

Praktikum Data-Mining und Sequenzanalyse

Aufgaben Clustering

Markus Fleischauer

1 Allgemeine Projektanforderungen

- Versionskontrolle mit Git
- Projektmanagement mit Gradle
- Implementierung der Klassen mit den entsprechenden Algorithmen
- API Dokumentation mit javadoc
- Benutzerinterface als Kommandozeilenprogramm (CLI) + README/Hilfe
- Erstellen Sie ein Klassendiagramm (UML) für das Projekt
- Erläutern Sie ihre Implementierung. Gab es Schwierigkeiten? Gibt es noch Fehler?

2 Implementierung - Klassenstruktur

Entwerfen Sie eine sinnvolle Klassenstruktur für ihr Clustering tool. Bedenken Sie dabei, dass man die Implementierung sich auch als Library eignet.

- Allgemein:
 - Datenstruktur(en) zur Darstellung gewurzelter und ungewurzelter Bäume
 - Clustering Algorithmen (WPGMA, UPGMA, NJ) zur Berechnung eines Baumes aus einer Distanzmatrix
 - Darstellung einer Distanzmatrix
 - Bestimmung der Metrik einer Distanzmatrix
 - Berechnung einer Distanzmatrix aus einer Menge aus Sequenzen (Paarweise Alignments)
- Package: *io*
 - Reader und Writer fuer *FASTA* Dateien
 - **FastaReader** Sequenzen aus Datei im (multiple-)FASTA Format lesen
 - **NewickWriter** Bäume im *Newick* Format in Datei schreiben.

```
> Name1
CTAGAGTGTAGTCATGTCAGTGCAGC
> Name2
GTAGCGCGGTTTAAGTCTAGTGCAGC
```

- Package: *cli*
Paket für den Kommandozeilenparser und die main Methode:
 - Klasse mit `main()` Methode
 - Wahlweise Eingabe:
 - * mehrerer Sequenzen in einer Datei
 - * einer Distanzmatrix
 - Ausgabe des resultierenden Baumes im Newick format (Es sollte std-out und Datei möglich sein)
 - Parameter zur Wahl des Clustering Algorithmus (WPGMA, UPGMA, NJ)

3 Validierung der Algorithmen - Tests

Gewährleistet die Korrektheit eurer Algorithmen durch die Implementierung folgender (Unit)Tests. *Hinweis: Die frühzeitige Implementierung von Tests erleichtert den Debuggingprozess erheblich.*

1. Implementiert geeignete Unit Tests für alle 3 Algorithmen (WPGMA, UPGMA, NJ), welche die korrekte Funktionsweise gewährleisten. (Toy-Examples, Grenzfälle, etc.)
2. Testen sie eine zufällige Anzahl zufällig erzeugter DNA Sequenzen (in geeigneter Größenordnung) und verwendet sie als Eingabe für alle 3 Implementierungen (WPGMA, UPGMA, NJ).

4 Anwendung der Implementierung

Allgemeine Hinweise:

- Die folgenden Aufgaben, sollen mit Hilfe ihres Kommandozeilenprogramms gelöst werden.
- Geben sie stets die Ausgeführten Befehle an, sodass die Ergebnisse reproduzierbar und nachvollziehbar sind.
- Geben sie das verwendete Testsystem an (CPU, RAM, HDD/SSD, OS(-Version), Java-Version, JVM Parameter).

Daten: Folgende Testdaten liegen im FASTA-Format vor ¹

- `aquifex-tRNA-clean.fasta`: tRNAs von *Aquifex aeolicus*

Aufgaben:

1. Generiert per Hand für je ≥ 5 Taxa eine additive Baummetrik (Distanzmatrix) und eine Ultrametrik und testet damit die Algorithmen.
2. Erzeugt ≥ 6 beliebige zueinander ähnliche DNA-Sequenzen und testet damit wieder Euer Programm.
3. Bestimmt Stammbäume für die tRNAs von *Aquifex aeolicus* und diskutiert das Ergebnis.
4. Geht auf `www.ncbi.nlm.nih.gov/BLAST` und durchsucht die NCBI Nucleotide Collection nach der Sequenz der menschlichen Hexokinase (NM_000162). In welcher Reihenfolge tauchen Sequenzen anderer Spezies auf? Ladet euch die menschliche Gensequenz und fünf gefundene Sequenzen von anderen Organismen herunter. Bestimmt die Stammbäume dieser Sequenzen.
5. Diskutiert prinzipbedingte Vor- und Nachteile der Verfahren.

¹aus NCBI Nucleotide und Uniprot