

# Riptide: Fast Protein Identification from Mass Spectrometer Data

Richard J. Carter

*Computational BioScience Research, Advanced Studies*

*HP Laboratories, Palo Alto, CA*

*dick.carter@hp.com*

## Abstract

*The biotech firm Target Discovery Incorporated (TDI) has developed a relatively fast and inexpensive method for protein identification. The final step in their approach involves an algorithm to deduce the terminal amino acid sequence of an unknown intact protein from its fragmentation mass spectrum. TDI's web-published algorithm was taken as a starting point for further research. The algorithm Riptide was developed that matches the output of TDI's algorithm, but demonstrates a 193X speed improvement on a 6-deep sequencing.*

## 1. Introduction

Mass spectrometry is used in a variety of methods for protein identification including peptide mass mapping and mass ladder sequencing [1]. The biotech firm Target Discovery Inc. (TDI) has developed one approach it terms Inverted Mass Ladder Sequencing (IMLS™) in which protein identification is made based on a deduced terminal amino acid sequence [2]. With IMLS, a label molecule is chemically attached to one end of the proteins in a mixture of proteins. Next, the proteins are separated and each is brought to a gaseous and ionized state. Each intact protein is then fragmented by the nozzle potential of a mass spectrometer, which outputs the  $m/z$  (mass/charge) ratios of the protein fragments. Finally, an algorithm described on the company's website (henceforth referred to as the "prior art" algorithm) is used to derive a terminal amino acid sequence. In about 90% of the cases, knowing the sequence of the first 6 amino acids from one end of a protein is sufficient to identify the protein from a database of known proteins. IMLS is described as offering at least a 100-fold improvement in the protein sample processing time [2]. This paper describes this author's research toward making commensurate speed improvements to the offered prior art sequencing algorithm.

## 2. Prior art sequencing algorithm

Riptide is an algorithm that solves the problem of deducing the most likely sequence of amino acid residues

that comprise one end of a protein from the mass spectrum of the fragments of that protein. Since Riptide and the prior art algorithm produce the same answer (by design), it is useful to describe the simpler prior art algorithm first after establishing some definitions.

Assume that the label-attached protein molecule is depicted as  $LABEL-R_1-R_2-R_3-R_4-R_5-R_6 \dots$  with the  $n^{th}$  amino acid residue as measured from the label-attached end of the protein referred to symbolically as  $R_n$ . The fragmentation process that occurs in the mass spectrometer typically cleaves the protein singly along the protein "backbone", so that after fragmentation one would find a relative abundance of the following molecules:  $LABEL$ ,  $LABEL-R_1$ ,  $LABEL-R_1-R_2$ ,  $LABEL-R_1-R_2-R_3$ ,  $LABEL-R_1-R_2-R_3-R_4$ ,  $LABEL-R_1-R_2-R_3-R_4-R_5$ , etc.

The prior art algorithm converts a hypothetical candidate sequence into a ranking through a "ranking function" that seeks to reward the relative abundance of each of these expected fragments in the mass spectrometer dataset. Given say the problem of determining the first 6 terminal amino acid residues of a protein, the prior art ranking function of a hypothetical candidate sequence  $R_1-R_2-R_3-R_4-R_5-R_6$  is:

$$\begin{aligned} \text{Ranking}(R_1, R_2, R_3, R_4, R_5, R_6) = & \\ & MS(m_{LABEL} + m_{R_1}) + \\ & MS(m_{LABEL} + m_{R_1} + m_{R_2}) + \\ & MS(m_{LABEL} + m_{R_1} + m_{R_2} + m_{R_3}) + \\ & MS(m_{LABEL} + m_{R_1} + m_{R_2} + m_{R_3} + m_{R_4}) + \\ & MS(m_{LABEL} + m_{R_1} + m_{R_2} + m_{R_3} + m_{R_4} + m_{R_5}) + \\ & MS(m_{LABEL} + m_{R_1} + m_{R_2} + m_{R_3} + m_{R_4} + m_{R_5} + m_{R_6}) \end{aligned}$$

We assume here the  $LABEL$  has mass  $m_{LABEL}$ , an amino acid residue  $R$  has mass  $m_R$ , and the fragmentation molecules occur primarily as singly charged ions. Further assumed is a "Mass Spectrum look-up function" named  $MS()$ . This function takes an  $m/z$  value and converts it into a score value based on the current mass spectrum dataset. This function is computationally intensive and involves more than simply a look-up of the occurrences at the given  $m/z$ . It includes interpolation between actual

mass spectrometer output samples, backbone fragmentation variation effects (ion classes) and statistical normalization of occurrence counts. Minimizing the number of  $MS()$  calls is an important aspect of Riptide's speed improvements over the prior art algorithm.

The prior art algorithm evaluates the above ranking function for each of the  $19^6$  sequence permutations of 6 amino acid residues (19, not 20, because leucine and isoleucine have the same mass and are lumped together). Further, the algorithm names the highest-ranked sequence as the predicted sequence for the unknown protein. On the surface, it would appear that each  $Ranking()$  function evaluation would require 6  $MS()$  calls. However, the first 5  $MS()$  call results can be shared amongst all sequences having the same first 5 amino acid residues. Only the last  $MS()$  call is made uniquely for each sequence permutation in the prior art algorithm's nested-loop approach to evaluating the sequence space. Thus the prior art algorithm, in evaluating  $19^6$   $Ranking()$  functions, makes 49,659,540  $MS()$  calls.

### 3. Riptide sequencing algorithm

Riptide is faster than the prior art algorithm because it performs its ranking function evaluations, not for every amino acid sequence *permutation*, but for every amino acid *combination*. For sequencing depths of interest, the number of permutations is vastly greater than the number of combinations. Despite the apparent simplification of the problem, Riptide is able to derive the best-ranked *ordered* amino acid sequence from its database of amino acid combination rankings with little extra computation.

Revisiting the 6-deep sequencing problem presented in the prior art algorithm description, Riptide would only need to evaluate 134,596  $Ranking()$  functions, and would make 177,099  $MS()$  calls. Riptide's theoretical speed improvement over the prior art algorithm for the 6-deep sequencing is thus a factor of  $49,659,540 / 177,099 = 280$ . This speed advantage could translate to a cheaper computer system for a given sequencing throughput goal. Alternatively, Riptide's advantage could be used to perform a deeper sequencing for a given computer system and this could result in a higher probability of a unique or complete protein identification.

To explore further where the Riptide speed advantage

comes from, consider the last term of the prior art algorithm's ranking function:

$$Ranking(R_1, R_2, R_3, R_4, R_5, R_6) = \dots + MS(m_{LABEL} + m_{R_1} + m_{R_2} + m_{R_3} + m_{R_4} + m_{R_5} + m_{R_6})$$

The last  $MS()$  call has an argument value that is independent of the ordering of the ranking function arguments, because of the commutativity of addition. Riptide effectively makes this  $MS()$  call once for all the sequences that are re-orderings (i.e. permutations) of each other. The prior art algorithm needlessly repeats this  $MS()$  function call for all of these sequence permutations.

### 4. Results

The Riptide algorithm and the prior art algorithm were both coded in the C-language and timings made on an HP XW8000 PC Workstation (2.8GHz Xeon) with 1.5GB main memory. The programs were compiled with "gcc -O2". Table 1 shows the relative speed-up achieved by Riptide over the prior art algorithm for various interesting sequencing depths.

### 5. Conclusions

The presented Riptide algorithm offers a novel speed improvement over protein sequencing algorithms that seek the highest-ranked amino acid sequence over the amino acid permutation space. Riptide demonstrates that the maximum of certain ranking functions can be more quickly determined by a traversal of the amino acid combination space. Actual timed improvements were consistent with the calculated reduction in calls to the compute-intensive mass spectrum look-up function  $MS()$ .

### 6. References

- [1] Michael Kinter and Nicholas E. Sherman, *Protein Sequencing and Identification Using Tandem Mass Spectrometry*, John Wiley & Sons, Inc., NY, NY, Sept. 2000.
- [2] [www.targetdiscovery.com/sys-tmpl/nss-folder/asmspowerpointpdf/ASMS.ppt](http://www.targetdiscovery.com/sys-tmpl/nss-folder/asmspowerpointpdf/ASMS.ppt)

**Table 1. Riptide speed-up from timed C programs on an HP XW8000 (2.8GHz Xeon)**

Scenario	Prior art algorithm	Riptide algorithm	Actual Speed-up
19 amino acids, 5-deep sequencing	0.58 sec	0.014 sec	41X
19 amino acids, 6-deep sequencing	11 sec	0.057 sec	193X
19 amino acids, 7-deep sequencing	192 sec	0.211 sec	910X