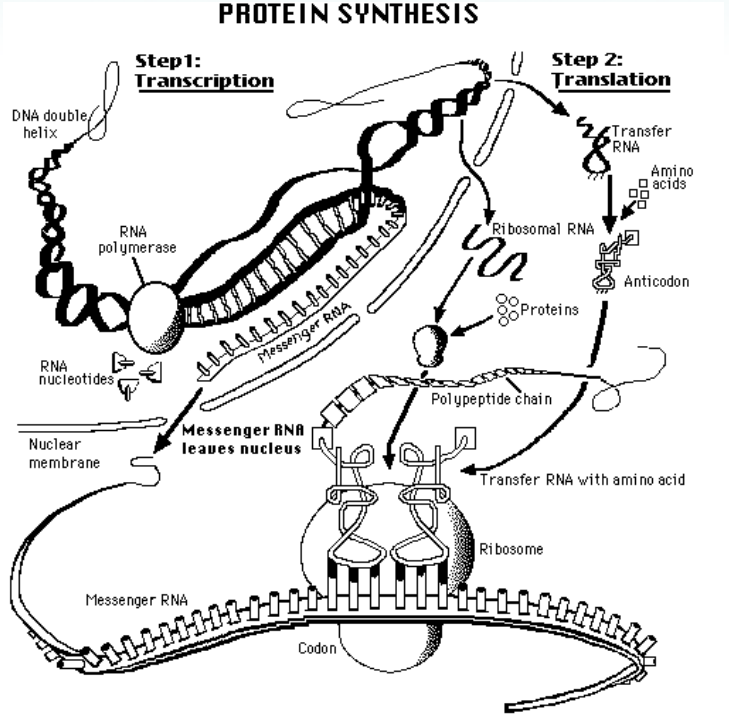
Lecture 15: Protein Sequencing

Study Chapter 8.10-8.15

From DNA to Proteins

- DNA sequences
  - "*OS*" that controls living biological systems
  - Sections of DNA (Genes) encode proteins, like programs
  - Triplets of nucleotides (codons) encode the amino-acid sequences, as well as the stop codes, used to assemble proteins
  - Complications in going from DNA → Protein: introns, RNA editing prior to translation, post-translational modifications



Proteins

- Proteins are the "machinery" or "hardware"
  - Compose the cellular structures
  - Control the biochemical reactions in cells
  - Regulate and trigger the chain reactions (metabolic pathways) that result in the cell's life cycle
  - Determine which parts of the DNA "code" are activated, executed, and when
- Like DNA, proteins are long molecular chains
  - Sequences of 20 amino acid residues rather than 4 nucleic acids

Protein Components

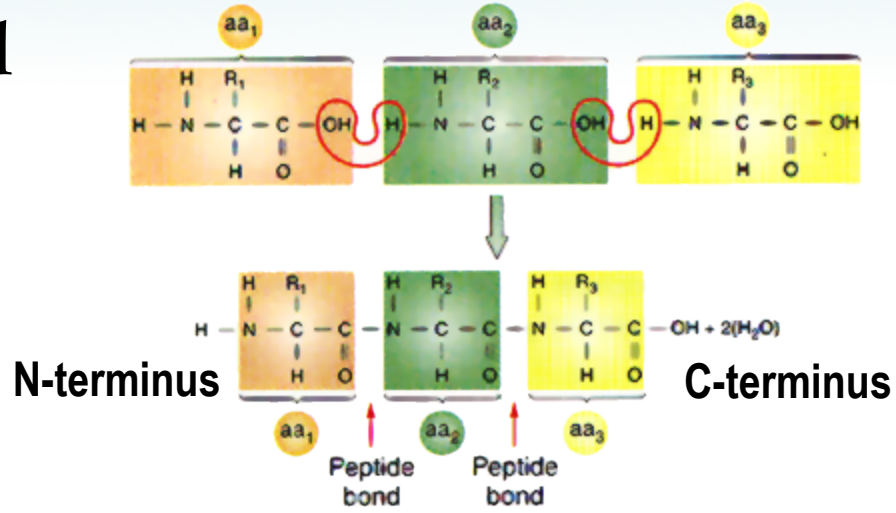
- Proteins are made from 20 amino acids
- Peptide bonds join amino acids into long chains
- 100's to 1000's of amino acid residues long

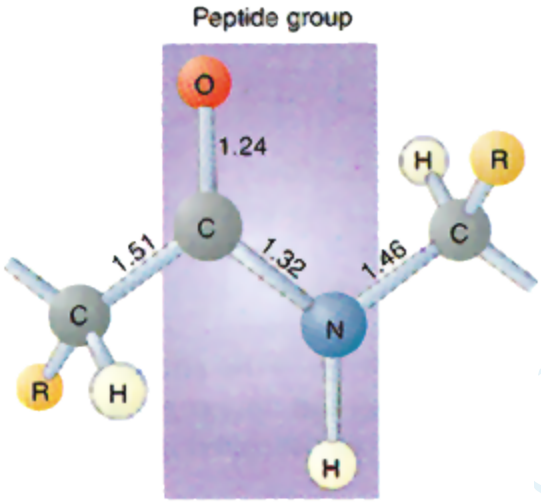
Amino Acid	3-Letter Code	1-Letter Code	Molecular Weight
Alanine	Ala	A	89.09
Cysteine	Cys	C	121.16
Aspartate	Asp	D	133.10
Glutamate	Glu	E	147.13
Phenylalanine	Phe	F	165.19
Glycine	Gly	G	75.07
Histidine	His	H	155.16
Isoleucine	Ile	I	131.18
Lysine	Lys	K	146.19
Leucine	Leu	L	131.18

Amino Acid	3-Letter Code	1-Letter Code	Molecular Weight
Methionine	Met	M	149.21
Asparagine	Asn	N	132.12
Proline	Pro	P	115.13
Glutamine	Gln	Q	146.15
Arginine	Arg	R	174.20
Serine	Ser	S	105.09
Threonine	The	T	119.12
Valine	Val	V	117.15
Tryptophan	Trp	W	204.23
Tyrosine	Tyr	Y	181.19

Protein Assembly

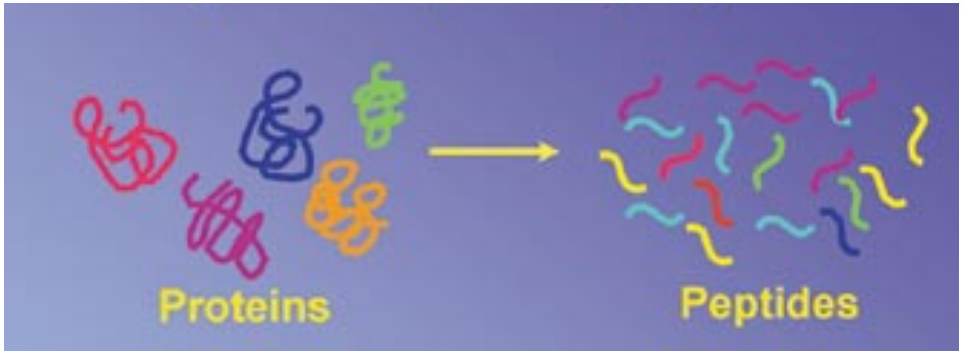
- Amino acids are joined by peptide bonds into long chains
- These chains "fold" into proteins
- Interact with other proteins and large molecules





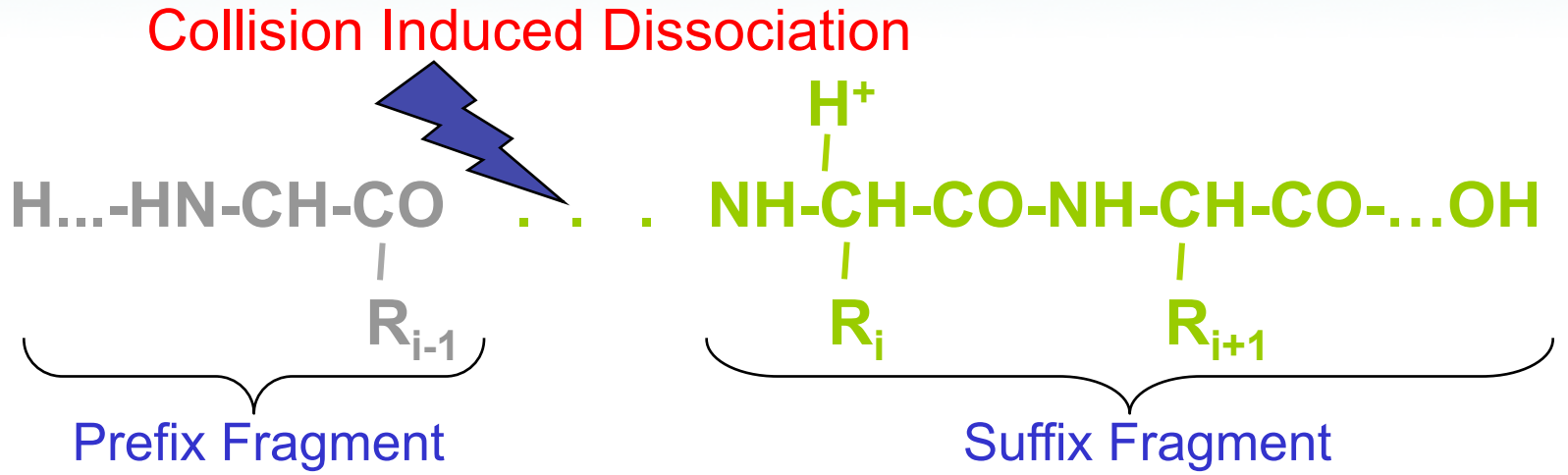
Protein Sequencing

- Purify a sample
- Break into pieces
  - Proteases cleave proteins into smaller "peptide" chains



- Read fragments
  - Edman degradation for short peptide sequences
  - Mass spectrometry measures mass/charge
  - The "Hard" part
- Reassemble
  - Relatively easy

Peptide Fragmentation

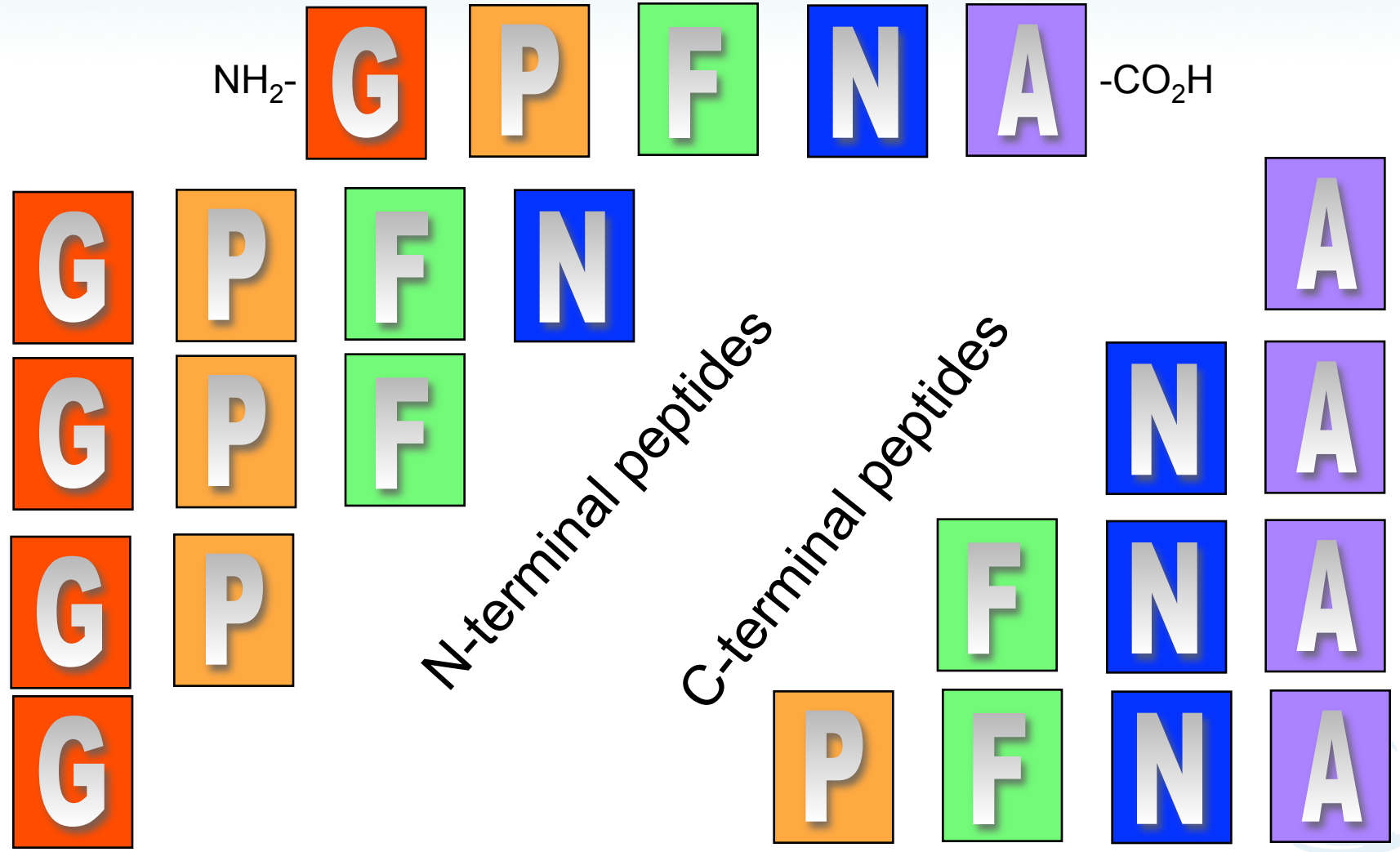


- Peptides tend to fragment along the backbone.
- Fragments can also lose neutral chemical groups like NH_3 and H_2O.

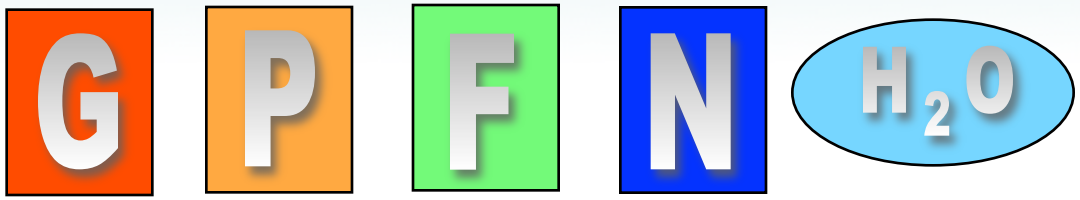
Breaking Peptides into Fragment Ions

- Proteases, e.g. trypsin, break proteins into *peptides*.
- A Tandem Mass Spectrometer further breaks the peptides down into *fragment ions* and measures the mass of each piece.
- Mass Spectrometer accelerates the fragmented ions; heavier ions accelerate slower than lighter ones.
- Mass Spectrometer measure *mass/charge* ratio of an ion.

N- and C-terminal Peptides



Terminal peptides and ion types

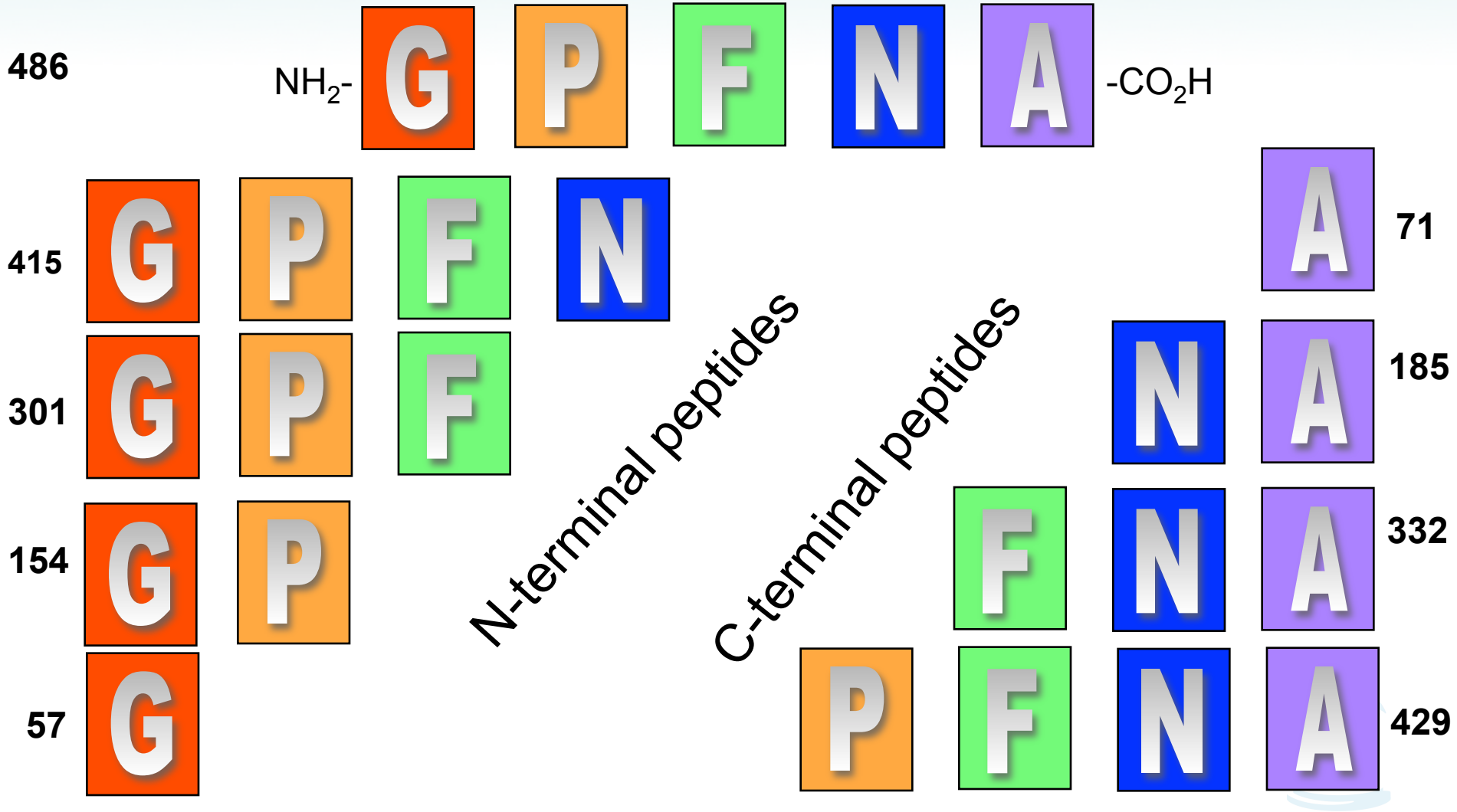
Peptide 

Mass (D) 57 + 97 + 147 + 114 = 415

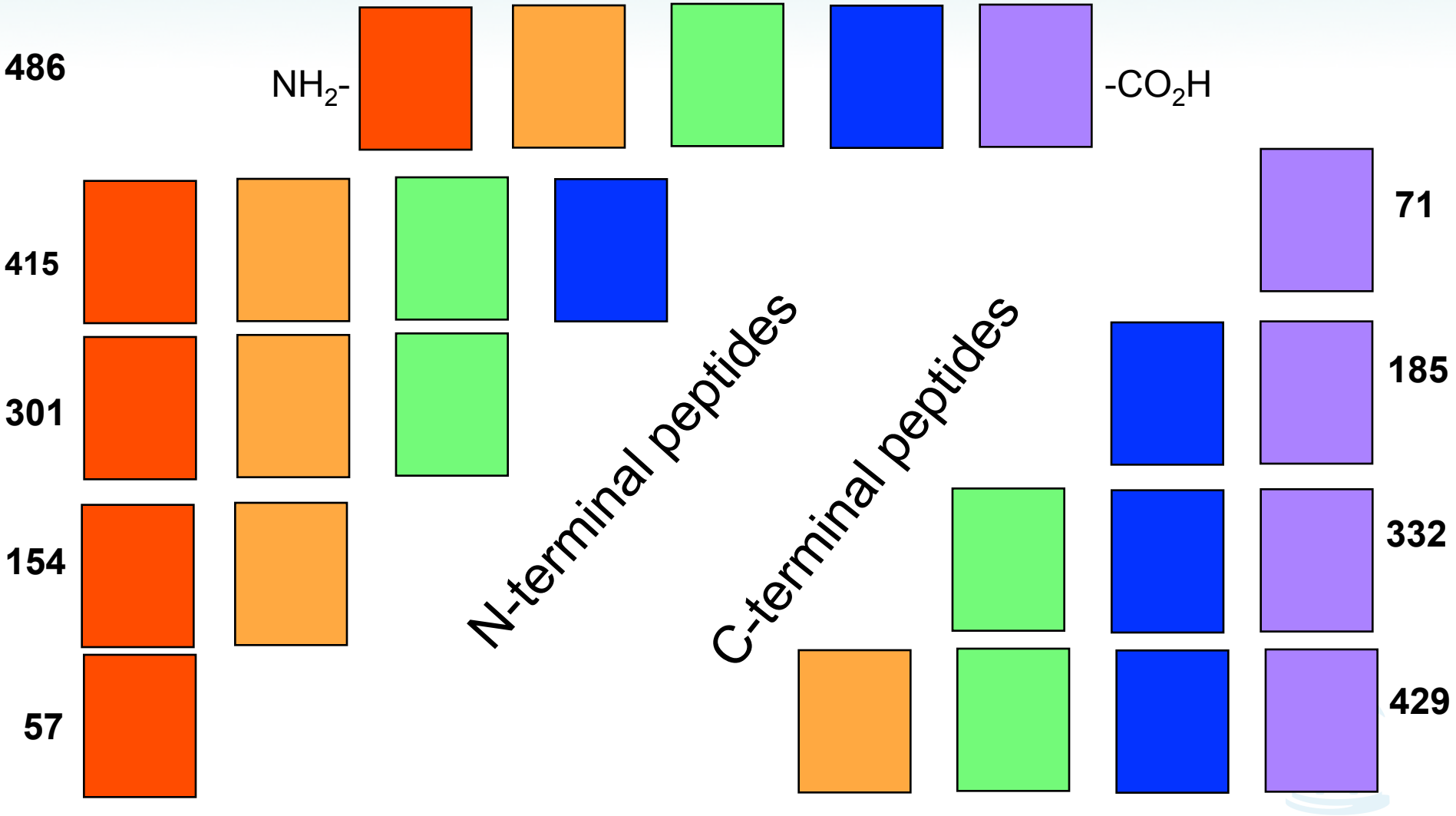
Peptide 

Mass (D) 57 + 97 + 147 + 114 - 18 = 397

N- and C-terminal Peptides



N- and C-terminal Peptides



N- and C-terminal Peptides

486

415

301

154

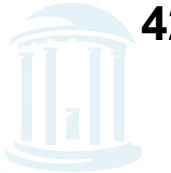
57

71

185

332

429



N- and C-terminal Peptides

486

71

415

Reconstruct peptide from the set of masses of fragment ions

301

(mass-spectrum)

185

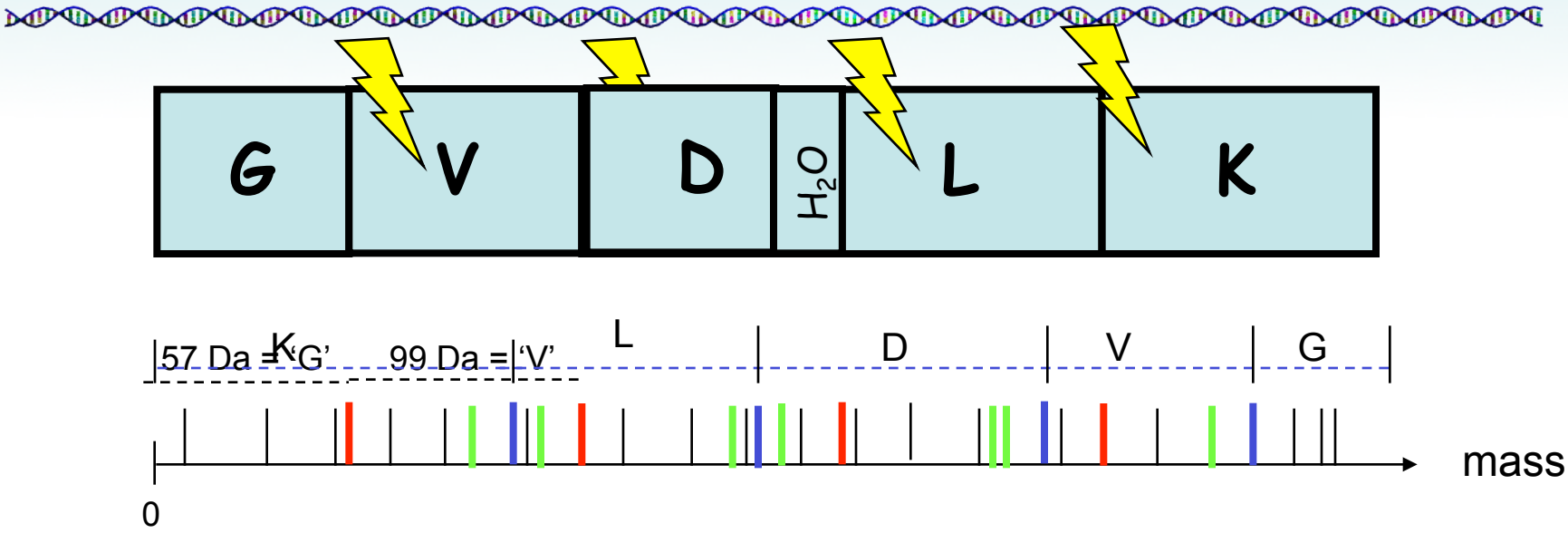
154

332

57

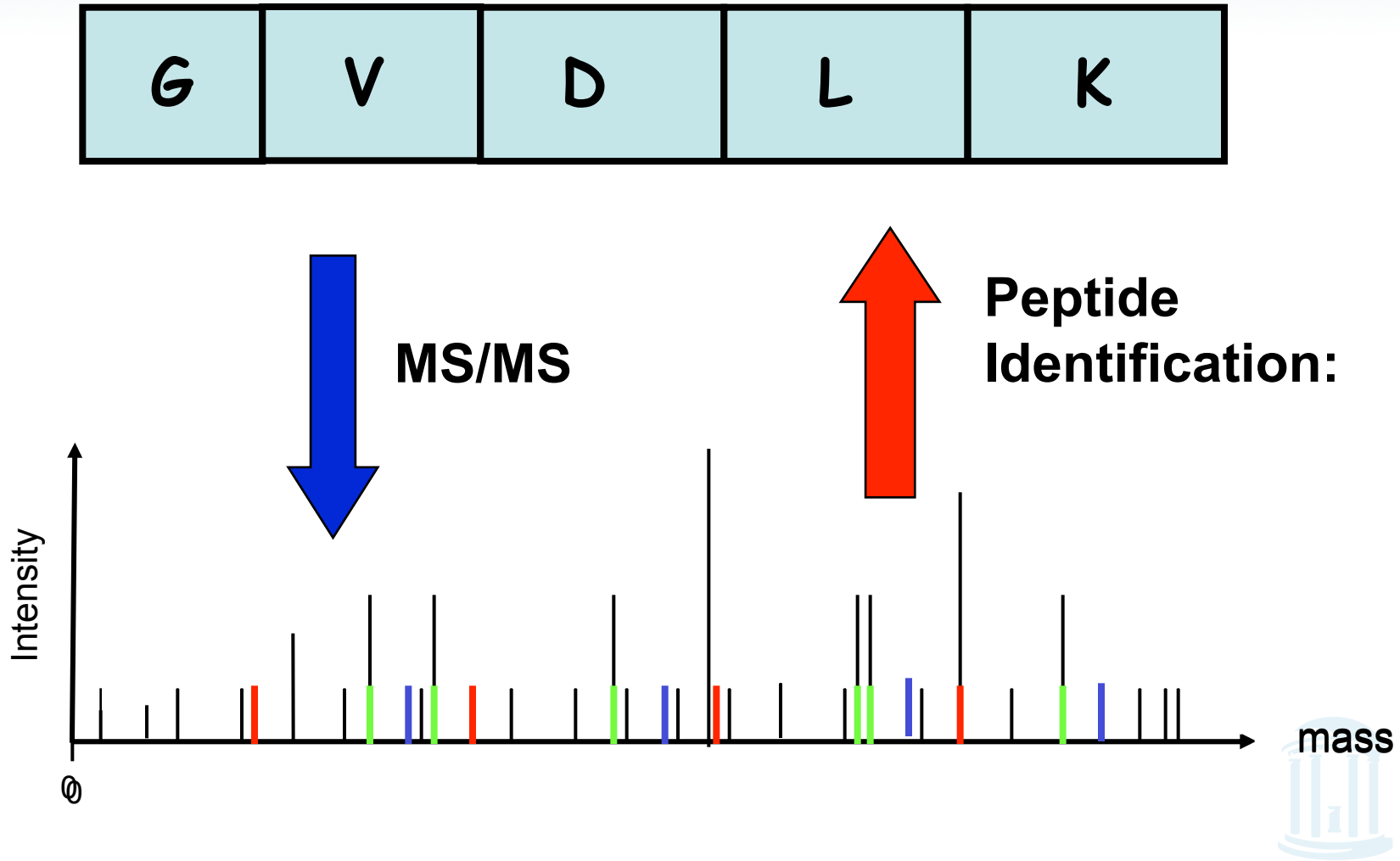
429

Mass Spectra

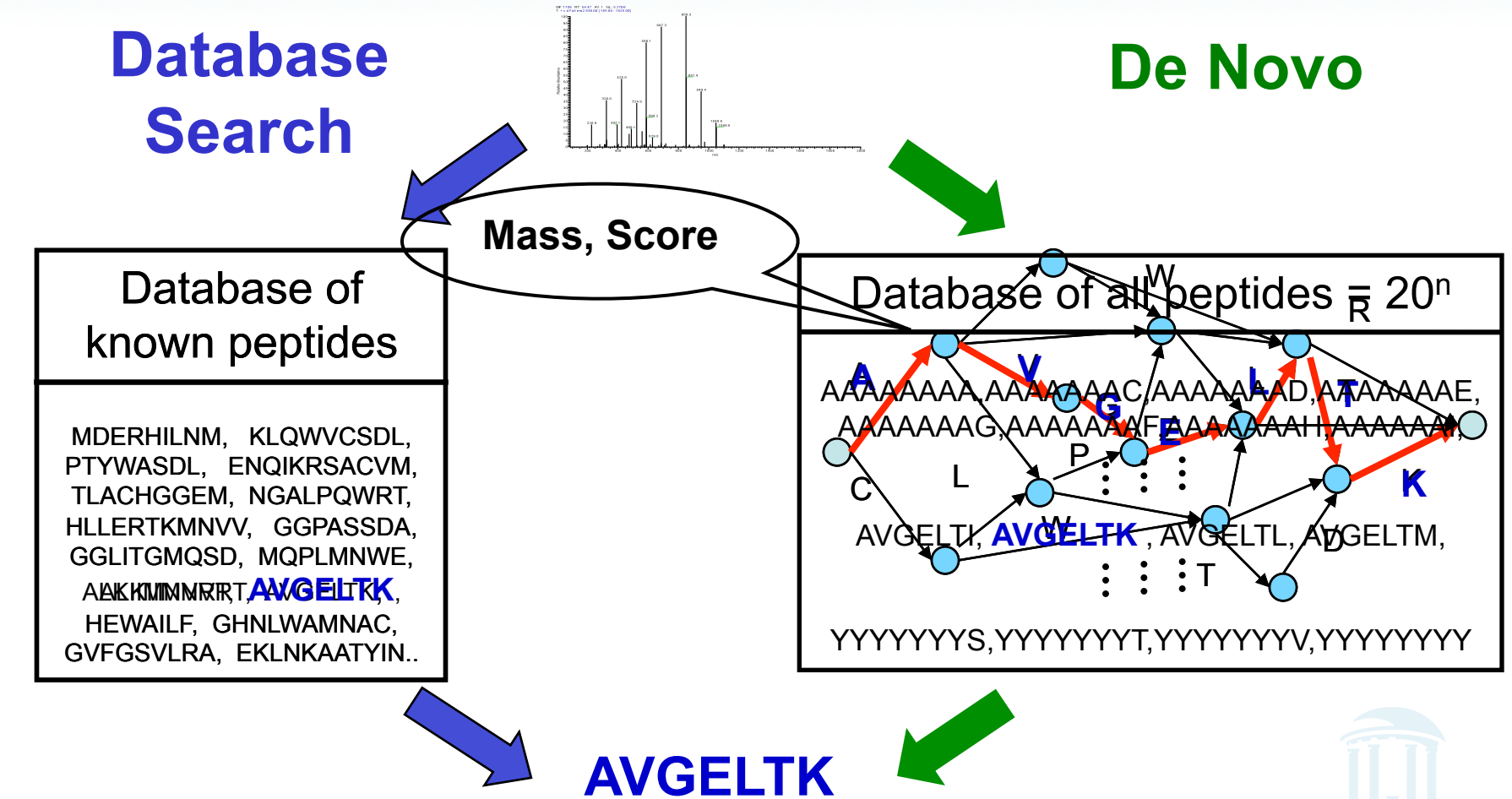


- The peaks in the mass spectrum:
  - Prefix and Suffix Fragments.
  - Fragments with **neutral losses** ($-H_2O$, $-NH_3$)
  - Noise and missing peaks.

Protein Identification with MS/MS



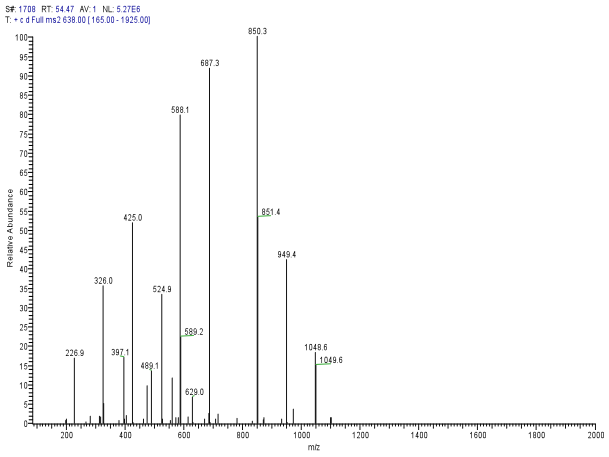
De Novo vs. Database Search

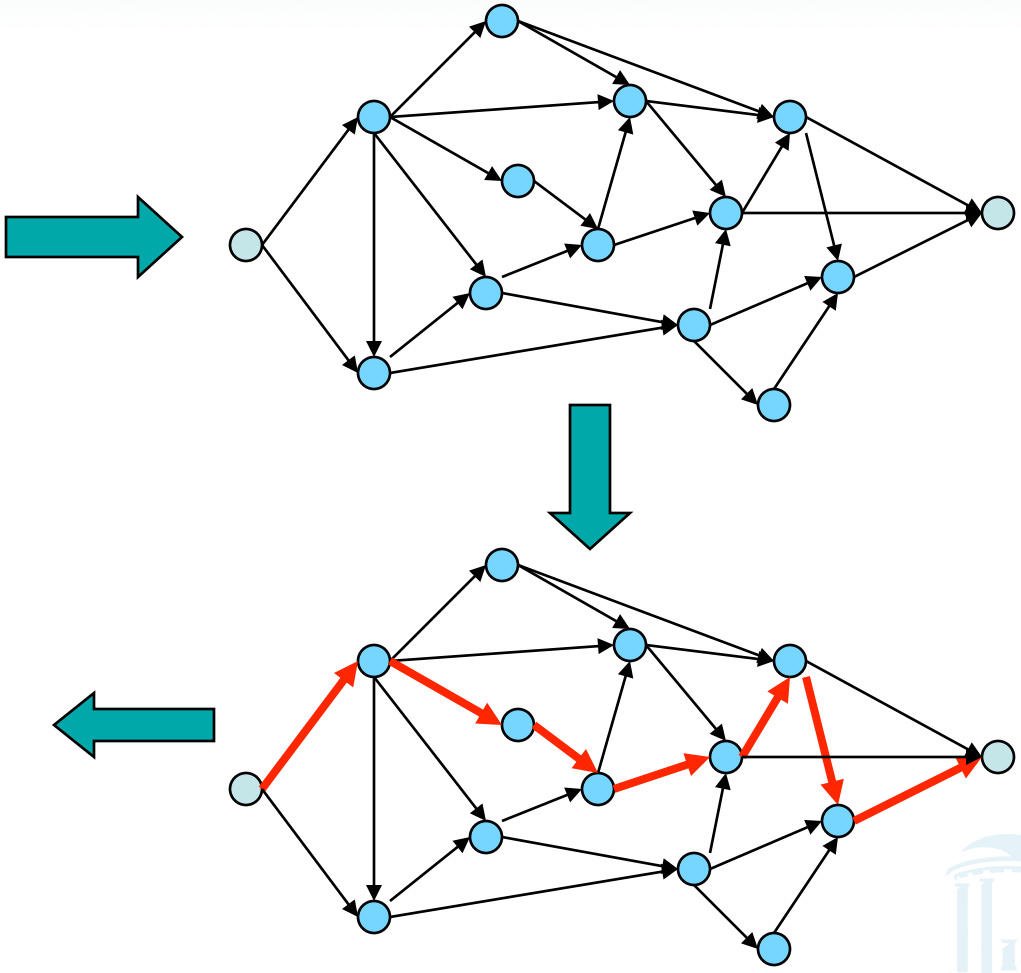


A Paradox

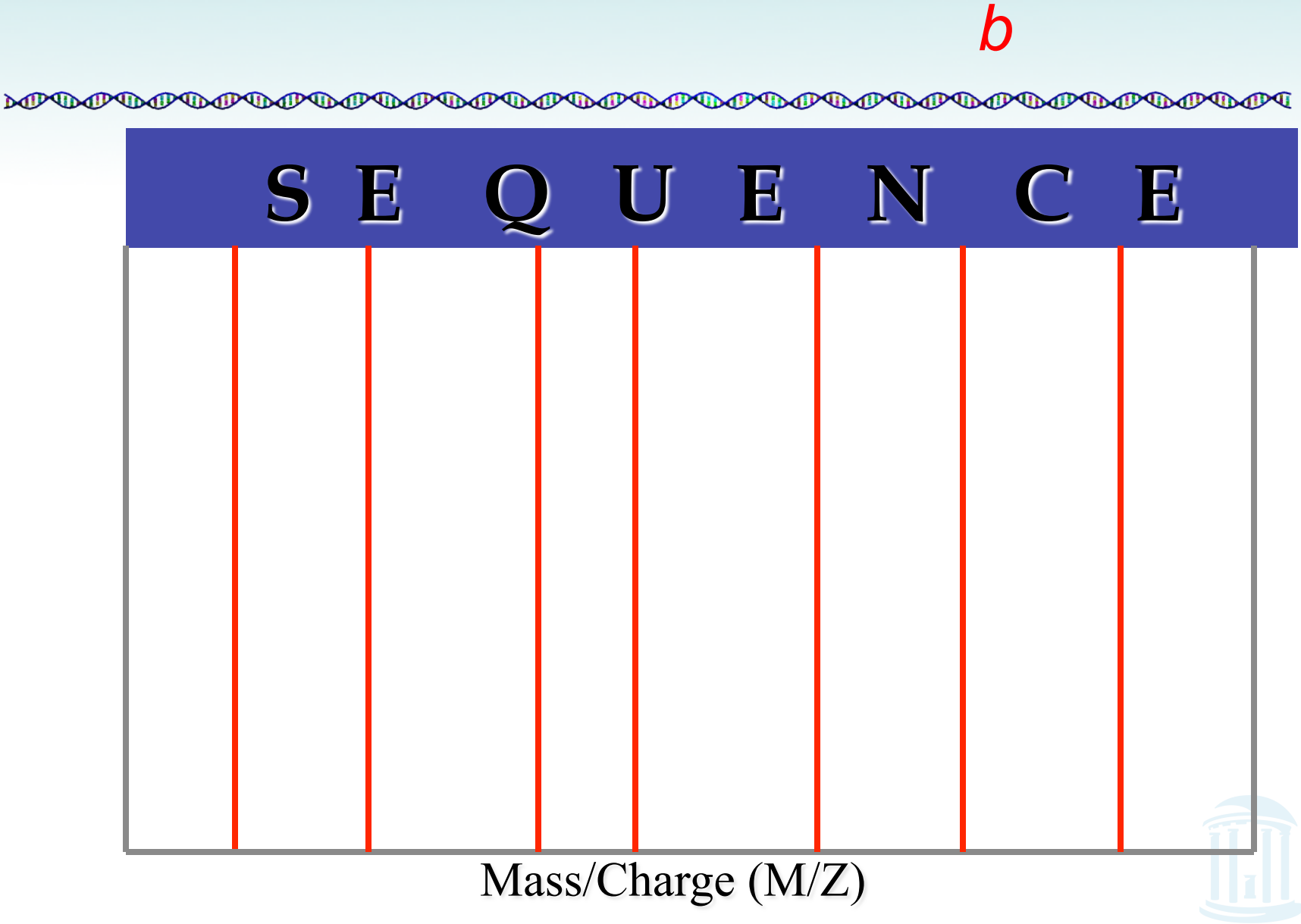
- Database of all peptides is huge ≈ $O(20^n)$.
- Database of all known peptides is much smaller ≈ $O(10^8)$.
- However, *de novo* algorithms can be much *faster*, even though their search space is much *larger!*
- A database search scans all peptides in the ***database of all known peptides*** search space to find best one.
- De novo eliminates the need to scan ***database of all peptides*** by modeling the problem as a graph search.

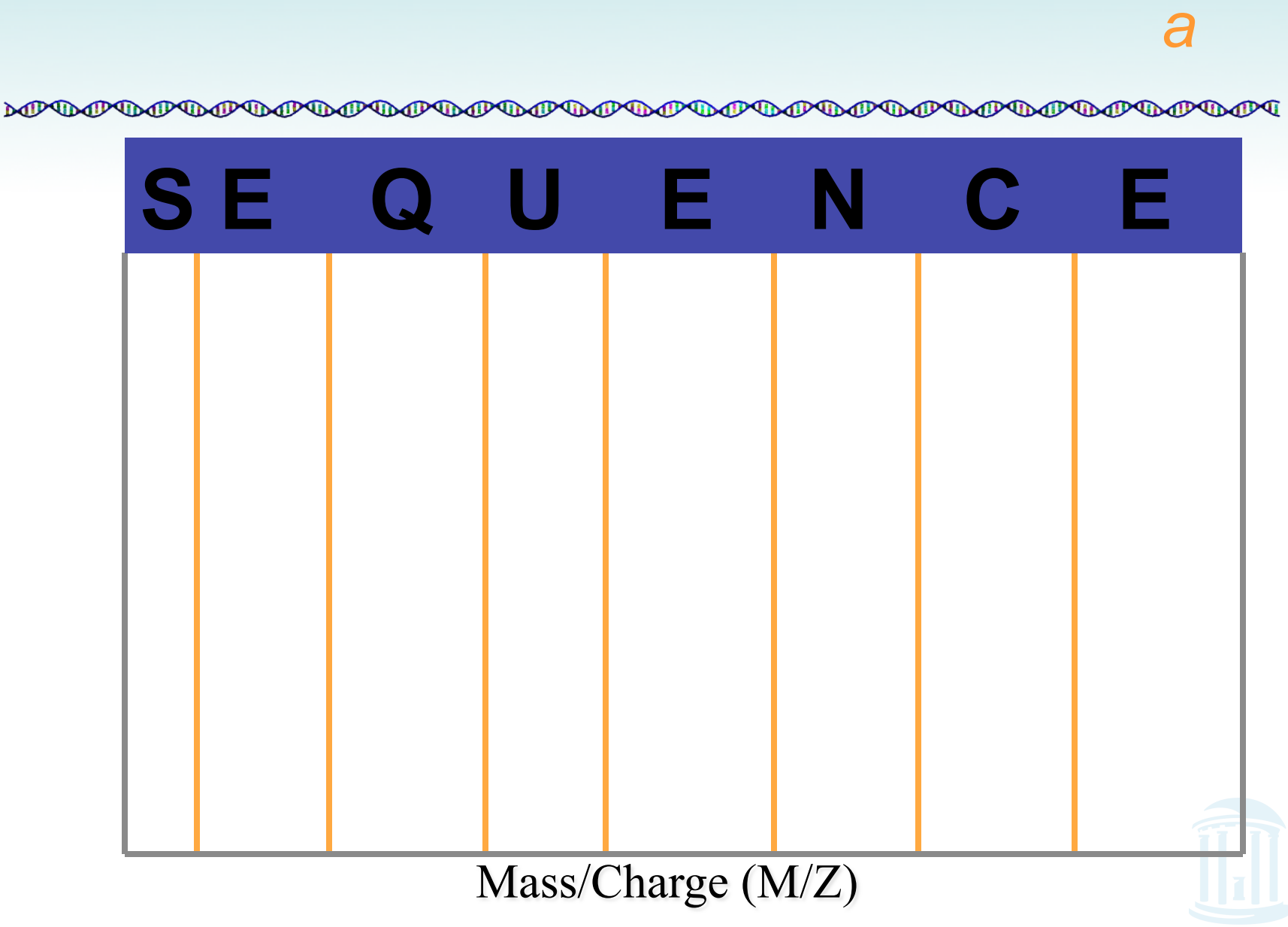
De novo Peptide Sequencing



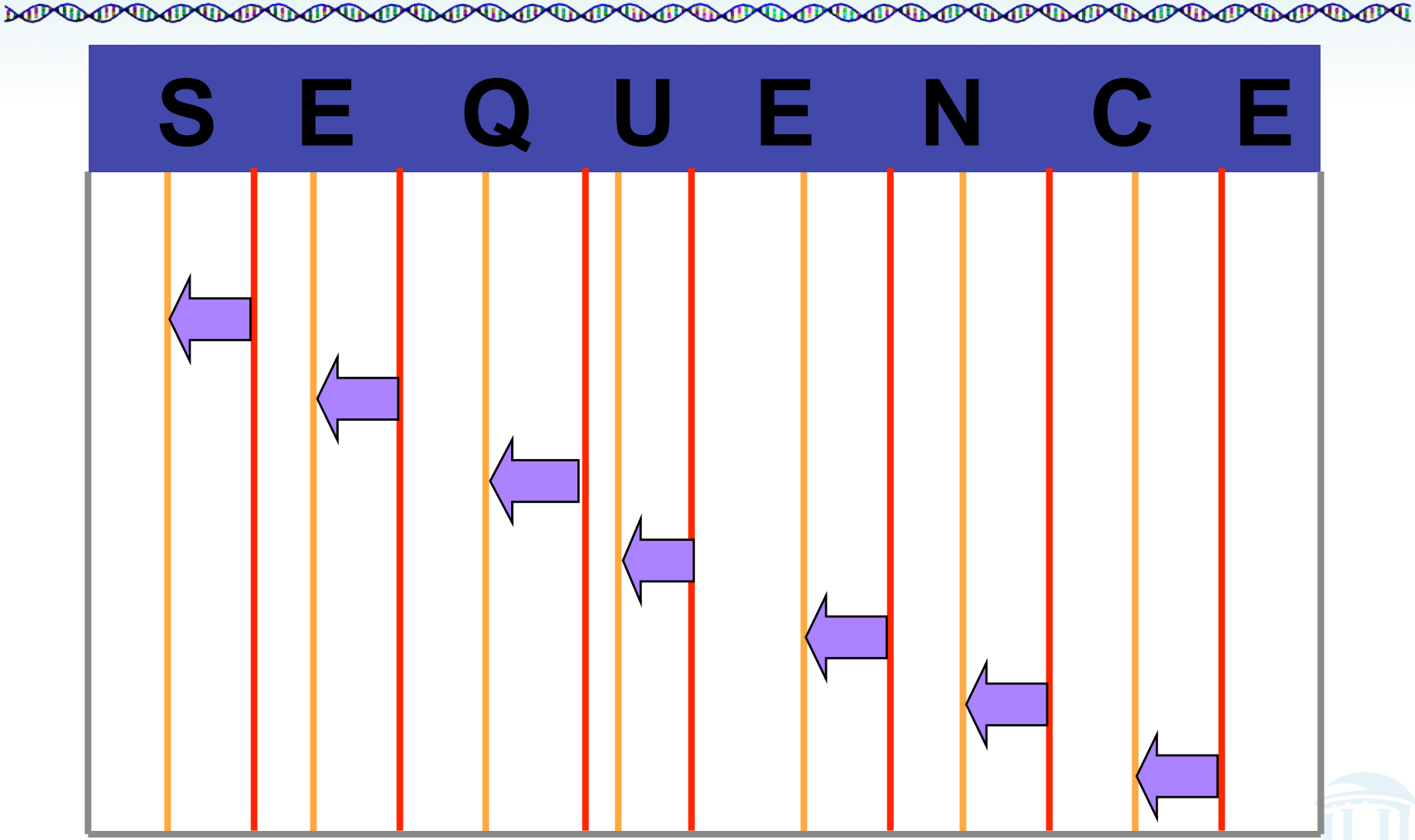


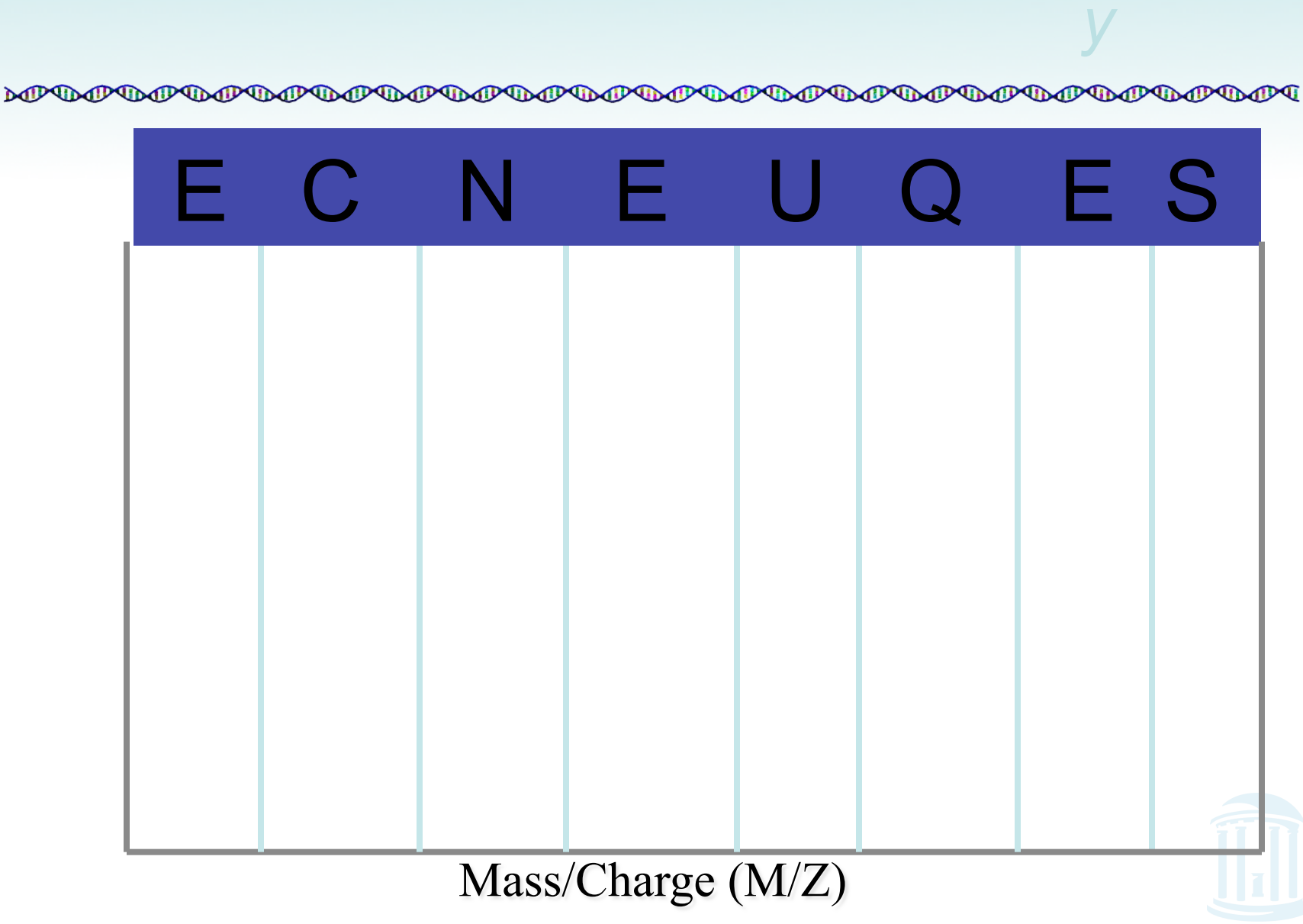
Sequence

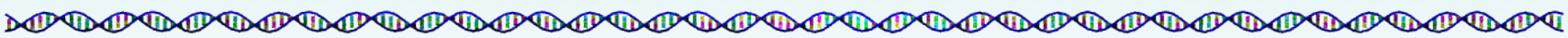

b
S E Q U E N C E
Mass/Charge (M/Z)

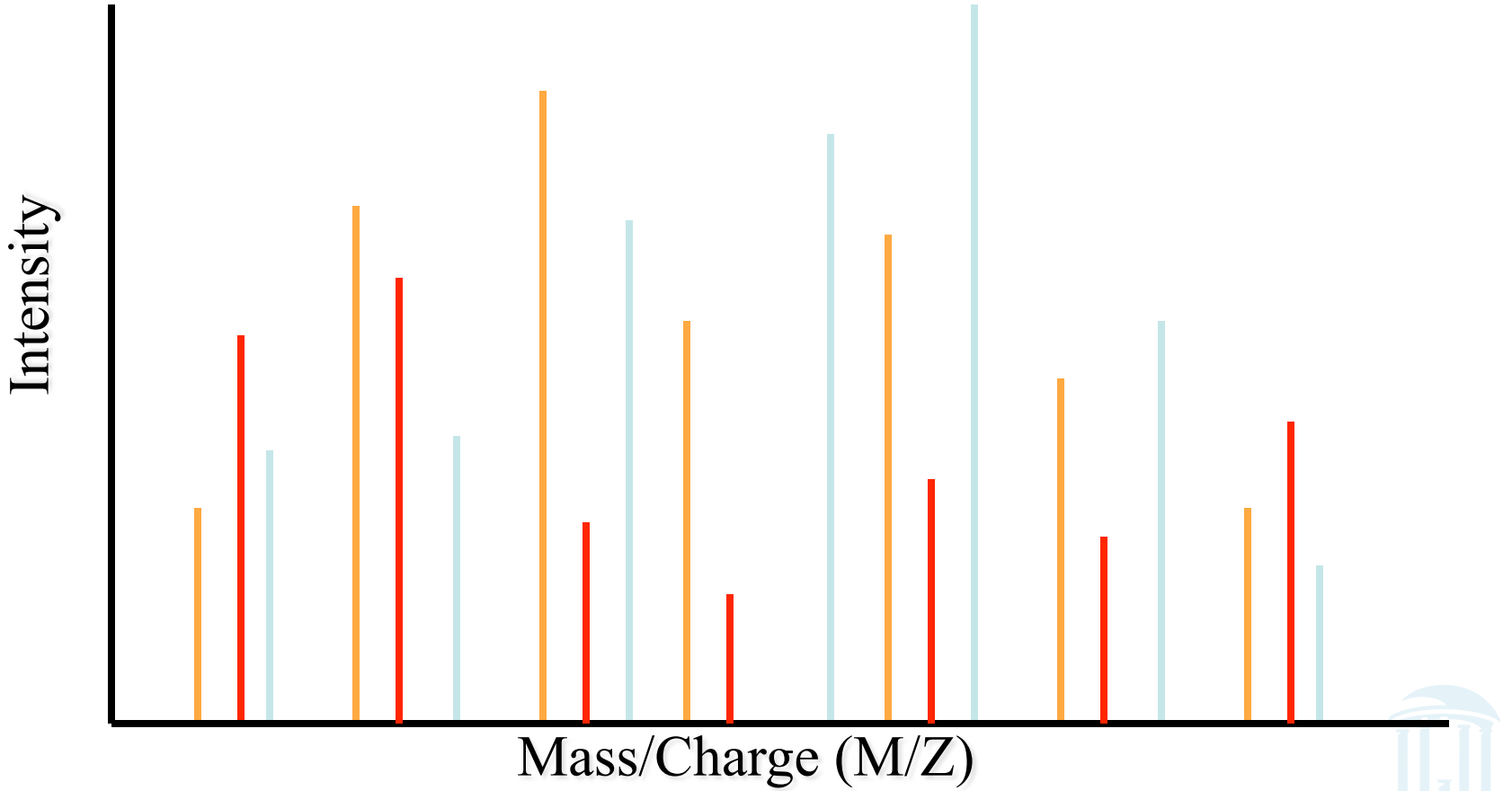

a
SEQUENCE
Mass/Charge (M/Z)

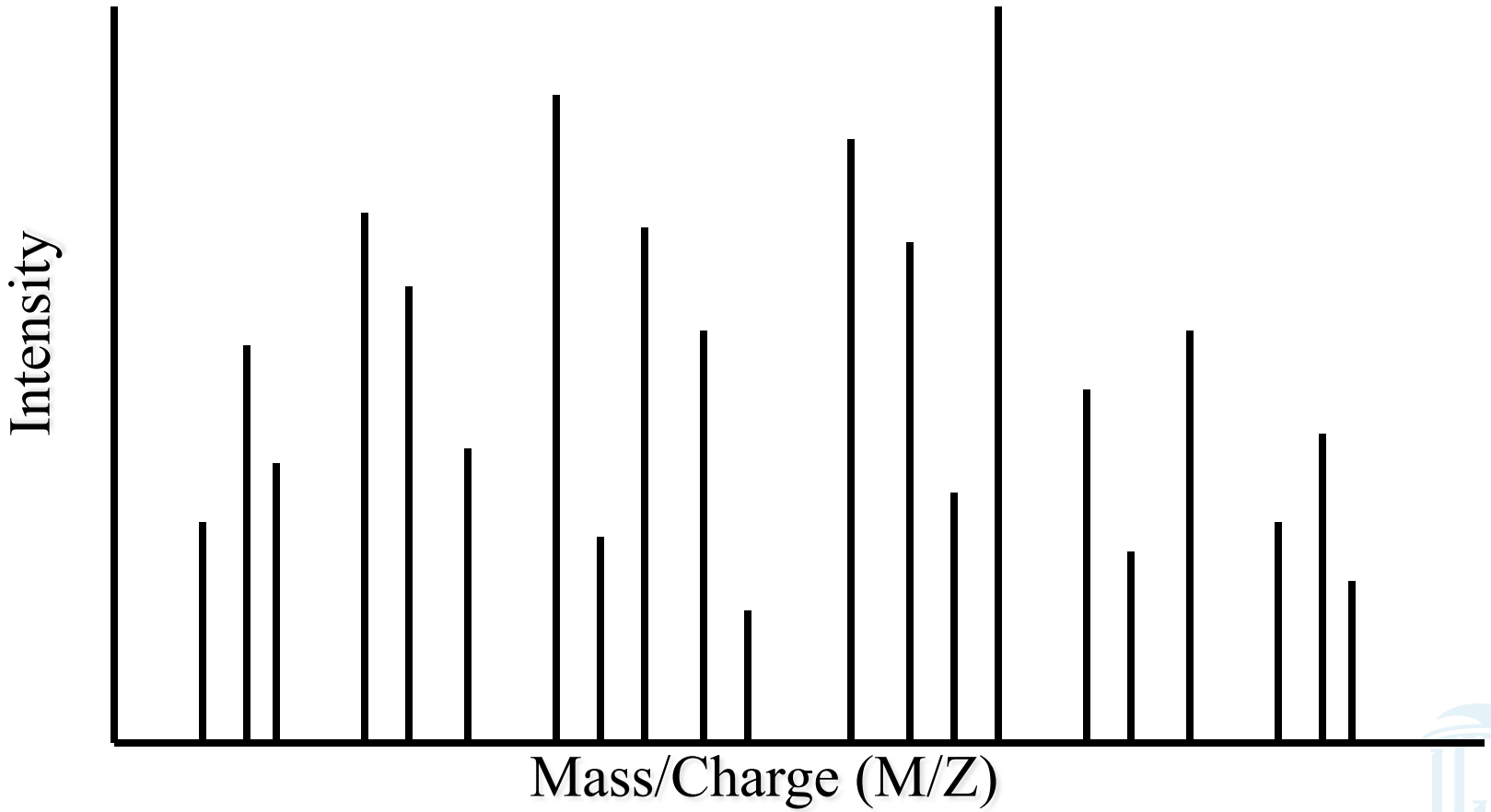
a is an ion type shift in *b*



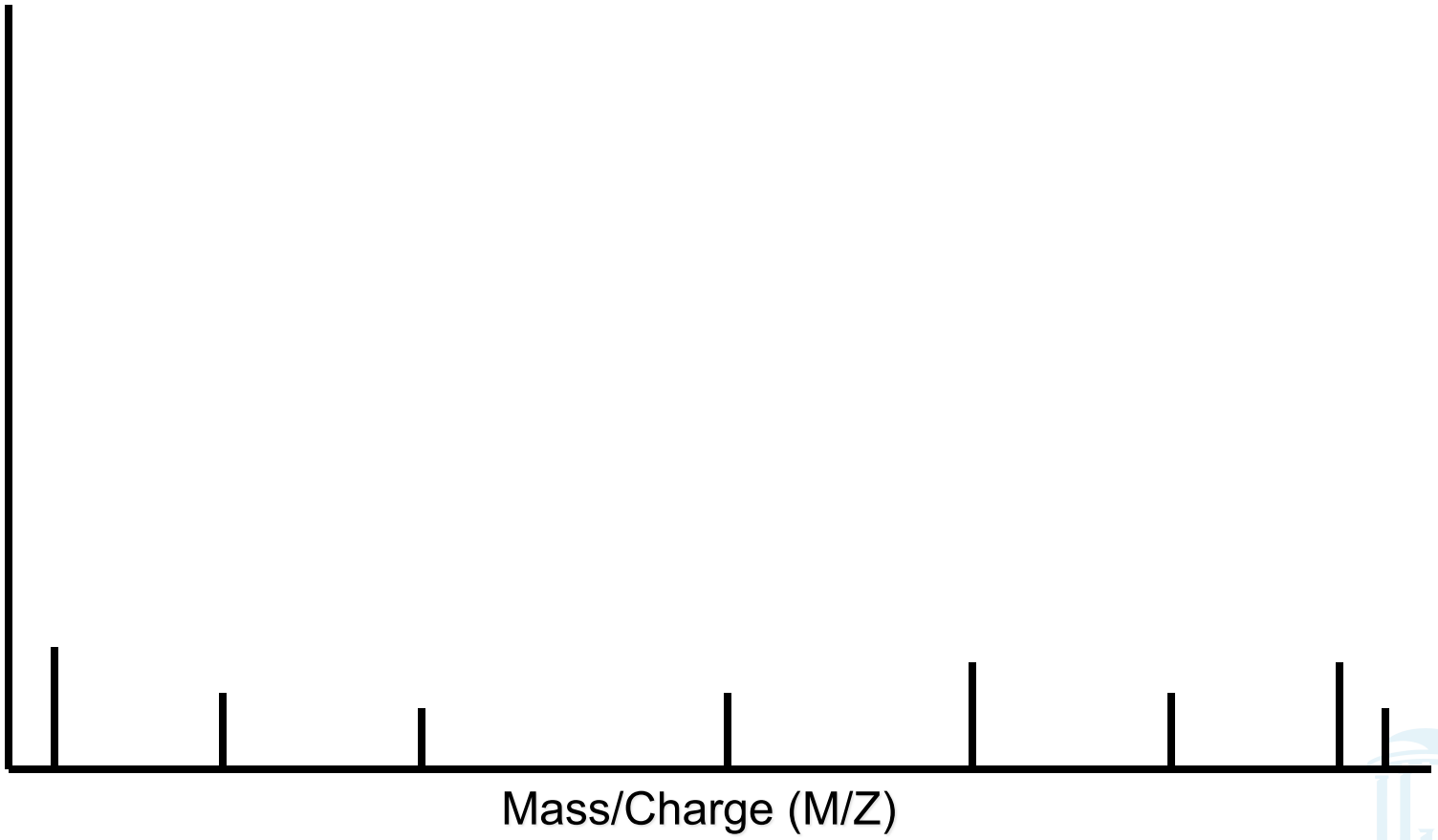

y
E C N E U Q E S
Mass/Charge (M/Z)



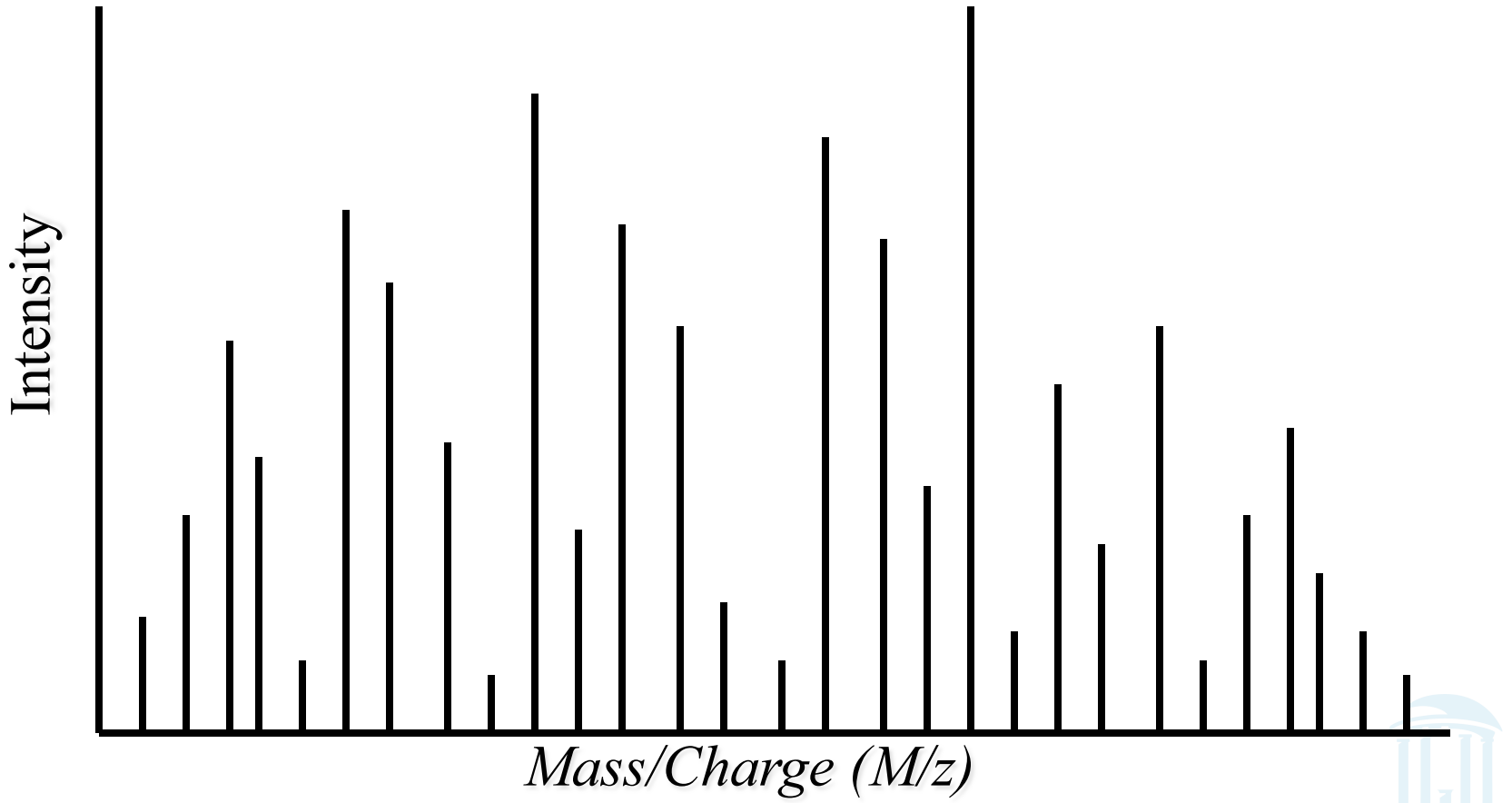

Intensity
Mass/Charge (M/Z)


Intensity
Mass/Charge (M/Z)

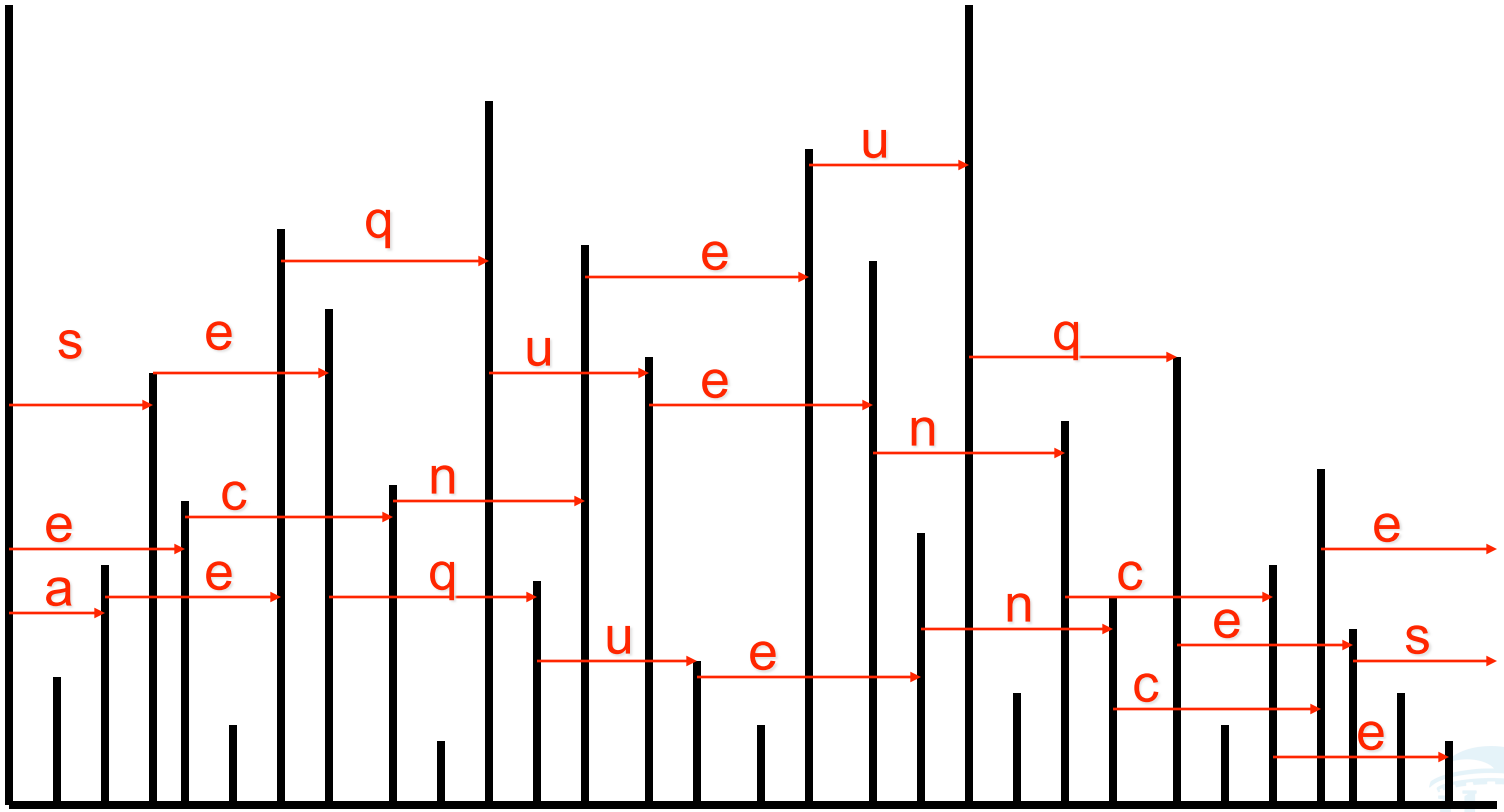
noise



MS/MS Spectrum



Some Mass Differences between Peaks Correspond to Amino Acids



Ion Types

- Some masses correspond to fragment ions, others are just random noise
- Known ion types $\Delta=\{\delta_1, \delta_2,\ldots, \delta_k\}$ allow us distinguish fragment ions from noise
- We can learn ion types δ_i and their probabilities q_i by analyzing a large test sample of annotated spectra.

Example of Ion Type

- $\Delta=\{\delta_1, \delta_2,\ldots, \delta_k\}$
- Ion types

$$\{b, b\text{-}NH_3, b\text{-}H_2O\}$$

correspond to

$$\Delta=\{0, 17, 18\}$$

*Note: In reality the δ value of ion type *b* is -1 but we will "hide" it for the sake of simplicity

Matching Spectra

- The match between two spectra is the number of masses (peaks) they share (**Shared Peak Count or SPC**)
- In practice mass-spectrometrists use the weighted SPC that reflects intensities of the peaks
- Match between experimental and theoretical spectra is defined similarly

Peptide Sequencing Problem

Goal: Find a peptide with maximal match between an experimental and theoretical spectrum.

Input:

- S: experimental spectrum
- Δ: set of possible ion types
- m: parent mass

Output:

- P: peptide with mass m, whose theoretical spectrum best matches the experimental S spectrum

Vertices of Spectrum Graph

- Masses of potential N-terminal peptides
- Vertices are generated by reverse shifts corresponding to ion types

$$\Delta=\{\delta_1, \delta_2,\ldots, \delta_k\}$$

- Every N-terminal peptide can generate up to k ions

$$m\text{-}\delta_1, m\text{-}\delta_2, \ldots, m\text{-}\delta_k$$

- Every mass s in an MS/MS spectrum generates k vertices

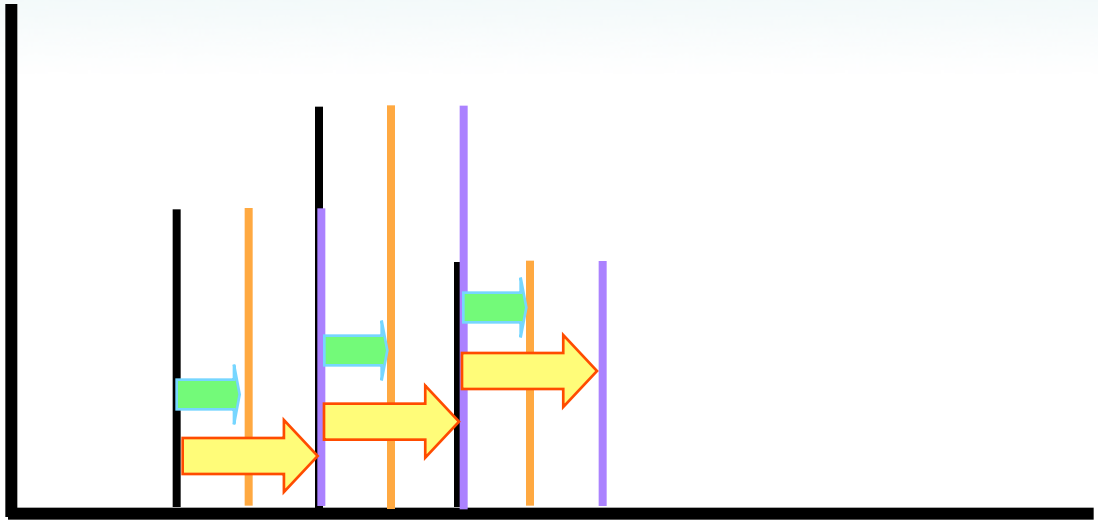
$$V(s) = \{s+\delta_1, s+\delta_2, \ldots, s+\delta_k\}$$

corresponding to potential N-terminal peptides

- **Vertices of the spectrum graph:**

$$\{initial\ vertex\} \cup V(s_1) \cup V(s_2) \cup \ldots \cup V(s_m) \cup \{terminal\ vertex\}$$

Reverse Shifts



Shift in H_2O

Shift in H_2O+NH_3

Edges of Spectrum Graph

- Two vertices with mass difference corresponding to an amino acid *A*:
  - Connect with an edge labeled by *A*

Paths

- Paths in the labeled graph spell out amino acid sequences
- There are many paths, how to find the correct one?
- We need scoring function to evaluate paths

Path Score

- $p(\boldsymbol{P},\boldsymbol{S})$ = probability that peptide $\boldsymbol{P}$ produces spectrum $\boldsymbol{S}= \{s_1,s_2,\ldots s_q\}$
- $p(\boldsymbol{P}, s)$ = the probability that peptide P generates a peak s
- Scoring = computing probabilities
- $p(\boldsymbol{P},\boldsymbol{S}) = \prod_{s \epsilon \boldsymbol{S}} p(\boldsymbol{P}, s)$

Peak Score

- For a position ***t*** that represents ion type d_j:

$$p(\boldsymbol{P}, s_t) = \begin{cases} q_j, & \text{if peak is generated at } \boldsymbol{t} \\ 1\text{-}q_j, & \text{otherwise} \end{cases}$$

Peak Score (cont'd)

- For a position ***t*** that is not associated with an ion type:

$$p_R(\boldsymbol{P}, s_t) = \begin{cases} q_R, & \text{if peak is generated at } \boldsymbol{t} \\ 1\text{-}q_R, & \text{otherwise} \end{cases}$$

- q_R = the probability of a noisy peak that does not correspond to any ion type

Optimal Paths in the Spectrum Graph

- For a given MS/MS spectrum ***S***, find a peptide ***P′*** maximizing $p(\boldsymbol{P},\boldsymbol{S})$ over all peptides ***P***:

$$p(P',S) = \max_P \; p(P,S)$$

- Peptides = paths in the spectrum graph

- ***P′*** = the optimal path in the spectrum graph

Ions and Probabilities

- Tandem mass spectrometry is characterized by a set of ion types $\{\delta_1,\delta_2,..,\delta_k\}$ and their probabilities $\{q_1,...,q_k\}$

- δ_i−ions of a partial peptide are produced ***independently*** with probabilities q_i

Ions and Probabilities

- A peptide has all k peaks with probability $\prod_{i=1}^{k} q_i$
- and no peaks with probability $\prod_{i=1}^{k} (1 - q_i)$
- A peptide also produces a "random noise" with ***uniform*** probability q_R in any position.

Ratio Test Scoring for Partial Peptides

- Incorporates **premiums** for observed ions and **penalties** for missing ions.

- Example: for k=4, assume that for a partial peptide P' we only see ions $\delta_1, \delta_2, \delta_4$.

  The score is calculated as: $\frac{q_1}{q_R} \cdot \frac{q_2}{q_R} \cdot \frac{(1-q_3)}{(1-q_R)} \cdot \frac{q_4}{q_R}$

Scoring Peptides

- T- set of all positions.
- T_i=$\{t_{\delta 1}, t_{\delta 2}, \ldots, t_{\delta k}\}$- set of positions that represent ions of partial peptides P_i.
- A peak at position $t_{\delta j}$ is generated with probability q_j.
- R=T- $(\cup T_i)$ - set of positions that are not associated with any partial peptides (noise).

Probabilistic Model

- For a position $t_{\delta j} \in T_i$ the probability $p(t, P, S)$ that peptide P produces a peak at position t.

$$P(t,P,S) = \begin{cases} q_j & \text{if a peak is generated at position } t_{\delta j} \\ 1-q_j & \text{otherwise} \end{cases}$$

- Similarly, for $t \in R$, the probability that P produces a random noise peak at t is:

$$P_R(t) = \begin{cases} q_R & \text{if a peak is generated at position t} \\ 1-q_R & \text{otherwise} \end{cases}$$

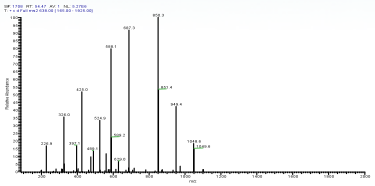
Probabilistic Score

- For a peptide P with n amino acids, the score for the whole peptides is expressed by the following ratio test:

$$\frac{p(P,S)}{p_R(S)} = \prod_{i=1}^{n} \prod_{j=1}^{k} \frac{p(t_{i\delta_j}, P, S)}{p_R(t_{i\delta_j})}$$

De Novo vs. Database Search

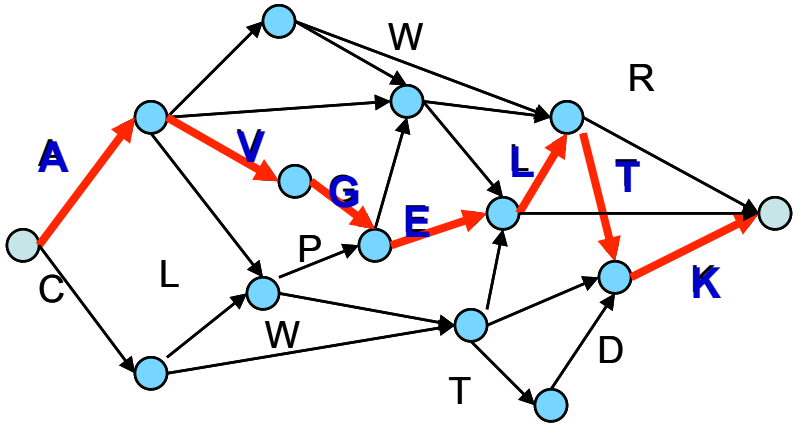
Database Search



De Novo

Database of known peptides

MDERHILNM, KLQWVCSDL, PTYWASDL, ENQIKRSACVM, TLACHGGEM, NGALPQWRT, HLLERTKMNVV, GGPASSDA, GGLITGMQSD, MQPLMNWE, AEKMMNRT, AVGELTK, HEWAILF, GHNLWAMNAC, GVFGSVLRA, EKLNKAATYIN..



AVGELTK

Peptide Identification Problem

Goal: Find a peptide *from the database* with maximal match between an experimental and theoretical spectrum.

Input:

- S: experimental spectrum
- *database of peptides*
- Δ: set of possible ion types
- m: parent mass

Output:

- A peptide of mass $\boldsymbol{m}$ *from the database* whose theoretical spectrum matches the experimental S spectrum the best

MS/MS Database Search

Database search in mass-spectrometry has been very successful in identification of **already known** proteins.

Experimental spectrum can be compared with theoretical spectra of database peptides to find the best fit.

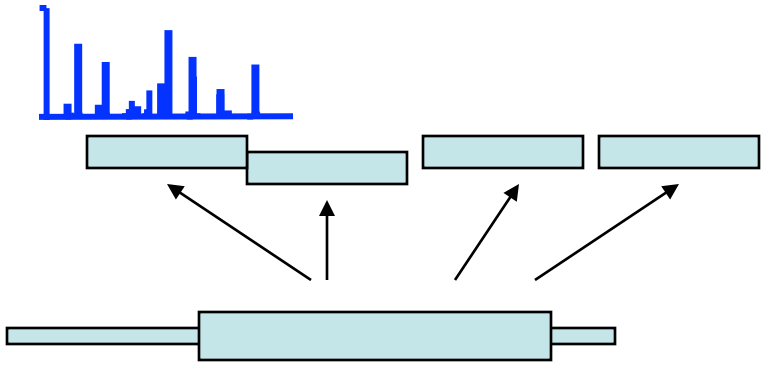
SEQUEST (Yates et al., 1995)

But reliable algorithms for identification of new protein forms via mutation is a much more difficult problem.

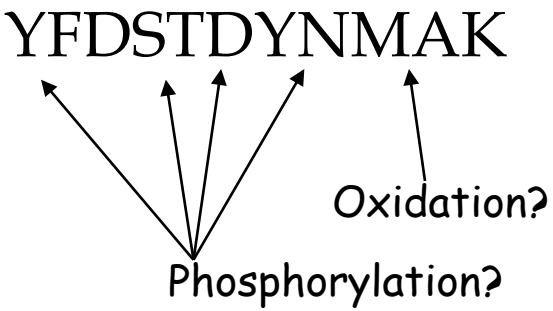
Modified Peptides

- Virtual Database Approach
- Yates et al.,1995: an exhaustive search in a virtual database of all modified peptides.
- Exhaustive search leads to a large combinatorial problem, even for a small set of modifications types.
- **Problem** (Yates et al.,1995). Extend the virtual database approach to a large set of modifications.

Exhaustive Search for modified peptides.



- For each peptide, generate all modifications.
- Score each modification.



- 2^5=32 possibilities, with 2 types of modifications!

Peptide Identification Challenge

Very similar peptides may have very different spectra!

Goal: Define a notion of spectral similarity that correlates well with the sequence similarity.

If peptides are a few mutations/modifications apart, the spectral similarity between their spectra should be high.

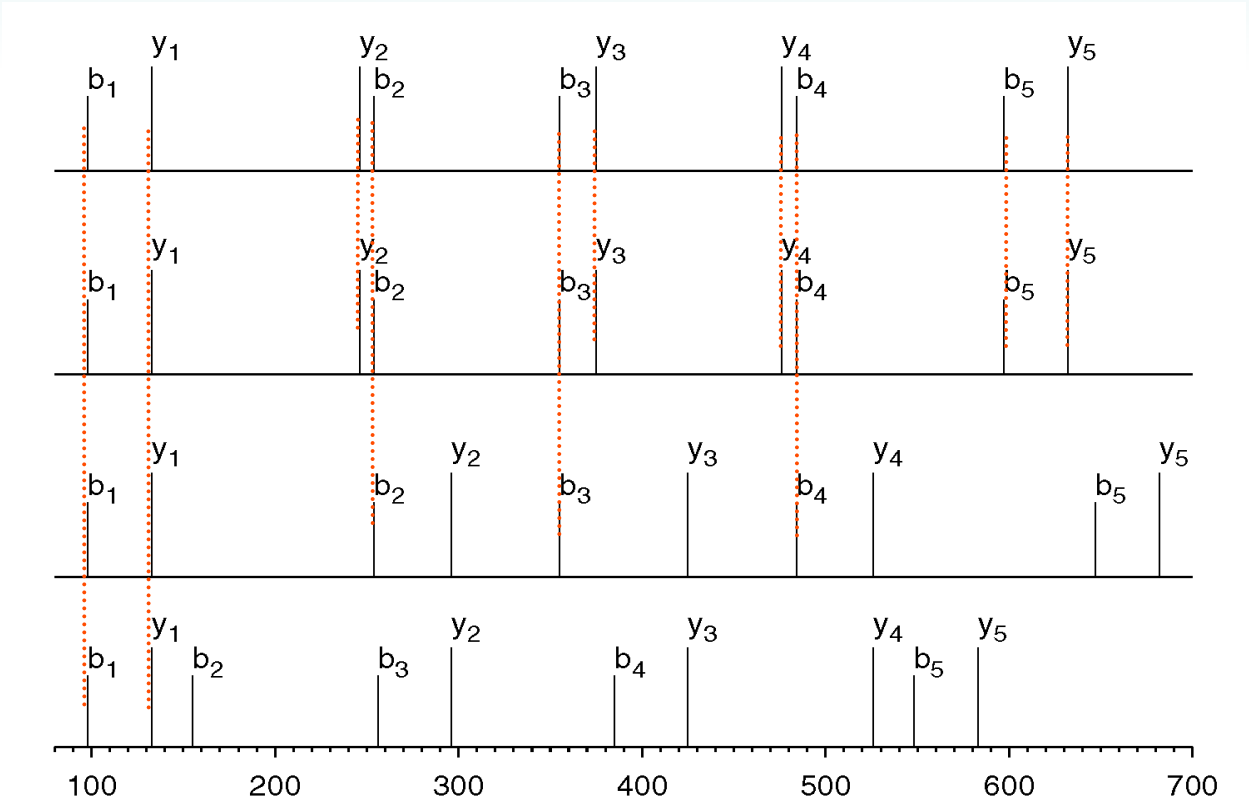
Deficiency of Shared Peaks Count

Shared peaks count (SPC): intuitive measure of spectral similarity.

Problem: SPC diminishes very quickly as the number of mutations increases.

Only a small portion of correlations between the spectra of mutated peptides is captured by SPC.

SPC Diminishes Quickly



no mutations
SPC=10

1 mutation
SPC=5

2 mutations
SPC=2

S(PRTEIN) = {98, 133, 246, 254, 355, 375, 476, 484, 597, 632}

S(PRTEYN) = {98, 133, 254, 296, 355, 425, 484, 526, 647, 682}

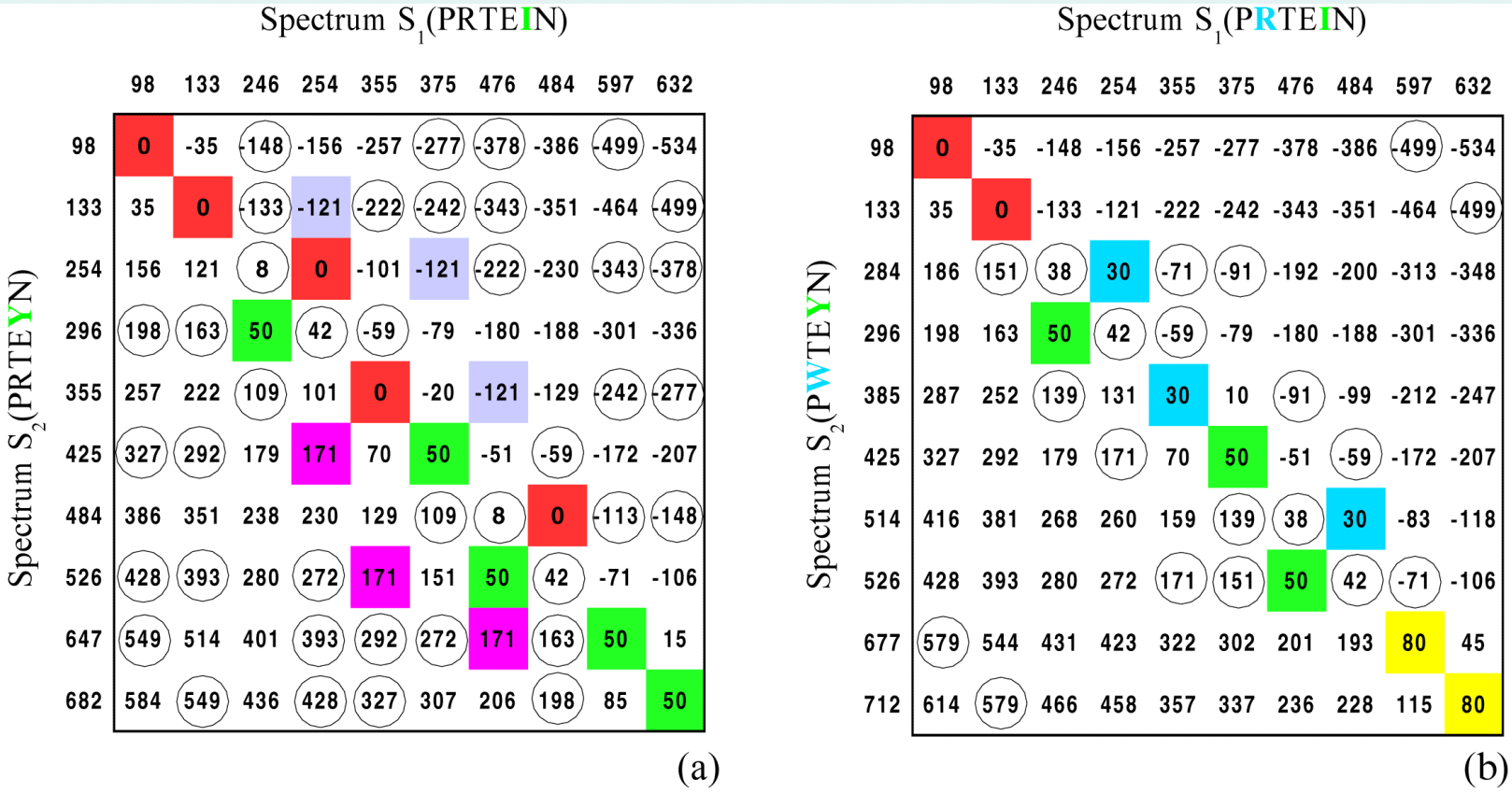
S(PGTEYN) = {98, 133, 155, 256, 296, 385, 425, 526, 548, 583}

Spectral Convolution

$$S_2 \ominus S_1 = \{s_2 - s_1 : s_1 \in S_1, s_2 \in S_2\}$$

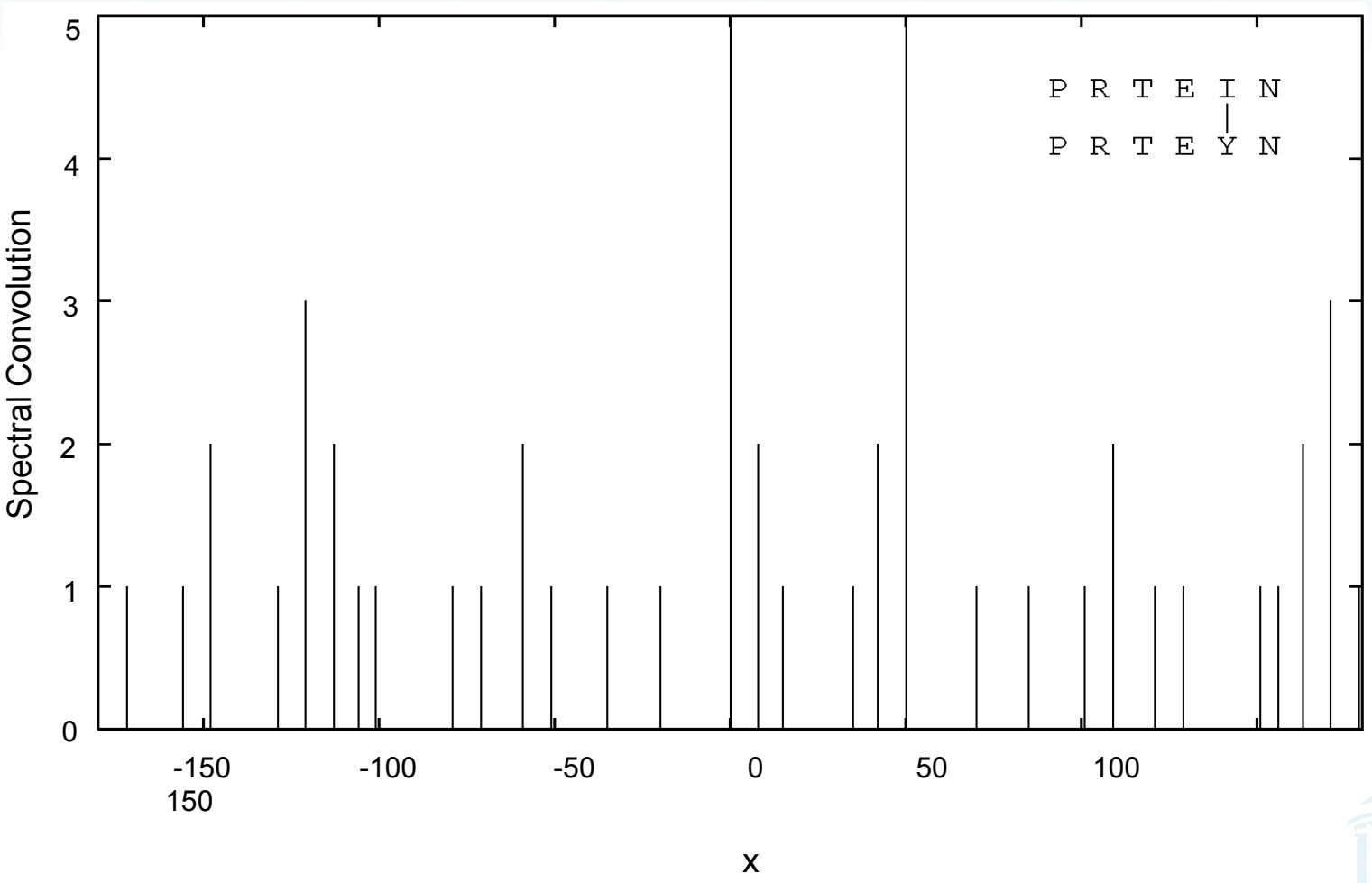
Number of pairs $s_1 \in S_1, s_2 \in S_2$ with $s_2 - s_1 = x$:
$(S_2 \ominus S_1)(x)$

The shared peaks count (SPC peak):
$(S_2 \ominus S_1)(0)$



Elements of $S_2 \ominus S_1$ represented as elements of a **difference matrix**. The elements with multiplicity >2 are colored; the elements with multiplicity =2 are circled. The SPC takes into account only the red entries

Spectral Convolution: An Example



Spectral Comparison: Difficult Case

S = {**10, 20, 30, 40, 50**, **60, 70, 80, 90, 100**}

Which of the spectra

S' = {**10, 20, 30, 40, 50**, 55, 65, 75,85, 95}

or

S'' = {**10**, 15, **30**, 35, **50**, 55, **70**, 75, **90**, 95}

fits the spectrum S the best?

SPC: both S' and S'' have 5 peaks in common with S.

Spectral Convolution: reveals the peaks at *0* and *5*.

Spectral Comparison: Difficult Case

$S \ominus S'$

S′ \ S	0	-10	-20	-30	-40	-50	-60	-70	-80	-90
10	**0**	-10	-20	-30	-40	-50	-60	-70	-80
20	10	**0**	-10	-20	-30	-40	-50	-60	-70
30	20	10	**0**	-10	-20	-30	-40	-50	-60
40	30	20	10	**0**	-10	-20	-30	-40	-50
45	35	25	15	**5**	**-5**	-15	-25	-35	-45
55	45	35	25	15	**5**	**-5**	-15	-25	-35
65	55	45	35	25	15	**5**	**-5**	-15	-25
75	65	55	45	35	25	15	**5**	**-5**	-15
85	75	65	55	45	35	25	15	**5**	**-5**

$S \ominus S''$

S′′ \ S	0	-10	-20	-30	-40	-50	-60	-70	-80	-90
0	-10	-20	-30	-40	-50	-60	-70	-80	-90
5	**-5**	-15	-25	-35	-45	-55	-65	-75	-85
20	10	**0**	-10	-20	-30	-40	-50	-60	-70
25	15	**5**	**-5**	-15	-25	-35	-45	-55	-65
40	30	20	10	**0**	-10	-20	-30	-40	-50
45	35	25	15	**5**	**-5**	-15	-25	-35	-45
60	50	40	30	20	10	**0**	-10	-20	-30
65	55	45	35	25	15	**5**	**-5**	-15	-25
80	70	60	50	40	30	20	10	**0**	-10
85	75	65	55	45	35	25	15	**5**	**-5**

Limitations

Spectral convolution does not reveal that spectra *S* and *S'* are similar, while spectra *S* and *S''* are not.

Clumps of shared peaks: the matching positions in *S'* come in clumps while the matching positions in *S''* don't.

This important property was not captured by spectral convolution.

Shifts

$A = \{a_1 < \ldots < a_n\}$: an ordered set of natural numbers.

A *shift* (i,Δ) is characterized by two parameters, the starting position (i) and the shift distance (Δ).

The shift (i,Δ) transforms

$$\{a_1, \ldots., a_n\}$$

into

$$\{a_1, \ldots.,a_{i-1},a_i+\Delta,\ldots,a_n+\Delta\}$$

Shifts: An Example

The shift (i,Δ) transforms $\{a_1, \ldots, a_n\}$

into $\{a_1, \ldots, a_{i-1}, a_i+\Delta, \ldots, a_n+\Delta\}$

e.g.

10 20 30 40 50 60 70 80 90

shift (4, -5)

10 20 30 35 45 55 65 75 85

shift (7,-3)

10 20 30 35 45 55 62 72 82

Spectral Alignment Problem

- Find a series of k shifts that make the sets

  $A=\{a_1, \ldots., a_n\}$ and $B=\{b_1, \ldots., b_n\}$

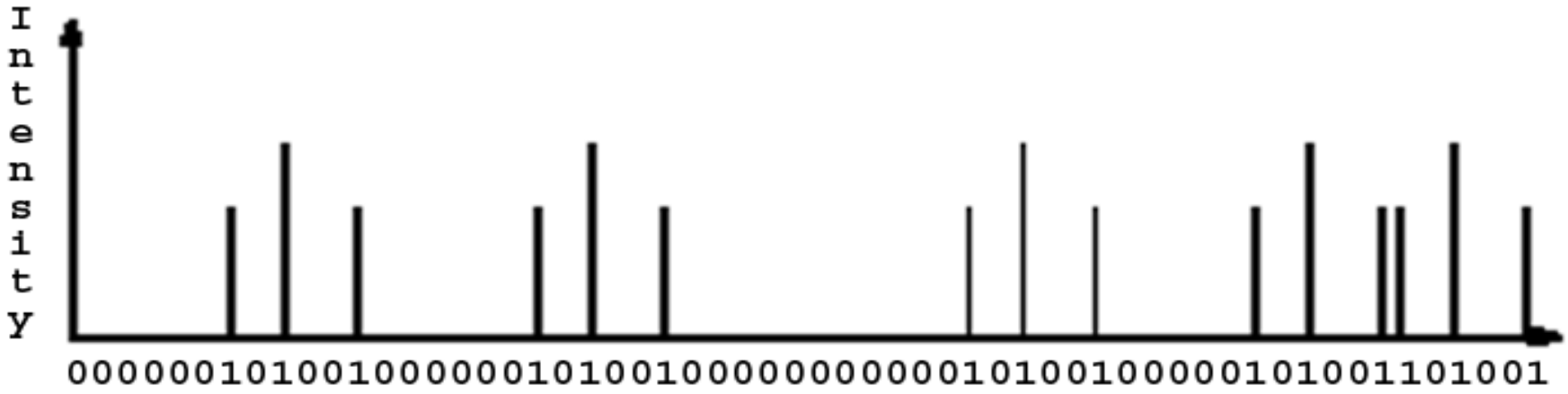
  as similar as possible.

- Provides a notion of **"k-similarity"** between sets

- $D(k)$ - the maximum number of elements in common between sets after k shifts (Like SPC).

Representing Spectra in 0-1 Alphabet

- Quantize (bin) the mass dimension
- Convert spectrum to a 0-1 string with 1s corresponding to the positions of the peaks.



Comparing Spectra=Comparing 0-1 Strings

- A modification with positive offset corresponds to inserting a block of 0s
- A modification with negative offset corresponds to deleting a block of 0s
- Comparison of theoretical and experimental spectra (represented as 0-1 strings) corresponds to a (somewhat unusual) **edit distance/alignment** problem where elementary edit operations are insertions/deletions of blocks of 0s
- **Use sequence alignment algorithms!**

Spectral Alignment vs. Sequence Alignment

- Manhattan-like graph with different alphabet and scoring.
- Movement can be diagonal (matching masses) or horizontal/vertical (insertions/deletions corresponding to PTMs).
- At most k horizontal/vertical moves.

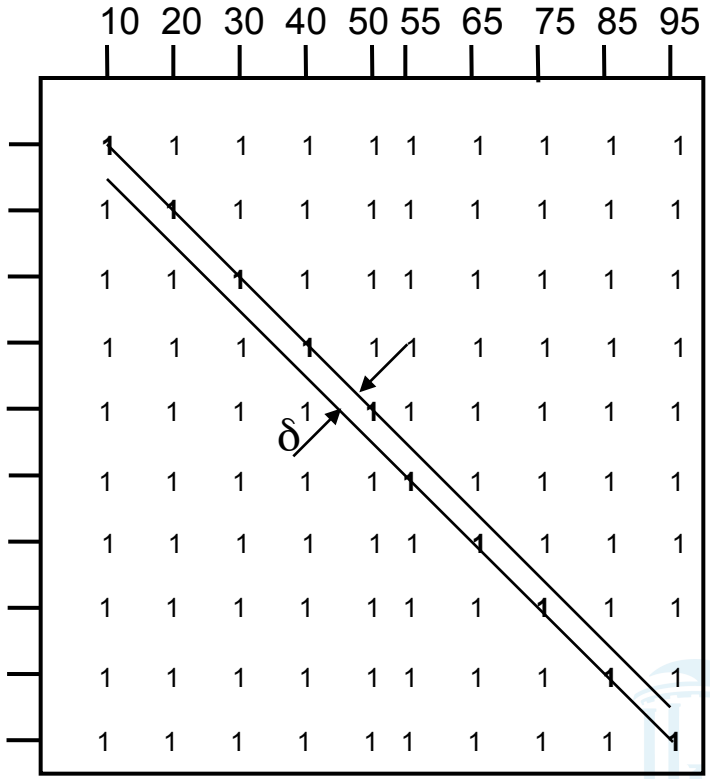
Spectral Product

$A=\{a_1, \ldots, a_n\}$ and $B=\{b_1, \ldots, b_n\}$

Spectral product $A \otimes B$: two-dimensional matrix with nm 1s corresponding to all pairs of indices (a_i, b_j) and remaining elements being 0s.

SPC: the number of 1s at the main diagonal.

δ-shifted SPC: the number of 1s on the diagonal $(i, i+\delta)$

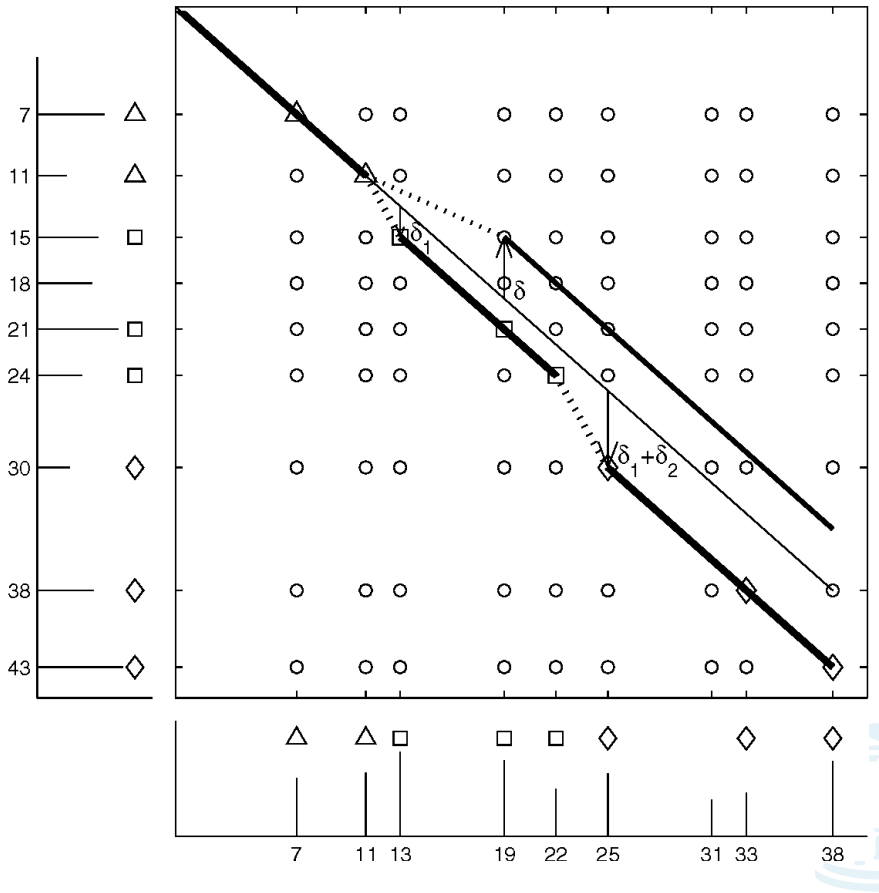


Spectral Alignment: *k*-similarity

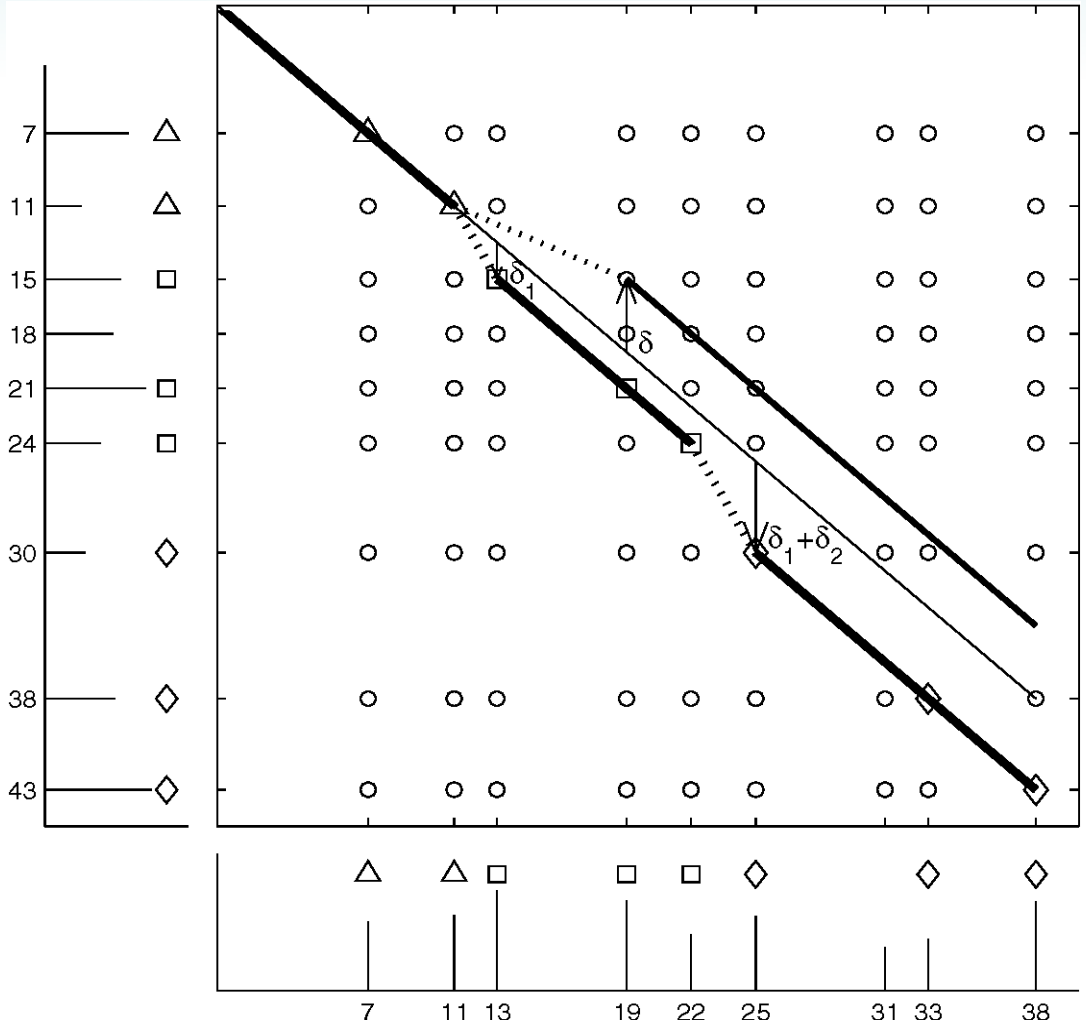
***k*-similarity between spectra**: the maximum number of 1s on a path through this graph that uses at most *k+1* diagonals.

***k*-optimal spectral alignment = a path.**

The spectral alignment allows one to detect more and more subtle similarities between spectra by increasing ***k***.

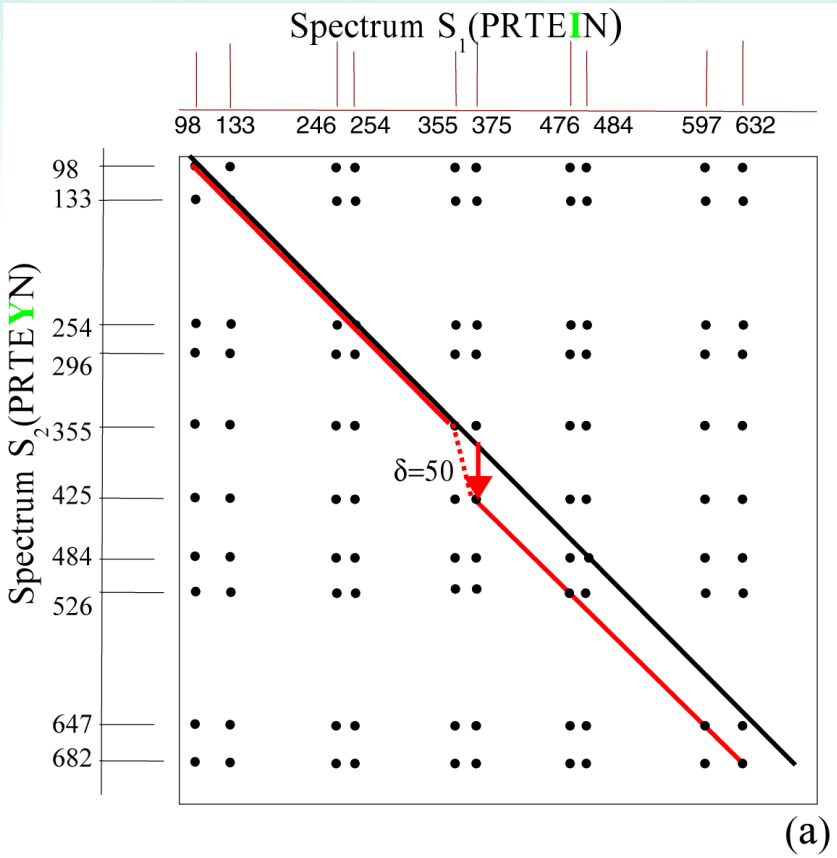


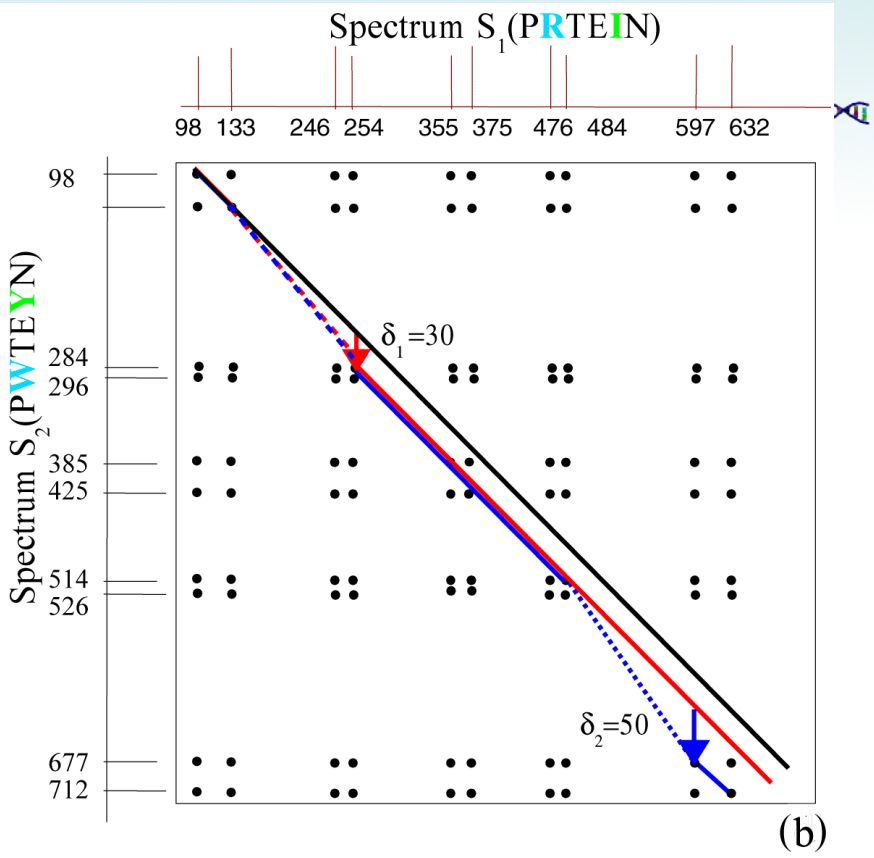
Use of k-Similarity



SPC reveals only $D(0)=3$ matching peaks.

Spectral Alignment reveals more hidden similarities between spectra: $D(1)=5$ and $D(2)=8$ and detects corresponding mutations.





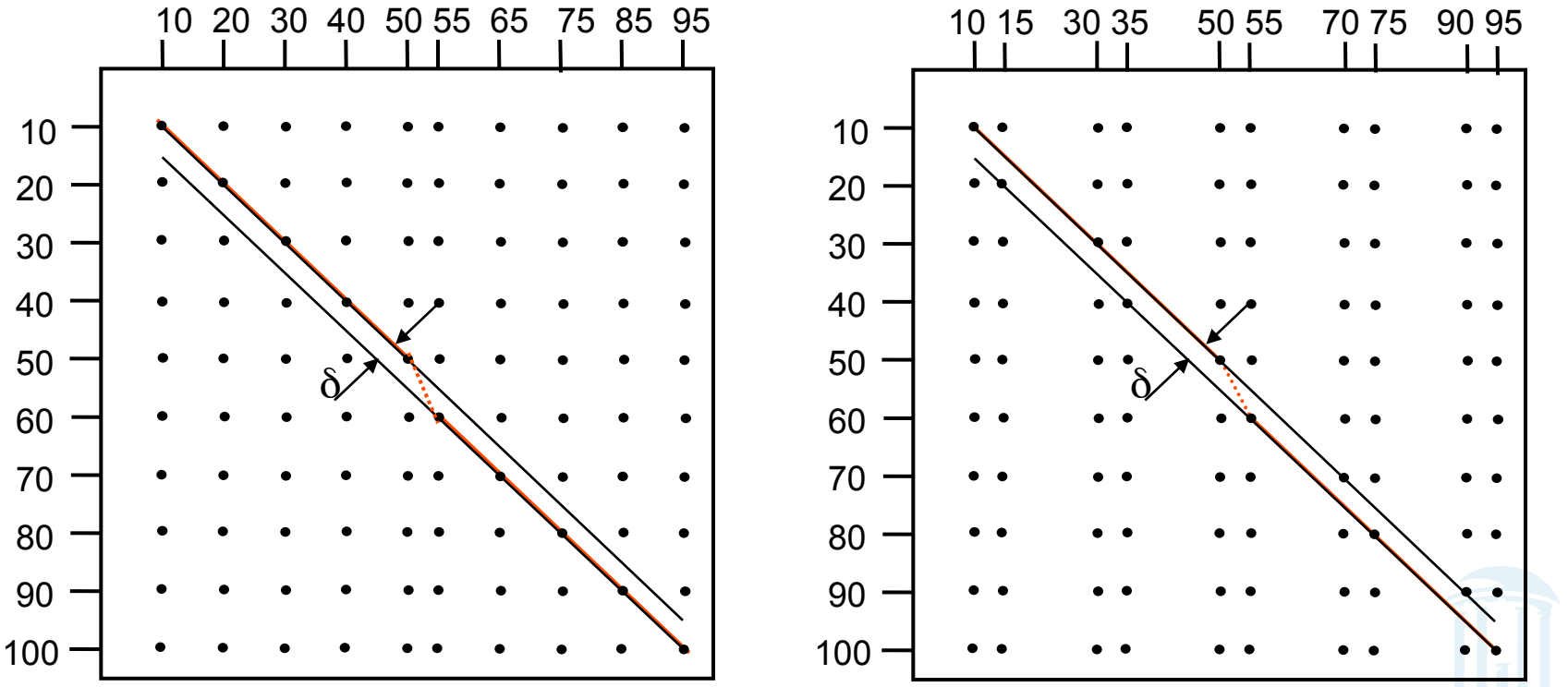
Black line represent the path for ***k=0***

Red lines represent the path for ***k=1***

Blue lines (right) represents the path for ***k=2***

Spectral Convolution's Limitation

The spectral convolution considers diagonals separately without combining them into feasible mutation scenarios.



D(1) =10 shift function score = 10 D(1) =6

Dynamic Programming for Spectral Alignment

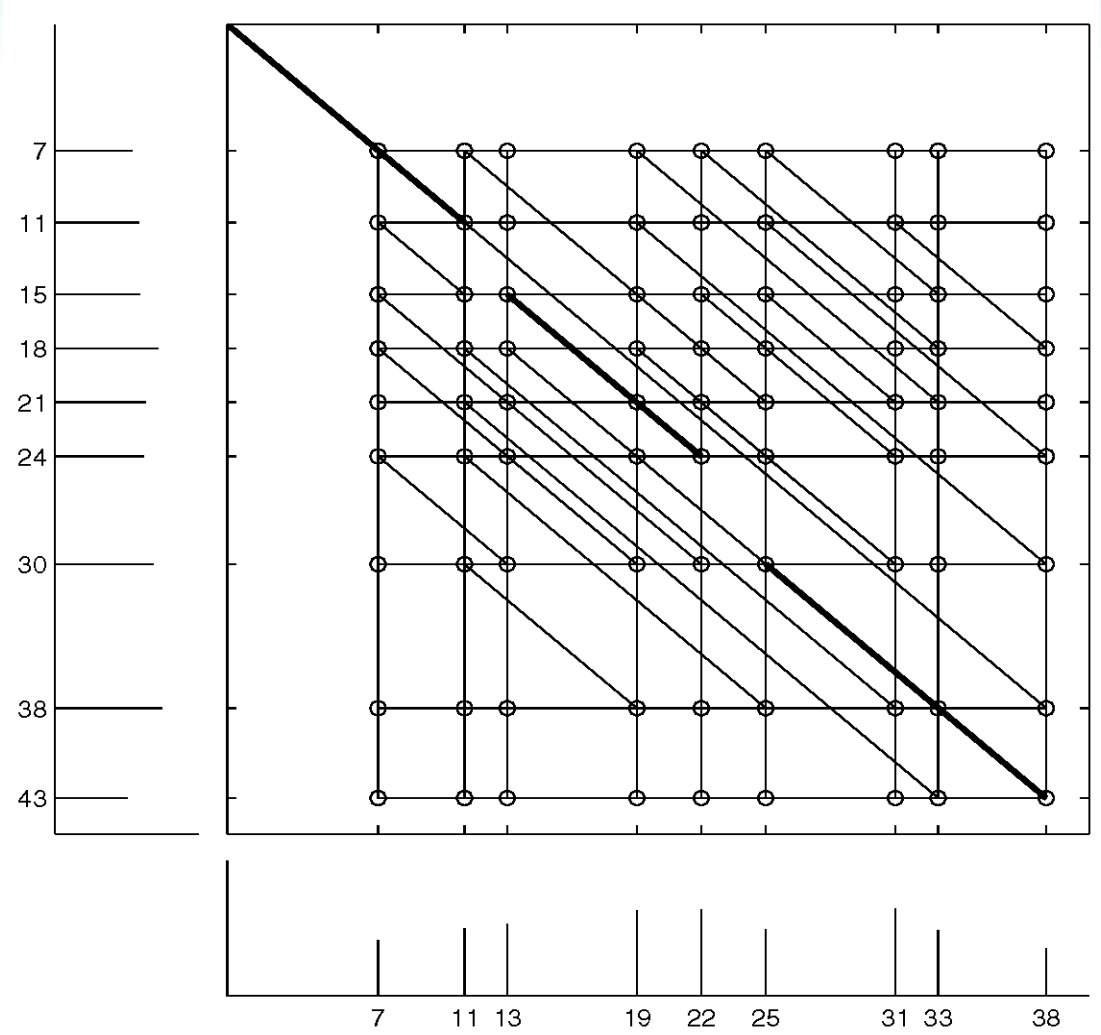
$D_{ij}(k)$: the maximum number of 1s on a path to (a_i, b_j) that uses at most $k+1$ diagonals.

$$D_{ij}(k) = \max_{(i',j')<(i,j)} \begin{cases} D_{i'j'}(k) + 1, & \text{if } (i',j') \sim (i,j) \\ D_{i'j'}(k-1) + 1, & \text{otherwise} \end{cases}$$

$$D(k) = \max_{ij} D_{ij}(k)$$

Running time: $O(n^4 k)$

Edit Graph for Fast Spectral Alignment



diag(i,j) – the position of previous 1 on the same diagonal as *(i,j)*

Fast Spectral Alignment Algorithm

$$M_{ij}(k) = \max_{(i',j')<(i,j)} D_{i'j'}(k)$$

$$D_{ij}(k) = \max \begin{cases} D_{diag(i,j)}(k) + 1 \\ M_{i-1,j-1}(k-1) + 1 \end{cases}$$

$$M_{ij}(k) = \max \begin{cases} D_{ij}(k) \\ M_{i-1,j}(k) \\ M_{i,j-1}(k) \end{cases}$$

Running time: $O(n^2 k)$

Spectral Alignment: Complications

Spectra are combinations of an increasing (N-terminal ions) and a decreasing (C-terminal ions) number series.

These series form two diagonals in the spectral product, the main diagonal and the perpendicular diagonal.

The described algorithm deals with the main diagonal only.

Spectral Alignment: Complications

- Simultaneous analysis of N- and C-terminal ions
- Taking into account the intensities and charges
- Analysis of minor ions