AuDeNS: A Tool for Automatic De Novo Peptide Sequencing

TECHNICAL REPORT no. 383, ETH Zurich, Dept. of Computer Science

Sacha Baginsky[1], Mark Cieliebak[2], Wilhelm Gruissem[1], Torsten Kleffmann[1], Zsuzsanna Lipták[2], Matthias Müller[2], and Paolo Penna[2]*

[1] Swiss Federal Institute of Technology (ETH) Zurich
Institute of Plant Sciences
Universitätsstrasse 2, CH-8092 Zurich
``firstname.lastname''@ipw.biol.ethz.ch

[2] Swiss Federal Institute of Technology (ETH) Zurich
Institute of Theoretical Computer Science
Clausiusstrasse 49, CH-8092 Zurich
{cielieba,zsuzsa,muellerm,penna}@inf.ethz.ch

Abstract. We have developed and implemented a framework for de novo sequencing of peptides using tandem mass spectrometry data. It first cleans the input spectrum with a number of data cleaning algorithms ("grass mowers"), followed by a sequencing algorithm that is a modification of a dynamic programming algorithm introduced in [CKT00]. In first experiments, our prototype performs well (but not better) in comparison with Sequest, a frequently used software for peptide identification with database–lookup, and Lutefisk, a de novo peptide identification tool. In this paper, we present first results in the development of an efficient de novo sequencing tool.

Keywords: de novo sequencing, dynamic programming algorithm, proteomics, grass mowing

1 Introduction

Automatic interpretation of mass spectrometry data is becoming increasingly important in high–throughput protein identification.

In *tandem mass spectrometry* (MS/MS) [HYS86], data are obtained in two steps. In the first step, a *mass fingerprint* of the protein is generated: The protein is cleaved into peptides, using a sequence–specific proteinase (e.g. trypsin, cleaving on the C–terminal side of arginine or lysine). Then the masses of the peptides are determined by mass spectrometry. In a

[*] Authors appear in alphabetical order.

second step, some of the peptides are selected (e.g. those with the highest intensities) and dissociated into fragment ions (CID: collision induced dissociation). There are two types of ions: b–ions, which have the N–terminus of the peptide, and y–ions, which have the C–terminus ([RF84]). b–ions correspond to prefixes of the amino acid sequence of the peptide, and y–ions to its suffixes. The *tandem mass spectrum* of a peptide consists of a list of molecular masses measured during the experiment, along with their abundance values (m/z). Ideally, a spectrum contains exactly all b–ions and y–ions of the peptide. However, due to contamination, measurement inaccuracies and other problems, a typical real–world spectrum is a long list of pairs of masses and abundance values, much of which is noise, and only few of which are actually derived from the original peptide.

In peptide identification using mass spectrometry, the problem is to determine the peptide's primary structure, i.e., its sequence of amino acids, given a tandem mass spectrum.

One prominent approach for protein identification is to identifiy peptides using a protein database: Given a tandem mass spectrum of a peptide, at first peptides from the database with matching parent masses (the total mass of the peptide) are suggested. Then the given spectrum is compared to each of the theoretical spectra of the database peptides, and the peptides are ranked according to how well their spectra match to the given spectrum. Software tools which implement this approach such as Sequest ([Sequest:www]) show that this method is very successful in identifying proteins listed in the database. However, its shortcoming is its dependence on a database and on the correctness of its entries. Even though Sequest allows for inclusion of some post–translational modifications, unsequenced proteins and proteins that are the products of alternative splicing processes are not considered. In these cases, techniques for interpreting the tandem mass spectrum without using information from a database are needed. This is referred to as *de novo sequencing*. In general, de novo sequencing consists of two steps: first, a set of theoretical peptides which match the given tandem mass spectrum is generated. This set can be very large due to contamination and measurement errors. In a second step, these theoretical peptides are ranked using heuristics, and those peptides with the highest ranking are output.

Several (more or less efficient) algorithms have been proposed to generate the set of matching peptides, e.g., Fernandez-de-Cossio et al. [FGS97], Taylor and Johnson [TJ01], and Dančík et al. [DAC99] transform the spectrum to a graph in which every connected path represents a possible sequence. They use different algorithms to select good matching sequences among the very large number of possible paths. There are several software packages that implement de novo sequencing algorithms, such as Lutefisk ([TJ97],[TJ01],[Lutefisk:www]), and others ([BioAnalyst:www], [biotools:www] [BioWorks:www], [MSfast:www]).

Recently, Chen et al. introduced a de novo sequencing algorithm that uses dynamic programming ([CKT00]). The algorithm has two variants, one for clean data, and one for noisy data. Since clean data do not exist in biological experiments, only the noisy variant is applicable. Chen et al. proved that the algorithm for noisy data has running time at most cubic in the number of peaks of the given spectrum. However, they did not provide an implementation of their algorithms. Naïve use of the noisy variant is computationally too complex, since the number of potential solutions is too large. In addition, measurement errors need to be taken into account.

We have developed heuristics ("grass mowers") for assigning relevance values to the input peaks, and have implemented a framework, AuDeNS, that first uses the grass mowers to preprocess the spectrum, and then employs a modification of the noisy sequencing algorithm of [CKT00] that can handle measurement errors. Also, we have solved the problem of the potentially exponential number of solutions by assigning relevances to the solutions and only enumerating those within a user–specified threshold relative to the maximal relevance value. The output of AuDeNS is a ranked list of "multi–sequences" (sequences that take inherent ambiguities of the input into account).

Even though our tool does not sequence as well as Lutefisk at the moment, we believe that it can be developed to match or even outperform Lutefisk for a number of reasons:

1. In our experiments, AuDeNS has much lower running times than either Lutefisk or Sequest, due to a fast algorithm and efficient implementation.
2. AuDeNS is a framework that is capable of having new mowers added to it with minimal effort. The mowers we employ at the moment are heuristics that are plausible but need further fine–tuning, esp. with regard to the parameters.
3. Even without having algorithmically tuned the parameters of AuDeNS, our output compares relatively well with that of both Lutefisk and Sequest.

This paper constitutes a first report on work in progress.

1.1 Overview

The paper is organized as follows: We first give a formal problem definition in Section 2. In Section 3, we describe our program in detail, first explaining the grass mowers (3.1), then the sequencing algorithm (3.2), followed by details of our implementation (3.3) and future work on AuDeNS (3.4). Finally, in Section 4, we present first experimental results.

2 Problem Definition

A *peptide* is a string over the 20–letter alphabet of *amino acids*, where each amino acid A is assigned a non–negative molecular mass $m(A)$ (measured in Daltons (Da)). Typical length of peptides in our setting is between 10 and 20 amino acids. Given a peptide $\mathcal{P} = A_1 \dots A_k$, its *dissociation pattern* is the set $D_{\mathcal{P}} = \{m_{\text{parent},\mathcal{P}}, m_{b,1}, \dots, m_{b,k-1}, m_{y,1}, \dots, m_{y,k-1}\}$, where $m_{\text{parent}} := \sum_{i=1}^{k} m(A_i) + \text{offset}_{\text{parent}}$ is the mass of the *parent ion* (the entire peptide), $m_{b,r} := \sum_{i=1}^{r} m(A_i) + \text{offset}_b, 0 < r < k$, is the mass of the *b–ions* (its prefixes), and $m_{y,r} := \sum_{i=r}^{k} m(A_i) + \text{offset}_y, 0 < r < k$, is the mass of its *y–ions* (its suffixes). Hereby, $\text{offset}_{\text{parent}}, \text{offset}_b, \text{offset}_y$ are positive real numbers[1].

A *tandem mass spectrum* S as found in a `.dta`–file[2] contains the *parent mass* $m_{\text{parent},S}$, followed by a list of pairs $(m(i), a(i))$, $i = 1, \dots, n$, where the $m(i)$ are molecular masses, and $a(i)$ is the abundance of $m(i)$. The entries are ordered w.r.t. their m–values. Typical values for n are between 35 and 900. A pair $(m(i), a(i))$ is often referred to as a *peak*, which derives from the customary visualization of mass spectra (see Figure 1). For the same reason, entries with large, resp. small abundance values are called high resp. small peaks. We will refer to peak $(m(i), a(i))$ simply by i. In the following, we will call peaks that derive from ions of the original peptide *real peaks*, and the others *noise* or *grass*.[3] In addition, we are given a *mass tolerance* ϵ, the measurement error of the mass spectrometer.[4]

A *solution* to a given spectrum S is a peptide $\mathcal{P}$ s.t. $|m_{\text{parent},\mathcal{P}} - m_{\text{parent},S}| \leq \epsilon$. In addition, we would like to match the masses of $D_{\mathcal{P}}$ with

[1] In fact, $\text{offset}_{\text{parent}} = \text{offset}_y = 19$ Da, and $\text{offset}_b = 1$ Da.

[2] An ASCII–formatted file as output by Micromass Qtof and Thermofinnigan ion trap programs.

[3] Because of their appearance in the visualization, groups of small peaks are sometimes referred to as *grass*. Since much of this part of the input is not well interpretable, some of the data preprocessing is concerned with getting rid of this grass. This is the reason we call our data cleaning algorithms *grass mowers*.

[4] For our data, we use $\epsilon = 0.5$ Da.

the real peaks of the spectrum, i.e., find pairs $(m, m(i))$, $m \in D_{\mathcal{P}}$ and real peak i of S, for which $|m - m(i)| \leq \epsilon$ holds. However, since it is not clear from the outset which peaks of S are real peaks, we allow that peaks not be matched, and in the extreme, that no peaks match with $D_{\mathcal{P}}$. Thus, since the only necessary condition for a solution is that the parent masses match within the given mass tolerance, the solution cannot be unique. In particular, given one solution of length k, all of its permutations will match, typically a number exponential in k. Another reason for non–uniqueness of solutions is that the two amino acids isoleucine (I) and leucine (L) have exactly the same molecular mass. Increasing the mass tolerance causes further pairs of amino acids to become indistinguishable. Missing real peaks in the spectrum account further for non–uniqueness of the solution, e.g., $m(\mathrm{N}) = 2 \cdot m(\mathrm{G})$.

The aim, therefore, is to output a ranked list of solutions such that the peptide that gave rise to the spectrum has high ranking. A *multi–sequence* is a finite set of sequences that we write as a regular expression, e.g., V(N|GG)GYSE(I|L)ER is short for the set {VNGYSEIER, VNGYSELER, VGGGYSEIER, VGGGYSELER}. Rather than listing feasible solutions individually, AuDeNS outputs a ranked list of multi–sequences.

3 AuDeNS: A Tool for Automatic de Novo Peptide Sequencing

AuDeNS works in the following way: In a first step, it applies the mowers to the input data, assigning to each input peak i a relevance value $r(i)$, with the default being $r(i) = 1$. Hereby, each mower M uses a relevance factor Rel_M (which can be set as a parameter of AuDeNS), and the relevance value of peak i is then given by $r(i) := 1 + \sum_{M \text{ mower}} \mathrm{Rel}_M \cdot M(i)$, where $M(i)$ is the value assigned to peak i by mower M. The relevance of a solution is then the sum of the relevances of the peaks matched by this solution. All mowers output values between 0 and 1, and thus, their output can be weighted against each other by the relevance factors specified by the user. In addition, the mowers each have parameters that can be specified (see Section 3.1 for details). It is an important aspect of AuDeNS that new mowers can be integrated with minimal effort.

In a second step, AuDeNS applies the sequencing algorithm. Hereby, the minimal quality of the solutions can be specified as a relative value as measured in comparison to the relevance $r_{\max}$ of a best solution, i.e., a $0 \leq \delta \leq 1$ s.t. all solutions with relevance greater or equal to the threshold

$r_\delta = (1 - \delta) \cdot r_{\max}$ are to be computed. First, a table is built up and $r_{\max}$ is computed. Then, all solutions with relevance greater or equal r_δ are computed and output, using backtracking in the table.

Global parameters such as mass tolerance and relevance factor of the mowers allow for a fine–tuning of AuDeNS.

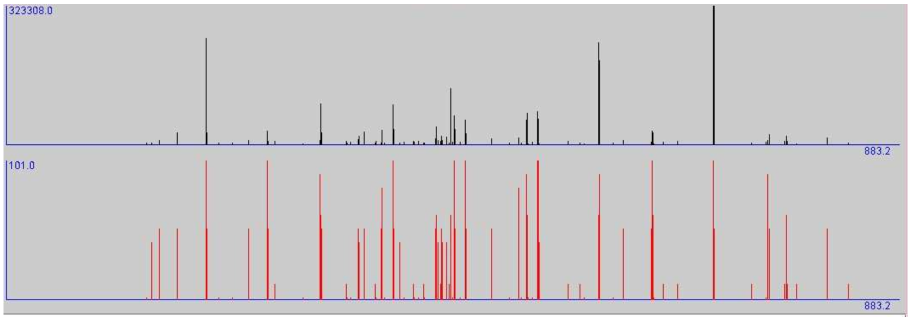

Fig. 1. A tandem mass spectrum with corresponding relevance values. The x–axis represents the masses. The upper graph shows the abundance values of the masses on the y–axis, and the lower graph their relevance values.

3.1 The Mowers

Threshold Mower Peaks with very small abundance values, e.g., under 10000, are unlikely to be real peaks. The threshold mower marks all peaks with an abundance value above a given (low) threshold.

Window Mower The window mower has two parameters: the size of a window W and a number k of peaks per window. It moves along the input, and, for each peak i, marks the k peaks with highest abundance within the window starting at $m(i)$, i.e., those k peaks with highest abundance in the set $\{j \mid m(i) \le m(j) \le m(i) + W\}$. The mower assigns each peak i a value proportional to the number of times it has been marked.

Roughly speaking, high peaks are more likely to be caused by peptide ions than low peaks. The rationale for the window mower then is twofold: First, within any window of the approximate size of the smallest amino acid mass, there can be at most two real peaks, namely one b–ion and one y–ion.

Second, when sequencing manually, contiguous regions of m/z values can be found such that the abundance of the peaks within *one* region do not differ very much, while they do differ between *different* regions. Regions are then scaled with different factors in order to level the height of the peaks, and then peaks which are high within their region are considered for the sequencing process ([SJ01]).

The reasons are inherent technical characteristics of an ion trap that result in differential efficiencies of mass measurements over the entire mass range. During an MS/MS cycle, a selected peptide is excited by resonance excitation to accomplish collision induced dissociation. However, resonance excitation and resonance ejection are virtually identical, resulting in the possible loss or inefficient measurement of product ions during resonance excitation. As a consequence, low molecular mass ions are often underrepresented in an MS/MS spectrum (e.g., product ions with less than 30% of the parent mass are only rarely observed, [A01]). Another reason for "regional differences" in mass measurement efficiencies are the inherent biochemical characteristics of amino acids, either resulting in efficient (e.g. Proline) or inefficient (e.g. Glycine) dissociation of a peptide bond ([SJ01]).

Therefore, considering only the absolute abundance of the peaks is not sufficient to identify real peaks, but a method that takes the differences between regions into account is more appropriate, such as employed by the window mower.

Isotope Mower Typically, single ions give rise to more then one peak in an MS/MS spectrum due to isotopes. Isotopes differ in the number of neutrons they have in the nucleus, and they occur in nature with different probabilities (e.g., carbon has either 6 neutrons, with probability 98.892%, or 7 neutrons, with probability 1.108%). Thus, peaks without corresponding isotope peaks are rather unlikely to be caused by ions, and the number of isotope peaks of a single peak can be used to adjust the relevance of a peak.

The isotope mower has a parameter k, the number of isotopes required. It assigns a value to each peak i proportionate to how many isotopes are present in the spectrum, i.e., for each $j, 1 \leq j \leq k$, it checks whether there is a peak with mass m such that $|m - (m(i) + j)| \leq \epsilon$.

Intersection Mower The intersection mower considers all spectra obtained from the same experiment as the current spectrum S that have the same parent mass $m \in [m_{\text{parent},S} - \epsilon, m_{\text{parent},S} + \epsilon]$. It then assigns each peak i in S a value proportionate to the number of other spectra in which it is also contained.

The rationale is the consideration that other spectra with the same parent mass that stem from the same experimental setup are likely to have been derived from the same peptide. Even though in theory, many different peptides will have the same parent mass (e.g., all permutations of the same amino acids), in reality—as some preliminary database analysis has shown—it is not very likely[5] that different peptides stemming from the same protein or from the same small number of proteins have the same mass.

Complement Mower If i is a peak in the spectrum which arose from a b–ion, then we expect the corresponding y–ion to be present in the spectrum, and vice versa. Therefore, for any peak i in the spectrum, we increase the relevance of i if the complement peak i' with $m(i') = m_{\text{parent}} - \text{offset}_{\text{parent}} + \text{offset}_b + \text{offset}_y - m(i)$ (within ϵ) is present. This mower is very closely related to the sequencing algorithm we use, since the algorithm heavily relies on pairs of complement peaks.

3.2 The Sequencing Algorithm (Weighted-Chen-et-al-Algorithm)

Our sequencing algorithm is based on the dynamic programming algorithm for noisy data introduced in [CKT00]. The algorithm in [CKT00] maximizes the sum of weights of peak pairs (edges), while our algorithm maximizes the sum of the relevance values assigned to the peaks. We refer to this algorithm as Weighted-Chen-et-al-Algorithm.

The idea of the Weighted-Chen-et-al-Algorithm is to generate a directed vertex–labelled graph $G = (V, E)$ with two special vertices x_0 and y_0, such that any directed path from x_0 to y_0 satisfying an additional constraint will correspond to a solution. For each peak i, there are two vertices $x_i, y_i \in V$, whose masses are the smaller and the larger value, respectively, of the mass of peak i and its complement w.r.t. the parent mass. The relevance $r(v)$ of a vertex v is the relevance of the correspondig peak assigned by the mowers. The reason for generating pairs (x_i, y_i) of vertices is that if a peak is real, then it is either a prefix (a b–ion) or a suffix (a y–ion)—and if the spectrum were perfect, then it would also contain its complement (see Section 3.1).

If two vertices have the same mass within the mass tolerance ϵ, then we merge them, and assign the new vertex the maximal relevance value among the merged vertices. The vertices are sorted such that $m(x_0) <$

[5] only between 2 and 7%

$m(x_1) < \ldots < m(x_n) < m(y_n) < \ldots < m(y_1) < m(y_0)$[6]. Hereby, x_0 and y_0 are two new vertices with masses $m(x_0) = \text{offset}_b$ and $m(y_0) = m_{\text{parent}} - \text{offset}_{\text{parent}} + \text{offset}_b$, and both relevance 1. At this point, for each pair $(x_i, y_i), i = 1, \ldots, n$, we know that it either constitutes noise, or one is a prefix of the peptide and the other a suffix—but we do not know which is which.

G contains a directed edge (u, v) if $m(v) - m(u)$ can be written as the sum of some amino acid masses within the mass tolerance (see Section 3.3 for details). Call a directed path P in G *k–compatible* if P contains at most one vertex of each pair $(x_i, y_i), i = 1, \ldots, k$. Any n–compatible directed path P in G from x_0 to y_0 corresponds to a solution to the input, because it represents a partial list of prefixes.

We will now fill in a table Q of size $(n + 1) \times (n + 1)$ that will be used to compute paths from x_0 to y_0. Define $w(P)$, the *pathweight* of the directed path P in G, as $w(P) := \sum_{v \in P} r(v)$. Set

$$\begin{aligned} Q(i,j) := \max\{w(L) + w(R) \mid\ & L \text{ directed path from } x_0 \text{ to } x_i, \\ & R \text{ directed path from } y_j \text{ to } y_0, \\ & \text{and } L \cup R \text{ is } \max(i,j)\text{–compatible.}\} \end{aligned}$$

The table Q has the property that $Q(i, j) > 0$ if and only if there is a path L from x_0 to x_i and a path R from y_j to y_0 s.t. $L \cup R$ is $\max(i, j)$–compatible. It can be filled in using the crucial observation that the maximum path for a given pair $x_i, y_j, i < j$, can be computed using all maximal paths for pairs x_i, y_k, for $k < j$. Since $j > i$, y_j can be added to any such pair $L \cup R$ without violating the compatibility–condition. The situation is analogous for the case where $i > j$. Thus, $Q(i, j)$ can be computed as follows:

$$Q(i,j) = \begin{cases} \max_{0 \le k \le j}\{Q(i,k) \mid (y_j, y_k) \in E, Q(i,k) > 0\} + r(y_j) & \text{if } i < j \\ 0 & \text{if } i = j \\ \max_{0 \le k \le i}\{Q(k,j) \mid (x_k, x_i) \in E, Q(k,j) > 0\} + r(x_i) & \text{if } i > j \end{cases}$$

The value of a maximal path is now $r_{\max} = \max\{Q(i, j) \mid (x_i, y_j) \in E\}$. Note that $r_{\max} = 0$ means that there is no feasible solution to the input, i.e., the parent mass cannot be written as a sum of amino acid masses within the given error tolerance ϵ.

Now all paths within the given threshold are enumerated recursively via backtracking in the table.

[6] Because of the merging of vertices, the new value of n may have decreased, but we ignore this detail here.

3.3 Details of Efficient Implementation

Enumerating the Multi–Sequences Entry $Q(i, j)$ contains the maximum weight of any path from x_0 to x_i and from y_j to y_0. Thus, the table Q can be used in a backtracking algorithm to recursively enumerate all paths from x_0 to y_0 whose weights are above a given threshold. The use of a threshold allows for pruning the tree of computation generated by the backtracking process in early stages. This makes the time spent in the recursion proportional, not to the total number of possible paths, but to the number of paths that are of interest (whose weights are above the threshold).

Deciding Whether a Mass is a Sum of Amino Acids To decide whether a given mass can be represented by a sum of masses of certain amino acids and to list all such amino acid sequences, we work with an array of Boolean variables $b_0, \ldots, b_N$. Variable b_i represents masses $m \in [i\Delta m, (i+1)\Delta m)$. Let $m_i = i\Delta m + \Delta m/2$ be the center mass of the interval represented by b_i. The maximal index N depends on the maximal mass $M_{\max}$ considered and is computed as $N = \lceil M_{\max}/\Delta m \rceil$. We use $\Delta m = 0.01$ Da and $M_{\max} = 1000$ Da.

The variables b_i are initialized as follows: If the mass interval represented by b_i contains the mass of any single amino acid, b_i is set true, otherwise b_i is set to false. This can be done in $20 + N = O(N)$ time. In a second phase, we run from b_0 to b_N and set b_i true, if there is an amino acid mass a such that the variable b_j containing $m_i - a$ is true. The second pass takes $20N = O(N)$ time steps since there are 20 amino acids.

To answer the question whether a mass sum m measured with error ϵ can be represented by a sum of masses of certain amino acids, we check all variables b_i that represent part of the interval $(m - \epsilon, m + \epsilon)$. If one of them is true, the answer is yes, if all are false, then the answer is no.

Enumeration of Subsequences To enumerate all amino acid sequences for a mass sum m and an error ϵ, we proceed as follows: For all true b_i's that represent part of the interval $(m - \epsilon, m + \epsilon)$, and for all amino acid masses a, we test whether the variable b_j containing $m_i - a$ is true. If so, we store the letter of amino acid a and recursively enumerate all sequences for mass m_j. This algorithm, however, enumerates all permutations of all possible sequences. To avoid this, in recursion depth d we only consider amino acids whose letters are lexicographically larger or equal to the animo acid letter chosen in depth $d-1$. This way, only distinct sequences with respect to permutation are output.

3.4 Future Work

Parameters At the moment, parameters of the mowers (such as thresholds or number of isotopes) as well as their relevance factors are set to hand selected values. Tuning these parameters algorithmically will be a major part of our future work. We will run AuDeNS on input spectra where the correct sequence is known in advance, either from the experimental setup (using known peptides), or from the output of other sequencing programs (such as Sequest or Lutefisk), and use machine learning algorithms to adapt the parameters of our tool.

Artificial noise We are working on a model for "artificial noise" in tandem mass spectra. This model will allow to generate an artificial spectrum from a given amino acid sequence, and to introduce noise of different types in different amounts (such as measurement errors or addition and ommision of peaks). Once the model of artificial noise maps the "real world noise" in an adequate way, we will use it to investigate the influence of different types of noise and their combination on the quality of the result of AuDeNS.

Mowers We will investigate and implement new types of mowers, e.g., an offset mower, which is similar to the intersection mower, except that it allows for an offset between the peak matches. This offset mower will enable us to include data from experiments such as methyl ester derivatisation, where each y–ion is shifted by 14 Da.

In addition, existing mowers will be improved, e.g., for the window mower, the fixed size window might be replaced by windows of flexible sizes, which will allow to better identify regions of peaks with almost identical height.

Ranking At the moment, ranking of the result multi–sequences in AuDeNS is only based on the mowers' relevance values. Postprocessing of the result will allow for even more appropriate rankings. E.g., statistical information from protein databases (such as the distribution of all tuples of amino acids) can be used to indicate how likely a certain result multi–sequence is.

4 First Experimental Results

The running time of AuDeNS depends on the number of sequences computed. To create the best 30 sequences, AuDeNS takes less than one second to read, mow, and sequence a spectrum.

We compared the output of AuDeNS to the results of Sequest and Lutefisk. Lutefisk needs 2 seconds up to several minutes on the same computer and same spectrum to output the best 0 to 5 solutions. However, Lutefisk outputs individual peptide sequences, as opposed to multi–sequences of AuDeNS. Even without algorithmically tuned parameters of the mowers, the best sequence found by Sequest for a spectrum is very often among the first 30 sequences created by AuDeNS. Otherwise, there are many almost correct sequences among the output. Three selected example outputs are shown in Figures 2 to 4. The parameters had been set as follows:

threshold mower:	relevance	40	threshold 8000
window mower:	relevance	10	no. of peaks 2, window 50
isotope mower:	relevance	10	no. of isotopes 1
complement mower:	relevance	40	
intersection mower:	relevance	0	

```
740.0   V(N|GG)GYSE(I|L)E(R|GV)
735.0   V(N|GG)GY(I|L)C(I|L)E(R|GV)
716.0   V(N|GG)GYAGS(I|L)E(R|GV)
715.0   V(N|GG)GYES(I|L)E(R|GV)
715.0   V(N|GG)GYDT(I|L)E(R|GV)
715.0   V(N|GG)GYTPME(R|GV)
711.0   V(N|GG)GYSGA(N|GG)E(R|GV)
705.0   V(N|GG)GYTD(I|L)E(R|GV)
```

Fig. 2. 2894.dta: Sequest sequence VNGYSEIER has the highest rating in the AuDeNS output.

5 Acknowledgements

We would like to thank Luzi Anderegg, Birgitta Weber, and Peter Widmayer for many very helpful discussions.

```
655.0   (AG|Q|K)A(I|L|N|GG)AAA(I|L)(N|GG)(AG|Q|K)
655.0   (AG|Q|K)(N|GG)AAAA(I|L)(N|GG)(AG|Q|K)
644.0   (AE|IS|LS|TV|CP)(I|L)AAA(I|L)(N|GG)(AG|Q|K)
615.0   (AG|Q|K)A(I|L|N|GG)AAA(I|L|N|GG)(I|L)(AG|Q|K)
615.0   (AG|Q|K)(N|GG)AAAA(I|L|N|GG)(I|L)(AG|Q|K)
605.0   (AG|Q|K)PSAAA(I|L)(N|GG)(AG|Q|K)
605.0   (AG|Q|K)(N|GG|D)AAAA(I|L)(N|GG)(AG|Q|K)
```

Fig. 3. 3165.dta: Sequest sequence AEIAAALNK is at third position in the AuDeNS output.

```
496.0   (AG|Q|K)AE(N|GG)(AG|Q|K)SGFFE
495.0   (AG|Q|K)AE(N|GG)AAEFFE
495.0   (AG|Q|K)AE(N|GG)(AG|Q|K)GSFFE
486.0   (AG|Q|K)AE(I|L)GS(N|GG)YFE
486.0   (AG|Q|K)AE(AG|Q|K)(N|GG)SGFFE
485.0   (AG|Q|K)AE(I|L)GS(N|GG)YFE
485.0   (AG|Q|K)AE(N|GG)AA(N|GG)YFE
485.0   (AG|Q|K)AE(AG|Q|K)(AG|Q|K)EFFE
485.0   (AG|Q|K)AE(AG|Q|K)(N|GG)GSFFE
485.0   (AG|Q|K)AE(I|L)E(AG|Q|K)YFE
485.0   (AG|Q|K)AE(N|GG)(AG|Q|K)(AG|Q|K)YFE
485.0   (AAG|AQ|AK)E(N|GG)(AG|Q|K)SGFFE
484.0   (AAG|AQ|AK)E(N|GG)AAEFFE
484.0   (AAG|AQ|AK)E(N|GG)(AG|Q|K)GSFFE
481.0   (AG|Q|K)AE(I|L)ESGFFE
480.0   (AG|Q|K)AE(I|L)EGSFFE
475.0   (AAG|AQ|AK)E(I|L)GS(N|GG)YFE
475.0   (AAG|AQ|AK)E(AG|Q|K)(N|GG)SGFFE
474.0   (AAG|AQ|AK)E(I|L)GS(N|GG)YFE
474.0   (AAG|AQ|AK)E(N|GG)AA(N|GG)YFE
474.0   (AAG|AQ|AK)E(AG|Q|K)(AG|Q|K)EFFE
474.0   (AAG|AQ|AK)E(AG|Q|K)(N|GG)GSFFE
474.0   (AAG|AQ|AK)E(I|L)E(AG|Q|K)YFE
474.0   (AAG|AQ|AK)E(N|GG)(AG|Q|K)(AG|Q|K)YFE
471.0   (AG|Q|K)AE(AG|Q|K)CST(I|L)FE
470.0   (AG|Q|K)AE(I|L)E(AG|Q|K)YFE
470.0   (AAG|AQ|AK)E(I|L)ESGFFE
469.0   (AAG|AQ|AK)E(I|L)EGSFFE
466.0   (AG|Q|K)AE(N|GG)(AG|Q|K)SGFFE
465.0   (AG|Q|K)AE(N|GG)AAEFFE
```

Fig. 4. 3717.dta: Sequest sequence AKELQEYFK does not appear within the first 30 sequences of AuDeNS but many similar sequences do, e.g., (AAG|AQ|AK)E(I|L)E(AG|Q|K)YFE.,

References

[A01] D. Arnott, Basics of triple-stage quadropole/ion-trap mass spectrometry: precursor and neutral loss scanning. Electrospray ionisation and nanospray ionisation. In [J01], pp 11-29, 2001.

[CKT00] Ting Chen, Ming-Yang Kao, Matthew Tepel, John Rush, George M. Church, A Dynamic Programming Approach to De Novo Peptide Sequencing via Tandem Mass Spectrometry. Proc. of the 11th SIAM-ACM Symposium on Discrete Algorithms (SODA 2000), pp 389-389, 2000.

[DAC99] V. Dančík, T.A. Addona, K.R. Clauser, J.E. Vath, P.A. Pevzner, De Novo Peptide Sequencing via Tandem Mass Spectrometry: A Graph-Theoretical Approach. RECOMB 99, pp 135-144, 1999.

[FGS97] J. Fernandez-de-Cossio, J. Gonzalez, T. Takao, Y. Shimonishi, G. Padron, and V. Besada, A Software Program for the Rapid Sequence Analysis of Unknown Peptides Involving Modifications, Based on MS/MS Data. 45th ASMS Conference on Mass Spectrometry and Allied Topics, Slot 074 (online proceedings at `http://www.asms.org/confASMS.php`), 1997.

[HYS86] D.F. Hunt, J.R. Yates 3rd, J. Shabanowitz, S. Winston, C.R. Hauer, Protein sequencing by tandem mass spectrometry. Proc. National Acad. Sci. USA. **83**:6233-7, Sep. 1986.

[J01] Peter James (ed.), Proteome Research: Mass Spectrometry. Springer, 2001.

[SJ01] W. Staudenmann and P. James, Interpreting peptide tandem mass-spectrometry fragmentation spectra. In [J01], pp 143-165, 2001.

[RF84] P. Roepstorff and J. Fohlman, Proposal for a common nomenclature for sequence ions in mass spectra of peptides. Biomed. Mass Spectrom. **11**:601, Nov. 1984.

[TJ97] J. A. Taylor and R. S. Johnson, Sequence database searches via de novo peptide sequencing by tandem mass spectrometry. Rapid Comm. Mass Spec. **11**:1067-1075, 1997.

[TJ01] J. A. Taylor and R. S. Johnson, Implementation and uses of automated de novo peptide sequencing by tandem mass spectrometry. Anal. Chem. **73**:2594-2604, 2001.

[BioAnalyst:www] `http://www.appliedbiosystems.com`

[biotools:www] `http://www.daltonics.bruker.com`

[BioWorks:www] `http://www.thermo.com`

[Lutefisk:www] `http://www.immunex.com/researcher/lutefisk/`

[MSfast:www] `http://www-hto.usc.edu/ tingchen/resume.html`

[Sequest:www] `http://fields.scripps.edu/sequest/`