

Bioinformatik

Für Biophysiker

Wintersemester 2005 / 2006

Ulf Leser

Wissensmanagement in der
Bioinformatik



Wissensmanagement in der Bioinformatik

- Lehrstuhl seit 10/2002
- Schwerpunkte
 - Algorithmen der Bioinformatik
 - Management molekularbiologischer Daten
 - Datenintegration
 - Text Mining
- Wann Sie mich noch sehen werden
 - Im Nebenfach Bioinformatik
 - In einer Diplomarbeit?

Mein Teil

- Algorithmen der Bioinformatik
 - 6 Termine
 - Letzter Termin: 2.1.2006
- Praktikum: 20.2.2005 – 24.2.2006
 - Mit Silke Trissl
 - Programmierung Dotplots & Smith-Waterman
- Klausur
 - Entweder am Montag, 27.2.2006
 - Alternative: Am Freitag, 24.2.2006, nachmittags?

Informationen

- Zu Vorlesung und Praktikum

- http://www.informatik.hu-berlin.de/wbi/teaching/ws0506/bioinfo_bpi/index.html

The screenshot shows a web page for the Bioinformatics course at Humboldt University Berlin. The page is titled "Bioinformatik" and is for students of the "Theoretische Biophysik und Bioinformatik" program. It lists the lecturers: Prof. Herzel, Prof. Frömmel, Prof. Leser, Dr. Meisel, and Dr. Preissner. The course is held in the winter semester of 2003/2004. The page includes a sidebar with navigation links (Home, Open positions, Staff, Research, Events, Teaching, Networking, Projects) and a main content area with details about the course, including the location (Zentrallabor, Invalidenstr. 42) and a list of topics to be covered in the lectures.

Stand: 28.11.2003

Humboldt Universität zu Berlin Institut für Informatik

Bioinformatik

Für Studenten des Studiengangs "Theoretische Biophysik und Bioinformatik"
Vorlesung im Wintersemester 2003 / 2004
Prof. Herzel, Prof. Frömmel, Prof. Leser, Dr. Meisel, Dr. Preissner

Die zweistündige Vorlesung behandelt grundlegende Fragestellungen moderner Bioinformatik. Sie wird ergänzt um ein Blockpraktikum im Zeitraum vom 23.2.2004 - 27.2.2004 am Institut für Informatik.

Ort / Zeit:

- Montag, 16.00 - 18.00 Uhr, Zentrallabor, Invalidenstr. 42

Die Vorlesung wird in Abschnitten von den verschiedenen Dozenten gehalten.

Literatur zur Vorlesung

- Abschnitt Prof. Leser: Dan Gusfield, "Algorithms on Strings, Trees, and Sequences", Cambridge University Press

Themen im Einzelnen (Folien sind hier teilweise jeweils vor der Vorlesung als PDF verfügbar. Änderungen möglich):

- 27.10.2003, Prof. Frömmel: Einführung / Sequenz->Struktur->Funktion
- 03.11.2003, Dr. Preissner: Funktion und Dynamik von Proteinen
- 10.11.2003, Prof. Herzel: Genome
- 17.11.2003, Prof. Herzel: Bernoulli Sequenzen und statistische Unabhangigkeit
- 24.11.2003, Prof. Herzel: Markov-Modelle und Bayes-Formel
- 01.12.2003, Prof. Leser: [Einleitung, Stringalgorithmen](#)
- 08.12.2003, Prof. Leser: Suffixbaume und Suffixarrays
- 15.12.2003, Prof. Leser: Alignment und dynamisches Programmieren
- Weihnachten / Neujahr
- 05.01.2004, Prof. Leser: Heuristiken zum Alignment; BLAST und FASTA
- 12.01.2004, Prof. Leser: Multiples Sequenzalignment
- 19.01.2004, Dr. Preissner: Strukturvorhersage, Homology Modelling
- 26.01.2004, Dr. Preissner: Strukturalignment und in silico Screening
- 02.02.2004, Prof. Frömmel: Protein-Struktur-Alignment
- 09.02.2003, Dr. Meisel: tba
- 16.02.2003, Dr. Meisel: tba

- Bei Fragen

- [leser\(at\)informatik.hu-berlin.de](mailto:leser(at)informatik.hu-berlin.de), Tel. 2093 3902
- [trissl\(at\)informatik.hu-berlin.de](mailto:trissl(at)informatik.hu-berlin.de), Tel. 2093 5487

Literatur

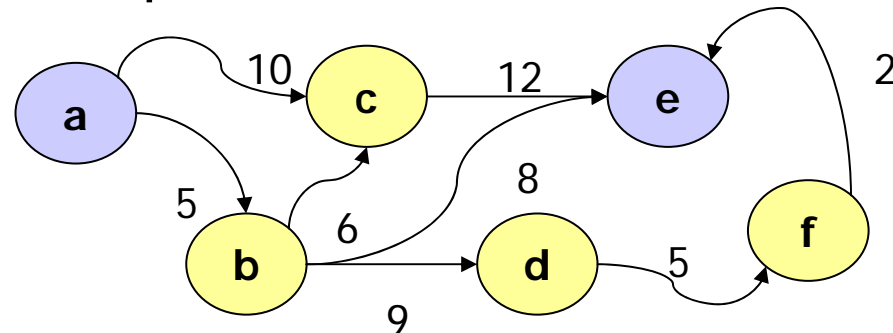
- Primär
 - Dan Gusfield: „Algorithms on Strings, Trees, and Sequences“, Cambridge University Press, 1997 (ca. 60 Euro)
- Weitere
 - Joachim Böckenhauer, Dirk Bongartz: „Algorithmische Grundlagen der Bioinformatik“, Teubner, 2003 (ca. 30 Euro)

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 der Sequenzen, die es zu vergleichen gilt
 - ...
 - [Nicht immer einfach zu bestimmen]
- Bei uns meistens
 - Anzahl und Länge von Zeichenketten

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
- Worst-Case Komplexität: Exponentiell in der Anzahl Kanten

- Dijkstra

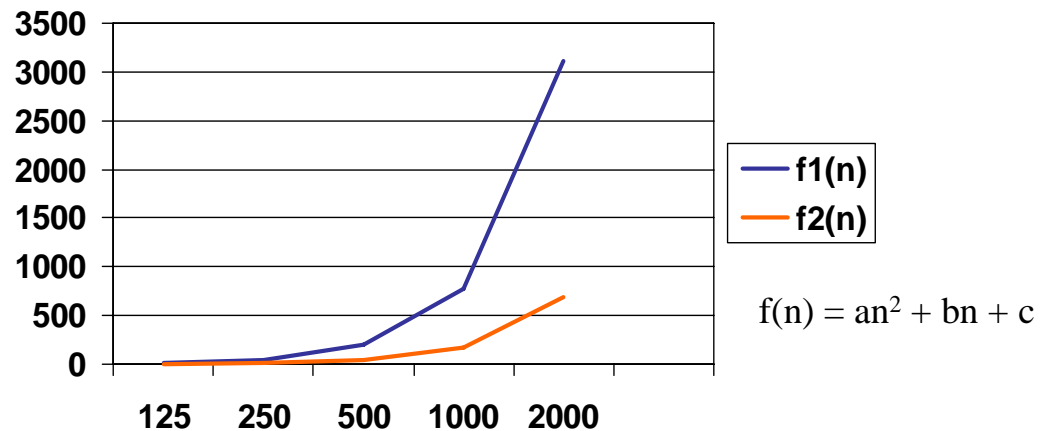
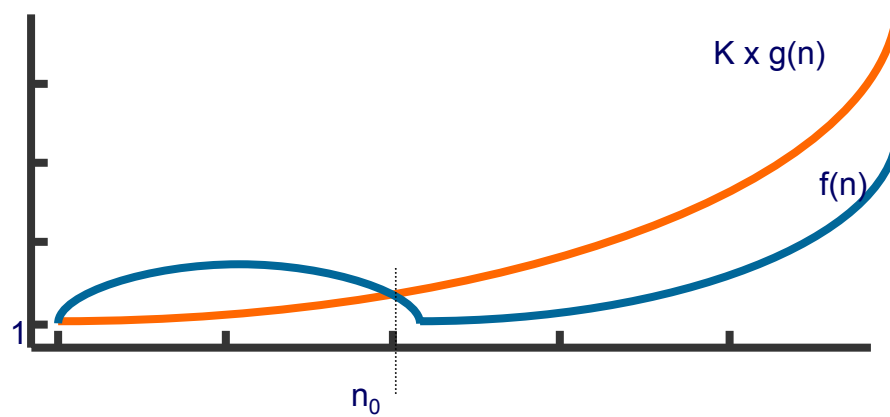
- Wege abbrechen, wenn
 - Kosten höher als bisher bestes Ergebnis
 - Erreichter Knoten schon auf kürzerem Weg erreicht wurde
- Komplexität: $O((n+m) * \log(n))$

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



Take Home Message

- Informatiker lieben O-Notation
- Wichtig für alle Anwendungen am Computer, die
 - mit nicht-trivialen Problemen oder
 - Mit nicht-trivialen Datenmengen zu tun haben
- Kleine O sind gut; n im Exponenten ist schlecht
- Schon $O(n^2)$ ist für grosse (?) Datenmengen zu viel

1. Stringalgorithmen

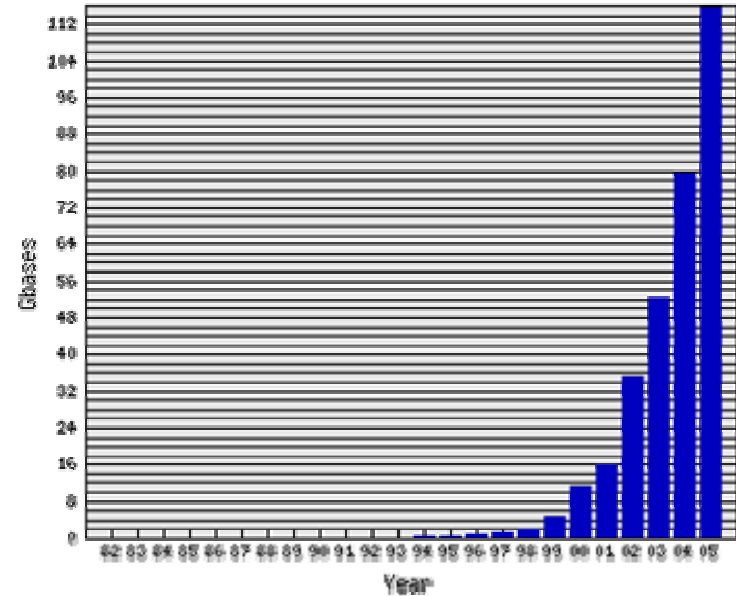
- Gegeben ein Template T und ein Pattern P. Finde alle Vorkommen von P in T in möglichst kurzer Zeit
 - Exaktes Matching
- Z-Box Fundamentaler linearer Algorithmus
- Boyer-Moore Schnellster Algorithmus in der Praxis
- Varianten
 - Suche nach mehreren P
 - Suche mit regulären Ausdrücken (= endlichen Automaten)

2. Approximatives Stringmatching

- Approximatives Matching
 - Gegeben ein Template T und ein Pattern P. Finde alle Substrings in T die „ähnlich“ zu P sind – in möglichst kurzer Zeit
- Variante
 - Gegeben ein Template T und ein Pattern P. Finde alle Substrings in T die „ähnlich“ zu einem Substring in P sind
- Beides sind **fundamentale Fragestellung** der Bioinformatik
 - Ähnlich Sequenz – ähnliche Struktur – ähnliche Funktion
- Was heißt überhaupt **ähnlich**?
 - Edit-Abstand, Alignierung
- Naiver Algorithmus benötigt exponentielle Laufzeit
 - Verbesserung durch tabellarische Berechnung

3. Heuristiken zur approx. Suche

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



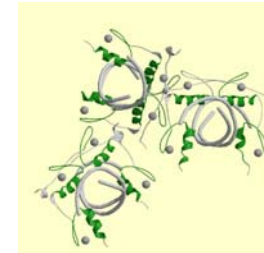
- Grundidee von BLAST
 - In jedem guten approximativen Match steckt ein exakter Match
 - Mischung von exaktem und approximativen Match
 - Findet nicht alle Hits, aber die „meisten“

4. Multiples Alignment

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

```
YVCK...LCN...FAFKTKGNLTKHMKSK.AH
YRCPR.ENCD...RTYTTKFNLSHILT..FH
FRCGY.KGCG...RLYTTAHLKVERA...H
YRCE...KCG...KMYKTERCLKVHNLV...H
FSCS...QCD...ESFVQRSELELHRQL...H
FPCE...QCD...EKFKTEKQLERHVKT...H
FQCN...QCG...ASFTQKGNLLRHIKL...H
FKCH...LCY...RCFQQTNLDRHLKK...H
FRCK...RCR...TRFRQOSELKHHMT...H
FECN...VCG...SAFRLQLYLSEHQKT...H
MSCKV...CD...RVFYRLDNLRSHLKQ...H
FSCQ...HCH...RAFADRSNLRHLQT...H
FRCG...YCG...RAFTVKDYLNKHLTT...H
HVCWV.PGCH...RAFSDNLRNAHYTK...TH
LTCAH...CD...WSFDNVMKLVRRGV...H
```

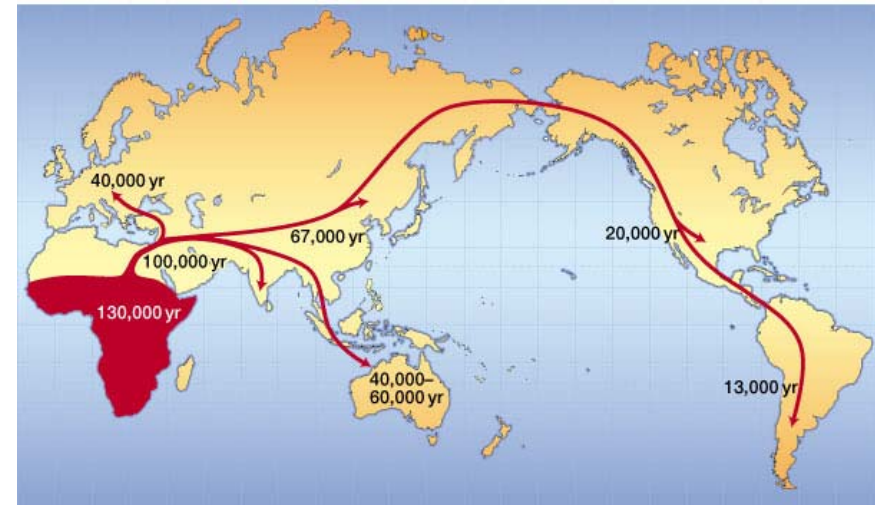
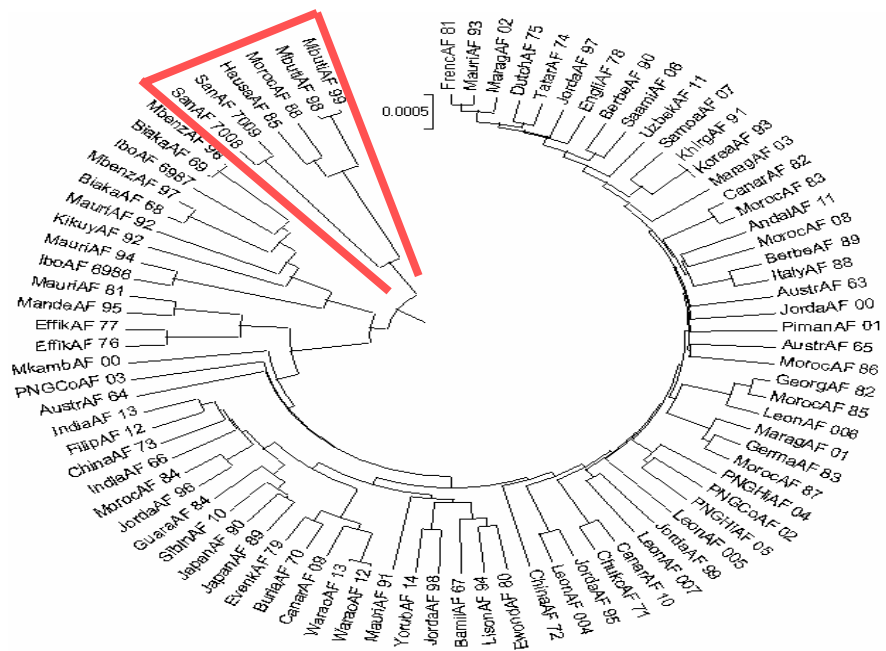
Quelle: Pfam, Zinc finger domain



- Ziel: 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?

5. Phylogenetische Algorithmen

- Sequenzierung der mitochondrialen DNA (16 KB) von 86 geographisch verteilt lebenden Personen
- Ergebnis: Mitochondriale DNA scheint nach einer molekularen Uhr abzulaufen; Divergenz ist ca. $1,7E-8$ pro Base und Jahr

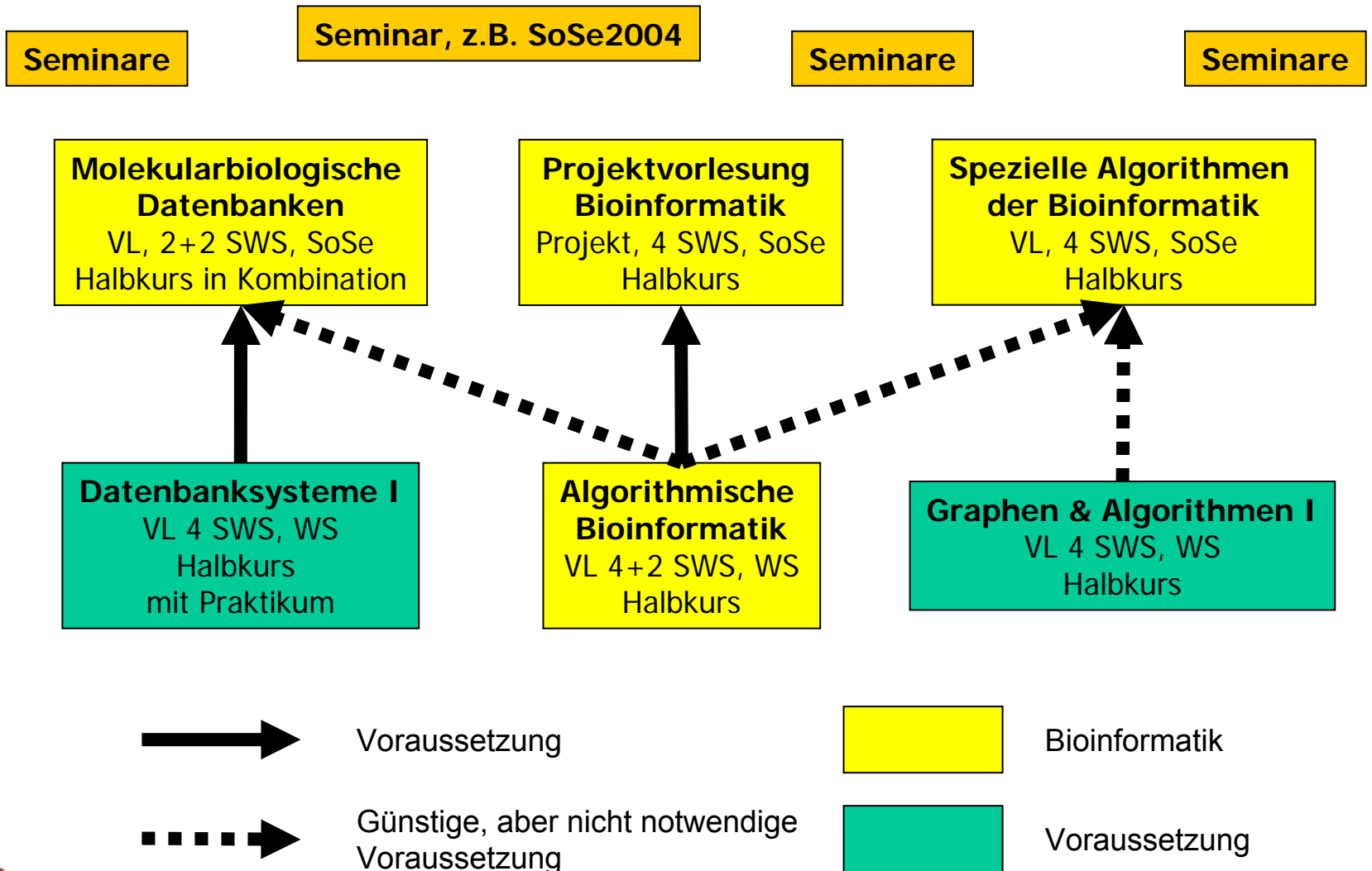


Quelle:
Ingman, M., Kaessmann, H., Pääbo, S. & Gyllensten, U. (2000)
Nature 408: 708-713

Quelle:
<http://www.genpat.uu.se/mtDB/sequences.html>

Methode: UPGMA

Bioinformatik@Informatik



Feedback vom letzten Mal

- Behandelte Stoffmenge angemessen (1-5) 2,9+-0,08
- Stoff zu leicht/zu schwer (1-5): 3+-0.05
- Vorbereitung und Kenntnisse des Dozenten (1-5): 1+-0
- Geschwindigkeit des Vortrags (1-5): 2,4+-0.6
Anregung zum Mitdenken: 2,2+-1.1
 - Langsamer Reden; Stoff entrümpeln; Mehr Tafelbild
 - [Fragen Sie nach](#)
- Weiteres
 - Plus/Minus Wiederholung am Anfang
 - Vorlesungsbegleitende Übung
 - Mehr Beispiele