

# Bioinformatik

## Substitutionsmatrizen

Ulf Leser

Wissensmanagement in der  
Bioinformatik



# Alignment mit linearem Platzbedarf

---

- Bisherige Algorithmen haben  $O(m \cdot n)$  Zeit- und  $O(m \cdot n)$  Platzbedarf
  - Woher kommt der Platzbedarf?
- Platzbedarf ist ein Problem für Alignierung großer Sequenzen (Genom-Genom)
- Gesucht: **Speicherplatzeffizienterer Algorithmus**

# Strings und reverse Strings

---

- Definition

- Mit  $A^r$  bezeichnen wir das *Reverse eines Strings A*
- $A^r[1..i]$  bezeichnet die ersten  $i$  Zeichen von  $A^r$ 
  - Also die letzten  $i$  Zeichen von  $A$
- Für Strings  $A, B$  sei  $v^r(i, j) = v(A^r[1..i], B^r[1..j])$

- Bemerkung

- Es gilt :  $v^r(i, j) = v(A[n-i+1..n], B[m-j+1..m])$ 
  - Denn:  $v(A^r, B^r) = v(A, B)$
- Berechnung von  $v^r$  kann exakt wie die von  $v$  erfolgen

A . . . . . ATGCGGT  
B . . . . . GGTCGTAG  
  
A<sup>r</sup> TGGCGTA . . . . .  
B<sup>r</sup> GATGCTGG . . . . .

# Problemhalbierung

---

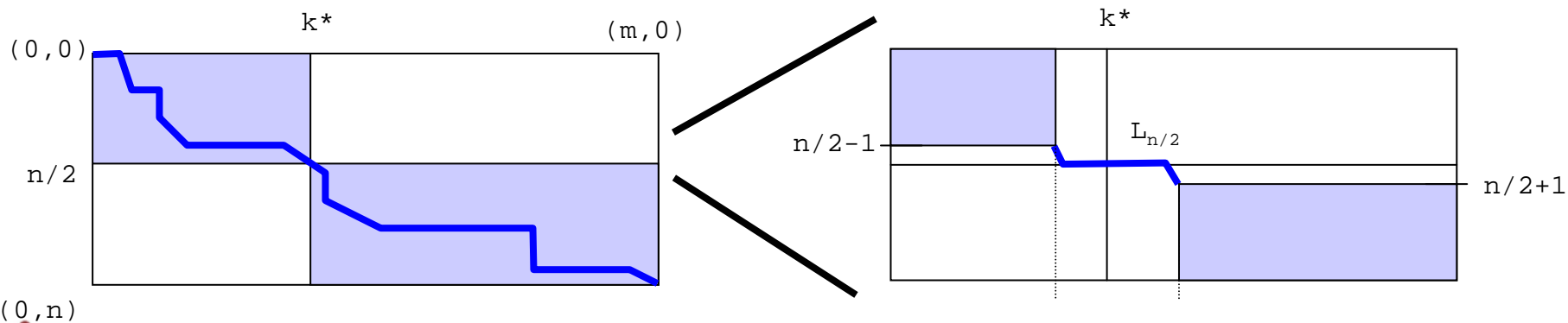
- Lemma  
*Gegeben zwei Strings A, B*

$$v(n, m) = \max_{0 \leq k \leq m} (v(n/2, k) + v^r(n/2, m - k))$$

- Beweisidee / Intuition
  - Wir alignieren
    - A[1..n/2] mit B[1..k] **vorwärts**
    - A[n/2+1..n] mit B[k+1..m] **rückwärts**
  - Im optimalen Alignment muss irgendein Präfix von B mit A[1..n/2] **alignieren**
  - Durch das laufende k erwischen wir diesen Punkt auf jeden Fall
- Bemerkung
  - Das **Problem wird also bzgl. |A| halbiert**

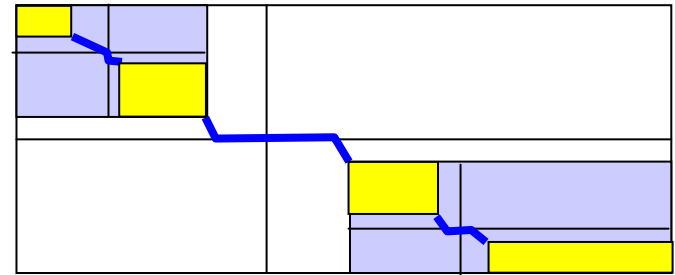
# Teilpfade

- Definition
  - Sei  $k^*$  das  $k$ , für das das Maximum  $v(n, m)$  erreicht wird
  - Sei  $L$  der Pfad von  $(0, 0)$  bis  $(n, m)$
  - Sei  $L_{n/2}$  der Pfad zwischen dem letzten Knoten in der Zeile  $n/2-1$  und dem ersten Knoten in  $n/2+1$
- Lemma
  - $L$  und  $L_{n/2}$  müssen  $k^*$  enthalten
- Beweisidee:  $L$  muss irgendwo die Zeile  $n/2$  passieren



# Rekursion

- Wo sind wir?
  - Wir halten die gewünschten Komplexitätsschranken
  - Wir haben das Mittelstück von L berechnet
- Damit können wir **rekursiv absteigen**
  - Löse rekursiv die Probleme
    - für  $A[1..n/2] / B[1..k^*]$  und
    - für  $A[n/2+1..n] / B[k^* + 1..m]$
- Platzkomplexität steigt nicht
- **Zeitkomplexität** steigt auch nicht
  - Zeitbedarf steigt um Faktor 2



# K-Band Algorithmus

---

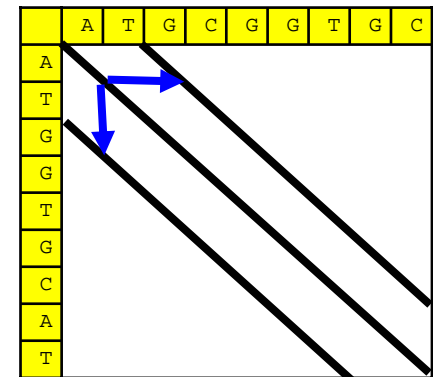
- Den Platzbedarf haben wir erfolgreich reduziert
- Kann man auch was an der Laufzeit machen?
  - An der Komplexität zum Finden der optimalen Lösung in allen Fällen: Nein
  - Aber oft sucht man nur **besonders gute Alignments**
    - Z.B. Forensik, Vaterschaftstests, Suche nach homologen Gensequenzen
    - Ein Treffer ist entweder fast identisch oder uninteressant

# K-Band Algorithmus

- Ansatz: Wir erlauben ein **Abweichungen nur um  $+k/-k$  Schritte**
- Algorithmus
  - Berechnet das beste globale Alignment innerhalb des Bandes der Breite  $2*k$

```
for i= 1 to n do
  for j= i-k to i+k do
    if (j<1) or (j>n) continue;
    M[i,j]= M[i-1,j-1] + t(A[i],B[j]);
    if inband(i-1,j) then
      M[i,j]= max( M[i,j], M[i-1,j]+b);
    if inband(i,j-1) then
      M[i,j]= max( M[i,j], M[i,j-1]+b);
  end for;
end for;
return M[n,n]
```

Beispiel:  $k=2$



# Optimalität

---

- Theorem

*Gegeben Strings  $A, B$  mit  $n=|A|=|B|$ . Sei  $v_K(A,B)$  der optimale  $K$ -Band Score für  $A$  und  $B$ . Wenn*

*$v_K(A,B) \geq s^*(n-k-1) + 2b^*(k+1)$ , dann ist  $v_K(A,B) = v(A,B)$ .*

- Beweis

- Wenn das optimale Alignment im  $k$ -Band verläuft, gilt auf alle Fälle  $v_K = v(A,B)$
- Wenn nicht, dann muss es irgendwo aus dem  $K$ -Band laufen. Im besten solchen Fall haben wir dann  $n-k-1$  Matches und dann  $k+1$  Insertions zum Verlassen des Bandes und weitere  $k+1$  Deletions, um am Ende noch  $(n,n)$  zu erreichen

# Iteratives K-Band

---

- Das können wir ausnutzen, um das optimale Alignment **iterativ** zu finden

```
k = 1;
while (true) do
    compute  $v_k$ ; // Costs  $O(k*n)$ 
    if  $v_k \geq s(n-k-1)+2b(k+1)$  then
        return  $v_k$ ;
    else
        k = 2*k;
    end if;
end while;
```

# Komplexität

---

- Theorem.  
*Sei  $v=v(A,B)$ . Der iterative K-Band Algorithmus benötigt  $O(sn^2-vn)$  Laufzeit.*
- Beweis
  - Beachte:  $v_k$  wird mit wachsendem  $k$  nie kleiner
  - Der Algorithmus stoppt, wenn  $v_k \geq s(n-k-1)+2b(k+1)$ , also 
$$k \geq \frac{sn - v_k}{s - 2b} - 1$$
  - Bis dahin wurden  $O(1n+2n+4n+\dots+kn) \sim O(2kn)$  Berechnungen durchgeführt
  - Wenn wir bei  $k$  stoppen, dann kann bei  $k/2$  die Abbruchbedingung noch nicht erfüllt gewesen sein, und damit gilt:

$$\frac{k}{2} < \frac{sn - v_{k/2}}{s - 2b} - 1$$

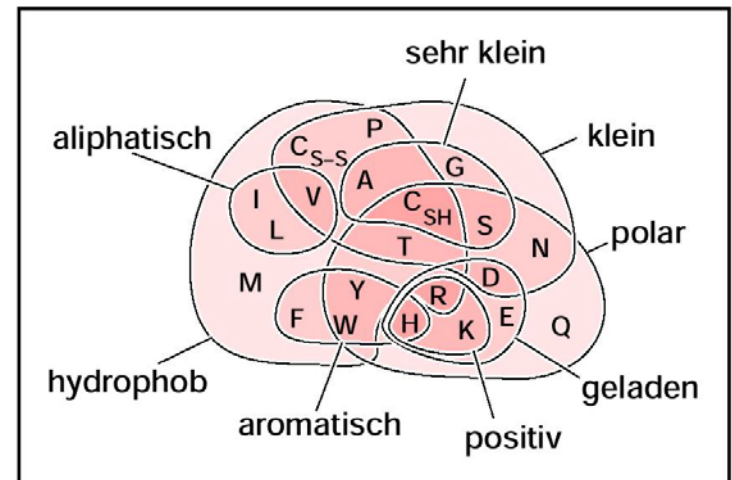
# Inhalt dieser Vorlesung

---

- Substitutionsmatrizen
  - PAM: Point-Accepted Mutations
  - BLOSUM: Blocks Substitution Matrices

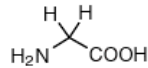
# Hintergrund

- Schon öfters angesprochen ...
  - Ähnlichkeitsmatrizen, Substitutionsmatrizen, Scorefunktionen, ...
- Ersetzung einer Base/Aminosäure durch eine andere hat **unterschiedliche Bedeutung**
  - Aminosäuren
    - Ersetzung mit „sehr ähnlichen“ Aminosäuren ändert Proteinstruktur kaum
    - Ersetzung mit „wenig ähnlichen“ Aminosäuren kann Struktur vollkommen ändern

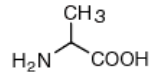


# Unterschiedliche Aminosäuren

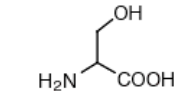
## Small



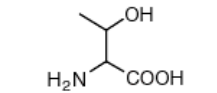
Glycine (Gly, G)  
MW: 57.05



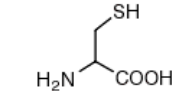
Alanine (Ala, A)  
MW: 71.09



Serine (Ser, S)  
MW: 87.08, pK<sub>a</sub> ~ 16

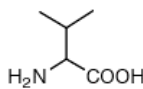


Threonine (Thr, T)  
MW: 101.11, pK<sub>a</sub> ~ 16

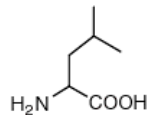


Cysteine (Cys, C)  
MW: 103.15, pK<sub>a</sub> = 8.35

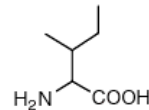
## Hydrophobic



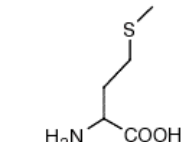
Valine (Val, V)  
MW: 99.14



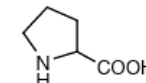
Leucine (Leu, L)  
MW: 113.16



Isoleucine (Ile, I)  
MW: 113.16

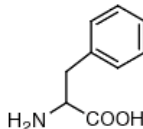


Methionine (Met, M)  
MW: 131.19

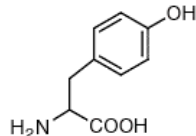


Proline (Pro, P)  
MW: 97.12

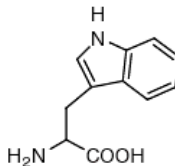
## Aromatic



Phenylalanine (Phe, F)  
MW: 147.18

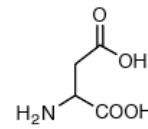


Tyrosine (Tyr, Y)  
MW: 163.18

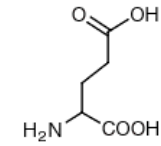


Tryptophan (Trp, W)  
MW: 186.21

## Acidic

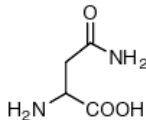


Aspartic Acid (Asp, D)  
MW: 115.09, pK<sub>a</sub> = 3.9

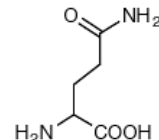


Glutamic Acid (Glu, E)  
MW: 129.12, pK<sub>a</sub> = 4.07

## Amide

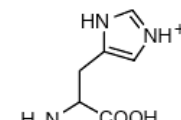


Asparagine (Asn, N)  
MW: 114.11

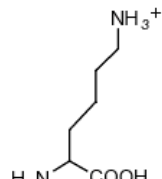


Glutamine (Gln, Q)  
MW: 128.14

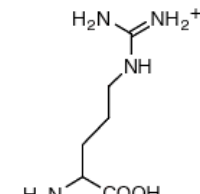
## Basic



Histidine (His, H)  
MW: 137.14, pK<sub>a</sub> = 6.04



Lysine (Lys, K)  
MW: 128.17, pK<sub>a</sub> = 10.79



Arginine (Arg, R)  
MW: 156.19, pK<sub>a</sub> = 12.48

# Substitutionsmatrizen

- Bewertung **individueller Substitutionen**
- Alignmentalgorithmus ist davon unberührt, nur Parameter ändern sich
- Das Ergebnis kann sich aber vollkommen ändern
- Beispiele: Blosom62, Identitätsmatrix

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0	-2	-1
R	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3	-1	0
N	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3	3	0
D	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3	4	1
C	0	-3	-3	-3	9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2	-1	-3	-3
Q	-1	1	0	0	-3	5	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2	0	3
E	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2	1	4
G	0	-2	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-2	-3	-3	-1	-2
H	-2	0	1	-1	-3	0	0	-2	8	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3	0	0
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-3	-1	3	-3	-3
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	2	2	0	-3	-2	-1	-2	-1	1	-4	-3
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2	0	1
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1	-3	-1
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6	-4	-2	-2	1	3	-1	-3	-3
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7	-1	-1	-4	-3	-2	-2	-1
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4	1	-3	-2	-2	0	0
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-2	-2	0	-1	-1
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11	2	-3	-4	-3
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1	-3	-2
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4	-3	-2
B	-2	-1	3	4	-3	0	1	-1	0	-3	-4	0	-3	-3	-2	0	-1	-4	-3	-3	4	1
Z	-1	0	0	1	-3	3	4	-2	0	-3	-3	1	-1	-3	-1	0	-1	-3	-2	-2	1	4

	A	C	G	T
A	5	-4	-4	-4
C	-4	5	-4	-4
G	-4	-4	5	-4
T	-4	-4	-4	5

# Ist das alles notwendig?

Code	Häufigkeit	Mutierbarkeit
L	0.091	54
A	0.077	100
G	0.074	50
S	0.069	117
V	0.066	98
E	0.062	77
K	0.059	72
T	0.059	107
I	0.053	103
D	0.052	86
P	0.051	58
R	0.051	83
N	0.043	104
Q	0.041	84
F	0.040	51
Y	0.032	50
M	0.024	93
H	0.023	91
C	0.020	44
W	0.014	25

- Häufigkeiten der einzelnen Aminosäuren im Verhältnis zur Gesamtzahl
- Häufigkeiten der Ersetzung einer Aminosäure im Verhältnis zu allen Ersetzungen
- Alanin (A) willkürlich als 100% gesetzt
- **Keinesfalls Gleichverteilung**
- Es gibt klar bevorzugte Mutationen
  - Besser: durch Selektion klar benachteiligte Mutationen
  - Tryptophan (W) sehr selten (25)
  - Serin (S) sehr häufig (117)

# Woher nehmen?

---

- Wie kann man sinnvolle Werte für die Matrix bestimmen?
  - Wir wollen **Ähnlichkeit der biologischen Bedeutung** messen
- Möglichkeit 1: Chemische Eigenschaften
  - Ladung, Größe, Polarität, ...
  - Viele Faktoren mit unklaren Gewichten
  - Wie soll man das durch **ein Bewertungsschema** ausdrücken?
  - Keine Verwendung in der Praxis
- Möglichkeit 2: Beobachtung
  - **Beobachtung der Evolution** statt analytischer Vorhersage
  - Lernen aus Beispielen, also „tatsächlich“ vorgekommener Mutationen
  - Benötigt große Menge homologe Sequenzen

# PAM: Point-Accepted Mutations

---

- Dayhoff et al., „A model of evolutionary change in proteins“, 1978
- PAM: Zwei Bedeutungen
  - 1 PAM – **Einheit** für den Abstand von Proteinsequenzen
  - PAM-X Matrix – Berechnete **Substitutionsmatrix** für zwei Sequenzen die X PAM entfernt sind

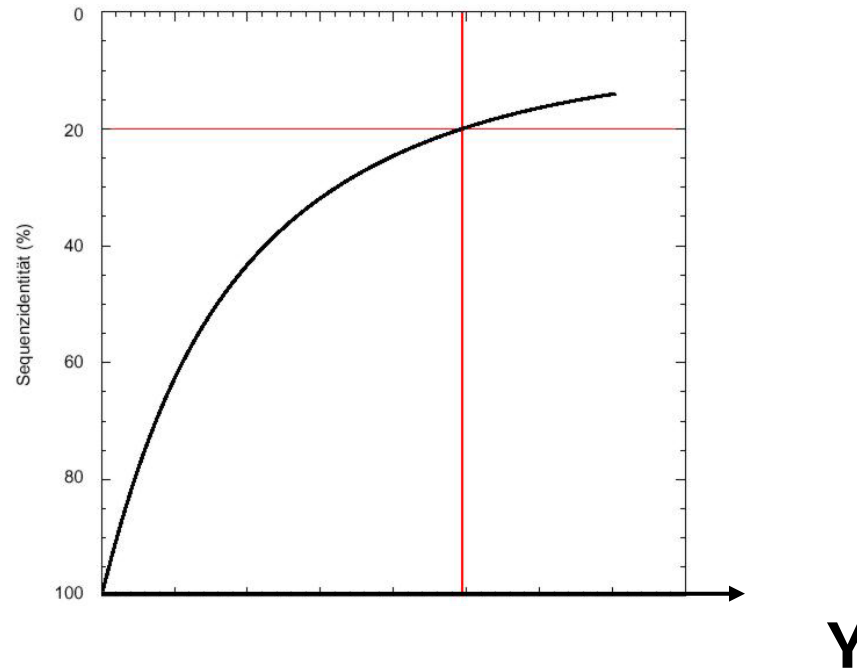
# Vorbetrachtung

---

- Annahme
  - Sequenz von 100 Aminosäuren an
  - **Mit jeder Generation** ändert sich mit Wahrscheinlichkeit  $p$  eine zufällig ausgewählte Position
- Betrachten wir  $y$  Generationen
  - Menge der **stattgefundenen Änderungen** korreliert mit  $y$  und  $p$ 
    - $p$  nehmen wir als konstant über Zeit und Position an
      - **Molecular Clock theory**
    - Wenn man Startpunkt und Ergebnis hat, kann man also  $y$  abschätzen
  - Mit welcher Wahrscheinlichkeit hat sich eine gegebene Position verändert?
  - Wie viele Positionen haben sich wahrscheinlich verändert?
  - Die Menge der **a-posteriori vorhandenen Änderungen** korreliert auch mit  $y$ , aber auf eine unterschiedliche Weise
    - Und wir wissen nur diese Änderungen

# Mutationshäufigkeit und Sequenzidentität

---



- Rückmutationen, Doppelmutationen, etc.
- Mit steigendem  $y$  wächst der a-priori Abstand ab einem bestimmten Punkt kaum noch
- In diesem Bereich kann man über [Homologie](#) nichts mehr sagen

# 1. PAM als Sequenzabstand

---

- Definition

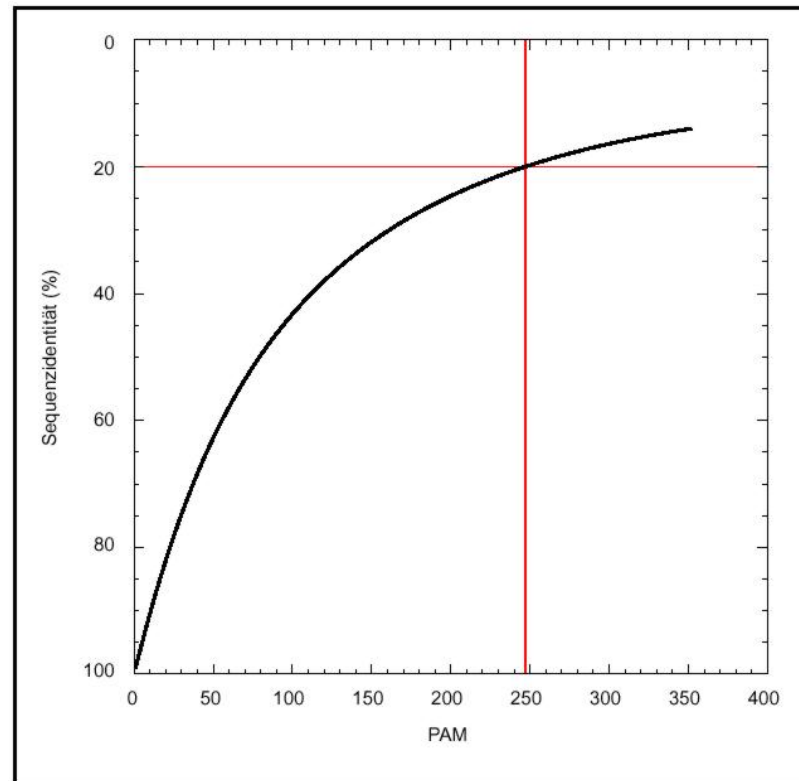
*Seien  $S_1$  und  $S_2$  zwei Proteinsequenzen.  $S_1$  und  $S_2$  heißen  $x$  PAM entfernt, wenn  $S_1$  in  $S_2$  überführt wurde mit  $x$  Punktmutationen pro 100 Aminosäuren*

- Eigenschaften

- PAM beachtet keine Inserts und Deletions
- $x$  schätzt man aus den a-posteriori Unterschieden von  $S_1$  und  $S_2$ 
  - Kann man analytisch rechnen: Gegeben Mutationswahrscheinlichkeit  $p$ ; mit welcher Wahrscheinlichkeit entstehen dadurch die beobachteten Unterschiede?
  - Oder durch Simulation bestimmen
- 50 PAM Abstand heißt also nicht 50 Veränderungen pro 100 Aminosäuren

# PAM Abstand und Sequenzidentität

---



- Jenseits von PAM 250: Rauschen

## 2. PAM Matrizen

---

- Seien  $(S_{1,1}, S_{2,1}), \dots, (S_{1,n}, S_{2,n})$  Paare von Sequenzen die jeweils  $x$  PAM entfernt sind. Dann berechnet sich die **PAM- $x$  Matrix  $M_x$**  wie folgt
  - Messe absolute Häufigkeit  $f(A_i)$  für alle Aminosäuren  $A_i$  in allen Sequenzen
  - Aligniere alle Paare entsprechend der evolutionären Wahrheit
    - $S_{k,l}'$  sei  $S_{k,l}$  mit den durch das Alignment eingefügten Leerzeichen
  - Messe **Übergangshäufigkeit  $f(i,j)$**  zwischen allen Paaren  $(A_i, A_j)$ , normiert auf Gesamtzahl aller Paare
    - Anzahl von Positionen  $k$  mit  $S_{1,z}'[k]=A_i$  und  $S_{2,z}'[k]=A_j$  über alle Positionen  $k$  in allen Paaren
    - Paare  $(A_x, \_)$  werden ignoriert
    - Übergang ist „richtungslos“;  $f(i,j) = f(j,i)$
  - Berechne Matrixelemente

$$M_x(i, j) = \log \left( \frac{f(i, j)}{f(i) * f(j)} \right)$$



# Erläuterung

---

- Typische Formel für **Log-Odds Ratio**
- Benutzung des Logarithmus zur Ersetzung von Multiplikation mit Addition
- Bruch
  - Normierung der Übergangshäufigkeit mit Wahrscheinlichkeit des zufälligen „Umkippens“

$$M_x(i, j) = \log\left(\frac{f(i, j)}{f(i) * f(j)}\right)$$

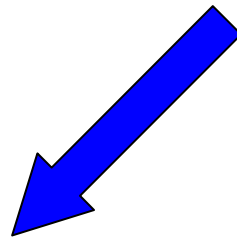
- $M(i, j) = 0$  (Bruch = 1)
  - **Keine Selektion** - Anzahl Übergänge entspricht statistischer Erwartung
- $M(i, j) < 0$  (Bruch < 1)
  - **Negative Selektion** – Übergang wird unterdrückt
- $M(i, j) > 0$  (Bruch > 1)
  - **Positive Selektion** – Übergang wird bevorzugt

# Beispiel

$S_{1,1}$ : ACGGTGAC  
 $S_{2,1}$ : AGG\_TGCC  
 $S_{1,3}$ : GTT\_AGCTA  
 $S_{2,4}$ : TTTCAG\_TA  
 $S_{1,2}$ : GGTCAA  
 $S_{2,2}$ : AGTC\_A

Absolute Häufigkeiten

A: 11/42	C: 8/42	G: 12/42	T: 11/42
----------	---------	----------	----------



Übergangshäufigkeiten



	A	C	G	T
A	4/19	1/19	1/19	0/19
C		2/19	1/19	0/19
G			4/19	1/19
T				5/19

Substitutionsmatrix

	A	C	G	T
A	0,48	0,02	-0,15	-
C		0,46	-0,01	-
G			0,41	-0,15
T				0,58

# Probleme mit PAM Matrizen

- Evolutionäre Wahrheit ist nicht bekannt
  - Besonders bei großem Abstand schwierig und subjektiv

einfach

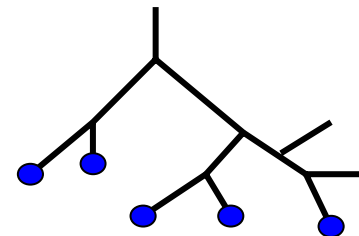
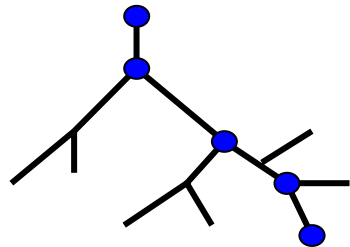
FMM\_IYVVYL  
FMMUIYV\_YL

schwierig

\_\_FMMFYVVYL  
UFPHVYL\_YL  
\_\_UFPHVYLYL  
FMMFYVVYL\_\_  
\_\_UFPHVYLYL

- Man hat keine Sequenzen, die x PAM entfernt sind
  - Wahren evolutionären Abstand schätzt man aus %-Identität

X PAM {  
X PAM {  
X PAM {  
X PAM {



# Reale PAM Matrizen

---

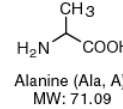
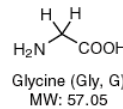
- Annahme der „ **Molecular Clock Theory**“
  - Evolution verläuft gleichmäßig in der Zeit und in den Sequenzpositionen
  - Hochrechnung langer Distanzen aus kurzen
- Vorgehen von Dayhoff et al.
  - Paare eng verwandter Sequenzen auswählen
    - >85% Identität, 34 Proteinfamilien
  - Manuell alignieren
  - PAM-1 Matrix  $M_1$  aus Häufigkeiten berechnen
  - PAM-x Matrizen wie folgt berechnen:  $M_n = (M_1)^n$

# PAM 250

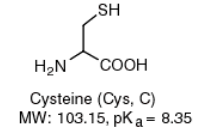
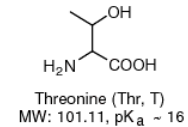
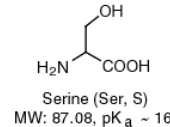
## 250 Multiplikationen der PAM-1 Matrix mit sich selber

Cys	12																		
Gly	-3	5																	
Pro	-3	-1	6																
Ser	0	1	1	1															
Ala	-2	1	1	1	2														
Thr	-2	0	0	1	1	3													
Asp	-5	1	-1	0	0	0	4												
Glu	-5	0	-1	0	0	0	3	4											
Asn	-4	0	-1	1	0	0	2	1											
Gln	-5	-1	0	-1	0	-1	2	2											
His	-3	-2	0	-1	-1	-1	1	1											
Lys	-5	-2	-1	0	-1	0	0	0											
Arg	-4	-3	0	0	-2	-1	-1	-1											
Val	-2	-1	-1	-1	0	0	-2	-2											
Met	-5	-3	-2	-2	-1	-1	-3	-2											
Ile	-2	-3	-2	-1	-1	0	-2	-2											
Leu	-6	-4	-3	-3	-2	-2	-4	-3											
Phe	-4	-5	-5	-3	-4	-3	-6	-5											
Tyr	0	-5	-5	-3	-3	-3	-4	-4											
Trp	-8	-7	-6	-2	-6	-5	-7	-7											
Cys	Gly	Pro	Ser	Ala	Thr	Asp	Glu	Asn	Gln	His	Lys	Arg	Val	Met	Ile	Leu	Phe	Tyr	Trp

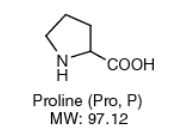
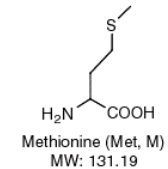
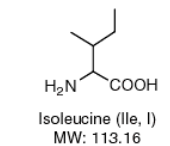
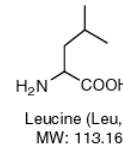
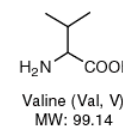
### Small



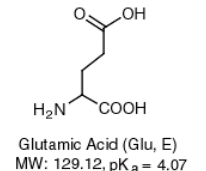
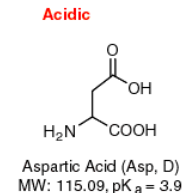
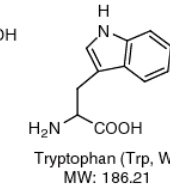
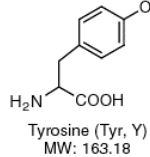
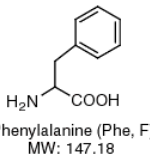
### Nucleophilic



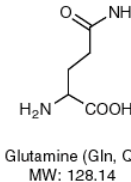
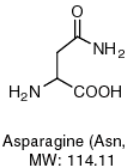
### Hydrophobic



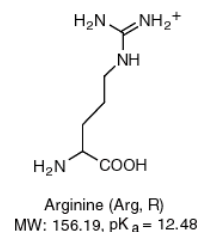
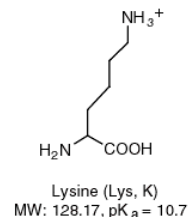
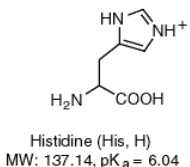
### Aromatic



### Amide



### Basic



# Verwendung

---

- Welche PAM Matrix soll man nun zur Alignierung zweier Sequenzen verwenden?
  - Die, die dem PAM-Abstand der Sequenzen entspricht
  - Den kennt man aber nicht – schätzen
  - Schätzung benötigt Alignments
    - Zur Berechnung der Sequenzidentität
  - Alignments basieren auf Substitutionsmatrizen
  - **Henne – Ei Problem**
- Also
  - Iteratives Verfahren einsetzen du auf Konvergenz warten
  - Verschiedene Matrizen testen
  - Externes Wissen (Chemie, Strukturen, etc.) hinzuziehen

# BLOSUM Matrizen

---

- Hauptkritikpunkte am PAM Ansatz
  - Nur Verwendung sehr ähnlicher Sequenzen
    - Realistische Zahlen für evolutionär weiter entfernte Sequenzen?
  - Einbeziehung kompletter Proteinsequenzen
    - Unterliegen die alle der selben Mutationsrate?
  - PAM-x vervielfältigen Fehler in PAM-1
- Anderer (neuerer) Ansatz: BLOSUM
  - BLOcks SUbstitution Matrix
  - Multiplen Alignments evolutionär entfernter, aber homologer Proteinsequenzen
  - Benutzung nur der konservierten Blöcke
  - Populärer als PAM Matrizen

# BLOSUM Vorarbeiten

---

- PROSITE

- Beschreibung funktionaler (=konservierter) Bereiche in [homologen Proteinsequenzen](#) durch reguläre Ausdrücke
- Expertenwissen - manuelle Pflege der Datenbank am EBI

- BLOCKS

- Alignierung von durch PROSITE Ausdrücke gematchten Sequenzen in [Multiple Alignments](#) (MSA)
  - Multiple Sequence Alignment –später mehr
  - Heute: Verwendung weiterer Domänen aus PRINTS, PFAM, ...
- Ein [BLOCK](#) ist zusammenhängendes Stück in einem MSA

```
FMYMFYVVPL_PQ_QVY
FYQQF__VQLYP_MFQV_
FMY__YUVOQP_UMUQ__
```

# BLOSUM Matrizen

---

- Berechnung der BLOSUM Matrizen verläuft identisch zur Berechnung der PAM-1 Matrix
  - Alle BLOCKS werden betrachtet
  - Absolute Häufigkeiten aller Aminosäuren
  - Häufigkeiten aller Übergänge in allen Paaren
- BLOSUM-x Matrizen
  - Bias in BLOCKS durch zu viele sehr ähnliche Sequenzen
  - Zur Berechnung der BLOSUM-x Matrix werden in jedem Block alle Sequenzen mit >x% Identität zu einer Sequenz zusammengefasst
  - **Gänzlich andere Bedeutung** des „x“ als in PAM-x
  - Aber ähnliche Verwendung: x ~ evolutionärem Abstand

$$M_1(i, j) = \log\left(\frac{f(i, j)}{f(i) * f(j)}\right)$$

# Unterschiede PAM - BLOSUM

---

- BLOSUM verwendet nur hochkonservierte Bereiche, PAM komplette Alignments
- PAM rechnet große evolutionäre Abstände nur hoch, BLOSUM verwendet gezielt entfernte Sequenzen
- BLOSUM basiert auf deutlich mehr Sequenzen
- Heutige BLOSUM-Matrizen sind heuristisch verbessert
  - „Feedback-Schleife“: Mit initialer BLOSUM 62 Matrix erneute Alignierung
  - Bestimmung der BLOCKS verwendet BLOSUM Matrix
- BLOSUM-62 heute meist der Default in Alignmentprogrammen

# BLOSUM 45 Matrix

- Ausblendung Sequenzen >45% Sequenzidentität

Gly	7																			
Pro	-2	9																		
Asp	-1	-1	7																	
Glu	-2	0	2	6																
Asn	0	-2	2	0	6															
His	-2	-2	0	0	1	10														
Gln	-2	-1	0	2	0	1	6													
Lys	-2	-1	0	1	0	-1	1	5												
Arg	-2	-2	-1	0	0	0	1	3	7											
Ser	0	-1	0	0	1	-1	0	-1	-1	4										
Thr	-2	-1	-1	-1	0	-2	-1	-1	-1	2	5									
Ala	0	-1	-2	-1	-1	-2	-1	-1	-2	1	0	5								
Met	-2	-2	-3	-2	-2	0	0	-1	-1	-2	-1	-1	6							
Val	-3	-3	-3	-3	-3	-3	-3	-2	-2	-1	0	0	1	5						
Ile	-4	-2	-4	-3	-2	-3	-2	-3	-3	-2	-1	-1	2	3	5					
Leu	-3	-3	-3	-2	-3	-2	-2	-3	-2	-3	-1	-1	2	1	2	5				
Phe	-3	-3	-4	-3	-2	-2	-4	-3	-2	-2	-1	-2	0	0	0	1	8			
Tyr	-3	-3	-2	-2	-2	2	-1	-1	-1	-2	-1	-2	0	-1	0	0	3	8		
Trp	-2	-3	-4	-3	-4	-3	-2	-2	-2	-4	-3	-2	-2	-3	-2	-2	1	3	15	
Cys	-3	-4	-3	-3	-2	-3	-3	-3	-3	-1	-1	-1	-2	-1	-3	-2	-2	-3	-5	12
Gly	Pro	Asp	Glu	Asn	His	Gln	Lys	Arg	Ser	Thr	Ala	Met	Val	Ile	Leu	Phe	Tyr	Trp	Cys	

# Zusammenfassung

---

- Willkommen im Reich der Heuristiken
- Substitutionsmatrizen
  - Beobachten tatsächlich evolutionär entstandene Unterschiede
  - Diese können nur geschätzt werden
  - Implizite Beachtung vielfältiger Faktoren
    - Ladungen, Nachbarschaft, Brücken über entfernte Aminosäuren durch 3D-Struktur, ...
  - Berechnung von Matrizen immer mit Hilfe „bekanntere“ echter Homologien – Henne-Ei Problem