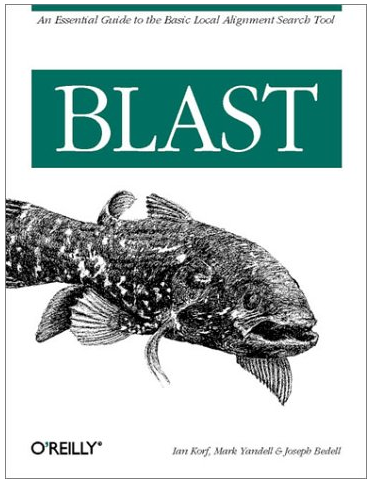
V2 Paarweises Sequenzalignment

- Methoden des Sequenzalignments
- Austauschmatrizen
- Bedeutsamkeit von Alignments
- BLAST, Algorithmus – Parameter – Ausgabe http://www.ncbi.nih.gov



Diese Vorlesung lehnt sich eng an das BLAST Tutorial-Buch (links) an, Kapitel 3-9

Sequenz-Alignment

Wenn man 2 oder mehr Sequenzen vorliegen hat, möchte man zunächst einmal

- ihre Ähnlichkeiten quantitativ erfassen
  Die ähnlichen Regionen können hierbei die ganze Sequenz, oder Teile von ihr umfassen! Lokales Alignment ↔ globales Alignment

- Entsprechungen zwischen einzelnen Bausteinen beider Sequenzen erfassen

- Gesetzmässigkeiten der Konservierung und Variabilität beobachten

- Rückschlüsse auf entwicklungsgeschichtliche **Verwandschaftsverhältnisse** ziehen

Wichtiges Ziel: **Annotation**, d.h. Zuordnung von strukturellen und funktionellen Merkmalen zu Gensequenzen.

Ähnlichkeit von Aminosäuren

Margaret Dayhoff stellte die Ähnlichkeit (beobachtete Austauschhäufigkeiten zwischen verwandten Sequenzen) zwischen Aminosäuren als $\log_2$ odds Verhältnis, oder *lod score* dar.

Margaret Dayhoff
http://www.nlm.nih.gov/changingthefaceofmedicine/gallery/photo_76_7.html

Lod score einer Aminosäure: nehme den Logarithmus zur Basis 2 ($\log_2$) von dem Verhältnis der beobachteten Häufigkeit für ein Paar durch die zufällig für das Paar erwartete Häufigkeit.

Lod score = 0 → beobachtete und erwartete Häufigkeiten sind gleich
> 0 → ein Austauschpaar tritt häufiger auf als zufällig erwartet
< 0 → unwahrscheinlicher Austausch

Allgemeine Formel für die Bewertung s_{ij} zweier Aminosäuren *i* und *j*.

$$s_{ij} = \log \frac{q_{ij}}{p_i p_j}$$

mit den individuellen Häufigkeiten p_i und p_j, und der Paarungsfrequenz q_{jj},

Ähnlichkeit der Aminosäuren

Beispiel: die relative Häufigkeiten von Methionin und Leucin seien 0.01 und 0.1.

Durch zufällige Paarung erwartet man 1/1000 Austauschpaare Met – Leu.

Wenn die beobachtete Paarungshäufigkeit 1/500 ist, ist das Verhältnis der Häufigkeiten 2/1.
Im Logarithmus zur Basis 2 ergibt sich ein *lod score* von +1 or 1 bit.

Wenn die Häufigkeit von Arginin 0.1 und die Paarung mit Leu die Häufigkeit 1/500 hat, dann ergibt sich ein *lod score* für ein Arg – Leu Paar von -2.322 bits.

Gewöhnlich berechnet man *nats*, multipliziert die Werte mit einem Skalierungsfaktur und rundet sie dann auf Integer Werte
→ **Austauschmatrizen** PAM und BLOSUM.

Diese ganzzahligen Werte (Integers) nennt man *raw scores.*

Bewertungs- oder Austausch-Matrizen

– dienen um die Qualität eines Alignments zu bewerten

– Für Protein/Protein Vergleiche:
eine *20 x 20* Matrix für die Wahrscheinlichkeit, mit der eine bestimmte Aminosäure gegen eine andere durch zufällige Mutationen ausgetauscht werden kann.

– Matrizen werden als symmetrisch angenommen, besitzen also die Form einer Dreiecksmatrix.

Substitutions-Matrizen

Nicht alle Aminosäuren sind gleich

- Einige werden leichter ausgetauscht als andere. Der Austausch von Aminosäuren ähnlichen Charakters (Ile, Leu) ist wahrscheinlicher (hat eine höhere Bewertung bzw. tritt häufiger in der Natur auf) als der von Aminosäuren unterschiedlichen Charakters (e.g. Ile, Asp).
- Bestimmte Mutationen geschehen leichter als andere
- Einige Austausche bleiben länger erhalten als andere

Mutationen bevorzugen bestimmte Austausche

- Einige Aminosäuren besitzen ähnliche Codons (siehe Codon-Sonne)
- Diese werden eher durch Mutation der DNA mutiert

Selektion bevorzugt bestimmte Austausche

- Einige Aminosäuren besitzen ähnliche Eigenschaften und Struktur

PAM250 Matrix

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	12																			
S	0	2																		
T	−2	1	3																	
P	−3	1	0	6																
A	−2	1	1	1	2															
G	−3	1	0	−1	1	5														
N	−4	1	0	−1	0	0	2													
D	−5	0	0	−1	0	1	2	4												
E	−5	0	0	−1	0	0	1	3	4											
Q	−5	−1	−1	0	0	−1	1	2	2	4										
H	−3	−1	−1	0	−1	−2	2	1	1	3	6									
R	−4	0	−1	0	−2	−3	0	−1	−1	1	2	6								
K	−5	0	0	−1	−1	−2	1	0	0	1	0	3	5							
M	−5	−2	−1	−2	−1	−3	−2	−3	−2	−1	−2	0	0	6						
I	−2	−1	0	−2	−1	−3	−2	−2	−2	−2	−2	−2	−2	2	5					
L	−6	−3	−2	−3	−2	−4	−3	−4	−3	−2	−2	−3	−3	4	2	6				
V	−2	−1	0	−1	0	−1	−2	−2	−2	−2	−2	−2	−2	2	4	2	4			
F	−4	−3	−3	−5	−4	−5	−4	−6	−5	−5	−2	−4	−5	0	1	2	−1	9		
Y	0	−3	−3	−5	−3	−5	−2	−4	−4	−4	0	−4	−4	−2	−1	−1	−2	7	10	
W	−8	−2	−5	−6	−6	−7	−4	−7	−7	−5	−3	2	−3	−4	−5	−2	−6	0	0	17

Beispiel für eine Bewertung

Wenn sich 2 Sequenzen in 2 (oder mehreren) Positionen unterscheiden, möchte man die Wahrscheinlichkeit berechnen, daß Änderung A an Position 1 auftritt UND Änderung B an Position 2 (usw).

Man braucht also log (A × B), wobei das Malzeichen für die UND-Verknüpfung steht.

Es gilt allgemein log (A × B) = log A + log B

→ die Bewertung (Score) eines Alignments ist daher einfach die **Summe** aller Bewertungen für die Paare an Aminosäuren (Nukleinsäuren) des Alignments:

Sequenz 1: `TCCPSIVARSN`

Sequenz 2: `SCCPSISARNT`

`1 12 12 6 2 5 -1 2 6 1 0` → Alignment Bewertung = 46

Dayhoff Matrix (1)

– wurde von Margaret.O. Dayhoff aufgestellt, die statistische Daten über die Austauschhäufigkeit von Aminosäuren in paarweisen Sequenzalignments sammelte

– Datensatz enthält eng verwandte Paare von Proteinsequenzen (> 85% Identität). Diese können nämlich zweifelsfrei aligniert werden.

– Aus der Frequenz, mit der Austausche auftreten, stellte sie die 20 x 20 Matrix für die Wahrscheinlichkeiten auf, mit der Mutationen eintreten.

– Diese Matrize heisst PAM 1. Ein evolutionärer Abstand von 1 PAM (point accepted mutation) bedeutet, dass es 1 Punktmutation pro 100 Residuen gibt, bzw. dass die beiden Sequenzen zu 99% identisch sind.

Dayhoff Matrix (2)

Aus PAM 1 kann man Matrizen für größere evolutionäre Entfernungen herstellen, indem man die Matrix **mehrfach mit sich selbst multipliziert.**

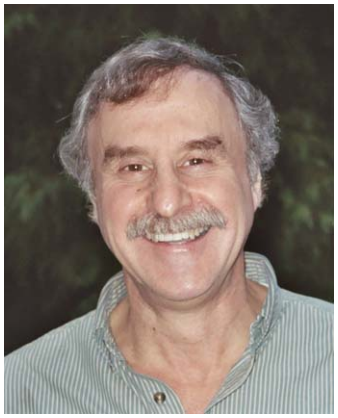
PAM250:

- 2,5 Mutationen pro Residue
- entspricht 20% Treffern zwischen zwei Sequenzen, d.h. man beobachtet Änderungen in 80% der Aminosäurepositionen.
- Dies ist die Default-Matrize in vielen Sequenzanalysepaketen.

BLOSUM Matrix

Einschränkung der Dayhoff-Matrix:
Die Matrizen, die auf dem Dayhoff-Modell der evolutionären Raten basieren, sind von eingeschränktem Wert, da ihre Substitionsraten von Sequenzalignments abgeleitet wurden, die zu über 85% identisch sind.

S. Henikoff und J.G. Henikoff: verwendeten später lokale Multiple Alignments von entfernter verwandten Sequenzen
➔ **Blosum-Matrix**



Steven Henikoff

Dies war möglich, da es nun bereits mehr Sequenzen sowie Algorithmen für multiple Alignments gab.

Vorteile dieses Ansatzes:
- größere Datenmengen (es gibt mehr Sequenzen, die entfernt miteinander verwandt sind als nah verwandte)
- multiple Alignments sind robuster als paarweise Alignments

BLOSUM Matrix (2)

Die BLOSUM Matrizen (BLOcks SUbstitution Matrix) basieren auf der BLOCKS Datenbank.

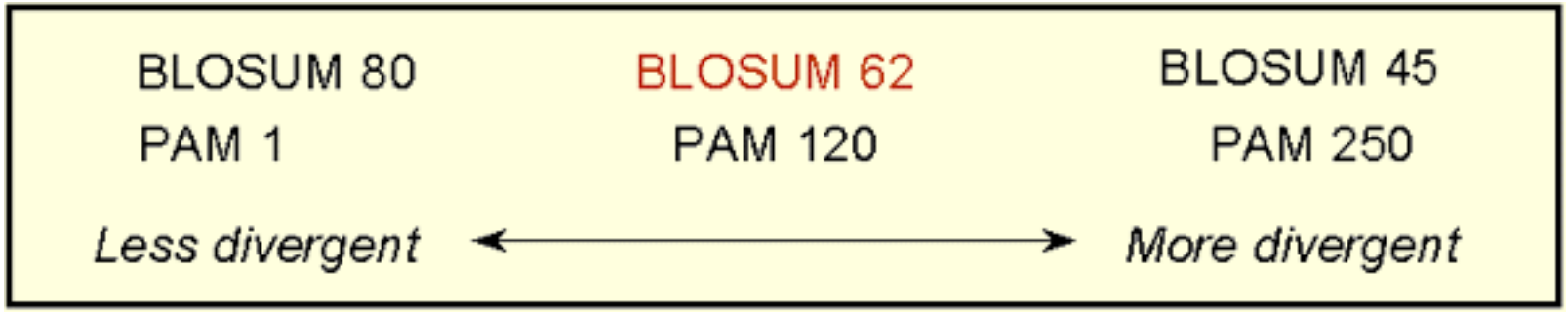
Die BLOCKS Datenbank verwendet das Konzept von Blöcken (lückenlose Aminosäure-Signaturen), die charakteristisch für eine Proteinfamilie sind.

Aus den beobachteten Mutationen innerhalb dieser Blöcke wurden Austauschwahrscheinlichkeiten für alle Aminosäurepaare berechnet und als Einträge für eine *log odds* BLOSUM matrix benutzt.

Man erhält unterschiedliche Matrizen indem man die untere Schranke des verlangten Grads an Identität variiert.
z.B. wurde die BLOSUM80 Matrix aus Blöcken mit > 80% Identität abgeleitet.

Welche Matrix soll man benutzen?

Enge Verwandtschaft (Niedrige PAM, hohe Blosum)
Entfernte Verwandtschaft (Hohe PAM, niedrige Blosum)



Vernünftige Default-Werte: PAM250, BLOSUM62

Gewichtung von Lücken (Gaps)

Neben der Substitutionsmatrix braucht man auch eine Methode zur Bewertung von Lücken.
Welche Bedeutung haben Insertionen und Deletionen im Verhältnis zu Substitutionen?

Unterscheide Einführung von Lücken:

```
aaagaaa
aaa-aaa
```

von der Erweiterung von Lücken:

```
aaaggggaaa
aaa----aaa
```

Verschiedene Programme (CLUSTAL-W, BLAST, FASTA) empfehlen unterschiedliche Default-Werte, die man wohl erst einmal verwenden sollte.

Needleman-Wunsch Algorithmus

- allgemeiner Algorithmus für Sequenzvergleiche
- maximiert eine Bewertung der Ähnlichkeit
- bester Match = größte Anzahl an Residuen einer Sequenz, die zu denen einer anderen Sequenz passen, wobei Deletionen erlaubt sind.
-Der Algorithmus findet durch dynamische Programmierung das bestmögliche GLOBALE Alignment zweier beliebiger Sequenzen

- NW beinhaltet eine iterative Matrizendarstellung

alle möglichen Residuenpaare (Basen oder Aminosäuren) – je eine von jeder Sequenz – werden in einem 2-dimensionalen Gitter dargestellt.

alle möglichen Alignments entsprechen Pfaden in diesem Gitter.

- Der Algorithmus hat 3 Schritte: 1 Initialisierung 2 Auffüllen 3 Trace-back

Needleman-Wunsch Algorithm: Initialisierung

Aufgabe: aligniere die Wörter “COELACANTH” und “PELICAN” der Länge $m = 10$ und $n = 7$. Konstruiere $(m + 1) \times (n + 1)$ Matrix.

Ordne den Elementen der ersten Zeile und Reihe die Werte $-m \times gap$ und $-n \times gap$ zu.

Die Pointer dieser Felder zeigen zurück zum Ursprung.

		C	O	E	L	A	C	A	N	T	H
	0	-1 ←	-2 ←	-3 ←	-4 ←	-5 ←	-6 ←	-7 ←	-8 ←	-9 ←	-10 ←
P	↑ -1										
E	↑ -2										
L	↑ -3										
I	↑ -4										
C	↑ -5										
A	↑ -6										
N	↑ -7										

Needleman-Wunsch Algorithm: Auffüllen

Fülle alle Matrizenfelder mit Werten und Zeigern mittels simpler Operationen, die die Werte der diagonalen, vertikalen, und horizontalen Nachbarzellen einschließen. Berechne

match score: Wert der Diagonalzelle links oben + Wert des Alignments (+1 oder -1)

horizontal gap score: Wert der linken Zelle + gap score (-1)

vertical gap score: Wert der oberen Zelle + gap score (-1).

Ordne der Zelle das Maximum dieser drei Werte zu. Der Pointer zeigt in Richtung des maximalen Werts.

		C	O	E	L	A	C	A	N	T	H
	0	-1 ←	-2 ←	-3 ←	-4 ←	-5 ←	-6 ←	-7 ←	-8 ←	-9 ←	-10 ←
P	↑ -1	↖ -1	↖ -2								

max(-1, -2, -2) = -1

max(-2, -2, -3) = -2

(Lege Konvention fest, damit Pointer bei gleichen Werten immer in eine bestimmte Richtung zeigen soll, z.B. entlang der Diagonalen.

Needleman-Wunsch Algorithmus: Trace-back

Trace-back ergibt das Alignment aus der Matrix.

Starte in Ecke rechts unten und folge den Pfeilen bis in die Ecke links oben.

```
COELACANTH
-PELICAN--
```

		C	O	E	L	A	C	A	N	T	H
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10
P	-1	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10
E	-2	-2	-2	-1	-2	-3	-4	-5	-6	-7	-8
L	-3	-3	-3	-2	0	-1	-2	-3	-4	-5	-6
I	-4	-4	-4	-3	-1	-1	-2	-3	-4	-5	-6
C	-5	-3	-4	-4	-2	-2	0	-1	-2	-3	-4
A	-6	-4	-4	-5	-3	-1	-1	1	0	-1	-2
N	-7	-5	-5	-5	-4	-2	-2	0	2	1	0

Smith-Waterman-Algorithmus

Smith-Waterman ist ein lokaler Alignment-Algorithmus. SW ist eine sehr einfache Modifikation von Needleman-Wunsch. Es gibt lediglich 3 Änderungen:

- die Matrixränder werden auf 0 statt auf ansteigende Gap-Penalties gesetzt.
- der maximale Wert sinkt nie unter 0. Pointer werden nur für Werte größer als 0 eingezeichnet.
- Trace-back beginnt am größten Wert der Matrix und endet bei dem Wert 0.

ELACAN

ELICAN

		C	O	E	L	A	C	A	N	T	H
	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	1	0	0	0	0	0	0	0
L	0	0	0	0	2	1	0	0	0	0	0
I	0	0	0	0	1	1	0	0	0	0	0
C	0	1	0	0	0	0	2	0	0	0	0
A	0	0	0	0	0	1	0	3	2	1	0
N	0	0	0	0	0	0	0	1	4	3	2

BLAST –
Basic Local Alignment Search Tool

- Findet das am besten bewertete lokale optimale Alignment einer Testsequenz mit allen Sequenzen einer Datenbank.

- Sehr schneller Algorithmus, 50 mal schneller als dynamische Programmierung.

- Kann verwendet werden um sehr große Datenbanken zu durchsuchen, da BLAST eine vor-indizierte Datenbank benutzt

- Ist ausreichend sensititv und selektiv für die meisten Zwecke

- Ist robust – man kann üblicherweise die Default-Parameter verwenden

BLAST Algorithmus, Schritt 1

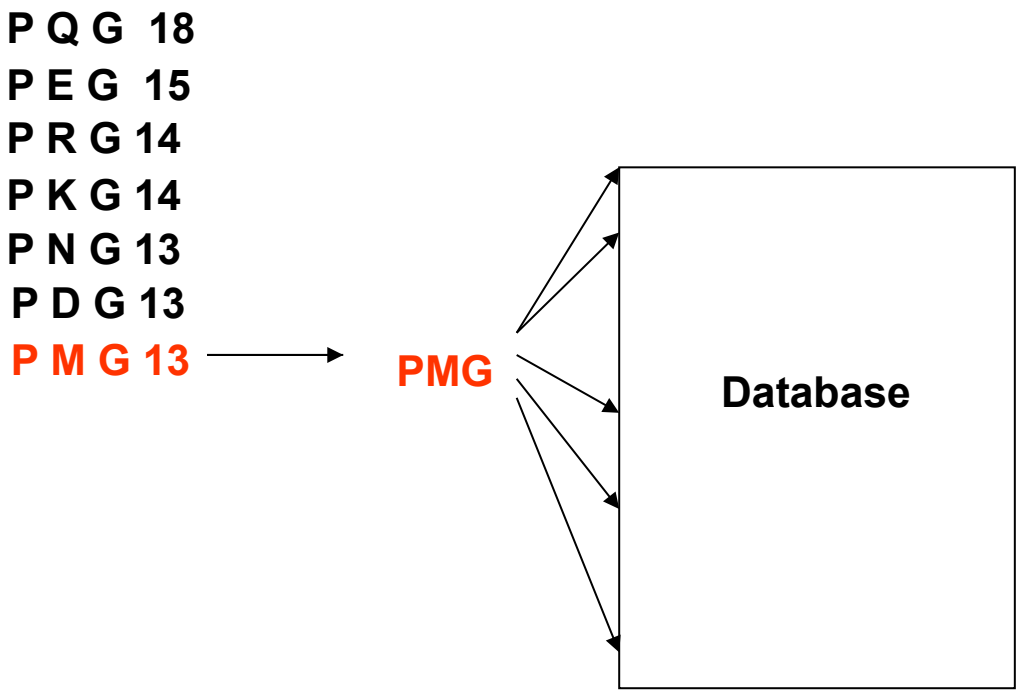
- Für ein gegebenes Wort der Länge *w* (gewöhnlich 3 für Proteine) und eine gegebene Bewertungs-Matrix
  erzeuge eine Liste aller Worte (*w*-mers), die eine Bewertung > *T* erhalten, wenn man sie mit dem *w*-mer der Eingabe vergleicht

Test Sequenz **L N K C K T P Q G Q R L V N Q**

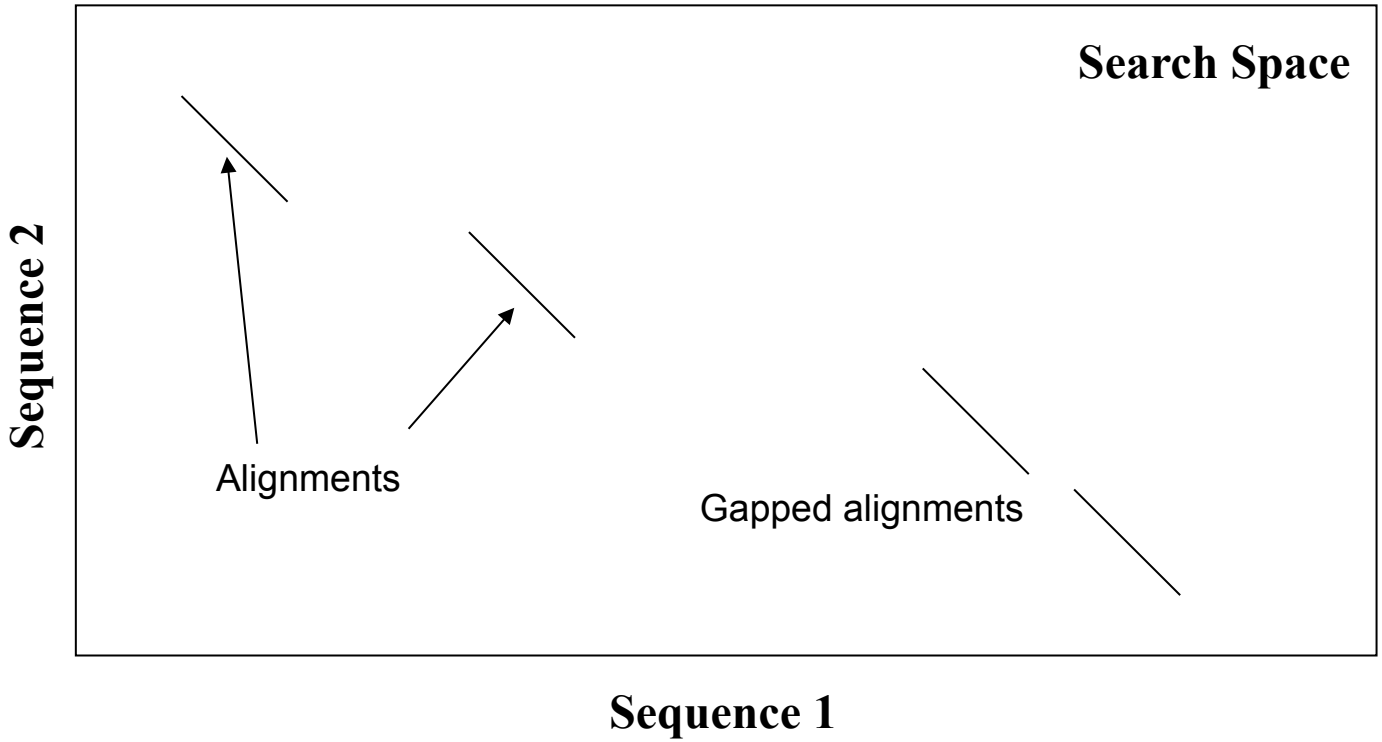
	P Q G 18	**Wort**
	P E G 15	**benachbarte Wörter**
	P R G 14	
	P K G 14	
	P N G 13	
	P D G 13	
	P M G 13	
unterhalb Schranke (T=13)	**P Q A 12**	
	P Q N 12	
	etc.	

BLAST Algorithmus, Schritt 2

jedes benachbarte Wort ergibt alle Positionen in der Datenbank, in denen es gefunden wird (hit list).

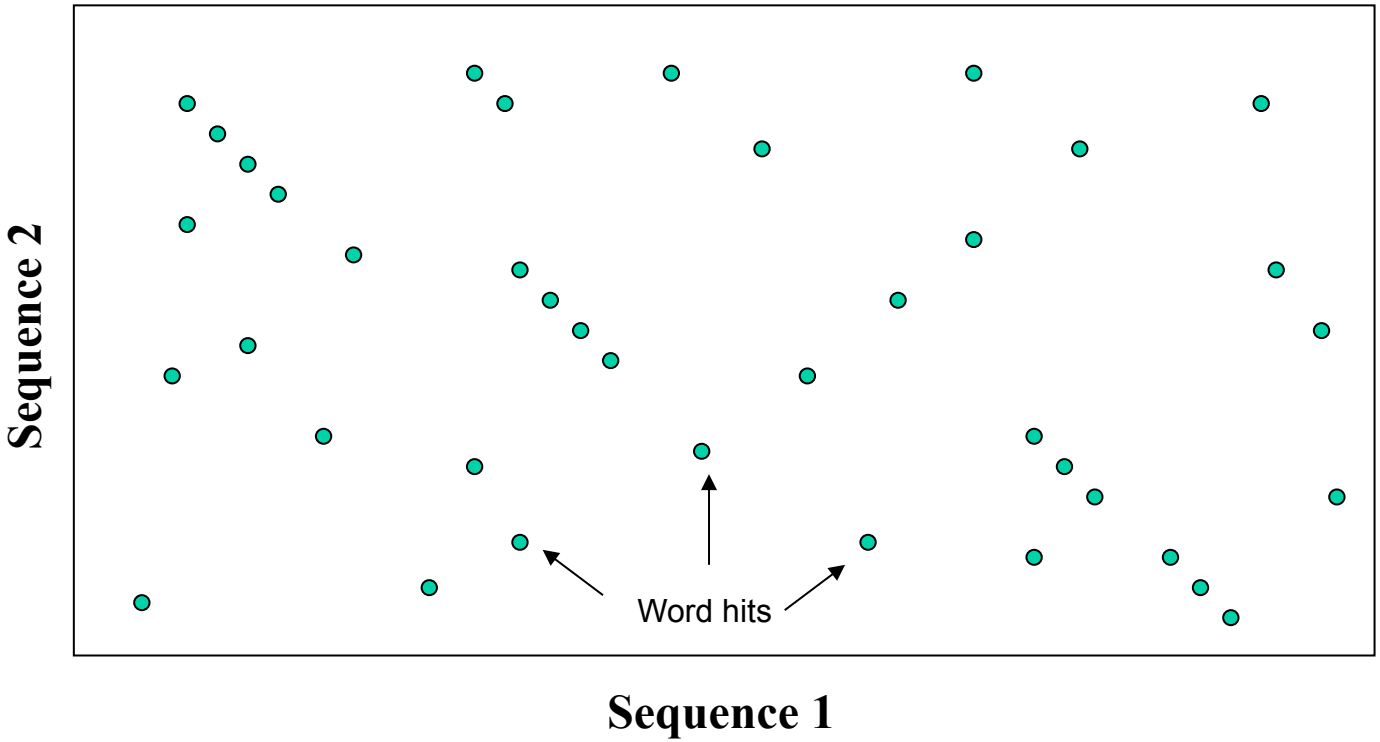


Was ist gesucht?



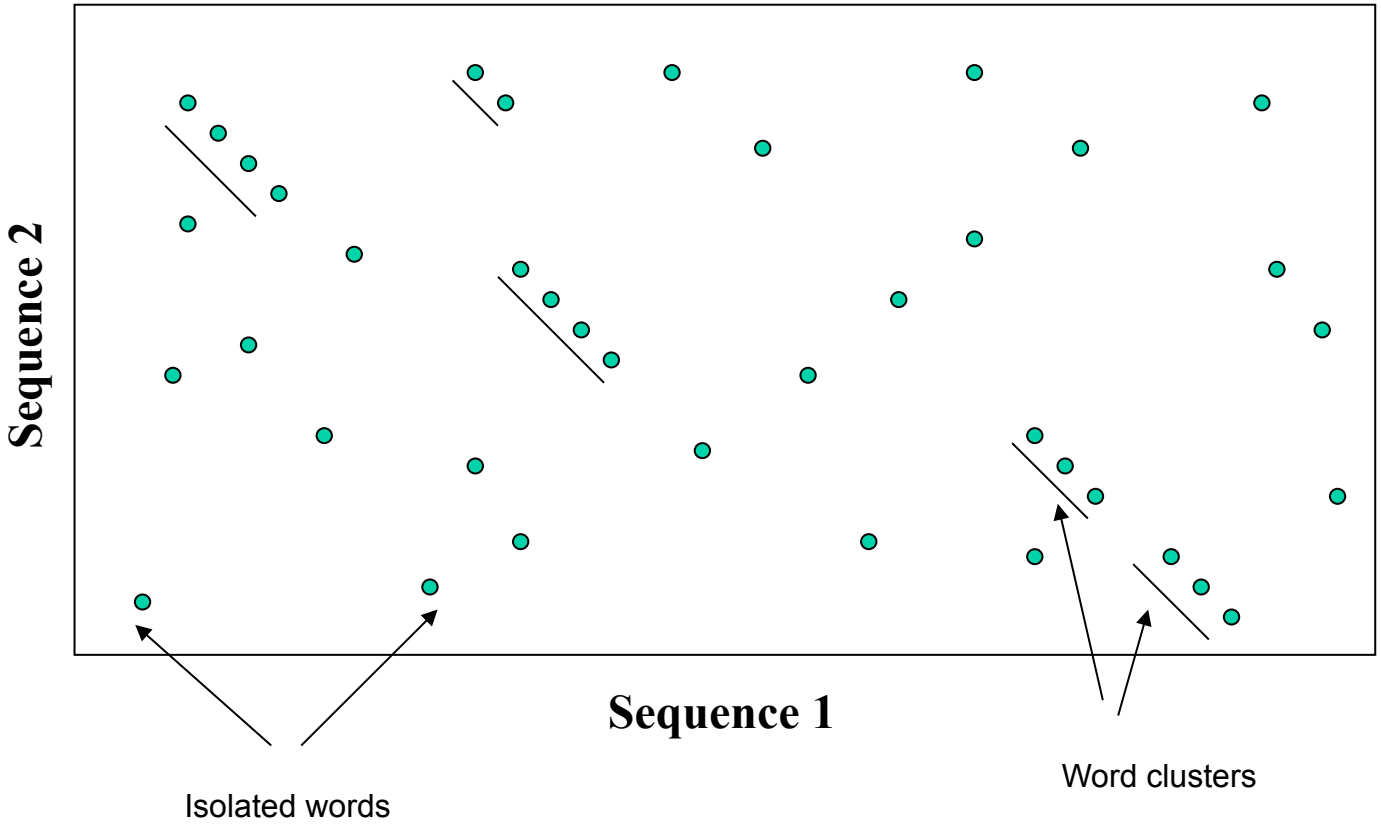
Das beste Mapping von Sequenz 1 auf Sequenz 2 entspricht einem unterbrochenen Pfad in dieser Diagonalmatrix.

Seeding



Können wir aus diesen „Word hits“ ein gutes Alignment konstruieren?

Seeding



BLAST Algorithmus: Erweiterungsschritt

- das Programm versucht, den Seed in beide Richtungen auszudehnen indem solange Residuenpaare hinzugefügt werden bis die zusätzliche Bewertung kleiner als ein Schrankenwert ist.

- Nachdem die Ausdehnung beendet wurde, wird das Alignment so “zurückbeschnitten” dass es die maximale Bewertung erhält.

```
Query:   325 SLAALLNKCKTPQGQRLVNQWIKQPLMDKNRIEERLNLVEA 365
             +LA++L+   TP G R++ +W+  P+ D   + ER   + A
Sbjct:   290 TLASVLDCTVTPMGSRMLKRWLHMPVRDTRVLLERQQTIGA 330
```

High-scoring Segment Pair (HSP)

Nachbarschaft für 3-Buchstaben-Worte

BLOSUM62		PAM200	
Wort	Bewertung	Wort	Bewertung
RGD	17	RGD	18
KGD	14	RGE	17
QGD	13	RGN	16
RGE	13	KGD	15
EGD	12	RGQ	15
HGD	12	KGE	14
NGD	12	HGD	13
RGN	12	KGN	13
AGD	11	RAD	13
MGD	11	RGA	13
RAD	11	RGG	13
RGQ	11	RGH	13
RGS	11	RGK	13
RND	11	RGS	13
RSD	11	RGT	13
SGD	11	RSD	13
TGD	11	WGD	13

Kommentar:
Sowohl die Auswahl der Austauschmatrix wie die Wahl des Cut-offs T wird den Seeding-Schritt beeinflussen.

BLAST Eingabe

Notwendige Schritte um BLAST einzusetzen (im Zeitalter des Internets!):

Wähle einen **Webserver** (EBI = European Bioinformatics Institute, NCBI = National Center for Biotechnology Information …)

- gib Testsequenz ein (cut-and-paste)
- wähle die Nukleotid bzw. Aminosäure-Sequenzdatenbank, die durchsucht werden soll
- wähle Parameter um Output zu steuern (Zahl der Sequenzen …)
- wähle Parameter für das Alignment (z.B. Austauschmatrix, Filter,….)

Testsequenz =
MAFIWLLSCYALLGTTFGCGVNAIHPVLTGLSKIVNGEEAVPGTWPWQVTLQDRSGFHF
CGGSLISEDWVVTAAHCGVRTSEILIAGEFDQGSDEDNIQVLRIAKVFKQPKYSILTVNND
ITLLKLASPARYSQTISAVCLPSVDDDAGSLCATTGWGRTKYNANKSPDKLERAALPLLT
NAECKRSWGRRLTDVMICGAASGVSSCMGDSGGPLVCQKDGAYTLVAIVSWASDTCS
ASS GGVYAKVTKIIPWVQKILSSN

BLAST Ausgabe (1)

Please wait ...

```
BLASTP 2.2.2 [Dec-14-2001]


Reference:
Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schäffer,
Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997),
"Gapped BLAST and PSI-BLAST: a new generation of protein database search
programs",  Nucleic Acids Res. 25:3389-3402.

Query= blast.seq  [Unknown form], 261 bases, 32E76D0B checksum.
         (261 letters)

Database: swissprot
           101,602 sequences; 37,315,215 total letters

Searching..................................................done
```

BLAST Ausgabe (2)

Kleine Wahrscheinlichkeit deutet an, dass der Treffer wohl nicht zufällig zustande kam.

```
                                                                  Score     E
Sequences producing significant alignments:                       (bits)  Value

swissprot:CTRB_HUMAN Chymotrypsinogen B precursor (EC 3.4.21.1).    433  e-121
swissprot:CTR2_CANFA Chymotrypsinogen 2 precursor (EC 3.4.21.1).    386  e-107
swissprot:CTRB_RAT Chymotrypsinogen B precursor (EC 3.4.21.1).      383  e-106
swissprot:CTRB_BOVIN Chymotrypsinogen B (EC 3.4.21.1).              348  4e-96
swissprot:CTRA_BOVIN Chymotrypsinogen A (EC 3.4.21.1).              330  1e-90
swissprot:CTRA_GADMO Chymotrypsin A precursor (EC 3.4.21.1).        286  2e-77
```

BLAST Ausgabe (3)

swissprot:CO2_HUMAN	Complement C2 precursor (EC 3.4.21.43) (C3/C...	55	1e-07
swissprot:CO2_MOUSE	Complement C2 precursor (EC 3.4.21.43) (C3/C...	53	3e-07
swissprot:ACH2_LONAC	Achelase II protease (EC 3.4.21.-).	52	1e-06
swissprot:GD_DROME	Serine protease gd precursor (EC 3.4.21.-) (G...	46	4e-05
swissprot:ACRO_CAPHI	Acrosin (EC 3.4.21.10) (Fragment).	39	0.009
swissprot:CTRP_PENMO	Chymotrypsin (EC 3.4.21.1) (Fragment).	36	0.047
swissprot:VSPA_CERCE	Cerastotin (EC 3.4.21.-) (Fragments).	35	0.098
swissprot:EL2B_HORSE	Neutrophil elastase 2B (EC 3.4.21.-) (Prote...	35	0.13
swissprot:CERC_SCHMA	Cercarial protease precursor (EC 3.4.21.-) ...	34	0.26
swissprot:EL2A_HORSE	Neutrophil elastase 2A (EC 3.4.21.-) (Prote...	33	0.42
swissprot:HPT_RABIT	Haptoglobin beta chain (Fragment).	31	1.4
swissprot:NMT1_ASPPA	NMT1 protein homolog.	30	4.8

Niedrige Bewertungen mit hohen Wahrscheinlickeiten deuten an, dass dies wohl keine guten Treffer sind.

Karlin-Altschul Statistik: E-value

Karlin und Altschul leiteten die Bewertung der Signifikanz eines Alignments ab (hier ohne Herleitung):

$$E = kmne^{-\lambda S}$$

Die Anzahl an Alignments (*E*), die man während einer Suche in einer Sequenz-datenbank mit *n* Sequenzen mit einer *m* Buchstaben langen Suchsequenz zufällig erhält, ist eine Funktion der Größe des Suchraums *(m × n)*, der normalisierten Austauschbewertungen *(λS),* und einer Konstanten *(k)*.

Bedeutung des Alignments in BLAST

E-Wert (Erwartungswert)

- $E = P \times$ Anzahl der Sequenzen in Datenbank
- E entspricht der Anzahl an Alignments einer bestimmten Bewertung, die man zufällig in einer Sequenz-Datenbank dieser Grösse erwartet (wird z.B. für ein Sequenzalignment E=10 angegeben, erwartet man 10 zufällige Treffer mit der gleichen Bewertung).
  Dieses Alignment ist also nicht signifikant.
- Treffer werden in BLAST nur ausgegeben, wenn der E-Wert kleiner als eine vorgewählte Schranke ist.

Grobe Anhaltspunkte

E-Wert (Erwartungswert)

$E \leq 0{,}0001$	genaue Übereinstimmung
$0{,}0001 \leq E \leq 0{,}02$	Sequenzen vermutlich homolog
$0{,}02 \leq E \leq 1$	Homologie ist nicht auszuschließen
$E \geq 1$	man muss damit rechnen, dass diese gute Übereinstimmung Zufall ist.

Traditionelle BLAST Programme

Program	Database	Query	Typical uses
BLASTN	Nucleotide	Nucleotide	Mapping oligonucleotides, cDNAs and PCR products to a genome, screening repetitive elements; cross-species sequence exploration; annotating genomic DNA sequencing reads
BLASTP	Protein	Protein	Identifying common regions between proteins; collecting related proteins for phylogenetic analyses
BLASTX	Protein	Nucleotide translated into protein	Finding protein-coding genes in genomic DNA; determining if a cDNA corresponds to a known protein
TBLASTN	Nucleotide translated	Protein	Identifying transcripts, potentially from multiple organisms, similar to a given protein; mapping a protein to genomic DNA into protein
TBLAST	Nucleotide translated into protein	Nucleotide translated into protein	Cross-species gene prediction at the genome or transcript level; searching for genes missed by traditional methods or not yet in protein databases

BLAST Ausgabe (4)

```
>swissprot:CTRB_HUMAN Chymotrypsinogen B precursor (EC 3.4.21.1).
          Length = 263

 Score =  433 bits (1222), Expect = e-121
 Identities = 220/263 (83%), Positives = 252/263 (95%), Gaps = 2/263 (0%)

Query: 1   MAFIWLLSCYALLGTTFGCGVNAIHPVLTGLSKIVNGEEAVPGTWPWQVTLQDRSGFHFC 60
           MAF+WLLSC+ALLGTTFGCGV AIHPVL+GLS+IVNGE+AVPG+WPWQV+LQD++GFHFC
Sbjct: 1   MAFLWLLSCWALLGTTFGCGVPAIHPVLSGLSRIVNGEDAVPGSWPWQVSLQDKTGFHFC 60

Query: 61  GGSLISEDWVVTAAHCGVRTSEILIAGEFDQGSDEDNIQVLRIAKVFKQPKYSILTVNND 120
           GGSLISEDWVVTAAHCGVRTS++++AGEFDQGSDE+NIQVL+IAKVFK PK+SILTVNND
Sbjct: 61  GGSLISEDWVVTAAHCGVRTSDVVVAGEFDQGSDEENIQVLKIAKVFKNPKFSILTVNND 120

Query: 121 ITLLKLASPARYSQTISAVCLPSVDDD--AGSLCATTGWGRTKYNANKSPDKLERAALPL 178
           ITLLKLA+PAR+SQT+SAVCLPS DDD  AG+LCATTGWG+TKYNANK+PDKL++AALPL
Sbjct: 121 ITLLKLATPARFSQTVSAVCLPSADDDFPAGTLCATTGWGKTKYNANKTPDKLQQAALPL 180

Query: 179 LTNAECKRSWGRRLTDVMICGAASGVSSCMGDSGGPLVCQKDGAYTLVAIVSWASDTCSA 238
           L+NAECK+SWGRR+TDVMIC  ASGVSSCMGDSGGPLVCQKDGA+TLV IVSW SDTCS
Sbjct: 181 LSNAECKKSWGRRITDVMICAGASGVSSCMGDSGGPLVCQKDGAWTLVGIVSWGSDTCST 240

Query: 239 SSGGVYAKVTKIIPWVQKILSSN 261
           SS GVYA+VTK+IPWVQKIL++N
Sbjct: 241 SSPGVYARVTKLIPWVQKILAAN 263
```

BLAST Ausgabe (5)

```
>swissprot:VSP5_TRIMU Mucrofibrase 5 precursor (EC 3.4.21.-).
          Length = 257

 Score =  103 bits (280), Expect = 3e-22
 Identities = 74/232 (31%), Positives = 110/232 (46%), Gaps = 10/232 (4%)

Query: 34   IVNGEEAVPGTWPWQVTLQDRSGFHFCGGSLISEDWVVTAAHCGVRTSEILIAGEFDQGS 93
            I+ G+E      P+ V +      + CGG+LI+E+WV+TAAHC     EI +     +
Sbjct: 25   IIGGDECNINEHPFLVLVYYDD--YQCGGTLINEEWVLTAAHCNGENMEIYLGMHSKKVP 82

Query: 94   DEDNIQVLRIAKVFKQPKYSILTVNNDITLLKLASPARYSQTISAVCLPSVDDDAGSLCA 153
            ++D  + +   K F     +    N DI L++L  P R S  I+ + LPS     GS+C
Sbjct: 83   NKDRRRRVPKEKFFCDSSKNYTKWNKDIMLIRLNRPVRKSAHIAPLSLPSSPPSVGSVCR 142

Query: 154  TTGWGRTKYNANKSPDKLERAALPLLTNAECKRSW-GRRLTDVMICGA--ASGVSSCMGD 210
              GWG          PD    A + LL    C+ ++ G   T   +C      G  SC GD
Sbjct: 143  IMGWGTISPTKVTLPDVPRCANINLLDYEVCRAAYAGLPATSRTLCAGILEGGKDSCGGD 202

Query: 211  SGGPLVCQKDGAYTLVAIVSWASDTCS-ASSGGVYAKVTKIIPWVQKILSSN 261
            SGGPL+C  +G +    IVSW  D C+     G+Y  V   + W++ I++ N
Sbjct: 203  SGGPLIC--NGQFQ--GIVSWGGDPCAQPHEPGLYTNVFDHLDWIKGIIAGN 250
```

BLAST Ausgabe (6)

```
>swissprot:HPT_RABIT Haptoglobin beta chain (Fragment).
          Length = 40

 Score = 31.3 bits (74), Expect = 1.4
 Identities = 15/41 (36%), Positives = 22/41 (53%), Gaps = 1/41 (2%)

Query: 34 IVNGEEAVPGTWPWQVTLQDRSGFHFCGGSLISEDWVVTAA 74
          I+ G     G++PWQ  +  R      G +LISE W++T A
Sbjct: 1  IIGGSLDAKGSFPWQAKMVSRHNL-VTGATLISEQWLLTTA 40
```

```
>swissprot:NMT1_ASPPA NMT1 protein homolog.
          Length = 342

 Score = 29.6 bits (69), Expect = 4.8
 Identities = 11/34 (32%), Positives = 22/34 (64%)

Query: 72  TAAHCGVRTSEILIAGEFDQGSDEDNIQVLRIAK 105
           TA  CG+  ++ +I G+ D G   +N+Q++ +A+
Sbjct: 137 TAVRCGMNVTKAIIRGDIDAGIGLENVQMVELAE 170
```

Obwohl ein hoher Anteil an identischen und positiven Positionen vorliegt, haben beide Treffer aufgrund ihrer kurzen Länge sehr hohe E-Werte.

Solche „Treffer“ für kurze Sequenzabschnitte können oft zufällig sein.

Tips für den Einsatz von BLAST

Verwende nicht stur die Standardparameter “You get what you look for”.

Führe Kontrollen durch, besonders in der twilight zone.

z.B. Schüttle die Sequenz durcheinander und wiederhole die Suche.

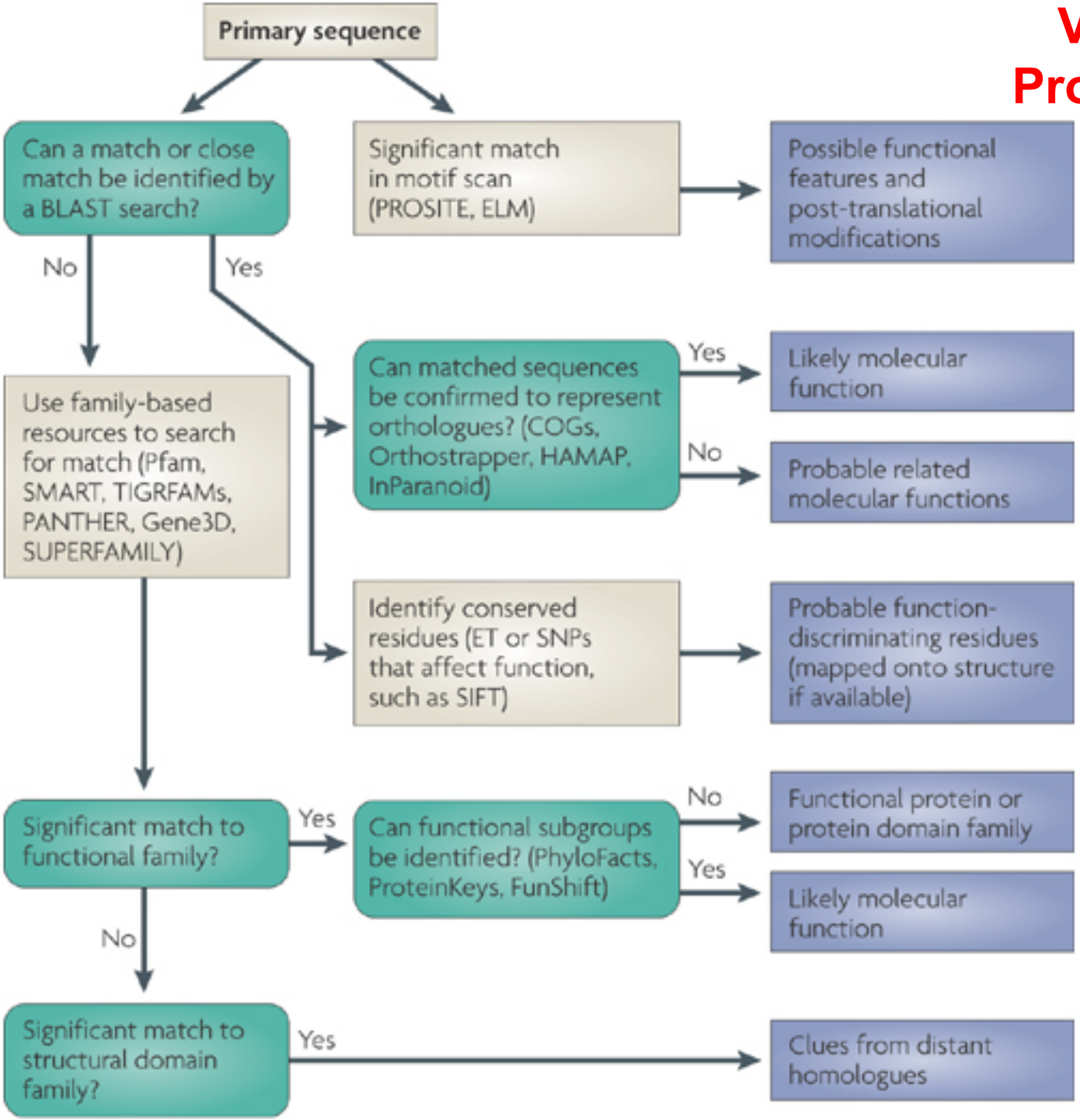
Falls die variierte Sequenz ähnliche Ergebnisse liefert, beruht das Alignment auf einer systematischen Verfälschung, oder die Parameter sind nicht empfindlich genug gewählt

Setze **Komplexitätsfilter** ein, wenn erforderlich.

Maskiere Repeats in genomischer DNA.

Teile große Genomsequenzen in Stücke auf um die Suche zu beschleunigen.

Vorhersage von Proteinfunktion u.a. mit BLAST



"There is no perfect workflow that can guarantee prediction of the correct function of a protein from its sequence."

Lee, Redfern, Orengo
Nature Rev Mol Cell Biol
8, 995-1005 (2007)

Wieviel Sequenzidentität ist erforderlich?

Beispiel: EC-Klassifizierung

•*EC 3* Enzyme sind **Hydrolasen** (Enzyme, die Wasser nutzen um ein anderes Moleküle zu spalten)

•*EC 3.4* sind Hydrolasen, die auf **Peptidbindungen** einwirken

•*EC 3.4.11* sind Hydrolasen, die die N-terminale Aminosäure von einem **Polypeptide** abspalten

•*EC 3.4.11.4* sind diejenigen, die die N-terminale Aminosäure von einem **Tripeptide** abspalten

40% paarweiser Sequenzidentität zwischen 2 Proteinsequenzen ist eine zuverlässige untere Schranke um die ersten 3 Ziffern der EC-Nummer des einen Proteins dem anderen zuordnen.

Um alle 4 EC-Ziffern mit mindestens 90% Trefferquote zuzuordnen, benötigt man >60% Sequenzidentität.

Lee, Redfern, Orengo
Nature Rev Mol Cell Biol
8, 995-1005 (2007)

PSI-BLAST: Position-Specific Iterated BLAST

- Entfernte Verwandtschaften lassen sich besser durch Motiv- oder Profil-Suchen entdecken als durch paarweise Vergleiche
- PSI-BLAST führt zunächst eine BLAST-Suche mit Gaps durch und identifiziert signifikante Treffer (z.B. 500 beste Treffer mit E-value < 0.001)
- Berechne aus den beobachteten Häufigkeiten der 20 Aminosäuren in den einzelnen Positionen des Alignments die Wahrscheinlichkeit, mit der die Aminosäuren an den Positionen auftauchen (können). Dies nennt man ein **Sequenzprofil** (siehe Vorlesung 6).
- Das PSI-BLAST Programm konvertiert das Sequenzprofil in eine Menge von **positionsspezfischen Austauschmatrizen**, die in der nächsten Runde der Datenbank-Suche verwendet werden.
- PSI-BLAST kann iterativ verwendet werden bis keine neuen signifikanten Alignments mehr gefunden werden.
- Fazit: benutze PSI-BLAST um entfernt verwandte Sequenzen zu finden.

Zusammenfassung

Paarweises Sequenzalignment ist heute Routine, aber nicht trivial.

Mit **dynamischer Programmierung** (z.B. Smith-Waterman) findet man garantiert das Alignment mit optimaler Bewertung.
Vorsicht: die Bewertungsfunktion ist nur ein Modell der biologischen Evolution.

Die schnellste Alignmentmethode ist BLAST und seine Derivate wie BLAT.
Es ergibt sehr robuste und brauchbare Ergebnisse für Proteinsequenzen.

Multiple Sequenzalignments sind in der Lage, entferntere Ähnlichkeiten aufzuspüren und bieten ein besseres funktionelles Verständnis von Sequenzen und ihren Beziehungen

Kommt nächste Woche dran.