Algorithm Engineering-Projekt

String Alignment mit Neeldeman-Wunsch

Tom Wegener, 18INM/TZ

2019-02-10

Inhaltsverzeichnis

1 Einleitung

1.1 generelle Aufgabenstellung

Am Anfang des Semesters soll sich für ein Projekt entschieden werden, dieses dann in vier Schritten über den Zeitraum des Semesters ausprogrammiert werden. Zu diesen vier Schritten gehört jeweils eine Abgabe.
Zuerst sollte ein Parser entwickelt werden, der die ausgewählten Daten ausliest und abspeichert. Anschließend eine erste Version des Algorithmus ausprogrammiert werden und erste Laufzeiten gemessen werden, die anschließend auch graphisch dargestellt werden. Aus den Laufzeiten sollten dann als dritte Aufgabe Optimierungsmöglichkeiten aufzeigen. Außerdem wurde die Verwendung eines Profilers empfohlen. In der vierten Aufgabe werden dann zwei Algorithmen auf unterschiedliche Art und Weise verglichen.

1.2 spezielle Problembeschreibung

In dieser Arbeit wird das Problem String-Alignment behandelt. Die Daten werden aus der Datenbank des "National Institute of biotechnical Information"(ncbi) in Form von Fasta-Dateien entnommen und durch einen Parser eingelesen.
Anschließend wird ein Wert ausgegeben, der der Ähnlichkeit von den zwei Strings entspricht. Dafür gibt es verschiedene Algorithmen.
Das Projekt wurde zuerst teilweise in Python umgesetzt und anschließend in Go (bzw golang) übersetzt und fertig gestellt, um die Laufzeiten niedrig zu halten.

1.3 Charakteristika der Eingabedaten

Die Fasta-Files haben den Aufbau, dass sie mehrere Blöcke an DNA-Sequenzen enthalten, jeder Block wird über ein »ëingeleitet, dahinter steht dann eine Beschreibung, die auch eine ID enthält. In den nachfolgenden Zeilen sind dann die DNA-Strings durch Zeichenketten bestehend aus Buchstaben enthalten. Diese Zeichenketten haben eine maximale Länge von 80 zeichen pro Zeile und sind deshalb auf mehrere Zeilen verteilt.

1.4 Messumgebung

Getestet wurden zwei Fasta-Dateien, Aedes Aegypti und Aedes Albopictus. Zusätzlich werden bei Programmstart mehrere String zufällig generiert.
Programmiert wurde zuerst in Python3 mit der Version 3.6.7 und anschließend in go (golang) mit der Version 1.10.4. Audgeführt wurde der Code hauptsächlich auf einem Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz mit 16GB Arbeitsspeicher. Das Betriebssystem ist eine Linux-Distribution mit Ubuntu 18.10 als Basis.

2 Algorithmen und Optimierung

2.1 Needleman-Wunsch

2.2 Paraleller Needleman-Wunsch

3 Laufzeitmessungen

4 Ausblick