NRC·CNRC

NRC Publications Archive
Archives des publications du CNRC

Simple Self-Adjusting Data Structure Use: An Empirical Investigation of the Unsupervised Construction of Conceptual Units from Mass Spectrometry Data

Barton, Alan

For the publisher's version, please access the DOI link below./ Pour consulter la version de l'éditeur, utilisez le lien DOI ci-dessous.

https://doi.org/10.4224/8913672

NRC Publications Record / Notice d'Archives des publications de CNRC:
https://nrc-publications.canada.ca/eng/view/object/?id=94664e77-1490-47ed-8c07-81b0e34d83a4
https://publications-cnrc.canada.ca/fra/voir/objet/?id=94664e77-1490-47ed-8c07-81b0e34d83a4

Access and use of this website and the material on it are subject to the Terms and Conditions set forth at
https://nrc-publications.canada.ca/eng/copyright
READ THESE TERMS AND CONDITIONS CAREFULLY BEFORE USING THIS WEBSITE.

L'accès à ce site Web et l'utilisation de son contenu sont assujettis aux conditions présentées dans le site
https://publications-cnrc.canada.ca/fra/droits
LISEZ CES CONDITIONS ATTENTIVEMENT AVANT D'UTILISER CE SITE WEB.

Questions? Contact the NRC Publications Archive team at PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca. If you wish to email the authors directly, please see the first page of the publication for their contact information.

Vous avez des questions? Nous pouvons vous aider. Pour communiquer directement avec un auteur, consultez la première page de la revue dans laquelle son article a été publié afin de trouver ses coordonnées. Si vous n'arrivez pas à les repérer, communiquez avec nous à PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca.

National Research Council Canada | Conseil national de recherches Canada

Canada

National Research Council Canada
Conseil national de recherches Canada

Institute for Information Technology
Institut de technologie de l'information

NRC · CNRC

Simple Self-Adjusting Data Structure Use: An Empirical Investigation of the Unsupervised Construction of Conceptual Units from Mass Spectrometry Data *

Barton, A.
May 2006

* published as NRC/ERB-1139. May 2006. 10 Pages. NRC 48728.

Copyright 2006 by
National Research Council of Canada

Permission is granted to quote short excerpts and to reproduce figures and tables from this report, provided that the source of such material is fully acknowledged.

Canada

National Research Council Canada
Conseil national de recherches Canada

Institute for Information Technology
Institut de technologie de l'information

ERB-1139

NRC·CNRC

Simple Self-Adjusting Data Structure Use: An Empirical Investigation of the Unsupervised Construction of Conceptual Units from Mass Spectometry Data

Barton, A.
May 2006

Copyright 2006 by
National Research Council of Canada

Permission is granted to quote short excerpts and to reproduce figures and tables from this report, provided that the source of such material is fully acknowledged.

Canada

NRC 48728

Simple Self-Adjusting Data Structure Use: An Empirical Investigation of the Unsupervised Construction of Conceptual Units from Mass Spectrometry Data*

Alan J. Barton†
National Research Council Canada
Institute for Information Technology
M50, 1200 Montreal Rd.
Ottawa, ON K1A 0R6
alan.barton@nrc-cnrc.gc.ca

ABSTRACT

One stage, of many, in the collection and analysis of mass spectrometry data is chosen for this investigative study. Closely related problems that are of interest for this research are the determination of: *i)* which points in the plane should have membership to which lines, *ii)* when a collection of points should be described as a line, and *iii)* how one knows that all lines have been found and that none have been missed in the particular data studied. This paper presents a brief survey of possibly related solutions along with the construction, experimentation and results of the use of a simple data structure for attempting to provide answers to the aforementioned problems. Future studies involving the aid of domain experts is required in order to further elaborate these preliminary time-constrained findings.

Keywords

data structures, applied problem domain, mass spectrometry data

1. INTRODUCTION

In the design of a solution to both theoretical and applied problems, a computer scientist usually faces a decision point in which the selection of the most appropriate data structure will need to be made. Various aspects of the particular problem under study are taken into account during the construction of such a potential solution, including: *i)* conflicting problem goals possibly requiring solution tradeoffs, *ii)* theoretical considerations from the mind of a capable theoretician, and *iii)* applied constraints imposed by observations of the natural world. Once the data structure has been chosen, a theoretical or applied algorithmic solution may be derived that is as efficient as possible in terms of, for example, both memory usage and query/search times. To know whether one has reached the "efficient as possible" lower bound one usually requires a restriction of a solution to a particular model of computation, in which a bound may be possibly proven, for example in the asymptotic or amortized [13] senses. This paper does not attempt to theoretically prove lower or upper bounds of efficiency. Rather, it hopes to attempt to properly empirically investigate one particular data set's properties in order to aid the selection of a data structure that will lead to an efficient real world solution for this one particular problem related to mass spectrometry data.

*Report for the course COMP5408 entitled *Advanced Data Structures* at Carleton University

†Master of Computer Science (in progress)

In the study of real world data, one may start by asking the question, "What is data?" To begin to answer the question, it may be noted that a datum may exist in different forms in many possible spaces containing various kinds of structure. To be more specific, a datum is a statement accepted at face value (a "given"), with a large class of practically important statements being measurements or observations of a variable[15]. Such a set of datum is called a space if the points are endowed with a structure[2]; and in Mathematics, a structure on a set, or more generally a type, consists of additional mathematical objects that in some manner attach to the set, making it easier to visualize or work with, or endowing the collection with meaning or significance (e.g. metric structures – geometries) [14].

For the particular case of Computer Science, the concepts of "data" and "structure" may be combined in order to yield a data structure. This is an organization of information for better algorithm efficiency (e.g. queue, dictionary, tree, etc.) or conceptual unity (e.g. name and address of a person). It may include redundant information (e.g. number of nodes in a subtree). [1]

The goal of this paper is to preliminarily investigate possible solutions by presenting a survey of potentially related work and experimenting with implementation variants for the purpose of improving the collection of peptide ions measured using a high pressure liquid chromatography (HPLC) mass spectrometry system. In particular, a comparison with results obtained using a previously reported domain heuristic based implementation ([8], [9]) is of interest; but not possible without the aid of the domain expert, and so will not be reported.

2. THE PROBLEM

The overall problem may be taken from four different perspectives: *i) a Biological problem* in which particular types of molecules are investigated with a measuring device (such as a mass spectrometer) for the purpose of further understanding a disease, biological process, etc. *ii) an optimization problem* in which an unknown number of lines need to be found such that the lines properly represent a particular data set. *iii)* a *geometrical problem* in which points are assigned to horizontal lines. However, the sense of a geometrical line needs to be broadened to allow points to lie on either side of the line and not directly on the line as traditionally defined. *iv)* a *data structures problem* in which both efficient use of memory and rapid data access times (e.g. query/search times) are reduced as much as possible for a particular algorithm that would need to be designed for use in a computer.

To be specific, the particular problems are: *i)* Given a point in the plane, to what line should it have membership? *ii)* Given a collection of points, when should all (or a subset) of the points be promoted to the concept of a line? and *iii)* Given a collection of points (a data set), how does one know that all lines have been discovered and that none have been missed?

3. RELATED WORK

Some related work is discussed.

For the general problem of feature extraction [10] cites Nilsson who comments that:

1. No general theory exists to allow us to choose what features are relevant for a particular problem.
2. Design of feature extractors is empirical and uses many ad hoc strategies.
3. We can get some guidance from biological prototypes.

A further point is made in [10] who cites Selfridge and Neisser:

> At present the only way the machine can get an adequate set of features is from a human programmer. The effectiveness of any particular set can be demonstrated only by experiment. In general there is probably safety in numbers. The designer will do well to include all the features he can think of that might plausibly be useful.

In addition, such operations as smoothing, thinning and/or filtering [10] may be applied when dealing with real data. Or the concept of edging [10] such as that employed by the SLEN (short line extractor neuron), which acts as an optical edge detector, may be applied, but with difficulty [10] if the data is noisy, and especially if the edges are irregular and occur at various angles. They state that "the main criteria in making a choice in a particular case are simplicity and the amount of computer time required".

Possibly one way to decide to which point a line belongs would be to apply the nearest neighbour rule [4], [3] which states "the nearest neighbour decision rule assigns to an unclassified sample point the classification of the nearest of a set of previously classified points". The only drawback with this supervised approach, is that the problem within this paper is an unsupervised one.

The range queries that are possible in space $\theta(N)$ [16] seem to potentially not be applicable given a set of N, unbounded reals. Of course an artificial bound may be applied and then changed over time.

A classical pattern classification book[5] contains the following 3 algorithms: *i)* The minimum squared error line fitting algorithm. Given a set of points $(x_i, y_i), i = 1, \ldots, n$ in the plane, find two numbers c_0 and c_1 such that the following error function is minimum:

$$\sum_{i=1}^{n} [(c_0 + c_1 x_i) - y_i]^2$$

In other words (they state), find a straight line such that the sum of squares of the vertical distances from each point to the line is minimum. *ii)* Eigenvector line fitting[5] in which the perpendicular (to the line) distance is wanted to be minimized, and *iii)* Line fitting by clustering[5] in which a set of points is partitioned such that each partition is reasonably represented by a single line.

Another important related work[11] contains a chapter on hash functions, which could facilitate a particular algorithm's reduction in execution time, due to their O(1) nature.

Range searching[6]: Let S be a set of n points in $\mathbb{R}^d$, and let $\Re$ be a family of subsets of $\mathbb{R}^d$; elements of $\Re$ are called *ranges*. The goal is to preprocess S into a data structure so that for a query range R, the points in $S \cap R$ can be reported or counted efficiently.

Geometric properties of sets of lines[12] is interesting from the point of view that a Möbius Hough space is described. It also discusses a fuzzy subset of Hough space, which is interesting because instead of the classical notions of geometry, the Fuzzy Sets notion [17] and appropriate previously extended work, such as the work of Fuzzy plane projective geometry [7] could also be used to fuzzify the crisp mass spectrometry data. This would be interesting work to pursue, but time constraints do not allow it. One idea would be to define triangular (or trapezoidal or guassian, etc) fuzzy numbers for each point measured by the mass spectrometer and perform experiments in order to analyse which approach may lead to better results. Potentially such things as range queries might also be investigated with respect to fuzzification.

3.1 Previous Work

Previous work published in the proteomics and systems biology communities [8], [9] has been made. That approach was constructed by domain experts and might be very loosely described as a point-based algorithm.

ALGORITHM 1. A domain heuristic approach.

(0) Determine specific charge values (z) for each of the measured ions via the following equation:

$$\frac{MW_{pep} + (MW_H) \cdot z}{z}$$

where MW_{pep} is the molecular weight of the peptide ion and MW_H is the molecular weight of Hydrogen.
(1) Remove all peaks with intensity less that 150 (Mass Spectrometer in Sequencing mode: lower limit)
(2) Sort and partition based on scan, mass to charge and intensity.
(3) Sort each partition and split if contiguous missing values.
(4) Delete multiple points with same scan in a partition.
(5) Repeat $2 - 4$ until no more partitions.
— End of Algorithm —

A base-line for which the current proposal is attempting to surpass is shown in Fig-1 with the first three isotopic peaks and their sum, for one of the largest lines that was constructed using the previous approach.

4. THE SOLUTION

A desirable solution would take the least amount of time to build, require the least amount of storage space at any point in time and produce more accurate results than that of the previously used algorithmic point-based solution.

One possibility could be to build a constrained (in the breadth not depth sense) Euclidean (or other metric) based minimum spanning tree (MST). This idea was envisioned during the investigation of the construction algorithm for α-shapes and the way in which α could be estimated via the use of a MST. The essential idea would be that of building a forest of MSTs such that each MST would represent a line.

Another possibility could be that of a data structure that uses persistence (i.e. an external memory data structure). This data structure would have the feature that when a point is added to a line, only the information necessary to retain the line description would

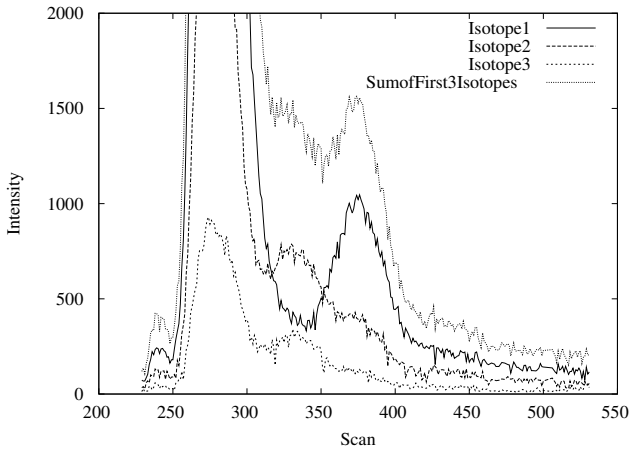

Figure 1: View of one representative spectrum for the first three isotopic peaks and their sum as extracted from the data (Fig-3) after processing by MassLynx Software using the deisotope algorithm. The spectrum has been truncated at $2,000$, but rises to approx. $11,000$.

be kept, and the point itself would be written to disk. This approach leads to a set of line summary descriptions being stored in memory with the points on disk.

At least three possible representations may be chosen for a datum: *i)* a classical point in Euclidean space, *ii)* a line spanning the measurement error of the instrument, and *iii)* a fuzzy number centered on the datum (triangular, gaussian, etc.). In particular, case *(iii)* subsumes case *(i)* in the event that a "spike" fuzzy number is defined, and subsumes case *(ii)* in the event that a uniform fuzzy number is defined. Therefore, case *(iii)* is the most general case, for which at least three representations may be investigated. However, due to time-constraints, only case *(i)* has been used in the empirical investigation.

The proposed algorithm is that based on the concept of a sweeping algorithm from Computational Geometry. The general idea is to consider one step at a time and then move to the next step... greedily constructing lines in an unsupervied fashion as the processing of points is performed. For the particular solution proposed, each step may consist of k scans (where a scan is those measurements obtained from the mass spectrometry instrumentation at time t_i) and these set of k scans are used in the construction of lines.

The concept of a line may be considered in more than one way. The simplest is to consider all points with exactly the same mass to charge ratio (one of the point co-ordinates) as belonging to a line. The next simplest approach would be to consider fixed small intervals, such that all points falling within the interval would be considered to belong to the line. Of course, during the construction of lines, a point may be introduced into the data structure that is related to the line in such a way that the point should not belong to that line. This should trigger the growth of a new line. Such line growing triggers are, for example, *i)* if the last inserted point's time t_l and the new point's time t_n are far apart in time ($t_n - t_l > \theta$) then a new line should be constructed. *ii)* if the intensity of the last point is very different than the new point, then a new line should be constructed, *iii)* if k new points near a position are within $k + \epsilon$ scans, for a fixed ϵ then all of the points should be considered to belong to the concept of a line. *iv)* etc.

If the k scan approach is taken such that $k > 1$ then one approach for the sweeping algorithm to take would be that of finding equivalence classes every j scans (e.g. the Numerical Recipes in C page 345 describes such an algorithm) performing a "FindLines" operation. What is an appropriate k value? This would need to be experimentally determined for a particular data set investigated. This paper does not address such an issue.

In summary, a sweep based algorithm that absorbs points into its data structure is proposed, with the properties that no supervision is provided and lines containing data points that are significantly older than the current sweeping position are stored on disk and removed from memory. In other words, the data structure self adjusts to the data at a particular time point such that line structures emerge from the data.

To be specific, the insertion of a point requires a hash into an array $\mathrm{O}(1)$ and then an append into a singly linked list $\mathrm{O}(1)$ along with checks that the currently stored line is sufficient to contain the new point (if it isn't, then that line is recorded to disk and a new line is constructed based on the single new point).

The 1-based hashing function ($h()$) used is based on *i)* a $width$ as heuristically determined to be related to the resolution of the mass spectrometer (e.g. a $1-1$ correspondance would yield $width = 0.1$, while a $1 - 4$ correspondance yields $width = 0.025$) and *ii)* the mass to charge ratios a and b, where a is the lowest currently known value based on the processed data up until that point and b is the point to hash.

$$h\left(a, b, width\right) = \left\lfloor \frac{b-a}{width} \right\rfloor + 1$$

The hashing function may be used to hash a point to its location, or in the case that a new point exceeds the extremes of the current data structure, to determine the total number of new line containers (bins) to which the simple data structure should be extended.

5. MASS SPECTROMETRY DATA

The central dogma of Biology revolves around the idea that DNA molecules give rise to RNA molecules which give rise to protein molecules through a very complex and currently not completely understood process. One aspect that is being tackled, is that of attempting to understand the differences in quantity of protein molecules between different cells or tissues of various organisms. This problem is important because it has been noted that a lot of the changes in proteomics data are very subtle, but may lead to large phenotypic differences.

Proteomics platforms that can efficiently identify and quantify changes in proteins related to disease (e.g., stroke) offer great promise for advancing biomedical research and the development of novel medicines.

Mass spectrometry is an analytical technique used to measure the mass-to-charge ratio (m/z) of ions. It is most generally used to find the composition of a physical sample by generating a mass spectrum representing the masses of sample components. The technique has several applications, including: *i)* identifying unknown compounds by the mass of the compound and/or fragments thereof, *ii)* determining the isotopic composition of one or more elements in a compound, *iii)* determining the structure of compounds by observing the fragmentation of the compound *iv)* quantifying the amount of a compound in a sample using carefully designed methods (mass spectrometry is not inherently quantitative), *v)* studying the fundamentals of gas phase ion chemistry (the chemistry of ions and neutrals in vacuum), *vi)* determining other physical, chemical or even biological properties of compounds with a variety of other approaches.

Two of the most commonly used methods for quantitative proteomics are *i)* two-dimensional electrophoresis (2DE) coupled

to either mass spectrometry (MS) or tandem mass spectrometry (MS/MS) and *ii)* liquid chromatography coupled to mass spectrometry (LC-MS).

In the 2DE-based approach, intact proteins are separated by 2DE, and the abundance of a protein is determined based on the stain intensity of the protein spot on the gel. The identity of the protein is now generally determined by MS analysis peptides after proteolysis of the protein spot. Since its inception in the mid-1970s, the 2DE-based approach has been routinely used for large scale quantitative proteomics analysis. The 2DE method, however, is limited in sensitivity and can be inefficient when analyzing hydrophobic proteins or those with very high or low mass. In addition, 2DE approach is difficult to automate and has a limited detection capacity for proteins with extreme ranges in pI values (the isoelectric point of proteins, which is the pH at which the net charge of the protein is zero), and for low abundance proteins.

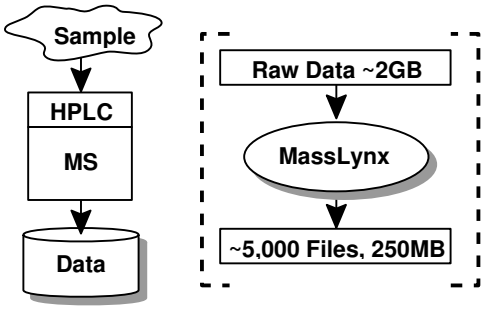

Figure 2: One biological sample is injected into the Mass Spectrometer (MS); first passing through the HPLC. 2GB of data are collected, which are then processed with MassLynx Software yielding the data set used as input for this study.

The LC-MS-based approach, on the other hand, can be automated and can identify proteins with extreme masses and pI values. This approach is also more sensitive and can detect very low abundant peptide peaks. However, to correctly quantify the low abundant peaks, they need to be properly resolved from the background "noise". The LC-MS/MS based approach often uses stable isotope labeling techniques, e.g. with 15N, 13C, stable isotope labeling by amino acids in cell culture (SILAC), and isotope-coded affinity tags (ICAT), to provide relative quantification. While potentially providing the greatest accuracy, isotopic labeling has some disadvantages. Labeling with stable isotopes is expensive, and some labeling procedures involve complex processes and yield artifacts.

A "label-free" LC-MS approach is based on the principle that the MS signal intensity of each peptide in a substantially similar sample analyzed under identical conditions is proportional to the abundance of the peptide within the dynamic range of the instrument. Therefore one may evaluate the relative abundance of a peptide in different, related samples by analyzing the samples under identical LC-MS conditions and by comparing MS signal intensity of the same peptide in different LC-MS runs. A disadvantage of such a label-free approach is that biological samples are usually very complex, and as a result, overlapping peptide peaks are often observed, which may be difficult to resolve. In order to accurately quantify peptide levels in LC/MS sample, not only do we need to identify and subtract the background noise but also need to deconvolve overlapping peaks.

Data was collected after one biological sample was injected into a mass spectrometer operating in survey mode (See Fig-2). MassLynx software (available from `http://www.waters.com`) was

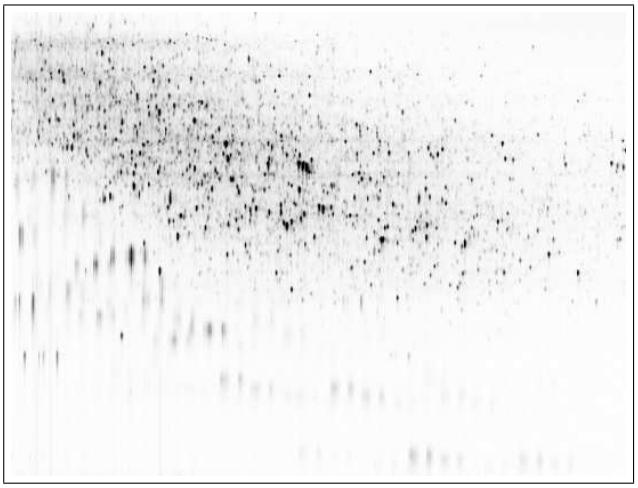

Figure 3: Visualization of data after processing by MassLynx Software for a set of eluting peptides from one biological sample. Time increases down the y-axis. Mass over charge increases along the x-axis. Intensity (pixel grey levels) represent ion counts.

used to generate peak lists for each of the MS survey scans (usually $2,000 - 4,000$ per sample). Each list contains three types of information: *i)* mass over charge, which is very accurate with an error of ± 0.05 Daltons (for peptides, the range is between about 400 and 1600 , *ii)* intensity (ion counts) from 0 to, for example, $11,000$ and *iii)* time, which can have a high error of ± 10 min. Fig-3 shows an example of raw mass spectrometry data for a set of eluting peptides from one sample.

6. EXPERIMENTAL SETTINGS

Experiments using the previously described mass spectrometry data were performed in order to attempt to justify the data structure selection. Table-1 shows the particular parameters for which a program was developed. The current implementation work has not been completed, but was capable of being used for performing the experiments and should be straightforward to extend and modify for other solutions that have been proposed.

Table 1: Experimental Settings for obtained results under the time constraints of the course submission deadline.

Start Scan	1
End Scan	-1 (determine dynamically)
Lowest M/z	-1 (determine dynamically)
Highest M/z	-1 (determine dynamically)
M/z Estimation using	scan 1
Resolution (r)	0.025 and 0.1
Sliding Window Size (k)	1 and 10
Datum Type (t)	Crisp
Distance (d)	Euclidean
Max Allowable Missing	7 points
Min Required Points	5 for 1 line
Contiguous Window Size (c)	882
Line Construction Policy (p)	$InsertPointsOnly$, $InsertAndWriteOldLines$

7. RESULTS

Table-2 has the locations of the disruptive events – those points that cause a resizing on the low or high side of the data structure. In this experiment, the first scan was used to obtain initial minimum and maximum mass/z estimates. Row 1 of the table lists the number of points (46) in scan 1 along with the minimum and maximum mass/z values (429.087311 and 1561.113525). This initial information was then used at a resolution of 0.025 to construct $45,282$ bins. The next row of Table-2 reports that scan 2 contains 69 points and has caused 843 new bins to be constructed on the low side of the data structure. During the course of all 2646 scans, there were 10 update minimums and 8 update maximums leading to a final bin interval of $[399.587311, 1601.312378]$ representing a data range of $[399.612213, 1601.311401]$. Analysing the bin interval closely, yields a span of 1201.725067; and with a resolution of 0.025 results in $\frac{1201.725067}{0.025} = 48069.00268$ bins. From a different perspective, summing all of the increases in the number of bins on the low side of the data structure ($= 1180$) across all 2646 scans with all of the increases in the bins on the high side of the data structure ($= 1606$) results in $45,282 + 1,180 + 1,606 = 48,069$ bins as we expected based on the previous calculation.

It can be observed that not all decreases (respectively increases) in the value of the data points will cause an increase in the data structure size at the top-most level (i.e. the array of pointers to lines). This occurs when a point falls within the bounds of the last "overhanging" interval.

Table 2: Disruptive events for particular scans over all 2646 scans. Resolution 0.025. Total number of data points: $5,480,201$ NC = No Change in number of bins.

Scan	Num Points	Minimum Data/Scan	+ Bins	Maximum Data/Scan	+ Bins
		∞		$-\infty$	
1	46	429.087311	45282	1561.113525	$*$
2	69	408.015808	843		
5	80	400.220703	312	1564.464478	134
6	85			1579.637451	607
7	72			1598.740356	764
16	73	399.993011	9		
41	47	399.912903	3		
43	66	399.904785	1		
82	57	399.677185	9		
86	874	399.661011	1		
92	146			1599.796021	42
108	1076			1601.046631	50
123	1063			1601.213867	7
126	1080			1601.285767	2
218	1388	399.620300	1		
536	1991	399.613403	NC		
556	2183			1601.290527	1
577	3367			1601.295288	NC
591	1563			1601.306641	NC
822	3184			1601.311401	NC
891	2618	399.612213	1		
2646	788				

7.1 Memory Usage

The $InsertPointsOnly$ line construction policy lead to a maximum memory usage of $20,920K$ while processing the first 200 scans and over $600MB$ while processing all $2,646$ scans. This indicates the necessity for an approach that leads to a reduction in run-time memory usage. The $InsertPointsAndWriteOldLines$ policy lead to an approximate maximum run-time memory usage of $25MB$ (down from over $600MB$) while processing all $2,646$ scans. This resulted in the unsupervised construction of $165,819$ lines. Further experiments in consultation with domain experts are required to evaluate which lines are appropriate for the investigated problem.

7.2 Line-based Algorithm Result Example

A portion of the implemented and examined line-constructing approach is shown in Fig-9. The constructed lines may not be properly representative of real-world molecules. For example, in Fig-9 it may be observed that some spurious lines exist (which might need to be deleted). However, it is interesting to observe that there is at least one example of a line that looks like a curve by the juxtaposition of line segments at their endpoints. A domain expert would need to be consulted in order to determine which lines may be joined/merged and which should be deleted.

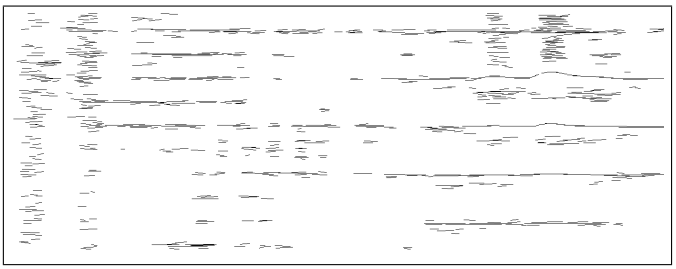

Figure 9: Selected subset of the constructed $165,819$ lines.

These results clearly indicate that a data structure that stores all points from the input data source is not the best solution. Therefore, the notion that only those points in the current set of lines should be stored within the data structure is put forth. In fact, a subset of the points within any particular line may be all that is required if the points are written "through" the data structure at an appropriate time. Such a data structure would have the property that the points will be stored in external memory (e.g. on disk).

8. CONCLUSION

The notion and use of a finite structure to potentially represent uncountable infinite data has been investigated through a brief survey of related work. A proposal of several possible data structure and algorithm variants has been made and three simple data structures were implemented using combinations of arrays and linked lists in order to collect and report experimental results. The use of hashing for O(1) access was made. Brute force search on small problem sizes was also noted to be of use. Counting based analysis (non-parametric) of the experiments was made. The conclusion that high bin counts corrolate to high intensities was not collaborated by the data due to the time-constrained deadline imposed by the course attended. A careful investigation of different point insertion approaches will need to be made with an examination of the possibility of post-processing the lines in order to merge and/or delete inappropriate line structures. The data structures proposed may potentially be parallelized or failing that, a distributed or grid computing environment could be considered. Further evaluation of the results with domain experts is required.

9. ACKNOWLEDGMENTS

The author would like to thank Professor Pat Morin from the University of Carleton for providing a productive learning environment. The author also appreciates Arsalan Haqqani and John Kelly from the National Research Council Canada, Institute for Biological Sciences (NRC-IBS) for providing the mass spectrometry data and guidance. Finally, the author certainly appreciates Julio Valdés, Bob Orchard and Fazel Famili from the NRC's Institute for Information Technology (NRC-IIT) for their continued support.

10. REFERENCES

[1] P. E. Black. "data structure", from Dictionary of Algorithms and Data Structures, NIST, 2006. [Online; accessed 8-April-2006].

[2] E. Borowski and J. Borwein. *Dictionary of Mathematics*. HarperCollins, 1989.

[3] T. M. Cover. Estimation by the nearest neighbor rule. In *IEEE Trans. Information Theory*, volume 14, pages 50–55, 1968.

[4] T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. In *IEEE Transactions on Information Theory*, number 13, pages 21–27, 1967.

[5] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, USA, 1973.

[6] J. Goodman and R. Pollack. *Handbook of Discrete and Computational Geometry*. CRC Press, 1997. P.K. Agarwal. Range searching. pages 575–598.

[7] K. Gupta and S. Ray. Fuzzy plane projective geometry. In *Fuzzy Sets and Systems*, volume 54, pages 191–206, 1993.

[8] A. Haqqani, A. Barton, M. Giguere, I. Rasquinha, F. Famili, W. Costain, D. Stanimirovic, and J. Kelly. Quantitative Proteomics on Consecutive LC-MS Runs of Unlabeled Peptides Using A New Software: A Comparison With ICAT. In *HUPO 4th Annual World Congress*, Munich, Germany, August 28 to September 1, 2005.

[9] A. Haqqani, M. Giguere, I. Rasquinha, A. Barton, F. Famili, W. Costain, D. Stanimirovic, and J. Kelly. Quantitative Proteomics Using LC-MS: ICAT vs. Label-Free Method. In *The 1st Annual Symposium on "Progress in Systems Biology"*, Ottawa, Ontario, Canada, November 17-18, 2005.

[10] M. Levine. Feature extraction: A survey. In *Proc. IEEE*, number 57, pages 1391–1407, August 1969.

[11] D. P. Mehta and S. Sahni, editors. *Handbook of Data Structures and Applications*. Number ISBN: 1584884355 in Computer and Information Science Series. Chapman & Hall/CRC, 2004.

[12] A. Rosenfeld. "geometric properties" of sets of lines. In *Pattern Recognition Letters*, volume 16, pages 549–556, May 1995.

[13] R. Tarjan. Amortized computational complexity. *SIAM J. Alg. Disc. Meth.*, 6(2):306–318, Apr 1985.

[14] Wikipedia. Mathematical structure — wikipedia, the free encyclopedia, 2005. [Online; accessed 8-April-2006].

[15] Wikipedia. Data — wikipedia, the free encyclopedia, 2006. [Online; accessed 8-April-2006].

[16] D. E. Willard. Log-logarithmic worst-case range queries are possible in space theta(n). In *Information Processing Letters*, number 17, pages 81–84, 1984.

[17] L. A. Zadeh. Fuzzy sets. In *Information and Control*, volume 8, pages 338–353, 1965.

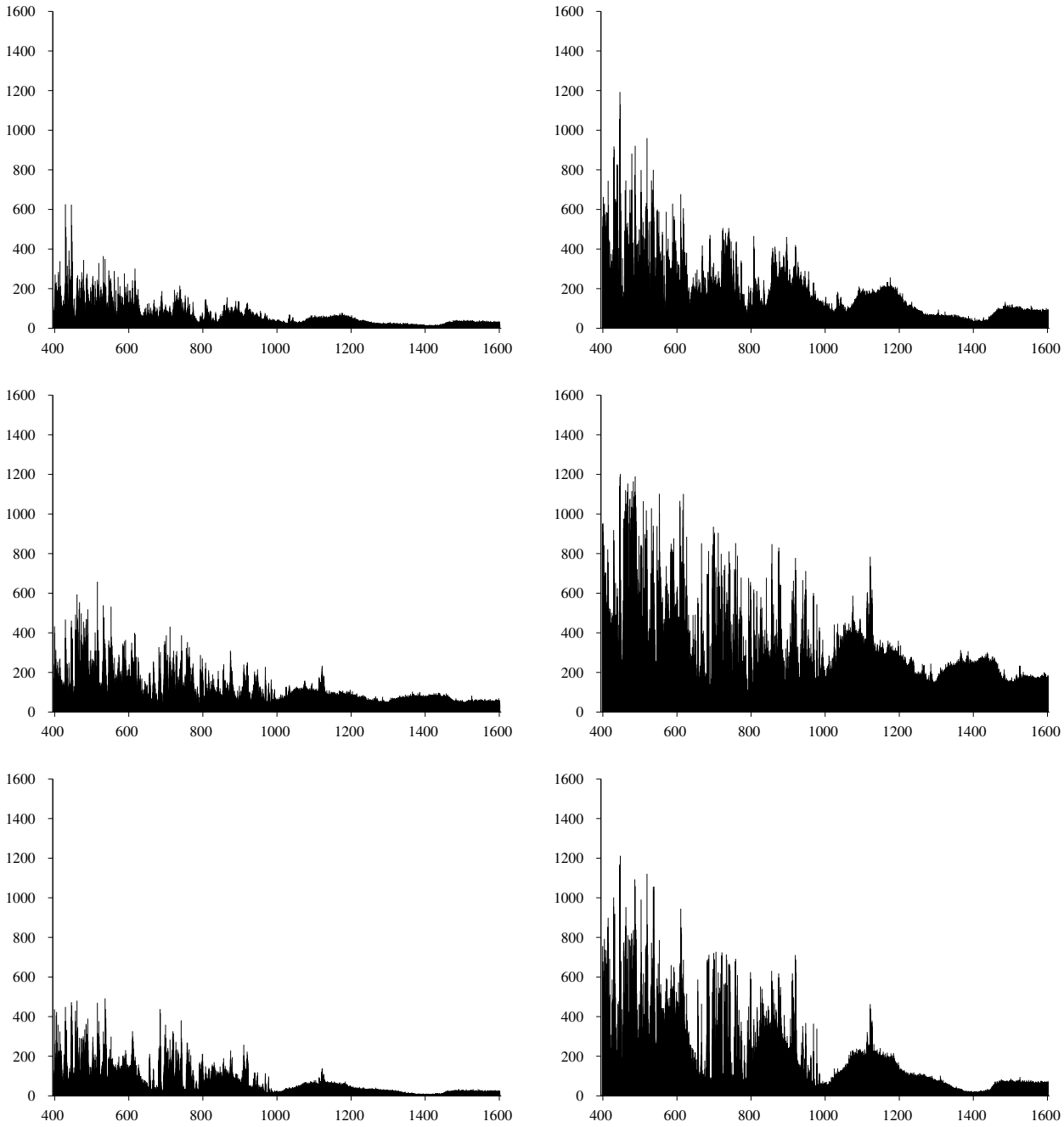

Figure 4: Effect of changing resolution on the data distribution for selected contiguous time (scan) periods $[a..b]$**. Left Column: Bin width** $= 0.025$ **Right column: Bin width** $= 0.1$ **Top Row:** $[1..882]$ **Middle Row:** $[883..1764]$ **Bottom Row:** $[1765..2646]$

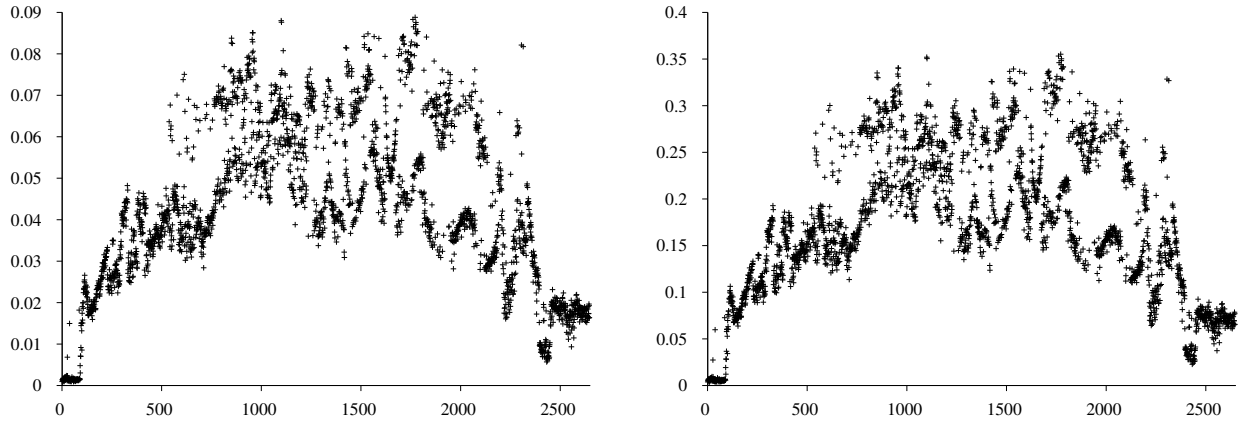

Figure 5: Effect of changinging resolution on average number of points per bin over all 2646 scans. Left: Resolution 0.025 Right: Resolution 0.1 For left: $48,069 \frac{bins}{scan} \cdot 2646$ scans $= 127,190,574$ bins, Divided by $5,480,201$ points $\approx 23 \frac{bins}{point} \approx 0.043 \frac{points}{bin}$

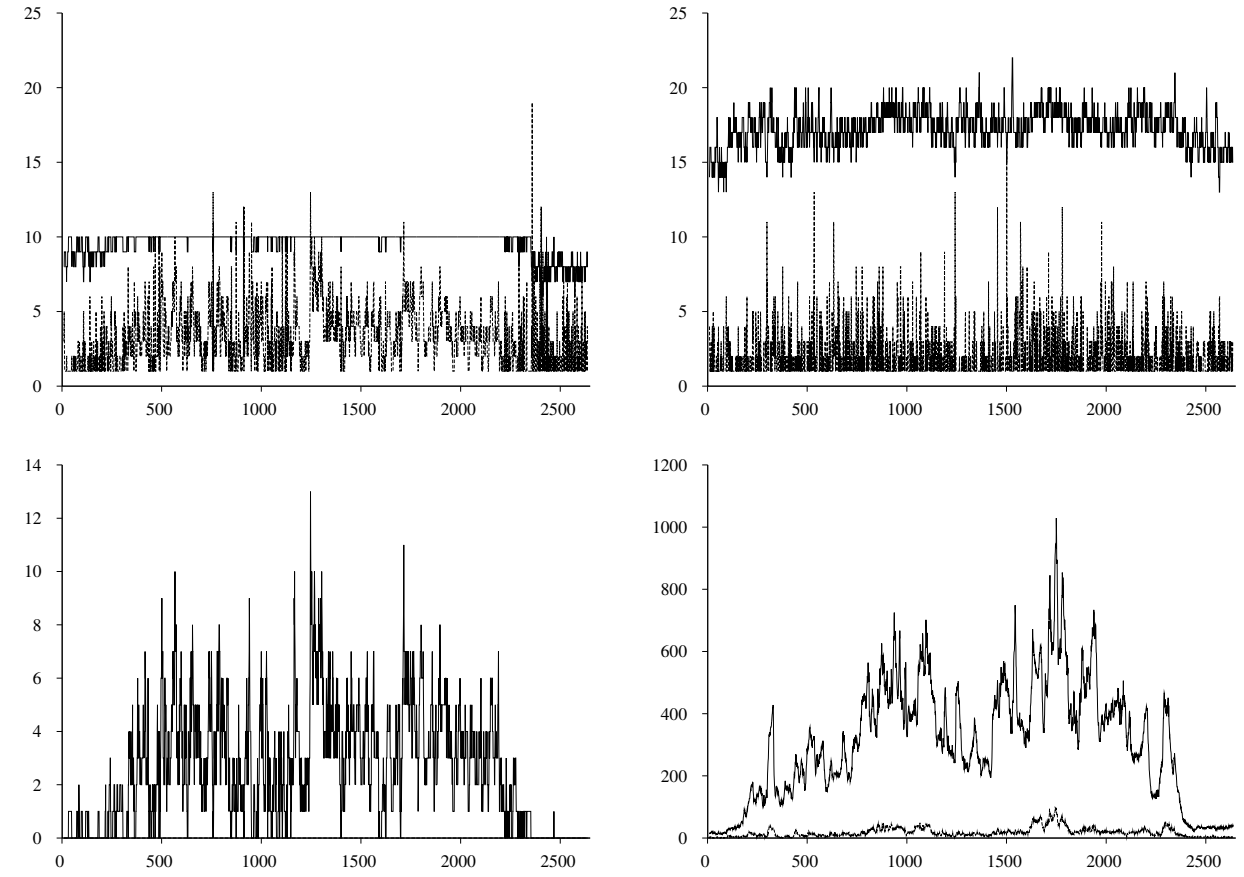

Figure 6: Effect of resolution on behaviour of largest bins for sliding windows of size $k = 10$. Left column: Resolution 0.025 Right column: Resolution 0.1 Let θ be the total number of points summed over the previous k scans for a particular bin. Top row: Each plot has 2 lines (upper and lower). Upper line: Value of maximum θ for each scan $> k$. Lower line: Number of bins having maximum θ points (upper line's size) for each scan $> k$. Bottom row: Each plot has 2 lines (upper and lower). Upper line: Number of bins having $\theta > k$ for all scans $> k$. Lower line: Number of bins having $\theta > 1.5 \cdot k$ for all scans $> k$.

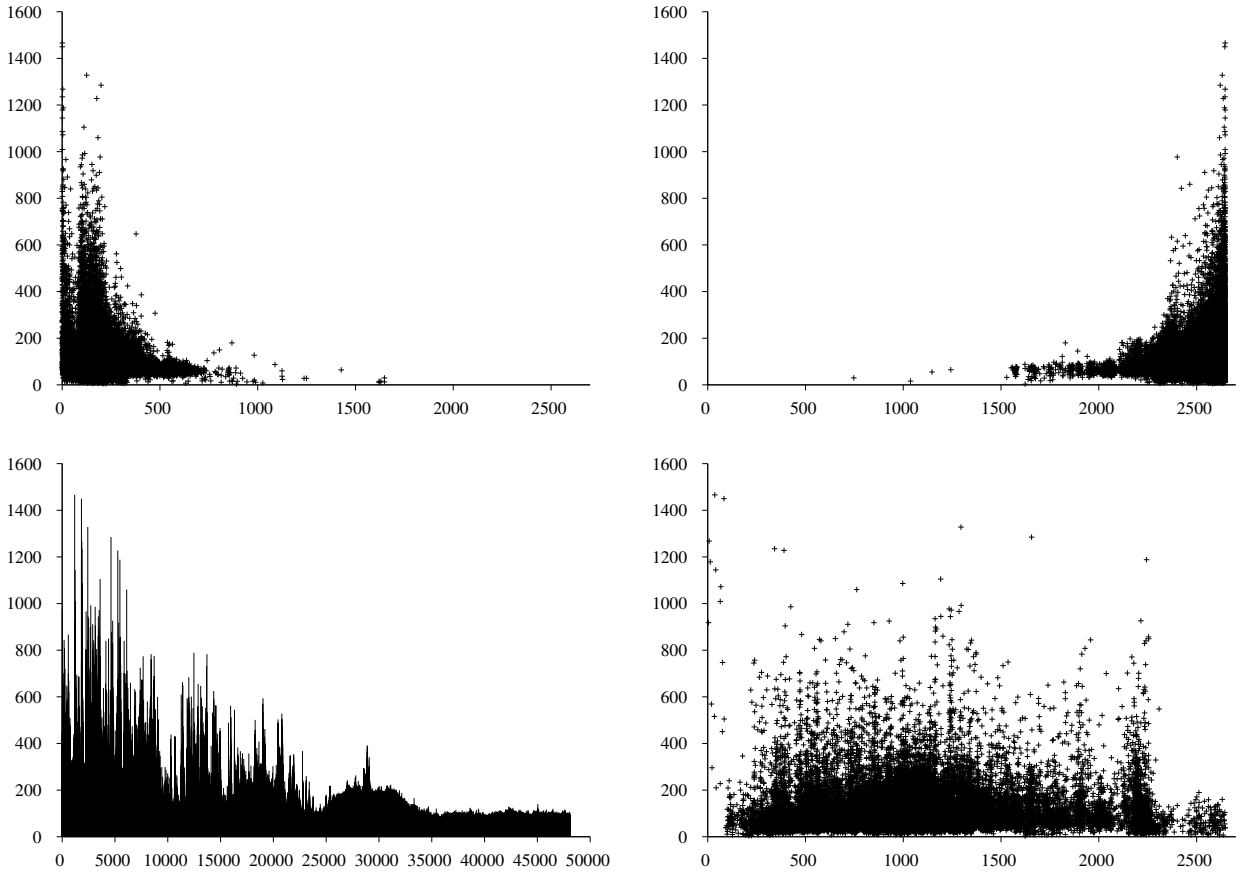

Figure 7: Investigation of properties when one line is used per bin (Resolution 0.025) throughout the 2646 scans (Points are not deleted after insertion in this scheme). Top Left: Time of first point insertion into the line data structure. Top Right: Time of last point insertion into the line data structure. Bottom Left: Number of points per line ($\equiv$ bin) for all $48,069$ bins (See Fig-4). Bottom Right: Time of point in line data structure having maximum intensity.

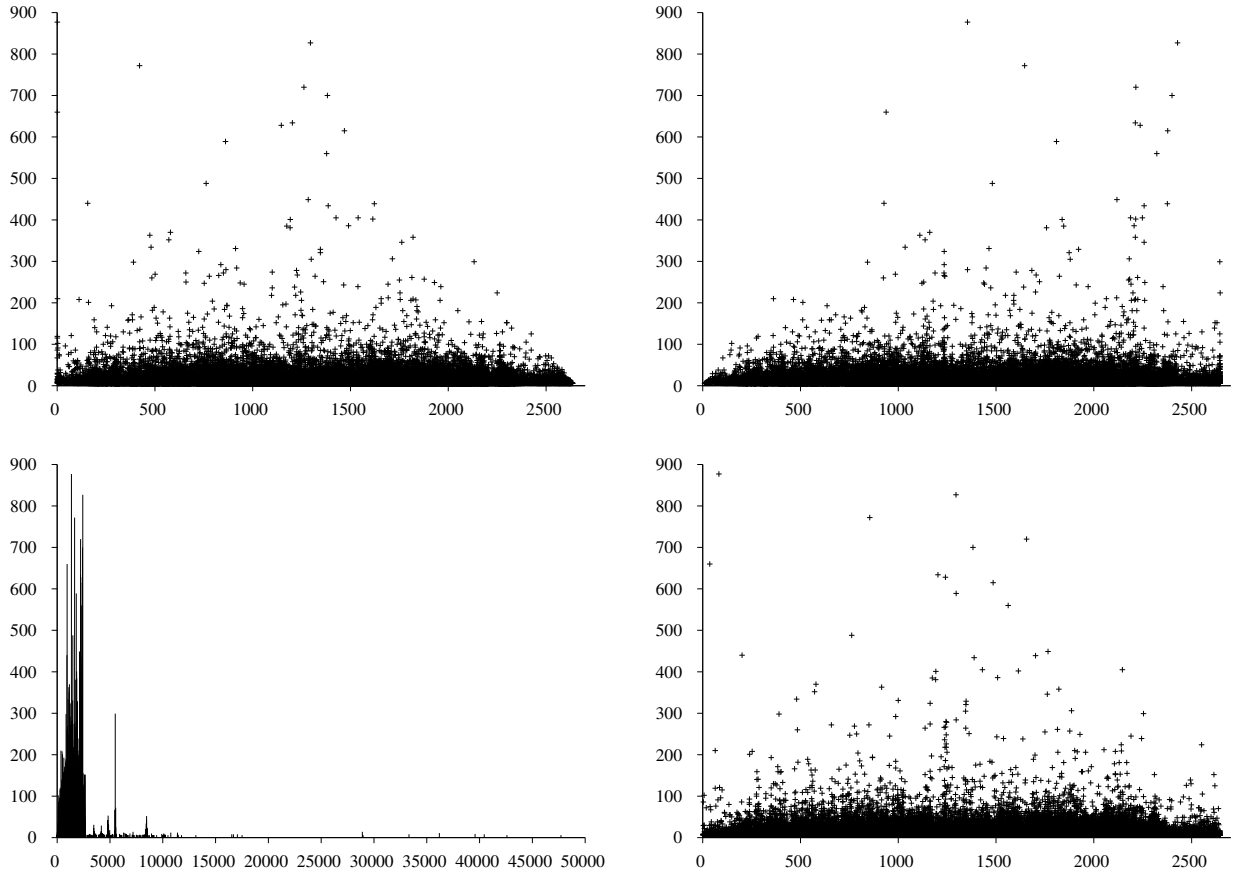

Figure 8: The policy was changed from $InsertPointsOnly$ in Fig-7 to $InsertAndWriteOldLines$. See Fig-7 for a description of the graphics. The essential difference lies in the fact that now the lines are observed to be distributed throughout the $2,646$ scans as one might hope when peptides are eluting.