technische universität dortmund

UNIVERSITÄT DUISBURG ESSEN
Offen im Denken

Bioinformatik, Fakultät für Informatik
Genominformatik, Medizinische Fakultät

Prof. Dr. Sven Rahmann
Wintersemester 2019/20

# Algorithmische Bioinformatik
# Übungsblatt 9

**Ausgabe:** 14. Januar 2020 · **Besprechung:** 21. Januar 2020

**Aufgabe 9.1** Für die Anzahl $U_n$ ungewurzelter ungerichteter Binärbäume hatten wir die Formel

$$U_n = (2n-5)\,!!$$

bewiesen, wobei die ungerade Fakultät als $k\,!! := 1 \cdot 3 \cdot 5 \cdot \cdots \cdot k$ für ungerade $k$ definiert ist, also $U_n = 1 \cdot 3 \cdot \cdots \cdot (2n-5)$ für $n \geq 3$. Beweise mit der "normalen" Fakultätsfunktion

$$U_n = \frac{(2n-5)!}{(n-3)! \cdot 2^{n-3}}\,.$$

**Aufgabe 9.2** Schreibe ein Programm, das alle $W_n$ gewurzelten Bäume auf $n$ OTUs (operational taxonomic units) $A, B, C, \ldots$ aufzählt und im Newick-Format ausgibt. Achtung: $((A,B),C)$, $((B,A),C)$, $(C,(A,B))$, $(C,(B,A))$ sind derselbe Baum. Es soll nur eine (kanonische) Variante ausgegeben werden! Wie sieht die Ausgabe deines Programms für $n = 5$ aus?

Zusatz: Gib statt des Newick-Formats eine svg-Datei (andere Formate sind auch erlaubt) aus, in der alle Baumtopologien gezeichnet sind.

**Aufgabe 9.3** Sei $M$ eine binäre $n \times m$ Merkmalsausprägungsmatrix mit $n$ Zeilen (OTUs) und $m$ Spalten (Merkmalen). Man erhält eine $3 \times 2$-Untermatrix, indem man drei verschiedene Zeilen $\{i, i', i''\}$ und zwei verschiedene Spalten $\{j, j'\}$ auswählt.

Zeige: Es existiert genau dann eine Perfekte Phylogenie mit Null-Ausprägungen in der Wurzel, wenn es in $M$ keine $3 \times 2$-Untermatrix der Form $\begin{pmatrix} M_{ij} & M_{ij'} \\ M_{i'j} & M_{i'j'} \\ M_{i''j} & M_{i''j'} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix}$ gibt.

**Aufgabe 9.4** Gegeben ist die folgende Matrix $M$ auf fünf OTUs ($A$–$E$) mit fünf Merkmalen (1–5). Existiert eine Perfekte Phylogenie? Wenn ja, konstruiere sie (irgendwie).

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $A$ | 1 | 1 | 0 | 0 | 0 |
| $B$ | 0 | 0 | 1 | 0 | 0 |
| $C$ | 1 | 1 | 0 | 0 | 1 |
| $D$ | 0 | 0 | 1 | 1 | 0 |
| $E$ | 0 | 1 | 0 | 0 | 0 |

**Aufgabe 9.5** (***) Wir wollen einen effizienten Algorithmus in Laufzeit $O(nm)$ bei $n$ OTUs und $m$ Merkmalen entwickeln, der entscheidet, ob eine PP existiert und ggf. eine konstruiert. Vollziehe den Algorithmus an der Matrix aus der vorigen Aufgabe nach, sowie an einer Matrix, zu der *keine* PP existiert.

Schritt 1: Wir sortieren die Merkmale (Spalten) der Matrix $M$ wie folgt um: Wir interpretieren jede Spalte als Binärzahl (höhchstwertiges Bit oben) und sortieren nach absteigendem Zahlenwert. Beschreibe, wie man dies in $O(nm)$ Zeit realisieren kann. Sei $M'$ die resultierende Matrix; wir benennen jetzt darin die Spalten wieder mit $1, 2, \ldots$. Argumentiere: Es gibt genau dann eine PP für $M'$, wenn es eine für $M$ gibt.

Schritt 2: Für jede Zeile (OTU) $i$ von $M'$, erzeuge die Sequenz $s_i$ der Merkmalsnamen $j$ in aufsteigender Reihenfolge mit $M'_{ij} = 1$, gefolgt von einem Endmarker \$. Zeige: Hat $M'$ eine PP, dann gilt, dass je zwei Sequenzen $s_i$, $s_{i'}$ die folgende Eigenschaft haben: Bis zu einer Position sind sie identisch, danach haben sie keine Zeichen mehr gemeinsam. (Der Fall, dass es beispielsweise 134\$ und 14\$ gibt, kann also bei einer PP nicht auftreten!) Dieser Schritt kann offensichtlich in $O(nm)$ Zeit durchgeführt werden.

Schritt 3: Konstruiere den Trie (Präfixbaum, engl. trie, prefix tree oder keyword tree) der Sequenzen; beschrifte dabei die Kanten mit den entsprechenden Merkmalsnamen und lasse die Beschriftung \$ an den Blattkanten weg. Warum kann dieser Schritt in $O(nm)$ durchgeführt werden? Zeige: Wenn es eine PP gibt für $M'$ gibt, dann ist dieser Baum eine solche.

Schritt 4 (optional): Ersetze an den Kanten Merkmalsnamen aus $M'$ durch die von $M$ durch Anwenden der inversen Permutation. Dies geschieht in Zeit $O(m)$.