INSTITUT FÜR INFORMATIK
Algorithmische Bioinformatik

Universitätsstr. 1 D–40225 Düsseldorf

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Contig-Assembly der MHC-Region mittels Linearer Programmierung

Marvin Lindemann

Bachelorarbeit

Beginn der Arbeit:	09. Juni 2019
Abgabe der Arbeit:	09. September 2019
Gutachter:	Prof. Dr. Gunnar W. Klau
	Dr. Alexander Dilthey

Erklärung

Hiermit versichere ich, dass ich diese Bachelorarbeit selbstständig verfasst habe. Ich habe dazu keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Düsseldorf, den 09. September 2019

_____

Marvin Lindemann

Zusammenfassung

Hier kommt eine ca. einseitige Zusammenfassung der Arbeit rein.

Inhaltsverzeichnis

1 Einleitung **1**

2 Formalisierung **3**

2.1 Anforderungen an die Lösung 3

3 Das lineare Programm **5**

3.1 Eine kurze Einführung 5

3.2 Assembly als LP 6

4 Wie man eine Lösung Grafisch darstellt **7**

5 Algorithmus Phase 1: Grobe Lösung aufbauen **7**

6 Algorithmus Phase 2: Lösung mithilfe vom LP verfeinern **7**

7 Auswertung der Ergebnisse **7**

8 Diskussion **7**

Abbildungsverzeichnis **8**

Tabellenverzeichnis **8**

1 Einleitung

Der Haupthistokompatibilitätskomplex, kurz MHC, ist ein Teilstück der DNA von Wirbeltieren, welches unter anderem eine tragende Rolle bei Vorgängen des Immunsystems besitzt. Durch seine große Variabilität dient es als Ausweis der körpereigenen Zellen, um sich vor dem Immunsystem von Fremdgewebe zu unterscheiden. Daraus ergibt sich ein großes Interesse für Organtransplantationen, den Aufbau dieses Komplexes zu kennen. Leider resultiert aus der Variabilität ein eben so großes Problem bei der Analyse dieses Aufbaus. Um dieses Problem zu verstehen, müssen wir erst verstehen, wie bei der Assemblierung, also der Bestimmung der DNA-Sequenz, vorgegangen wird.

Bei langen DNA-Sequenzen können die Basenpaare nicht direkt ermittelt werden. Daher werden sie in kürzere Stücke zerlegt, für die dann die Basenpaare bestimmt werden können. Dies sind die *Reads*. Für das Erstellen solcher Reads gibt es verschiedene Verfahren, die alle verschiedene Vor- und Nachteile haben. So werden bei der sogenannten Illuminar-Sequenzierung sehr kleine Reads mit einer Länge von ungefähr 200 Basenpaaren erzeugt. Diese können sich teilweise überlappen, wodurch es möglich ist, einige Reads zu so genannten *Contigs* zusammenzufügen. Im Idealfall passen die Reads so gut zueinander, dass der gesamte untersuchte DNA-Bereich rekonstruiert werden kann. Ein Fall, bei dem dies nicht möglich ist, zeigt diese Situation:

ATTAAGCCTTAGGGTTATATCATATATATATATA

TATATATATATGTAAGCGTTCGTTGTCTC

Durch den sehr simplen Aufbau des Zwischenbereiches aus abwechselnden Adenin- und Thyminbasen ist es nicht möglich, die genaue Positionierung zu bestimmen. Auch wenn die Contigs nicht exakt zueinander positioniert werden können, so ist dies zumindest ungefähr möglich. Noch größere Schwierigkeiten machen Regionen in der DNA, die über eine längere Strecke komplett identisch sind, sogenannte Repeats. Das Problem wird in Abbildung 1 verdeutlicht. Zu sehen sind fünf Contigs, wobei ein Pfeil von Contig a zu Contig b verdeutlicht, dass in der DNA Contig a direkt vor Contig b kommt. Da Contig 63 (blau) zwei mal in der DNA vorkommt, ist es nicht trivial zu bestimmen, wie der Strang verläuft. Und MHC ist sehr repetitiv, besitzt also viele Wiederholungen. Daher ist diese Methode ungeeignet um MHC zu assemblieren.

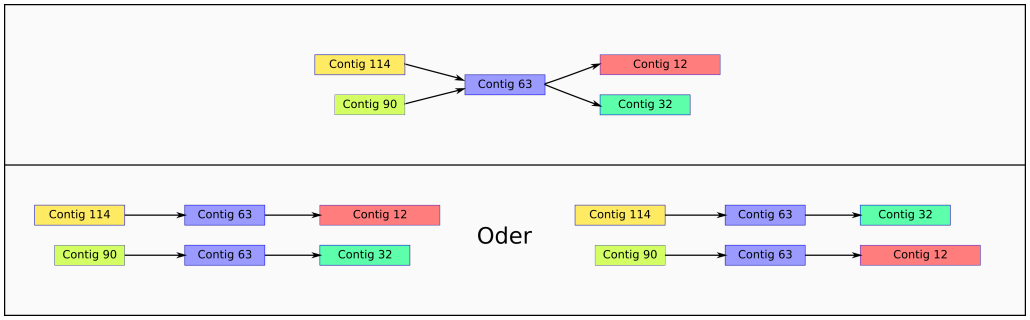

Abbildung 1: Repeat in einer Sequenz

Abbildung 2: Ein Long-Read über mehrere Contigs

Eine weitere Möglichkeit der Sequenzierung ist die Nanopore-Sequenzierung. Hierbei können sehr lange Reads von über 10 000 Basenpaaren gelesen werden. Aufgrund des Längenunterschieds werden die Reads aus der Illuminar Sequenzierung auch *Short-Reads* genannt, und die Reads aus der Nanopore-Sequenzierung *Long-Reads*. Durch ihre Länge umfassen die Long-Reads oftmals bereits vollständige Repeats plus Umgebung. In unserer Abbildung entspräche dies einem Read, in dem Contig 132, Contig 63 und Contig 32 (gelb, blau und grün) Teil von einem Read sind und es keine Schwierigkeiten diesbezüglich gibt. Leider ist die Fehlerrate bei dieser Methode sehr hoch. So werden sehr viele Long-Reads aus der selben Region benötigt, um die richtigen Basenpaare verlässlich zu bestimmen.

Daher hat sich die Manchot-Forschungsgruppe vom Institut für Medizinische Mikrobiologie und Krankenhaushygiene der HHU, mit deren Zusammenarbeit diese Arbeit entstand, eine Kombination der beiden Methoden entwickelt. Die verlässlichen Contigs aus der Illuminar-Sequenzierung wurden auf die Long-Reads gemappt und so deren Abstände zueinander bestimmt. Betrachten wir zur Anschauung Abbildung 2 eines Long-Reads. Die kleinen rot eingefärbten Bereiche stellen Fehler dar, die bunten Bereiche stellen Gebiete dar, in denen die Basenpaarsequenzen mit denen eines Contigs aus den Illuminar-Daten übereinstimmen. In der selben Situation wie in Abbildung 1 hätten wir nun die Information erhalten, dass Contig 114 und Contig 32 zusammengehören, also die rechte Auflösung der Abbildung richtig ist. Die Manchot-Forschungsgruppe hat diese Daten gesammelt und daraus eine Liste aus paarweisen Daten extrahiert. Diese besitzt die Form: Contig a hat zu Contig b die Entfernung d (in Anzahl von Basenpaaren zwischen a und b). Dieses Dreiertuple aus zwei Contigs und einer Distanz nennen wir einen *Constraint*.

Es bleiben noch einige Schwierigkeiten für die Assemblierung zu beachten. Die Distanzwerte zwischen den Contigs sind meistens durch Fehler in den Long-Reads verfälscht. Dadurch sind erst mehrere Constraints, die eine ähnliche Distanz zwischen zwei Contigs prognostizieren, wirklich belastend. Bis zu welchen Distanz sich Constraints noch bestätigen und ab wann sie sich widersprechen ist hierbei ein entscheidender Aspekt, der betrachtet werden muss. Die Repeats lassen sich nicht immer so eindeutig auflösen wie in unserem Beispiel. In manchen Fällen kann erst im Gesamtzusammenhang erkannt werden, wie der Strang verläuft. Letztlich bleibt noch die große Datenfülle als Herausforderung zu nennen: Bei rund 122 000 auftretenden Distanzen zwischen 2 124 Contigs ist eine manuelle Zusammenfügung nicht zielführend und mindestens eine Teilautomatisierung der Prozesse obligatorisch. Auf der anderen Seite sind die Constraints nicht gleichmäßg verteilt, sodass es Regionen gibt, bei denen mit sehr wenig Informationen ausgekommen werden muss.

Hier stellt sich die Frage, mit welchen Methoden diese Probleme bewältigt werden können. Denkbar wäre eine Umsetzung mittels Linearer Programmierung. Das Ziel dieser Arbeit wird sein, zu untersuchen, ob lineare Programmierung hierbei anwendbar ist, und welche Vor- und Nachteile lineare Programme mit sich bringen.

2 Formalisierung

Die vorliegenden Daten der Manchot-Forschungsgruppe bestehen aus zwei Dateien:

1. einer Liste von allen Contigs und deren Länge gemessen in der Anzahl an Basenpaaren, die im Contig auftreten
2. einer Datei mit den eingangserwähnten Constraints.

Letztere besitzt folgenden dreispaltigen Aufbau:

$$\begin{array}{ccc} a & b & 1000 \\ a & b & 1100 \\ b & c & 3000 \\ a & c & 6000 \end{array}$$

Dabei entspricht jede Zeile einem Constraint. Die ersten beiden Zeilen geben jeweils die beiden involvierten Contigs an. In der dritten Spalte sind die gemessenen Distanzen angegeben. Diese sind als Entfernung vom rechten Rand des ersten Contigs zum linken Rand des zweiten Contigs in Basenpaaren zu interpretieren. Negative Distanzen sind auf Überlagerungen einzelner Contigs zurückzuführen. Zusätzlich zu den Constraints aus den Long-Reads stammen etwa 2% der Constraints direkt aus der Illumina Sequenzierung. Hier wurde mehrmals die Distanz zwischen benachbarte Contigs gemessen und der Durchschnitt daraus berechnet. Dadurch können auch nicht ganzzahlige Werte als Distanzen auftreten.

2.1 Anforderungen an die Lösung

Nun wollen wir einige Gütekriterien für mögliche Lösungen festlegen, um während der Bearbeitung Orientierungspunkte für die weitere Optimierung des Algorithmus zu haben und um diesen nach Abschluss anhand dieser Kriterien zu bewerten. Folgende Punkte sollen beachtet werden:

1. Wenn mehrere Constraints eine ähnliche Distanz zwischen zwei Contigs sehen, sollten diese Contigs auch die Distanz möglichst gut erfüllen. Der Schwellwert hierbei liegt bei zwei Constraints.
2. Ein Großteil der restlichen Constraints sollte auch zu der Positionierung passen.
3. Es sollten möglichst wenig Contigs nahe zueinander positioniert werden, die keine gemeinsamen Constraints aufweisen. Diese Situation wird durch Abbildung 3 illustriert. In der Abbildung werden die Constraints durch gerichtete Kanten zwischen den Contigs dargestellt. Der oberer Teilabschnitt zeigt eine Lösung, bei der die Bedingung nicht erfüllt wurde. Realistischer ist jedoch, dass hierbei ein Teilblock bestehend aus Contig 12 und Contig 32 doppelt in dem DNA-Strang vorkommt und bei der Lösung des Problems zwei separate Stränge ineinander verflochten wurden. Dies ist in der unteren Bildhälfte dargestellt.

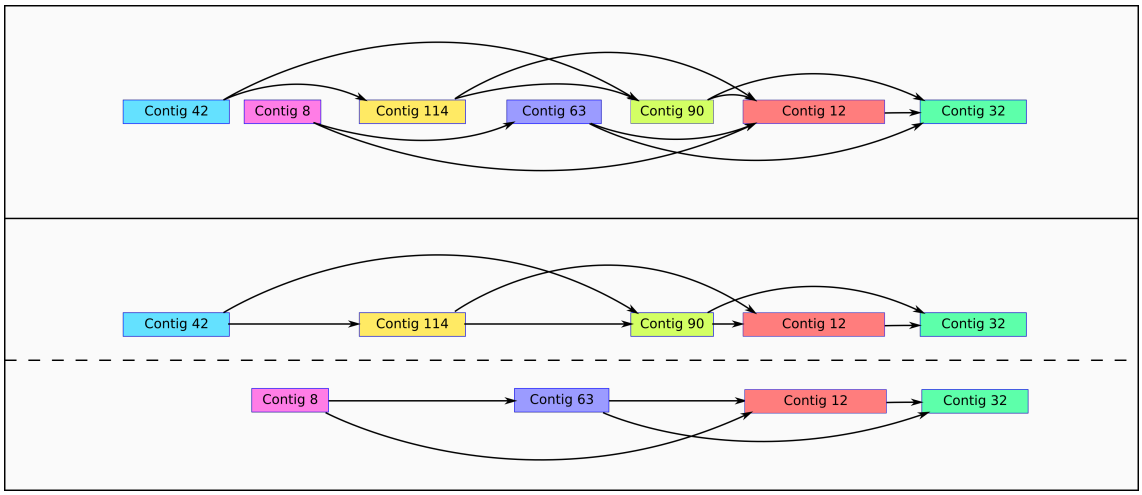

Abbildung 3: verflochtene Stränge

4. Die Entfernung vom ersten zum letzten Contig sollte zwischen 4,8 Millionen und 5 Millionen Basenpaare lang sein, da dies die ungefähre Gesamtlänge des Strangs ist.

5. Bei einer Lösung wäre es zudem gut, wenn man Informationen darüber besitzen würde, wie wahrscheinlich es ist, ob verschiedene Teilgebiete tatsächlich so im Strang auftreten. Dabei wäre zum Beispiel eine Unterteilung in „sichere“ und „unsichere“ Gebiete interessant.

Nun wollen wir konkretisieren, bis zu welchem Abstand Constraints ähnliche Distanzen haben. Dazu betrachten wir, wie die Standardabweichung der Distanzen von Constraints verteilt ist. In der Abbildung 4 werden die zu einem Basenpaar zugehörigen Constraints gegen ihre Standardabweichung geplottet. Dabei wurden pro Basenpaar die jeweiligen Distanzen der Constraints zu einer (Multi-)Menge zusammengefasst. Es wurde symmetrisch vorgegangen, das heißt Paare der Form (a,b) und (b,a) werden in der gleichen Menge behandelt. Ferner wurden Mengen mit einem Element nicht berücksichtigt, da es hier keine Abweichung gibt. Für die jeweiligen Mengen wurden dann die Standardabweichungen berechnet, der Größe nach geordnet und dann mit Berücksichtigung dieser Ordnung geplottet. Der Plot wird mit zwei Skalierungen angegeben: Auf der linken Seite sieht man eine logarithmische Skalierung, während der rechte Plot eine lineare Skalierung verwendet.

Eine optische Betrachtung der Plots legt folgende Interpretation nahe: Es gibt einen Bereich der natürlichen Abweichung in der Datenmenge. Dies entspricht dem relativ flachen Anfangsbereich des Graphen. Ab einem gewissen Punkt ëxplodieren"die Werte. Hier ist die Standardabweichung innerhalb der Constraints so hoch, dass man nicht mehr von natürlicher Abweichung innerhalb der Daten ausgehen kann. Die orangene Linie grenzt diese Bereiche intuitiv voneinander ab. Diese liegt bei einer Standardabweichung von 500 Basenpaaren. Somit unterstützen sich Constraints, deren Distanzwerte sich nicht um mehr als 500 Basenpaare unterscheiden.

Um die oben genannten Forderungen an eine Lösung zu erfüllen, ist es notwendig die

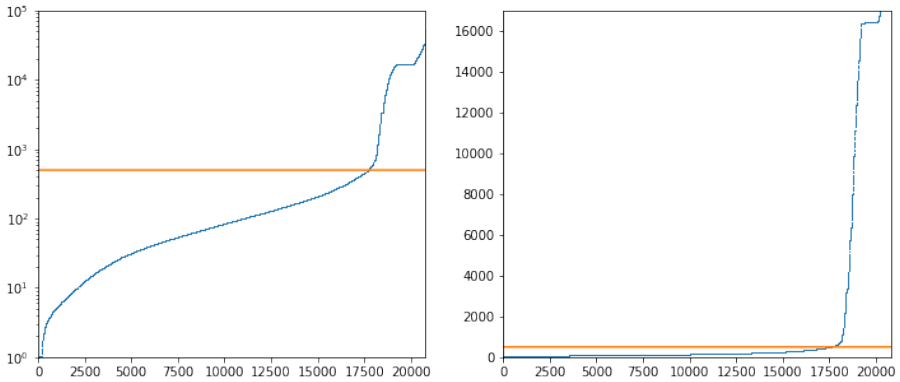

Abbildung 4: Standardabweichung der Distanzwerte

Repeats auszumachen und die Constraints auf diese Repeats aufzuteilen. Dazu halten wir uns an das Prinzip: „so viel wie nötig, so wenig wie möglich“. Um dies sicherzustellen, sollte ein Contig, der mehrfach in der DNA vorkommen soll, zwei Punkte für jede seiner Versionen erfüllen:

1. Es sollte mindestens ein Contig in der Nachbarschaft liegen, zu welchem es zwei oder mehr Constraints gibt.
2. Sowohl unter den Vorgängern als auch unter den Nachfolgern des Contigs, sollte es je einen Contig geben, der einen gemeinsamen Constraint aufweist.
3. Sowohl zu einem der Vorgängern als auch zu einem der Nachfolgern des Contigs, sollte es ein Constraint mit dem Contig geben.

Der erste Punkt soll sicherstellen, dass es nicht einfach ein Fehler in den Daten ist. Der zweite Punkt stellt sicher, dass der richtige Contig als Repeat markiert wurde. Wenn der Distanzwert eines Constraints nicht erfüllt ist, ist erstmal nicht klar, welcher der beiden beteiligten Contigs eine Repeat-Version haben soll.

3 Das lineare Programm

3.1 Eine kurze Einführung

Die Problemstellung in dieser Arbeit lässt sich als lineares Programm (kurz LP) formulieren. In der linearen Programmierung möchten wir, unter Berücksichtigung von linearen Nebenbedingungen an die Funktionsparameter, eine lineare Funktion maximieren oder minimieren. Formal mathematisch lässt sich der Minimierungsfall so fassen:

$$\begin{aligned} &\textit{Gegeben}: \quad c \in \mathbb{R}^n,\ A \in \mathbb{R}^{n \times m},\ b \in \mathbb{R}^m \\ &\textit{Gesucht}: \quad \arg\min_{x \in \mathbb{R}^n} \{c^T x \mid Ax \leq b\} \end{aligned}$$

Dabei wird folgende Notation verwendet:

Variablen $x_1, ..., x_n$

Zielfunktion $c^T x = \sum_{i=1}^{n} c_i x_i$

Nebenbedingungen $Ax \leq b \Leftrightarrow \sum_{i=1}^{n} a_{ji} x_i \leq b_j,\ j = 1, ..., n$

Lineare Programme lassen sich in Polynomialzeit berechnen. Daher eignen sie sich oft als Werkzeug für verschiedenste Probleme. Es ist auch möglich, den Definitionsraum der Variablen (teilweise) auf ganze Zahlen zu beschränken. Dies bezeichnet man dann als ein ganzahliges lineares Programm (kurz ILP). ILPs bieten einige Möglichkeiten die mit LPs nicht umzusetzen wären. Sie sind dafür aber NP-schwer und somit wahrscheinlich nicht polynomialzeitberechenbar.

3.2 Assembly als LP

Im Folgenden bezeichnet C die Menge aller Contigs und D die Multimenge aller Constraints. Wir möchten die Contigs so positionieren, dass der durchschnittliche Fehler aller Constraints möglich klein ist. Als Formel:

$$\mathrm{pos} = \underset{\mathrm{pos}\,:\,C \to \mathbb{N}}{\arg\min} \sum_{(a,b,\delta) \in D} |\,\mathrm{pos}(b) - \mathrm{pos}(a) - \delta|$$

Um daraus ein lineares Programm zu machen, führen wir für jeden Constraint (a, b, δ) aus D eine Hilfsvariable ε ein:

$$\varepsilon = |\,\mathrm{pos}(b) - \mathrm{pos}(a) - \delta|$$

Mit Hilfe dieser Variablen können wir die zu minimierende Zielfunktion wie folgt darstellen:

$$\sum_{d \in D} \varepsilon_d$$

Nun müssen wir noch die Informationen über die Fehler, also $\varepsilon = |\,\mathrm{pos}(b) - \mathrm{pos}(a) - \delta|$, einbauen. Da wir ohnehin die Summe der Fehler minimieren wollen, ist es ausreichend, folgende Ungleichung zu fordern:

$$\varepsilon \geq |\,\mathrm{pos}(b) - \mathrm{pos}(a) - \delta|$$

Dies liegt daran, dass bei Minimierung immer die untere Schranke angenommen wird, welche in diesem Fall die Gleichheit ist. Nun ist die Betragsfunktion aber nicht linear. Sie lässt sich aber äquivalent durch die folgenden beiden linearen Ungleichung darstellen:

$$\begin{aligned} \mathrm{pos}(b) - \mathrm{pos}(a) - \delta &\leq \varepsilon \\ -\mathrm{pos}(b) + \mathrm{pos}(a) + \delta &\leq \varepsilon \end{aligned}$$

Zusammengefasst erhalten wir also folgendes lineares Programm für die Berechnung der optimalen Positionierung:

$$
\begin{aligned}
\text{Variablen:} \quad & \mathrm{pos}(c)\ \forall c \in C \ \text{ und } \ \varepsilon_d\ \forall\, d \in D \\
\text{Zielfunktion:} \quad & \sum_{d \in D} \varepsilon_d \\
\text{Bedingungen:} \quad & \begin{aligned} \mathrm{pos}(b) - \mathrm{pos}(a) - \delta &\leq \varepsilon_d \\ -\mathrm{pos}(b) + \mathrm{pos}(a) + \delta &\leq \varepsilon_d \end{aligned} \quad \forall\ (a, b, \delta) = d \in D
\end{aligned}
$$

Distanzwerte weisen zu große Schwankungen auf, um auf ein Basenpaar genau zu sein. Daher bietet es sich an, die Relaxierung des LPs zu betrachten, also auch reelle Positionen zuzulassen. Durch diese Lockerung der Bedingungen lässt sich das Programm wesentlich schneller lösen.

4 Wie man eine Lösung Grafisch darstellt

5 Algorithmus Phase 1: Grobe Lösung aufbauen

6 Algorithmus Phase 2: Lösung mithilfe vom LP verfeinern

7 Auswertung der Ergebnisse

8 Diskussion

Abbildungsverzeichnis

1 Repeat in einer Sequenz 1
2 Ein Long-Read über mehrere Contigs 2
3 verflochtene Stränge 4
4 Standardabweichung der Distanzwerte 5

Tabellenverzeichnis