



An efficient error correction algorithm using FM-index

Seminararbeit

vorgelegt von:

Eike Gebauer

Matrikelnummer: 408126

Studiengang: MSc. Informatik

Zusammenfassung

An dieser Stelle soll eine Zusammenfassung der wissenschaftlichen Arbeit stehen. Idealerweise ist die Zusammenfassung genau eine Seite lang, was in etwa 350 - 450 Wörtern entspricht. Sie bildet zusammen mit dem Fazit einen Rahmen der gesamten wissenschaftlichen Arbeit und sollte auch für Außenstehende einfach zu verstehen und ansprechend formuliert werden.

Inhaltsverzeichnis

1	Einleitung	1
1.1	DNA-Sequenzierung	1
1.2	Fehlerkorrektur	2
2	Methodik	5
2.1	Mathematische Notation	5
2.2	Algorithmus	5
3	Implementierung	7
3.1	Quelltext-Abschnitte	7
4	Ergebnisse	9
5	Diskussion	11
6	Fazit	13

1 Einleitung

DNA Sequenzierung ist aus der biologischen und medizinischen Forschung nicht mehr wegzudenken. Die Entwicklung immer günstigerer und leistungsfähigerer Sequenzierungsverfahren hat in den letzten Jahrzehnten für einen rapiden Anstieg der zu verarbeitenden Datenmengen gesorgt. Hinzu kommt, dass die verwendeten Verfahren nicht fehlerfrei sind, was die Verarbeitung der Daten weiter erschwert.

In dieser Arbeit wird ein Paper behandelt, in dem der Fehlerkorrekturalgorithmus FMOE vorgestellt wird. Zu erst wird aber der Hintergrund in den Bereichen Biologie und Bioinformatik erläutert.

1.1 DNA-Sequenzierung

Das Ziel der de-novo DNA-Sequenzierung ist die Bestimmung der Nukleotidfolge in DNA ohne ein bereits bekanntes Referenzgenom. Wo bei alten Verfahren noch alle Basen nacheinander bestimmt wurden, werden seit dem sogenannten Shotgun Sequencing viele Teilstücke der DNA parallel analysiert. Das macht es möglich, lange DNA Stränge mit Verfahren zu sequenzieren, die nur kurze Fragmente verarbeiten können. Gleichzeitig ermöglicht es die parallele Sequenzierung von mehreren Fragmenten, was den Vorgang extrem beschleunigt.

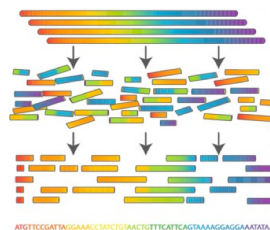


Abbildung 1.1: Shotgun Sequenzierung: Kopien des DNA-Strangs (oben) werden an zufälligen Positionen fragmentiert und sequenziert(darunter), in die korrekte Reihenfolge gebracht (3. von oben) und zur Gesamtsequenz zusammengesetzt (unten)

Die Reihenfolge der DNA-Fragmente aufrecht zu erhalten, ist möglich, aber relativ langsam. Die sogenannten next-gen Sequenzierungsverfahren, welche den aktuell höchsten Durchsatz erzielen, verzichten daher darauf. Um trotzdem auf die ursprüngliche Reihenfolge und damit auf die gesamte DNA-Sequenz

1 Einleitung

zurückschließen zu können, wird das Verfahren für mehrere Kopien desselben DNA-Strangs durchgeführt.

Wie in 1.1 zu sehen, trennt man also mehrere Kopien des Strangs an zufälligen Positionen in ungefähr gleich lange Fragmente. Alle Fragmente können dann unabhängig von einander - und damit parallel - sequenziert werden. Das Ergebnis der Sequenzierung eines Fragments wird Read genannt. Aufgrund der zufälligen Trennungspositionen überlappen sich viele dieser Reads. Vorausgesetzt die Abdeckung¹ reicht an allen Stellen der Sequenz aus, kann man mit den Überlappungen die Reads in die ursprüngliche Reihenfolge bringen und so die gesuchte DNA-Sequenz bestimmen. Dieser Prozess heißt **Assembly**.

Wie bereits erwähnt, sind die aktuellen Verfahren zur Bestimmung der Reads nicht fehlerfrei. Es kann zum Überspringen (Deletion) oder Vertauschen (Substitution) richtiger Basen oder zur Einfügung (Insertion) falscher Basen kommen. Kollektiv werden diese Fehler Indels genannt.

Die Assembly üblicherweise durch Überlappungsgraphen oder de Bruijn Graphen bewältigt. Überlappungsgraphen können Fehler bis zu einem gewissen Maß kompensieren, bringen allerdings hohe Speicher- und Rechenanforderungen mit sich. De Bruijn Graphen sind deutlich ressourcensparender, tolerieren aber keine Fehler. Aus diesem Grund wird bei großen Datenmengen oft auf de Bruijn Graphen mit vorheriger Fehlerkorrektur zurückgegriffen.

Fehler in den Reads sorgen für verschiedene Probleme bei der Assembly mithilfe eines de Bruijn Graphen:

Fragmentierte Assembly

Fehlerhafte Reads verhindern das Erkennen von Überlappungen und sorgen für Graphen, die keine valide Assembly zulassen.

Misassembly

Fehlerhafte Reads überlappen sich in manchen Fällen mit korrekten Reads und sorgen für eine valide Assembly, die aber eine falsche Sequenz ergibt.

Dieser Fall ist im Nachhinein schwer zu erkennen.

Zusätzlich erhöht sich die Komplexität des Graphen bei fehlerhaften Reads unter Umständen enorm.

1.2 Fehlerkorrektur

Aufgabe der Fehlerkorrektur ist es, diese Komplikationen zu verhindern. Da bei der de-novo Sequenzierung kein Referenzgenom zur Verfügung steht, kann die Fehlerkorrektur nur auf Grundlage der Reads geschehen. Die üblichen Verfahren stützen sich dabei auf statistische Modelle, um fehlerhafte Reads zu finden und zu verwerfen oder zu korrigieren.

¹Abdeckung (englisch Coverage) beschreibt die Anzahl einzigartiger Reads, die dieselbe Base enthalten.

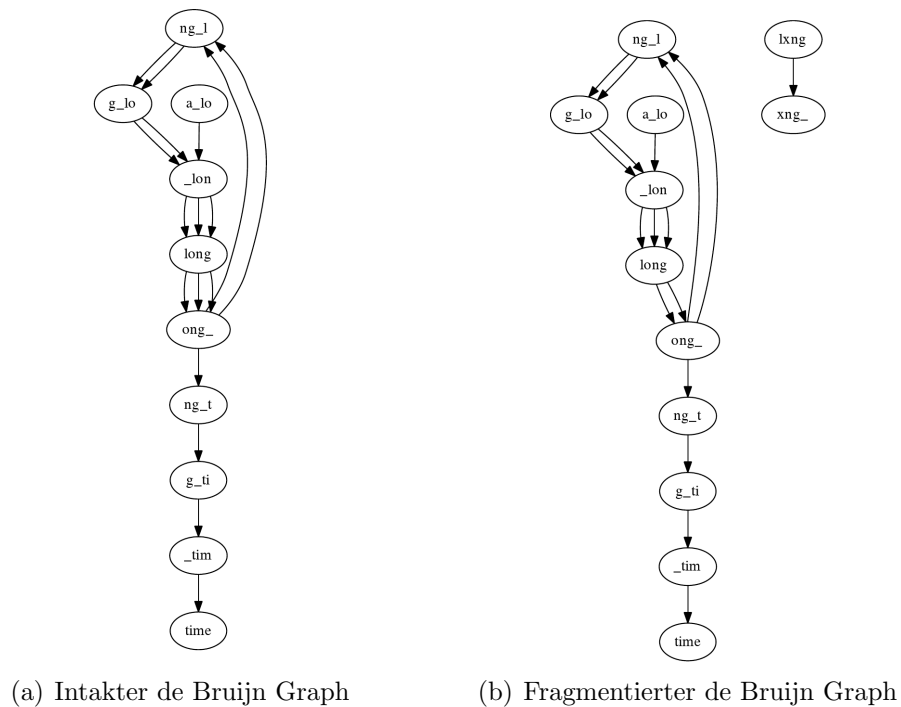


Abbildung 1.2: De Bruijn Graph des Texts „a long long time“. Oben der Graph aus fehlerfreien Fragmenten, unten der fragmentierte Graph, hervorgerufen durch einen Substitutionsfehler.

Das hier behandelte Paper teilt Fehlerkorrekturalgorithmen in drei Kategorien:

k-mer Frequenzspektrum Methoden erzeugen eine Erhebung aller k-mere² und ihrer Häufigkeiten aus den Reads und ersetzen seltene k-mere durch Häufige. Diese Verfahren sind empfindlich gegenüber Wiederholungen in der DNA-Sequenz.

Suffix-Tree / -Array Methoden erzeugen einen Index der Suffixe der Reads in Form eines Arrays oder Baums und verhalten sich ansonsten ähnlich zu den k-mer-basierten Methoden. Die variable Länge der Suffixe macht diese Verfahren etwas resistenter gegen Wiederholungen.

Überlappungsbasierte Methoden suchen für jeden zu korrigierenden Read alle überlappenden Reads und korrigieren potenzielle Fehler mit einem

²k-mer wird eine k Zeichen lange Teilkette einer Zeichenkette genannt. Beispiel: Die 3-mere von „ABCDE“ sind „ABC“, „BCD“ und „CDE“.

1 Einleitung

MSA³ der gefundenen Reads.

Durch die Verwendung der ganzen Reads sind diese Verfahren nicht anfällig für Wiederholungen, allerdings ist die Berechnung des MSA Berechnung sehr langsam.

Weitere Vor- und Nachteile der verschiedenen Vorgehensweisen werden später noch erläutert.

³MSA ist die Abkürzung für Multiple Sequence Alignment und beschreibt das alignen von mehr als drei DNA-Sequenzen. Mit alignment ist an dieser Stelle der methodische Vergleich von Sequenzen gemeint.

2 Methodik

Im Methodik-Kapitel werden die mathematischen Ausführungen des Verfahrens bzw. des Algorithmus vorgestellt, jedoch technische Details zur Umsetzung und Implementierung (falls nötig) auf ein darauffolgendes Kapitel verlagert.

2.1 Mathematische Notation

Mathematische Formeln können mittels der `\begin{align}...\end{align}` Umgebung gesetzt werden:

$$f(n) = \begin{cases} n/2, & \text{wenn } n \text{ gerade,} \\ 3n + 1, & \text{wenn } n \text{ ungerade.} \end{cases} \quad (2.1)$$

$$g(n) = \frac{n}{2} \quad (2.2)$$

2.2 Algorithmus

Eigene Algorithmen beschreibt man am Besten mit Hilfe von Pseudo-Code und dem Paket `algorithm`.

Algorithm 1 Algorithmus

Require: Argument $n \in \mathbb{N}$

```
 $a = 0$   
for  $i = 0, \dots, n$  do  
     $a = a + 1$   
end for  
return  $a$ 
```

3 Implementierung

In diesem Kapitel werden Technische Details, wie zum Beispiel die Verwendete Laufzeitumgebung, Laufzeitanalyse oder wichtige Implementierungsdetails behandelt. Dabei sollte beachtet werden inwiefern diese Angaben für die Arbeit von Bedeutung sind. Die Namen einzelner Funktionsaufrufe oder Klassendiagramme sind zum Beispiel im Allgemeinen für den Leser uninteressant.

3.1 Quelltext-Abschnitte

Eigene Auszüge aus dem Quelltext bindet man am Einfachsten mit dem Befehl `\lstinputlisting{}` ein. Das Ergebnis ist in 3.1 zu sehen. Optionen für die Sprache, Tab-Breite etc. von `\lstinputlisting` können auch am Anfang des Dokuments mit `\lstset{}` gesetzt werden. Das Paket erlaubt mit `firstline=...` und `lastline=...` auch die Einbindung von einzelnen Zeilen einer Datei.

Quelltext 3.1: Codebeispiel

```
int myTest(n) {  
    a = 0;  
    for (int i = 0; i <= n; i++)  
        a = a + 1;  
    return a;  
}
```


4 Ergebnisse

Dieses Kapitel sollte die Ergebnisse beinhalten, die mit den Methoden aus Kapitel 2 erstellt wurden.

Tabelle 4.1: Beispieltabelle

Spalte1	Spalte2	Spalte3
1	2	3

5 Diskussion

In diesem Kapitel werden die zuvor vorgestellten Ergebnisse der Arbeit diskutiert. Häufig wird es mit dem Fazit zusammengelegt.

6 Fazit

Dieses Kapitel bildet die abschließende Zusammenfassung der Arbeit. Dazu können die folgende Punkte behandelt werden:

- Reflexion: wurden die Ziele der Arbeit erreicht?
- mögliche Erweiterungen und Verbesserungen („future work“)

Abbildungsverzeichnis

- 1.1 Shotgun Sequenzierung: Kopien des DNA-Strangs (oben) werden an zufälligen Positionen fragmentiert und sequenziert(darunter), in die korrekte Reihenfolge gebracht (3. von oben) und zur Gesamtsequenz zusammengesetzt (unten) 1
- 1.2 De Bruijn Graph des Texts „a long long time“. Oben der Graph aus fehlerfreien Fragmenten, unten der fragmentierte Graph, hervorgerufen durch einen Substitutionsfehler. 4

Tabellenverzeichnis

4.1	Beispieltabelle	9
-----	---------------------------	---

Literaturverzeichnis

- [1] JELE, Harald: *Wissenschaftliches Arbeiten: Zitieren*. R. Oldenbourg Verlag, 2010

Eidesstattliche Erklärung

Hiermit versichere ich, dass die vorliegende Arbeit über „*Titel*“ selbstständig verfasst worden ist, dass keine anderen Quellen und Hilfsmittel als die angegebenen benutzt worden sind und dass die Stellen der Arbeit, die anderen Werken – auch elektronischen Medien – dem Wortlaut oder Sinn nach entnommen wurden, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht worden sind.

Vorname Nachname, Münster, 13. Januar 2019

Ich erkläre mich mit einem Abgleich der Arbeit mit anderen Texten zwecks Auffindung von Übereinstimmungen sowie mit einer zu diesem Zweck vorzunehmenden Speicherung der Arbeit in eine Datenbank einverstanden.

Vorname Nachname, Münster, 13. Januar 2019