

Bioinformatik

Für Biophysiker

Wintersemester 2003 / 2004

Dozenten: Cornelius Frömmel, Robert Preisner,
Hans-Peter Herzl, Ulf Leser, Christian Meisel

Ulf Leser

Wissensmanagement in der
Bioinformatik



Wissensmanagement in der Bioinformatik

- Neuer Lehrstuhl seit 10/2002
- Schwerpunkte
 - Algorithmen der Bioinformatik
 - Management molekularbiologischer Daten
 - Datenintegration
 - Text Mining
- Laufend möglich
 - Studienarbeiten
 - Diplomarbeiten
 - ... oft in Kooperation mit Bioinformatikfirmen
 - ... oft in Kooperation mit Bioinformatikinstituten

Mein Teil

- Vorlesung
 - Stringvergleiche
 - Suffixbäume
 - Dynamische Programmierung
 - BLAST + friends
 - Multiples Stringalignment
- Praktikum: 23.2.2003 – 27.2.2003
 - Mit Silke Trissl
 - Programmierung Dotplots & Smith-Waterman

Informationen

- Zu Vorlesung und Praktikum

- http://www.informatik.hu-berlin.de/wbi/teaching/ws0304/bioinfo_bpi/index.html

- Bei Fragen

- Leser@informatik.hu-berlin.de, Tel. 2093 3902

- Trissl@informatik.hu-berlin.de, Tel. 2093 5487 (tw)

Literatur

- Primär

- Dan Gusfield: „Algorithms on Strings, Trees, and Sequences“, Cambridge University Press, 1997 (ca. 60 Euro)

- Weitere

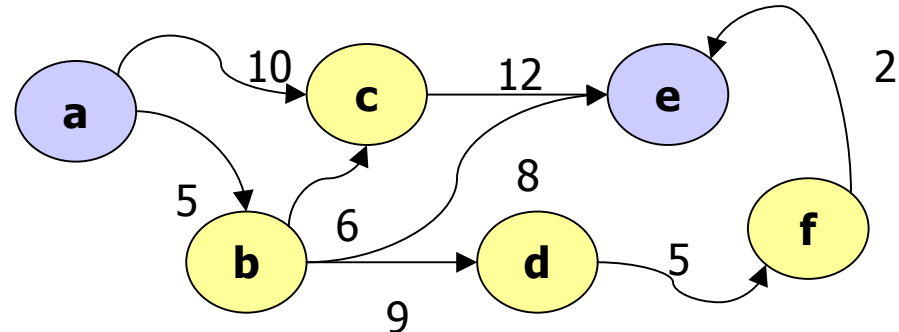
- Joachim Böckenhauer, Dirk Bongartz: „Algorithmische Grundlagen der Bioinformatik“, Teubner, 2003 (ca. 30 Euro)
- David Mount: „Bioinformatics. Sequence and Genome Analysis“, Cold Spring Harbour Press, 2001 (ca. 70 Euro)
- Gibson & Muse: „A primer of genome sciences“, Sinauer Associates, 2001 (ca. 50 Euro)

Wichtiger Begriff: Komplexität

- Ziel: Angabe der Effizienz eines Algorithmus unabhängig von
 - Rechner, Programmiersprache, Hauptspeicher, ...
 - Erfassen der „Essenz“ eines Algorithmus
- Komplexität hat viele Facetten
 - Speicherplatzverbrauch, Laufzeit
 - Average-Case, Worst-Case
- Komplexität wird gemessen in der Länge der Eingabe n
 - Anzahl Zahlen, die es zu sortieren gilt
 - Anzahl Knoten eines Graphen, den es zu durchsuchen gilt
 - Länge Sequenzen, die es zu vergleichen gilt
 - ...
 - [Nicht immer einfach zu bestimmen ...]

Beispiel

- Kürzeste Wege in einem Graphen



- Naives Verfahren

- Alle Wege ausprobieren
 - Ausgehend von a
 - Alle Wege a-X beschreiten
 - Rekursiv weitermachen und Wegsumme dabei addieren
 - Kleinstes Ergebnis merken
- Komplexität: Exponentiell

- Dijkstra

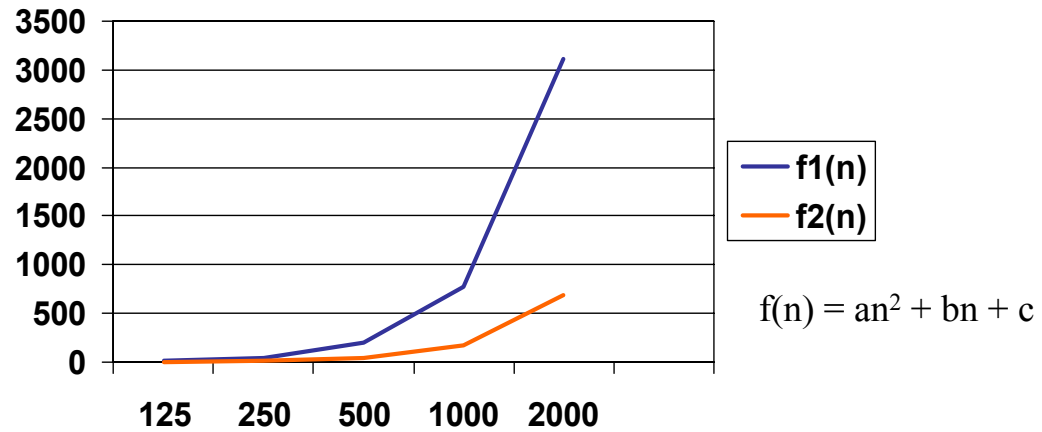
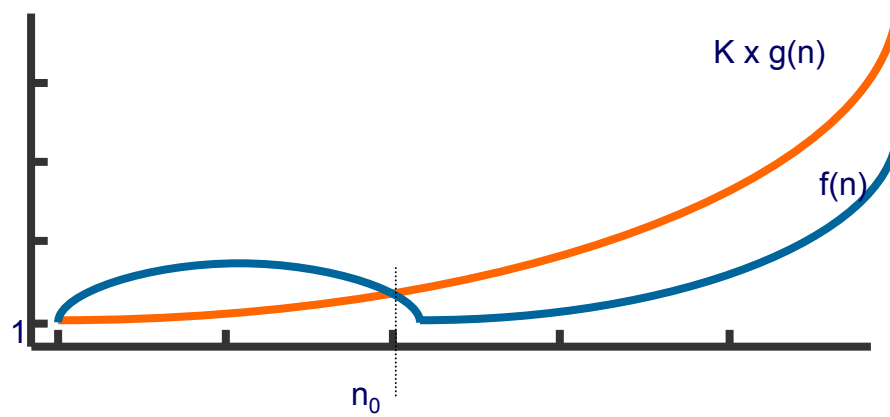
- Wege abbrechen, wenn
 - Kosten höher als bisher bestes Ergebnis
 - Erreichter Knoten schon auf kürzerem Weg erreicht wurde
- Komplexität: $O((n+m) * \log(n))$

O-Notation

- Komplexität wird angegeben als $O(g(n))$
- Das hat folgende Bedeutung
 - Algorithmus A ist $O(g(n))$, gdw. es eine Funktion $f(n)$ gibt, die die Laufzeit von A berechnet, und es gilt:
 - Es gibt positive Konstanten K und n_0 mit:
 - $|f(n)| \leq K \cdot |g(n)|$ für alle $n \geq n_0$
- Beispiel
 - $8n^3 + n^2 + 76$ ist $O(n^3)$
 - Da $8n^3 + n^2 + 76 \leq 85n^3$ für alle $n \geq 1$
- Für g wählt man i.d.R. nur den Grad des Polynoms bzw. den höchsten Exponenten
 - $O(1), O(n), O(n^2), o(n^3), O(2^n), O(\log(n)), \dots$
- Ein Großteil der Informatik widmet sich
 - der Suche nach Algorithmen mit geringerer Komplexität
 - der Suche nach besten Algorithmen für Klassen von Problemen



Unterschiede



1. Stringalgorithmen

- Gegeben ein Template T und ein Pattern P. Finde alle Vorkommen von P in T in möglichst kurzer Zeit
 - Exaktes Matching
- Z-Box **Fundamentaler linearer Algorithmus**
- Boyer-Moore **Schnellster Algorithmus in der Praxis**
- Knuth-Morris-Prath **Elegante Analyse; Erweiterbar**
- Varianten
 - Suche nach mehreren P: Aho-Corasick, Keyword Trees
 - Suche mit Wildcards
 - Suche mit regulären Ausdrücken (= endlichen Automaten)

Beispiel

- Ziel: Statt

T ctgagatcgcgta
P gagatc
gagatc
gagatc
gagatc
gagatc
gatatc
gatatc
gatatc

- ... lieber ...

bbcggbcbbaaggbbbaacabaabgbaac

cababgbaa

bbcggbcbbaaggbbbaacabaabgbaac

cababgbaa

bbcggbcbbaaggbbbaacabaabgbaac

cababgbaa

bbcggbcbbaaggbbbaacabaabgbaac

cababgbaa

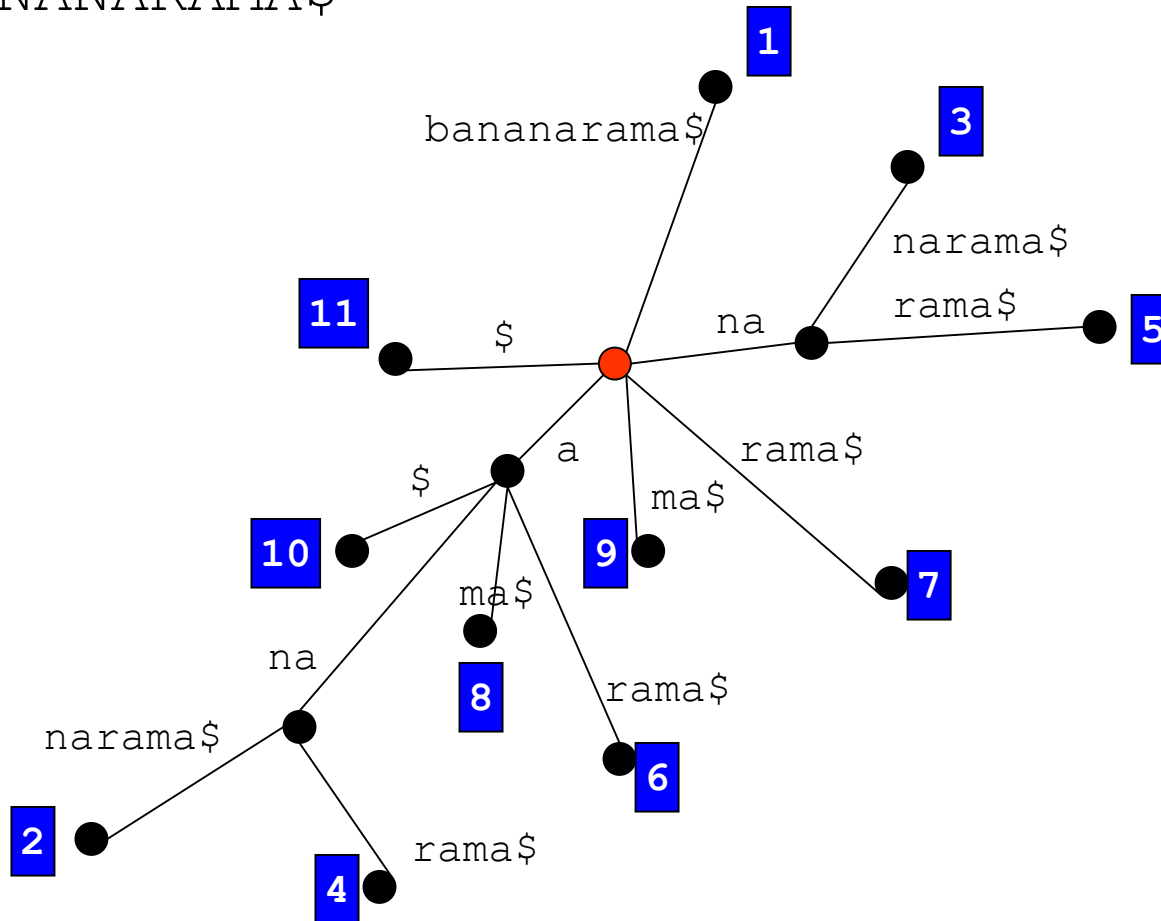
2. Suffixbäume

- Z-Box Algorithmus ist schnell für ein T und ein P
- Es geht noch schneller, wenn man viele T und ein P hat
 - Oder viele P und ein T - Keywordtrees
- Gegeben ein festes T und dauernd wechselnde P. Finde eine Datenstruktur für T, die die Suche nach P's in möglichst kurzer Zeit gestattet
 - Grundlegende Datenstruktur: Suffixbäume
 - Ukkonen's lineare Algorithmus
 - Verschiedene Anwendungen

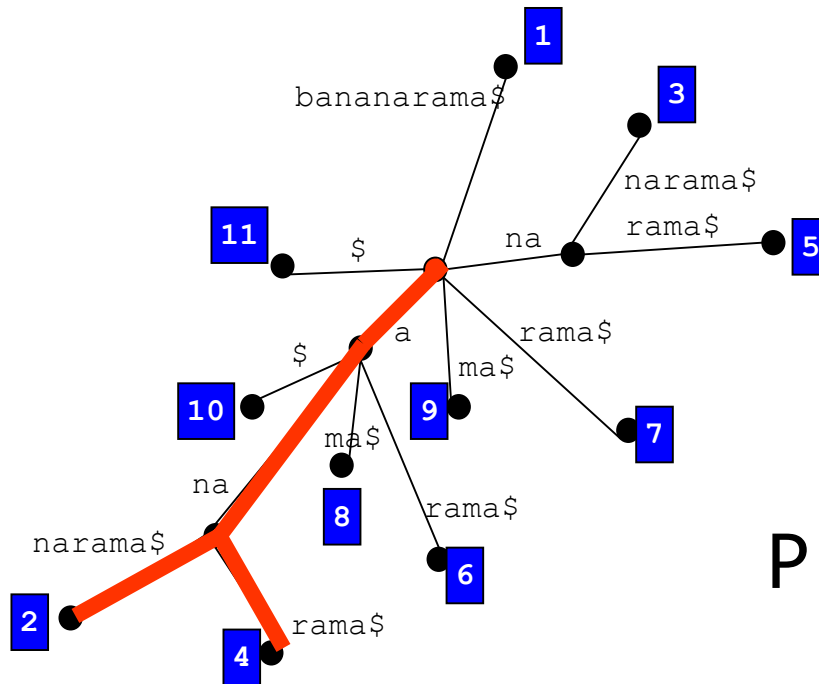
Beispiel

12345678901

- S= BANANARAMA\$



Suche im Suffixbaum



P = „ana“

3. Approximatives Stringmatching

- Gegeben ein Template T und ein Pattern P. Finde alle Vorkommen von Substrings ähnlich zu P in T in möglichst kurzer Zeit
 - Approximatives Matching
- Was heißt überhaupt ähnlich?
 - Edit-Abstand, Alignierung
- Naiver Algorithmus benötigt exponentielle Laufzeit
 - Verbesserung durch dynamische Programmierung

Dynamische Programmierung

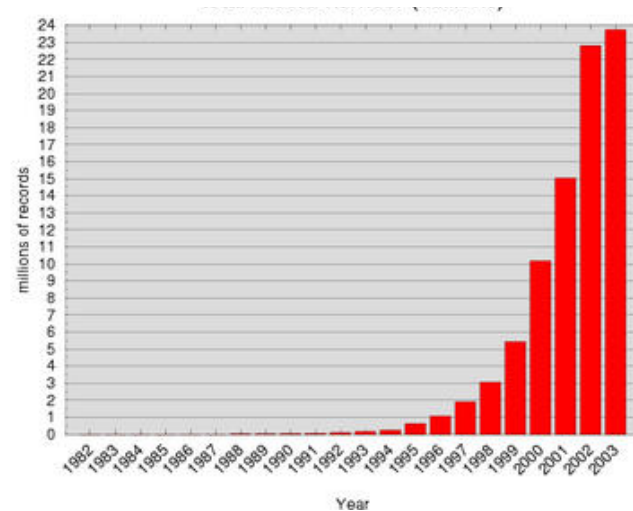
$$d(i, j) = \min \left\{ \begin{array}{l} d(i, j-1) + 1 \\ d(i-1, j) + 1 \\ d(i-1, j-1) + t(i, j) \end{array} \right\}$$

		A	T	G	C	G	G	T
	0	1	2	3	4	5	6	7
A	1	0	1	2	3	4	5	6
T	2	1	0	1	2	3	4	5
G	3	2	1	0	1	2	3	4
G	4	3	2	1	2	1	2	3

- Inkrementeller Aufbau einer Tabelle mit Abstandsmaßen
- Suche nach Wegen in dieser Tabelle
- Quadratische Komplexität

4. BLAST

- Quadratische Laufzeit ist zu teuer
 - Genomanalyse benötigt Suche auf allen bekannten Sequenzen
 - Celera Sequenzierung: All-against-all Vergleich von 28.000.000 Teilsequenzen



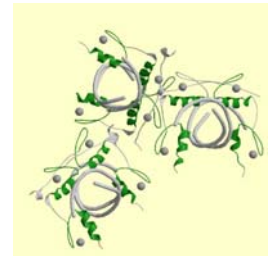
- Grundidee von BLAST
 - Suche nach „Seeds“ mit exakten Matches
 - Verlängerung und Zusammenfügung der Seeds zu Matches
 - Findet nicht alle Hits, aber die meisten „interessanten“
 - „BLAST“ ist fast Synonym für „Bioinformatik“ geworden

5. Multiples Alignment

- Gegeben eine Menge von Strings. Ein Multiple Sequence Alignment (MSA) ist eine Anordnung der Strings mit Spaces untereinander

```
YVCK...LCN...FAFKTKGNLTKHMMSK..AH
YRCPR..ENCD...RTYTTKFNLKSHILT..FH
FRCGY..KCGG...RLYTTAHLKVVHERA...H
YRCE...KCG...KMYKTERCLKVHNLV...H
FSCS...QCD...ESFVORSELELHRQL...H
FPCE...QCD...EKFKTEKQLERHVKT...H
FQCN...QCG...ASFQKGNLLRHIKL...H
FKCH...LCY...RCFGQQTNLDRHLK...H
FRCK...RCR...TRFRQOSELKHKMKT...H
FECN...VCG...SAFRLQLYLSEHOKT...H
MSCKV...CD...RVFYRLDNLRSHLKQ...H
FSCQ...HCH...RAFADRSNLR AHLQT...H
FRCG...YCG...RAFTVKDYLNKHLTT...H
HVCWV..PGCH...RAFSRSDNLN AHVTK...TH
LTC AH...CD...WSFDNVMKLVRRHGV...H
```

Quelle: Pfam, Zinc finger domain



- Hauptziel von MSA's: Finde das „Gemeinsame“ der Sequenzen
 - Funktionen werden oft von sehr kurzen Sequenzstücken bestimmt
 - Welcher Teil eines Proteins bestimmt die Funktion?
 - Wie kann man Proteine in Familien anordnen?

Bioinformatik@Informatik

