HTW Berlin
Angewandte Informatik

18.10.2022


Exposee für eine Bachelorarbeit zum Thema:

# Recognizing and correcting reading-errors in DNA sequences using machine learning

Vorgelegt von:

**Fabian Vogt**
Friedrichsbrunner Straße 40
12347 Berlin
Email: fabian.vogt@student.htw-berlin.de
Matrikelnr.: s0570800

Erstgutachter:

**Prof. Dr. Christian Herta**

Zweitgutachter:

**Dr. Christian Krumnow**

# Inhaltsverzeichnis

# Motivation

Beim Einlesen von DNA-Sequenzen können bestimmte Fehler auftreten, beispielsweise, dass einzelne Basen nicht korrekt erkannt werden. Dies kann im schlimmsten Fall verheerende Konsequenzen haben, wie z.B. Fehldiagnosen.
Deshalb ist es wichtig diese Fehler zu erkennen und zu beseitigen. In dieser Bachelorarbeit soll untersucht werden, wie gut sich ein Machine-Learning-Ansatz zu Lösung des Problems eignet.

# Vorläufige Zielsetzung

Für die Implementierungsphase lassen sich bereits einige Zwischenziele definieren (siehe Methoden und Vorgehen):

- 1) Erstellen von Testdaten
- 2) Implementierung/Verwendung eines Sequenzmodells
- 3) POC für Fehlererkennung durch Reinforcement-Learning-Modell
- 4) Wechsel zu Transformer-Model – falls es nicht bereits in Verwendung ist
- 5) Optimierung der Fehlererkennung

Das Ziel der Implementierung ist, dass Lesefehler in DNA-Sequenzen erfolgreich erkannt und korrigiert werden sollen. Eine Korrektur war dann erfolgreich, wenn

- Im Durchschnitt von *N* Fehlern mindestens einer richtig korrigiert wird
- Im Durchschnitt mehr Fehler korrigiert werden, als dass neue erzeugt werden

Am Ende der Bachelorarbeit soll die Frage beantwortet werden, ob und wie gut sich ein Reinforcement-Learning Modell in Kombination mit einem Sequenzmodell eignet, um Lesefehler in DNA-Sequenzen zu korrigieren.

# Methoden und Vorgehen

## Trainingsdaten

Im ersten Schritt sollen Daten erzeugt werden. Ein Set von Trainingsdaten ist notwendig, um das ML-Model zu trainieren. Es müssen also realistische Fehler in reale DNA-Sequenzen eingefügt werden. Um die Machbarkeit der Lösung zu prüfen, liegt der Fokus zunächst auf den simpelsten Fehlern:
Die Veränderung einzelner Basen der Sequenz (z.B. ATGC → ATAC).

Die unverfälschten Ausgangsdaten ergeben den Validierungsdatensatz.

## Vektorisierung der DNA-Sequenzen

Die fehlerhaften DNA-Sequenzen werden mit Hilfe eines Sequenzmodels verarbeitet und als abstrakter Vektor repräsentiert. Die Idee ist, dass fehlerhafte DNA-Sequenzen anhand ihrer Positionen im Vektorraum erkennbar sein müssten. Die Positionen fehlerhafter Sequenzen sollten sich in der Theorie also eindeutig von den Positionen fehlerfreier Sequenzen abgrenzen lassen.

Für die Wahl des Sequenz-Models bieten sich zwei Optionen:

- Option 1: Verwendung eines bereits trainierten Transformer-Models (Vgl. NGS read classification using AI)
- Option 2: Implementierung eines eigenen, simplen Models (z.B. Recurrent Neural Network) um ein Proof-Of-Concept der DNA-Vektorisierung zu erstellen

## Erkennung und Korrektur der DNA-Fehler

Der Output des Sequenzmodels (abstrakter Vektor) kann anschließend als Input für ein weiteres Machine-Learning Modell verwendet werden. Es soll darauf trainiert werden, fehlerhafte Vektoren zu erkennen und die DNA-Sequenz zu korrigieren.

Dieses Modell ist der Kern Implementierung und der Bachelorarbeit.

### Reinforcement Learning

Für die Wahl der Art des Modells, eignet sich Reinforcement-Learning (RL). Beim RL wird ein Agent darauf trainiert, eine Sequenz von Aktionen auszuführen, um eine möglichst hohe Belohnung zu bekommen (Vgl. A Brief Survey of Deep Reinforcement Learning).

Im Falle der DNA-Fehlerkorrektur könnte man die Belohnungen des Agenten beispielsweise so definieren:

- Fehler richtig korrigiert: +5
- Fehler erkannt, aber falsch korrigiert: +1
- Kein Fehler erkannt, jedoch Fehler vorhanden: -1
- Fehler erkannt, jedoch kein Fehler vorhanden: -5

Die Aktionen, die der Agent ausführen kann, sind im simpelsten Fall:

- Base austauschen (z.B. für die Base A):
  - Tausch A → T
  - Tausch A → G
  - Tausch A → C
- Nichts tun (A → A)

Nachdem eine Aktion erfolgt ist, wird die Belohnung anhand der Validierungsdatensätze berechnet.

## Erfordernisse und Randbedingungen

Keine (?)

## Erwartete Ergebnisse

- Implementierung Sequenzmodell
- Implementierung Reinforcement-Learning-Modell
- Dokumentation

## Zeitplan

Woche 1:
- Research zu DNA-Sequenzierung
- Planung des ML-Models
- Aufstellen von Trainingsdaten

Woche 2-4:
- Implementierung POC
- Tests
- Notizen für BA sammeln

Woche 5-6:
- Fertigstellung der Implementierung
- Optimierung / Performance
- Struktur der BA, Einleitung / Abstract

Woche 7-9:
- Fertigstellung BA

Woche 10:
- Pufferzeit

## Grob-Gliederung

- Einleitung
- Lesen von DNA-Daten
    - Was ist DNA?
    - Wie wird DNA eingelesen?
    - Was sind die häufigsten Ursachen für Fehler beim Einlesen?
    - Welche Arten von Fehlern gibt es?
    - Wie können Lesefehler identifiziert werden?
    - Wie können Lesefehler korrigiert werden?
- Machine Learning
    - Was ist ML?
    - Warum eignet sich ML zur Korrektur von Lesefehlern in DNA-Daten?
    - Welche ML-Modelle gibt es?
    - Welches Modell ist für die Fehlererkennung geeignet? --> RL
    - Wie lassen sich Sequenzmodell und RL-Modell kombinieren?
- Projektumsetzung
    - Definitionsphase
        - Anforderungsanalyse
    - Planungsphase
        - Wie sieht das Training aus?
        - Welche Modelle werden konkret genutzt?
    - Implementierungsphase
    - Training
    - Tests
- Fazit
    - Hat sich die Annahme, dass ein RL-Ansatz für das Problem geeignet ist, bewahrheitet?
    - Eigenschaften des Modells z.B. Lernkurve
    - Vergleich mit statischem Lösungsansatz
    - Funktionsanalyse
- Quellen

# Literaturangaben


| Author | Titel | Link | Zugriffsdatum |
|---|---|---|---|
| James M. Heather, Benjamin Chain | The sequence of sequencers: The history of sequencing DNA | https://www.sciencedirect.com/science/article/pii/S0888754315300410 | 18.10.2022 |
| Benjamin Voigt, Oliver Fischer, Christian Krumnow, Christian Herta, Piotr Wojciech Dabrowski | NGS read classification using AI | https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0261548 | 18.10.2022 |
| Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin | Attention Is All You Need | https://arxiv.org/pdf/1706.03762.pdf | 18.10.2022 |
| Eduardo Muñoz | Attention is all you need: Discovering the Transformer paper | https://towardsdatascience.com/attention-is-all-you-need-discovering-the-transformer-paper-73e5ff5e0634 | 18.10.2022 |
| Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, Anil Anthony Bharath | A Brief Survey of Deep Reinforcement Learning | https://arxiv.org/pdf/1708.05866.pdf | 18.10.2022 |
| Ke Yu, Chao Dong, Liang Lin, Chen Change Loy | Crafting a Toolchain for Image Restoration by Deep Reinforcement Learning | https://arxiv.org/abs/1804.03312 | 18.10.2022 |