimates.

Peptide	$\widehat{p}_{IS}$	$\widehat{p}_{MC}$	$\widehat{p}_{DPR}$
PPAEDSQK	$4.87 \cdot 10^{-7}$	$4.20 \cdot 10^{-7}$	$6.6 \cdot 10^{-7}$
GQGDPGSNPNK	$4.70 \cdot 10^{-7}$	$6.40 \cdot 10^{-7}$	$1.5 \cdot 10^{-8}$
HSNAAQTQTGEANR	$2.39 \cdot 10^{-6}$	$2.22 \cdot 10^{-6}$	$4.9 \cdot 10^{-8}$
GEEEPSQGQK	$1.03 \cdot 10^{-6}$	$1.04 \cdot 10^{-6}$	$3.6 \cdot 10^{-7}$
$(10, 20, 40)$	0.00184	0.00184	0.00197
$(10, 20, 40, 80)$	$7.35 \cdot 10^{-6}$	$7.34 \cdot 10^{-6}$	$9.36 \cdot 10^{-6}$
$(10, 20, 40, 80, 160)$	$6.76 \cdot 10^{-9}$	N/A	$4.49 \cdot 10^{-9}$
$(10, 20, 40, 80, 160, 320)$	$1.74 \cdot 10^{-12}$	N/A	$1.56 \cdot 10^{-12}$
$(10, 20, 40, 80, 160, 320, 640)$	$4.08 \cdot 10^{-16}$	N/A	N/A
Surfactin	$1.18 \cdot 10^{-5}$	$1.13 \cdot 10^{-5}$	$1.01 \cdot 10^{-5}$

Table 2 Comparison of Monte Carlo, MCMC and MS-DPR approaches: 95% confidence intervals.

Peptide	Conf. interval, $\widehat{p}_{IS}$		Conf. interval, $\widehat{p}_{MC}$	
PPAEDSQK	$4.74 \cdot 10^{-7}$	$4.99 \cdot 10^{-7}$	$2.40 \cdot 10^{-7}$	$6.00 \cdot 10^{-7}$
GQGDPGSNPNK	$4.53 \cdot 10^{-7}$	$4.87 \cdot 10^{-7}$	$4.18 \cdot 10^{-7}$	$8.62 \cdot 10^{-7}$
HSNAAQTQTGEANR	$2.30 \cdot 10^{-6}$	$2.48 \cdot 10^{-6}$	$1.81 \cdot 10^{-6}$	$2.63 \cdot 10^{-6}$
GEEEPSQGQK	$9.96 \cdot 10^{-7}$	$1.07 \cdot 10^{-6}$	$7.57 \cdot 10^{-7}$	$1.32 \cdot 10^{-6}$
$(10, 20, 40)$	$1.80 \cdot 10^{-3}$	$1.88 \cdot 10^{-3}$	$1.82 \cdot 10^{-3}$	$1.85 \cdot 10^{-3}$
$(10, 20, 40, 80)$	$7.12 \cdot 10^{-6}$	$7.58 \cdot 10^{-6}$	$6.60 \cdot 10^{-6}$	$8.10 \cdot 10^{-6}$
$(10, 20, 40, 80, 160)$	$6.4 \cdot 10^{-9}$	$7.10 \cdot 10^{-9}$	N/A	N/A
$(10, 20, 40, 80, 160, 320)$	$1.51 \cdot 10^{-12}$	$1.97 \cdot 10^{-12}$	N/A	N/A
$(10, 20, 40, 80, 160, 320, 640)$	$3.60 \cdot 10^{-16}$	$4.55 \cdot 10^{-16}$	N/A	N/A
Surfactin	$1.14 \cdot 10^{-5}$	$1.22 \cdot 10^{-5}$	$1.03 \cdot 10^{-5}$	$1.23 \cdot 10^{-5}$

Tables 1 and 2 summarize these results. As can be seen from these tables, the confidence intervals constructed from $\widehat{p}_{IS}$ lie within the confidence intervals for $\widehat{p}_{MC}$ and often have significantly smaller lengths. $\widehat{p}_{DPR}$ falls outside the confidence intervals and often is biased downwards. We must note that this property of $\widehat{p}_{DPR}$ could easily lead to false discoveries and certainly inflates the number of significant PSMs in the applications. Also, the sample size N of $50 \cdot 10^6$ was not enough to estimate $\widehat{p}_{MC}$ for $(10, 20, 40, 80, 160)$, $(10, 20, 40, 80, 160, 320)$, and $(10, 20, 40, 80, 160, 320, 640)$ and MS-DPR failed to calculate $\widehat{p}_{DPR}$ for the last peptide.

In the next series of experiments we study the variance of $\widehat{p}_{IS}$ and compare it to that of $\widehat{p}_{MC}$. In order to do so, we calculate $\widehat{p}_{MC}$ using the same number of simulations N as it was used to calculate $\widehat{p}_{IS}$[2]. Table 3 shows the reduction of variance of $\widehat{p}_{IS}$ compared to the $\widehat{p}_{MC}$. Overall, it could be observed that the smaller the probability is, the larger does the difference between the variances of MCMC and Monte Carlo estimators end up being.

Finally, in order to verify the scalability and applicability of the proposed method, the run of the Dereplicator algorithm was performed on the entirety of the Global Natural Products Social (GNPS) molecular network [20] database. This allowed us to compare the

[2] We have to increase the sample size to obtain observations with desired target score 13 to allow $\widehat{\sigma}^2_{MC}$ estimation for GQGDPGSNPNK.

Table 3 Comparison of Monte Carlo and MCMC approaches: variances.

Peptide	$\hat{\sigma}^2_{IS}$	$\hat{\sigma}^2_{MC}$	$\hat{\sigma}^2_{MC}/\hat{\sigma}^2_{IS}$	Sample Size
PPAEDSQK	$2.09 \cdot 10^{-10}$	$4.94 \cdot 10^{-7}$	2358.98	5000000
GQGDPGSNPNK	$2.33 \cdot 10^{-10}$	$1.49 \cdot 10^{-7}$	639.49	20000000[2]
HSNAAQTQTGEANR	$5.56 \cdot 10^{-9}$	$2.24 \cdot 10^{-6}$	403.23	2800000
GEEEPSQGQK	$1.23 \cdot 10^{-9}$	$7.89 \cdot 10^{-7}$	642.19	3800000
$(10, 20, 40)$	$5.47 \cdot 10^{-4}$	$1.88 \cdot 10^{-3}$	3.43	1500000
$(10, 20, 40, 80)$	$9.93 \cdot 10^{-8}$	$7.60 \cdot 10^{-6}$	76.53	7500000
Surfactin	$1.15 \cdot 10^{-7}$	$1.00 \cdot 10^{-5}$	86.96	2000000

Table 4 Comparison of $\widehat{p}_{IS}$ and $\widehat{p}_{DPR}$ on GNPS data. The number of target and decoy database matches and FDR estimates at different significance levels are shown.

	MSDPR			MCMC		
$-\log_{10} p$	target	decoy	$\widehat{FDR}$ %	target	decoy	$\widehat{FDR}$ %
7	762	188	19.78	744	179	19.39
8	619	110	15.08	610	104	14.56
9	505	52	9.33	473	51	9.73
10	443	33	6.93	415	30	6.74
11	393	21	5.07	354	20	5.34
12	354	15	4.06	312	12	3.70
13	322	11	3.30	271	7	2.51
14	293	11	3.61	238	2	0.83
15	264	7	2.58	201	1	0.49
16	238	5	2.05	169	0	0.0
17	211	2	0.93	138	0	0.0
18	188	0	0.0	104	0	0.0
19	157	0	0.0	87	0	0.0
20	139	0	0.0	76	0	0.0

devised algorithm with the MS-DPR estimate used by DEREPLICATOR by default. Table 4 shows an overview of the obtained results. The False Discovery Rate estimate is calculated in DEREPLICATOR via the target-decoy approach [1]. Table 4 shows that $\widehat{p}_{IS}$ yields a smaller number of significant decoy matches compared to $\widehat{p}_{DPR}$ and therefore less FDR. This could easily be explained by the fact that $\widehat{p}_{DPR}$ are biased downwards.

4 Summary

We presented the importance sampling-based estimator that is capable of accurately and quickly assess the significance of peptide spectrum matches. Given its generic nature, it could be easily modified to be used with a great number of different score functions, fragmentation models and amino acid mass distributions. The proposed estimation algorithm has been integrated into DEREPLICATOR[3] and VARQUEST[4] tools and publicly available as a part of these packages.

[3] `http://cab.spbu.ru/software/dereplicator/`
[4] `http://cab.spbu.ru/software/varquest/`

Acknowledgement. The authors would like to extend a special thanks to Alexey Gurevich for running DEREPLICATOR on GNPS, Seungjin Na for making the spectra and PSMs available for linear peptides and Alex Shlemov for all the fruitful discussions that were of great help in improving the algorithm.

References

1 J. E. Elias and S. P. Gygi. Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. *Nat. Methods*, 4(3):207–214, 2007.

2 David Fenyö and Ronald C. Beavis. A method for assessing the statistical significance of mass spectrometry-based protein identifications using general scoring schemes. *Analytical Chemistry*, 75(4):768–774, 2003.

3 James M. Flegal and Lei Gong. Relative fixed-width stopping rules for Markov Chain Monte Carlo simulations. *Statistica Sinica*, 25(2):655–675, 2015.

4 James M. Flegal and Galin L. Jones. Batch means and spectral variance estimators in Markov Chain Monte Carlo. *Ann. Statist.*, 38(2):1034–1070, 2010.

5 A. M. Frank. Predicting intensity ranks of peptide fragment ions. *J. Proteome Res.*, 8(5):2226–2240, 2009.

6 Lei Gong and James M. Flegal. A practical sequential stopping rule for high-dimensional Markov Chain Monte Carlo. *Journal of Computational and Graphical Statistics*, 25(3):684–700, 2016.

7 W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

8 James P. Hobert, Galin L. Jones, Brett Presnell, and Jeffrey S. Rosenthal. On the applicability of regenerative simulation in markov chain monte carlo. *Biometrika*, 89(4):731, 2002.

9 Yukito Iba, Nen Saito, and Akimasa Kitajima. Multicanonical MCMC for sampling rare events: an illustrative review. *Annals of the Institute of Statistical Mathematics*, 66(3):611–645, 2014.

10 Galin L. Jones, Murali Haran, Brian S. Caffo, and Ronald Neath. Fixed-width output analysis for Markov Chain Monte Carlo. *Journal of the American Statistical Association*, 101(476):1537–1547, 2006.

11 Sangtae Kim, Nitin Gupta, and Pavel A. Pevzner. Spectral probabilities and generating functions of tandem mass spectra: A strike against dgegeecoy databases. *Journal of Proteome Research*, 7(8):3354–3363, 2008.

12 Sangtae Kim, Nikolai Mischerikow, Nuno Bandeira, J. Daniel Navarro, Louis Wich, Shabaz Mohammed, Albert J. R. Heck, and Pavel A. Pevzner. The generating function of CID, ETD, and CID/ETD pairs of tandem mass spectra: Applications to database search. *Molecular & Cellular Proteomics*, 9(12):2840–2852, 2010.

13 D. P. Landau, Shan-Ho Tsai, and M. Exler. A new approach to Monte Carlo simulations in statistical physics: Wang-landau sampling. *American Journal of Physics*, 72(10):1294–1302, 2004.

14 H. Mohimani, A. Gurevich, A. Mikheenko, N. Garg, L. F. Nothias, A. Ninomiya, K. Takada, P. C. Dorrestein, and P. A. Pevzner. Dereplication of peptidic natural products through database search of mass spectra. *Nat. Chem. Biol.*, 13(1):30–37, 2017.

15 H. Mohimani, S. Kim, and P. A. Pevzner. A new approach to evaluating statistical significance of spectral identifications. *J. Proteome Res.*, 12(4):1560–1568, 2013.

16 Per Mykland, Luke Tierney, and Bin Yu. Regeneration in Markov chain samplers. *Journal of the American Statistical Association*, 90(429):233–241, 1995.

17 G. O. Roberts. Markov chain concepts related to sampling algorithms. In W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, editors, *Markov Chain Monte Carlo in Practice*, pages 45–58. Chapman & Hall, London, 1996.

18 Luke Tierney. Markov chains for exploring posterior distributions. *Ann. Statist.*, 22(4):1701–1728, 1994.

19 F. Wang and D. P. Landau. Efficient, multiple-range random walk algorithm to calculate the density of states. *Physical Review Letters*, 86:2050–2053, 2001.

20 M. Wang, J. J. Carver, V. V. Phelan, L. M. Sanchez, et al. Sharing and community curation of mass spectrometry data with Global Natural Products Social Molecular Networking. *Nat. Biotechnol.*, 34(8):828–837, 2016.

21 Stefan Wolfsheimer, Inke Herms, Sven Rahmann, and Alexander K. Hartmann. Accurate statistics for local sequence alignment with position-dependent scoring by rare-event sampling. *BMC Bioinformatics*, 12(1):47, 2011.

22 Wei Biao Wu. Recursive estimation of time-average variance constants. *Ann. Appl. Probab.*, 19(4):1529–1552, 2009.

23 Chun Yip Yau and Kin Wai Chan. New recursive estimators of the time-average variance constant. *Statistics and Computing*, 26(3):609–627, 2016.