

# Bioinformatik

BLAST

Basic Local Alignment Search Tool

Ulf Leser

Wissensmanagement in der  
Bioinformatik



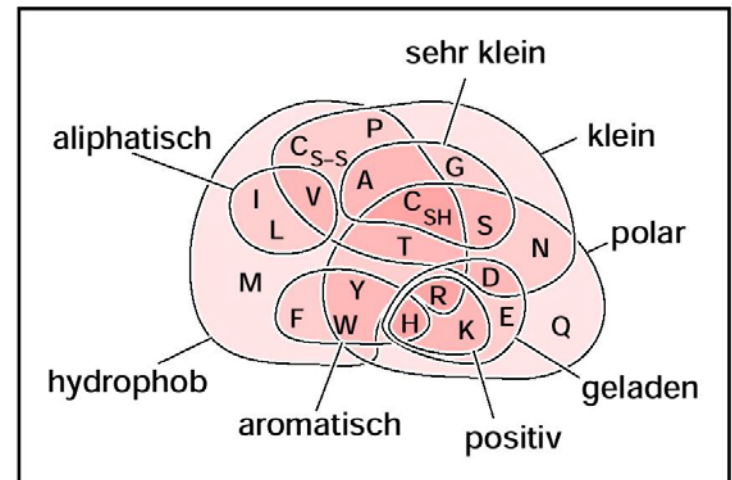
# Praktikum

---

- Anmeldung bis 10.02.2004 bei Silke Trißl
  - ([trissl@informatik.hu-berlin.de](mailto:trissl@informatik.hu-berlin.de))
  - Name, Vorname
  - Matrikelnummer
  - Semester, Studiengang
  - Account-Name in der Informatik bzw. ‚kein Account vorhanden‘
    - wenn kein Account vorhanden / Passwort abgelaufen um Zugang zu Rechnern der Informatik bemühen

# Hintergrund

- Schon öfters angesprochen ...
  - Ähnlichkeitsmatrizen, Substitutionsmatrizen, Scorefunktionen, ...
- Ersetzung einer Base/Aminosäure durch eine andere hat **unterschiedliche Bedeutung**
  - Basen: Auswirkungen auf kodiertes Protein nicht gleichverteilt über die drei Codon-Positionen
  - Aminosäuren
    - Ersetzung mit „sehr ähnlichen“ Aminosäuren ändert Proteinstruktur kaum
    - Ersetzung mit „wenig ähnlichen“ Aminosäuren kann Struktur vollkommen ändern



# 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

# 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

# 1. PAM als Sequenzabstand

---

- Definition

*Seien  $S_1$  und  $S_2$  zwei Proteinsequenzen.  $S_1$  und  $S_2$  heißen  $x$  PAM entfernt, wenn gilt*

- *$S_1$  wurde in  $S_2$  überführt mit  $x$  Punktmutationen pro 100 Aminosäuren*

- Eigenschaften

- PAM beachtet keine Inserts und Deletions
  - Großes Manko des PAM Ansatzes
- Der echten PAM Abstand zweier Sequenzen ist nicht bestimmbar
  - Dazu hätte man ein paar Millionen Jahre zusehen müssen
- Stattdessen: **Schätzen des PAM Abstandes**
  - Grundlage: durchschnittliche Veränderung nach X Mutationen
  - Ermitteltbar z.B. durch Simulation
- Vorsicht: 50 PAM heißt nicht 50 veränderte pro 100 Aminosäuren
  - Doppelmutationen, Rückmutationen, etc.

- Eigentlich besser

- $S_1, S_2$  sind  $x$  PAM entfernt, wenn:  $S_1$  wurde *am wahrscheinlichsten* in  $S_2$  überführt mit  $x$  Punktmutationen pro 100 Aminosäuren

## 2. PAM Matrizen - Grundidee

---

- Vorgehen

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  $A_i$  über alle Sequenzen, normiert auf Gesamtlänge aller Sequenzen
- Aligniere alle Paare entsprechend der **evolutionären Wahrheit**
  - $S_{k,l}'$  sei die Sequenz  $S_{k,l}$  mit den durch das Alignment eingefügten Leerzeichen
- Messe **Übergangshäufigkeiten  $f(i,j)$**  zwischen allen Paaren von Aminosäuren  $(A_i, A_j)$ , 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

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

# Reale PAM Matrizen

---

- 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:  $M_n = (M_1)^n$
- Dem liegen viele Annahmen zugrunde
  - **Evolutionary Clock Theory**: Evolution verläuft gleichmäßig
    - in der Zeit und in den Sequenzpositionen
    - Proportionalität von Veränderungen
    - Hochrechnung langer Distanzen aus kurzen
  - Keine Insertions oder Deletions
  - Unabhängigkeit der Mutationswahrscheinlichkeit von der Position in der Sequenz (und der Nachbarschaft)

# 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
  - Verschiedene Matrizen testen
  - Alignments prüfen
  - Externes Wissen (Chemie, Strukturen, etc.) hinzuziehen

# BLOSUM Matrizen

---

- Hauptkritikpunkte am PAM Ansatz
  - Nur Verwendung sehr ähnlicher Sequenzen
    - Realistische Zahlen für evolutionär weiter entfernte Sequenzen?
  - Manuelles Alignment
    - Stimmt das?
  - PAM-x für hohe x vervielfältigen eventuelle Fehler in PAM-1
  - Schwierigkeit, das korrekt „x“ für konkrete Alignmentaufgaben auszuwählen
- Anderer (neuerer) Ansatz: BLOSUM Matrizen
  - **B**LOcks **S**Ubstitution **M**atrix
  - Basiert auf **multiplen Alignments evolutionär entfernt, aber homologer Proteinsequenzen**
  - Populärer als PAM Matrizen

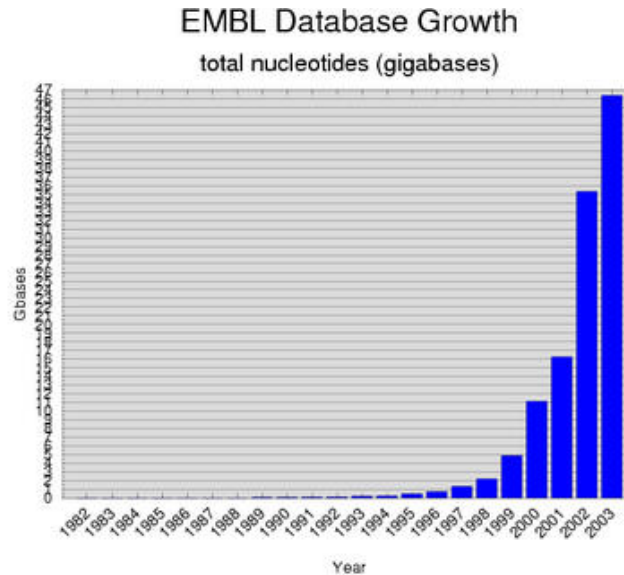
# Inhalt diese Vorlesung

---

- Heuristische Algorithmen zum lokalen Alignment
  - BLAST / BLAST-2
  - (PSI-BLAST, BLAT, PatternHunter, Quasar, OASIS, ...)

# Heuristische Alignierung

---



- Annotation neuer Sequenzen basiert auf Suche nach homologen Sequenzen in [Sequenzdatenbanken](#)
- Datenmenge wächst exponentiell – selbst lineare Algorithmen sind zu langsam
- Gesucht sind Verfahren
  - Sublinearer Laufzeit
  - Möglichst geringer Verlust an Ergebnissen



# Suche in Datenbanken

---

- Gedankenkette
  - Gegeben Sequenz P und Datenbank mit Sequenzen  $S_1, \dots, S_n$
  - Gesucht: Welche Sequenzen in DB sind homolog zu P?
    - Kann nicht beantwortet werden
  - Annäherungen
    - Welche Sequenzen in DB sind sehr ähnlich zu P?
      - Was heißt ähnlich?
    - Welche Sequenzen haben einen hohen Alignment-Score zu P?
      - Berechnung Alignmentsscore dauert zu lange
    - Näherung: Welche Sequenzen haben wahrscheinlich einen hohen Alignmentsscore zu P?
      - BLAST
- Wie viele Hits verlieren wir dabei?

# Sensitivität und Spezifität

---

		Reality	
		+	-
Prediction	+	TruePositive (TP)	FalsePositive (FP)
	-	FalseNegative (FN)	TrueNegative (TN)

- **Spezifität** =  $TP / (TP + FP)$  (Precision)
  - Wie viele der Treffen des Verfahrens sind wirklich welche?
- **Sensitivität** =  $TP / (TP + FN)$  (Recall)
  - Wie viele der echten Treffer findet das Verfahren?
- Oftmals **eine Balance**
  - Algorithmen berechnen einen Score pro Sequenz
  - Hohe Score – Positive; Niedriger Score – Negativ
  - Wenn Score mit Wahrscheinlichkeit für korrekte Klassifikation korreliert, folgt daraus
    - Ergebnismenge klein: SP=hoch, SE=klein
    - Ergebnismenge groß: SP=niedrig, SE=hoch



# Beispiel

---

- Gegeben
  - Sequenz P und Datenbank mit 10.000 Sequenzen
- Algorithmus findet
  - 15 Sequenzen homolog zu P
- In Wahrheit
  - 20 Sequenzen homolog zu P
  - 10 davon hat der Algorithmus gefunden

	Real: Positive	Real: Negative
Alg: Positive	TP = 10	FP = 5
Alg: Negative	FN = 10	TN = 9.975

- Spezifität =  $TP / (TP + FP) = 10 / 15 = 66\%$
- Sensitivität =  $TP / (TP + FN) = 10 / 20 = 50\%$

# BLAST

---

- Altschul, Gish, Miller, Myers, Lipman: „Basic Local Alignment Search Tool“, J Mol Bio, 1990.
- **Heuristische Suche**
  - Sehr schnell, findet aber nicht alle optimalen Alignments
  - Berechnet statistische Signifikanz der Treffer
- **\*\*Die\*\* Erfolgsgeschichte der Bioinformatik**
  - In manchen Darstellungen äquivalent zu „Bioinformatik“
  - Eingesetzt auf NCBI/EBI Server – von der ganzen Welt benutzt
  - Software frei erhältlich
- **Diverse Weiterentwicklungen**
  - Gapped-BLAST und PSI-BLAST (1997) - später

# BLAST Varianten

---

- BLAST gibt es in verschiedenen Ausprägungen
  - Blastn : DNA-Anfrage/DNA-Datensammlung
  - Blastp : Protein-Anfrage/Protein-Datensammlung
  - Blastx : translatierte DNA-Anfrage/Protein-Daten
  - Tblastn : Protein-Anfrage/translatierte DNA-Daten
  - Tblastx: translatierte DNA-Anfrage/DNA-Daten
- Bei DNA/Proteinsuche immer Suche **mit allen sechs Reading Frames**
- Suche in Proteinen läuft etwas anders als in DNA
  - Wegen Unterschied in Größe des Alphabets

# BLAST Parameter

---

- Zunächst
  - Suche in DNA Sequenzen
  - BLAST berechnet lokale Alignments
- Gegeben
  - Suchsequenz P, Datenbank  $DB = \{S_1, \dots, S_n\}$
  - Parameter w: Länge der „Seeds“
  - Substitutionsmatrix M
  - Parameter t: initialer Schwellwert
  - Parameter c: Gesamtschwellwert
    - Wird berechnet in Abhängigkeit von t, M, |DB|, |P|
  - Parameter v: Erwünschte Anzahl Treffer
    - Blast berechnet die v ähnlichsten Subsequenzen

# BLAST Schritt 1 und 2

---

- Schritt 1
  - Bestimme alle **Teilwörter**  $P_1, \dots, P_m$  der Länge  $w$  in  $P$ 
    - Mit Überlappung – keine Partitionierung
- Schritt 2
  - Suche nach **Hits** von  $P_1, \dots, P_m$  in DB mit Score über  $t$ 
    - Idee: Ausreichend gute Alignments besitzen mit hoher Wahrscheinlichkeit einen guten „Seed“
    - Wir suchen nach diesem Seed; Sequenzen ohne guten Seed werden von weiterer Bearbeitung ausgeschlossen
    - Score wird ohne Leerzeichen berechnet
      - Blast (1) kann keine Gaps verarbeiten
    - Suche läuft linear in  $|DB|$  und simultan für alle  $P_i$ 
      - Dank „Keyword Trees“

# BLAST Schritt 3

---

- Schritt 3
  - Für jeden Hit H zwischen DB-Sequenz  $S_j$  und Wort  $P_i$
  - **Verlängere Bereich** um H sowohl in P als auch in  $S_j$ 
    - Extension
    - Alignment erst nach links, dann nach rechts wachsen lassen
      - Wiederum ohne Gaps
    - Solange, bis
      - Sequenz P oder  $S_j$  ist zu Ende
      - Alignment score fällt unter geschätzten Schwellwert c
      - Alignment score fällt signifikant unter bisherige beste Treffer
        - » „Signifikant“ heuristisch bestimmt, abhängig von c und v
    - Ergibt „**Maximal Segment Pairs**“
    - Die v besten MSP sind das Ergebnis

# Beispiel

$W=5$ ,  $t=5$ , Kosten:  $M=+1$ ,  $R=-3$   
 $P=ACGTGATA$   
 $S=GATTGACGTGACTGCAAGTGATACTATAT$

Schritt 1  
Teilwörter



$P_1=ACGTG$   
 $P_2=CGTGA$   
 $P_3=GTGAT$   
 $P_4=TGATA$

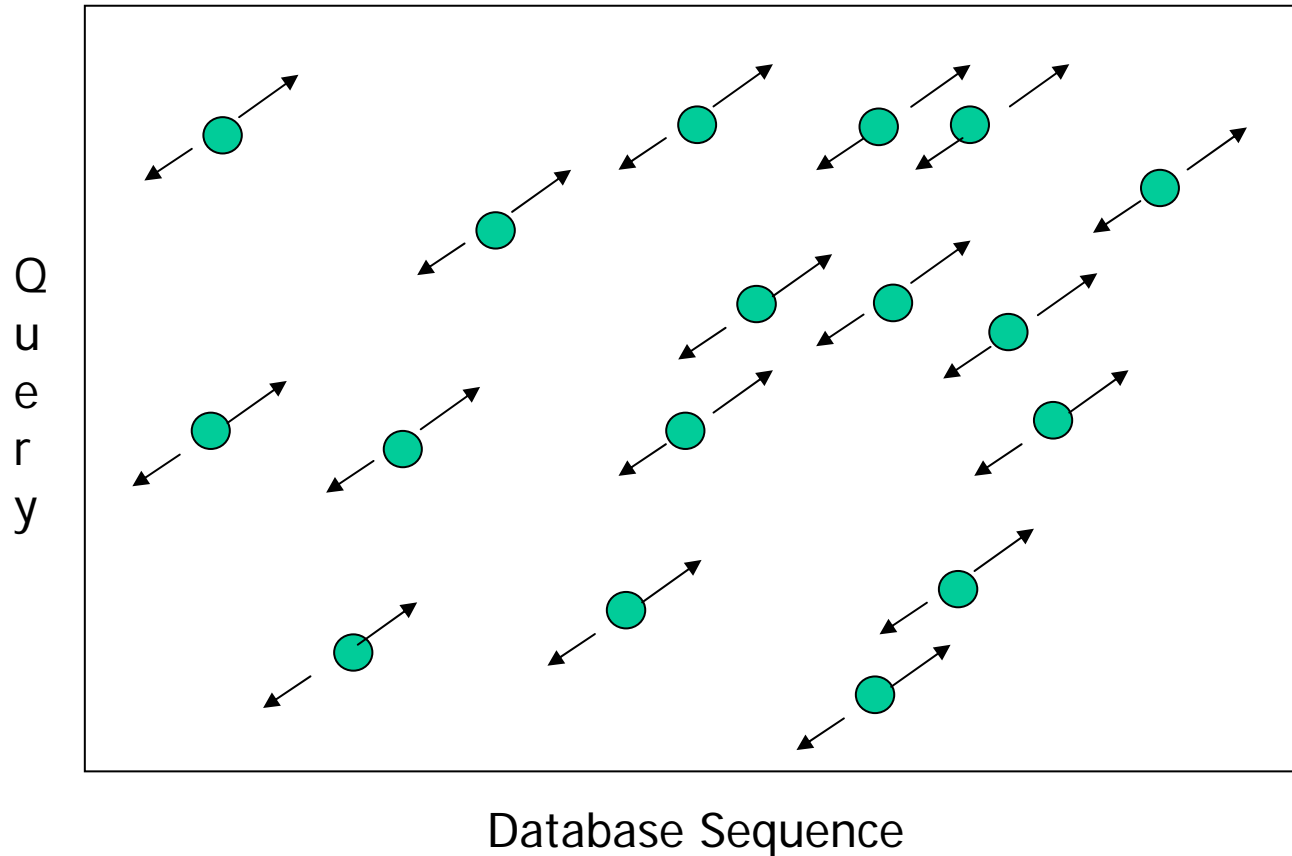
**GATTGACGTGACTGCAAGTGATACTATAT**  
**GATTGACGTGACTGCAAGTGATACTATAT**  
**GATTGACGTGACTGCAAGTGATACTATAT**

Schritt 2  
Hitsuche

Schritt 3  
Verlängerung

<b>GATTGACGTGACTGCAAGTGATACTATAT</b>	
<b>ACGTGATA</b>	5
<b>ACGTGATA</b>	$5+1=6$
<b>ACGTGATA</b>	$6-3=3$
...	...

# Veranschaulichung



- Kein Zusammenfügen mehrerer MSP zu einem Alignment
- Keine Beachtung von Gaps

# Eigenschaften

---

- w oder t vergrößern
  - Anforderungen an Seeds wachsen
  - Es werden weniger Seeds gefunden
  - Es werden weniger Hits verlängert
    - Performanz steigt
  - Gefundene Matches sind mit höherer Wahrscheinlichkeit gute lokale Alignments
    - Spezifität steigt
  - Aber mehr gute Alignments werden übersehen
    - Sensitivität sinkt

# Defaulteinstellungen

---

- Default Einstellungen
  - Proteine
    - BLOSUM62 Matrix
    - $w=4, t=14$
  - DNA
    - Einheitsmatrix (Match +1, Mismatch -3)
    - $w=12, t=12$
    - Also Extension nur um exakte Matches von Teilwörtern der Länge 12

# BLAST Screenshot

NCBI results of **BLAST**

BLASTP 2.2.1 [Apr-13-2001]

**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.

RID: 1011021848-13603-5892

**Query=** gi|2501594|sp|Q57997|Y577\_METJA PROTEIN MJ0577  
(162 letters)

**Database:** All non-redundant GenBank CDS translations+PDB+SwissProt+FIR+PRF  
846,869 sequences; 266,854,569 total letters

Sequences producing significant alignments:

			Score (bits)	E Value
<a href="#">gi 15668757 ref NP_247556.1 </a>	(NC_000909)	conserved hypothet...	248	2e-65
<a href="#">gi 14590690 ref NP_142758.1 </a>	(NC_000961)	hypothetical prote...	89	1e-17
<a href="#">gi 15679011 ref NP_276128.1 </a>	(NC_000916)	conserved protein ...	76	1e-13
<a href="#">gi 15668711 ref NP_247510.1 </a>	(NC_000909)	conserved hypothet...	76	1e-13
<a href="#">gi 15790518 ref NP_280342.1 </a>	(NC_002607)	Vng1536c [Halobact...	72	1e-12
<a href="#">gi 15678181 ref NP_275296.1 </a>	(NC_000916)	conserved protein ...	72	2e-12
<a href="#">gi 15678918 ref NP_276035.1 </a>	(NC_000916)	conserved protein ...	71	3e-12
<a href="#">gi 15679076 ref NP_276193.1 </a>	(NC_000916)	conserved protein ...	69	2e-11
<a href="#">gi 15790787 ref NP_280611.1 </a>	(NC_002607)	Vng1898c [Halobact...	69	2e-11
<a href="#">gi 15887843 ref NP_353524.1 </a>	(NC_003062)	AGR_C_878p [Agroba...	68	4e-11
<a href="#">gi 16120145 ref NP_395733.1 </a>	(NC_002608)	Vng6205c [Halobact...	67	7e-11
<a href="#">gi 16080976 ref NP_391804.1 </a>	(NC_000964)	similar to hypothe...	66	1e-10
<a href="#">gi 17934409 ref NP_531199.1 </a>	(NC_003304)	conserved hypothet...	66	1e-10
<a href="#">gi 15790505 ref NP_280329.1 </a>	(NC_002607)	Vng1518h [Halobact...	65	2e-10
<a href="#">gi 15791176 ref NP_281000.1 </a>	(NC_002607)	Vng2386c [Halobact...	64	7e-10
<a href="#">gi 17544898 ref NP_518300.1 </a>	(NC_003295)	CONSERVED HYPOTHET...	62	2e-09
<a href="#">gi 15790676 ref NP_280500.1 </a>	(NC_002607)	Vng1752c [Halobact...	62	2e-09
<a href="#">gi 16330107 ref NP_440835.1 </a>	(NC_000911)	unknown protein [S...	60	8e-09
<a href="#">gi 17546223 ref NP_519625.1 </a>	(NC_003295)	CONSERVED HYPOTHET...	59	2e-08
<a href="#">gi 15794596 ref NP_284418.1 </a>	(NC_003116)	conserved hypothet...	59	2e-08
<a href="#">gi 15677353 ref NP_274508.1 </a>	(NC_003112)	conserved hypothet...	59	2e-08
<a href="#">gi 15921817 ref NP_377486.1 </a>	(NC_003106)	157aa long conserv...	59	2e-08
<a href="#">gi 11499518 ref NP_070759.1 </a>	(NC_000917)	conserved hypothet...	59	2e-08
<a href="#">gi 15221345 ref NP_176997.1 </a>	(NC_003070)	unknown protein [A...	58	3e-08
<a href="#">gi 17544903 ref NP_518305.1 </a>	(NC_003295)	CONSERVED HYPOTHET...	57	4e-08
<a href="#">gi 11499349 ref NP_070588.1 </a>	(NC_000917)	conserved hypothet...	57	5e-08
<a href="#">gi 15790608 ref NP_280432.1 </a>	(NC_002607)	Vng1658c [Halobact...	56	1e-07
<a href="#">gi 7262999 gb AAF44047.1 AF206717.1</a>	(AF206717)	hypothetical...	56	2e-07
<a href="#">gi 15899493 ref NP_344098.1 </a>	(NC_002754)	Conserved hypothet...	56	2e-07
<a href="#">gi 15675620 ref NP_269794.1 </a>	(NC_002737)	conserved hypothet...	55	2e-07
<a href="#">gi 17546078 ref NP_519480.1 </a>	(NC_003295)	CONSERVED HYPOTHET...	55	2e-07

# BLAST 2

---

- Altschul, et al.
  - „Gapped BLAST and PSI-BLAST: a new generation of protein database search programs“, NAR, 1997
- Verbesserungen
  - Performance verbessern durch andere Extensionsstrategie
    - Extensionen fressen >90% der gesamten Laufzeit
    - Die allermeisten davon werden am Ende trotzdem ignoriert
      - Man will nur die besten  $v$  Sequenzen
    - 1. Idee:  $w/t$  vergrößern – dann sinkt aber die Sensitivität
    - 2. Idee: „Mehr“ verlangen als ein Wort, bevor man Extension beginnt
  - Gaps beachten
    - Mehrere kurze (unterhalb  $t$  liegende) Alignments mit Gaps werden vom alten BLAST übersehen
      - Alle zusammen betrachtet können aber gut genug sein
    - Idee: Auch Ansammlungen kürzerer Hits als Seed benutzen

# Zwei-Hit-Strategie

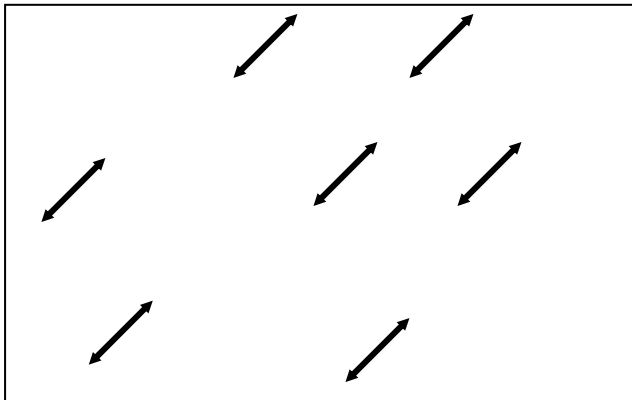
---

- Neue Strategie
  - Extension erfolgt nur, wenn **zwei nicht-überlappende Hits auf einer Diagonale** mit Abstand höchstens  $a$  gefunden wurden
  - Dadurch werden weniger Extensionen ausgeführt
    - Performance steigt
    - Sensitivität sinkt
  - Abhilfe:  $w/t$  verkleinern
- Ergebnis
  - Performance verdoppelt bei gleichbleibender Sensitivität

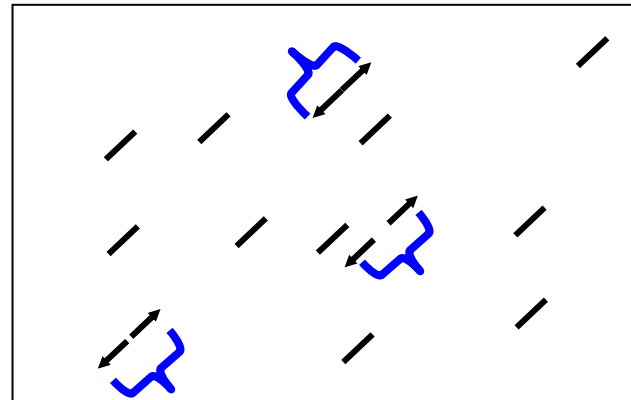
# Illustration

---

Blast 1



Gapped Blast



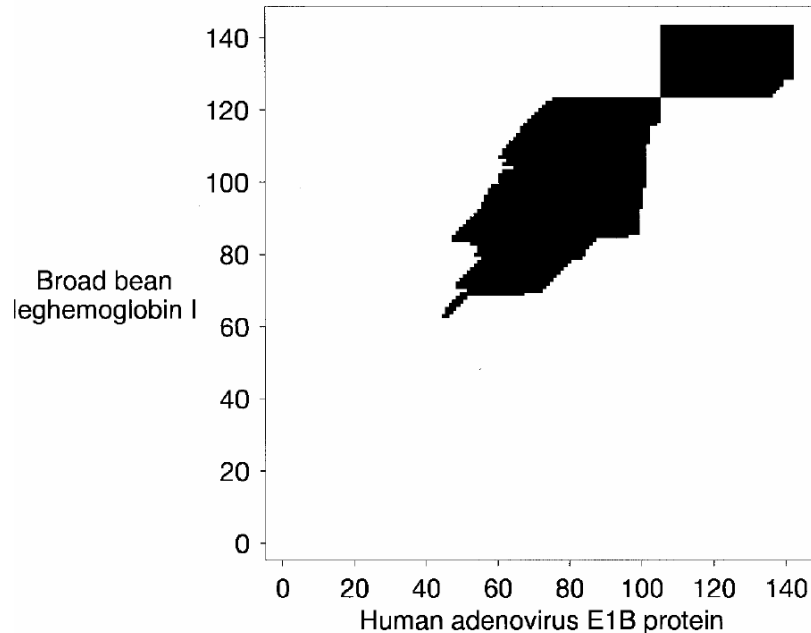
# Gaps in Gapped BLAST

---

- Original: Hits werden verlängert ohne Gaps
  - Verschiedene, nahe beieinander liegende Hits müssen manuell zu einem Alignment kombiniert werden
  - Da das **Alignment zweier benachbarter schwacher Hits statistisch signifikant sein** kann, müssen (relativ) kleine t-Werte benutzt werden
- Gapped Alignment
  - Wenn zwei Hits  $H_1$ ,  $H_2$  in Abstand  $a$  gefunden werden, wird in dieser Diagonale ein **Smith-Waterman Alignment** berechnet
  - Dazu sucht man das Sequenzstück zwischen  $H_1$  und  $H_2$  (incl.) der Länge  $w$  mit dem höchsten Score (ohne Gaps)
  - Von diesem „Seedpoint“ aus wird SW berechnet; Abbruch immer, wenn bestimmte Schranken im Score unterschritten werden
  - Da SW-Alignment sensitiver ist als Extensionen ohne Gaps, kann man  $t$  wieder erhöhen (musste man wegen der zwei-Hits Strategie verringern)
- Weitere **Performanceverbesserung** trotz besserer Sensitivität
  - 500x langsamere Extension durch SW, aber 4000x weniger Extensions durch Erhöhung von  $t$  von 11 auf 13

# Lokale Smith-Waterman Erweiterung

---



- SW wird ausgehend von Seed-Point **in beide Richtungen** berechnet
- Abbruch bei Unterschreiten bestimmter Schranken (abhängig von bisherigen besten Treffern)
- „Leere“ Zellen werden bei der Tabellenberechnung ignoriert
- Variante des Banded Alignment

# Zusammenfassung

---

- BLAST ist eine Heuristik zur Lösung des Alignierungsproblem bei der Suche in Datenbanken
  - Wesentlich schneller als Smith-Waterman (sublinear?)
  - Trotzdem gute Sensitivität
  - Signifikanz von Treffern durch e-Werte gemessen
  - **De-Facto Standard zur Suche in Datenbanken** (EMBL, SwissProt)
  - Weiterentwicklungen: PSI-BLAST, MEGA-Blast
- Es gibt viele weitere Heuristiken
  - FASTA: kaum noch in Gebrauch
  - **BLAT: 500\* schneller als BLAST bei gleicher Sensitivität**; speziell für sehr ähnliche Sequenzen (EST Suche)
  - BioHunter: Kommerzielles Produkt
  - Quasar, Oasis, ...
  - Tricks: Nicht zusammenhängende Seeds, Seedpaare, spezielle Indexstrukturen zur Seedsuche, ...