

# Computational methods enabling genome-based protein identification from large, complex genomes using mass spectrometry data

Michael S. Wisz, Jainab Khatun, Morgan C. Giddings  
University of North Carolina at Chapel Hill

Departments of Microbiology & Immunology and Biomedical Engineering

**The interpretation of mass spectrometry data for protein identification has become a vital component of proteomics research. Most current methods that transform a peptide mass fingerprint or collection of tandem mass spectra into identification of the protein(s) in a potentially complex sample are dependent on the quality of the protein database available for the organism under study. This presents a major impediment in many cases, since genome sequencing is progressing at an impressive rate, and annotation efforts are relatively slow and can be inaccurate. This problem is particularly pronounced for the larger, recently sequenced genomes containing multi-exon genes, where the annotation process is still in its early stages, and thus the potential for false negative identifications is high.**

## I. INTRODUCTION

We previously introduced a method, Genome Fingerprint Scanning[1], that circumvents the reliance on protein databases by mapping mass spectrometric data directly to unannotated genomic sequence to identify the coding locus of sample protein(s). It has performed well in studies involving relatively small genomes (*E. coli* and *S. cerevisiae*), but has had limited application to identifying the coding locus for protein samples expressed by larger genomes (*i.e.*, *Mus musculus*, *Homo sapiens*), due to their size ( $> 10^7$  nt) and more complex gene structure (multiple exons). The first hurdle that must be overcome to effectively apply such algorithms on a genome-wide scale is the computational bottleneck associated with enormous map of possible peptides generated by these large genomes.

## II. METHODS

In order to identify the coding locus for a protein, GFS searches the set of all putative peptide fragments obtained from translating and performing an *in silico* proteolytic digestion (starting peptides at

start codons, ending peptides after lysine or arginine for trypsin) of the genome in 6 reading frames (3 forward and 3 reverse). The number of putative peptides encoded by the entire Human Genome, allowing for 2 missed cleavages per peptide, is  $\sim 2 \times 10^9$ . While both the *time* to match theoretical fragment masses with experimental masses and the *memory* required to load all the sequence, fragment, and associated data are linearly proportional to the size of the genome being studied, the  $\sim 10^3$  scale difference between small and large genomes renders these processes impractical for the large genomes.

Since computer memory is usually limited to only a few gigabytes on most desktop-class machines, calculating and matching peptide maps for a large genome within memory is not possible as it is for the smaller genomes. To illustrate this point, consider the complete 6-frame trypsin digestion of the human genome, which generates  $\sim 2 \times 10^9$  peptide fragments. Each of these takes uses 34 bytes of memory using a compact representation (information on fragment location, length, mass, frame, number of missed cleavages, number of oxidized methionies is encoded), requiring 68 GB of memory for the peptide data. We therefore resort to pre-calculating a database of possible peptides for the genome of interest, and storing it on disk. However, the relative slowness of disks further compounds the speed limitations of handling such large peptide maps.

The time-limiting step is matching peptides against raw genomic sequence. This matching process is  $O(MN)$ , where  $M$  is the size of the mass list and  $N$  is number of peptide fragments produced by the genome under study. Therefore, if it takes  $\sim 10$  seconds to match 100 masses against the  $\sim 4 \times 10^6$  peptides of the *E. coli* genome digest with 2 missed cleavages, it will take  $\sim 83$  hours to perform a this process on the entire human genome, not accounting for the further speed hit of disk access required for the large peptide map. This is unacceptable when analyzing mass spectra in a high-throughput environment. We

tested the feasibility and performance of indexing peptide masses in a B-tree structure[2] with 100 masses per node. The B-tree is arranged such that any given node will contain 100 float values (representing peptide masses) in ascending order, and a pointer positioned to the right of each mass value, giving 101 total pointers (there is one extra pointer to the left of the first value). A given pointer points to another identically structured node, but whose values are all floats greater than the value to the pointer's left but less than the value to the pointer's right. The entire peptide mass B-tree structure is stored on disk, and only the necessary nodes are read into memory at any given time, reducing the memory footprint to < 1 kB from a potential 32 GB.

### III. RESULTS AND DISCUSSION

In the experiment performed here, 100 randomly generated peptide masses were matched against a varying number ( $4 \times 10^6$  to  $21 \times 10^6$ ) of disk cached peptide masses from the mouse genome, with a mass tolerance of 100 ppm, and the matching time for linear search (without a B-tree) was then compared to the time for B-tree search. In theory, applying a B-tree to such a matching problem will reduce matching speed complexity from  $O(MN)$  to  $O(\log_{100}[N])$ .

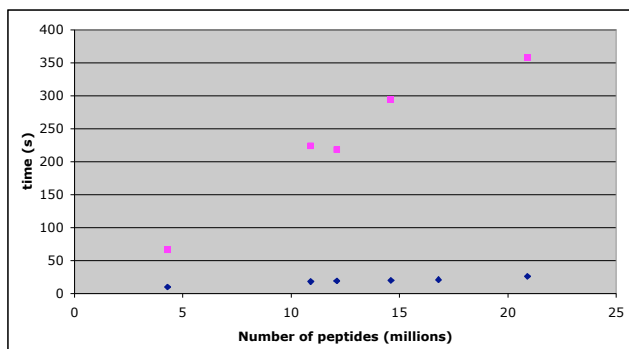


Fig 1. Linear peptide matching (pink squares) versus B-tree search matching (blue diamonds)

As shown in Figure 1, comparing the linear matching approach (pink squares) with the B-tree scheme (blue diamonds), our B-tree solution translates into a 10-fold speed improvement for ~20 million peptides. Extrapolating to an entire large genome means the conversion of a problem taking hours to one taking minutes. This has overcome a significant hurdle in genome-based protein identification, allowing us to begin testing protein matches in large genomes. We have already com-

bined this approach with a scheme for scoring peptide matches in multiple exons, and then narrowing down potential gene candidates by using sequence tag data obtained from tandem MS, ultimately leading us to the correct identification of a single gene in an experiment involving the entire mouse genome. Furthermore, we hope to advance this approach to identify proteins from alternatively spliced genes, currently an unsolved problem in proteomics.

### REFERENCES

- [1] Giddings, M.C.; Shah, A.A.; Gesteland, R.F.; Moore, M., "Genome-based peptide fingerprint scanning", *Proc. Natl. Acad. Sci. U.S.A.*, 100, 20-25, 2003.
- [2] Aho, A.V.; Hopcroft, J.E.; Ullman, J.D., *Data Structures and Algorithms*. 1983: Bell Telephone Laboratories, Inc.