

# **Computational Proteomics and Metabolomics**

**Knut Reinert  
(basierend auf Folien von  
Oliver Kohlbacher & Sven Nahnsen  
Eberhard-Karls Universität Tübingen)**

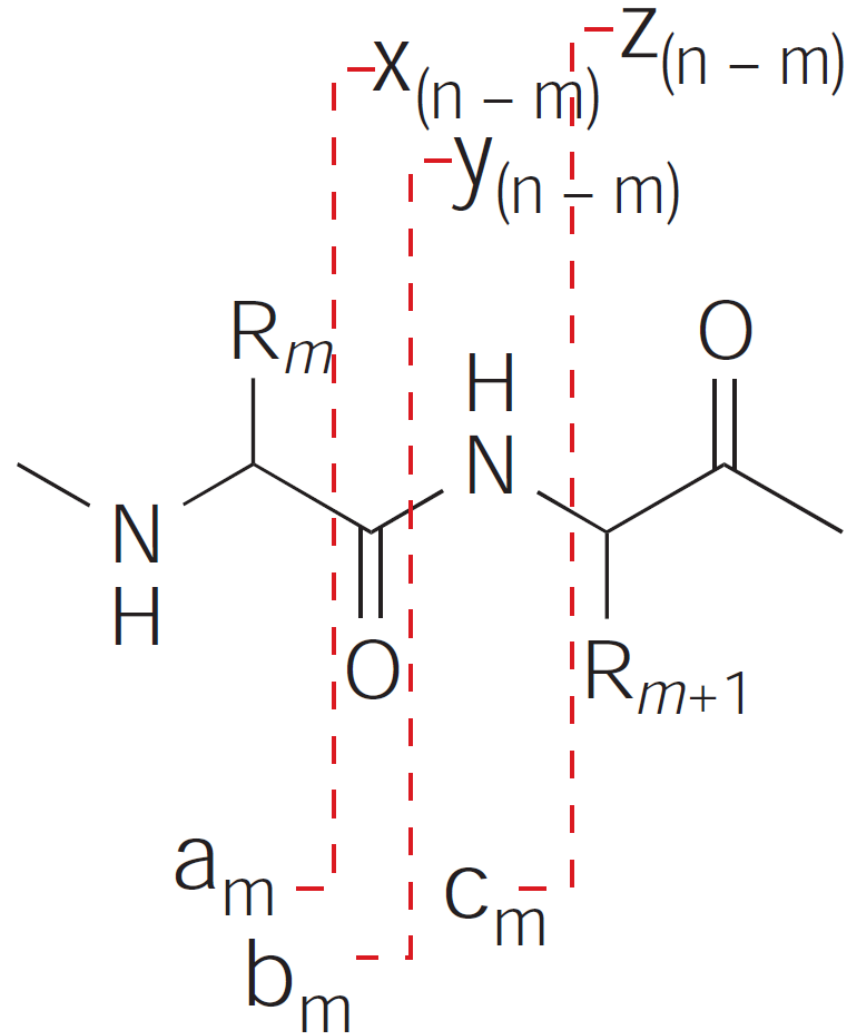
**WS 11/12**

**Peptide ID**

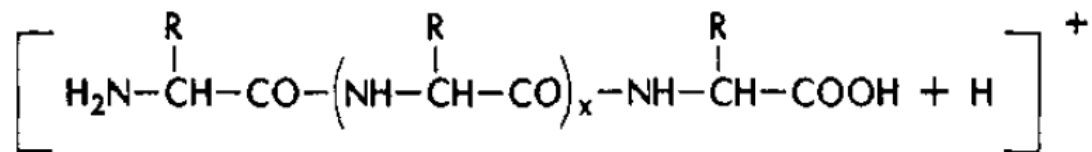
# Product ion generation

- A peptide of length  $n$  can potentially give rise to  $a, b, c$  and  $x, y, z$  ions. This example shows the fragments that can be produced between amino acids  $R_m$  and  $R_{m+1}$
- This nomenclature for fragment ions was first proposed by Roepstorff and Fohlman in 1984

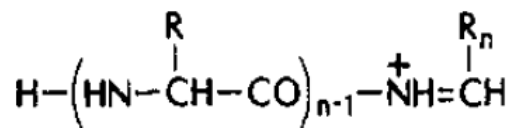
(Roepstorff and Fohlman, *Biological Mass Spectrometry*, Volume 11, Issue 11, page 601, November 1984)



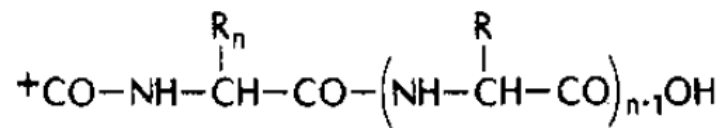
# abc and xyz ions



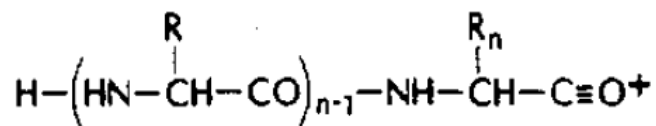
I



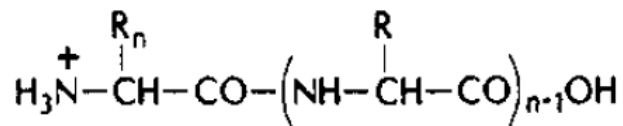
$a_n$



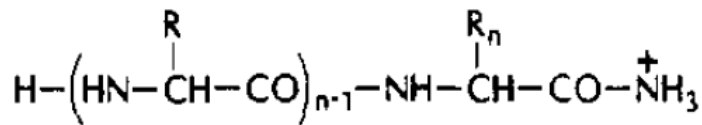
$x_n$



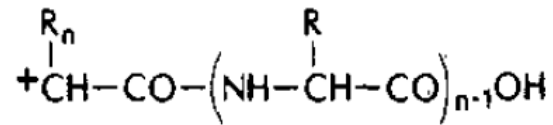
$b_n$



$y_n$



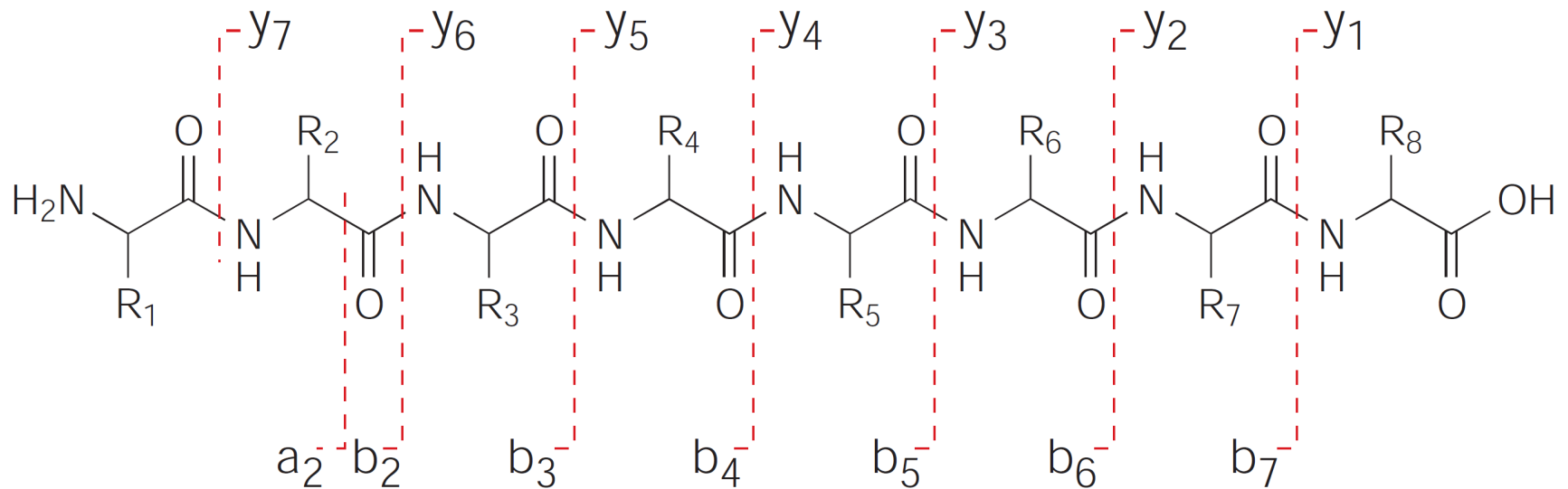
$c_n$



$z_n$

# b/y ions in CID

CID fragmentation predominately produces b and y ions



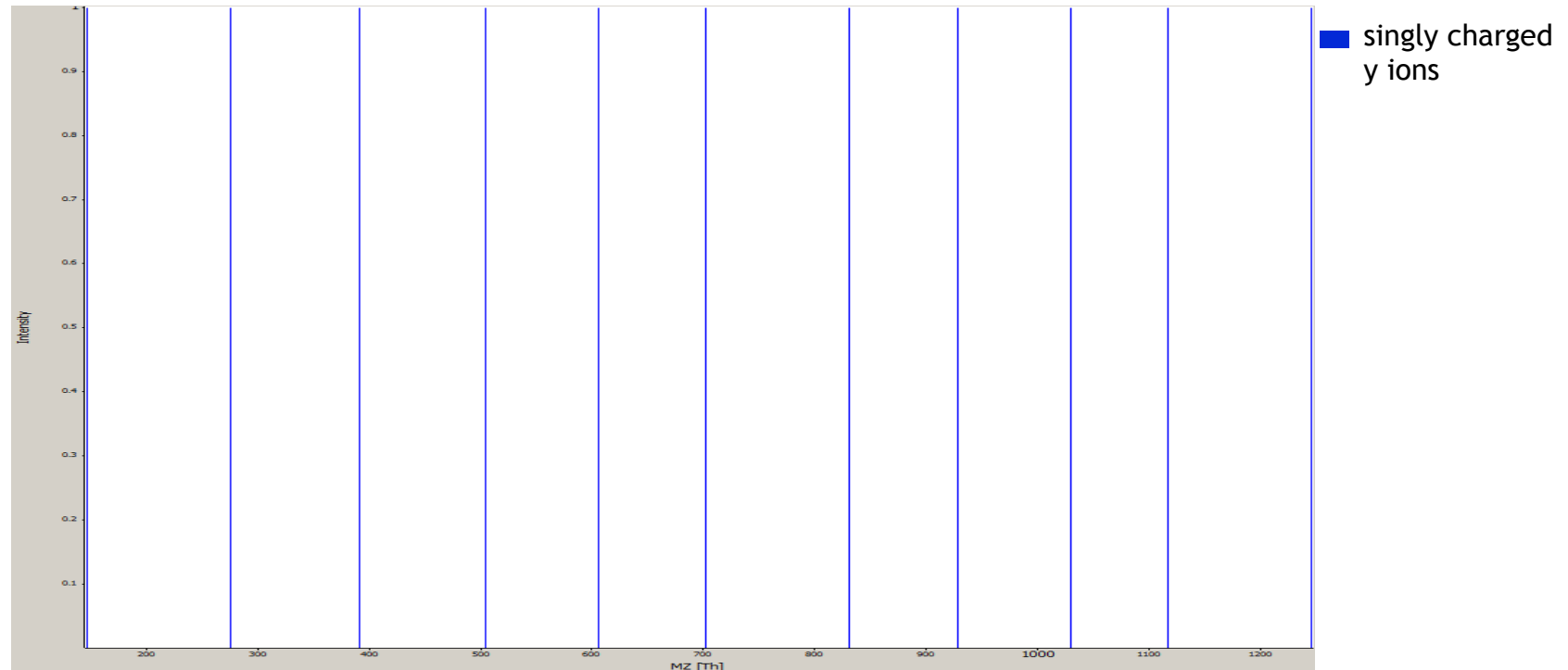
Note:  $y_i$  ion is also called the *sister fragment* of the  $b_{n-i}$  ion and vice versa

# TESTPEPTIDEK

- For simplicity we will consider theoretical spectra for the artificial TESTPEPTIDEK

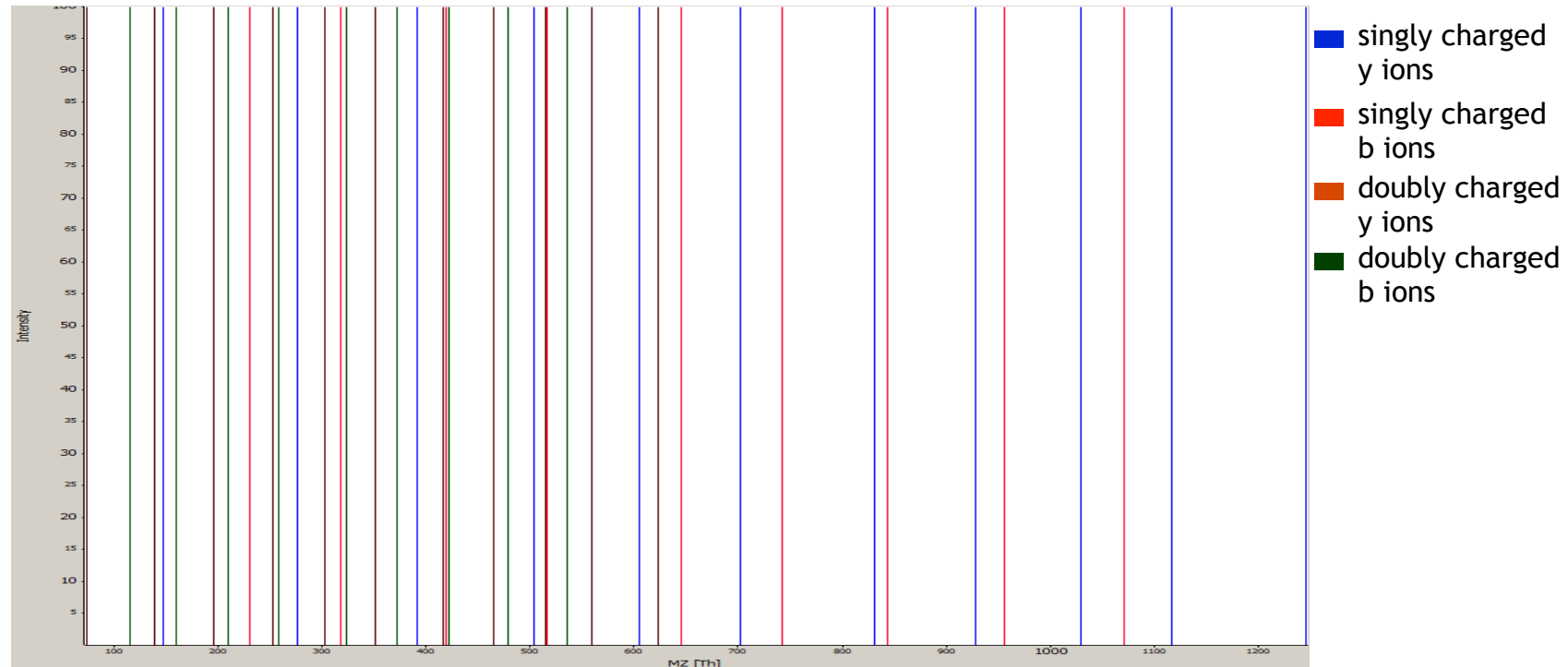
# Ion types in a tandem spectrum

- If a singly charged ion fragments, only one of the sister fragments will be observed (here: artificial peptide: 'TESTPEPTIDEK' with one charge at C-terminus)



# Ion types in a tandem spectrum

- If the same peptide was multiply charged; the charges are usually distributed across the product ions, the tandem spectrum is assumed to contain both sister ions and also doubly charged product ions



# Ion types in a tandem spectrum

- Theoretically, one also observes a, c, x and z ions





# Ion types in a tandem spectrum

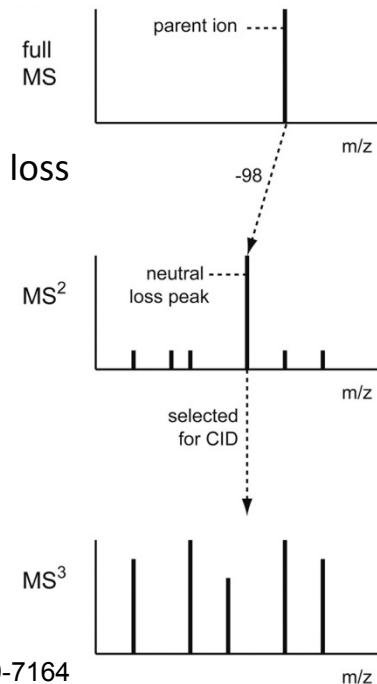
- Theoretically, one also observes a, c, x and z ions
- abc and xyz ions are called backbone ions.

This spectrum contains all theoretical backbone ions of charge 1-2  
(theoretically generated for TESTPEPTIDEK)



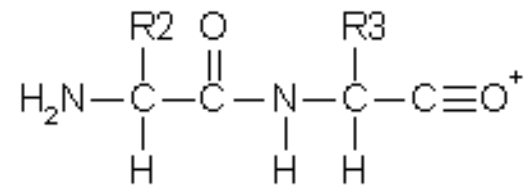
# Neutral losses

- Besides backbone ions, we also observe the precursor ions and precursor ions with neutral losses
- *Neutral losses* most frequently occur as water loss ( $\text{H}_2\text{O}$ : -18.011 Da) on S, T, D and E; as ammonia loss ( $\text{NH}_3$ : -17.027 Da) on R, K, N and Q and as loss of phosphoric acid ( $\text{H}_3\text{PO}_4$ : -98 Da) on S, T and Y
- Neutral losses are uncharged fragments, but result in an additional charged ion with  $\text{mass}_{\text{ion}} - \text{mass}_{\text{neutral loss}}$
- The problem of very intense ions, resulting from neutral losses of precursor ions, can be overcome by triggering an additional fragmentation.

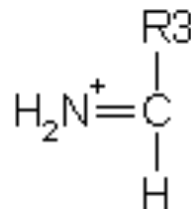


# Internal fragments

- Internal fragments result from double backbone fragmentation. Usually, these are formed by a combination of *b*-type and *y*-type ions, and consist of five residues or less

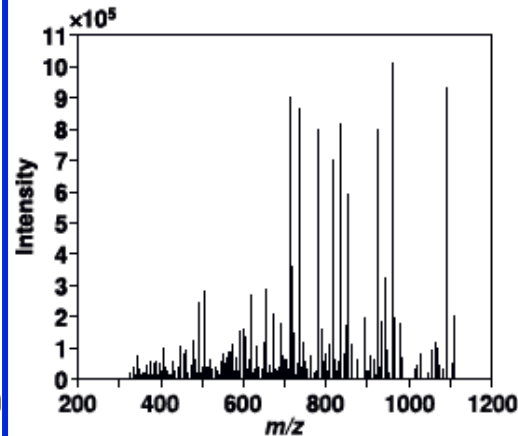
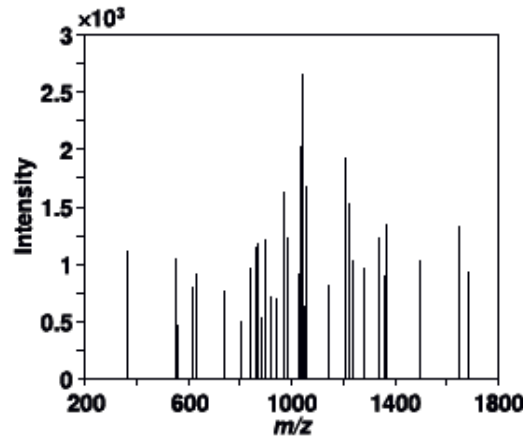


- Immonium ions are a special case of internal fragments. They are composed of a single side chain formed by a combination of *a*-type and *y*-type fragmentation

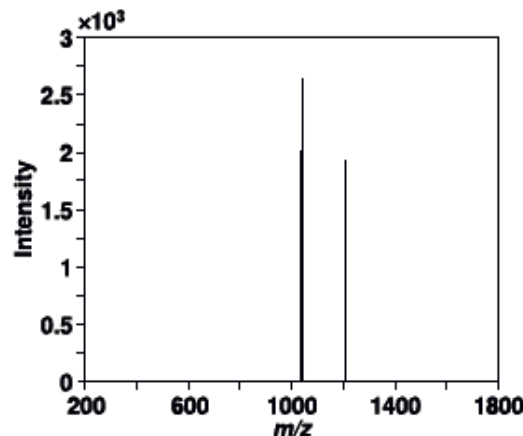


# Noise in tandem spectra

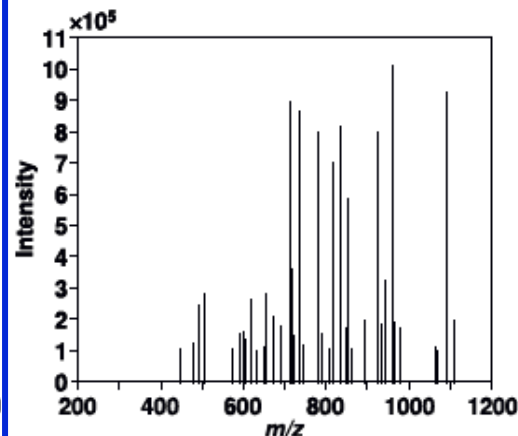
- In addition to the various types of ions, there is also noise in tandem spectra



*With noise*



*Blank run*



*Without noise*

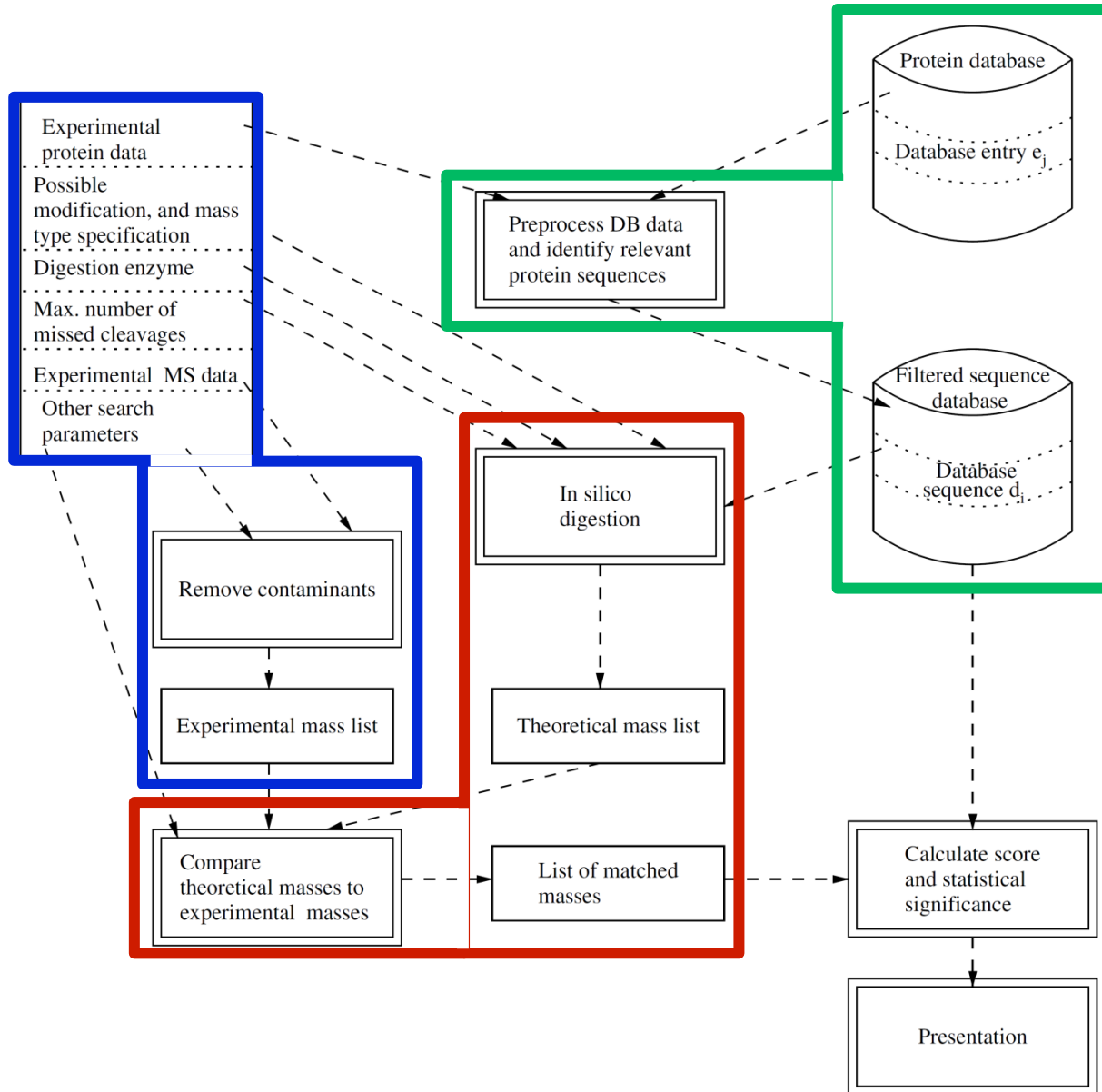
*One isolated peptide*

# Summary ion types

- Due to different fragmentation efficacies and different response factors, fragment ions will have different intensities
- These intensities can be predicted using machine learning techniques and appropriate fragmentation models, however, most search engines do **not** include intensity information, but only the masses
- In general, a simple peptide search engine should consider *b* and *y* type ions, doubly charged *b* and *y* type ( $b^{2+}$ ,  $y^{2+}$ ) ions and optionally  $b^{-NH_3}$ ,  $y^{-NH_3}$  and  $b^{-H_2O}$ ,  $y^{-H_2O}$

# Identification workflow

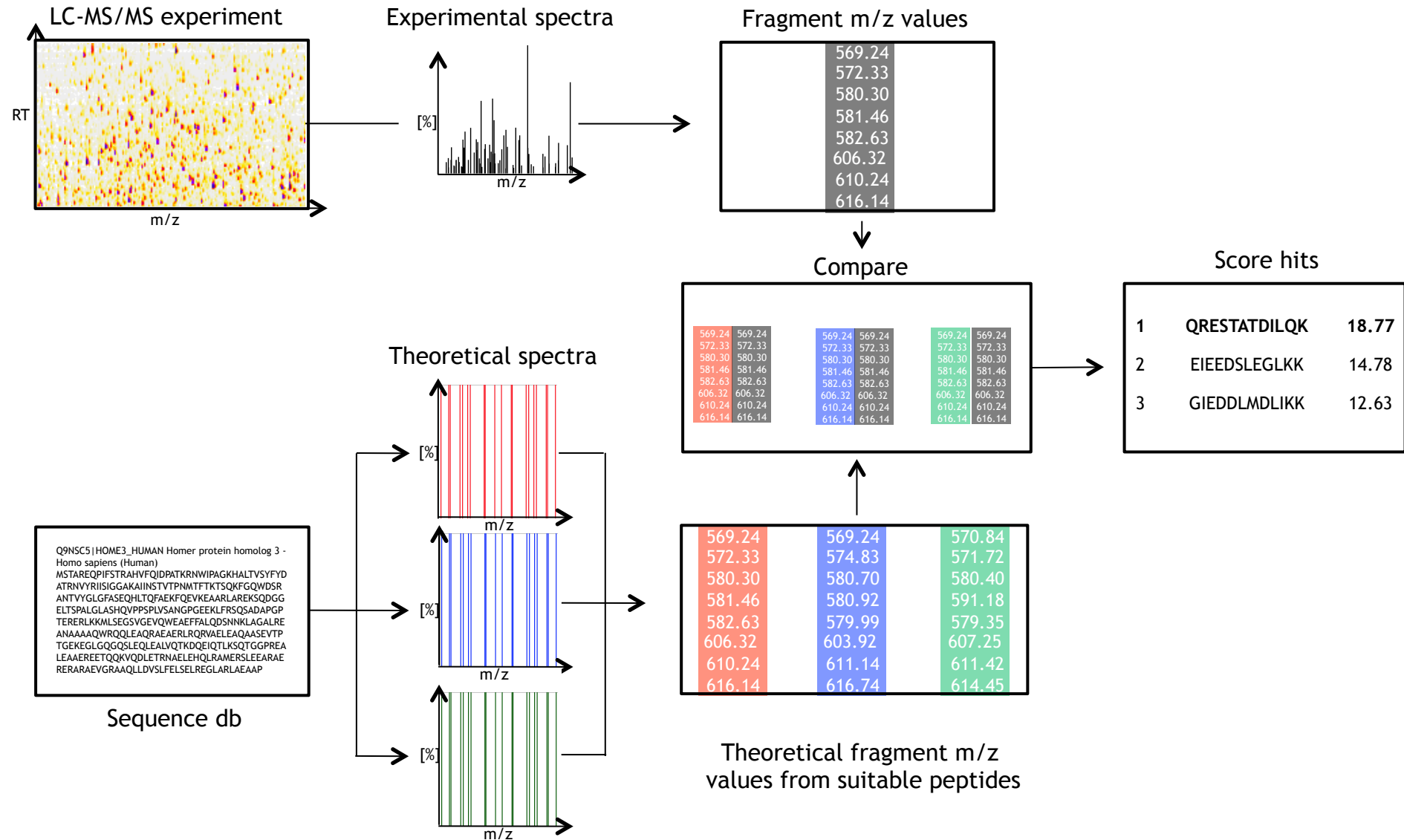
Experimental parameters →



← DB settings

Search engine →

# Peptide identification

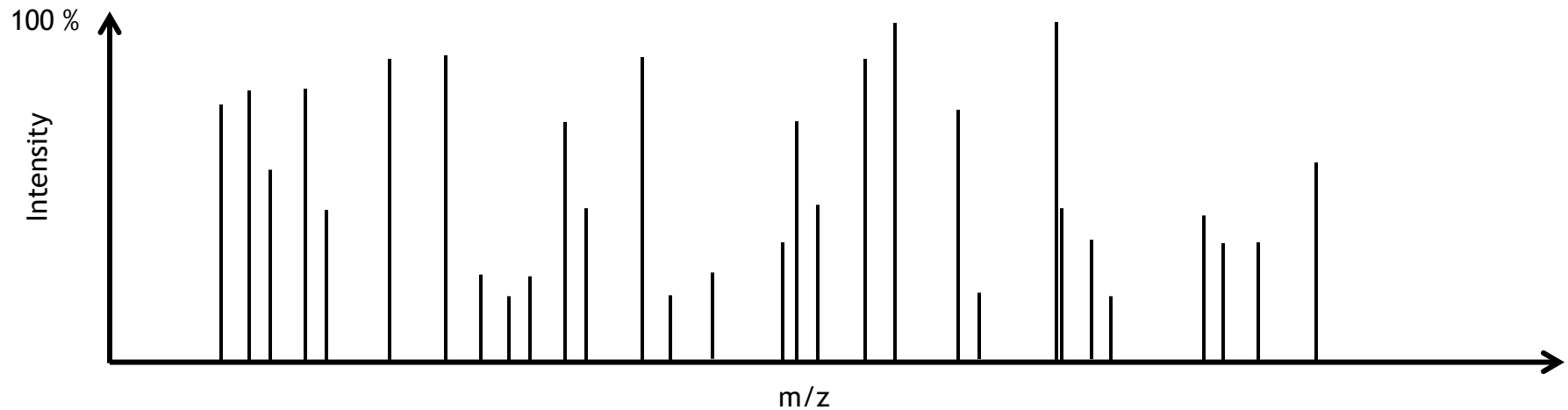


# Peptide identification

1. From the database, extract all sequences that fit the precursor mass of the *MS2* spectrum with a given error tolerance
2. For each of these candidates a theoretical spectrum is generated
3. All theoretical spectra are aligned / compared to the experimental spectrum
4. The alignments are scored and the candidates are ranked according to the score
5. The top ranked candidate is assumed to be the correct PSM (Peptide Spectrum Matching)



# 1. Extract all candidates (search space)



- Given: Experimental spectrum  $S$
  - Task: Identify the correct sequence for  $S$  from a given protein database
1. Define the search space for  $S$  for a given mass tolerance  $d$ :
    - $m_{prec}$  is the mass of the precursor ion of spectrum  $S$
    - From the database, extract all peptide sequences with mass  $m_{cand}$  given that

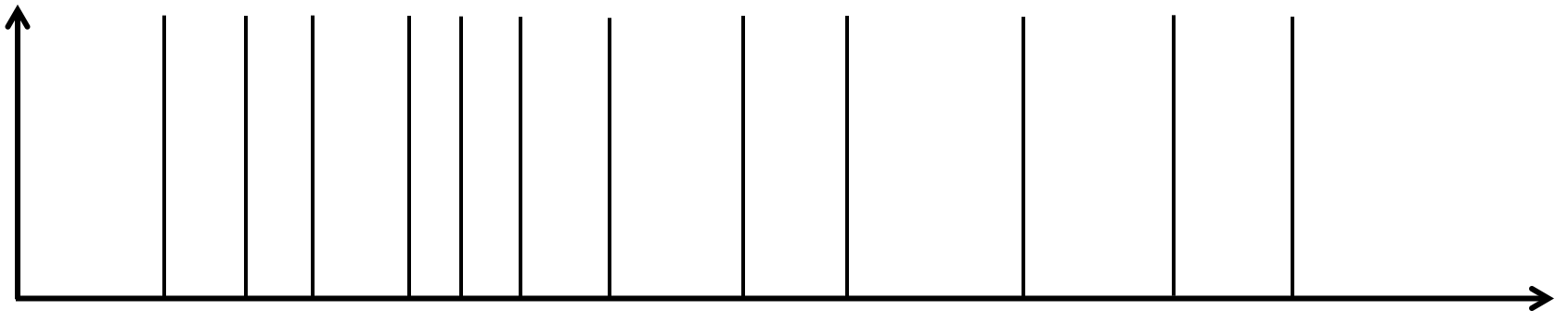
$$|m_{prec} - m_{cand}| \leq d$$

- This set of candidates is defined as the search space for spectrum  $S$  and denoted as

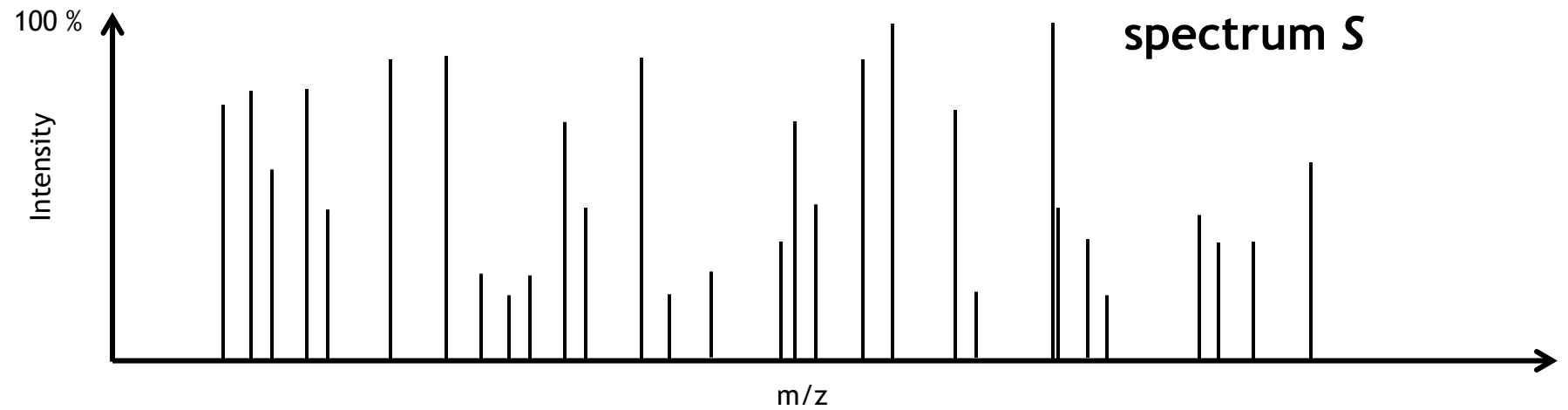
$$\Omega_S$$

## 2. Generate theoretical spectra

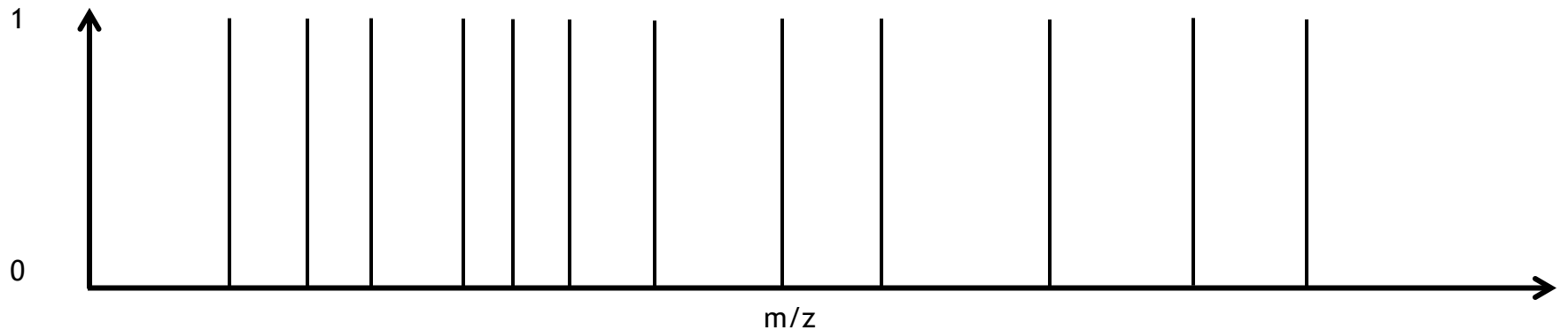
- 1<sup>st</sup> option: extract all masses from the MS2 spectrum
- 2<sup>nd</sup> option: try to model fragment ion intensities



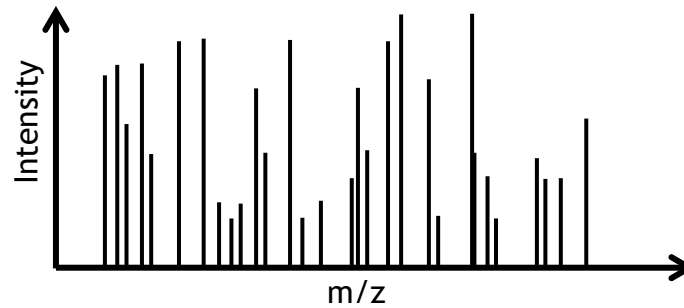
### 3. Comparison to experimental spectra



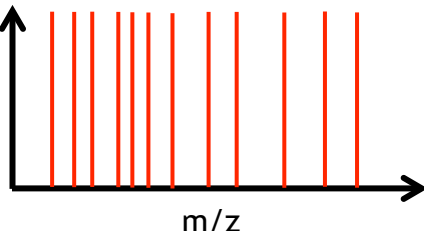
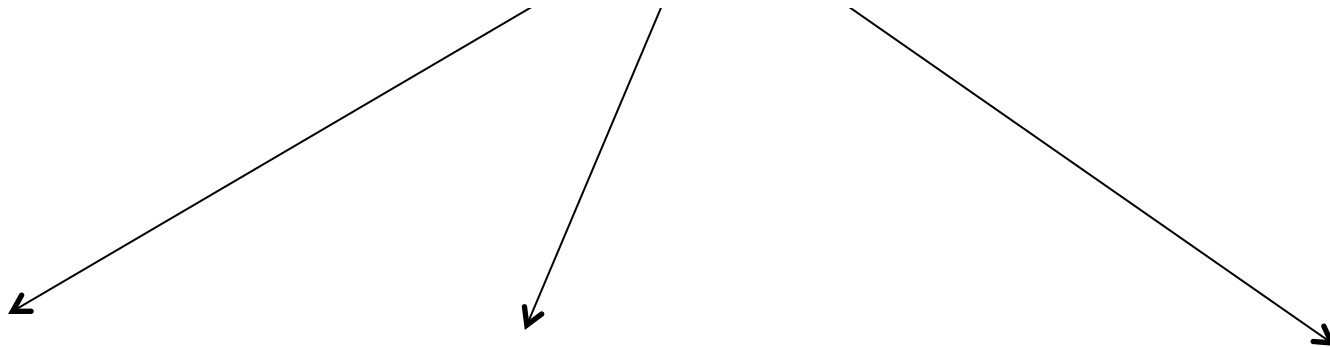
Theoretical spectrum **T**, generated from a sequence  $p_i \in \Omega_S$



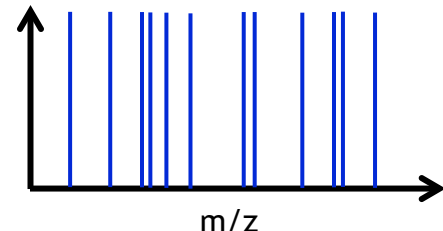
# 3. Comparison to experimental spectra



2. Compare theoretical spectra for all  $p_i \in \Omega_S$  to the experimental spectrum  $S$

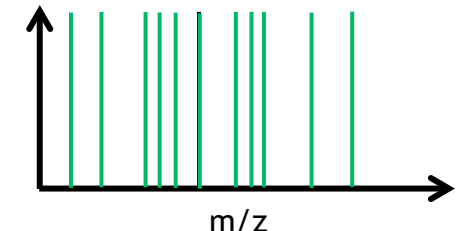


$p_1 \in \Omega_S$



$p_2 \in \Omega_S$

...



$p_n \in \Omega_S$

## 4. Scoring of peptide candidates

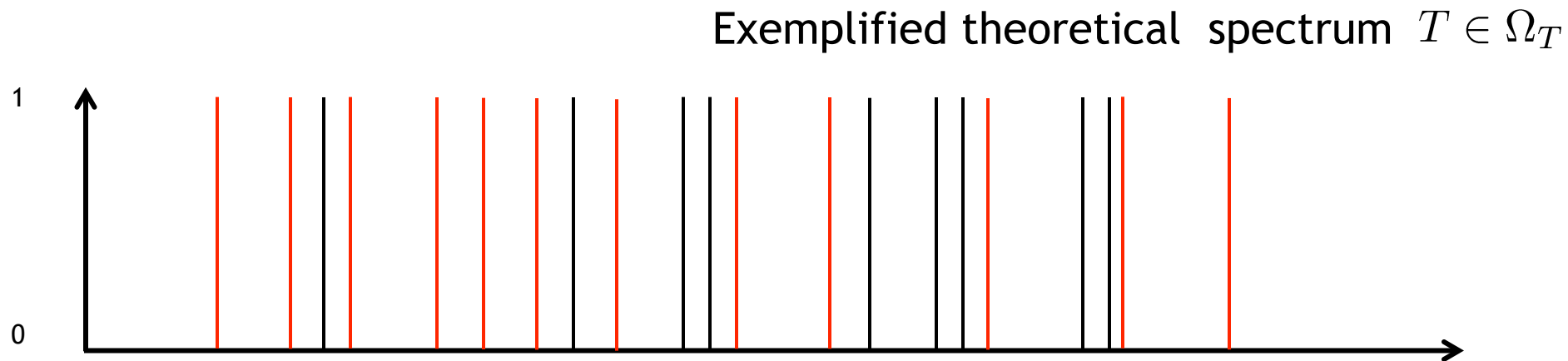
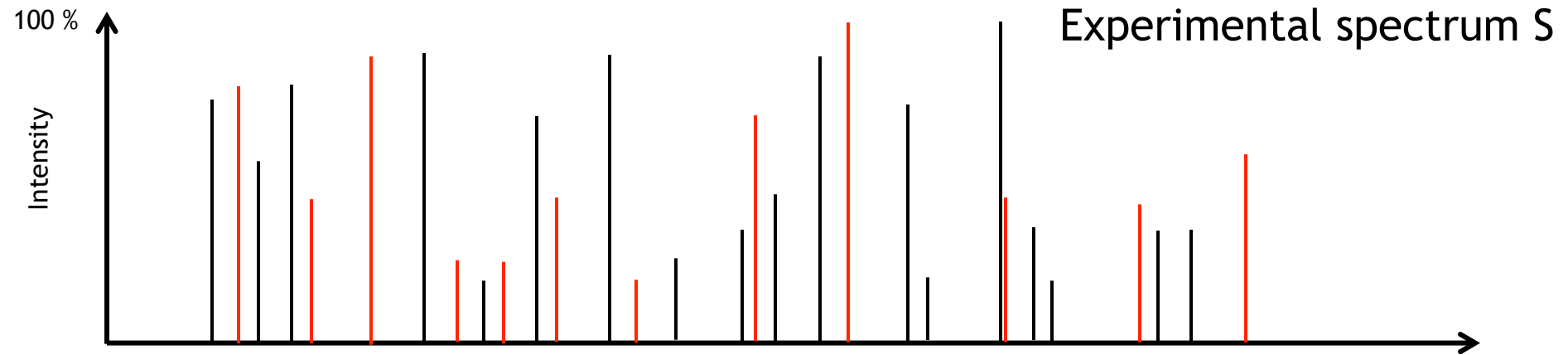
- There are numerous tools for the comparison of theoretical and experimental candidate peptides
- The main difference of search engines is the implementation of the scoring schemes (resulting in differences in runtime and performance)
- However, conceptually all search engine algorithms are based on fragment ion comparison
- In the following, we will discuss briefly
  - **X!Tandem**, Craig,R. and Beavis,R.C. (2003) *Rapid Commun. Mass Spectrom.*, 17, 2310-2316
  - **OMSSA**, Geer et al. (2004) *J Proteome Res.* 2004 Sep-Oct;3(5):958-64.
  - **Sequest** Eng et al., *J. Am. Soc. Mass Spectrom.* **1994**, 5, 976-989.

# X!Tandem

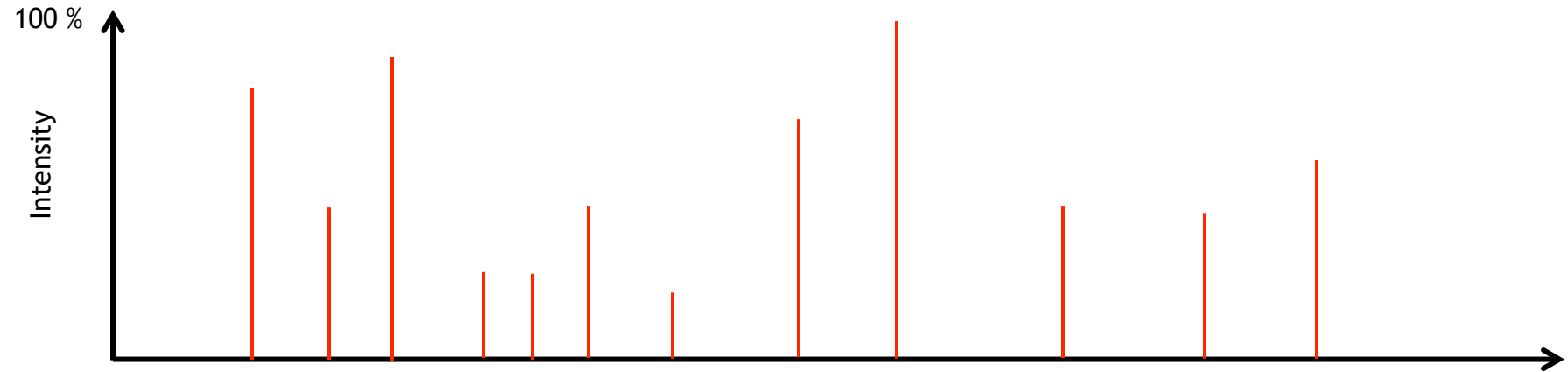
- Craig, R. and Beavis, R.C. (2003) *Rapid Commun. Mass Spectrom.*, **17**, 2310-2316.
- <http://www.thegpm.org/tandem/instructions.html>

# Find overlapping masses

To find overlapping masses, a maximal **fragment mass tolerance** window needs to be set (for ion traps this is usually 0.5 Da)



# X!Tandem's dot product



- Reduce the experimental spectrum to only those peaks that match peaks in the theoretical spectrum
- Calculate dot product (dp) (using ion intensities and the number of matching ions)

$$dp = \sum_{i=0}^n I_i P_i$$

*Intensities from experimental spectrum*  
 $I_i$  ... fragment ion intensities

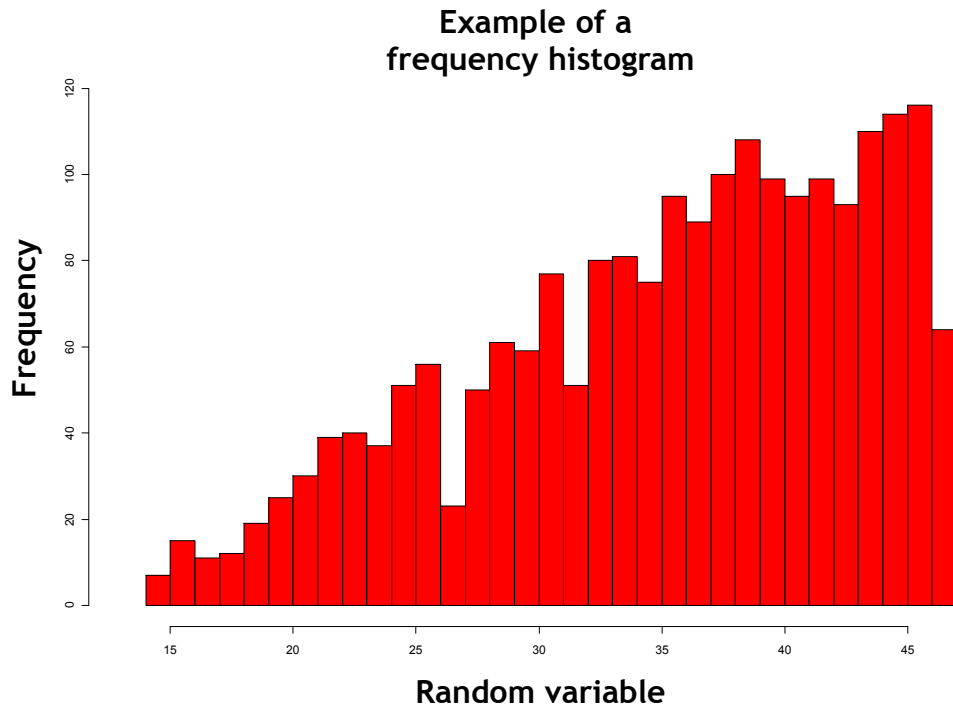
*Predicted or not in theoretical spectrum*  $P_i \in \{0, 1\}$



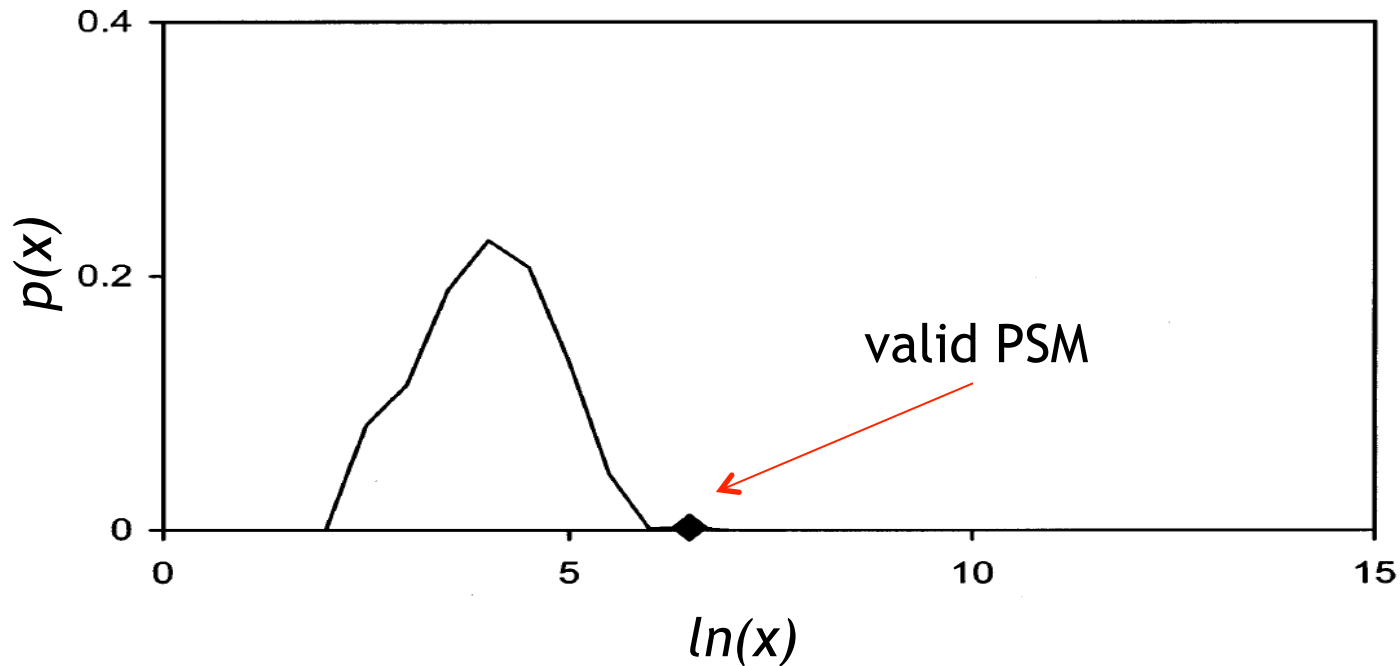
# Survival function and e-value

- Let  $x$  represent the dot product score for the experimental spectrum  $S$  and the theoretical spectrum  $T \in \Omega$
- $p(x)$  is calculated from the frequency histogram (counts of PSMs per score bin)
- With  $f(x)$ , the number of PSMs that are given the score  $x$ ,  $p(x)$  is calculated with

$$p(x) = \frac{f(x)}{N} \text{ , with } N \dots \text{ total number of PSMs}$$



# Survival function and e-value

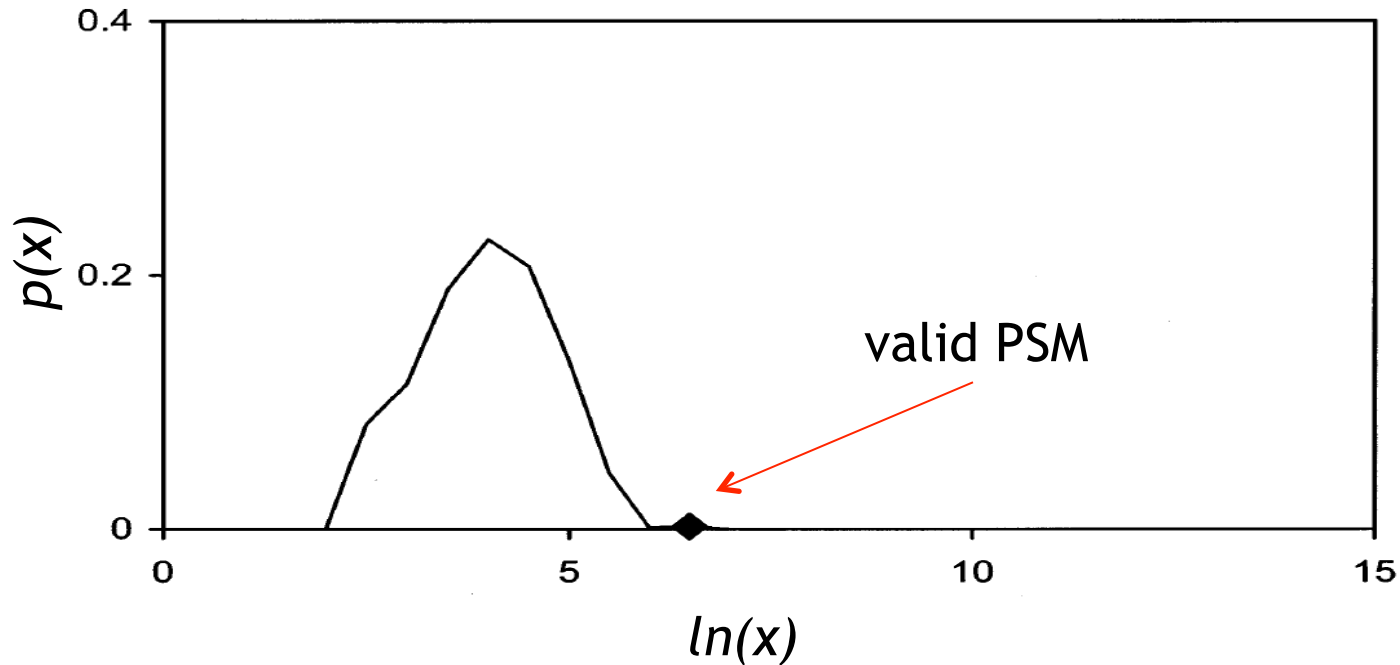


- The survival function,  $s(x)$ , for a discrete stochastic score probability distribution,  $p(x)$  is defined as

$$s(x) = P(X > x) = \sum_{X > x} p(x)$$

where  $P(X > x)$  is the probability to have a greater value than  $x$

# Survival function and e-value



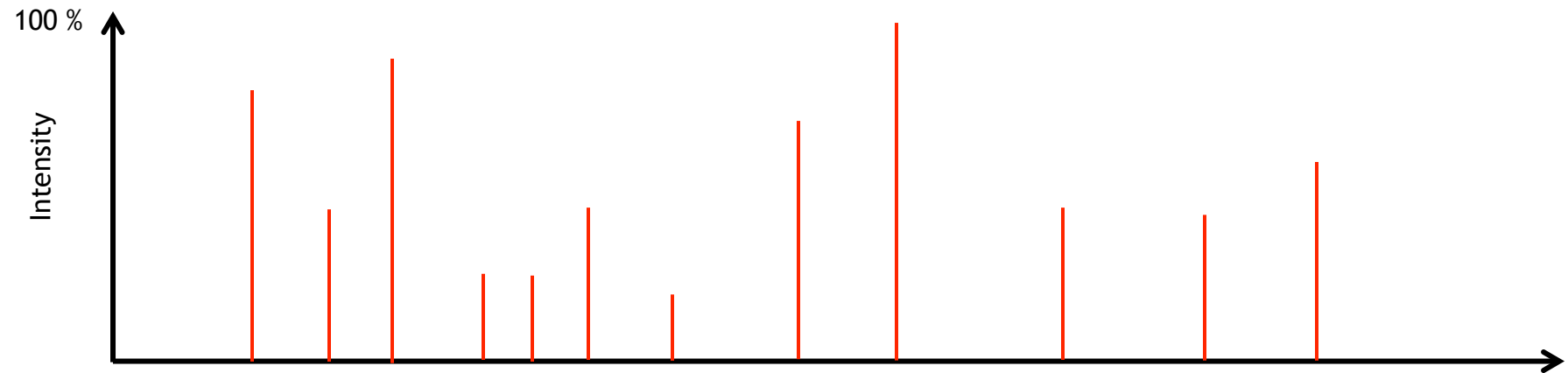
- With the survival function  $s(x)$ , we can calculate the E-value  $e(x)$ , indicating the number of PSMs that are expected to have scores of  $x$  or better

$$e(x) = ns(x)$$

where  $n$  is the number of sequences in  $\Omega_S$

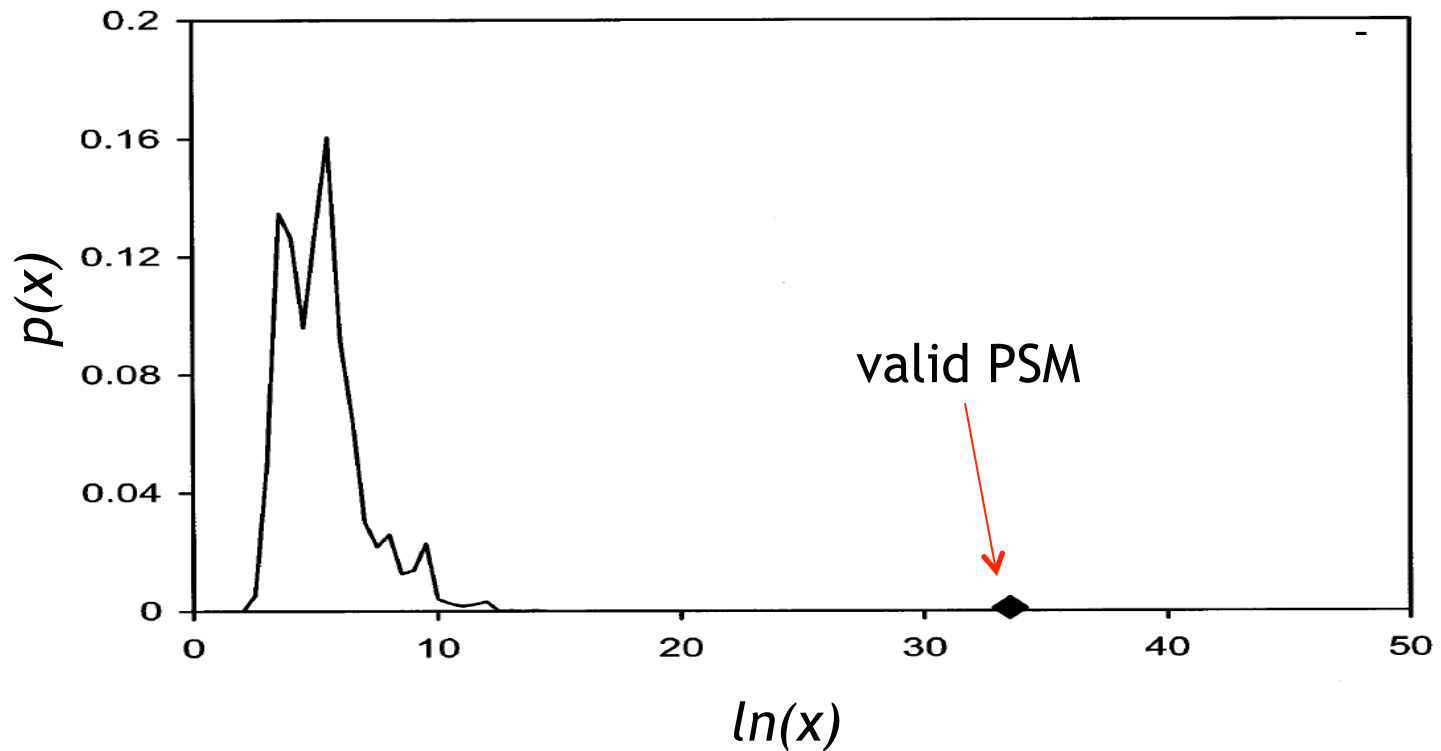
- Now, each PSM can be ranked according to  $e(x)$

# X!Tandem Hyperscore



- The hyperscore (HS) is calculated by multiplying with factorials of the number of assigned b and y ions.
- The use of the factorials is based on the hypergeometric distribution that is assumed for matches of product ions

$$HS = \left( \sum_{i=0}^n I_i P_i \right) N_b! N_y!$$



- If  $p(x)$  is now plotted as a function of their  $\log(\text{hyperscores})$ , the valid PSM is much better separated from the bulk of incorrect assignments

# OMSSA

- Geer et al. (2004) J Proteome Res. 2004 Sep-Oct; 3(5):958-64.
- <http://pubchem.ncbi.nlm.nih.gov/omssa/>

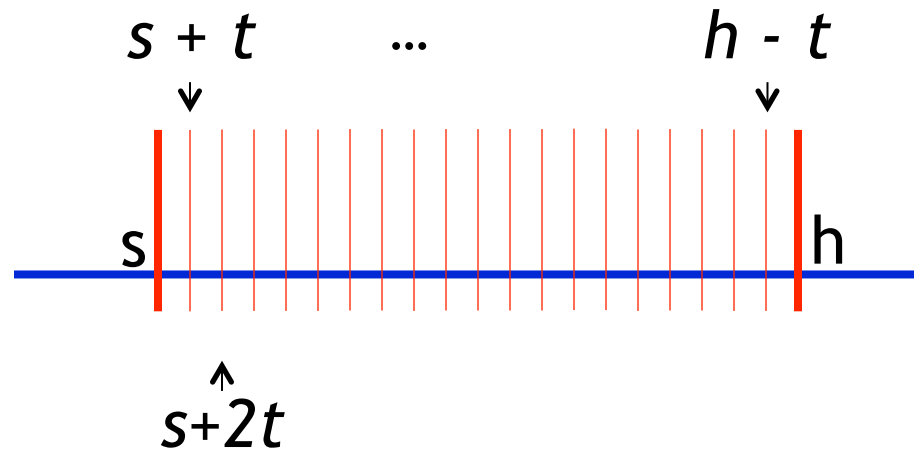
# OMSSA scoring

- The OMSSA algorithm calculates e-values based on how many product ions from theoretical candidates  $T \in \Omega_S$  can randomly hit the tandem spectrum  $S$
- In brief, it considers random matches to a given  $m/z$  value
- The distribution of these random matches allows to assign probabilities for PSMs
- This distribution is constructed separately for different charge states

# OMSSA scoring

For a given spectrum, assume charge state 1+:

- Let  $s$  be the smallest measured product ion mass, and  $h$  be the highest measured production ion mass
- Then, there can be  $\frac{h-s}{2t}$  possible matches if the fragment absolute mass tolerance is  $t$  [Da].

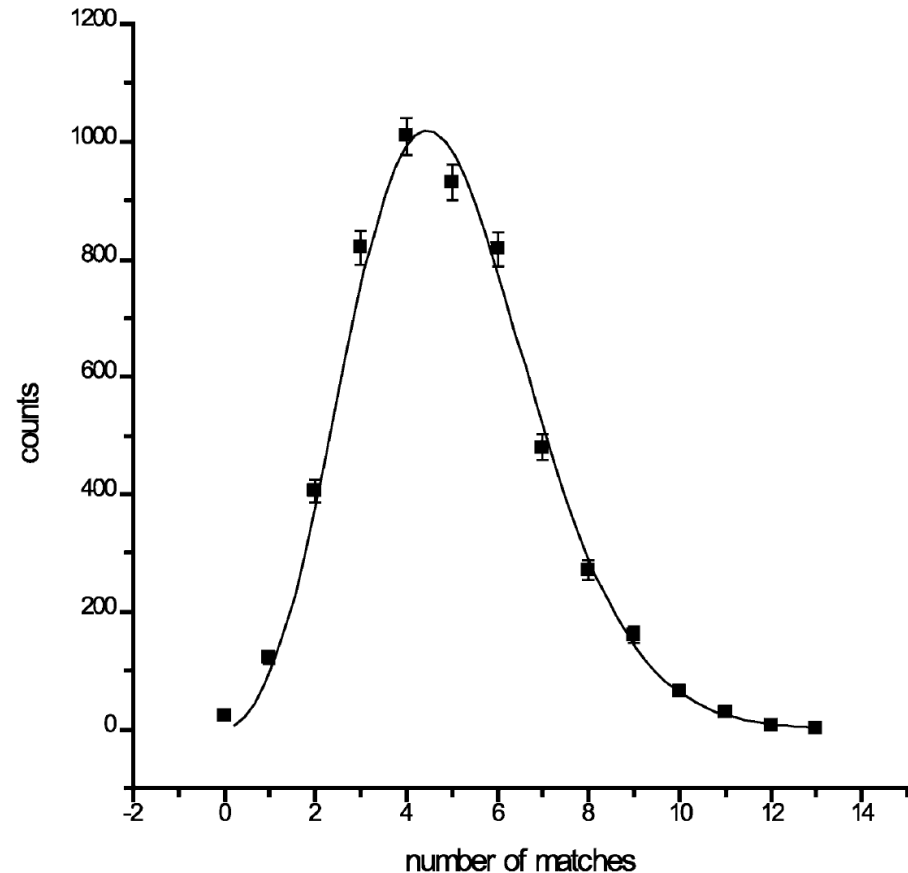


- If a fragment ion  $l$  is measured with  $m_{\text{measure}} = s + t$ , we can assume for the real mass ( $m_{\text{real}}$ ):  $m_{\text{real}} \in \{s, s + 2t\}$



# OMSSA scoring

- Furthermore, if  $m$  denotes the measured precursor mass, then there are  $\frac{k(h-s)}{m}$  product ions, if  $k$  is the total number of calculated  $m/z$  values
- This number needs to be matched to  $e$  experimental product ions
- The number of matches is assumed to follow a Poisson distribution



Geer et al., J Proteome Res. 2004 Sep-Oct;3(5):958-64.

Sadygov and Yates, Anal. Chem. 2003 Aug; 75(15):3792-3798.

# OMSSA – model for charge state 1

- Assuming an underlying Poisson distribution allows to determine the mean for the singly charged case

$$\mu_1 = \left( \frac{2t}{h-s} \right) \left( \frac{k(h-s)}{m} \right) e = \frac{2tke}{m}$$

- This is the mean value for the Poisson distribution  $P(x, \mu)$

$$P(x, \mu) = \frac{\mu^x}{x!} e^{-\mu}$$

$x$  is the number of matches and  $\mu$  is separately calculated for the different charge states ( $\mu_1$  denotes charge state 1)

- After the modeling of the distribution for charge state 1, a combined distribution for charge states +1 and +2 can be derived and so on

# OMSSA – model for charge state +1 and +2

- If +1 and +2 product ions are present, then the spectrum can be split in two ranges (A: containing only +1 ions  $\frac{m}{z} > \frac{m}{2}$  and B: containing +1 and +2 ions ( $\frac{m}{z} < \frac{m}{2}$ )
- Each region is modeled with a separate Poisson distribution
- Region A:

$$\mu_A = \left( \frac{2t}{h - \frac{m}{2}} \right) \left( \frac{k(h - \frac{m}{2})}{m} \right) e^{\frac{(h - \frac{m}{2})}{h - s}} = \frac{2tke}{m} \frac{(h - \frac{m}{2})}{h - s}$$

↑ Corresponds to the portion of e, that lies in A

Corresponds to the calculation of  $\mu_1$ ,  $s =$

$$s = \frac{m}{2}$$

2

$$\mu_B = \left( \frac{2t}{\frac{m}{2} - s} \right) \left( \frac{k(\frac{m}{2} - s)}{m} + \frac{k(\frac{m}{2} - s)}{\frac{m}{2}} \right) e^{\frac{(\frac{m}{2} - s)}{h - s}} = \frac{6tke}{m} \frac{(\frac{m}{2} - s)}{h - s}$$

- From elementary probability distribution,

$$\mu_2 = \mu_A + \mu_B = \frac{2tke}{m} \frac{(h + m - 3s)}{h - s}$$

- This can be continued for further charge states...

# OMSSA – improve performance

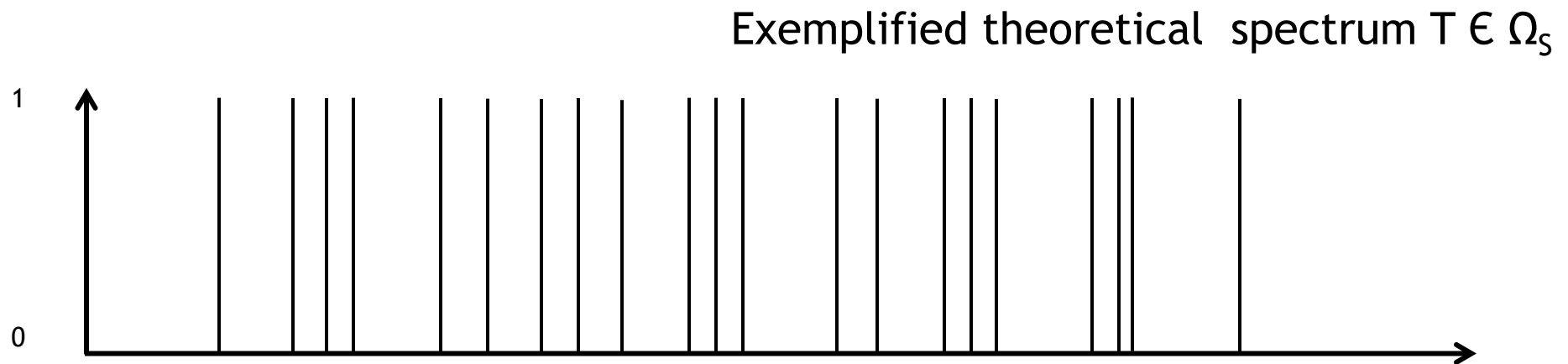
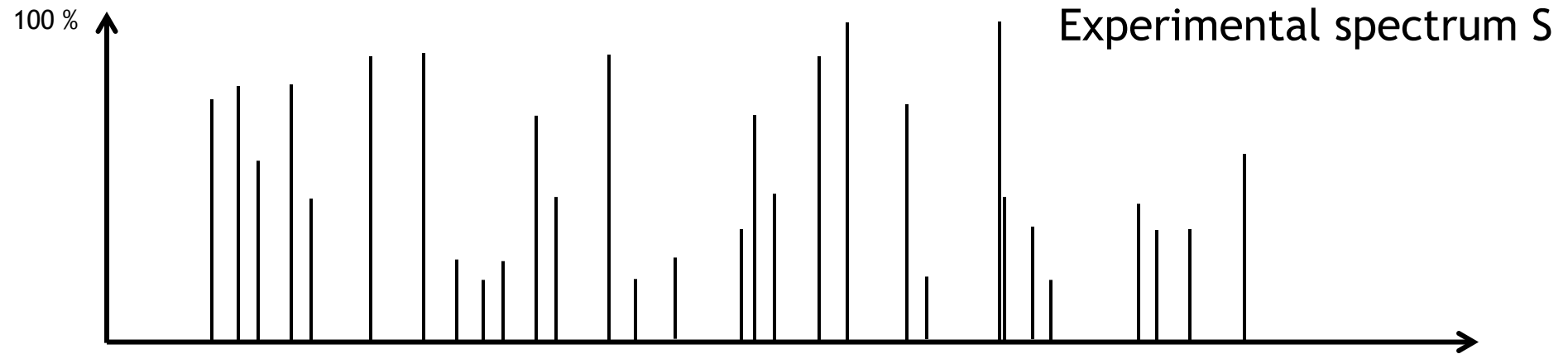
- OMSSA includes a small trick to make the algorithm more efficient and more sensitive
- For candidate selection:
  1. Require that at least one of the  $m/z$  in T (theoretical spectrum) matches one of top  $n$  ( $n=3$  by default) values in S (experimental spectrum).
  2. This selection modulates the probability distribution.  
We take  $q = \frac{n}{e}$  as the probability that a measured  $m/z$  value matches a theoretical one. Then the new distribution  $P'$  is:

$$P'(x, \mu) = \frac{1}{Q} (1 - (1 - q)^x) P(x, \mu)$$

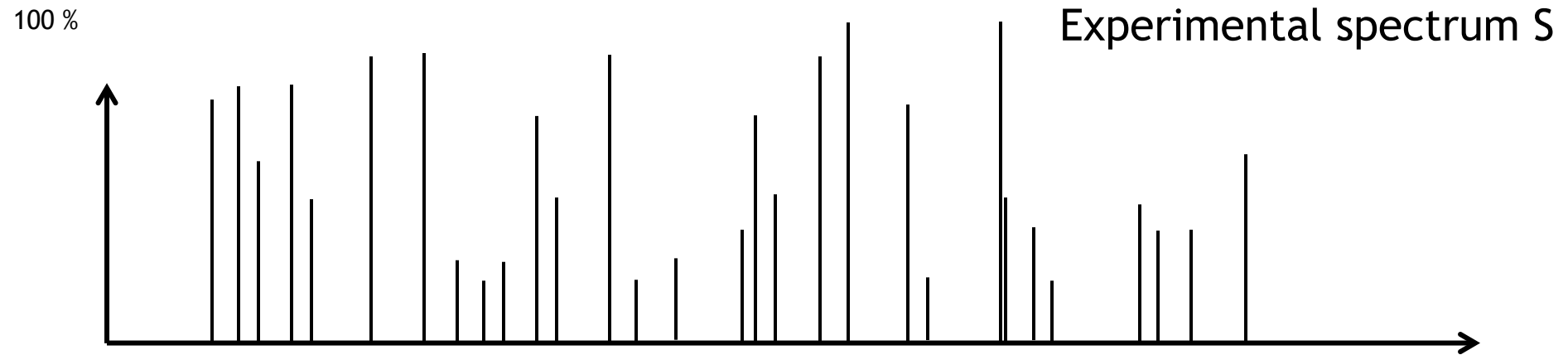
with the normalization factor  $Q$

$$Q = \sum_x (1 - (1 - q)^x) P(x, \mu)$$

# Sequest

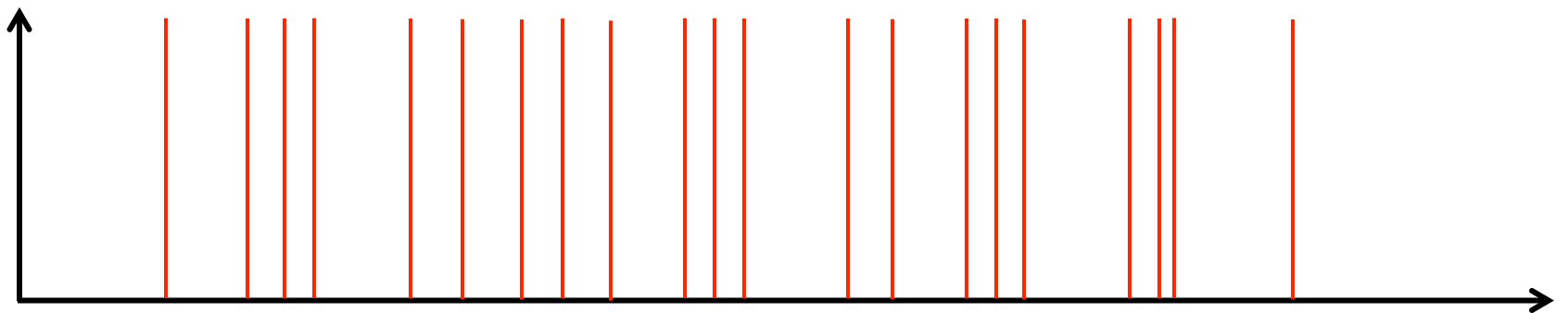
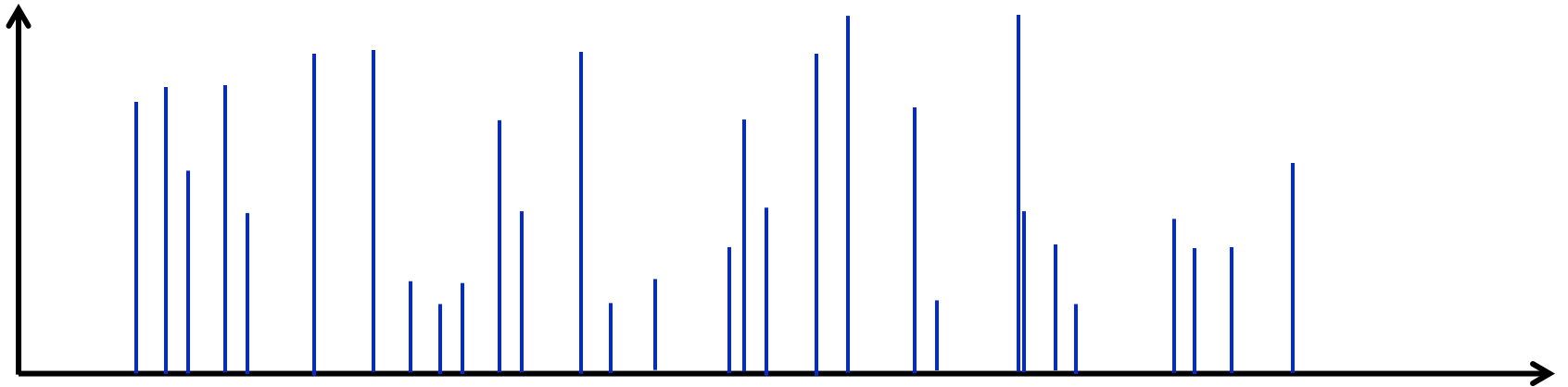


# Sequest – Cross correlation



- Sum all the peaks that overlap between theoretical and experimental spectrum
  - This score is called **Cross-correlation**
- 
- This figure shows a bar chart illustrating the cross-correlation between a theoretical spectrum and an experimental spectrum. The vertical axis is labeled '1' at the top and '0' at the bottom. The horizontal axis represents frequency. The plot shows a series of vertical black bars representing the theoretical spectrum. Overlaid on this are several vertical red bars representing the experimental spectrum. The red bars are positioned at the same frequency locations as the black bars, indicating a match between the two spectra. The text 'This score is called Cross-correlation' is written across the plot area.

# Sequest – Autocorrelation



# Sequest – $X_{corr}$ score

- By shifting the spectra, the assumption is that the peaks should not overlap. The spectra are displaced by  $x$  Da
- The peaks that overlap upon spectra shifting are used to calculate the autocorrelation

- Sequest reports

$X_{corr}$  scores

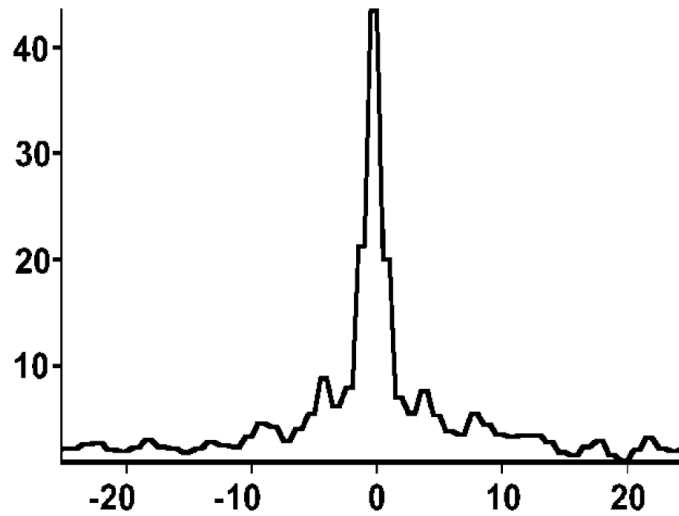
$$X_{corr} = \frac{Cross_{corr}}{Average(Auto_{corr})_{-75 \leq x \leq 75}}$$

for displacement

$$x \text{ [Da]} \in \{-75, 75\}$$

correlation count

$\times 10^3$



Displacement x in Da



# Sequest – $\Delta C_n$ score

- $X_{corr}$  scores can be calculated for every theoretical spectrum in the search space  $\Omega_S$  for an experimental spectrum  $S$
- Additionally to the  $X_{corr}$  score, Sequest also calculates the  $\Delta C_n$  score for the top scoring PSM (best  $X_{corr}$ )
- This score measures how good the best score is in relation to the second best

$$\Delta C_n = \frac{X_{cross1} - X_{cross2}}{X_{cross1}}$$

# Other search engines

- Mascot from Matrix Science (<http://www.matrixscience.com/>)
  - Mascot is one of the most popular search engines
  - Commercial software
  - Algorithmic details have never been published
  - Mascot calculates  $p$ -values for all candidates in the search space and ranks the output according to these  $p$ -values
- Phenyx
  - Commercial software
  - Colinge et al., Proteomics. Vol. 3, No. 8, August 2003, pp. 1454-1463.
- InsPecT
  - Very fast open-source search engine
  - Designed for the identification of posttranslational modification
  - Tanner et al., J Proteome Res. 2005 Jul-Aug;4(4):1287-95.
- Myrimatch
  - Open source
  - Tabb et al., J Proteome Res. 6(2) 654-61. 2007 Feb

# Search settings

- OpenMS offers TOPP tools for the most common search engines
- .ini files allow to adjust the parameters
- This is an example for X! Tandem settings for analyzing LTQ-Orbitrap data

parameter	value
XTandemAdapter	
1	
in	
out	
precursor_mass_tolerance	10
fragment_mass_tolerance	0.5
precursor_error_units	ppm
fragment_error_units	Da
database	choose a database
min_precursor_charge	2
max_precursor_charge	5
fixed_modifications	[Carbamidomethyl (C)]
variable_modifications	[Oxidation (M), Deamidated (Q), Deamidated (N)]
missed_deavages	2
xtandem_executable	
default_input_file	
minimum_fragment_mz	150
cleavage_site	[RK]{}{P}
max_valid_expect	10
no_refinement	false
threads	2
no_progress	false

Disables progress logging to command line

Show advanced parameters

# Mass tolerance settings

- Mass tolerance settings:
  - Easy to estimate when knowing the instrument, calibration runs
  - Precursor tolerance determines search space
    - should be stringent, but broad enough to have several entries per search space (e.g., for E-value calculation)
    - 5-10 ppm is commonly used for data acquired on well-calibrated Orbitrap instruments
  - Product (or fragment) tolerance determines the number of theoretical fragment ions that can be matched to the experimental spectrum
    - again, should be stringent, but also provide enough flexibility for statistical assessment (e.g., drawing the Poisson distribution in the OMSSA algorithm)
    - 0.5 Da is commonly used for data recorded by ion traps (e.g. LTQ)

# Charge states and missed cleavages

## Charge state settings

- Frequently, the mass spectrometer is set to only fragment features with charge  $> 1$
- If you know your data is restricted to several charge states (e.g., for your mass spectrometric settings), you can save time by not looking at these

## Missed cleavages

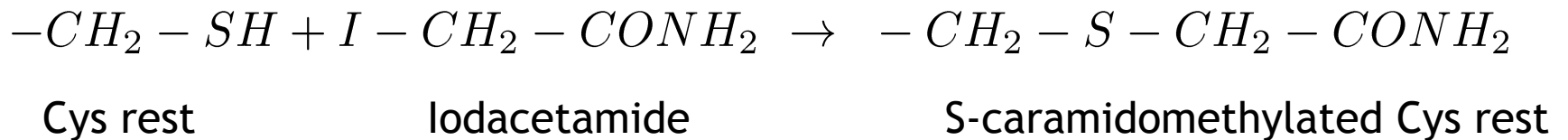
- Sometimes, proteases don't cleave perfectly
- 1 or 2 missed cleavages should be allowed, but be careful since the number of missed cleavages increases your search space sizes!

# Modifications

- The modification settings mostly depend on the biochemical assays used for sample preparation

## Fixed modifications

- **Carbamidomethylation of cysteins** is used as fixed modification in most experiments, since proteins are usually subjected to a DL-Dithiothreitol (DTT) treatment to reduce disulfide bonds built by cysteins. To protect the liberated –SH the samples are treated with Iodoacetamide. This leads to a stable modification of cysteins



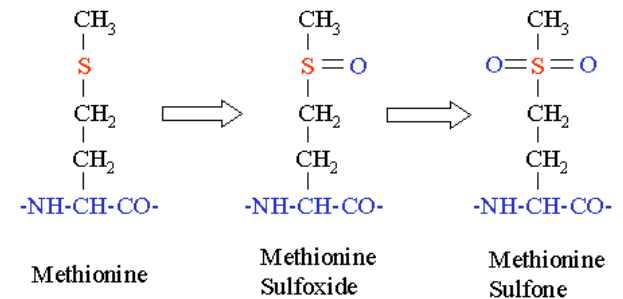
- A fixed modification on amino acid X replaces the original amino acid X during database search

# Modifications

- The modification settings mostly depend on the biochemical assays used for sample preparation

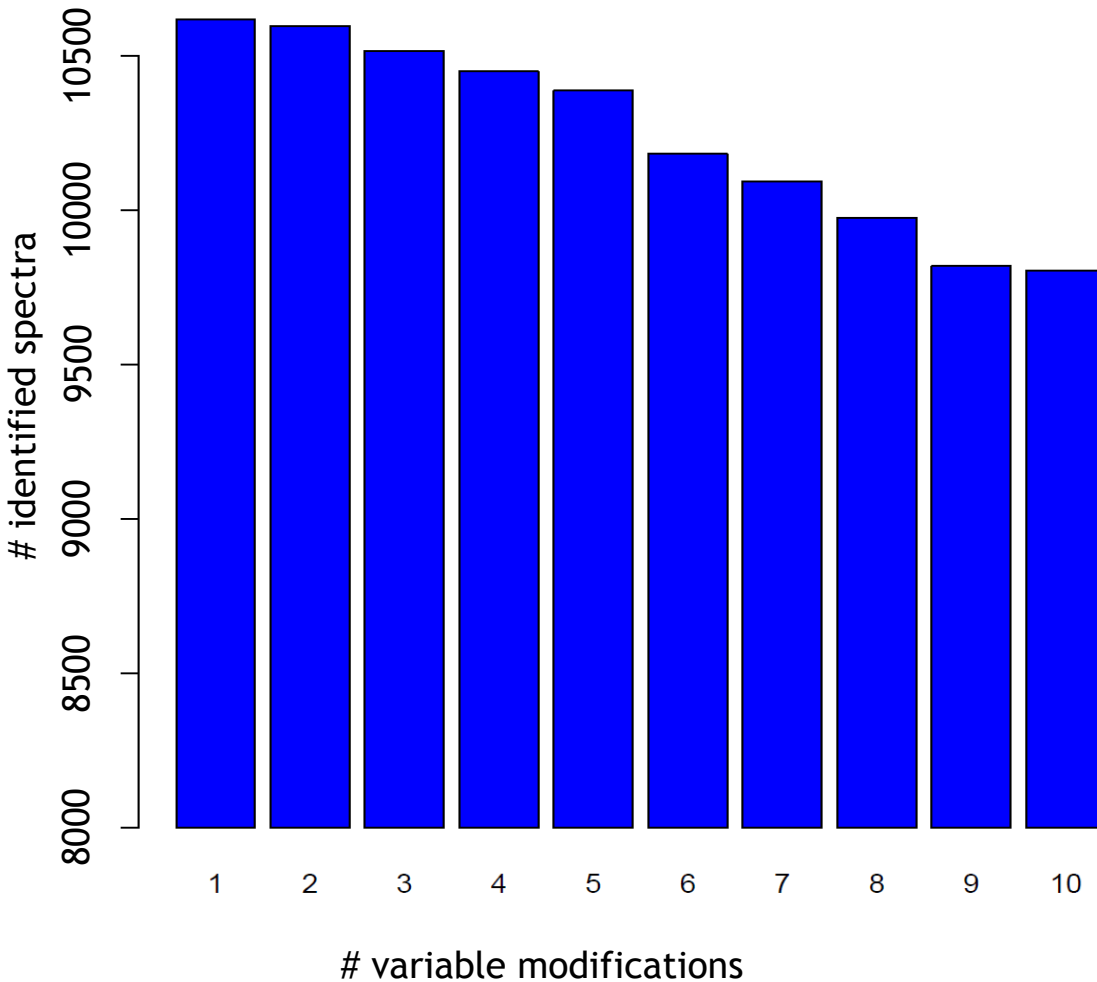
## Variable modifications

- Variable modifications should be set if you know that a subset of the amino acids are modified. Routinely oxidation of methionine should be set as variable modifications. During the electrospray ionization Met residues frequently react with the oxygen in the ionization source environment
- Note that variable modifications are considered as new amino acids and **have significant influence on the search space sizes**



<http://ionsource.com/Card/MetOx/metox.1.gif>

# Variable modifications



## Intuitively...

- More variable modifications should discover more peptides
- Large parts of the proteome are modified

## However...

- More 'amino acids': increase in search space
- Loss in sensitivity
- Variable modifications need to be carefully chosen



# Database settings

- The database should contain all protein sequences that are expected to be in the sample (e.g., all human proteins of your looking at proteomics data from human cell lines)
- From the database and the enzyme of 'cutting rule' settings, the peptide candidates are calculated
- Besides the expected proteins, the database should also contain common contaminants, such as trypsin (or other enzymes), keratins or BSA (bovine serum albumin) that is usually used for instrument calibration
- Databases can also be designed in a way to give an intuitive idea on False discovery rates -> target/decoy databases

# Target-decoy databases

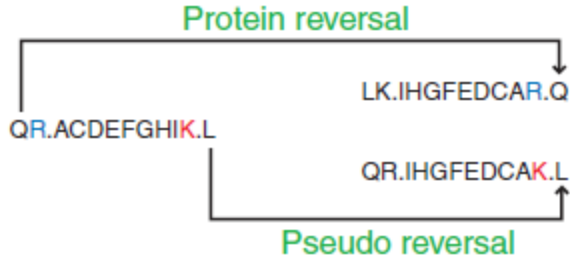
- Take the original protein sequences (target sequences) and reverse, pseudo-reverse, randomize or shuffle these sequences to create decoy sequences
- Either the data is searched twice (first versus the target and then versus the decoy database) or the data is searched once versus a database containing both target and decoy sequences
- The assumption here is that if a decoy peptide is annotated to spectra, the PSM scores can be used to estimate the number of *false* identifications

## Important:

- The decoy database design should provide equal numbers of decoy peptides as there are target peptides per search space (with randomized sequences this is hard to control)
- Ideally one should avoid large overlap between target and decoy peptides

# Target-decoy databases

## Design decoy sequences



### Random

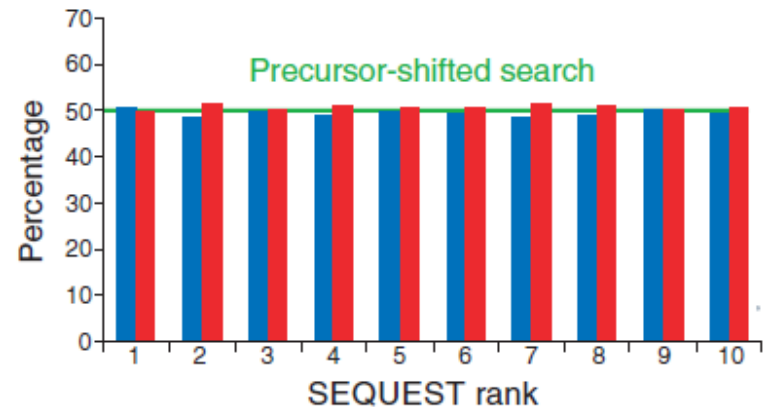
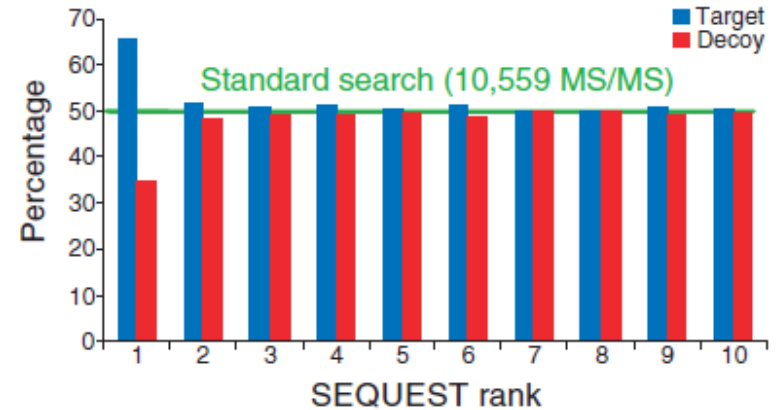
Residue	Frequency
A	0.070
C	0.023
D	0.046
E	0.070
F	0.036

### Markov

Residue	Frequency
A	0.047
C	0.003
D	0.043
E	0.087
F	0.020

[STEV]+

## Separation of target and decoy results



Although different decoy database designs produce very similar results, the most frequently used approaches are the reversed and pseudo-reversed decoy databases

# Calculation of FDRs using target-decoy databases

- General equation for FDR calculation (see statistics lecture)

$$FDR = \frac{FP}{FP+TP}$$

There are two ways how FDRs are calculated based on target-decoy search results:

- Käll et al. suggest (Käll et al., *Proteome Res.* 2008, 7, 29- 34)

$$FDR = \frac{\#decoy}{\#target}$$

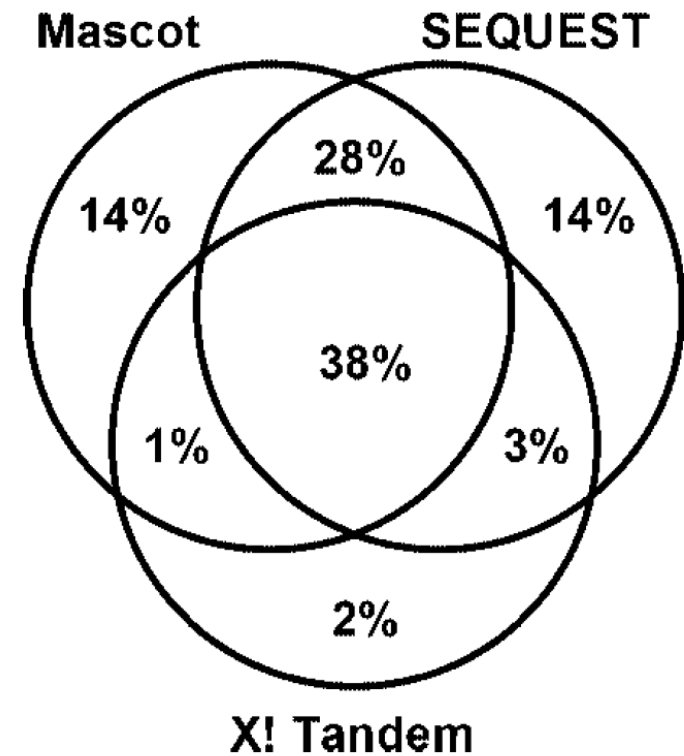
- Zhang et al. suggest (Zhang et al., *J Proteome Res* 2007;6(9):3549-3557)

$$FDR = \frac{2\#decoy}{\#target+\#decoy}$$



- OpenMS::TOPP::FalseDiscoveryRate uses the *Käll* metrics

# Comparison of search engines

- 18 protein mix
- The same dataset was searched with three different search engines
- Identical search parameters



# Multiple search engines

- Majority voting
  - Reliability  sensitivity 

# Multiple search engines

- Majority voting
  - Reliability  $\uparrow$  sensitivity  $\downarrow$
- All peptide IDs
  - Reliability  $\downarrow$  sensitivity  $\uparrow$
- Combine search engine scores:
  1. Scores are inherently different
  2. Different number of peptide candidates

# Multiple search engines

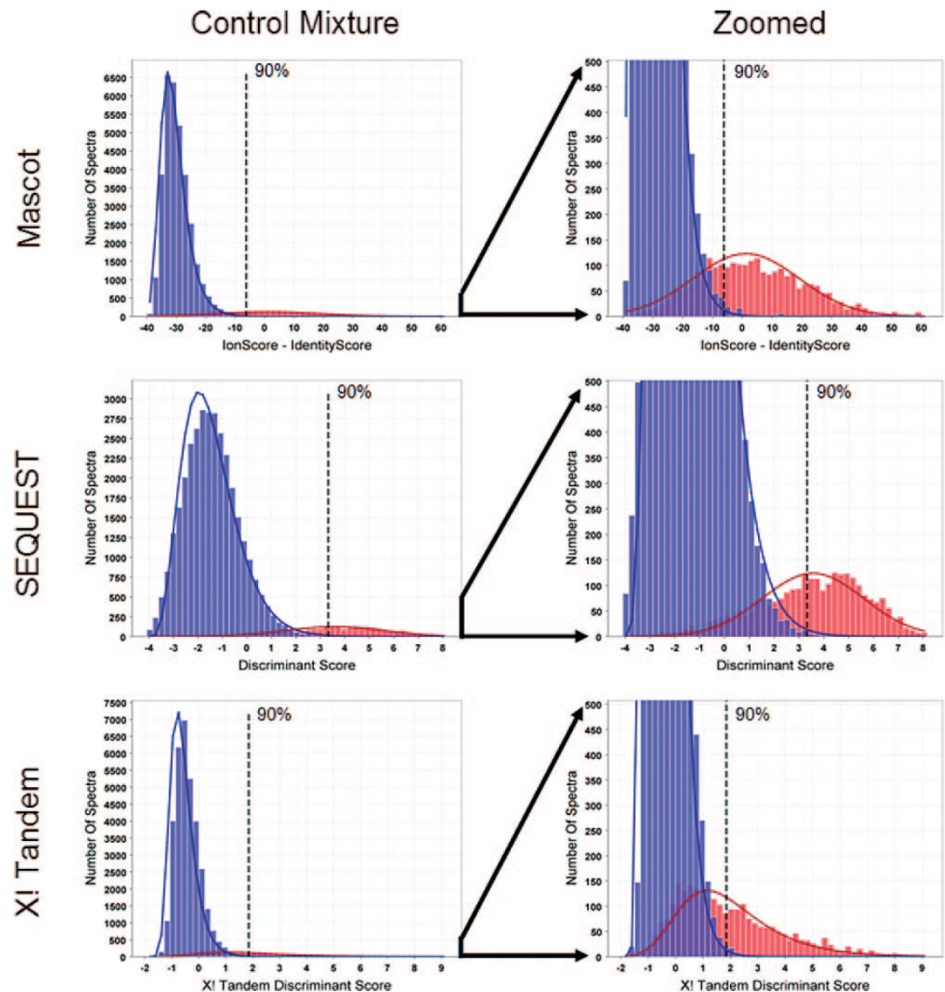
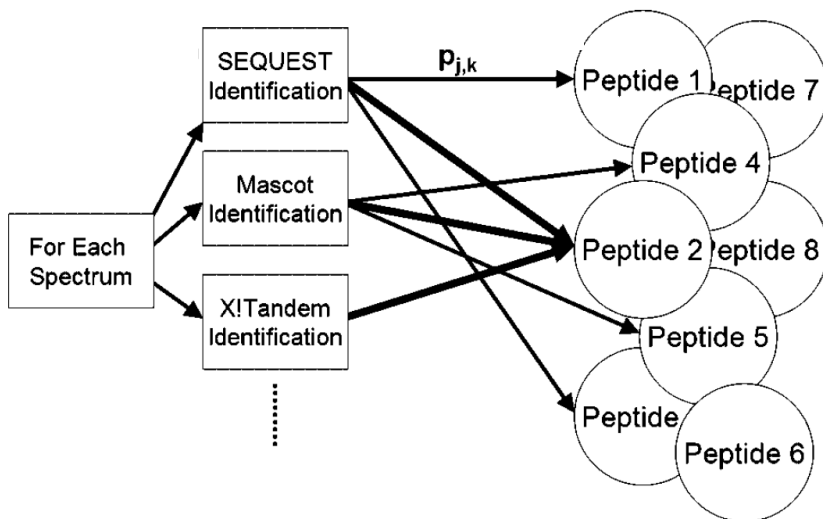
- Majority voting
  - Reliability ↑ sensitivity ↓
- All peptide IDs
  - Reliability ↓ sensitivity ↑
- Combine search engine scores:
  1. Scores are inherently different
  2. Different number of peptide candidates
- *Combination approaches*
  - Scaffold Searle et al., J Proteome Res. 2008, 7, 245-253 245
  - OpenMS::TOPP::ConsensusID Nahsen et al., J Proteome Res. 2011 Aug 5;10(8):3332-43.



# Scaffold

Scaffold integrates search results from Sequest, Mascot and X!Tandem

1. Use mixture models to normalize different scorings to probabilities



# Scaffold

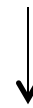
- Calculate agreement score for each PSM across all search engines

$D$  = PSM (Peptide spectrum matching)

$D_{i,j}$  = PSM: spectrum  $i$  to peptide  $j$

$p$  = probabilities for correct assignment (from mixture model)

Probability of correct assignment of peptide  $j$  to spectrum  $i$  by search engine  $k'$



peptide  $j$   
search engine  $k$   
spectrum  $i$

$$A_{i,j,k} = \sum_{k' \neq k} \left\{ \begin{array}{l} p(+|D_{i,j,k'}) < 0.05 \\ 0.05 \leq p(+|D_{i,j,k'}) < 0.5 \\ 0.5 \leq p(+|D_{i,j,k'}) \end{array} \right\} \begin{array}{l} 0.0 \\ 0.5 \\ 1.0 \end{array}$$

Conditional probability for A assuming a correct assignment

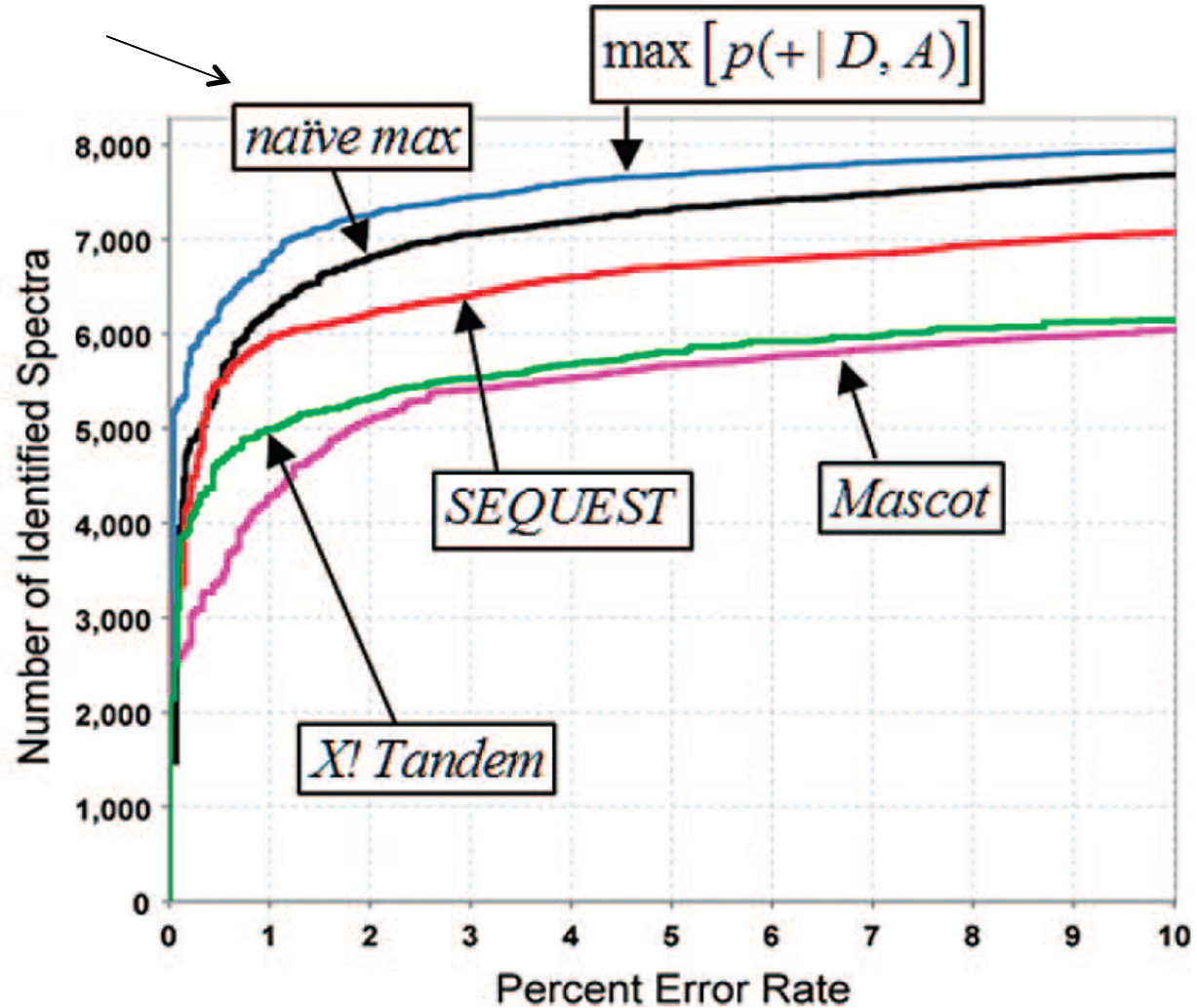


Conditional probability for being correct given a PSM  $D$



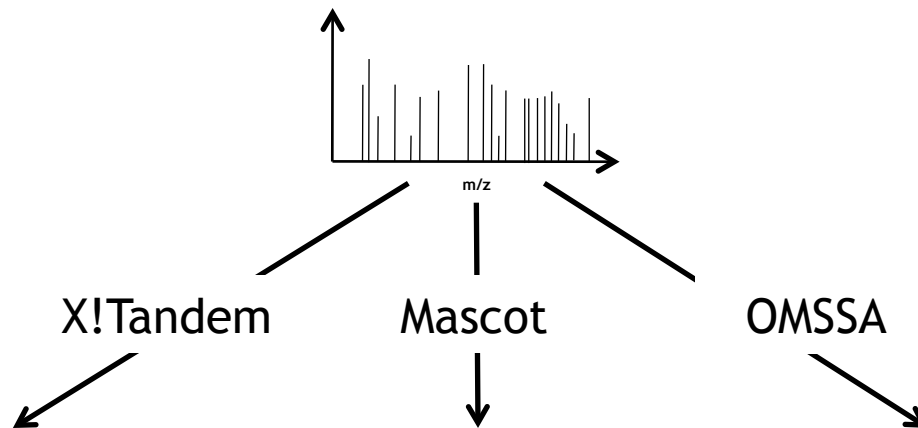
$$p(+|D, A) = \frac{p(A|+)p(+|D)}{p(A|+)p(+|D) + p(A|-)p(-|D)}$$

# Scaffold performance



# ConsensusID

ConsensusID integrates search results from OMSSA, Mascot and X!Tandem



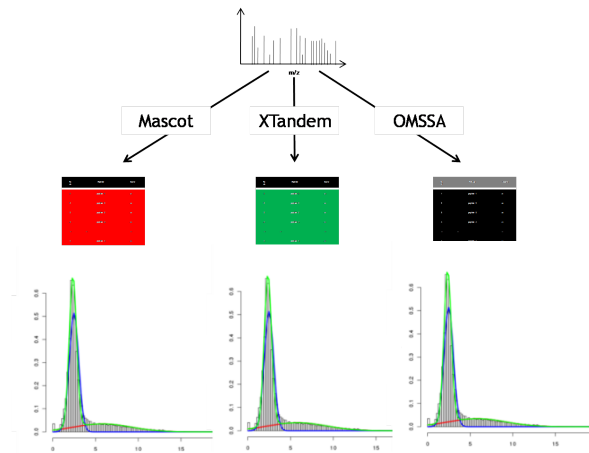
Rank	Peptide	Score
1	QRESTATDILQK	0.008

Rank	Peptide	Score
1	EIEEDSLEGLKK	14.78
2	GIEDDLMDLIKK	12.63
3	ISCAEGALEALKK	10.2

Rank	Peptide	Score
1	AELASCVVGDLGAK	1.2
2	ELM(Ox)SNGPGSIIGAK	1.2
3	ISCAEGALEALKK	4
4	QRESTATDILQK	10

1. Use mixture models to normalize different scorings to probabilities

# ConsensusID – mixture modeling



Rank	Peptide	Score
1	QRESTATDILQK	0.54

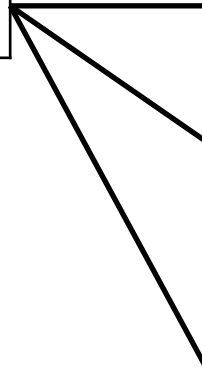
Rank	Peptide	Score
1	EIEEDSLEGLKK	0.96
2	GIEDDLMDLIKK	0.98
3	ISCAEGALEALKK	0.98

Rank	Peptide	Score
1	AELASCVVDLGAK	0.94
2	ELM(Ox)SNGPGSIIGAK	0.97
3	ISCAEGALEALKK	0.99
4	QRESTATDILQK	0.99

# ConsensusID – similarity scoring

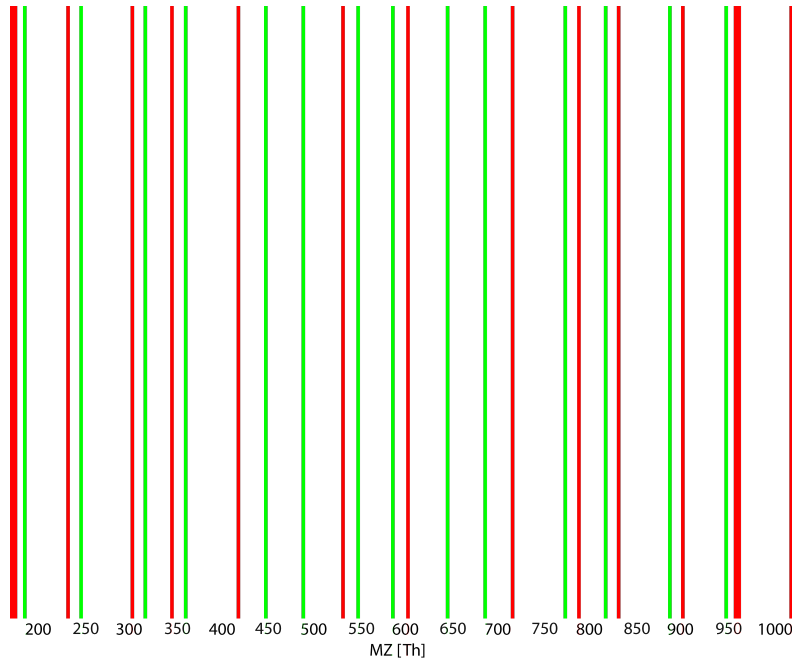
Rank	Peptide	Score
1	QRESTATDILQK	0.54

Rank	Peptide	Score
1	EIEEDSLEGLKK	0.96
2	IGIEDDLMDLIKK	0.98
3	ISCAEGALEALKK	0.98



# ConsensusID - similarity scoring

Rank	Peptide	Score
1	QRESTATDILQK	0.54



Rank	Peptide	Score
1	EIEEDSLEGLKK	0.96
2	IGIEDDLMDLIKK	0.98
3	ISCAEGALEALKK	0.98

47%

42%

21%

QRESTATDILQK	similarity $*s_2(p_1)$
--------------	------------------------

# ConsensusID - consensus score

- Score for every sequence from any engine

Rank	Peptide	Score
1	QRESTATDILQK	0.54
2	EIEEDSLEGLKK	$S_{1,2}$
3	GIEDDLMDLIKK	$S_{1,3}$
4	ISCAEGALEALKK	$S_{1,4}$
5	AELASCVVDLGAK	$S_{1,5}$
6	ELM(Ox)SNGPGSIIGAK	$S_{1,6}$

Rank	Peptide	Score
1	EIEEDSLEGLKK	0.96
2	GIEDDLMDLIKK	0.98
3	ISCAEGALEALKK	0.98
4	QRESTATDILQK	$S_{2,4}$
5	AELASCVVDLGAK	$S_{2,5}$
6	ELM(Ox)SNGPGSIIGAK	$S_{2,6}$

Rank	Peptide	Score
1	AELASCVVDLGAK	0.94
2	ELM(Ox)SNGPGSIIGAK	0.97
3	ISCAEGALEALKK	0.99
4	QRESTATDILQK	0.99
5	EIEEDSLEGLKK	$S_{3,5}$
6	GIEDDLMDLIKK	$S_{3,6}$

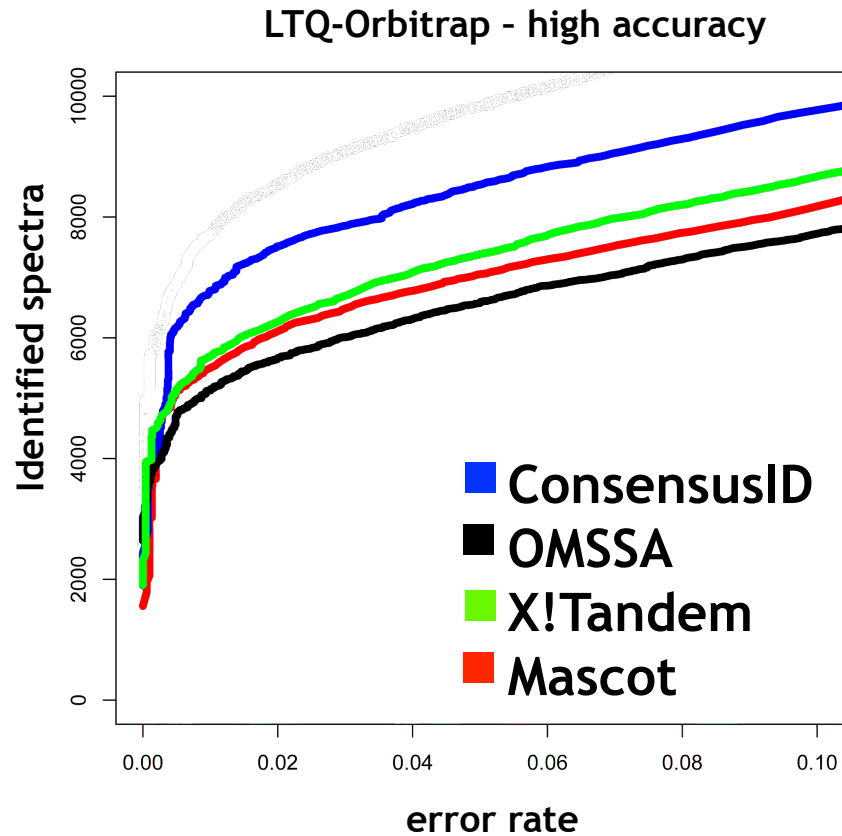
- Combination of scores

$$\text{ConsensusID } (p_1) = \frac{s_1(p_1) + \alpha s_2(p_i) + \beta s_3(p_j)}{(1 + \alpha + \beta)^2}$$

$$\text{ConsensusID } (\text{QRESTATDILQK}) = \frac{0.54 + 0.3 \cdot 0.96 + 1 \cdot 0.99}{(1 + 0.3 + 1)^2} = \mathbf{0.34}$$



# ConsensusID performance



error rates = *false discovery rates*

# References

- Eidhammer et al., Computational Methods for Mass Spectrometry Proteomics. Wiley. 2007.
- Freitas and Xu, BMC Bioinformatics. 2010, 11:436
- Roepstorff and Fohlman, Biological Mass Spectrometry, Volume 11, Issue 11, page 601, November 1984
- Steen and Mann. Nature Reviews, Molecular Cell Biology, Vol. 5 2004
- Johnson et al. Anal. Chem. 1987;59:2621-2625
- Hoffert J D et al. PNAS 2006;103:7159-7164
- Craig,R. and Beavis,R.C. (2003) Rapid Commun. Mass Spectrom., 17, 2310–2316
- Geer et al. (2004) J Proteome Res. 2004 Sep-Oct;3(5):958-64.
- Eng et al., *J. Am. Soc. Mass Spectrom.* 1994, 5, 976-989.
- Fenyö and Beavis, Anal. Chem.2003, 75, 768-774
- [http://www.proteomesoftware.com/pdf\\_files/XTandem\\_edited.pdf](http://www.proteomesoftware.com/pdf_files/XTandem_edited.pdf)
- Grenzel et al, Proteomics. 2003(3):1597-1610.
- Elias and Gygi, Nature Methods. Vol. 4, No. 3, March 2007
- Searle et al., Journal of Proteome Research. 2008, 7, 245–253 **245**
- Nahnsen et al., J Proteome Res. 2011 Aug 5;10(8):3332-43