Algorithmische Problemlösetechniken

Roland Wittler
roland ät cebitec.uni-bielefeld.de, U10-145

Wintersemester 2010/2011
28.2.–18.3.2011
im GZI: V2-221 (und V2-234)

1 Inhalt

In diesem Blockseminar soll eigenständiges Arbeiten an bisher ungelösten Problemen (z.B. Entscheidungs- oder Optimierungsprobleme) geübt werden. Zu Beginn werden verschiedene Probleme vorgestellt, für deren Lösung im Verlauf der Veranstaltung Algorithmen überlegt werden sollen.

Es kommt nicht darauf an, ein Problem vollständig oder effizient zu lösen; teilweise ist nicht einmal bekannt, ob effiziente Lösungen existieren. Vielmehr ist es wichtig das Problem vollständig zu verstehen, vorhandene Ansätze zu recherchieren, und schließlich verschiedene eigene Ansätze zu entwickeln, implementieren, evaluieren und dokumentieren.

Es soll in Kleingruppen an je einem Problem gearbeitet werden: theoretisch (Recherche, Verständnis, Analyse) als auch praktisch (Implementation, Dokumentation, Presentation). Der Fokus liegt dabei auf der Implementation, dem praktischen Ausprobieren verschiedener Lösungsansätze.

2 Ablauf

Montag, 28.2., 9 c.t. Einführung, Themenvergabe, Einarbeitung

Dienstag, 1.3. bis Donnerstag, 10.3. Anwesenheitspflicht: Bearbeitung der Problemstellungen (in Kleingruppen oder auch alleine).

Freitag, 11.3. bis Donnerstag, 17.3. Freie Arbeit: Arbeit am Problem, Vorbereiten eines Vortrags (20-25 Minuten), Schreiben einer Ausarbeitung (ca. 3 Seiten Reintext). GZI steht zur Verfügung.

Freitag, 18.3. Abgabe der Ausarbeitungen, Präsentation der Ergebnisse.

3 Problemdefinitionen

Im folgenden werden kurz einige Problemstellungen erläutert, die sich für die Bearbeitung in dieser Veranstaltung eignen. Die Erklärungen sind hier recht oberflächlich dargestellt und werden zu Beginn des Seminars ausführlicher erläutert werden.

Weitere Themenvorschläge sind willkommen!

3.1 Deletionen im Genom

Mithilfe von *paired-end mapping* kann man Deletionen in einem neu sequenzierten Genom vorhersagen, ohne die vollständige DNA-Sequenziert zu bestimmen. Die Deletionen können jedoch nur grob bestimmt werden: "In diesem Abschnitt des Genoms wurden zwischen min und max Basen gelöscht." Aufgrund fehlerhafter Daten und vereinfachender Annahmen, sind diese Vorhersagen nicht immer korrekt, teilweise sogar widersprüchlich. Frage: Kann es überhaupt eine Menge D von deletierten Basen im Genom geben, die allen Vorhergesagten Deletionen entsprechen?

[CDP] Consistency of deletion predictions.

Given: Set M of intervals over $\mathbb{N}$, functions $min : M \longrightarrow \mathbb{N}$ and $max : M \longrightarrow \mathbb{N}$ assigning a minimum and a maximum value to each interval.
Question: Is there a set $D \subseteq \mathbb{N}$ such that for all $I \in M$:
$min(I) \leq |I \cap D| \leq max(I)$?
Complexity: open

Nehmen wir an, wir hätten *minimal conflicting sets* von Deletionen identifiziert. Dies sind Mengen von Deletionen, die widersprüchlich sind; *minimal* meint, sobald eine Deletion aus der Menge herausgenommen würde, wäre der Widerspruch aufgehoben, die Menge also konsistent.

Das menschliche Genom in *diploid*: Jedes Chromosom liegt in zwei Kopien vor. Eine Deletion kann auch *heterogen* sein, d.h. nur in einer der zwei Kopien fehlen die Basen. Deletionen aus einem *minimal conflicting set* könnten also auf zwei Kopien aufgeteilt sein, so dass jede "Hälte" für sich konsistent ist.

Dies wird mit einem Hypergraphen modelliert. Das ist ein Graph, bei dem Kanten aus mehr als zwei Knoten bestehen können. Jede Deletion entspricht einem Knoten, jede (Hyper-)Kante einem *minimal conflicting set* von Deletionen. Nun versuchen wir jede Deletion einer der beiden Chromosomen-Kopien zuzuordnen (wir färben jeden Knoten im Graphen mit einer von zwei Farben ein), so dass jeder Konflikt durch Verteilen der Deletionen auf zwei Kopien gebrochen wird (jede Kante bekommt beide Farben ab).

[H2C] Hypergraph two-colorability.

Given: Hypergraph (V, E).
Question: Can the elements of V be colored by two colors such that each edge $e \in E$ contains at least one vertex of each color?
Complexity: NP-complete, even if $|e| < 4$ for all $e \in E$.

3.2 Ordnen von Contigs

Beim Sequenzieren eines Genoms kann meist nicht auf Anhieb die gesamte Sequenz bestimmt werden. Stattdessen erhält man nur Teilstücke des Genoms: *Contigs* (contiguous sequences). Man kennt nicht deren Reihenfolge und Abstand im Genom, einige Contigs könnten sogar mehrfach im Genom vorkommen. Durch Vergleiche der Contigs mit Genomen von eng verwandten Spezies kann man jedoch Hinweise auf die richtige Reihenfolge der Contigs erhalten.

Wir modellieren einen Contig-Graphen, in dem jeder Knoten einem Contig entspricht. Für jeden Contig/Knoten ist angegeben, wie oft er im Genom vorkommen kann. Eine Kante deutet darauf hin, dass zwei Contigs im Genom nebeneinander liegen. (Ein Gewicht der Kante kann zusätzlich angeben, wie (un)wahrscheinlich die Nachbarschaft ist.) Ein Weg durch den Graphen entspricht einer Anordnung der Contigs. Jeder Contig sollte "passend oft" auf diesem Weg benutzt werden.

[mHAM] Hamilton with multiplicities.

Given: Graph $G = (V, E)$, function $m : V \longrightarrow \mathbb{N}$ assigning a maximum multiplicity to each vertex $v \in V$.
Question: Is there a tour through G that visits each vertex $v \in V$ $m(v)$ times and each edge $e \in E$ at most once?
Complexity: NP-complete.

[mTSP] TSP with multiplicities.

Given: Edge-weighted complete graph $G = (V, E)$, function $m : V \longrightarrow \mathbb{N}$ assigning a maximum multiplicity to each vertex $v \in V$.
Question: Find a shortest tour through G that visits each vertex $v \in V$ $m(v)$ times and each edge $e \in E$ at most once.
Complexity: NP-complete.

3.3 Rekonstruktion von Genclustern

Unter einem *Gencluster* versteht man eine Gruppe von Genen, die in den Genomen von verschiedenen Spezies nebeneinander angeordnet sind. Es gibt Verfahren um ausgehend von den Genreihenfolgen von heute lebenden Spezies Gencluster von den (ausgestorbenen) Vorfahrspezies zu rekonstruieren um davon Informationen über die Evolution von Genordnungen, Genclustern und einzelnen Genen zu erhalten. Für eine gegebene Menge von rekonstruierten Genclustern stellt sich die Frage: "Kann es überhaupt eine Genreihenfolge geben, die all die rekonstruierten Gencluster enthält?"

Diese Fragestellung kann man als Binärmatrix modellieren. Jede Spalte entspricht einem Gen und jede Zeile einem Gencluster. Alle Gene die in einem Gencluster enthalten sind werden in der entsprechenden Zeile der Matrix durch eine 1 markiert. Alle andern Einträge sind 0. Eine Genreihenfolge, die alle Gencluster enthält, enstpricht nun einer Umordnung der Spalten, so dass in jeder Reihe alle Einsen in einem Block nebeneinander liegen. Dies ist das *Consecutive Ones Problem*, für welches schon vor vielen Jahren polynomielle Algorithmen gefunden wurden. Es gibt jedoch auch zahlreiche Varianten dieses klassischen Problems, teilweise NP-schwer, teilweise ist die Komplexität noch offen.

Das Genom vieler Organismen besteht nicht aus einem oder mehreren linearen Chromosomen, sondern aus zirkulären Chromosomen. Desweiteren können einige Gene auch mehrfach im Genom vorkommen.

[mC1Pc] mC1P on circular sequences.

Given: Binary matrix M, function m assigning a maximum multiplicity to each column c of M.
Question: Is there a set of circular sequences of the columns of M such that each column c is used at most $m(c)$ times and the entries 1 in each row are consecutive in at least one of the circular sequences?
Complexity: open.

Kleine Fehler in den Daten könnten toleriert werden.

[k-C1P] C1P allowing k blocks.

Given: Binary matrix M.
Question: Is there an ordering of the columns of M such that the entries 1 in each row appear consecutively in at most k blocks?
Complexity: NP-complete.

[(2,1)−C1P] C1P allowing two blocks of distance one.

Given: Binary matrix M.
Question: Is there an ordering of the columns of M such that the entries 1 in each row appear consecutively in at most two blocks separated by at most one entry 0?
Complexity: open.

Für einige Gene ist nicht sicher, ob sie in einem Cluster enthalten sind oder nicht.

[x−C1P] C1P with wild cards.

Given: Matrix M over $\{0, 1, x\}$.
Question: Is there an ordering of the columns of M such that in any row, no entries 1 are separated by an entry 0.
Complexity: open.

Bei der Rekonstruktion kann auch die Tatsache, dass Gene an den Extremitäten eines Chromosoms, den Telomeren, liegen berücksichtigt werden. Wir nehmen an, dass jedes Gen nur einmal vorkommt, aber verwenden ein "Pseudogen" als Repräsentant eines Telomers. Da nicht alle Telomere unterscheidbar sind, werden teilweise mehrere Vorkommen eines Telomers erlaubt. Jeder Gencluster enthält nur ein Telomer.

[C1Pt] C1P with telomeres.

Given: Binary matrix M, function m assigning a maximum multiplicity to each column c of M, each row in M contains at most one column c with $m(c) > 1$.
Question: Is there a sequence of the columns of M containing the entries 1 in each row consecutively and each column c is used at most $m(c)$ times?
Complexity: open.