Algorithmische Bioinformatik

Biologische Daten als Strings

Ulf Leser

Wissensmanagement in der Bioinformatik

Ziele für heute

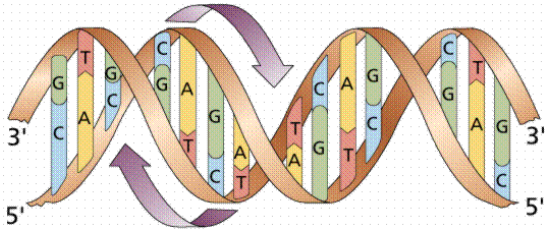
- Wert von Reduktionismus: Genome als Strings
- Anwendungen von Stringmatching in der Bioinformatik
- Varianten des Stringmatching

Inhalt dieser Vorlesung

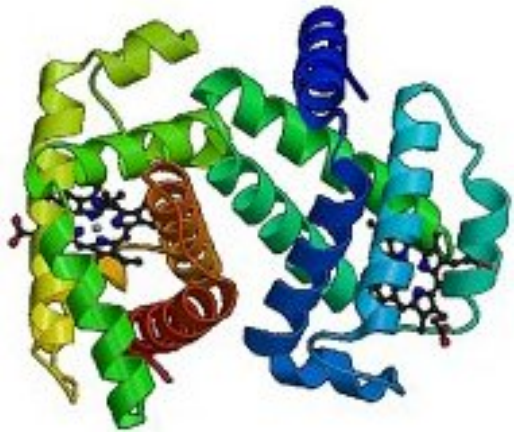
- Warum Stringmatching?
  - Von DNA zu Strings
  - Genomsequenzierung
  - Funktionale Annotation von Sequenzen
- Strings und Matching

Sequenz - Funktion

- DNA
  - Genotyp
  - Vererbung
  - Regulation
  - Produktion von Proteinen

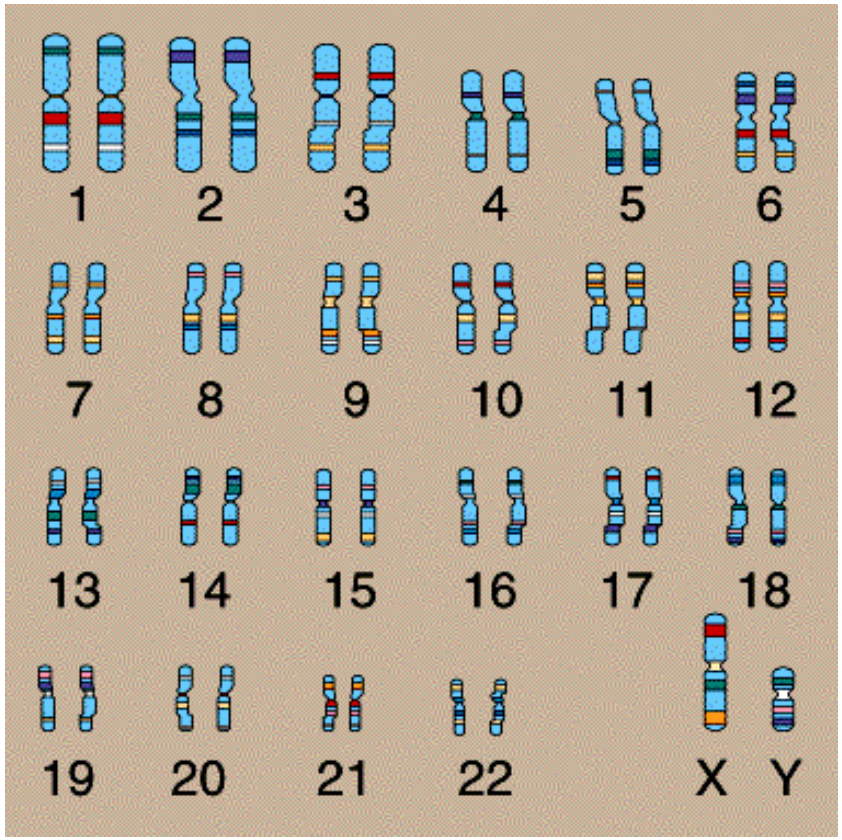


- Proteine
  - Phänotyp
  - Struktur
  - Bindungsverhalten
  - Vielfältigste Funktionen



Das menschliche Genom

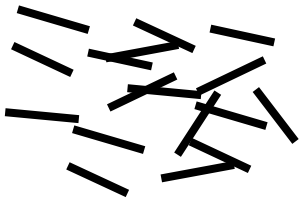
- 23 Chromosomenpaare
  - ~ 3.000.000.000 Basen
- Chromosomen kann man (noch) nicht direkt und als Ganzes sequenzieren
- Erste Aufgabe: Kurze, stabile und kopierbare Sequenzabschnitte produzieren



Inhalt dieser Vorlesung

- Warum Stringmatching?
  - Von DNA zu Strings
  - Genomsequenzierung
  - Funktionale Annotation von Sequenzen
- Strings und Matching

Sequenzierung



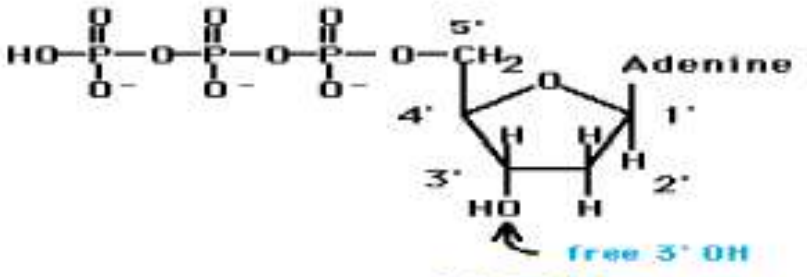
- Gegeben: Bruchstücke unbekannter Sequenz
- Gesucht: Deren Sequenz
- (Radioactive) Dideoxy Sequencing
  - Auch „Sanger sequencing"
  - Verfahren von Sanger et al., 1972

Frederick Sanger, 1918 - 2013

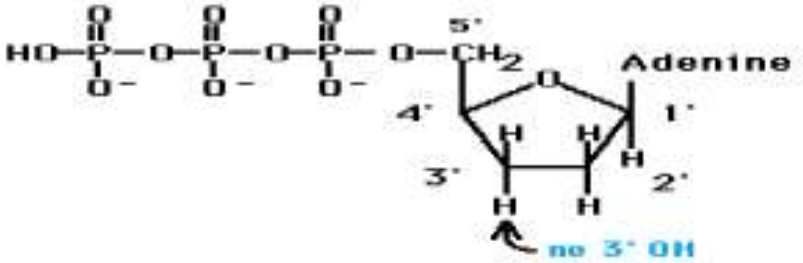
Sanger Sequencing

- Voraussetzungen
  - Sequenz hat einen definierten Anfang
    - Teil des Clonierungsvektors
    - Dient als Bindungsstelle für Primer
  - Polymerase
    - Bindet an doppelsträngigen Abschnitt
    - Verlängert einsträngige DNA entlang des Templates
- Deoxy versus Dideoxy Nucleotide
  - DNA besteht aus Deoxy Nucleotiden (dNTP)
  - Einbau von Dideoxy Nucleotiden (ddNTP) möglich
  - ddNTP stoppt Polymerase

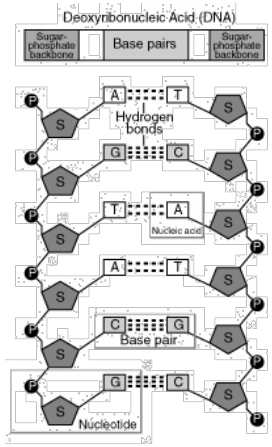
dNTP versus ddNTP

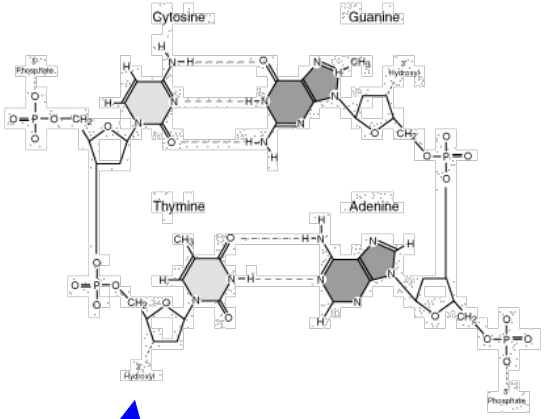


dNTP



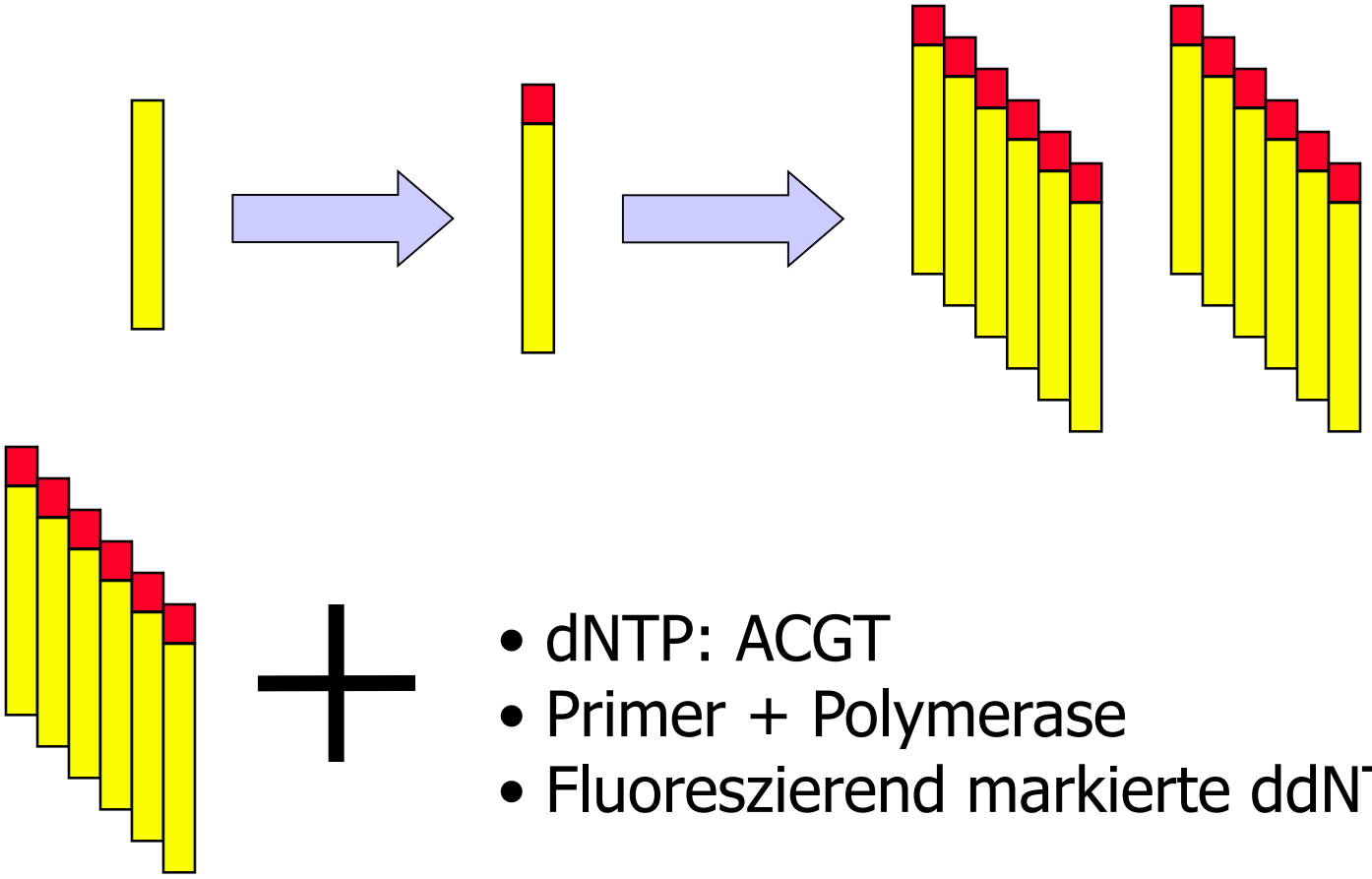
ddNTP





- Dideoxy-Base: keine freie OH Gruppe
  - Werden mit konzentrationsabhängiger Wahrscheinlichkeit eingebaut
  - Danach können keine weiteren Basen angehängt werden

Schritt 1 und 2

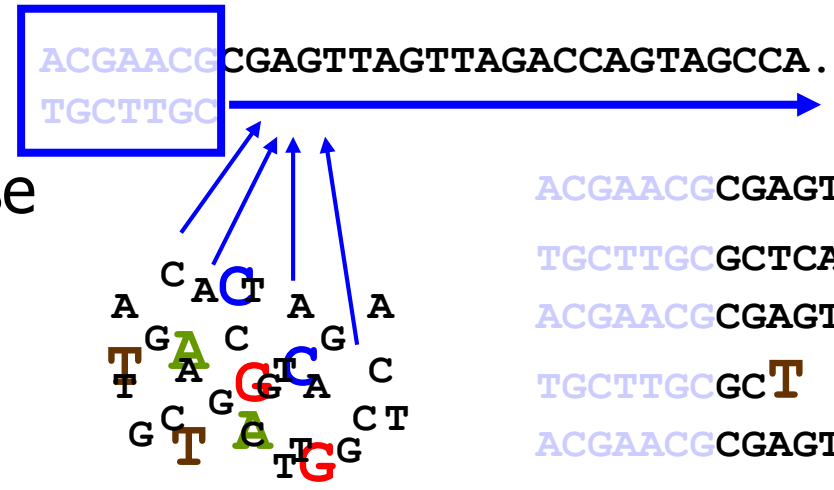


- dNTP: ACGT
- Primer + Polymerase
- Fluoreszierend markierte ddNTP: ACGT

Schritt 3

Primer Template

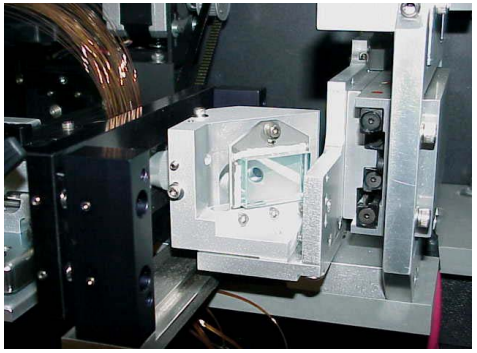
ACGAACGCGAGTTAGTTAGACCAGTAGCCA...



ACGAACGCGAGTTAGTTAGACCAGTAGCCA...
TGCTTGCGCTCAAT
ACGAACGCGAGTTAGTTAGACCAGTAGCCA...
TGCTTGCGCT
ACGAACGCGAGTTAGTTAGACCAGTAGCCA...
TGCTTGCTCTCAATC
ACGAACGCGAGTTAGTTAGACCAGTAGCCA...
TGCTTGCTCTCAATCAATCTG
ACGAACGCGAGTTAGTTAGACCAGTAGCCA...
TGCTTGCT

Schritt 4

Laser &
Detektoren



ACGAACGCGAGTTTA

ACGAACGCGAG

ACGAACGCGAGTTAGTT

ACGAACGCGAGTTAGTTAGT

ACGAACGCGA

Gel / Kapillar
Elektrophorese

ACGAACGC

ACGAACGCG

ACGAACGCGA

ACGAACGCGAG

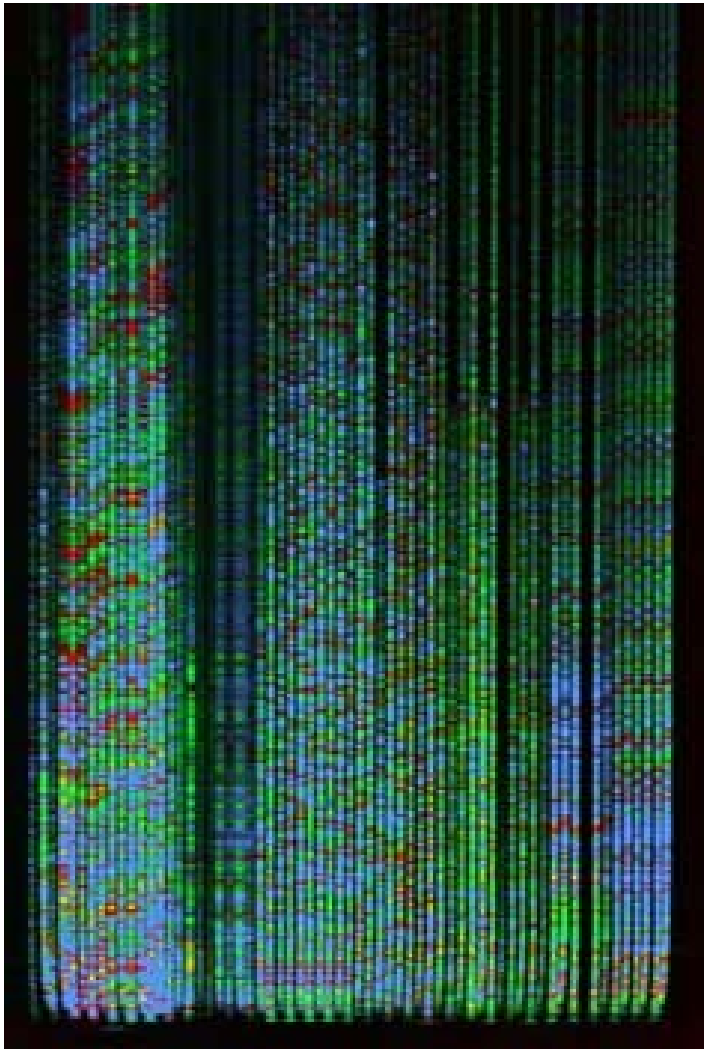
ACGAACGCGAGT

ACGAACGCGAGTTA

ACGAACGCGAGGTTAG

Primäres Ergebnis

- „Aktuelle" Geräte
  - Kapillarelektrophorese
  - Bis zu 96 parallele Kapillare
- Früher (original)
  - Radioaktive Markierung
  - 4 Mischungen (A,G,T,P)
  - 4 Gele (Linien)

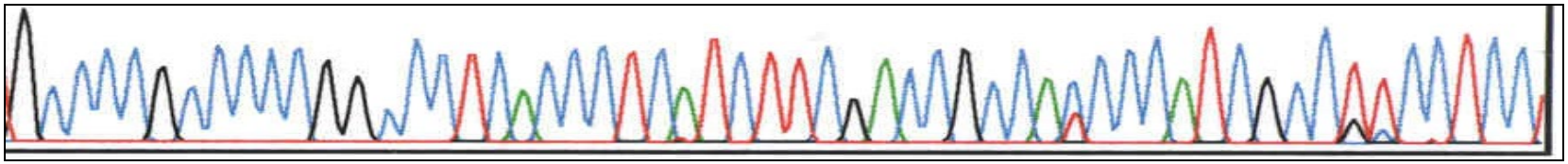


Vom Tracefile zur Sequenz

- Peak-Detection: Von Farbintensitäten zu Peaks
- Base Calling: Berechnung der wahrscheinlichsten Sequenz
- Assembly: Berechnung der wahrscheinlichsten ursprünglichen Gesamtsequenz
- Finishing: Füllen von Lücken durch gezielte weitere Experimente

Traces

- Signalverarbeitung



- Übersetzung in Traces
  - 4 Arrays, jedes für eine Farbe
  - Intensitätswerte entlang der Zeitachse
- Base calling
  - Peaks entdecken (Abstände werden sukzessive kleiner)
  - Base zuordnen und Qualität bewerten
  - Wahrscheinlichkeit der Hauptbase jeder Position

Assembly

- Basis: Finden von Überlappungen von Sequenzen
- Redundanz ist
  - Notwendig: Verbindung von Teilstücken nur durch Überlappungen
  - Konfliktträchtig: Widersprüche, Mehrdeutigkeit
- Berücksichtigung von Sequenzierfehlern
- Approximation der wahrscheinlichsten Anordnungen

Fehler ?

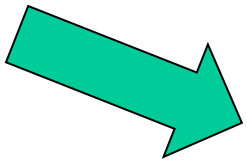
Greedy Algorithmus?

accgttaaagcaaagatta

aagattattgaaccgtt

aaagcaaagattattg

attattgccagta



accgttaaagcaaagatta
aaagcaaagattattg
aagattattgaaccgtt
attattgccagta

29 Basen

accgttaaagcaaagatta
aaagcaaagattattg
aagattattgaaccgtt
attattgccagta

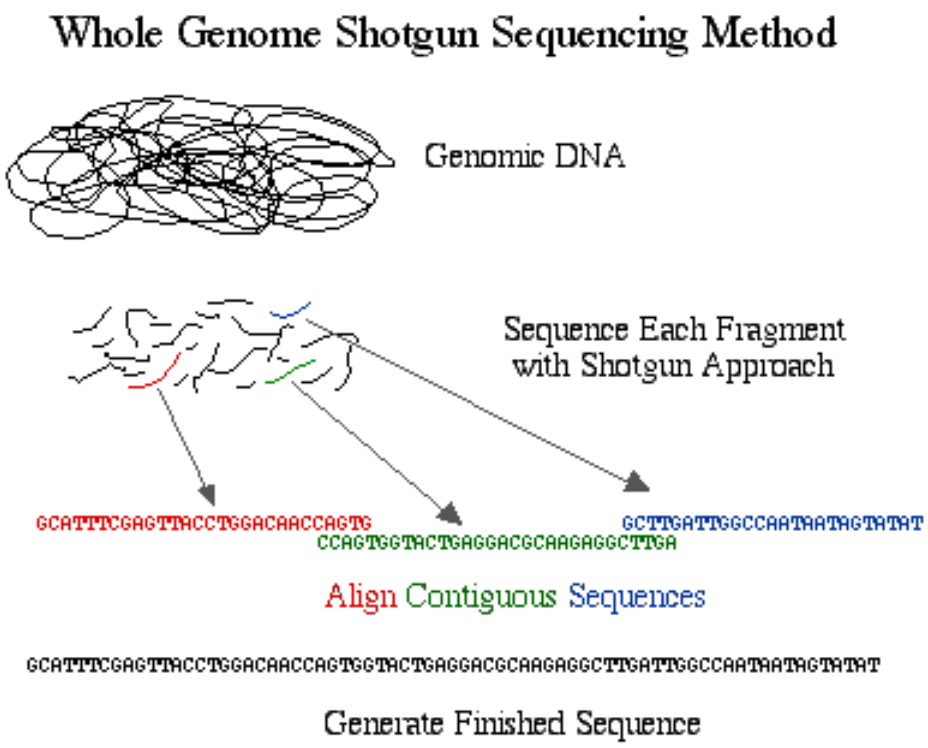
39 Basen

Abstrakte Formulierung

- SUPERSTRING
  - Geg.: Menge S von Strings
  - Ges.: String T so, dass
    - (a) $\forall s \in S: s \in T$ (s Substring von T)
    - (b) $\forall$ T', für die (a) gilt, gilt: $|T| \leq |T'|$ (T ist minimal)
- NP-vollständig (in der Zahl der Sequenzen)

- Assembly
  - Verschärfungen von SUPERSTRING wegen Fehler in Sequenzen (s „ungefähr Substring" von T)
  - Orientierung der Strings nicht klar

Problemdimension: Whole Genome Shotgun



- Zerbrechen von kompletten Genomen in Stücke 1KB-100KB
- Alle Stücke (an-) sequenzieren
- Celera:
  - Drosophila: Genom: 120 MB, 3.200.000 Reads
  - Homo sap.: Genom: 3 GB, 28.000.000 Reads
- Schnelle Algorithmen notwendig

Second (Next) Generation Sequencing, NGS

- New generation of sequencers since ~2005
  - Illumina, Solexa, 454, Solid, …
- Much higher throughput
  - ~15 TB raw data in 3-5 days
  - ~600 GB processed data/week
  - Cost for sequencing a genome down to ~1.000 USD
  - But: Shorter reads (~100bp)
- 3rd generation sequencers
  - Single molecule sequencing
  - A (human) genome in a day
  - Sequence every human
  - Sequence different cells in every human

Illumina HiSeq 2000. DNAVision

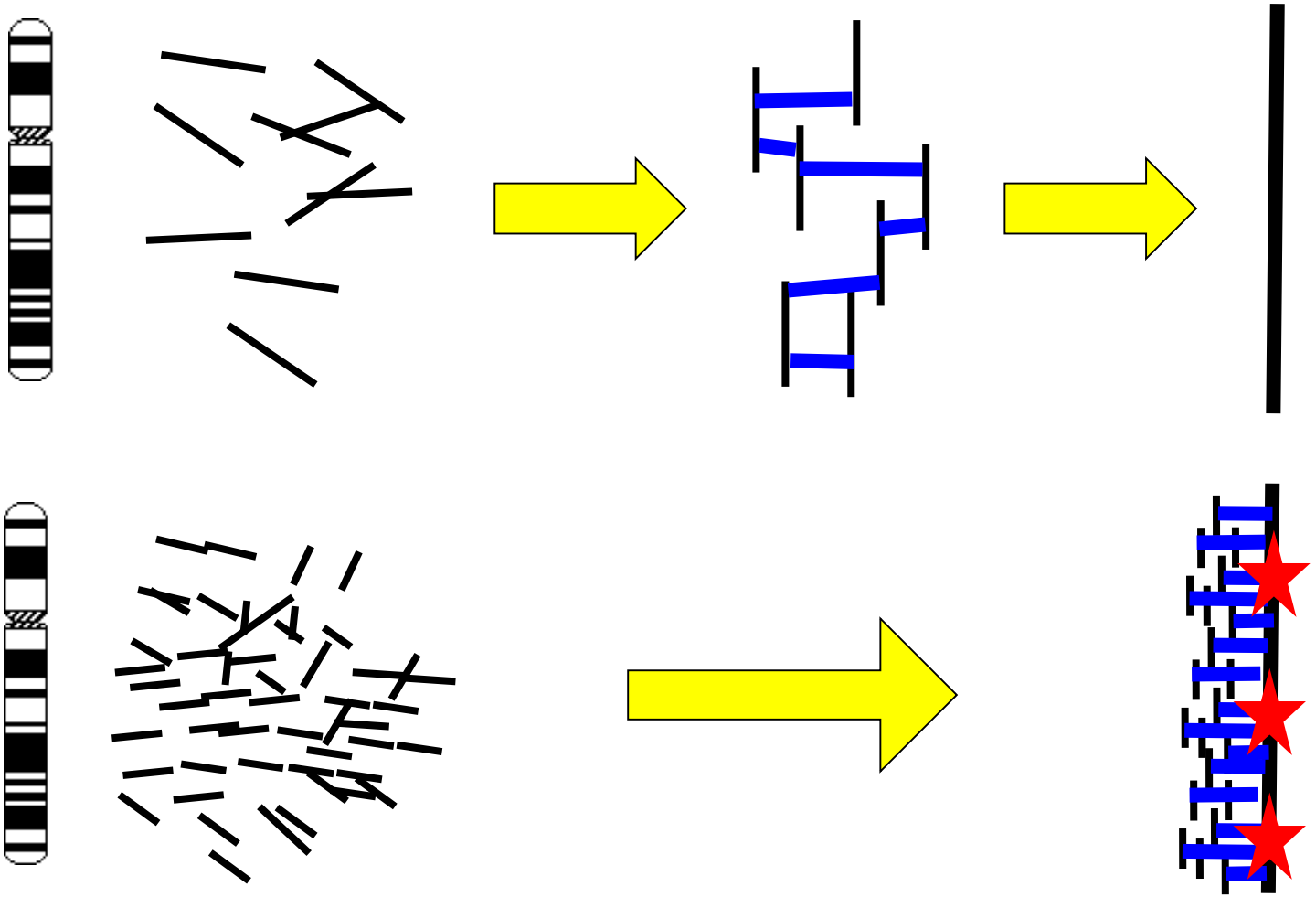
2016

- 600GB / day, 18.000 genomes per year
- $1,000 genome at 30x coverage
  - Amortized over 18,000 genomes per year over four-year period
- (Not cheap)

Read Mapping

- Sequenzieren mit NGS erzeugt kurze Reads
  - De-Novo Assembly kaum möglich (repeats)
- Daher: Read-Mapping gegen Referenzgenom
  - Problem: Mismatches können verschiedene Quellen haben
  - Fehler in den Reads
  - Fehler im Referenzgenom
  - Natürliche Variation
  - (Krankheitsassoziierte) Mutation
- Abdeckung und Entropy der Position, Häufigkeit der Base an der Stelle in Population, Qualität des Reads, Quality-Score der Base im Read, …

New Task: Read Mapping & SNV Detection



Inhalt dieser Vorlesung

- Warum Stringmatching?
  - Von DNA zu Strings
  - Genomsequenzierung
  - Funktionale Annotation von Sequenzen
- Strings und Matching

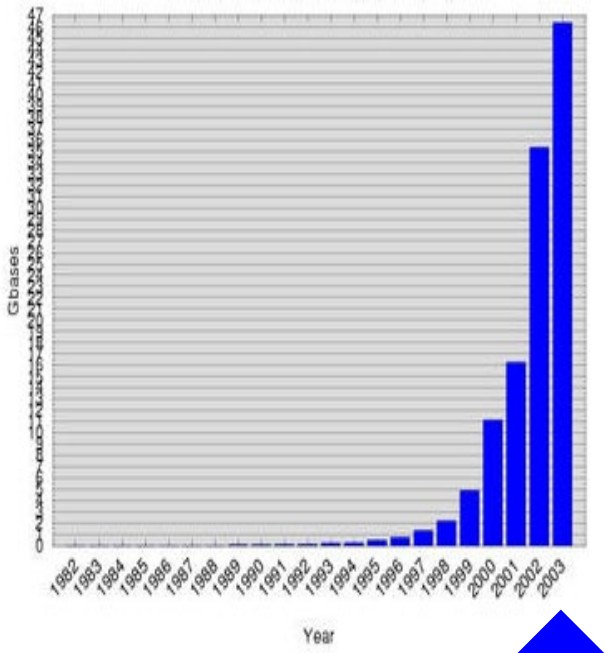
Funktionale Annotation

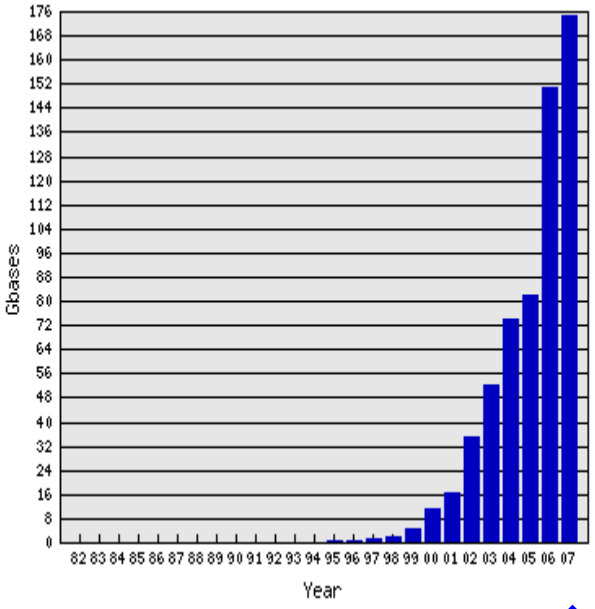
- DNA Sequenzen bestimmen Proteinfunktionen
  - Gensequenzen => Proteinsequenz
  - Proteinsequenzen => Struktur
  - Struktur => Funktion
- Beobachtung (Grundpfeiler der Bioinformatik)
  - Gleiche Sequenzen – gleiche Funktion
  - Sehr ähnliche Sequenzen – sehr ähnliche Funktion
  - Etwas ähnliche Sequenzen – verwandte Funktion?
- Insbesondere: Sequenzen aus verschiedenen Spezies

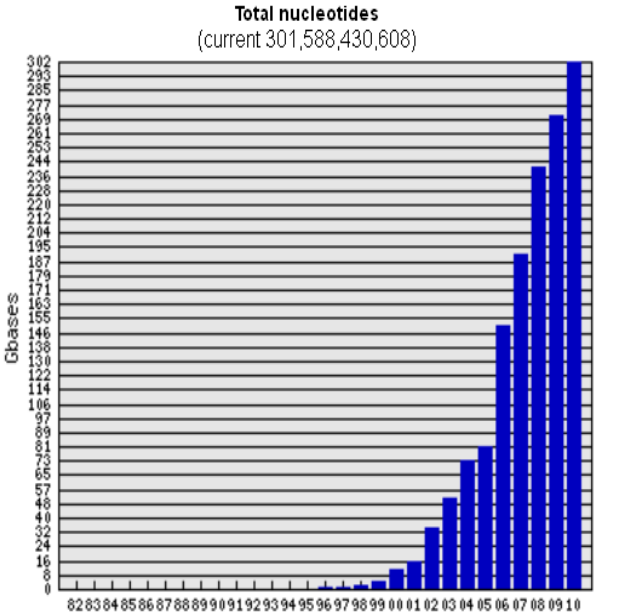
Standardvorgehen

- Gegeben: Eine frisch sequenzierte DNA Sequenz
- Annotationspipeline
  - Suche nach ähnlichen Gensequenzen
  - Suche nach ähnlichen Promotersequenzen
  - Suche nach ähnlichen Proteinen (Übersetzung- Rückübersetzung)
  - Vorhersage neuer Genen durch Programme (trainiert auf bekannten Gensequenzen)
  - Suche nach ähnlichen Proteindomänen durch Programme (trainiert auf bekannten Proteindomänen)
  - ...
- Alternative: Experimentelle Überprüfung
  - Teuer, auch nicht fehlerfrei
  - Ethische / technische Machbarkeit

Problemdimension







Inhalt dieser Vorlesung

- Warum Stringmatching?
  - Von DNA zu Strings
  - Genomsequenzierung
  - Funktionale Annotation von Sequenzen
- Strings und Matching

Zeichenketten

- Definition
  Ein String S ist eine von links nach rechts angeordnete Liste von Zeichen eines Alphabets Σ
  - *|S| ist die Länge des Strings*
  - *Positionen in S sind 1,...,|S|*
    - *Wir zählen ab 1*
  - *S[i] ist das Zeichen an der Position i im String S*
  - *S[i..j] ist der Substring, der an Pos. i beginnt und an Pos. j endet*
    - *S[i..j] ist ein leerer String, falls i > j*
  - *S[1..i] heißt Präfix von S bis zur Position i*
  - *S[i..] ist das Suffix von S, welches an Position i beginnt*
  - *Echte Präfixe und echte Suffixe umfassen nicht den gesamten String S und sind nicht leer*

Problemklassen

- Exaktes Substring-Matching (Patternmatching, Matching)
  - Gegeben: Strings P, T
  - Gesucht: Alle Auftreten von P in T
  - Variante: Gegeben $P_1,...,P_n$, T: Vorkommen aller P_i in T
- Approximatives Matchen
  - Gegeben: Strings S, T
  - Gesucht: Wie ähnlich sind sich S und T?
  - Variante: Ist S in T mit höchstens k Fehlern enthalten?
- Approximatives lokales Matching
  - Gegeben: Strings S, T
  - Gesucht (Read-Mapping): Substring in T, der ähnlich zu S ist?
  - Variante (BLAST): Substring in T, der ähnlich zu Substring in S ist?

Datenbank-Varianten

- Gegeben: Datenbank D von Sequenzen, String P
- Exaktes Substring-Matching: Alle $d \in D$, die P enthalten
- Approximatives Matchen: Alle $d \in D$, die zu P ähnlich sind
- Approximatives lokales Matching: Alle $d \in D$, die einen zu P ähnlichen Substring enthalten
- Top-K: Die k zu P ähnlichsten Sequenzen $d \in D$
- Lokales Top-K: Die k Sequenzen $d \in D$, die die k zu P ähnlichsten Substrings enthalten

Übersicht Exaktes Matching

- Naiver Algorithmus: O(n*m)
- Z Algorithmus: O(m+n)
  - Wird gerne in anderen Verfahren zum Pre-Processing verwendet
- Boyer-Moore: Sublinear im Average Case
  - Worst Case O(n*m), aber Average Case sublinear
  - Erweiterung zu linearem Worst-Case
- Knuth-Morris-Pratt: O(m+n)
  - Voraussetzung für Aho-Corasick zur Suche nach mehreren Pattern
- Später: Indexstrukturen, z.B. Suffixbäume
  - O(n+k) (nach Preprocessing: O(m))

Selbsttest

- Wie funktioniert Sanger-Sequenzierung? Warum ist die Länge der Reads dabei beschränkt?
- Nehmen wir an, dass in einer Mischung 95% dioxy und 5% dideoxy-Basen enthalten sind. Stellen Sie die Formel für die Wahrscheinlichkeit auf, dass dadurch Reads der Länge exakt k entstehen (k als Parameter)
- Beschreiben Sie formal das Read-Mapping Problem
- Führen Sie das Assembly-Problems auf ein klassisches NP vollständiges Problem zurück
- Was ist paired-end Sequenzierung?