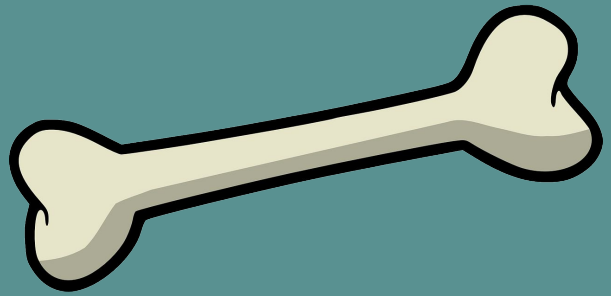
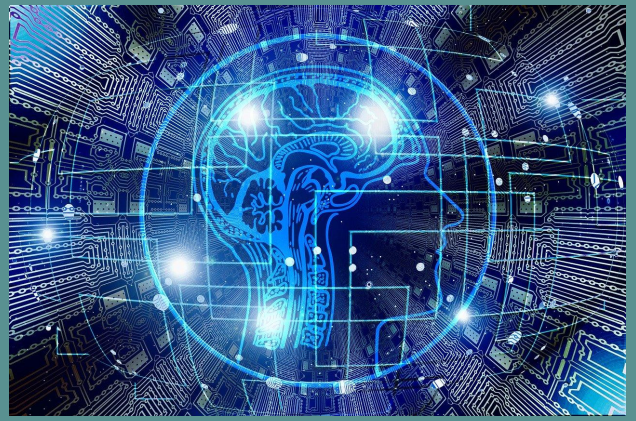
A - C - G - T

Quelle: https://pixabay.com/de/photos/bakterien-elektronenmikroskop-811861/

Quelle: https://pixabay.com/de/illustrations/wale-ozean-meer-natur-s%C3%A4ugetier-1696051/



Quelle: https://pixabay.com/de/vectors/knochen-hund-skelett-157272/



Quelle: https://pixabay.com/de/illustrations/k%C3%BCnstliche-intelligenz-gehirn-hirn-3382507/

Die vier Basen der **Desoxyribonukleinsäure** (DNA):

Adenin - **C**ytosin - **G**uanin - **T**hymin

Suchverfahren für Datenbanken im Bereich der Bioinformatik

Ein Vortrag von Jan-Niklas Neumann

Inhalt

1. Einleitung und Motivation
2. Biochemische Hintergründe und Grundlagen
3. Alignments
4. Suchverfahren
    a. BLAST
    b. Weitere Suchverfahren
    c. Das FastA Dateiformat
    d. BLAST in praktischer Anwendung
5. Herausforderungen und Möglichkeiten für das Datenmanagement in der Bioinformatik

Einleitung und Motivation

Welche Daten werden in der Bioinformatik vor allem gespeichert?

- DNA-Sequenzen
  - lineare Abfolge der Basen
- Proteinsequenzen
  - lineare Abfolge der Aminosäuren

Was ist das Ziel einer Suche in solchen Datenbanken?

- **Alignments** (Übereinstimmungen) zwischen Sequenzen finden
  - eine der wichtigsten Aufgaben in der Bioinformatik

Was motiviert Wissenschaftler, Folgen von Basen und Säuren zu analysieren?

- Forschungen haben Zusammenhang von DNA und Proteinen offengelegt
- physiologische und pathologische Prozesse

Ziele des Sequenzvergleichs

Ziele des DNA Sequenz Vergleichs:

- Gene identifizieren → Muster in Funktionsweise erkennen
- Verwandtschaftsbeziehungen untersuchen → Evolution

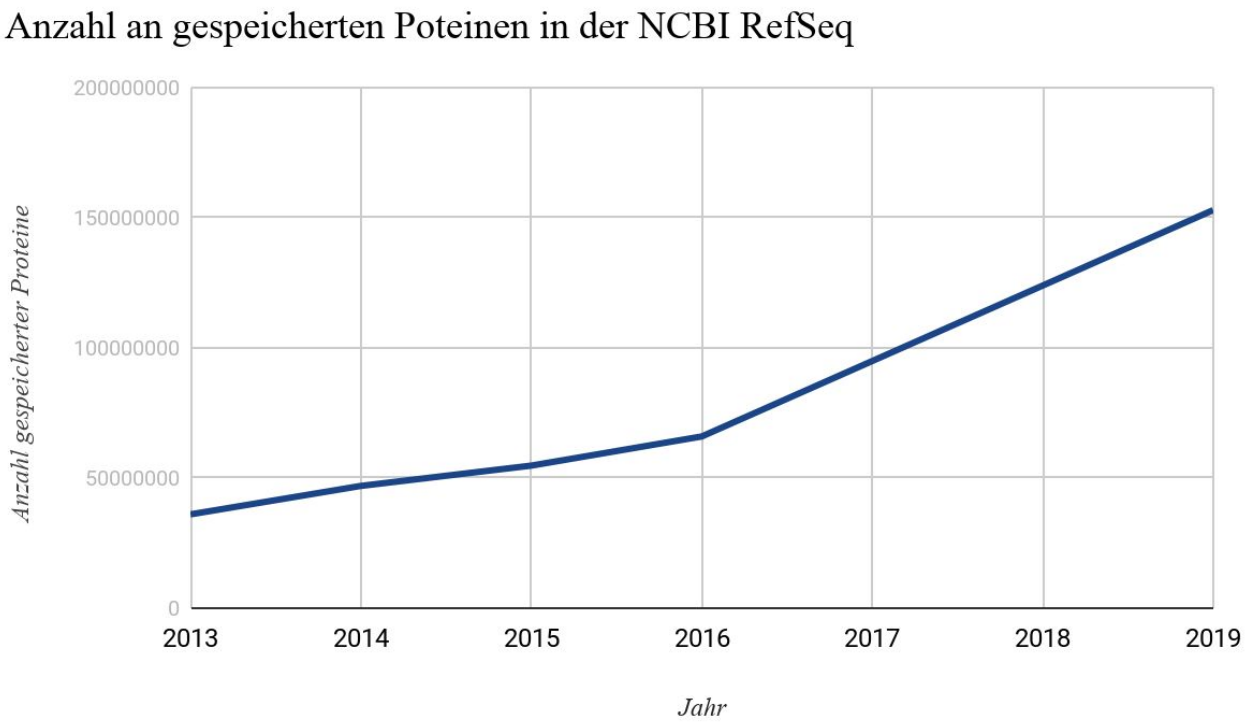
Ziele des Proteinsequenz Vergleichs:

- Struktur und Funktion von Proteinen untersuchen
- Verständnis für die grundlegende Molekularbiologie
- Entwicklungsprozess von Krankheiten

Warum werden dafür Datenbanken benötigt?

- Die Menge an Daten
  - biologische Daten verdoppeln sich ca. alle 15 bis 16 Monate
  - ca. 40.000 Anfragen an Datenbanken (Bioinformatik) täglich
- geforderte Qualität bei Datenanalyse

Zuwachs an gespeicherten Proteinen in der *Reference Sequence database* des *National Center for Biotechnology Information* (NCBI RefSeq):



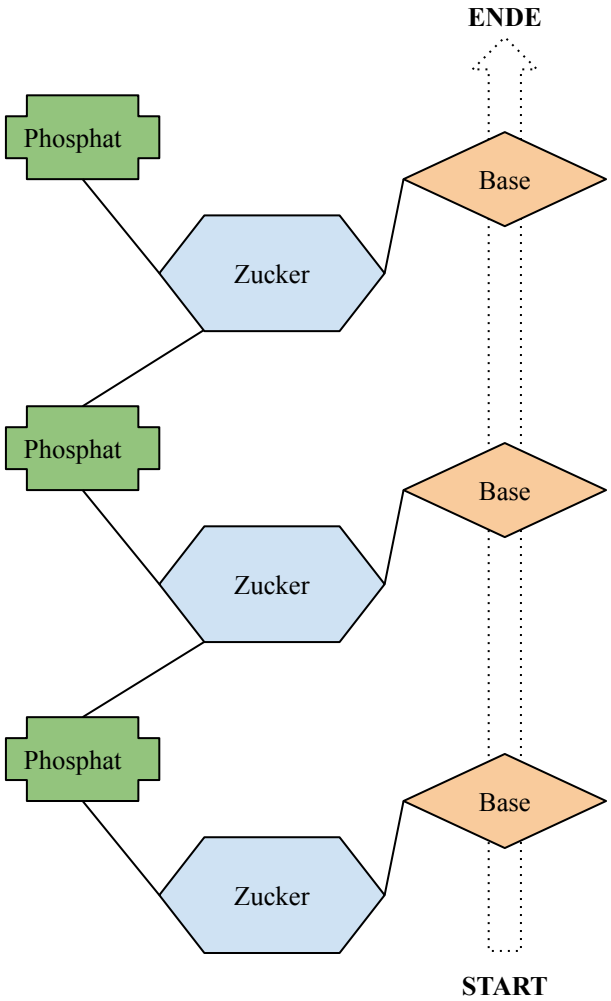
Biochemische Hintergründe und Grundlagen

Die DNA

- Die Desoxyribonukleinsäure -

Wie ist die Desoxyribonukleinsäure (DNA) grundlegend aufgebaut?

- lineare Kette verschiedener Nukleotide
- Bestandteile eines Nukleotids:
  - Phosphat-Anteil
  - Zucker
  - Base
    - Adenin
    - Cytosin
    - Guanin
    - Thymin
      - bei Ribonukleinsäure (RNA) Uracil statt Thymin


ENDE
Phosphat
Base
Zucker
Phosphat
Base
Zucker
Phosphat
Base
Zucker
START

Wie entsteht die Struktur der DNA?

- DNA Stränge über Basen verbunden
- Mögliche Paare:
  - Adenin und Thymin
    - Adenin und Uracil bei RNA
  - Cytosin und Guanin

    → ein Strang kann aus dem anderen abgeleitet werden


Ende Strang A
Start Strang B
C
G
T
A
T
A
Start Strang A
Ende Strang B

Welche Hauptaufgabe erfüllt die DNA?

- kodiert Informationen:
  - kodierende Abschnitte → Gen → genetischer Code
  - nicht kodierende Abschnitte → bestimmen Aktivität von Genen
- genetischer Code in zwei Phasen übersetzt
  - DNA → (m/t)RNA → **Proteinbiosynthese**
- von Generation zu Generation übertragen

Proteine

Wie sind Proteine grundlegend aufgebaut?

- Primärstruktur
  - speichert Proteinsequenz:
    - lineare Folge 20 verschiedener Arten von Aminosäuren
- Sekundärstruktur:
  - dreidimensionale Faltung
    - durch Proteinsequenz definiert
    - bestimmt Funktionsweise
- Tertiär-/Quartärstruktur:
  - spezifiziert Faltung
  - zusammenschließen mehrerer Untergruppen

Welche Hauptaufgaben erfüllen Proteine?

- Faltung bestimmt Funktion:
  - biochemische Reaktionen auslösen
  - interzelluläre Kommunikation
  - Bestandteile des Organismus bilden

Alignments (Übereinstimmungen)

- Das Ergebnis eines Sequenzvergleichs -

Wie werden Alignments gebildet?

- **Ziel:** Übereinstimmungen zwischen Sequenzen suchen
- Voraussetzung:
  - Alphabet **Z** inklusive Leerraum-Symbol (‘**-**’)
    - Leerraum für mehr Treffer
  - Sequenzen **X** und **Y** über **Z**
- Ergebnis:
  - 2D Matrix:
    - Buchstaben geordnet
    - beliebig viele Leerräume pro Zeile …
    - …, aber mindestens ein Buchstabe pro Spalte

Ein Alignment-Beispiel

- Beispieldaten:

(A = Adenin, C = Cytosin, G = Guanin, T = Thymin, Z = Alphabet, - = Leerraum)

Z = {A, C, G, T, -}

X = T A T G A C C

Y = C A G G C A C

Ein mögliches Alignment zwischen Sequenz X und Y

T	-	A	T	G	-	A	C	C
-	C	A	G	G	C	A	-	C

(A = Adenin, C = Cytosin, G = Guanin, T = Thymin, - = Leerraum)

Match und Mismatch

T	-	A	T	G	-	A	C	C
-	C	A	G	G	C	A	-	C

(A = Adenin, C = Cytosin, G = Guanin, T = Thymin, - = Leerraum, = Match, = Mismatch)

- Match → identische Buchstaben
- Mismatch → unterschiedliche Buchstaben
  - entspricht biologisch einer Substitution
    - Basen ausgetauscht

Indel

T	-	A	T	G	-	A	C	C
-	C	A	G	G	C	A	-	C

(A = Adenin, C = Cytosin, G = Guanin, T = Thymin, - = Leerraum)

- Indel → Spalte mit Leerraum

Indels können in Insertions und Deletions unterteilt werden

T	-	A	T	G	-	A	C	C
-	C	A	G	G	C	A	-	C

(A = Adenin, C = Cytosin, G = Guanin, T = Thymin, - = Leerraum, = Insertion, = Deletions)

- Insertion → Leerraum in erster Zeile
- Deletion → Leerraum in zweiter Zeile
- Basen hinzugefügt oder entfernt

Welche Arten von Alignments gibt es?

- Ungapped Alignments:
  - Indels **nicht** erlaubt
- Gapped Alignments:
  - Indels erlaubt

Problem bei Auswahl des Alignments

- viele mögliche Alignments
- abhängig von Länge der Sequenz

	0	1	2	3	4	5	6	7	8
0	1	1	1	1	1	1	1	1	1
1		3	5	7	9	11	13	15	17
2			13	25	41	61	85	113	145
3				63	129	231	377	575	833
4					321	681	1289	2241	3649
5						1683	3653	7183	13073
6							8989	19825	40081
7								48639	108545
8									265729

Tabelle 1: Anzahl möglicher *alignments* in Abhängigkeit von der Sequenzlänge. In der ersten Zeile und der ersten Spalte werden die Wortlängen beider Sequenzen abgebildet. Inhalte entnommen aus Chao, *Sequence Comparison: Theory and Methods*.

Lösung: Alignments bewerten

- Scoring Matrix → Match/Mismatch bewerten

A	2	-1	-1	-1
G	-1	2	-1	-1
C	-1	-1	2	-1
T	-1	-1	-1	2
	A	G	C	T

Abbildung 4: Beispielhafte *scoring matrix* für einen Vergleich von DNA Sequenzen. Inhalt entnommen aus Chao, *Sequence Comparison: Theory and Methods.*

- Affine gap Methode → Indels bewerten

  - (o + k * e) (o = Wert Lücke eröffnen, e = Wert Lücke erweitern, k = Länge Lücke)

- für Proteinsequenzen: Substitutions Matrix

  → Wahrscheinlichkeiten für Austausch von Aminosäuren

Suchverfahren

Grundlegender Aufbau einer Suchanfrage

- Voraussetzung:
  - Datenbank → DNA oder Proteinsequenzen
  - Suchsequenz/Suchstring
- Ziel:
  - abhängig vom Verfahren
    - das beste Alignment finden
    - die besten Alignments finden

Gütekriterien für Suchverfahren / Algorithmen

- Laufzeit
- Sensitivität:
  - richtig-positive Ergebnisse erkennen → Treffer annehmen
- Spezifität:
  - falsch-positive Ergebnisse erkennen → Treffer ablehnen

Smith-Waterman Algorithmus

- liefert optimales Alignment
- Brute-Force Algorithmus:
  - berücksichtigt jede Möglichkeit → sehr sensitiv
  - Einbußen in Performance → hoher Zeitaufwand

```
// ...
Container c = new Container();
for (Sequence s : d) {
        s.useSmithWaterman(q, c);
}
return c;
// ...
```

(c = Container zum Speichern der Ergebnisse, s = Sequenz aus der Datenbank, d = Datenbank, q = Suchstring)

Probleme / Lösungsansätze beim Aufsuchen des optimalen Alignments

- Probleme:
  - Zeitaufwand bei riesiger Masse an Daten
  - Brute-Force Algorithmen alleine zu ineffizient
- Lösungsansätze:
  - ungefähre Treffer zulassen
  - Suche nach passenden Wörtern, nicht nur Buchstaben
  - Treffer mit Punkteschwelle filtern

BLAST

- Das Basic Local Alignment Search Tool -

BLAST: Allgemeines

- mitunter populärstes Tool für Suchanfragen
- verschiedene Ausprägungen (z.B.):
  - BLASTN → DNA Sequenzvergleich
  - BLASTP → Proteinsequenzvergleich
  - BLASTX → übersetzter Proteinsequenzvergleich
- Anfrage über Webseite oder Applikation

BLAST: Ziel

- Sequenzvergleich in **angemessener** Zeit → Alignments finden
- Signifikanz der Treffer berechnen
- heuristischer Ansatz:
  - verbessert Rechenzeit
  - verringert Sensitivität

BLAST: Ergebnis

- Ergebnis in XML (Extensible Markup Language) oder ASN.1 (Abstract Syntax Notation 1)
- Resultate als C++ Objekt
  - z.B. über ASN.1 Format verteilt und später deserialisiert
- Anzeige (z.B.):
  - Alignments:
    - Bewertungen und Signifikanz
    - Beginn und Ende von Übereinstimmungen
    - Positionen von Insertions und Deletions

BLAST: Historische Entwicklung

- 1990:
  - ungapped BLAST → ungapped Alignments
- 1997:
  - gapped BLAST → gapped Alignments
  - PSI(Position-Specific Iterated)-BLAST
    - Verwandschaftsbezüge bei Proteinanalyse

Ungapped BLAST

Ungapped BLAST: Beispiel

- Sequenz X = C T A T C A T T C T G
- Sequenz Y = G A T C C A T C T T
- Scoring Matrix:

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

(A = Adenin, C = Cytosin, G = Guanin, T = Thymin)

Scoring Matrix anwenden

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

Abbildung 8: Zwei Sequenzen in einer Matrix dargestellt und mit der vorher definierten Bewertung versehen. Inhalte übertragen aus Chao, *Sequence Comparison: Theory and Methods.*

(A = Adenin, C = Cytosin, G = Guanin, T = Thymin)

Ungapped BLAST: MSP und HSP

- SP = segment pair
  - diagonaler, zusammenhängender Abschnitt
  - eine übereinstimmende Folge
- MSP = maximal-scoring segment pair
  - SP mit höchster Bewertung
  - hohe Rechenzeit für Berechnung
- HSP = high-scoring segment pair
  - näherungsweises MSP
  - schnellere Berechnung möglich

Ungapped BLAST: HSP berechnen (1/2)

- feste Wortlänge definieren
- Datenbank durchsuchen

Ungapped BLAST: HSP berechnen (1/2)

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

Abbildung 9: Exakte Treffer bei einer festen Wortlänge von $w = 3$. Inhalte übertragen aus Chao, *Sequence Comparison: Theory and Methods.* (A = Adenin, C = Cytosin, G = Guanin, T = Thymin)

Ungapped BLAST: HSP berechnen (2/2)

- Treffer beidseitig diagonal erweitern
  - Ende: aktueller Wert bestimmten Wert schlechter als das beste Ergebnis

    → lokales MSP wird gesucht
- ggf. Punkteschwelle um Suche abzubrechen

  → gefundener Abschnitt wird zurückgegeben

Ungapped BLAST: HSP berechnen (2/2)

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

Abbildung 9: Exakte Treffer bei einer festen Wortlänge von $w = 3$. Inhalte übertragen aus Chao, *Sequence Comparison: Theory and Methods.* (A = Adenin, C = Cytosin, G = Guanin, T = Thymin)

Ungapped BLAST: HSP berechnen (2/2)

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

Abbildung 10: Das durch diagonale Erweiterung gefundene MSP. Inhalte übertragen aus Chao, *Sequence Comparison: Theory and Methods.* (A = Adenin, C = Cytosin, G = Guanin, T = Thymin)

Ungapped BLAST: Probleme

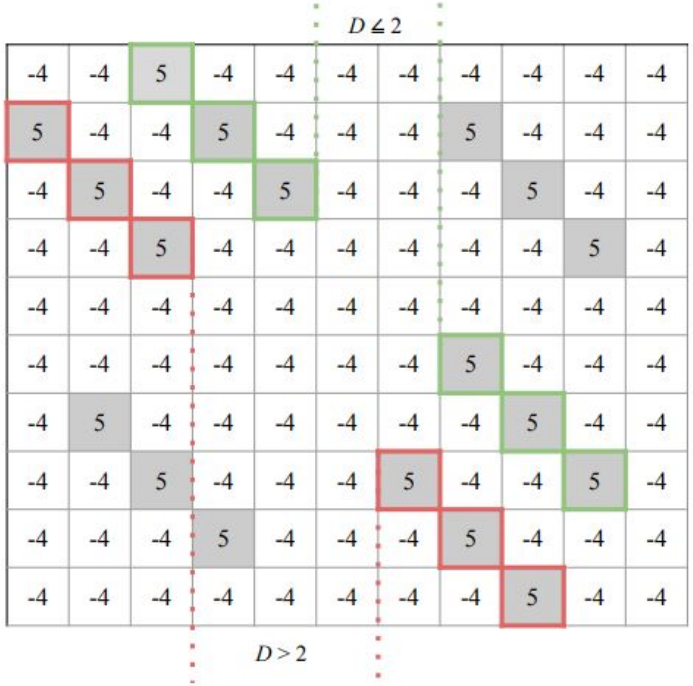
- Rechenaufwand der diagonalen Erweiterung:
  - ca. 90 % der gesamten Rechenzeit
- wichtige HSP’s häufig länger als Wortlänge
- keine konkreten gapped Alignments

Gapped BLAST

Gapped BLAST: Anpassungen

- **Ziel:** Anzahl der Erweiterungen verringern:
  - zwei Treffer benötigt
  - Abstand zwischen Treffern <= zuvor definierter Wert
  - Treffer schneller akzeptiert → Sensitivität
    - mehr Einzeltreffer
    - weniger Erweiterungen
  - für jede Diagonale Position des letzten Treffers speichern

Gapped BLAST: Anpassungen



-4	-4	5	-4	-4	-4	-4	-4	-4	-4	-4
5	-4	-4	5	-4	-4	-4	5	-4	-4	-4
-4	5	-4	-4	5	-4	-4	-4	5	-4	-4
-4	-4	5	-4	-4	-4	-4	-4	-4	5	-4
-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4
-4	-4	-4	-4	-4	-4	-4	5	-4	-4	-4
-4	5	-4	-4	-4	-4	-4	-4	5	-4	-4
-4	-4	5	-4	-4	-4	5	-4	-4	5	-4
-4	-4	-4	5	-4	-4	-4	5	-4	-4	-4
-4	-4	-4	-4	-4	-4	-4	-4	5	-4	-4

(D = Länge der zulässigen Lücke)

Abbildung 11: Darstellung, für welche Treffer eine Erweiterung durchgeführt wird. Chao, *Sequence Comparison: Theory and Methods*

Gapped BLAST: Gapped Alignment

- kann gapped Alignments generieren
- Durchführung von Erweiterungen mit Lücken abhängig von:
  - für den Start:
    - HSP Bewertungen
  - für das Ende:
    - Bereichs Bewertung

PSI-BLAST

PSI-BLAST: Merkmale

- PSI = Position-Specific Iterated
- Verwandschaftsbezüge bei Proteinanalyse
- gapped BLAST hierbei nicht effizient → Sequenzen zu unterschiedlich
- Ablauf:
  - initiales gapped BLAST
  - Ergebnisse → profile/position specific Scoring Matrix (PSSM) aufstellen
  - weitere gapped BLAST Durchläufe → Matrix mit PSSM austauschen
  - PSSM wird so verfeinert → Sensitivität erhöht

Weitere Suchverfahren

Weitere “klassische” Suchverfahren

- FastA:
  - vergleicht Wörter mit zuvor definierter Länge
    → schnell auf Kosten der Sensitivität
    → insgesamt dennoch langsamer als BLAST
- BLAT (BLAST-like alignment tool):
  - bei hoher Ähnlichkeit zwischen Suchsequenz und Datenbank
  - Führt Index für Wort-Matches
- PatternHunter:
  - genauere Definition von Treffern

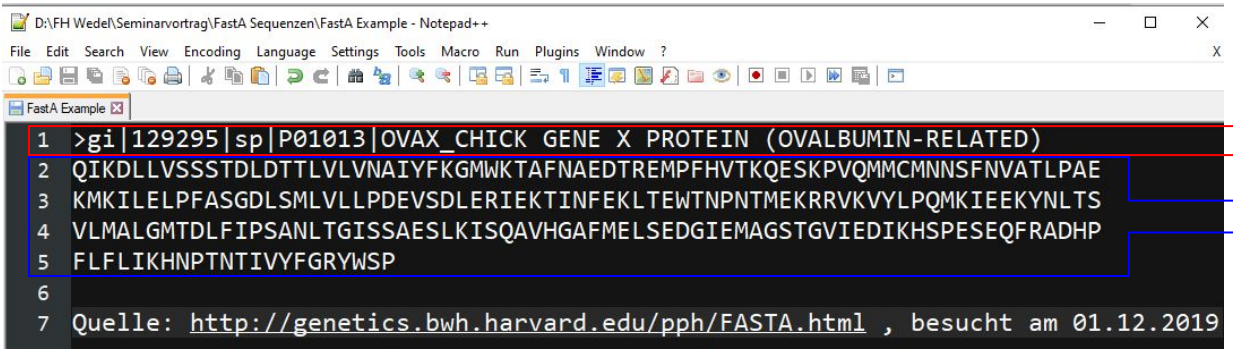
Weitere “neue” Suchverfahren

- SWORD:
  - für Proteinsequenzen
  - Sensitiver Modus: 8-16 x schneller als BLAST
  - Schneller Modus: 68 x schneller als BLAST
  - für die meisten Inputs bessere Sensitivität als BLAST
- DIAMOND:
  - für Proteinsequenzen
  - 20.000 x schneller als BLASTX bei kurzen Sequenzen
  - ähnliche Sensitivität wie BLASTX

Das FastA Dateiformat

FastA Format

Textbasierte Darstellung einer Sequenz



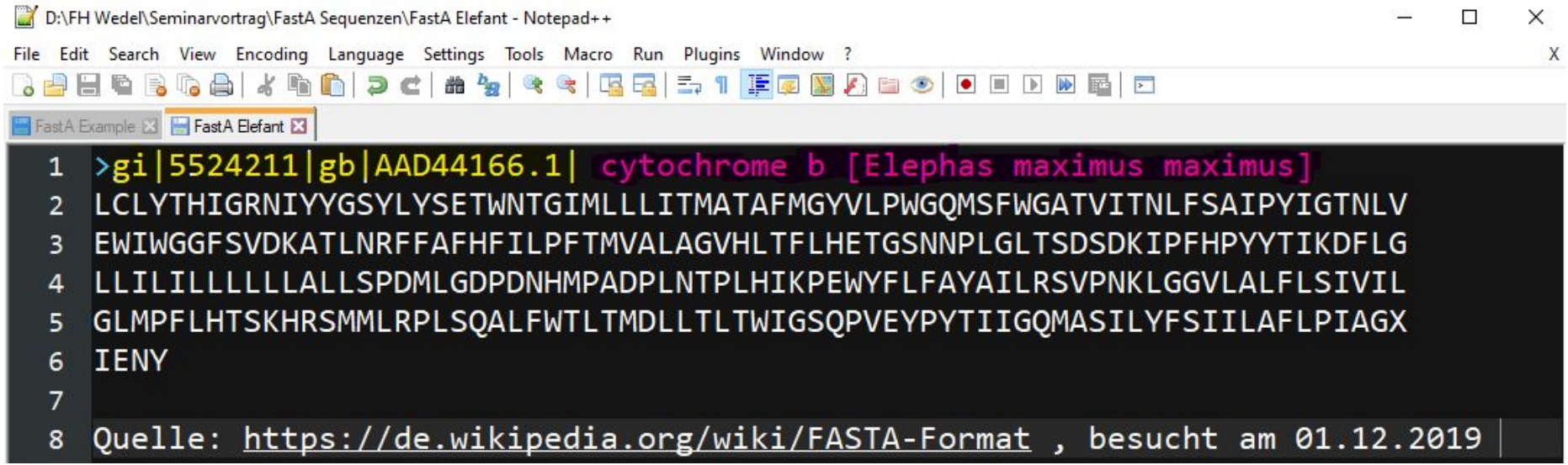
Beschreibung der Sequenz

Inhalt der Sequenz

- Kommentare zwischen Beschreibung und Inhalt möglich
- mit “;” eingeleitet
- in Anwendung häufig nicht erkannt

Aufbau der Beschreibung

>“Kennung””optionale Beschreibung”



Aufbau des Inhalts

Für DNA Sequenzen:

A → Adenin	C → Cytosin	G → Guanin
T → Thymin	U → Uracil	R → G/A
Y → T/C	K → G/T	M → A/C
S → G/C	W → A/T	B → G/T/C
D → G/A/T	H → A/C/T	V → G/C/A
N → A/G/C/T	- → Lücke, unbestimmt lang	

Quelle: Wikipedia, FastA-Format.

Aufbau des Inhalts

Für Protein Sequenzen:

A → Alanin	B → Asparaginsäure/Asparagin	C → Cystein
D → Aspartat	E → Glutamat	F → Phenylalanin
G → Glycin	H → Histidin	I → Isoleucin
K → Lysin	L → Leucin	M → Methionin
N → Asparagin	P → Prolin	Q → Glutamin
R → Arginin	S → Serin	T → Threonin
U → Selenocystein	V → Valin	W → Tryptophan
Y → Tyrosin	Z → Glutamat/Glutamin	X → jede Aminosäure
* → Stop der Translation	- → Lücke, unbestimmt lang	

Quelle: Wikipedia, FastA-Format.

Praktische BLAST Anwendung

https://blast.ncbi.nlm.nih.gov/Blast.cgi

Herausforderungen und Möglichkeiten für das Datenmanagement in der Bioinformatik

Zukunft: Ziele für das Datenmanagement

- Aktuelle und zukünftige Technologien nutzen:
  - Datenintegration → Zusammenführen von Daten
    - Aufstellen von Hypothesen
    - biologisches Wissen entdecken
    - Molekularbiologie erforschen
- Big Data Ansätze umsetzen

Zukunft: Big Data in der Bioinformatik

- relevant wegen Datenmasse
- **Ziele:** Speichermanagement
  - Daten verwalten
  - Daten analysieren
- relationale Datenbanken dafür generell nicht geeignet
- NoSQL Datenbanken:
  - Skalierbarkeit und Flexibilität
  - große Mengen heterogener Daten

Zukunft: Datenanalyse verbessern

- Cloud basierte Ansätze
- Machine Learning
- Data mining Algorithmen
- Ontologies:
  - Daten annotieren und integrieren
- Semantic Web:
  - Daten veröffentlichen
  - untereinander verbinden
  - Netzwerk für Lebenswissenschaften

Zukunft: Weitere potenzielle Anforderungen

- Repräsentation der Daten:
  - Relevanz mobiler Endgeräte

    → responsive Webseiten und Applikationen
  - optisch ansprechendes Design
  - Performance → schneller Erhalt der Informationen

Quellenverzeichnis

Einleitung und Motivation:

Chuming Chen Hongzhan Huang, Cathy H. Wu. “Protein Bioinformatics Databases and Resources". In: Methods in Molecular Biology 1558 (2017), S. 3-39. url: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5506686/ (besucht am 06. 11. 2019).

NCBI. RefSeq Announcements for 2013. url: https://www.ncbi.nlm.nih.gov/refseq/announcements/2013/ (besucht am 13. 11. 2019).

- RefSeq Announcements for 2014. url: https://www.ncbi.nlm.nih.gov/refseq/announcements/2014/ (besucht am 13. 11. 2019).
- RefSeq Announcements for 2015. url: https://www.ncbi.nlm.nih.gov/refseq/announcements/2015/ (besucht am 13. 11. 2019).
- RefSeq Announcements for 2016. url: https://www.ncbi.nlm.nih.gov/refseq/announcements/2016/ (besucht am 13. 11. 2019).
- RefSeq: NCBI Reference Sequence Database. url: https://www.ncbi.nlm.nih.gov/refseq/ (besucht am 13. 11. 2019).

Robert Vaser, Dario Pavlović, Mile Šikić. “SWORD - a highly efficient protein database search”. In: Bioinformatics 32.17 (2016), S. i680-i684. url: https://doi.org/10.1093/bioinformatics/btw445 (besucht am 06.11.2019).

Sung Wing-Kin. Algorithms in Bioinformatics: A Practical Introduction - Database Search. url: https://www.comp.nus.edu.sg/~ksung/algo_in_bioinfo/slides/Ch5_database.pdf (besucht am 06. 11. 2019).

Zvelebil, Baum. Understanding Bioinformatics. Taylor & Francis, 2007. isbn: 978-0815340249.

Quellenverzeichnis

Biochemische Hintergründe und Grundlagen:

Chao, Zang. Sequence Comparison: Theory and Methods. Springer, 2008. isbn: 978-1848003194.

Sung Wing-Kin. Algorithms in Bioinformatics: A Practical Introduction - Database Search. url: https://www.comp.nus.edu.sg/~ksung/algo_in_bioinfo/slides/Ch5_database.pdf (besucht am 06. 11. 2019).

Zvelebil, Baum. Understanding Bioinformatics. Taylor & Francis, 2007. isbn: 978-0815340249.

Alignments:

Chao, Zang. Sequence Comparison: Theory and Methods. Springer, 2008. isbn: 978-1848003194.

Suchverfahren:

Altschul, Madden, Schäfer, Zhang J., Zhang Z., Miller, Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. In: Nucleic Acids Research. 25.17 (1997), S. 3389-3402.

Buchfink, Xie, Huson. “Fast and sensitive protein alignment using DIAMOND”. In: Nature Methods 12.1 (2015), S. 59-60. url: https://www.ncbi.nlm.nih.gov/pubmed/25402007# (besucht am 03.12.2019).

Chao, Zang. Sequence Comparison: Theory and Methods. Springer, 2008. isbn: 978-1848003194.

Madden Thomas. “The BLAST Sequence Analysis Tool". In: The NCBI Handbook [Internet]. 2nd edition (2013). url: https://www.ncbi.nlm.nih.gov/books/NBK153387/ (besucht am 06. 11. 2019).

Robert Vaser, Dario Pavlović, Mile Šikić. “SWORD - a highly efficient protein database search”. In: Bioinformatics 32.17 (2016), S. i680-i684. url: https://doi.org/10.1093/bioinformatics/btw445 (besucht am 06.11.2019).

Sung Wing-Kin. Algorithms in Bioinformatics: A Practical Introduction - Database Search. url: https://www.comp.nus.edu.sg/~ksung/algo_in_bioinfo/slides/Ch5_database.pdf (besucht am 06. 11. 2019).

Wikipedia, FASTA-Format. url: https://de.wikipedia.org/wiki/FASTA-Format (besucht am 01.12.2019).

Quellenverzeichnis

Ausblick:

Chuming Chen Hongzhan Huang, Cathy H. Wu. “Protein Bioinformatics Databases and Resources". In: Methods in Molecular Biology 1558 (2017), S. 3-39. url: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5506686/ (besucht am 06. 11. 2019).