Algorithmen auf Sequenzen

Organisatorisches

Zeit Mo und Fr, 10-12 (Vorlesung 2-3h/Woche, Übung 1-2h/Woche)
Ort OH16, E07
SpG 4 (Algorithmen und Komplexität)

Prüfungsleistungen (über 4 SWS = 6 ECTS credits)

- Leistungsnachweis durch Bearbeiten der Übungsaufgaben (2 / Woche)
- mündliche Fachprüfung über Vorlesung & Übungen

Anmeldung zu den Übungen

Ausgabe der Aufgaben: Montag. Abgabe der Aufgaben: Freitag.
Bitte jetzt den Anmeldezettel ausfüllen!

Weitere Informationen

http://ls11-www.cs.tu-dortmund.de/people/rahmann/teaching/ss2008/AlgorithmenAufSequenzen
Sprechstunde: Mo 16-17 in OH14, R214

Literatur

Gonzalo Navarro, Mathieu Raffinot
Flexible Pattern Matching in Strings
Cambridge University Press
ISBN: 0-521-03993-2
(möglicherweise vergriffen, Neudruck im Oktober)

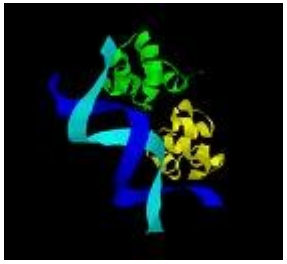
Dan Gusfield
Algorithms on Strings, Trees and Sequences
Cambridge University Press
ISBN: 0-521-58519-8

David Sankoff und Joseph P. Kruskal
Time Warps, String Edits, and Macromolecules
University of Chicago Press
ISBN: 1-575-86217-4
(Taschenbuchausgabe von 2000 des 1983 erschienenen Originals)

weitere Originalliteratur im Lauf der Vorlesung

Warum Untersuchung von Sequenzen?

Relevant für viele Anwendungen, z.B.



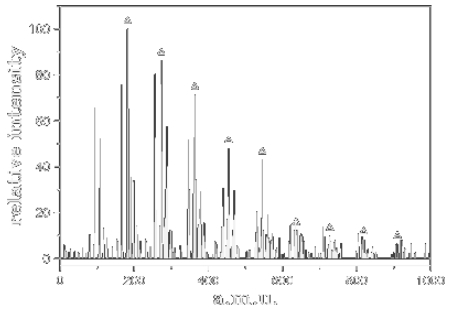
• **Biosequenzen** (DNA, RNA, Proteine)
(aber: Genome sind komplexer als „nur“ eine DNA-Sequenz)

• **Texte** (Literatur, wissenschaftliche Texte)
Die Kunst hinter guter Literatur und hinter guten wissenschaftlichen Arbeiten besteht darin, schwierige, komplex zusammenhängende Sachverhalte in eine logische Abfolge von einzelnen Sätzen zu bringen.

• **Programme**

• **Dateien, Datenströme**
Komplexe Datenstrukturen werden serialisiert, um sie persistent zu machen.

• **Zeitreihen, Spektren**
Audiosignale, Massenspektren, ...



Sequenzen sind serialisierte Information.

Sequenzen sind „eindimensional“
= einfach darzustellen und zu analysieren!
(Schwieriger sind z.B. Probleme auf Graphen)

Was ist eine Sequenz?

Sequenz (sequence) := Folge, im mathematischen Sinn

Bausteine

- Alphabet A (endlich oder unendlich)
- Linear geordnete Indexmenge I (endlich oder unendlich)

Eine durch I indizierte Folge s über A heißt Sequenz.

Schreibweisen

Funktionsschreibweise: $s: I \rightarrow A$

Tupelschreibweise: $s \in A^I$

Aufzählende Angabe von endlichem s :
Aneinanderreihung (Konkatenation) der Elemente, durch I gegebene Ordnung

Indizierung häufig durch Subskripte: $s = (s_i)_{i \in I}$

Beispiel s = AGGTC ist eine Sequenz mit
A = {A,C,G,T} (DNA-Alphabet),
I = {1,2,3,4,5} in der üblichen Ordnung,
Z.B. bildet s die 3 auf G ab, s_3=G

Beispiele für Sequenztypen

In dieser Vorlesung (und in konkreten Anwendungen fast immer)
- Endliches Alphabet A
- Indexmenge $I = \{1,2,...,N\}$ für ein endliches N

Beispiele

DNA-Sequenz	$A = \{A,C,G,T\}$
Protein-Sequenz	A = 20 Standard-Aminosäuren
C-Programme	A = ASCII-Zeichen (7-bit)
Java-Programme	A = Unicode-Zeichen
Audiosignal (16-bit samples)	$A = \{0,...2^{16}-1\}$
Massenspektrum	$A = [0,1]$ (unendlich) oder Double

Darstellung im Computer (Java)

String (wenn $A \subseteq$ Unicode) oder A[] oder ArrayList<A> oder Map<I,A>

Probleme auf Sequenzen (Auswahl)

- Sequenzvergleich: Quantifikation von Ähnlichkeit / Unterschieden
- Anwendung: Revisionskontrolle, Verfolgen von Änderungen (z.B. subversion)

- Quantifikation der „Komplexität“ einer Sequenz (Datenkompression)

- Suche nach Mustern in Sequenzen (z.B. Textverarbeitung: find-replace)
- Approximative Mustersuche (Meier vs. Mayer)
- Suche nach einem Wort in einem Wörterbuch, das dem gegebenem am ähnlichsten ist

- Entdecken von wiederholten Teilsequenzen (Genomanalyse, Kompression)

- Entwicklung von Codes zur fehlertoleranten Übertragung von sequenzieller Information

Brief Biography

Sven Rahmann became a professor for Bioinformatics for High-Throughput Technologies at the Chair of Algorithm Engineering, Computer Science Department, TU Dortmund in October 2007.

Between August and December 2007, he spent four months at HHMI Janelia Farm Research Campus as a visiting scientist in Gene Myers' lab.

Sven was an independent Junior Research Group leader of the Computational Methods for Emerging Technologies (COMET) group, formerly known as the *Algorithms and Statistics for Systems Biology* group, at Bielefeld University from March 2004 till September 2007. The group closely collaborated with the Genome Informatics group at the Faculty of Technology (Technische Fakultät) at Bielefeld University. During the same time, **Sven** was also a member of the Institute of Bioinformatics (IfB) at the Center for Biotechnology (CeBiTec), and part of the faculty of the Graduate School in Bioinformatics and Genome Research at Bielefeld University.

From January 2001 till February 2004, **Sven** wrote his doctoral thesis on oligonucleotide design for microarrays in the Computational Molecular Biology group at the Max Planck Institute for Molecular Genetics in Berlin.

Between 1994 and the end of 2000, **Sven** studied mathematics and computer science with a focus on statistical methods in bioinformatics at the universities of Göttingen, UC Santa Cruz, and Heidelberg. During this time, he worked as a freelance programmer for the Gothaer insurance company and as a student assistant in the Theoretical Bionformatics group of the German National Cancer Research Center (DKFZ), where he wrote his Diploma (M.Sc.) thesis on word statistics in random texts.

Grundlegende Definitionen

- Alphabet *A*: endliche Menge
- A^n: Menge der Sequenzen der Länge *n* über *A*
- A^*: Menge aller endlichen Sequenzen über *A*
- A^+: Menge aller nicht leeren endlichen Sequenzen
- ϵ: die leere Sequenz, das einzige Element in A^0

$$A^* := \cup_{n \geq 0} A^n$$
$$A^+ := \cup_{n \geq 1} A^n$$

Sequenz = String = Wort
Aber: Teilstring (substring) != Teilsequenz (subsequence)

Sei $s = (s_1, s_2, ..., s_n)$ ein String.
Dann ist $s[i..j] := (s_i, s_{i+1}, ..., s_j)$ für $i \leq j$ ein Teilstring. Für $j < i$ sei $s[i..j] := \epsilon$.
Ein Teilstring der Länge *q* heißt auch *q*-gram (Griechisch gramma = Buchstabe).

Eine Teilsequenz wird durch eine streng monoton steigende Folge von Indizes gegeben.
Jeder Teilstring ist auch eine Teilsequenz, aber nicht umgekehrt.

Ein Präfix ist ein Teilstring, der leer ist oder an Position 1 beginnt.
Ein Suffix ist ein Teilstring, der leer ist oder an der letzten Position endet.