

# COMPUTATIONAL PROTEOMICS AND METABOLOMICS

*Oliver Kohlbacher, Sven Nahnsen, Knut Reinert*

## *7. Peptide Identification I – Database Search*



# LEARNING UNIT 7A

## PEPTIDE DATABASE SEARCH

- Peptide fragmentation
- Database search concepts

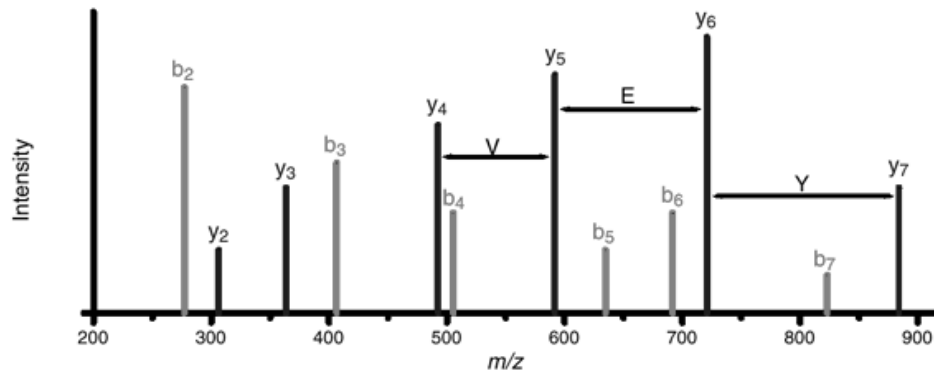
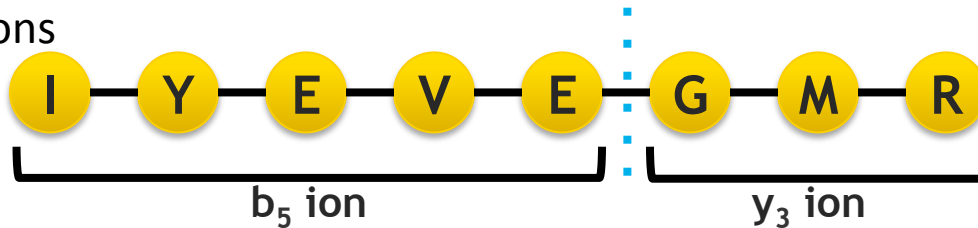
This work is licensed under a Creative Commons Attribution 4.0 International License.



# Peptide Identification

Why can we identify peptides from tandem MS spectra?

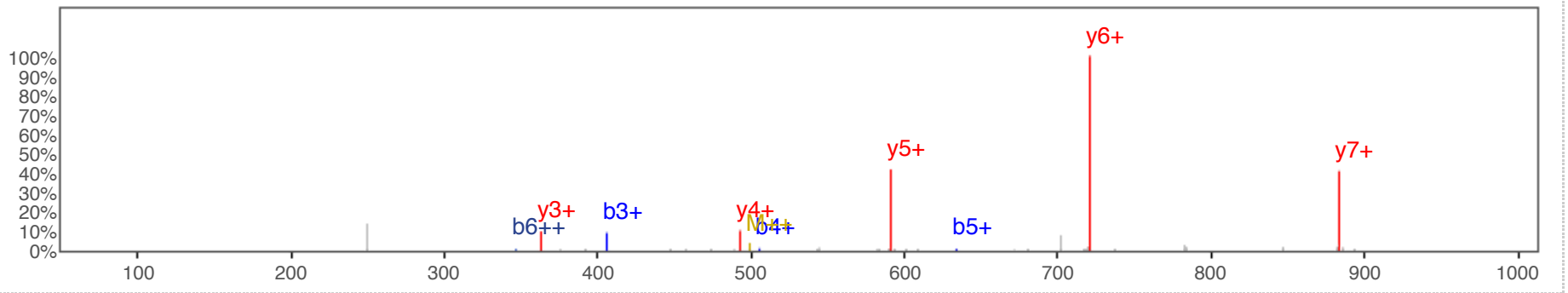
- **Goal: identify sequence**
- Tandem MS
  - Sequence consists of the **same 20 building blocks** (amino acids)
  - CID: peptide breaks preferentially along the **backbone**
  - Peptide **fragment ions correspond to prefixes and suffixes** of the whole peptide sequences
  - Complete ion series (ladders) reveal the sequence via mass differences of adjacent fragment ions



# Peptide Identification

- **Issues**

- Spectra are incomplete – ions are missing
- Missing information makes it very hard to reconstruct full sequence



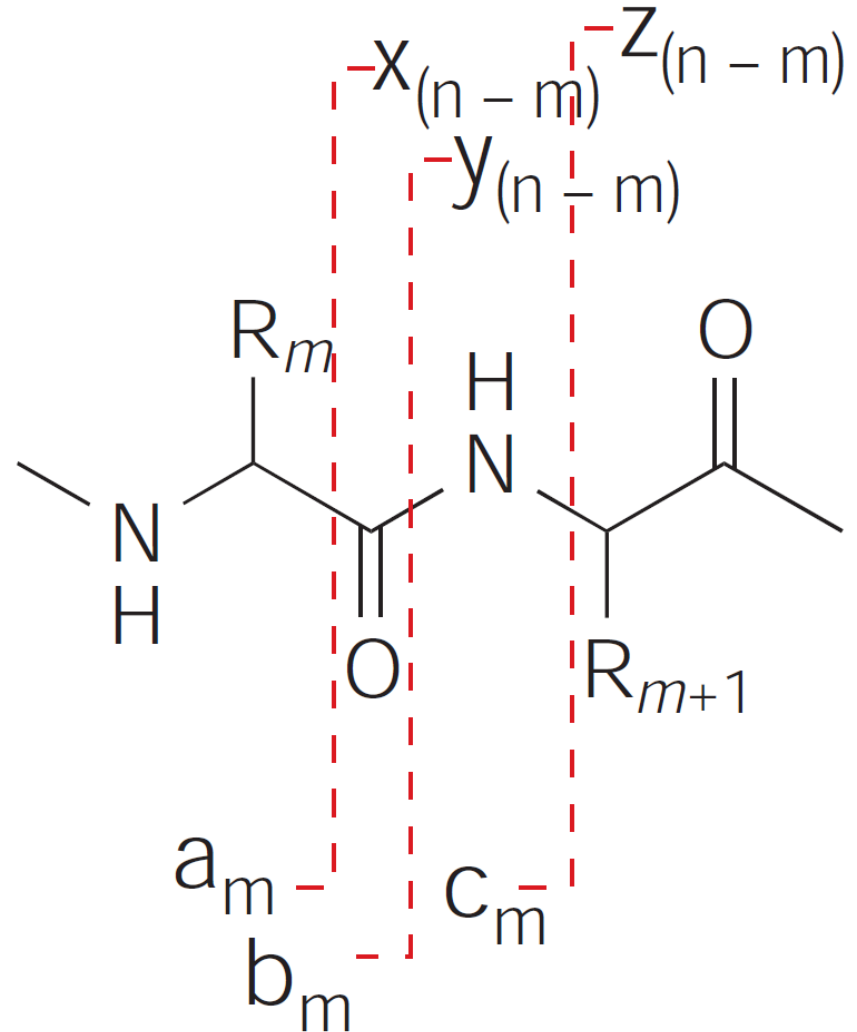
- **Database search**

- Not all sequences occur in a proteome – only a fraction of sequence space is used
- Try to find those sequences that match the ions present in the spectrum

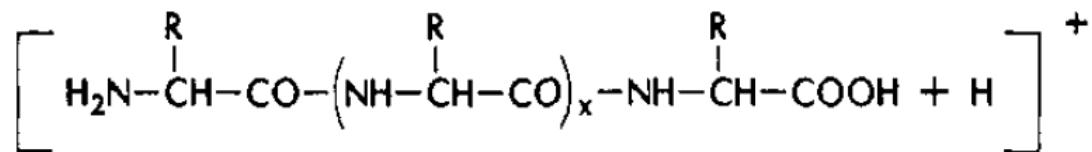
# Product ion generation

- 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

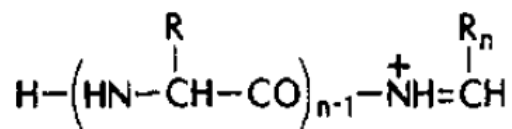
(Roepstorff and Fohlman, *Biological Mass Spectrometry*, Volume 11, Issue 11, page 601, November 1984)



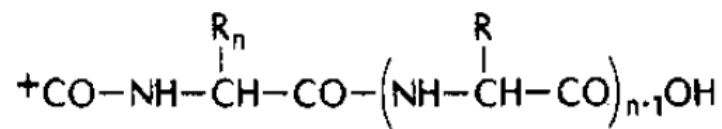
# Ion Series



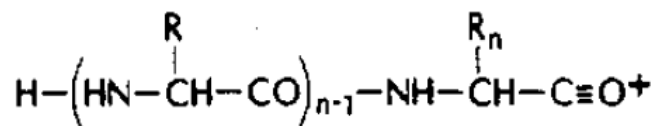
I



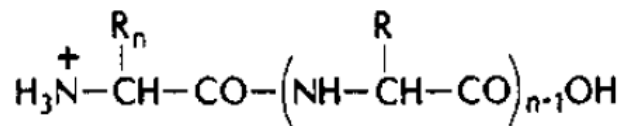
$a_n$



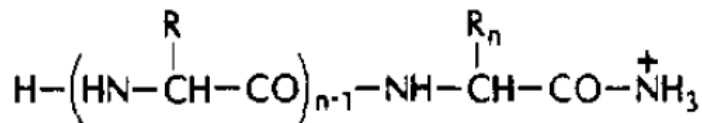
$x_n$



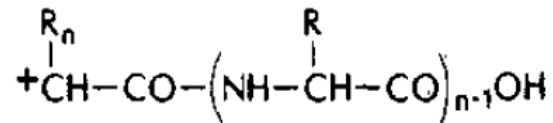
$b_n$



$y_n$



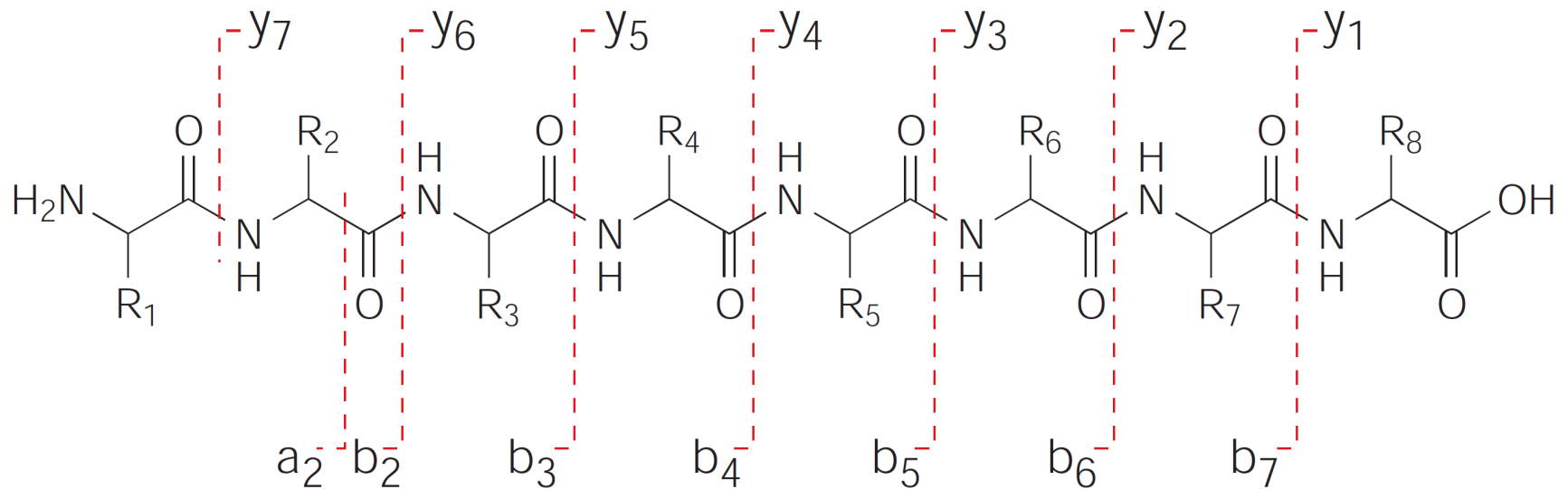
$c_n$



$z_n$

# b/y ions in CID

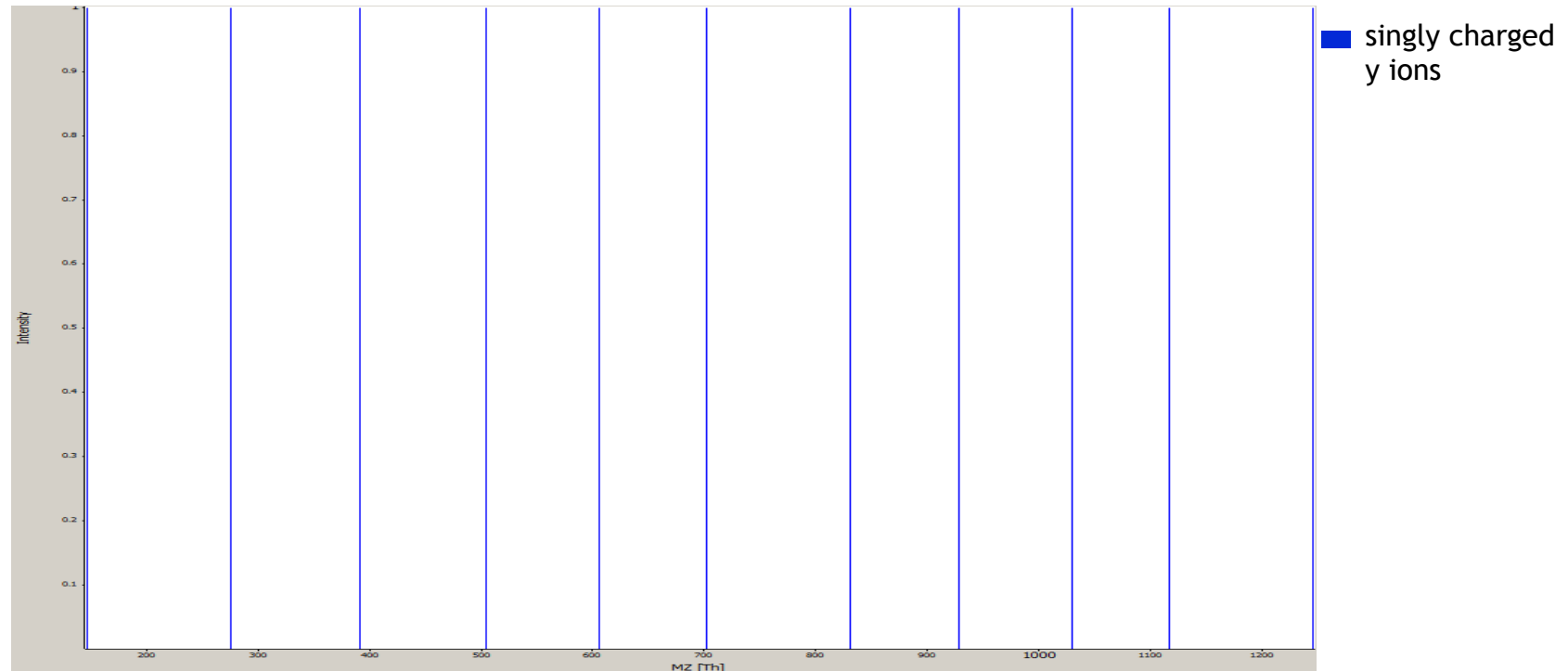
CID fragmentation predominately produces b and y ions



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

# Ion Series - Example

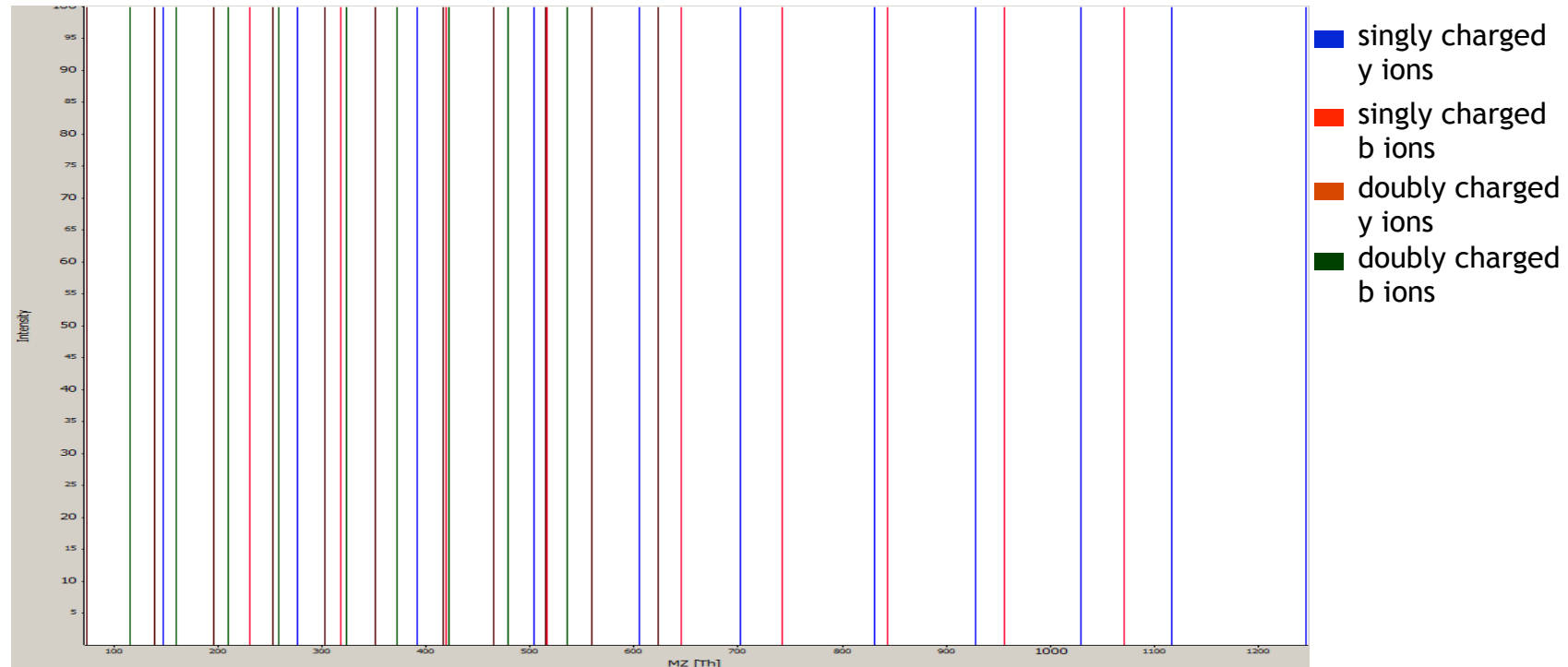
- For simplicity we will consider theoretical spectra for the artificial (tryptic) peptide TESTPEPTIDEK
- For singly charged ion fragments, only one of the sister fragments will be observed





# Ion Series - Example

- If the same peptide was multiply charged, the charges are usually distributed across the product ions
- Tandem spectrum then usually contains both sister ions and also doubly charged product ions



# Ion Series - Example

- Theoretically, one can also observe a, c, x and z ions



# Ion Series - Example

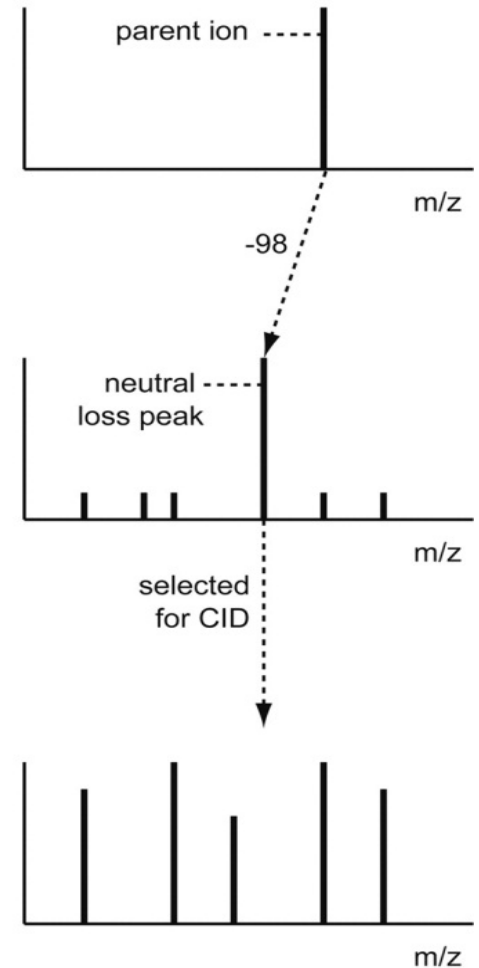
- Theoretically, one also observes a, c, x and z ions
- abc and xyz ions are called backbone ions.

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



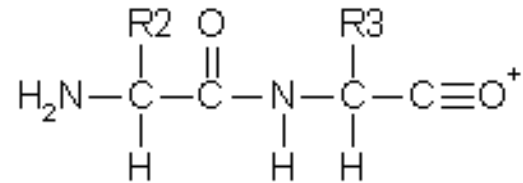
# Neutral Losses

- 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
  - ammonia loss** ( $\text{NH}_3$ : - 17.027 Da) on R, K, N and Q
  - 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  $\text{mass}_{\text{ion}} - \text{mass}_{\text{neutral}}$
- The problem of very intense ions resulting from neutral losses of precursor ions can be overcome by triggering an additional fragmentation

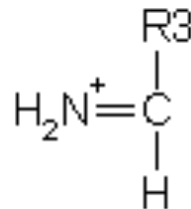


# Internal Fragments

- **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

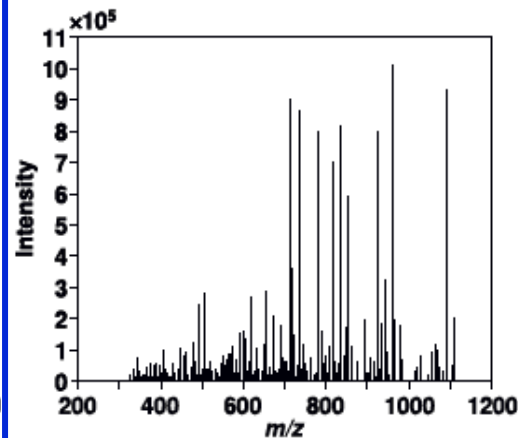
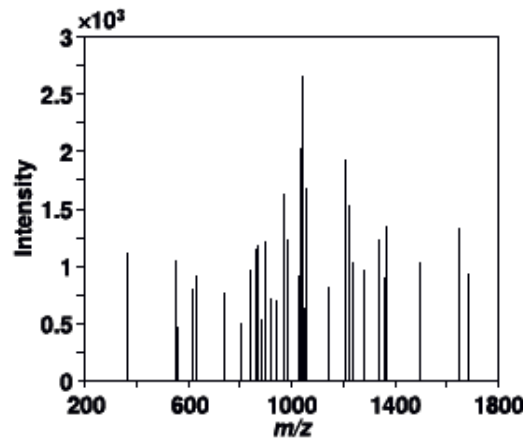


- **Immonium ions** are a special case of internal fragments. They are composed of a single side chain formed by a combination of *a*-type and *y*-type fragmentation

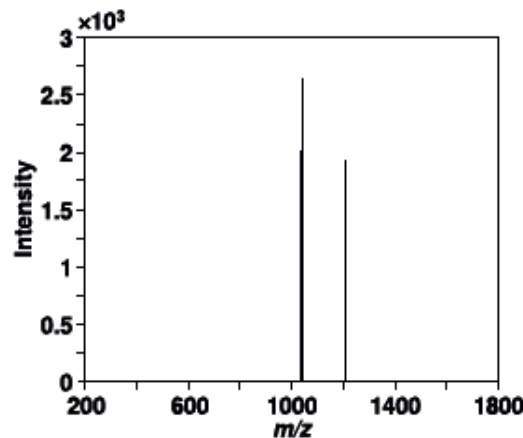


# Noise in Tandem Spectra

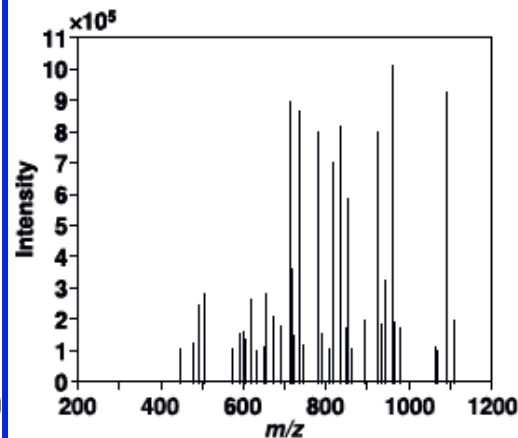
- In addition to the various types of ions, there is also noise in tandem spectra



*With noise*



*Blank run*



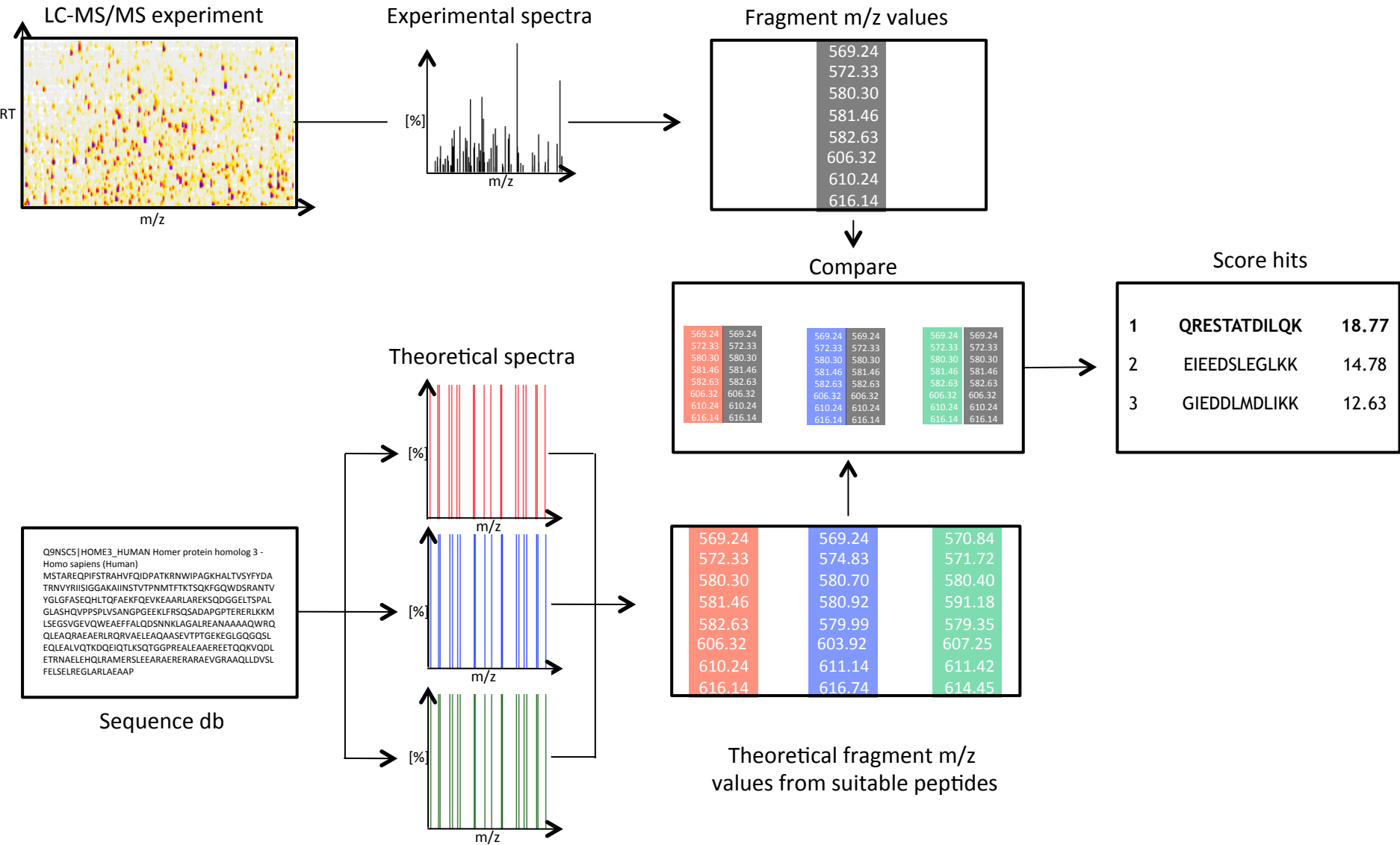
*Without noise*

*One isolated peptide*

# Ion Types – Summary

- 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  $b^{-H_2O}$ ,  $y^{-H_2O}$

# Database Search – Overview

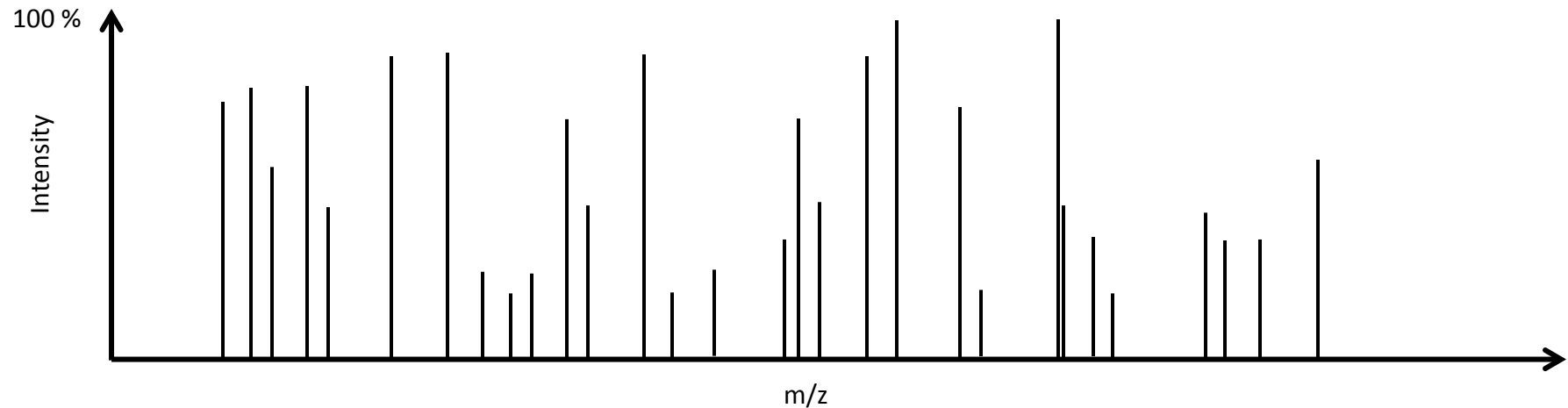




# Database Search – Key Steps

1. **Extract all sequence candidates** (usually tryptic) from the database matching the precursor mass of the MS<sup>2</sup> spectrum with a given error tolerance
2. **Generate theoretical spectrum** for each of the candidate sequences
3. **Align** the theoretical spectra to the experimental spectrum
4. **Score** the alignment
5. **Report all peptide-spectrum matches** above a certain score threshold

# Step 1. Generate Candidates



- 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

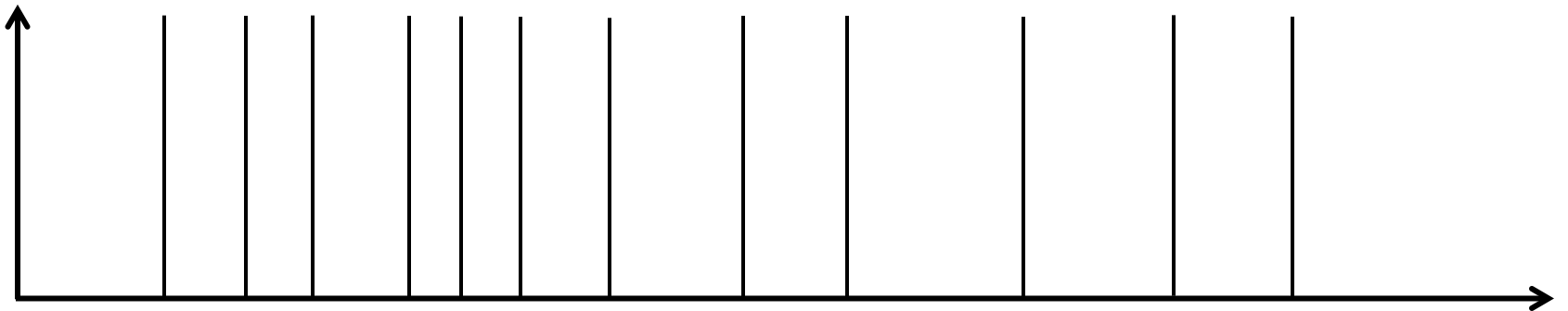
$$|m_{prec} - m_{cand}| \leq d$$

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

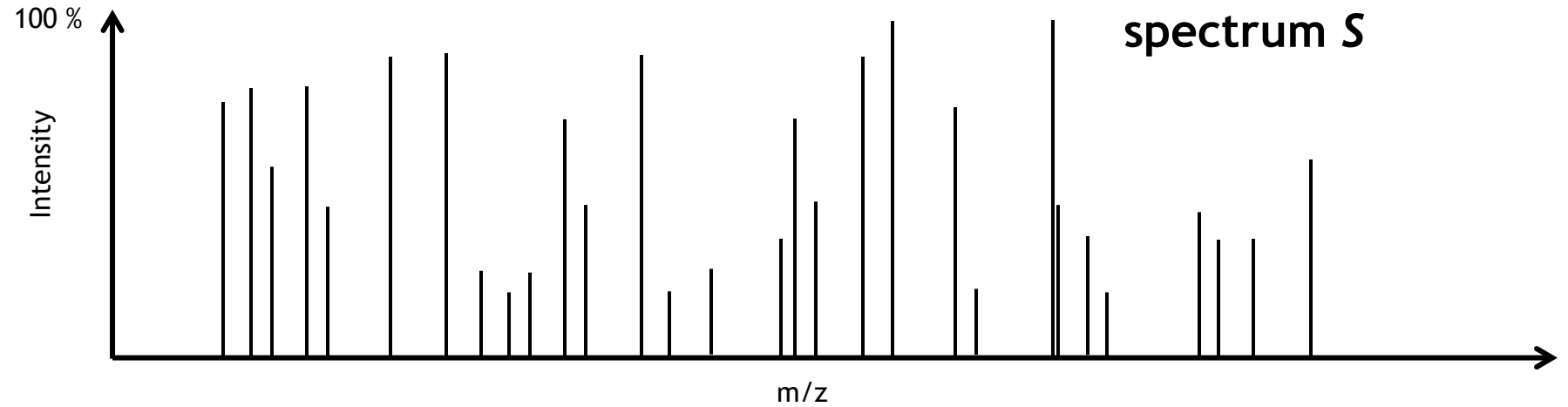
$$\Omega_S$$

## Step 2: Generate Theoretical Spectra

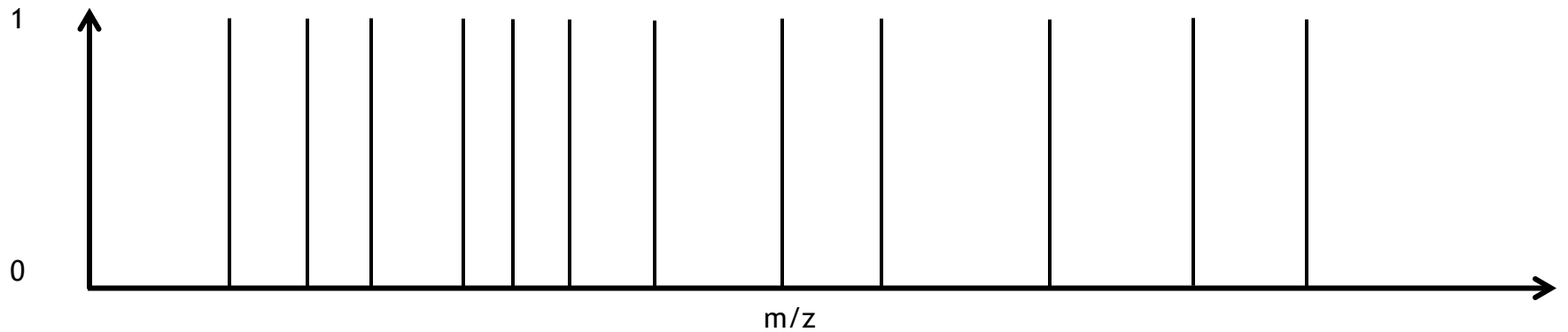
- 1<sup>st</sup> option: extract all masses from the MS<sup>2</sup> spectrum
- 2<sup>nd</sup> option: try to model fragment ion intensities



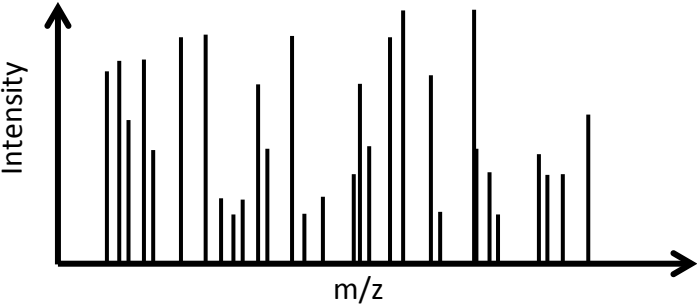
# Step 3. Align Spectra



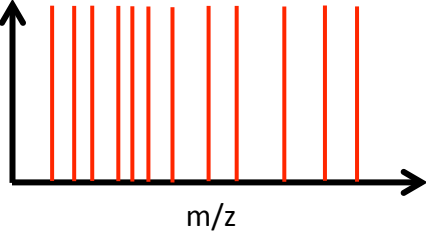
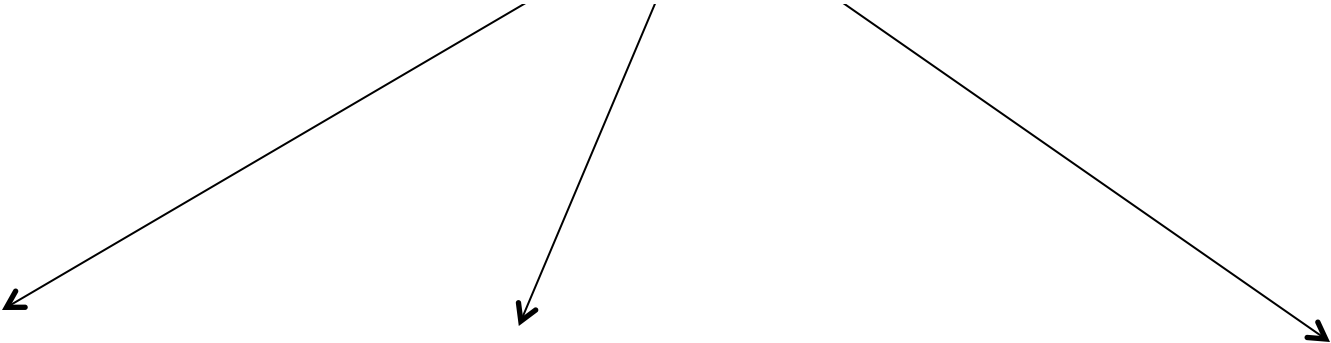
Theoretical spectrum  $T$ , generated from a sequence  $p_i \in \Omega_S$



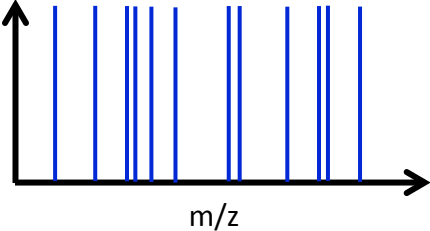
# Step 3: Align Spectra



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

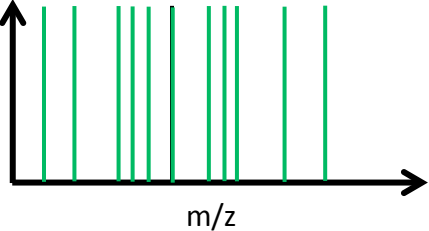


$p_1 \in \Omega_S$



$p_2 \in \Omega_S$

...



$p_n \in \Omega_S$

## Step 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

Discussed  
in detail

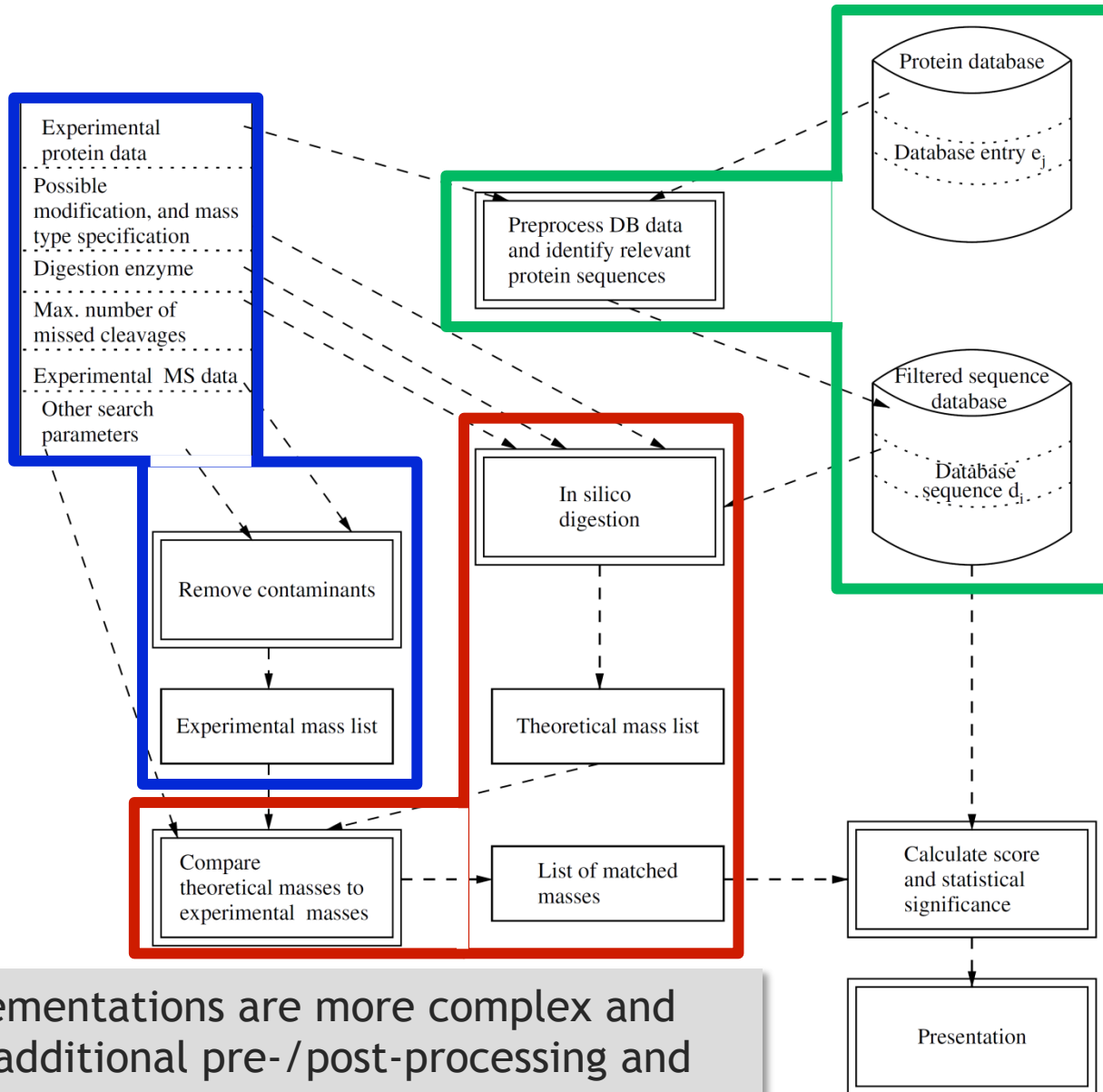
- **X!Tandem**, Craig,R. and Beavis,R.C. (2003) *Rapid Commun. Mass Spectrom.*, 17, 2310–2316

Drafted

- **Sequest** Eng et al., *J. Am. Soc. Mass Spectrom.* **1994**, 5, 976-989.

# More Complex Workflow

Experimental parameters →



← DB settings

Search engine →

Real-life implementations are more complex and often require additional pre-/post-processing and depend on a large number of parameters

# LEARNING UNIT 7B

## SEARCH ENGINES

- X!Tandem
- Sequest
- Other search engines





# Database Search Engines

- Dozens of different database search tools are currently being used
- Common to these tools are the fundamentals describe in Learning Unit 7A
- Tools differ with respect to
  - Spectrum pre-processing
  - Scoring of peptide-spectrum matches
  - Post-processing of peptide-spectrum matches
  - Score statistics
  - Speed
- Results for the same dataset will differ between search engines!

# X!Tandem

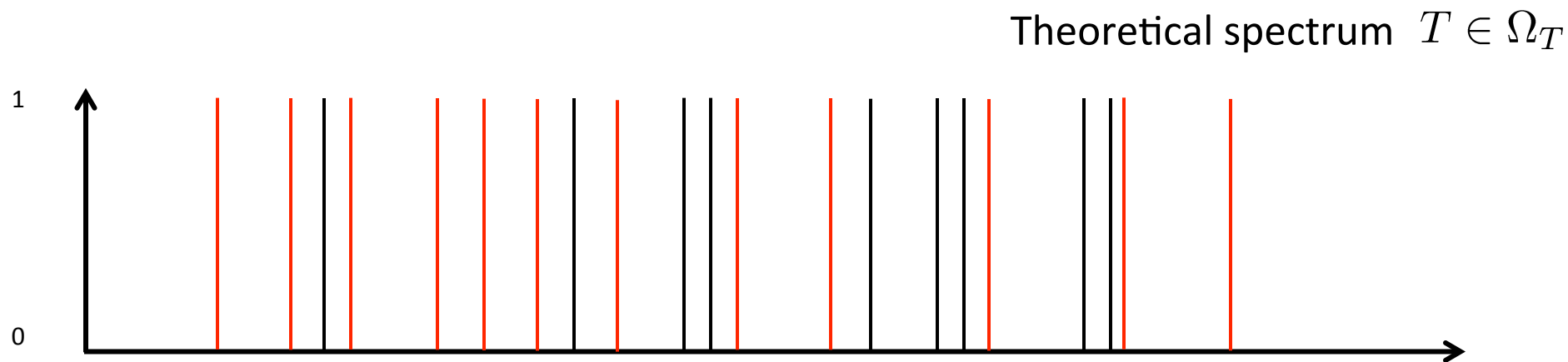
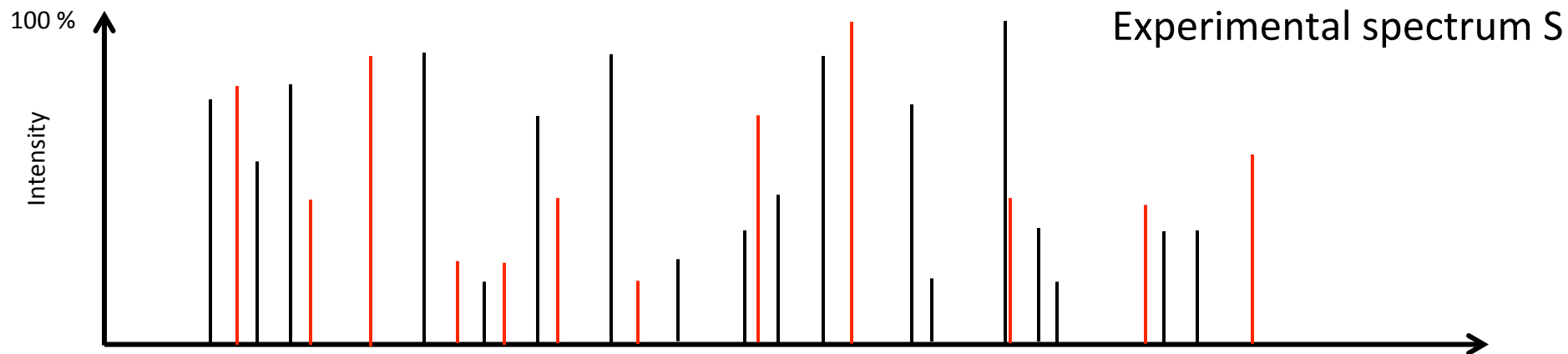
- X!Tandem
  - is a popular open-source database search engine
  - is fast
  - has been published in various versions including multiple refinements to the core algorithm sketched here (latest version: X!Tandem Sledgehammer, 2013)

## Original reference

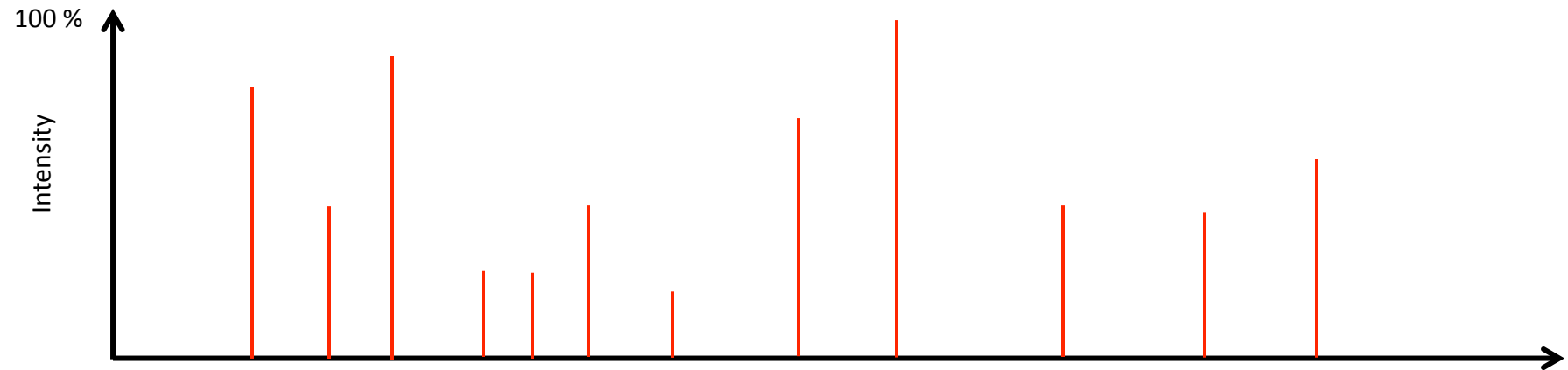
- Craig,R. and Beavis,R.C. (2003) *Rapid Commun. Mass Spectrom.*, **17**, 2310–2316.
- <http://www.thegpm.org/tandem/instructions.html>

# 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)

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

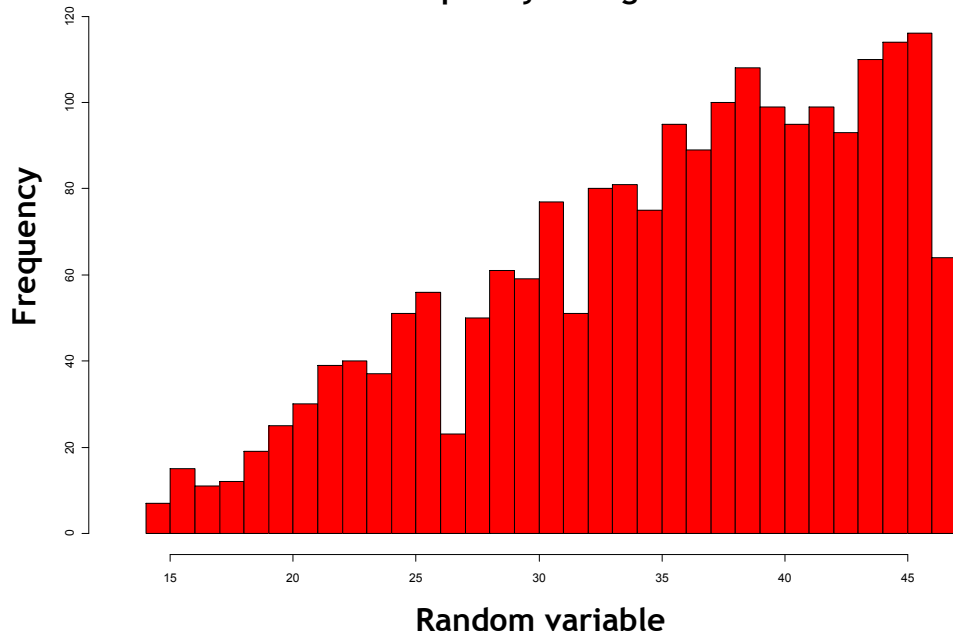
*Intensities from experimental spectrum*  
 $I_i$  ... fragment ion intensities

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

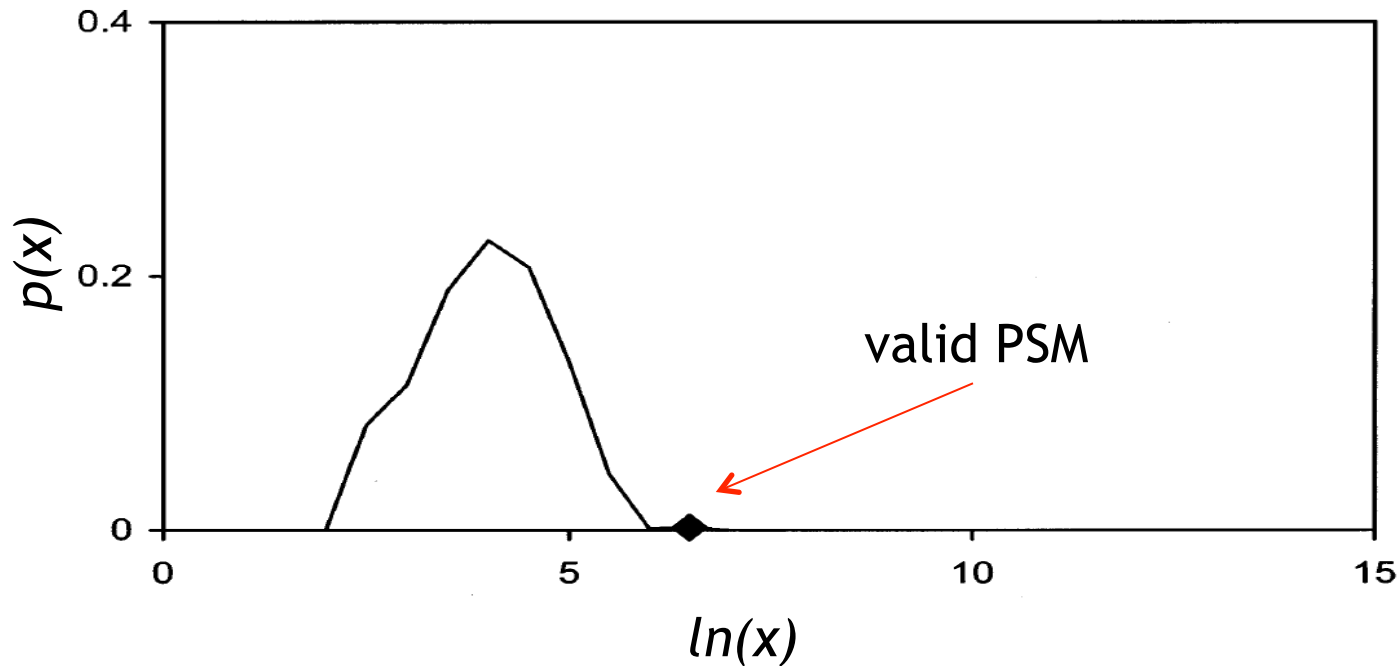
# Survival function and e-value

- 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 histograms (counts of PSMs per score bin).
- With  $f(x)$ , the number of PSMs that are given the score  $x$ ,  $p(x)$  is calculated as  $p(x) = f(x)/N$  with  $N$  being the total number of PSMs

Example of a  
frequency histogram



# Survival function and e-value

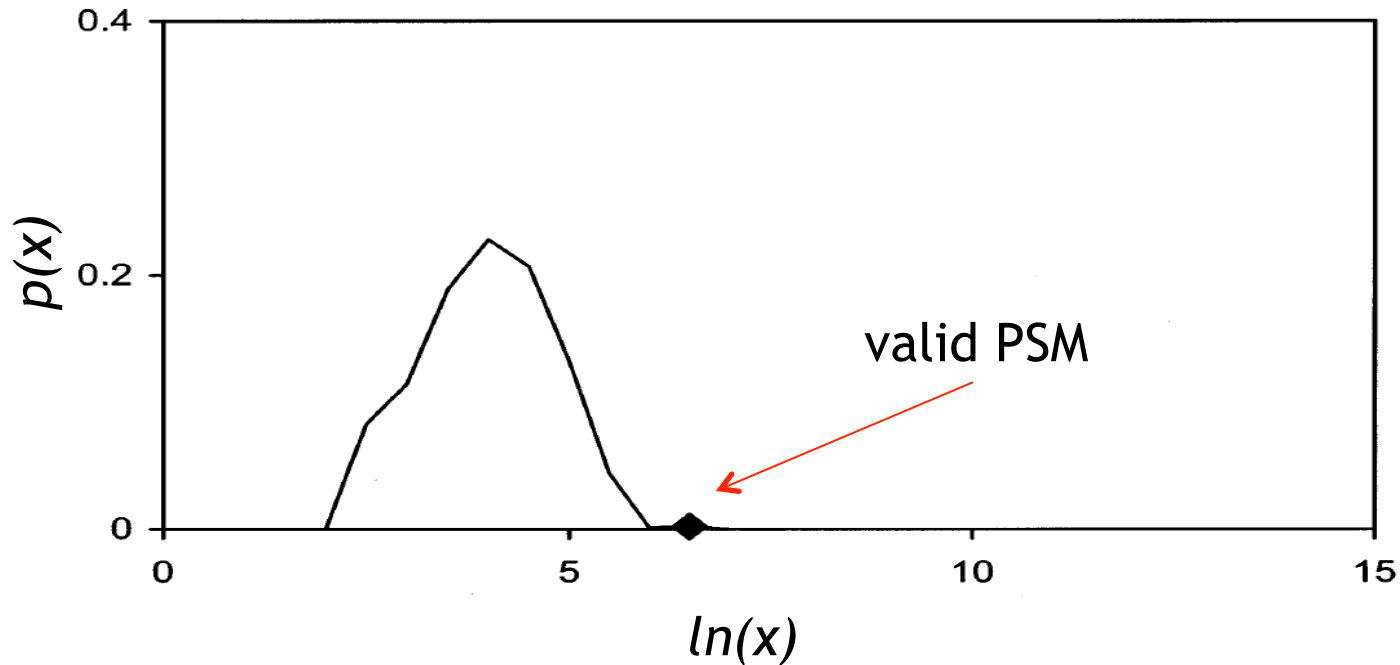


- The survival function,  $s(x)$ , for a discrete stochastic score probability distribution,  $p(x)$  is defined as

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

where  $P(X > x)$  is the probability to have a greater value than  $x$  by random matches in a database.

# Survival function and e-value



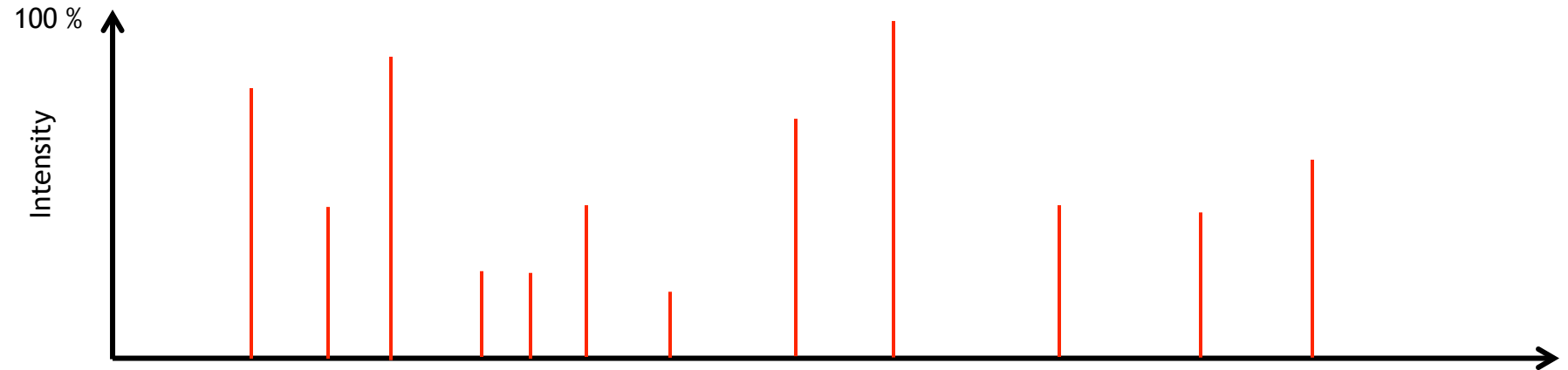
- With the survival function  $s(x)$ , we can calculate the E-value  $e(x)$ , indicating the number of PSMs that are expected to have scores of  $x$  or better

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

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

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

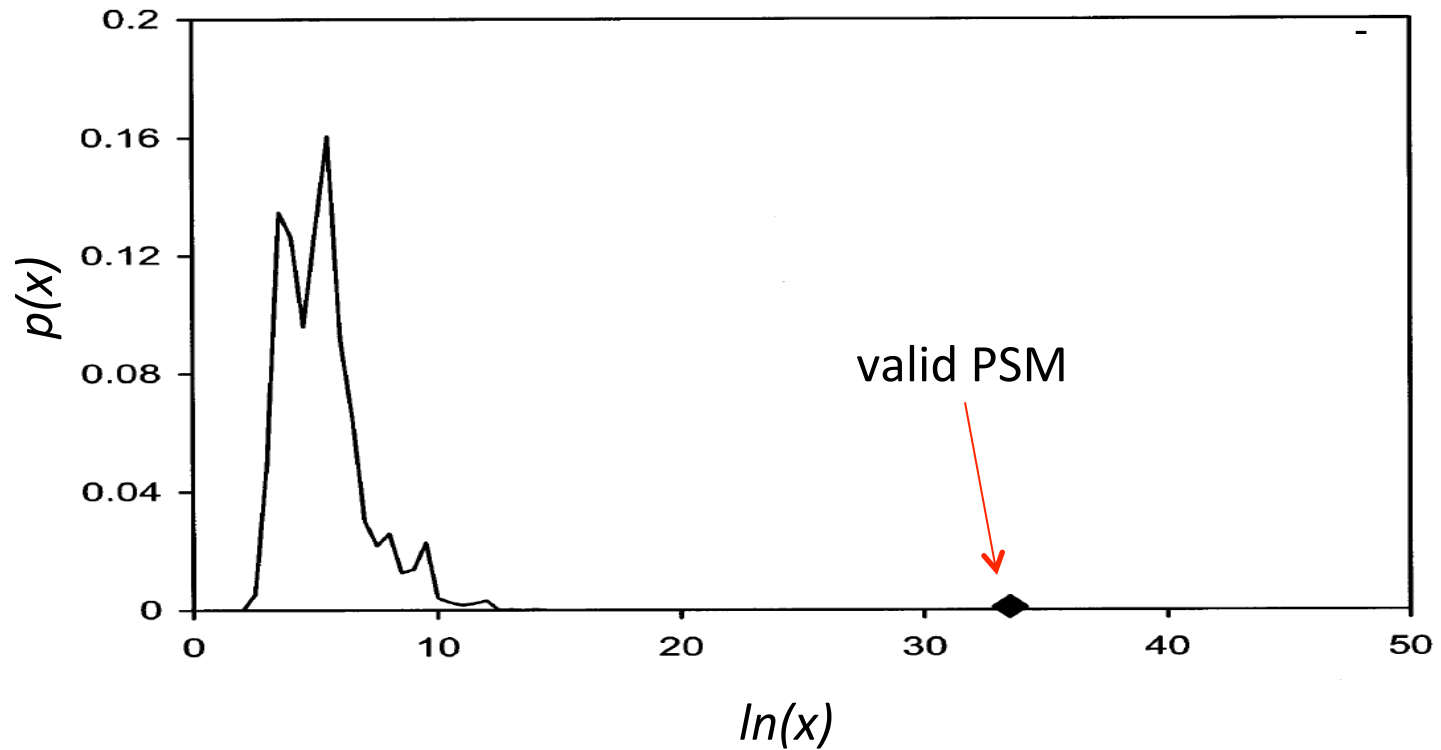
# X!Tandem Hyperscore



- 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

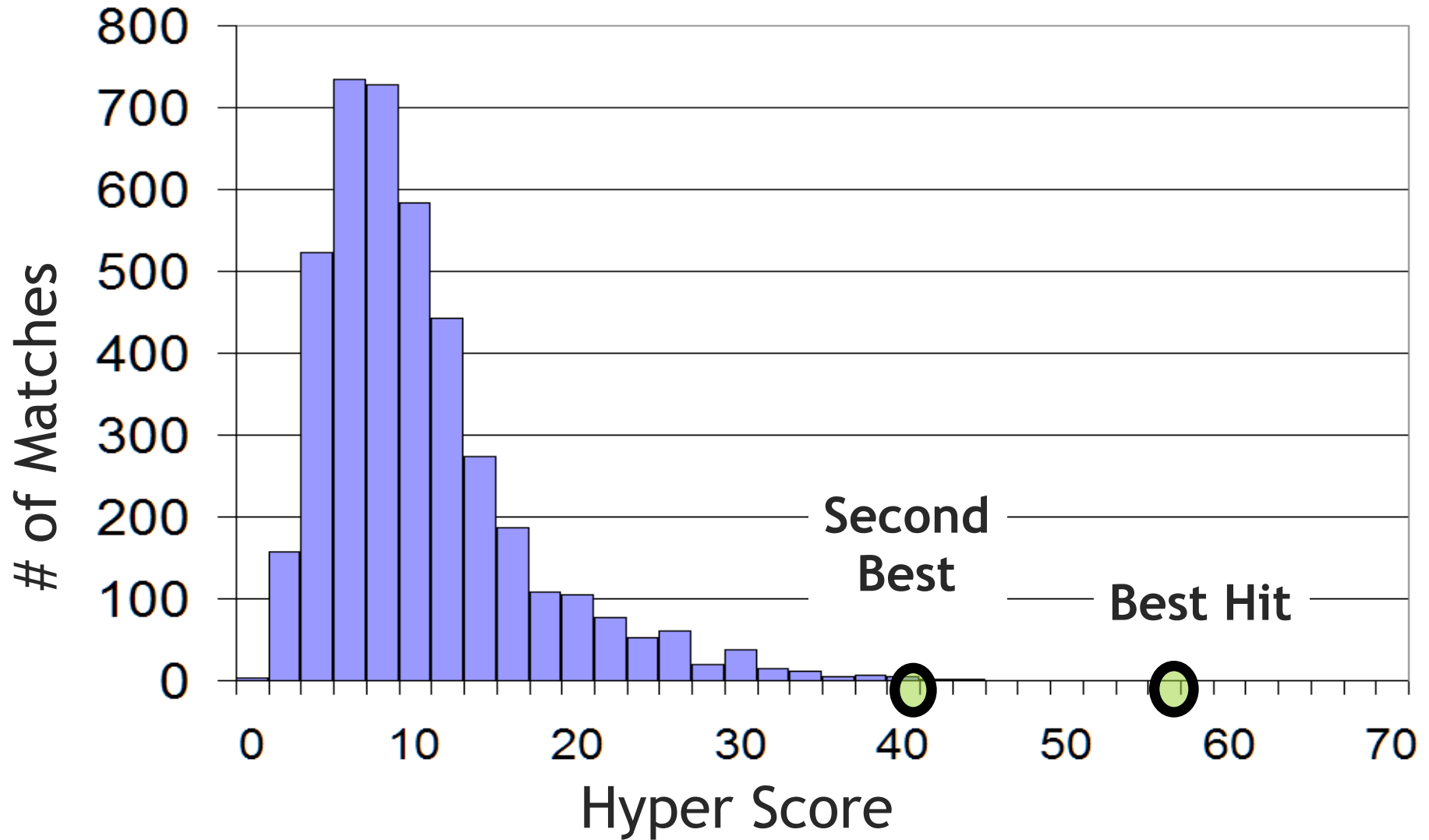


# X!Tandem Hyperscore

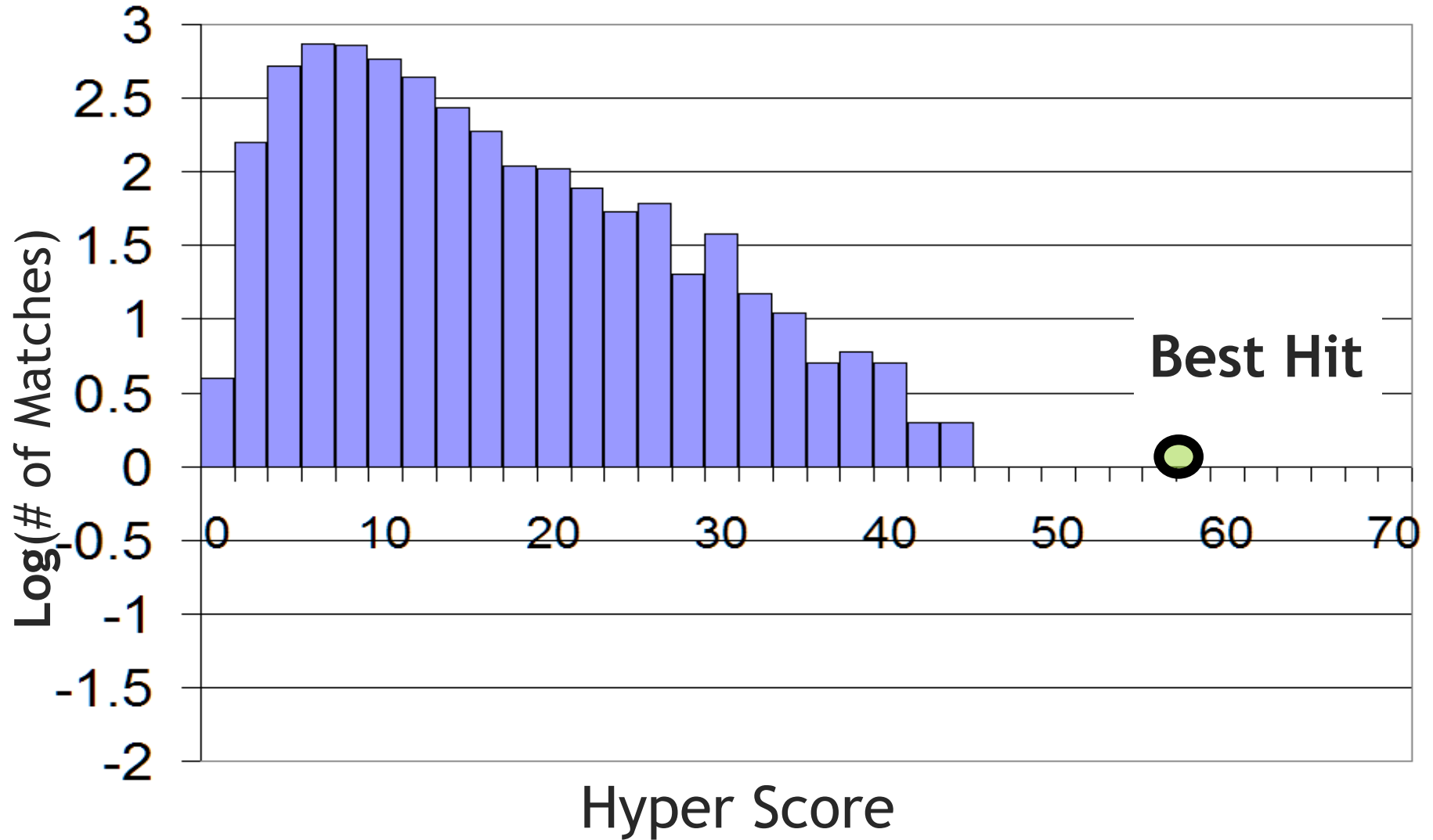


- If  $p(x)$  is now plotted as a function of their  $\log(\text{hyperscores})$ , the valid PSM is much better separated from the bulk of incorrect assignments

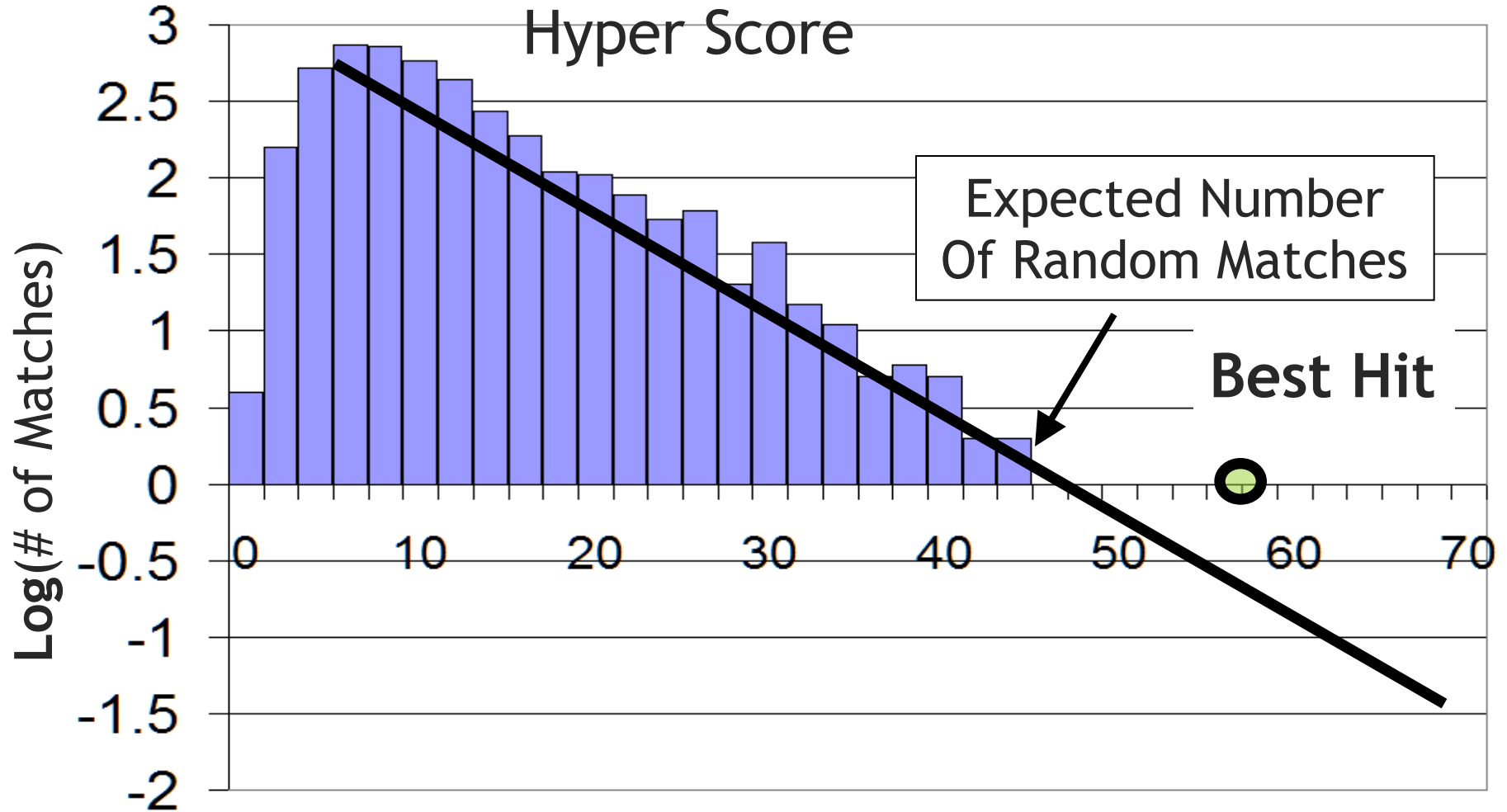
# Distribution of "Incorrect" Hits



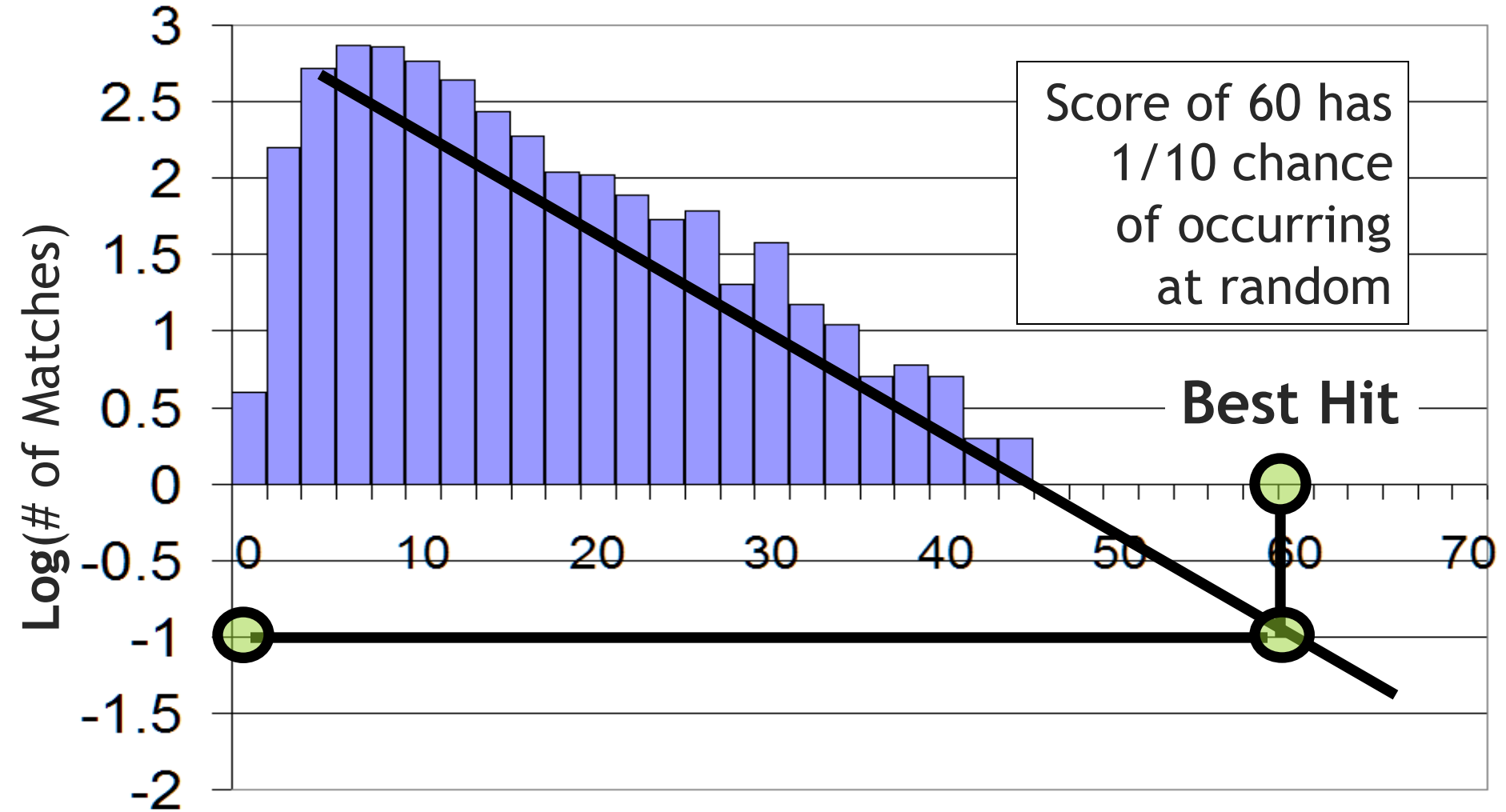
# Estimate Likelihood (E-Value)



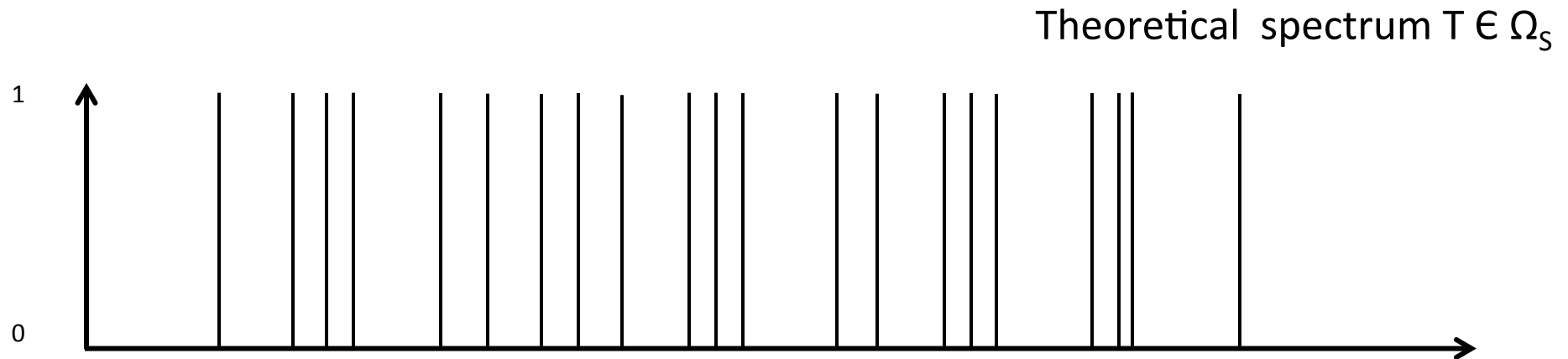
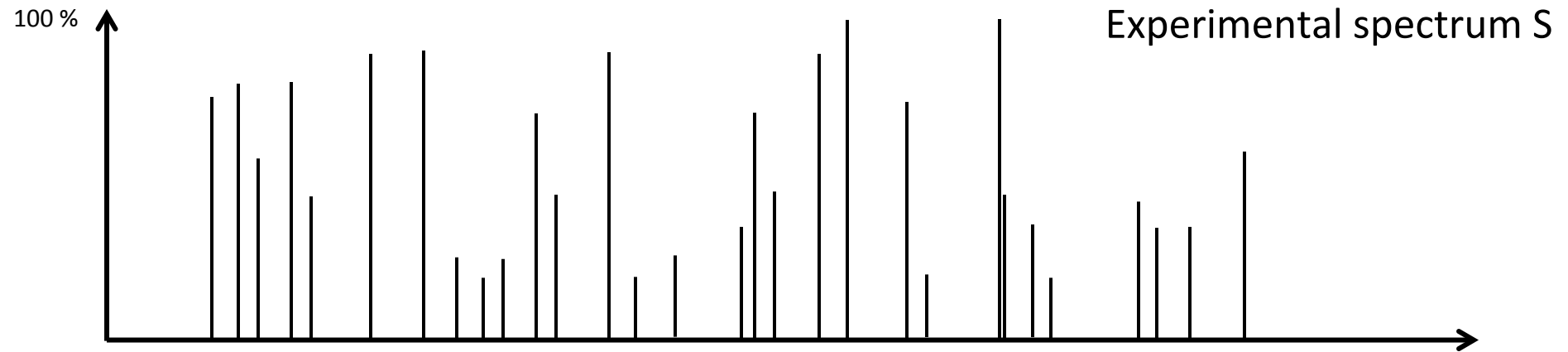
# Estimate Likelihood (E-Value)



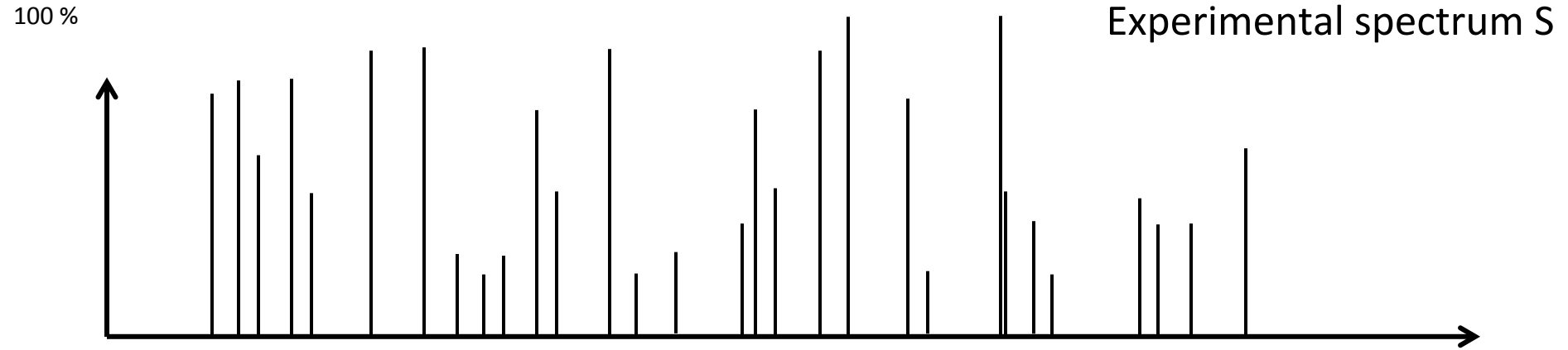
# Estimate Likelihood (E-Value)



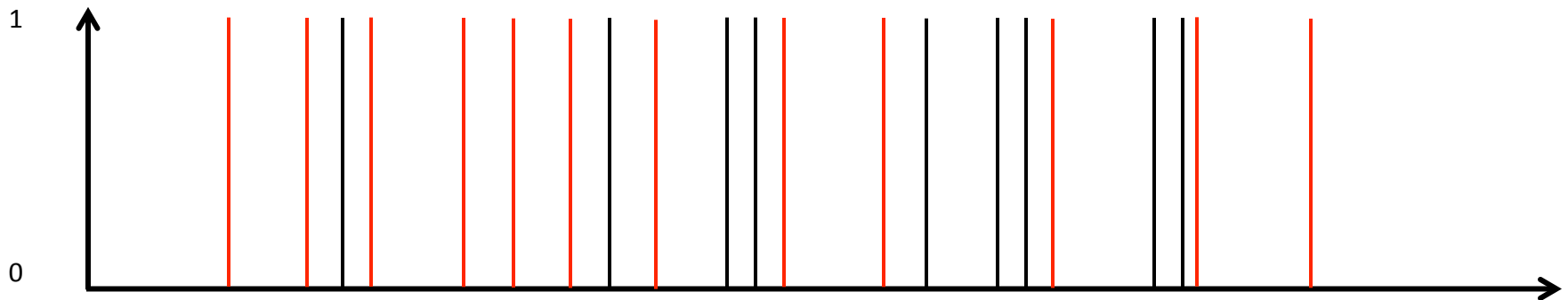
# Sequest



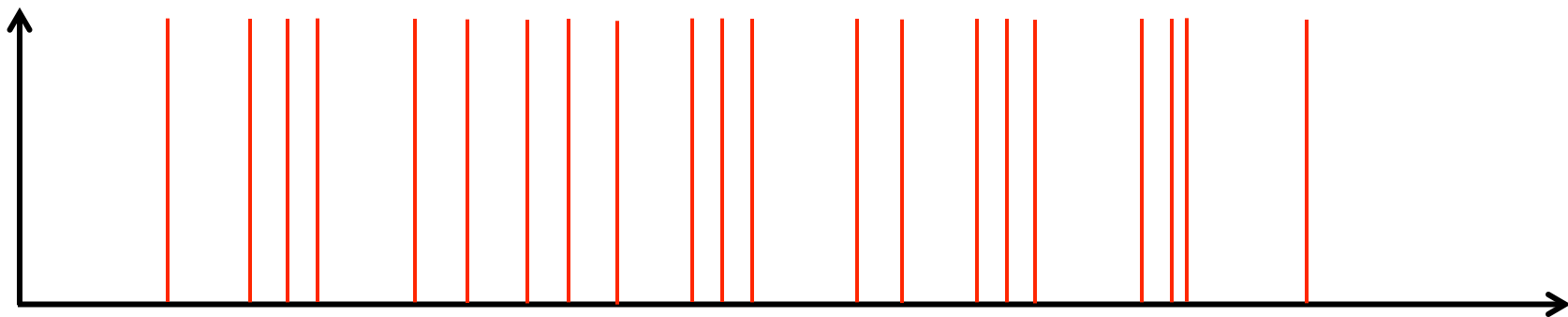
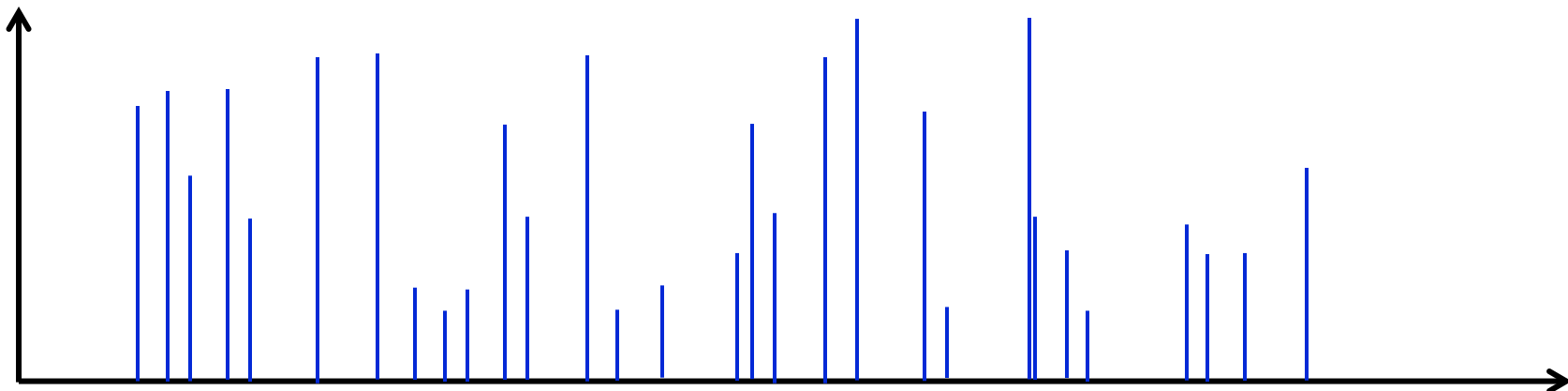
# Sequest – Cross Correlation



- Sum all the peaks that overlap between theoretical and experimental spectrum
- This score is called **cross-correlation**



# 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

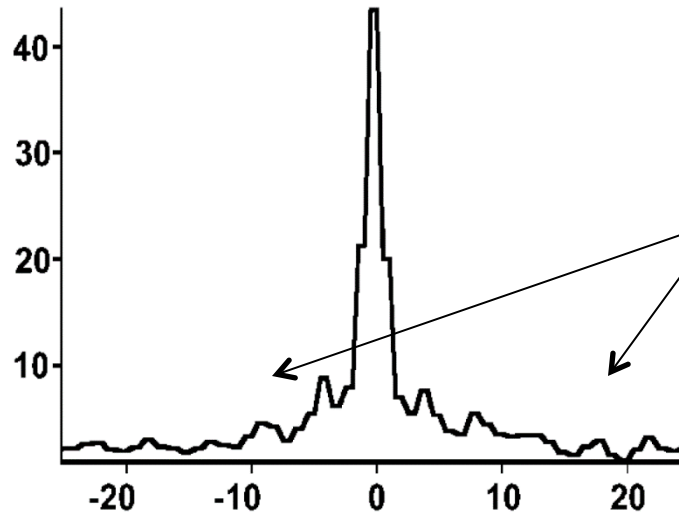
$X_{corr}$  scores

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

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

correlation count

$\times 10^3$



Displacement  $x = 0$ ,  
denotes the cross  
correlation

Displacement  $x \neq 0$ ,  
denotes the auto-  
correlation

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 State and Missed Cleavages

## Charge state

- 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

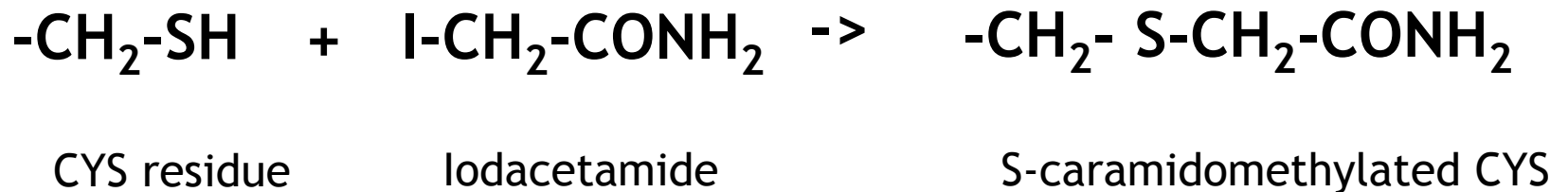
- Sometimes, proteases don't cleave perfectly
- 1 or 2 missed cleavages should be allowed, but be careful since the number of missed cleavages increases your search space sizes!

# Modifications

The modification settings mostly depend on the 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



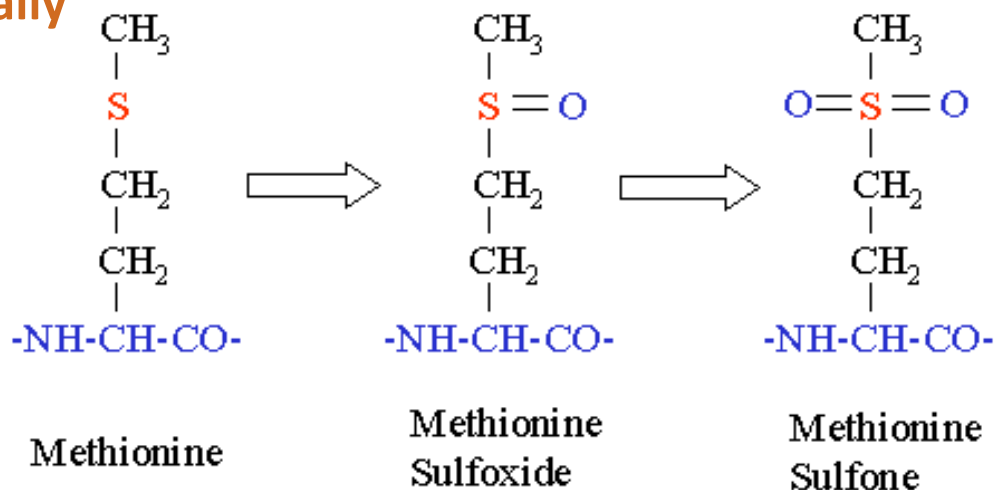
- A fixed modification on amino acid X replaces the original amino acid X during database search

# Modifications

The modification settings mostly depend on the sample preparation

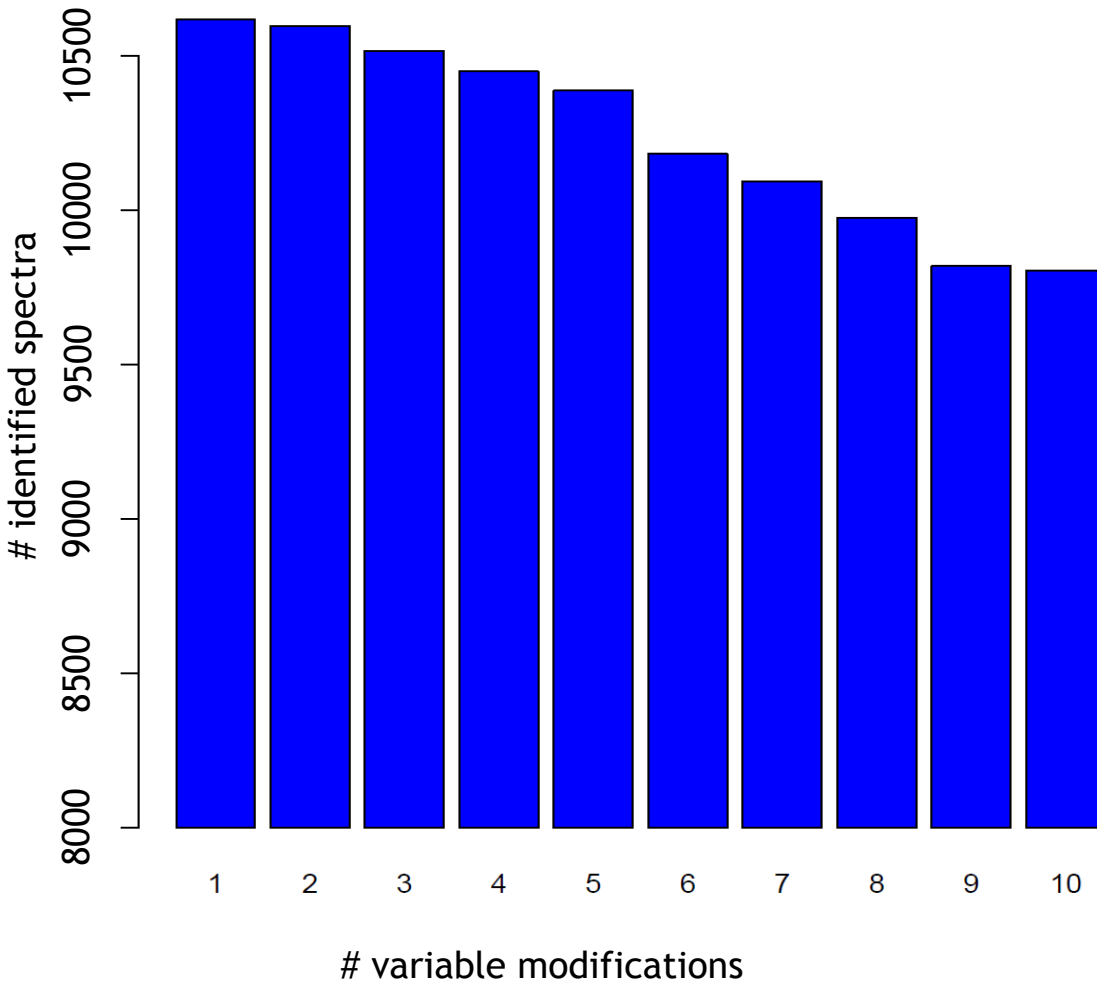
## Variable modifications

- 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 additional amino acids and **impact search space size drastically**





# Variable Modifications



## Intuitively...

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

## However...

- More 'amino acids': increase the search space (combinatorial explosion)
- Loss in sensitivity
- Variable modifications need to be carefully chosen

# LEARNING UNIT 7C

## FDR ESTIMATION

This work is licensed under a Creative Commons Attribution 4.0 International License.



# Database Settings

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

# Target-Decoy Databases

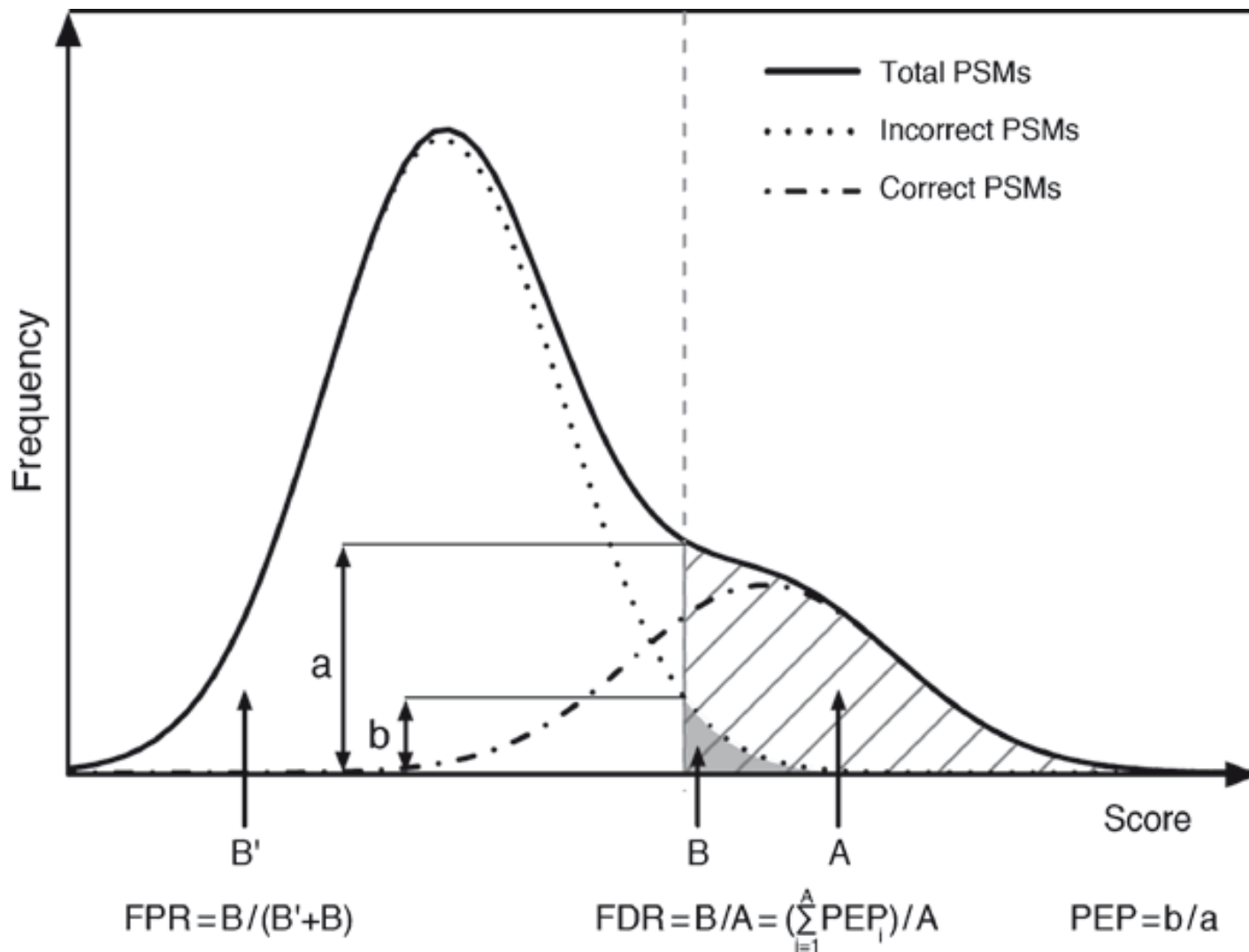
- Take the original protein sequences (target sequences) and reverse, pseudo-reverse, randomize or shuffle these sequences to create **decoy sequences**
- Spectra are either searched twice (first against target, then decoy database) or against the concatenated database (target + decoy)
- Decoy sequences are random sequences and should not be present in the sample
- PSMs against decoy proteins have to be **false positive identifications**
- Note
  - 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 Approach

Peptide identification	Search engine score	TARGET/DECOY
LCEVEEGDKEDVDK	$s_1$	TARGET
YTAQVDAEEKEDVK	$s_2$	TARGET
IVADKDYSVTANSK	$s_3$	TARGET
TGIEIIKK	$s_4$	TARGET
DLGEEHFK	$s_5$	TARGET
TASSDTSEELNSQDSPK	$s_6$	<b>DECOY</b>
GAGGENEPPAAAPEPR	$s_7$	TARGET
IKDPDAAKPEDWDDR	$s_8$	TARGET
VDEVGGEALGR	$s_9$	TARGET
SEEQLKEEGIEYK	$s_{10}$	<b>DECOY</b>
LHVDPENFK	$s_{11}$	TARGET
FSTVAGESGSADTVRDPR	$s_{12}$	TARGET
AEDEILNR	$s_{13}$	<b>DECOY</b>

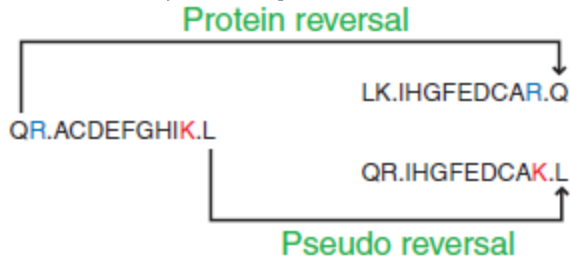
- PSMs are sorted by (deteriorating) score
- As the score gets worse, the likelihood of finding a decoy hit increases, likelihood for target hit decreases
- By choosing an appropriate score threshold, one can ensure a given false-discovery rate (FDR)

# PSM Score Distribution



# Target-Decoy Approach

## Design decoy sequences



### Random

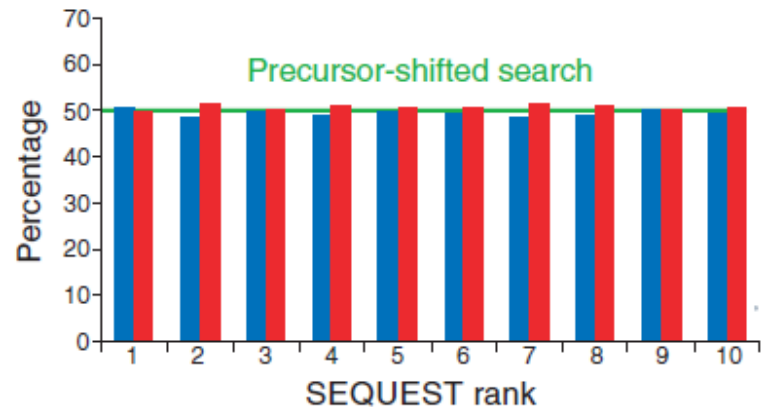
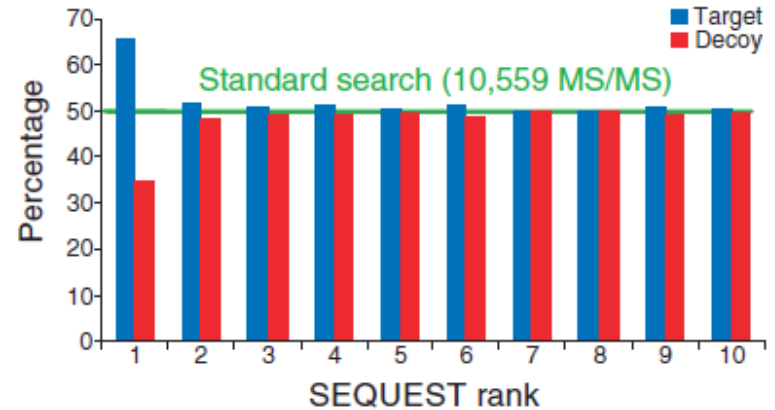
Residue	Frequency
A	0.070
C	0.023
D	0.046
E	0.070
F	0.036

### 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

- 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 tool **FalseDiscoveryRate** uses the *Käll* metrics



# LEARNING UNIT 7D

## CONSENSUS IDENTIFICATION

- Overlap between search engines
- Consensus identification idea
- Scaffold
- OpenMS Consensus ID



# Comparison of Search Engines

## Simple experiment

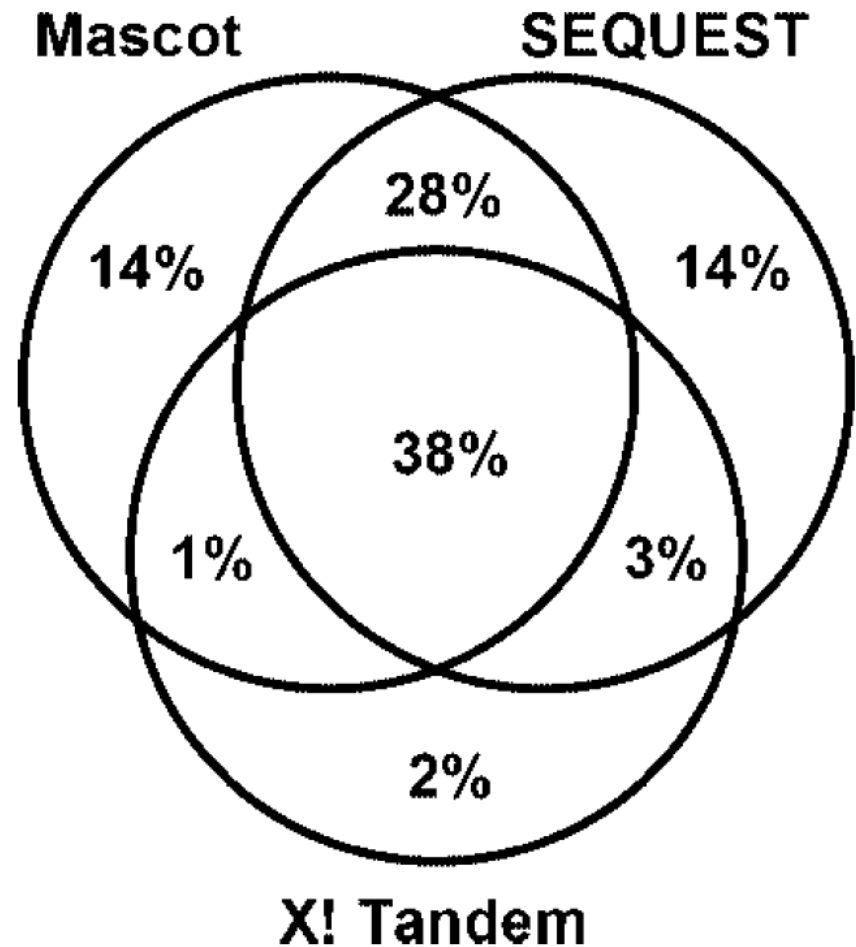
- 18 protein mix, digest, measure
- Same dataset searched with three different search engines
- Equivalent search parameters

## Result





- Overlap between search engines is rather limited
- Each search engine finds (correct) candidates none of the others finds

## Idea

- **Combine results from multiple search engines (consensus identification)**



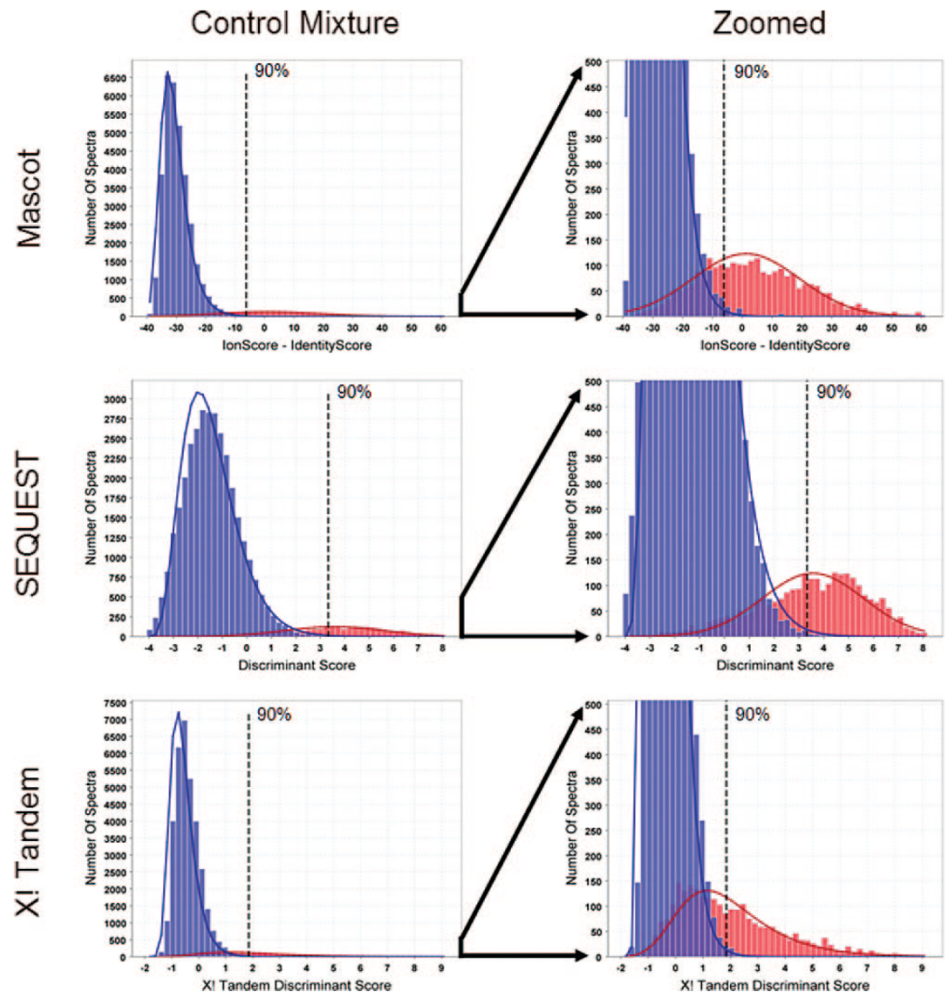
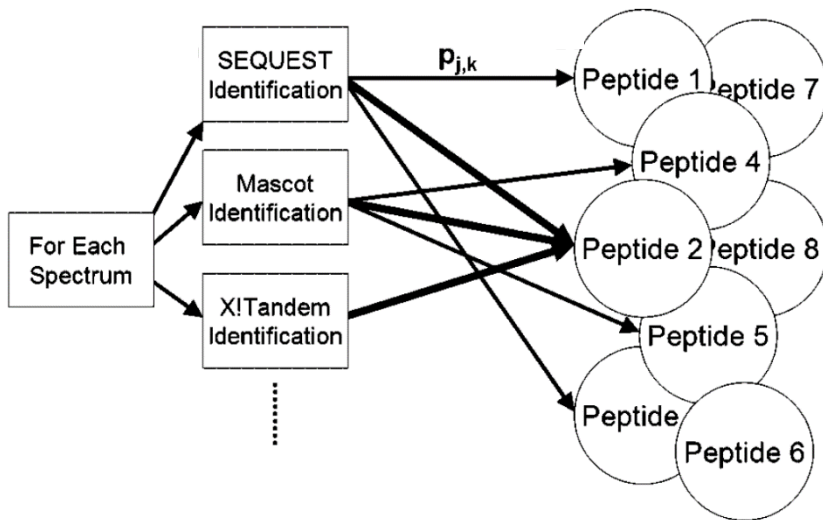
# Multiple search engines

- 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
  - 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 scores to probabilities



# Scaffold

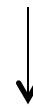
- Calculate agreement score for each PSM across all search engines

$D$  = 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



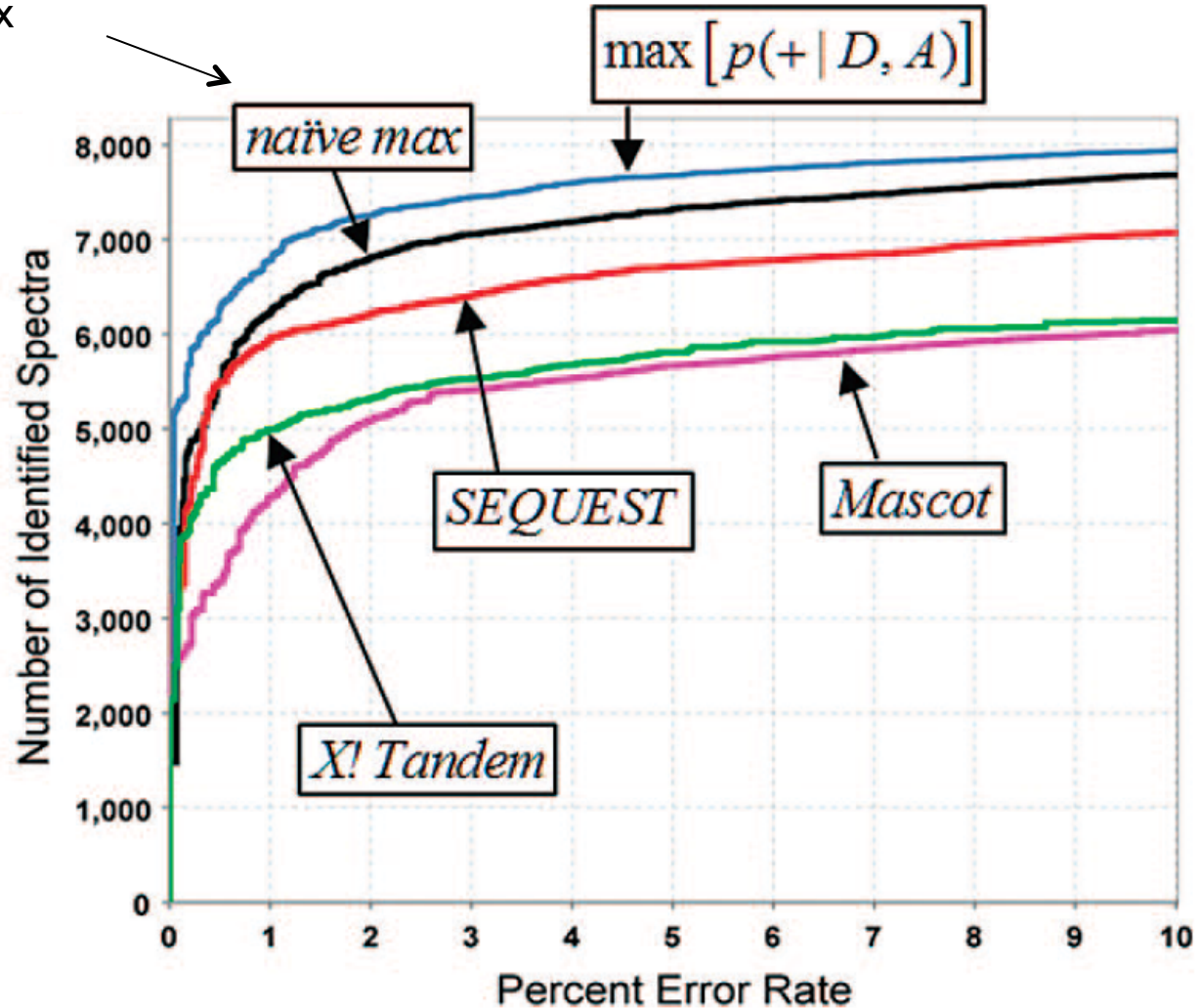
Conditional probability for being correct given a PSM  $D$



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

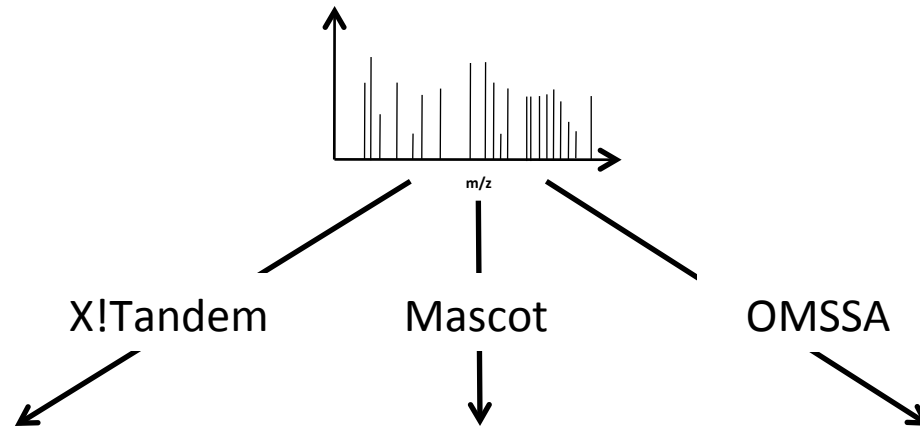
# Scaffold Performance

We did not discuss  
the naïve max



# ConsensusID

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



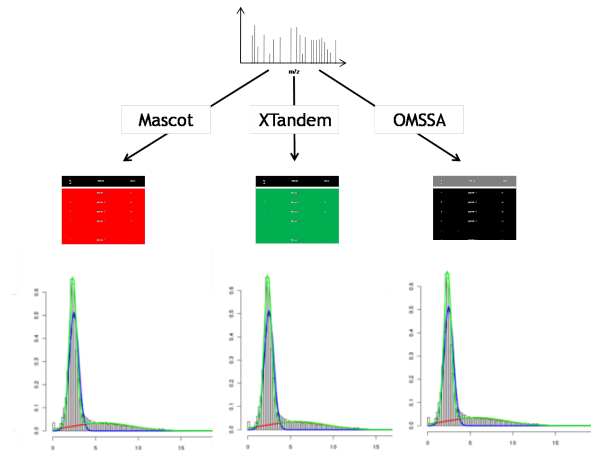
Rank	Peptide	Score
1	QRESTATDILQK	0.008

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

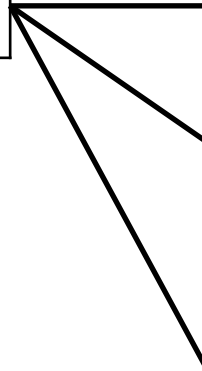
Rank	Peptide	Score
1	AELASCVVDLGAK	0.94
2	ELM(Ox)SNGPGSIIGAK	0.97
3	ISCAEGALEALKK	0.99
4	QRESTATDILQK	0.99



# 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

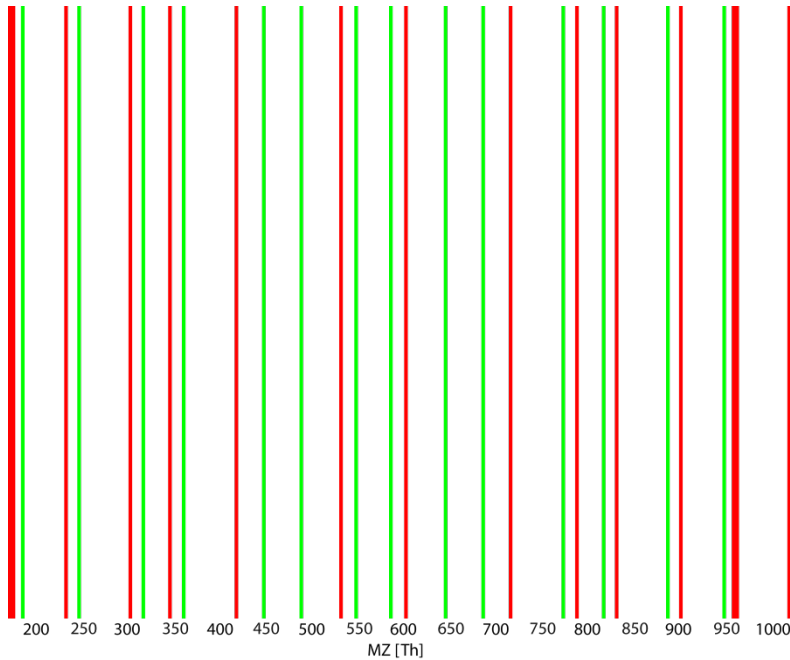
Rank	Peptide	Score
1	QRESTATDILQK	0.54

47%

42%

21%

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



QRESTATDILQK      similarity  $*s_2(p_1)$

# ConsensusID - Consensus Score

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

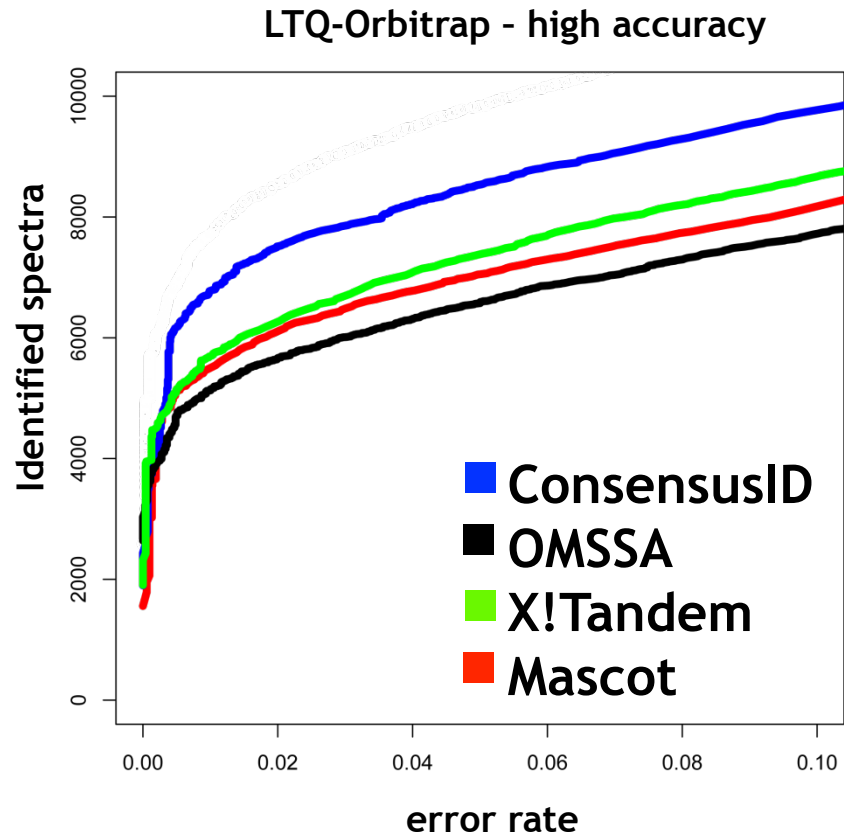
Rank	Peptide	Score
1	AELASCVVGDLGAK	0.94
2	ELM(Ox)SNGPGSIIGAK	0.97
3	ISCAEGALEALKK	0.99
4	QRESTATDILQK	0.99
5	EIEEDSLEGLKK	$S_{3,5}$
6	GIEDDLMDLIKK	$S_{3,6}$

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

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

**0.34**

# ConsensusID Performance



error rates = *false discovery rates*

# Materials

- Online Materials
  - Learning Unit 3B (statistics for FDR)
  - Learning Unit 7A, B, C, D
- Slides on peptide ID by Brian Searle
  - <https://proteome-software.wikispaces.com/file/view/interpreting-MS-MS-proteomics-results.ppt>

# 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.
- Fenyő 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