

Übungen zur Vorlesung Algorithmische Bioinformatik

Freie Universität Berlin, WS 2010/11
Roland Krause · Martin Vingron · Matthias Winkelmann

Blatt 1 · Ausgabe am 18.10.2010
Abgabe am 24.10.2010 vor Beginn der Vorlesung

Aufgabe 1 (Sequenzstatistik). Sie suchen eine DNA-Sequenz der Länge 5 in einer Datenbank von 10.000 zufälligen Sequenzen mit Länge 7. Bestimmen Sie den Erwartungswert (e-value) für die Anzahl exakter Treffer. [Theorie 2]

Aufgabe 2 (Reguläre Ausdrücke in Python). Laden Sie die Proteine *Saccharomyces cerevisiae*, die Sie in der Saccharomyces Genome Database finden ¹. Zur Vereinfachung der Analyse ersetzen Sie die Kopfzeile jedes Eintrags durch den Gen-Namen. Ersetzen Sie also

```
>YFL039C ACT1 SGDID:S000001855, Chr VI from 54695-54686,54377-53260, reverse  
complement, Verified ORF, "Actin, structural protein involved in  
cell polarization, endocytosis, and other cytoskeletal functions"
```

durch

```
>ACT1
```

Nutzen Sie dazu einen regulären Ausdruck und geben Sie ihn an. Das Programm soll so gestaltet sein, dass auf anderen Rechnern, auf denen Python3 installiert ist, ohne Modifikationen laufen kann. Die Ein- und Ausgabedateien sollen als Kommandozeilen-Parameter spezifiziert werden. [Anwendung 30]

Aufgabe 3 (Bestimmung einer Proteinfamilie). In einem vollständig sequenzierten Organismen wollen Sie die Proteine einer Familie bestimmen. Nutzen Sie die Proteine der Hefe, die Sie in der vorigen Aufgabe vorbereitet haben. Erstellen Sie daraus eine durchsuchbare Datenbank mit BLAST (`formatdb` oder `makeblastdb`) und suchen Sie die Homologen von FAL1. Wie viele Proteinsequenzen erhalten Sie?

Schreiben Sie ein Programm, um die BLAST-Treffer auszulesen und erstellen Sie eine neue Datei im FASTA-Format mit den dazugehörigen Sequenzen. Alignieren Sie die Proteinsequenzen mit Muscle² und erstellen mit `hmmbuild` aus dem HMMER-Paket³ ein Hidden Markov Model. Suchen Sie mit diesem Modell mittels `hmmsearch` in den Hefe-Proteinen nach weiteren Homologen.

Vergleichen Sie, wieviele Proteine sie mit BLAST und wie viele Sie mit der Profilsuche finden.

[Anwendung 120]

¹http://downloads.yeastgenome.org/sequence/genomic_sequence/orf_protein/orf_trans.fasta.gz

²<http://www.drive5.com/muscle/>

³<http://hmmer.janelia.org/>

Aufgabe 4 (Dynamische Programmierung). Betrachten Sie die folgende Matrix, die ein Alignment mittels dynamischer Programmierung entsteht.

1. Erläutern Sie das Prinzip der dynamischen Programmierung (200 - 300 Worte).
2. Welcher Algorithmus wurde verwendet, um die Matrix zu füllen?
3. Welche Scorefunktion (Matches, Mismatches, Gaps) liegt vor?
4. Füllen Sie die restlichen Felder der Matrix aus.
5. Geben Sie die optimalen lokalen Alignments der beiden Sequenzen an.

		A	A	G	C	C	T	T	G	C	A	A
	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	↖	2 ← 1	0	0	0	↖	2 ← 1	0	0
A	↖	2	↖	2 ← 1	↑ ↖	1	0	0	↑ ↖	1	↖	3 ← 2
C		↑ ↖	1	↑ ↖	1	3	3 ← 2	← 1	0	3 ← 2	2	2
G			↖	3 ← 2	↑ ↖	2	2 ← 1	↖	3 ← 2	2 ← 1	↖	↑
C			↑ ↖	2	5 ← 4	← 3	← 2	2	5 ← 4	← 3	↖	↑
A	↖	2	↖	2 ← 1	↑ ↖	4	4 ← 3	← 2	← 1	4	7 ← 6	↖
A	↖	2	↖	4 ← 3	↑ ↖	3	3	3 ← 2	← 1	3	6	9
G		↑	↑ ↖	3	6 ← 5	← 4	← 3	← 2	4 ← 3	5	8	↑
G			↑ ↖	2	5	5 ← 4	← 3	← 2	4 ← 3	4	7	↑
C			↑	4	7	7 ← 6	← 5	← 4	6			
C												
A												
		A	A	G	C	C	T	T	G	C	A	A

[Anwendung 60]