Bachelor Thesis

Detektierung von Hierachien topologisch assoziierter Domänen in Hi-C Daten

Sarah Domogalla

Gutachter: Prof. Dr. Rolf Backofen
Betreuer: Joachim Wolff

Albert-Ludwigs-Universität Freiburg

Technische Fakultät

Institut für Informatik

Lehrstuhl für Bioinformatik

4. März 2020

Bearbeitungszeit

04. 12. 2019 – 04. 03. 2020

Gutachter

Prof. Dr. Rolf Backofen

Betreuer

Joachim Wolff

Erklärung

Hiermit erkläre ich, dass ich diese Abschlussarbeit selbständig verfasst habe, keine anderen als die angegebenen Quellen/Hilfsmittel verwendet habe und alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen wurden, als solche kenntlich gemacht habe. Darüber hinaus erkläre ich, dass diese Abschlussarbeit nicht, auch nicht auszugsweise, bereits für eine andere Prüfung angefertigt wurde.

Ort, Datum	Unterschrift

Zusammenfassung

Das Verständnis, wie Chromatin im Zellkern organisiert ist und wie diese Architektur funktioniert, sind wichtige Fragen in der Zellbiologie. In den letzten Jahren wurde viel in diesem Gebiet geforscht und dennoch weiß man noch relativ wenig über diese Mechanismen. Inzwischen kann man Gene identifizieren, die zu schweren Krankheiten führen oder hat die Möglichkeit bessere Medikamente zu entwickeln. Man erkennt immer mehr die Wichtigkeit in der Erforschung der grundlegenden Struktur der DNA. Eine Art, Informationen über die Architektur der DNA darzustellen, sind Hi-C-Daten. Diese können unter anderem durch die Software HiCExplorer verarbeitet und visualisiert werden. Diese Arbeit befasst sich mit der Entwicklung und Implementierung eines Algorithmus, der als Erweiterung des HiCExplorers arbeiten soll. Dieser soll die Erkennung und Differenzierung von TADs in Hi-C-Daten verbessern, indem er TADs aus verschiedenen Matrix-Auflösungen zusammenfügt und unter Einbeziehung von biologischen Grundlagen der TAD-Bildung diese auch selektieren. Ebenso werden die Beziehungen zwischen den TADs ermittelt und angemessen abgebildet.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einführung

Die DNA ist der Träger unserer Erbinformation und somit die Grundlage unseres Lebens. In ihr befinden sich die gesamten vererbbaren Informationen eines Organismus. Bei Eukaryoten ist die DNA von einigen Millionen bis einige Milliarden Basenpaare lang und ist im Zellkern als Chromosomen organisiert [1]. Der Zellkern selbst ist einige μMeter groß und die DNA ist beispielsweise bei Menschen ca. 2 Meter lang [2]. Damit so ein langes Molekül in einen winzigen Zellkern passt, muss sie auf eine bestimmte Art und Weise verpackt werden. Um die Rolle der DNA besser verstehen zu können, reicht die lineare Abfolge der DNA-Basen nicht aus. Es ist wichtig die 3D-Struktur des Genoms genauer zu untersuchen, um zu verstehen ob und wenn ja wie, die Genaktivität beeinflusst wird.

Einer der ersten wichtigen Schritte wurde 2002 mit der Entwicklung von Chromosome Confirmation Capture (3C) gelegt [3]. Im dreidimensionalen Raum können verschiedene Bereiche der DNA in räumliche Nähe zueinander gebracht werden, die im linearen Raum durch viele Nukleotide voneinander entfernt sein können. Diese Berührungspunkte der DNA mit sich selbst werden als DNA-Interaktionen bezeichnet. Die 3C-Methode erlaubt einen Einblick auf die Anzahl der Interaktionen zwischen genomischen Loci. Sie quantifiziert jedoch nur die Interaktionen zwischen einem einzigen Paar genomischer Loci (one-vs-one).

Daraufhin wurde die 3C-Methode erweitert und die Chromosome conformation capture-on-chip (4C) und später die Chromosome conformation capture carbon copy (5C) entwickelt [3]. Die 4C-Method erfasst die Interaktionen zwischen einem Locus

und allen anderen gemischten Loci (one-vs-all) und die 5C-Methode erlaubt sogar die Erfassung der Interaktionen zwischen allen Restriktionsfragmenten innerhalb eines bestimmten Bereiches (many-vs-many).

Heutzutage ist eine der gängigsten Methoden zur Untersuchung der Chromosomenstruktur die Hi-C-Technik [4], die vierte der 3C-basierenden-Methoden. Diese ermöglicht uns eine Quantifizierung der Interaktionen zwischen allen möglichen Paaren von Fragmenten gleichzeitig (all-vs-all). Es wird eine Interaktionsmatrix für die Loci des gesamten Genoms erstellt, wobei jeder Eintrag die Anzahl der Kontakte zwischen den Regionen darstellt. Diese Kontaktmatrix wird als Hi-C-Matrix bezeichnet. Dank dieser Matrix bekommen wir Informationen über die Struktur von Chromosomen und ihre Beziehung zueinander.

Da im Chromatin Regionen miteinander interagieren, die im linearen Raum sehr weit entfernt voneinander liegen können, bilden sich neue Regionen, die wir als Kompartimente bezeichnen [5]. Chromatin kann in zwei grundlegenden Zuständen auftreten: Das Chromatin ist entweder offen und somit zugänglich oder es ist geschlossen und somit unzugänglich. Abgekürzt werden diese Regionen als A-Kompartiment für das offene Chromatin und B-Kompartiment für das geschlossene Chromatin. Regionen im Kompartiment A interagieren hauptsächlich mit Regionen aus dem gleichen Kompartiment. Das gleiche gilt für Interaktionen im Kompartiment B [5].

Man kann auf den Hi-C-Matrizen nicht nur die verschiedenen Kompartimente entdecken, sondern kann auch die topologisch assoziierten Domänen (TAD) sehr gut erkennen [6]. TADs stellen Regionen dar, in denen Loci häufig miteinander interagieren. Diese sind auf der Hi-C-Matrix durch Dreiecke leicht zu erkennen, da in diesen Regionen hohe Interaktionswerte auftreten. Dies zeigt, dass Loci innerhalb eines TAD eher zu Interaktionen neigen, als zu Loci außerhalb des TAD.

Es gibt bereits Algorithmen, die TADs in Hi-C-Matrizen erkennen. Diese sind jedoch oft noch nicht sehr genau, beziehungsweise erkennen bestimmte Strukturen und Probleme nicht. Diese wären zum Beispiel die doppelte Darstellung von TADs oder wie die TADs in Beziehung zueinander stehen. Ein weiteres Problem ist die Kontrolle,

ob die berechneten TADs tatsächlich welche sind oder nicht. Zur Bewältigung dieses Problems wird in dieser Arbeit ein Algorithmus entwickelt und implementiert, um die Architektur noch besser zu erkennen und differenzieren zu können.

2 Biologischer Hintergrund

Die Organisation des Genoms im Zellkern ist nicht zufällig und beeinflusst die Genomfunktion erheblich. Darunter fallen Funktionen wie Genexpression, DNA-Replikation, Chromosomenübertragung und die Aufrechterhaltung der Genomstabilität [7]. Es gibt spezifische genomische Regionen, in denen es zu Interaktionen kommt. Solche Interaktionen können zum Beispiel Promotor-Enhancer-Wechselwirkungen sein, die wichtig für die Transkription sind, oder zufällige Schleifenbildungen darstellen [8]. Inzwischen gibt es verschiedene Methoden, um die räumliche Organisation langer Genomregionen in Zellen und deren Genominteraktionen zu untersuchen.

2.1 3C basierende Methoden

Die einzelnen 3C basierenden Technologien (3C, 4C, 5C, Hi-C) unterscheiden sich in der Art der Erkennung und dem Umfang der Interaktionen, die sie in der Lage sind zu untersuchen (siehe Abbildung 1). Alle 3C-Methoden beginnen mit den gleichen vier Schritten:

1. Formaldehyd-Fixierung: diese Vernetzung von Chromatin führt dazu, dass die Stellen, an denen Interaktionen stattfinden, fixiert werden [10]

2. Spaltung von Chromatin durch Restriktionsenzym: dabei wird das Genom ungefähr alle 4000 Basenpaare zerschnitten [7]

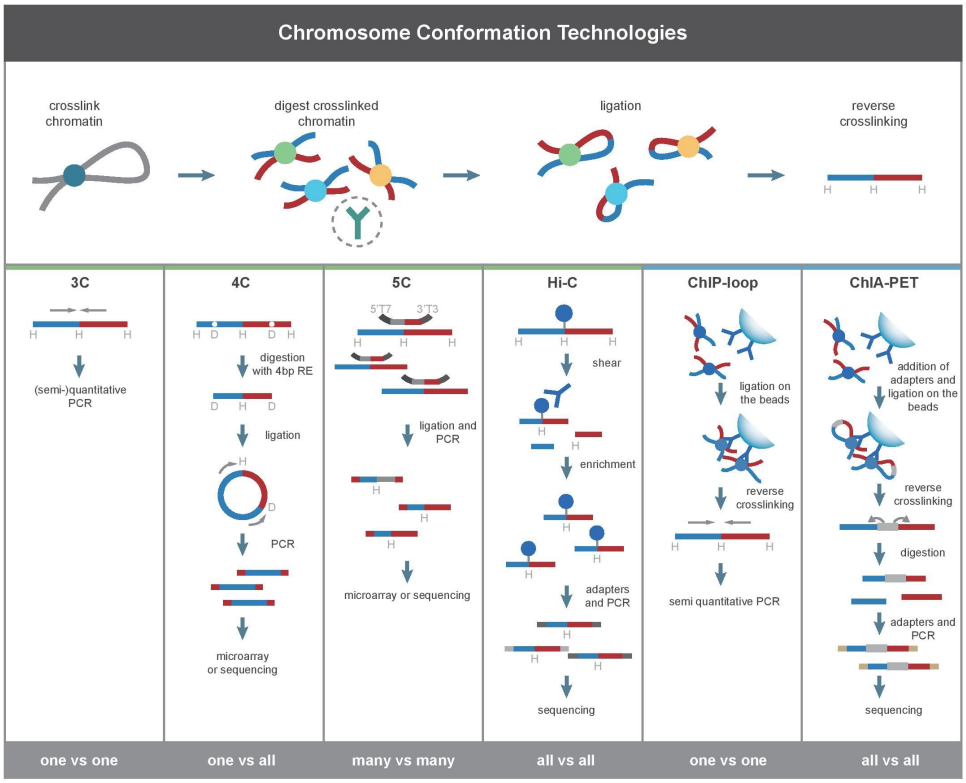

Abbildung 1: Vergleich zwischen 3C und seinen abgeleiteten Methoden. Quelle: [9]

3. Ligation (Verknüpfung) unter verdünnten Bedingungen: dadurch wird die Ligation zwischen interagierenden Fragmenten gegenüber der Verknüpfung zwischen nicht vernetzten Fragmenten bevorzugt [7]

4. Nachweis und Quantifizierung der Ligationsprodukten: zur Bestimmung der bei der Ligation des vernetzten Chromatins erfassten Interaktionsfrequenzen [11]

2.1.1 3C (one-vs-one)

Die 3C-Methode arbeitet mit der Polymerasen-Kettenreaktion (PCR) und wird verwendet, um die relative Häufigkeit von Interaktionsfrequenzen zwischen einem Paar ausgewählter genomischer Loci zu ermitteln. Der Nachteil der 3C-Methode liegt

darin, dass sich die Abfrage immer nur auf ein einziges Paar beschränkt. Das Ergebnis liefert die Häufigkeit, mit der zwei Restriktionsfragmente ligiert werden, und diese bilden ein Maß für die Häufigkeit, mit der sie im Zellkern interagieren [12].
Basierend auf dieser Methode entwickelten sich weitere 3C-basierende-Verfahren, um einerseits den Durchsatz zu erhöhen und andererseits eine unvoreingenommene genomweite Analyse zu ermöglichen [7].

2.1.2 4C (one-vs-all)

Die Chromosome conformation capture-on-chip (4C) ermöglicht die Erfassung einer Interaktion zwischen einem Locus und allen anderen genomischen Loci. Im Gegensatz zu der 3C-Methode beinhaltet dieses Verfahren einen zweiten Ligationsschritt, um selbstzirkuläre DNA-Fragmente zu erzeugen. Diese werden für die inverse PCR benötigt, die eine Vermehrung einer unbekannten Sequenz ermöglicht. Dies ist möglich, da durch den zweiten Ligationsschritt an dieser unbekannten Sequenz eine bekannte Sequenz gebunden ist, die man mittels der inversen PCR mehrfach duplizieren kann. Dadurch benötigt man im Gegensatz zu der 3C und der 5C Methode keine Vorkenntnisse über die miteinander interagierenden Regionen [8].

2.1.3 5C (many-vs-many)

Die Chromosome conformation capture carbon copy (5C) ermöglicht eine Erkennung der Interaktionen aller Restriktionsfragmenten innerhalb eines bestimmten Bereichs, also innerhalb einer gegebenen Genomdomäne [11]. Dieser Bereich ist typischerweise nicht größer als 1 MB. Der Nachteil der 5C Methode ist, dass sie nur eine relativ geringe Abdeckung hat. Für eine genomweite Interaktion ist dieser Ansatz zu aufwendig, da man Millionen von 5C-Primern benötigen würde [8].

2.1.4 Hi-C (all-vs-all)

Hi-C verwendet eine Hochdurchsatz-Sequenzierung und ist damit in der Lage, die Architektur ganzer Genome zu untersuchen. Hi-C verbindet die Proximity-basierte Ligation mit paralleler Sequenzierung [4]. Am Ende der vier ursprünglichen Schritte, die alle 3C-basierenden-Methoden gleich haben, hat man DNA-Proben, die Ligationsprodukte enthalten. Diese bestehen aus Fragmenten, die sich ursprünglich in räumlicher Nähe befanden. Diese werden an der Verbindungsstelle mit Biotin markiert und durch Scheren der DNA und Selektion der Biotin-haltigen Fragmente wird eine Hi-C-Bibliothek erstellt. Diese Bibliothek wird dann unter Verwendung einer massiv parallelen DNA-Sequenzierung analysiert und dabei wird ein Katalog mit interagierenden Fragmenten erzeugt [4]. Daraus können dann genomweite Chromatinkontaktkarten erzeugt werden, die die Chromosomenorganisation in einer Zellpopulation widerspiegeln [13].

2.2 Hi-C-Matrix

Mit Hi-C hat man die Möglichkeit Hi-C-Interaktionsmatrizen des menschlichen Genoms darzustellen und anhand dieser kann man die unterschiedlichen Chromosomenterritorien erkennen und die räumliche Nähe von Chromatin feststellen.

2.2.1 A/B-Kompartimente

Wenn man sich das Genom im großen Maßstab anschaut, kann man es in zwei voneinander abgeschlossene Arten von Chromatin unterteilen: das aktive A-Kompartiment und das inaktive B-Kompartiment. Das aktive Kompartiment mit dem aktiven Chromatin und das inaktiven B-Kompartiment mit dem kompakten Heterochromatin [5]. Wenn man sich jedoch die lokalen Muster in einer Kontaktmatrix genau anschaut,

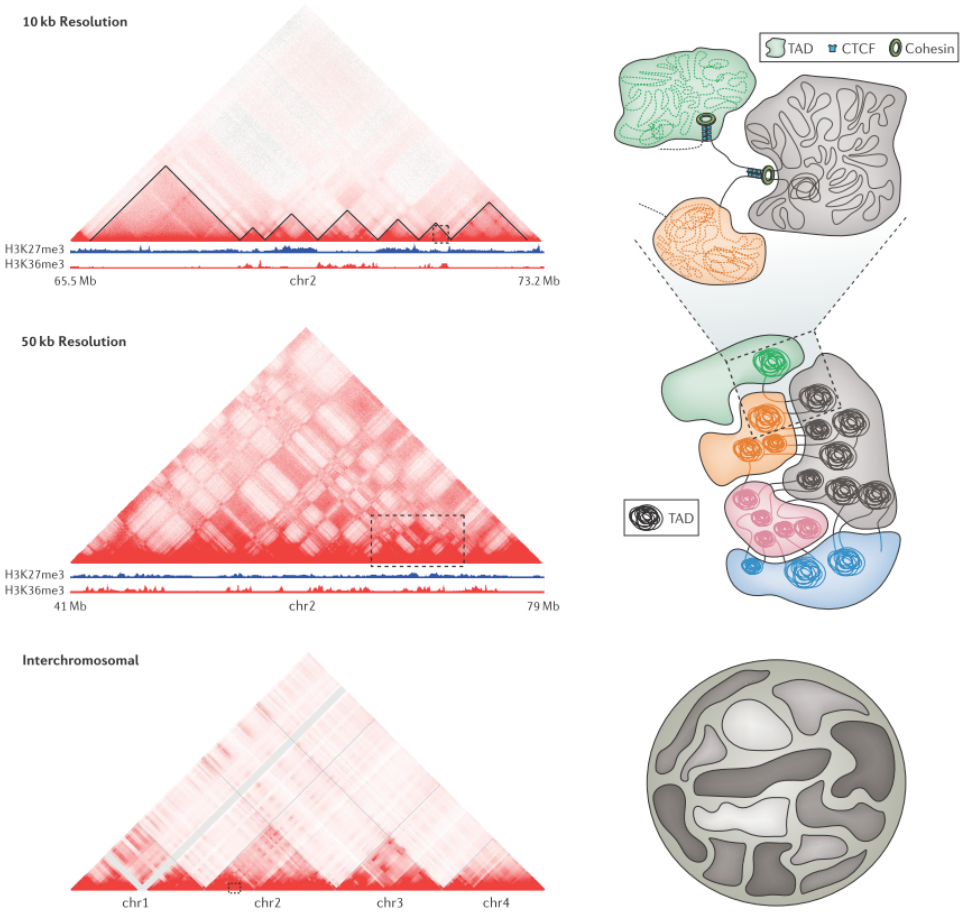

Abbildung 2: Hierarchische Organisation der Chromatinstruktur
Links sieht man die Unterschiede zwischen den Auflösungen der gleichen Hi-C-Matrix und rechts die allgemeine Organisation von Chromatin. Quelle: [3]

fallen einem stattdessen besonders die topologisch assoziierenden Domänen (TADs) als Schlüsselmerkmale auf (siehe Abbildung 2) [6].

2.2.2 Topologisch assoziierte Domäne

Topologisch assoziierte Domänen (TADs) (siehe Abbildung 3) sind räumlich abgegrenzte Domänen, die sich durch eine hohe Kontakthäufigkeit innerhalb dieser Domäne kennzeichnen. Gleichzeitig sind die Interaktionen zwischen den verschiedenen TADs

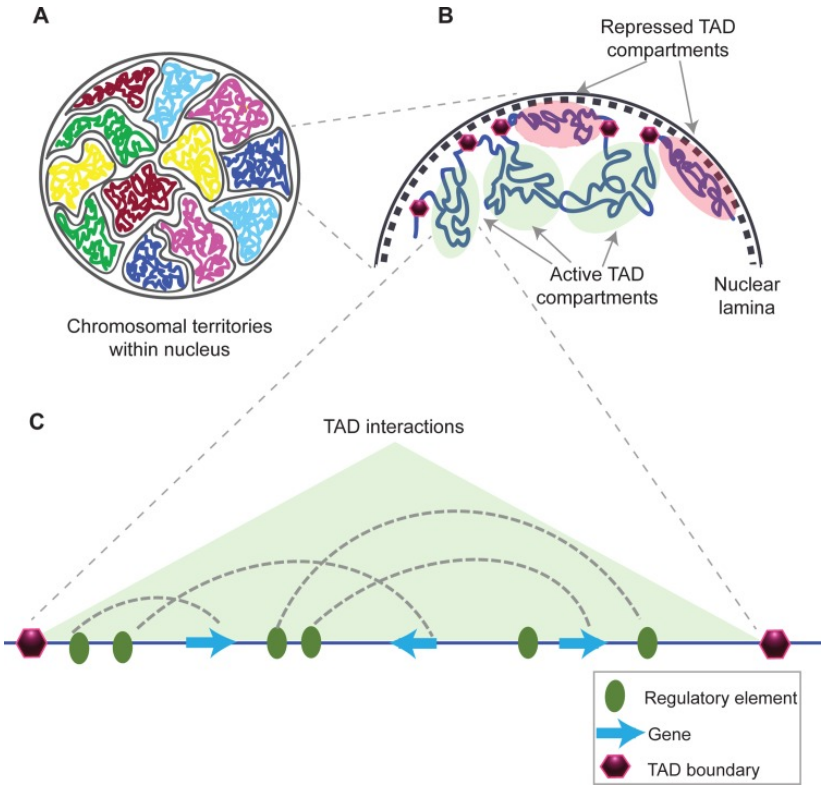

Abbildung 3: **a)** Chromosomengebiete innerhalb des Zellkerns. **b)** Jedes Chromosom ist in topologisch assoziierte Domänen (TAD) unterteilt. **c)** Ein Beispiel für ein TAD mit mehreren Wechselwirkungen zwischen regulatorischen Elementen und Genen. Quelle: [15]

nur schwach vorhanden (Abbildung 2) [6]. Man erkennt TADs auf Hi-C Matrizen sehr deutlich, da sie als sehr abgegrenzte Dreiecke auffallen. Diese Domänen wurden bei vielen Arten entdeckt und in ihnen finden wichtige Prozesse der Genregulation statt [5]. Eine Störung der TAD-Strukturen durch Veränderung ihrer Grenzen kann zu einer Fehlexpression der Gene führen, was Entwicklungsstörungen und Krebs auslösen kann [5]. Ihre Größe kann von Tausenden bis zu Millionen von DNA-Basen variieren. Bei Menschen liegt die Durchschnittsgröße bei 880 kB [5]. Es ist inzwischen bekannt, dass eine Reihe von Proteinen an der TAD Bildung beteiligt sind. Eine besondere Rolle bei Säugetieren spielen vor allem das Protein CTCF und der Proteinkomplex Cohesin [14].

2.2.3 Cohesin und CTCF

Um die einzelnen TADs voneinander unterscheiden zu können, muss man wissen, was die strukturellen Merkmale der Chromatin-Organisation sind. Ein Hauptmerkmal der meisten TAD-Grenzen von Säugetieren ist das Vorhandensein von CTCF zusammen mit dem Proteinkomplex Cohesin. Diese Grenzen sind als „Eckpunkte“ auf den Hi-C-Karten zu erkennen und sind an starken Interaktionen beteiligt [5].
Die Schlaufenbildung beginnt damit, dass sich ein Cohesin-Komplex an die DNA anheftet (Abbildung 4a). Cohesin besteht aus zwei verbundenen, ringförmigen Untereinheiten und die DNA verbindet sich mit dem Cohesin, indem sie durch ein Loch hinein- und durch das andere wieder heraustritt (Abbildung 4b). Nun wird die Schlaufe vergrößert, indem die einzelnen Cohesin-Ringe in entgegengesetzter Richtung der DNA entlanggleiten und erst stoppen, wenn sie an ein CTCF Molekül stoßen (Abbildung 4c).

Die CTCF Moleküle sind an die DNA gebunden und können zwei Richtungen aufweisen. Die Vergrößerung der Schleife endet erst, wenn der Cohesin-Ring über ein CTCF gleitet, dass zum inneren der DNA-Schlaufe hindeutet. Ansonsten wird das CTCF-Molekül ignoriert, der Cohesin-Ring gleitet darüber hinweg und die Schlaufe wächst weiter. Die Schlaufe ist abgeschlossen, wenn beide Cohesin-Ringe ein nach innen gerichtetes CTCF Molekül erreicht haben [14].

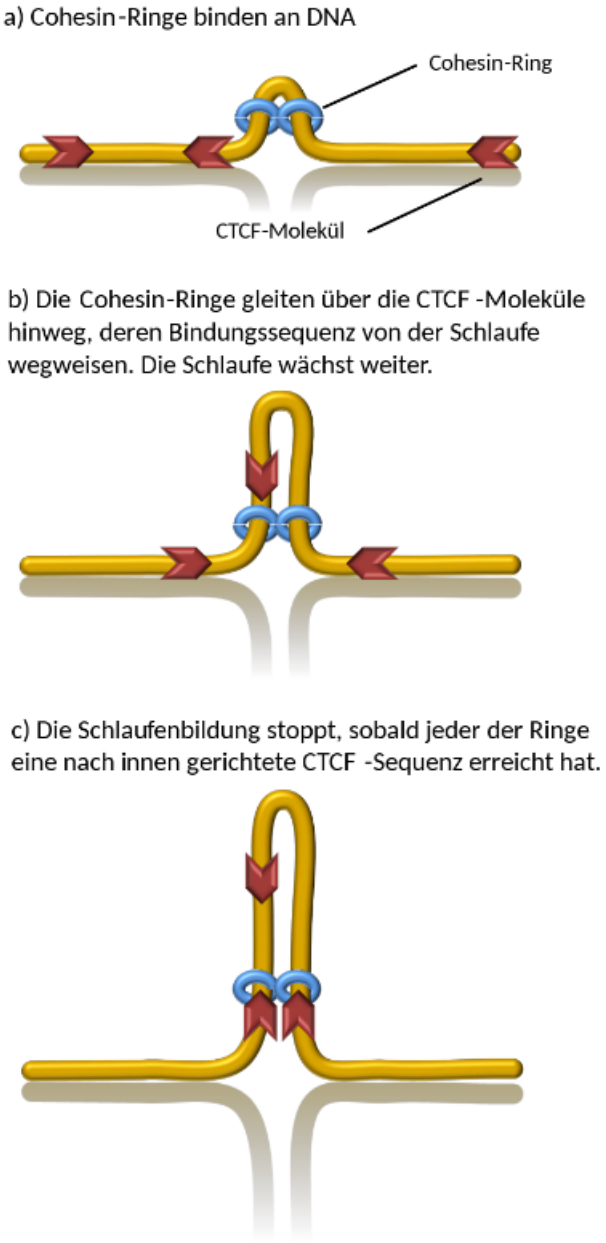

Abbildung 4: Bildung einer DNA-Schlaufe durch Cohesin und CTCF. Quelle: [14]

3 Verwendete Algorithmen

Um die Beziehung unter den TADs besser verstehen zu können, benötigt man erst einmal grobe Informationen über diese. Besonders interessant sind die Positionen der TADs und vor allem wo ihre Grenzen liegen. Ein Algorithmus, der uns diese Informationen liefert, ist unter anderem der hicFindTADs-Algorithmus aus der Software-Suite HiCExplorer [16].

3.1 hicFindTADs

Dieser Algorithmus identifiziert TAD-Grenzen, indem er mit einem bestimmten Score arbeitet, dem TAD-Separation-Score. Dieser Score berechnet einen TAD-Trennungswert um dann lokale Minima zu identifizieren, die TAD-Grenzen anzeigen. Ein Vorteil von diesem Algorithmus ist, dass er zusätzlich zu den TAD-Grenzen auch den TAD-Separation-Score zurück gibt. Dieser Score ist eine nützliche Information über die Stärke, der einzelnen Grenzen, und der Kontaktdichte innerhalb der TADs [16].
Für die Berechnungen benötigt der Algorithmus eine korrigierte Hi-C-Kontaktmatrix. Als Ausgabe bekommt man mehrere Dateien mit Informationen über diese Matrix.

Chrom	Start	End	Name	Score	Strand	ThickStart	ThickEnd	ItemRGB
chr1	1850000	2600000	ID_1	-1.110295918048	.	1850000	2600000	31,120,180
chr1	1900000	3400000	ID_2	-0.769206442711	.	1900000	3400000	31,120,180
chr1	2600000	3350000	ID_3	-0.896680703526	.	2600000	3350000	51,160,44
chr1	3350000	3800000	ID_4	-0.620943888002	.	3350000	3800000	31,120,180
chr1	3400000	6000000	ID_5	-0.389226278516	.	3400000	6000000	51,160,44
chr1	3800000	4700000	ID_6	-0.711443824609	.	3800000	4700000	51,160,44
chr1	4700000	5500000	ID_7	-0.045102389240	.	4700000	5500000	31,120,180
chr1	5500000	6000000	ID_8	-0.452283298292	.	5500000	6000000	51,160,44
chr1	6000000	6750000	ID_9	-0.683079372844	.	6000000	6750000	31,120,180
chr1	6000000	6700000	ID_10	-0.890486947679	.	6000000	6700000	31,120,180
chr1	6700000	8000000	ID_11	-0.505583355621	.	6700000	8000000	51,160,44
chr1	6750000	7050000	ID_12	-0.634443136119	.	6750000	7050000	51,160,44
chr1	7050000	7700000	ID_13	-0.039455147398	.	7050000	7700000	31,120,180

Abbildung 5: Ausschnitt der domain.bed Datei einer 50 kB Matrix

Als Ausgabedateien bekommt man folgende:

- **Boundaries Datei:** Darin sind die TAD-Grenzpositionen gespeichert
- **Boundaries und Score Datei:** Zusätzlich zu den TAD-Grenzpositionen wird auch der TAD-Score gespeichert
- **Domain Datei:** In dieser Datei befinden sich die TAD-Positionen
- **Score Datei:** Diese Datei beinhaltet den TAD-Separation-Score
- **Z-Score-Matrix**: Wird für die Berechnung des TAD-Separation-Score verwendet

Eine Datei davon ist die Domains-Datei, die für den hier vorgestellten Algorithmus als Grundlage dient. In Abbildung 5 sieht man einen Ausschnitt aus einer Domains-Datei einer 50 kB Matrix. Wichtig sind hierbei die ersten 5 Spalten. Diese sagen aus, auf welchem Chromosom das TAD sich befindet, die linke und rechte Grenze des TADs, seine ID und seinen Score.

4 Problemstellung

Das Problem bei bisherigen Algorithmen ist, dass sie als Eingabe meistens nur eine Matrix nehmen. Ist die Matrix groß und detailliert, werden vor allem die kleinen TADs erkannt. Das liegt daran, dass bei detaillierteren Matrizen die Bereiche, in denen die Basenpaare zusammengefügt werden, kleiner sind. Wenn wir zum Beispiel eine 5kB Matrix haben, sind jeweils 5.000 Basenpaare zusammengefügt. Hat man im Gegensatz dazu, eine weniger detailreiche 1MB Matrix, sind jeweils 1.000.000 Basenpaare zusammengefasst. Die Schritte sind also viel größer und so können TADs, die dazwischen liegen, übersehen werden. Es ist auch keine Lösung einfach eine große detailreiche Matrix zu nehmen, da diese oft die großen TADs nicht erkennen, weil die Schritte zu klein sind. Je nachdem wie groß die Matrizen sind, werden unterschiedliche TADs erkannt (siehe Abbildung 6). Deshalb ist für ein besseres Verständnis über den Aufbau der TADs wichtig, nicht nur die TADs aus einer Matrix-Auflösung angezeigt zu bekommen.
Schaut man sich die Hi-C-Matrix genauer an, erkennt man deutlich, dass die TADs auch innerlich verschiedene Bereiche aufweisen. Das bedeutet, dass Algorithmen, die nur mit einer Eingabematrix arbeiten, kein optimales Ergebnis liefern können. Um die verschiedenen Ebenen darstellen zu können, braucht man eine Vereinigung der TADs von verschiedenen Auflösungen.

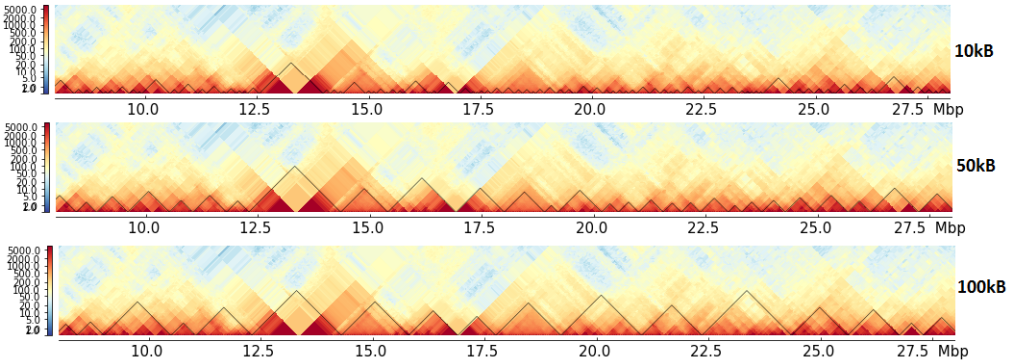

Abbildung 6: Darstellung der unterschiedlichen TAD-Erkennungen zwischen den verschiedenen Hi-C-Auflösungen

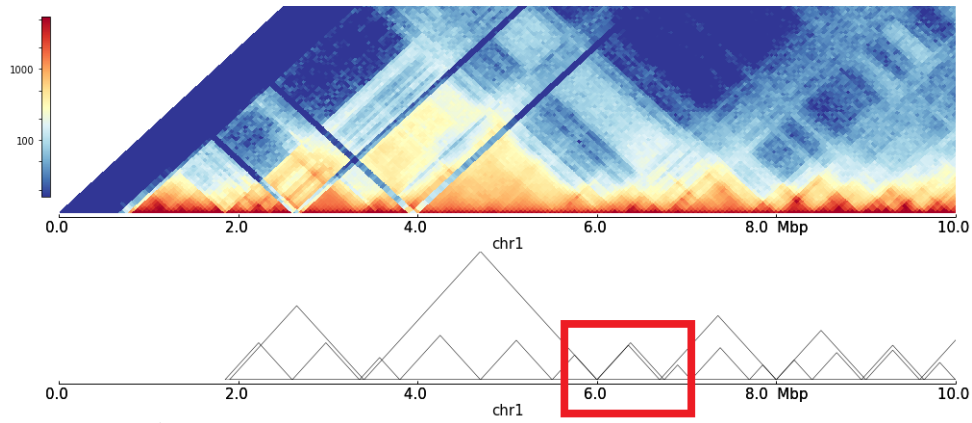

Abbildung 7: 50 kB Matrix mit doppeltem TAD

4.1 Unterscheidung von TADs

Wenn man TADs aus verschiedenen Auflösungen zusammenfügt, können aus mehreren Matrizen die gleichen TADs angezeigt werden. Wenn man jedoch berücksichtigt, dass die TADs aus verschiedenen Matrixgrößen stammen und somit nicht zu 100% identisch sind, kann es zu einem Problem werden, diese zu erkennen.

Schaut man sich den markierten TAD in Abbildung 7 an, sieht man das Problem: diese zwei TADs sind ähnlich, aber ihre Grenzen sind nicht zu 100% gleich. Es könnten zwei oder ein TAD sein. Dieses Problem tritt vor allem bei zwei TADs auf, die sich nur um wenige Basenpaare unterscheiden. Bei diesem Problem hilft der Parameter

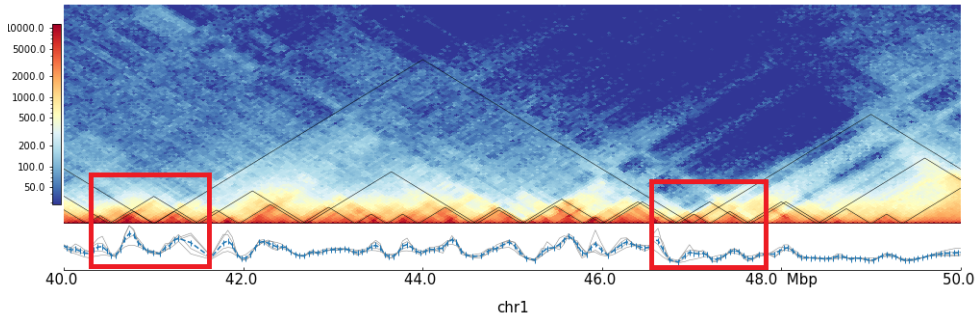

Abbildung 8: Beispiel von zwei überlappenden TADs in einer Hi-C-Matrix

value. Dieser gibt an, wie viele Basenpaare zwischen den Grenzen zweier TADs liegen müssen, damit diese als zwei eigenständige Bereiche erkannt werden. Dabei reicht es aus, dass nur eine Grenze diesen Mindestabstand haben muss. Dieser Parameter wird in Basenpaaren angegeben.

4.2 Überlappende TADs

Im Gegensatz zu Problem 4.1, bei dem es darum geht, zwei TADs zu unterscheiden, deren Fläche größtenteils übereinstimmt, geht es bei diesem Problem in den meisten Fällen um das Gegenteil. In einem zweidimensionalen Raum, was die Hi-C-Daten darstellen, ist es schwierig zu unterscheiden, welche TADs einander nur überlappen und welche TADs tatsächlich Untereinheiten von großen TADs darstellen. In Beispiel 8 sieht man zwei TADs, die sich jeweils mit ihren Nachbar-TADs links und rechts überschneiden. Je geringer die Fläche ist, mit der sie sich überschneiden, desto leichter kann man sagen, dass diese zwei TADs in keiner Beziehung zueinander stehen und sich wahrscheinlich nur überlappen. Schaut man sich jedoch den rechten markierten TAD in Abbildung 8 an, ist der Überschneidungsbereich mit dem rechten großen TAD sehr groß. Ist also der markierte TAD eine Untereinheit von dem sehr großen TAD im rechten Drittel?
In dem hier vorgestellten Algorithmus hat man die Möglichkeit, mit einem optionalen

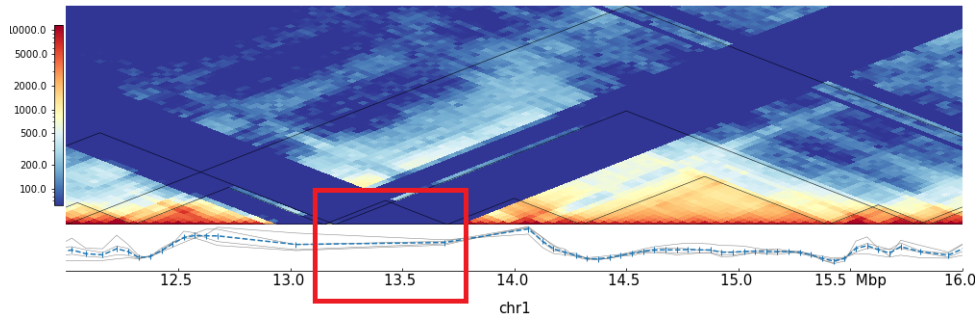

Abbildung 9: Beispiel eines auffälligen Bereiches in einer Hi-C-Matrix

Parameter *percent* selbst zu bestimmen, ab wann zwei TADs in Verbindung zueinander stehen und wann nicht. Dieser Parameter gibt an, wie viel Prozent der Flächen die TADs mindestens gemeinsam haben müssen, damit sie in irgendeiner Weise in Beziehung zueinander stehen. Gibt man zum Beispiel percent = 0.05 ein, so müssen sie mindestens 5% gleich sein.

4.3 Falsche TADs

Ein weiteres Problem ist, dass der Algorithmus aus 3.1 TADs an Stellen bestimmt, an denen gar keine zu sehen sind. Wenn man sich den gleichen Ausschnitt einer Matrix mit verschiedenen Auflösungen anschaut, werden oft TADs eingezeichnet, wo man keine erkennt. Eine Möglichkeit echte TADs von falschen zu unterscheiden, ist CTCF (siehe Section 2.2.3). Dieses Protein lagert sich an den TAD-Grenzen an und gibt uns damit eine Möglichkeit die Grenzen auf ihre Korrektheit zu überprüfen. Um diese überprüfen zu können, wird eine separate Datei benötigt, in denen die CTCF-Peaks gespeichert sind.
Es kann auch passieren, dass man in Hi-C-Matrizen sehr auffällige Bereiche entdeckt, wie man deutlich in Abbildung 9 sieht. Die deutlich herausstechenden dunkelblauen Streifen zeigen deutlich an, dass in diesen Gebieten keine Interaktionen vorkommen und trotzdem wurden durch den Algorithmus an dieser Stelle zwei TADs zugeordnet.

5 Implementierung

Das Ziel dieser Arbeit war es einen Algorithmus zu implementieren, der die TAD-Detektierung unter Beachtung der Probleme aus 4 verbessert. Der daraus resultierende Algorithmus *hicMergeDomains* vereint Domain-Dateien aus verschiedenen Auflösungen miteinander, um die verschiedenen TADs zu vereinen. Darüber hinaus überprüft er jeden einzelnen TAD auf seine Korrektheit, optimalerweise mit Einbeziehung von CTCF, und ermittelt die Beziehungen zwischen den TADs und stellt sie in angemessener Form dar.

5.1 Eingabedateien

Die Grundlage des Algorithmus ist das Vereinen der Domain-Dateien. Damit das geschehen kann, müssen mindestens zwei bis beliebig viele Domain-Dateien als Eingabe mitgegeben werden. Die erste Domain-Datei wird mittels dem Parameter $domain1$ definiert und alle weiteren Domain-Dateien werden über den Parameter $domainList$ übergeben. Damit der Algorithmus im weiteren Verlauf das best möglichste Ergebnis liefern kann, sollte darauf geachtet werden, auch die passende CTCF-Datei anzugeben. Diese kann als zusätzlicher Parameter $ctcfFile$ angegeben werden.

5.2 hicMergeDomains

Der Algorithmus 1 ist in Python geschrieben und kann in drei große Abschnitte unterteilt werden. Der erste Schritt ist das Einlesen der Eingabedateien und die CTCF-Sortierung. Der zweite Schritt ist das Vereinen aller Domain-Dateien und der letzte Schritt wäre die Ermittlung der Beziehungsverhältnisse und die Erstellung des Beziehungsbaums.

5.2.1 Einlesen der Dateien und CTCF-Sortierung

Zuerst werden alle Domain-Dateien eingelesen und in Listen abgespeichert (Algorithmus 1 Zeile 0). Das Gleiche passiert mit der CTCF-Datei. Damit man die TAD-Grenzen der Dateien mit der CTCF-Datei vergleichen kann, muss jeweils die CTCF-Datei auf die gleiche Auflösung, wie die jeweilige Domain-Datei, angepasst werden (Algorithmus 1 Zeile 4). Das erreicht man, indem man alle CTCF-Peaks aus einem bestimmten Bereich zusammenfasst. Zum Beispiel würde man bei einer 50 kB Domain-Datei alle Peaks innerhalb von 50.000 Basenpaaren zusammen nehmen. Wenn es mindestens einen Peak in diesem Bereich gibt, werden alle Grenzen akzeptiert, die in diesem Bereich liegen. Falls man jedoch mit sehr großen Auflösungen arbeitet, zum Beispiel mit einer 1 MB Matrix und findet, dass ein CTCF-Peak pro 1.000.000 Basenpaare zu wenig ist, hat man die Möglichkeit mittels dem optionalen Parameter *minPeak* einen anderen Wert einzustellen. Der Standard-Wert ist ein Peak bei jeder Auflösung.
Für jede Domain-Datei wird die CTCF-Datei auf die gleiche Auflösung angepasst (Algorithmus 1 Zeile 4). Danach wird für jede TAD-Grenze in der Domain-Datei überprüft, ob an dieser Position ein CTCF-Peak existiert oder nicht. Falls an dieser Position kein eindeutiger Peak zu sehen ist, wird im zweiten Schritt überprüft, ob es an dieser Stelle zu einem Anstieg von dem TAD-Score kommt, also ob an dieser Grenze verstärkte Interaktionen stattfinden. Wenn weder ein CTCF-Peak noch ein

Algorithm 1 hicMergeDomains

Require: –domain1 –domainList

```
read all domain files and save them in lists
if CTCF file is given then
    for each domains file from domainList and for domain1 do
        adapt CTCF file to domain resolution considering the minPeak
        all TADs that don't match any CTCF or Score Peak are deleted
        the resulting cleaned list is saved
    end for
end if
mergedList = cleaned List of domain1
for each domains of domainList do
    function mergeDomains(domain, mergedList)
        cleaned list is merged with the mergeList
        return mergedList
    end function
end for
function createRelationsshipList(mergedDomainList)
    create new List RelationList
    for every TAD in mergedList as TAD1 do
        for every TAD in mergedList as TAD2 do
            if TAD1 and TAD2 not at the same Chromosome then
                Break
            end if
            if TAD1 and TAD2 are overlapping then
                save the relationship into RelationList
            end if
            if TAD1 or TAD2 is an internal TAD then
                save the relationship into RelationList
            end if
        end for
    end for
    return RelationList
end function
save the RelationList in a File
function createRelationshipTree(RelationList)
    create a relationship tree for each chromosome
    return save relationship tree
end function
```

verstärkter TAD-Score vorliegen, wird der TAD entfernt (Algorithmus 1 Zeile 5).
Als Zwischenergebnis erhält man aus jeder Domains-Datei unter Einfluss von CTCF eine bereinigte Liste .

5.2.2 Domain-Dateien zusammenfügen

Im zweiten großen Schritt werden alle einzelnen Domain-Dateien mit ihren unterschiedlichen Auflösungen zu einer Liste verschmolzen. Zu Beginn wird als Merge-Liste die Domain1 Liste verwendet (Algorithmus 1 Zeile 9). Nun wird jede weitere Domain mit der Merge-Liste verschmolzen (Algorithmus 1 Zeile 12) und dieses Ergebnis wird wiederum mit der nächsten Domain-Liste vereint(Algorithmus 1 Zeile 13), da das Ergebnis als neue Merge-Liste dient. Daraus folgt, dass diese Funktion bei n Domain-Listen n-1 mal ausgeführt wird.

5.2.3 Beziehungen zwischen den TADs

Im letzten Schritt geht es darum, die Beziehungen zwischen den TADs zu ermitteln und darzustellen. Zuerst werden die einzelnen TADs miteinander verglichen und ihre Beziehungen zueinander erkannt. Dabei wird festgestellt, welcher TAD *Child* beziehungsweise *Parent* ist. In Abbildung 10 wäre TAD1 *Child* von TAD2, da TAD1 in TAD2 liegt und somit ist TAD2 auch gleichzeitig *Parent* von TAD1.

Um die Beziehungen richtig ermitteln zu können, wird in Algorithmus 1 Zeile 18 mit zwei Schleifen gearbeitet. Als Erstes wird überprüft, ob die beiden TADs noch auf dem gleichen Chromosom liegen (Algorithmus 1 Zeile 20). Wenn dies der Fall ist, wird überprüft, ob es sich bei den beiden TADs um überlappende TADs handelt (Algorithmus 1 Zeile 23).
Dafür benötigt man die Prozentzahl, die man als Parameter aus Problem 4.2 angeben kann. Es werden nun die Anzahl der Basenpaare berechnet, die die Prozentangabe für die jeweiligen TADs ausmachen. Dann wird mittels Addition überprüft, ob die

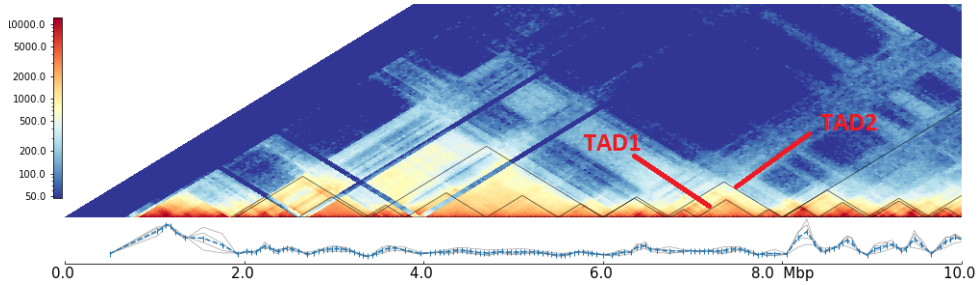

Abbildung 10: In diesem Beispiel werden die Begrifflichkeiten *Child* und *Parent* bei TADs erklärt. TAD1 stellt hierbei *Child* und TAD2 *Parent* dar.

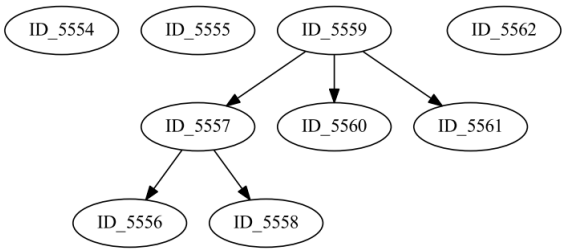

Abbildung 11: Ausschnitt aus dem Beziehungsbaum von Chromosom X

TADs die Mindestfläche überschreiten. Falls das der Fall ist, wird der kleine TAD als "Child"für den größeren TAD gemerkt.

Danach wird überprüft, ob einer der beiden TADs komplett im zweiten TAD liegt, also es wird überprüft, ob TAD1 mit seinen Grenzen komplett in TAD2 liegt. Falls das der Fall ist, wird auch diese Beziehung hinzugefügt (Algorithmus 1 Zeile 26).

Diese Beziehungen werden während der Schleifen (Algorithmus 1 Zeile 18) in Listen gespeichert und später in einer Datei abgespeichert (Algorithmus 1 Zeile 33).

In der zweiten Hälfte dieses Schrittes, werden die Beziehungen als Baum dargestellt. Damit es übersichtlich bleibt, werden die Chromosomen einzeln dargestellt (Algorithmus 1 Zeile 35) und in einem Ordner gesammelt gespeichert. In Abbildung 11 sieht man einen kleinen Ausschnitt aus dem Beziehungs-Graphen von Chromosom X.

6 Resultat

Durch die Kombination des Zusammenfügens verschiedener Domain-Dateien und Überprüfung der einzelnen TADs mit dem CTCF-Wert und des TAD-Scores, bekommt man eine bereinigte TAD-Liste geliefert. Um zu überprüfen, wie erfolgreich der Algorithmus aus 5.2 arbeitet, werden verschiedene Parameter getestet und die Ergebnisse miteinander verglichen.

6.1 Grundlage

Um die Ergebnisse vergleichen zu können, schauen wir uns einen Ausschnitt der Humanzelle GM12878 mit verschiedenen Parametern genau an. Wir konzentrieren uns bei dem Vergleich auf Chromosom 1 und hauptsächlich auf den Bereich von 8-18MB. Als Vergleichsbasis dient uns die Abbildung 12. Diese Matrix zeigt drei zusammengefasste Matrizen mit den Auflösungen von 10 kB, 50 kB und 100 kB an, die jedoch noch nicht bereinigt ist. Sie zeigt alle TADs an, die durch den Algorithmus hicFindTADs (siehe Kapitel 3.1) für die einzelnen Auflösungen berechnet wurden und mittels hicMergeDomains aus 5.2 zusammengefügt wurden, jedoch ohne irgendwelche TADs zu löschen.

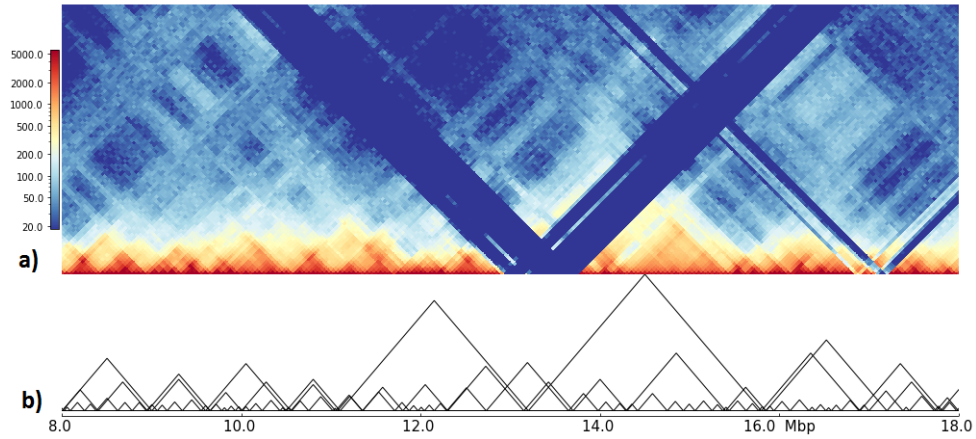

Abbildung 12: **a)** Dieser Ausschnitt zeigt die Hi-C-Matrix der Humanzelle GM12878 aus dem Bereich 8-18 MB in der Auflösung 50 kB an. **b)** Hier sieht man die passenden TADs zusammengefügt aus 3 Matrix-Auflösungen (10 kB, 50 kB und 100 kB) des hicFindTADs-Algorithmus

6.2 CTCF-Parameter

Die größte Erweiterung des hicMergeDomain-Algorithmus aus 5.2 ist die Überprüfung der TADs auf CTCF-Peaks und damit das Aussortieren der TADs, die keinen CTCF-Peak aufweisen können. Dieser Schritt löst das Problem aus 4.3. In Abbildung 13 sieht man deutlich, dass in diesem Bereich drei TADs entfernt wurden. Insgesamt wurden auf dem kompletten Chromosom 1 der Humanzelle GM12878 68 TADs entfernt. Davon waren 62 TADs aus der 10 kB Matrix, 4 TADs aus der 50 kB Matrix und 2 TADs aus der 100 kB Matrix.

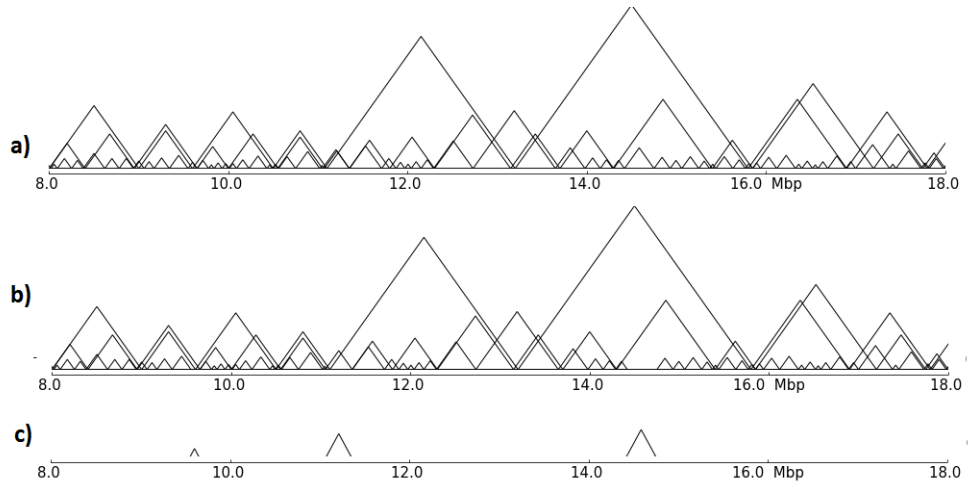

Abbildung 13: a) zeigt die unbereinigten TADs einer 50 kB Matrix **b)** zeigt die bereinigten TADs unter Beachtung des CTCF-Peaks **c)** verdeutlicht, welche TADs von a) nach b) entfernt wurden, da an diesen Grenzen kein CTCF-Peak vorgefunden wurde

6.2.1 MinPeak-Parameter

Im hicMergeDomain-Algorithmus hat man die Möglichkeit die Anzahl der *minPeak* selbst einzustellen (siehe Kapitel 5.2.1). Mit *minPeak* definiert man, wie viele CTCF-Peaks pro Abschnitt existieren müssen, damit in diesem Abschnitt eine TAD-Grenze erlaubt wird. Dies hilft besonders, wenn man mit sehr großen Matrizen arbeitet, da zum Beispiel ein CTCF-Peak auf einem Abschnitt von 1 MB nicht sehr aussagekräftig ist. In solchen Fällen, könnte man den *minPeak* höher stellen. Man sollte jedoch beachten, den *minPeak* nicht zu hoch zu stellen. Was die Auswahl des *minPeaks* für Auswirkungen haben kann, zeigt die Tabelle 1. Die Ergebnisse sagen aus, wie viel Prozent der CTCF-Peaks gelöscht werden, bei unterschiedlichen *minPeak*-Einstellungen. Diese Tabelle sagt nicht aus, wie viele TADs am Ende gelöscht werden, sondern nur wie viele Peaks durch das Anpassen der Auflösung, unter Beachtung des "minPeaks", verloren gehen. Bei minPeak = 1 wird natürlich jeder CTCF-Peak beachtet. Bei minPeak = 2 sieht man den Anstieg der Prozentzahlen deutlich, er steigt nämlich von 0% auf 65% an. minPeak = 2 unter der Auflösung 10 kB sagt aus, dass die CTCF

Datei auf die Auflösung von 10 kB angepasst wird und alle Peaks die innerhalb von 10 kB Schritten liegen, zusammengefasst werden.
Erreicht ein 10 kB Schritt nicht die Mindest-Peak-Anzahl (minPeak), so wird dieser Bereich gelöscht und TADs, deren Grenze in solchen Bereichen liegen würden, würden nicht akzeptiert werden. Man sieht auch, dass die Anzahl der gelöschten Peaks geringer werden, je größer die Auflösung ist. Das liegt daran, dass je größer die Auflösung ist, desto größer auch die Schritte und desto größer die Wahrscheinlichkeit das die Mindestanzahl als Peaks erreicht wird.

MinPeak / Auslösung	1	2	3	4	5	6	7	8
10 kB	0%	65%	88%	95%	97%	98%	98%	98%
50 kB	0%	28%	48%	68%	79%	89%	92%	94%
100 kB	0%	15%	30%	41%	55%	68%	77%	86%
500 kB	0%	0%	10%	18%	23%	28%	33%	38%

Tabelle 1: Verhalten des minPeak Parameters bei verschiedener Matrix-Auflösung
In dieser Tabelle wird gezeigt, wie viel Prozent der Peaks der kompletten CTCF-Datei gelöscht werden, bei unterschiedlicher minPeak-Einstellungen.

6.3 Value-Parameter

Der dritte Parameter, der bei der TAD-Darstellung eine Rolle spielt, ist der *value* Parameter (siehe Kapitel 4.1), der bei der Unterscheidung von Doppelten TADs helfen soll. Bei der Wahl dieses Parameters, sollte man sich an den Auflösungen der Matrizen orientieren, die man als Eingabe gegeben hat. Diese Matrizen spiegeln auch die Zwischenschritte zwischen den TADs wieder. Der kleinste Abstand zwischen zwei TADs spiegelt die kleine Auflösung der Eingabe-Matrizen wieder. In Abbildung 14 wurde value = 50000 gewählt, also die mittlere Auflösung der drei Eingabe-Matrizen.

6.4 Beziehungen zwischen den TADs

Eine weitere Besonderheit des Algorithmus hicMergeDomains ist die Darstellung der Beziehungen zwischen den TADs. Hier trifft man auf das Problem 4.2. Dieses Problem befasst sich damit, ab wann zwei TADs miteinander in Verbindung stehen, also miteinander interagieren und wann sie sich nur überlappen. Hierbei hilft der Parameter *percent*.

6.4.1 Percent-Parameter

Dieser Parameter hat keinen Einfluss darauf, ob TADs gelöscht werden oder nicht. Deswegen sieht man auch keinen Unterschied in der TAD-Darstellung. Dieser Parameter definiert, ab wie viel Prozent Flächenunterschied zwei TADs nicht mehr in Beziehung zueinander stehen. Wählt man percent = 0.5, werden alle TADs in Verbindung miteinander gestellt, die zu mindestens 51% über die gleiche Fläche verfügen. Als Beispiel nehmen wir den Ausschnitt von TADs, den man in Abbildung 15 sieht.

Diese sind zum besseren Verständnis der Beziehungen, mit ihren ID-Bezeichnungen versehen. Wählt man percent = 0, werden alle TADs in Beziehung zueinander gestellt, die sich in irgendeiner Weise berühren oder überlappen. Die Ausgabe sieht man in Abbildung 16.

In Abbildung 16 sieht man anhand von ID_89 deutlich, dass die Beziehungen zwischen den TADs nicht transitiv sind. ID_86 -> ID_87 und ID_87 -> ID_89, aber das gilt nicht für ID_86 -> ID_89.
Wählt man nun percent = 0.5, sieht der Beziehungs-Baum in Abbildung 17 sehr viel übersichtlicher aus. Was bei Abbildung 17 unter anderem auffällt ist, dass ID_86 in keiner Beziehung mehr zu den anderen TADs steht. Das liegt an der Parameter-Auswahl, da dieser TAD mit keinem anderen TAD mindestens zu 50% die gleiche Fläche teilt.

6.5 Resultat

Man sieht an den einzelnen Ergebnissen die Wirkung, die man mit dem Algorithmus hicMergeDomains erreichen kann. Das Endergebnis wird in Abbildung 18 verdeutlicht. Der Algorithmus fügt TADs aus verschiedenen Auflösungen zusammen, bereinigt diese zuverlässig auf erweiterter biologischer Grundlage und lässt genug Spielraum, um den Algorithmus auf seine individuellen Vorhaben anzupassen.

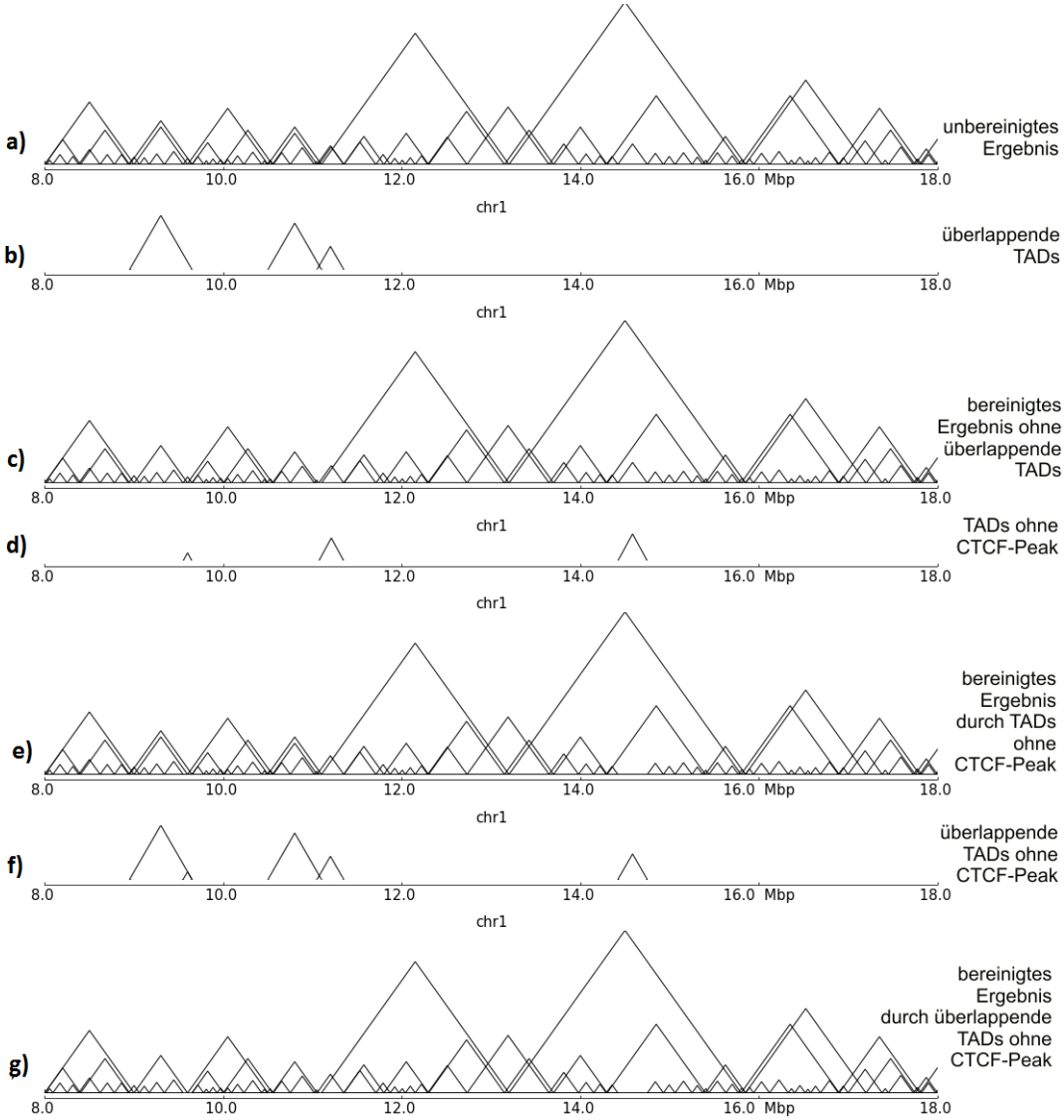

Abbildung 14: **a)** unbereinigte TAD-Darstellung ohne Paramter als Vergleichsbasis **b)** überlappende TADs aus a), die durch den Parameter *value* entfernt wurden **c)** bereinigte TADs aus a) ohne die überlappenden TADs aus b) **d)** TADs ohne CTCF-Peak **e)** bereinigte TADs aus a) ohne TADs aus d), da keine CTCF-Peaks vorliegen **f)** überlappende TADs ohne CTCF-Peaks (vereinte Darstellung aus b) und d)) **g)** bereinigtes Ergebnis ohne überlappende TADs und unter Beachtung der CTCF-Peaks

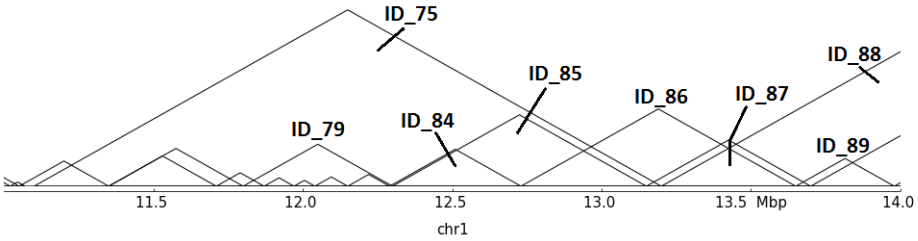

Abbildung 15: Gezeigt wird ein Ausschnitt von TADs mit ihren ID-Nummern

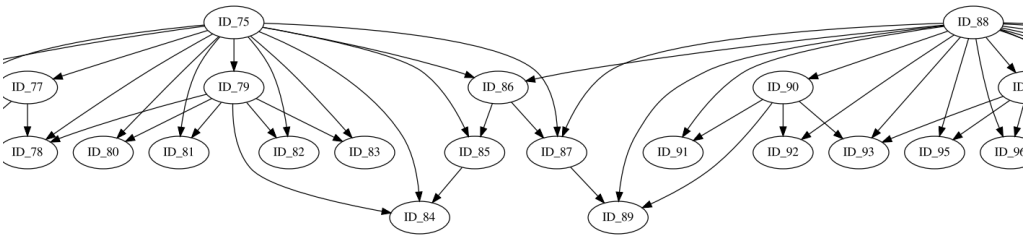

Abbildung 16: Gezeigt wird ein Ausschnitt der Beziehungen zwischen TADs mit percent = 0

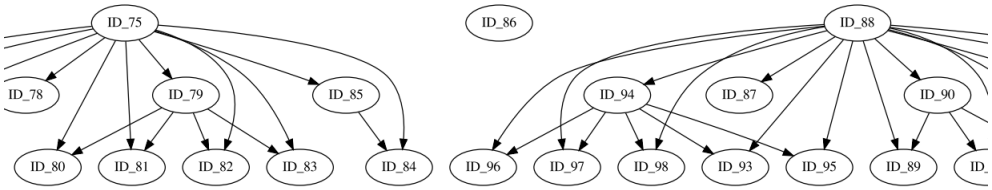

Abbildung 17: Gezeigt wird ein Ausschnitt der Beziehungen zwischen TADs mit percent = 0.5

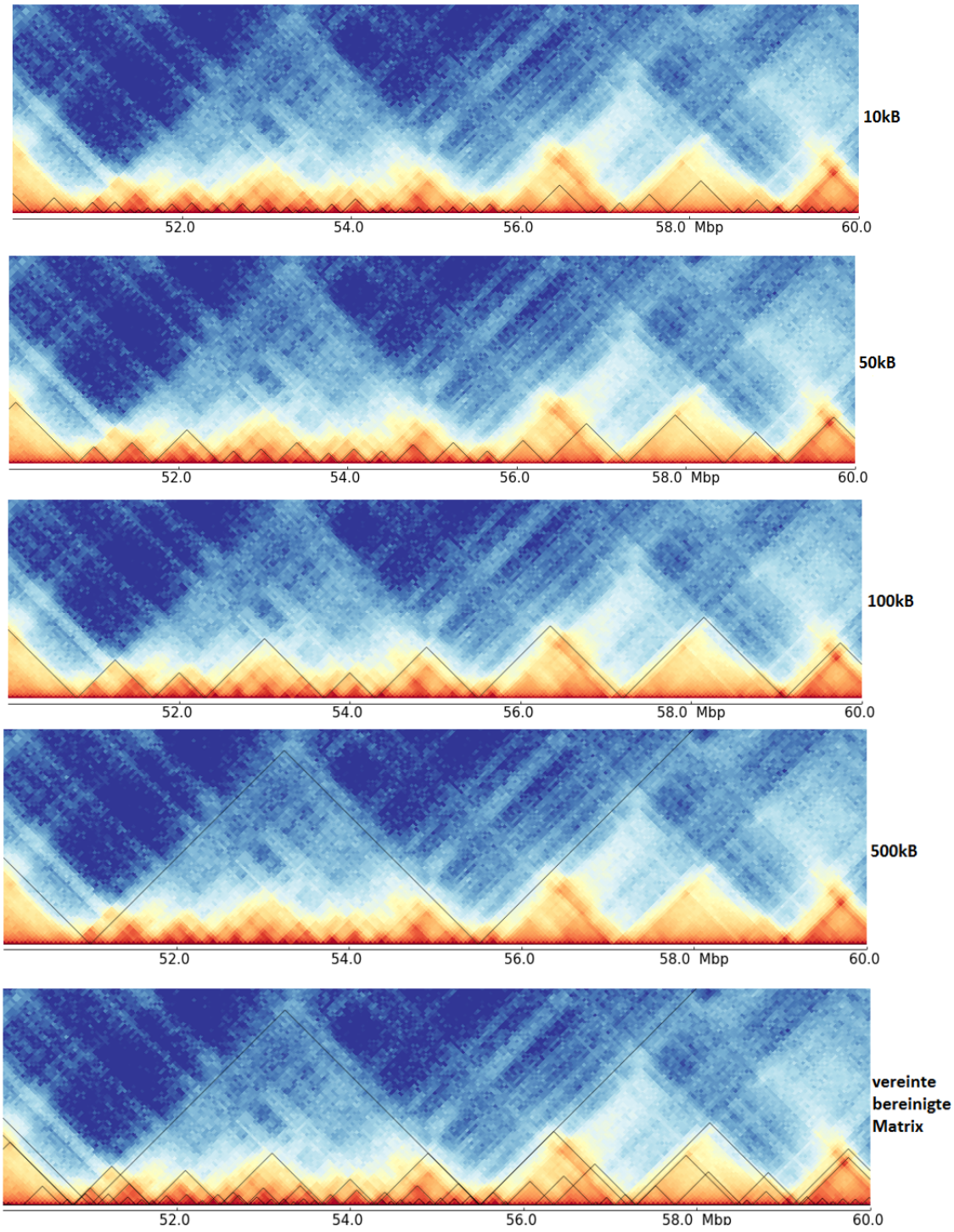

Abbildung 18: Die ersten vier Matrizen von oben, zeigen die TADs in den Auflösungen 10 kB, 50 kB, 100 kB und 500 kB. Ganz unten sieht man die zusammengefügte bereinigte Variante des hier vorgestellten Algorithmus, aus den 4 Auflösungen darüber.

7 Diskussion

Die Experimente aus dem vorherigen Kapitel haben gezeigt, dass der Algorithmus die Domain-Dateien und damit die TADs nach unseren Vorstellungen vereinen und bereinigen kann. Bei der Implementierung und Entwicklung des Algorithmus (auf Github zu finden, siehe Referenz [1]) gab es keine größeren Schwierigkeiten. Dennoch stößt man an manchen Stellen an die Grenzen des aktuellen Wissensstandes der Organisation des Genoms, vor allem bei der dreidimensionalen Organisation der TADs.

Ein Beispiel dafür findet man bei dem Problem (Kapitel 4.1), wenn zwei sehr ähnliche TADs in der vereinten Domain-Liste entstehen. In dieser Arbeit geht man davon aus, dass zwei sehr ähnliche TADs ein Fehler in dem Zusammenfügen verschiedener Auflösungen von Matrizen sein müssen, indem zwei gleiche TADs aus verschiedenen Auflösungen in die Liste hinzugefügt wurden. Jedoch kann es sich hierbei auch um zwei TADs handeln, die in keiner Beziehung zueinander stehen und sich nur durch die zweidimensionale Darstellung der Hi-C-Matrix sich überlappen. Da man noch keine Möglichkeit hat, dies zu überprüfen, behandelt der Algorithmus solche sehr ähnlichen TADs als ein TAD. Das gleiche Problem besteht bei den TADs, die deutlicher voneinander getrennt sind, aber man dennoch eine Beziehung zwischen den TADs vermuten könnte (Kapitel 4.2). Abschließend kann man sagen, dass der Algorithmus für den jetzigen Stand der Technik, zufriedenstellend arbeitet und für zukünftige Arbeiten ein umfangreicheres Wissen um die Organisation des Genoms nützlich wären, um das Bereinigen der TADs noch sinnvoller zu gestalten.

[1] https://github.com/SarahMaria27/HiCExplorer.git

Literaturverzeichnis

[1] "Aufbau von dna und rna." https://www.amboss.com/de/wissen/Aufbau_von_DNA_und_RNA. [Online; accessed 11.02.2020].

[2] "How long is your DNA?." https://www.sciencefocus.com/the-human-body/how-long-is-your-dna/. [Online; accessed 28.02.2020].

[3] B. Bonev and G. Cavalli, "Organization and function of the 3d genome," *Nature Reviews Genetics*, vol. 17, no. 11, p. 661, 2016.

[4] E. Lieberman-Aiden, N. L. Van Berkum, L. Williams, M. Imakaev, T. Ragoczy, A. Telling, I. Amit, B. R. Lajoie, P. J. Sabo, M. O. Dorschner, *et al.*, "Comprehensive mapping of long-range interactions reveals folding principles of the human genome," *science*, vol. 326, no. 5950, pp. 289–293, 2009.

[5] Q. Szabo, F. Bantignies, and G. Cavalli, "Principles of genome folding into topologically associating domains," *Science advances*, vol. 5, no. 4, p. eaaw1668, 2019.

[6] K. Pal, M. Forcato, and F. Ferrari, "Hi-c analysis: from data generation to integration," *Biophysical reviews*, vol. 11, no. 1, pp. 67–78, 2019.

[7] N. Naumova, E. M. Smith, Y. Zhan, and J. Dekker, "Analysis of long-range chromatin interactions using chromosome conformation capture," *Methods*, vol. 58, no. 3, pp. 192–203, 2012.

[8] "Chromosome conformation capture." https://de.wikipedia.org/wiki/Chromosome_conformation_capture. [Online; accessed 11.02.2020].

[9] G. Li, L. Cai, H. Chang, P. Hong, Q. Zhou, E. V. Kulakova, N. A. Kolchanov, and Y. Ruan, "Chromatin interaction analysis with paired-end tag (chia-pet) sequencing technology and application," *BMC genomics*, vol. 15, no. 12, p. S11, 2014.

[10] A. Gavrilov, E. Eivazova, I. Pirozhkova, M. Lipinski, S. Razin, and Y. Vassetzky, "Chromosome conformation capture (from 3c to 5c) and its chip-based modification," in *Chromatin Immunoprecipitation Assays*, pp. 171–188, Springer, 2009.

[11] O. Hakim and T. Misteli, "Snapshot: chromosome conformation capture," *Cell*, vol. 148, no. 5, pp. 1068–e1, 2012.

[12] J. Dostie, T. A. Richmond, R. A. Arnaout, R. R. Selzer, W. L. Lee, T. A. Honan, E. D. Rubio, A. Krumm, J. Lamb, C. Nusbaum, *et al.*, "Chromosome conformation capture carbon copy (5c): a massively parallel solution for mapping interactions between genomic elements," *Genome research*, vol. 16, no. 10, pp. 1299–1309, 2006.

[13] A. D. Schmitt, M. Hu, and B. Ren, "Genome-wide mapping and analysis of chromosome architecture," *Nature reviews Molecular cell biology*, vol. 17, no. 12, p. 743, 2016.

[14] "Topologically associating domain." https://de.wikipedia.org/wiki/Topologically_associating_domain. [Online; accessed 11.02.2020].

[15] N. Matharu and N. Ahituv, "Minor loops in major folds: enhancer–promoter looping, chromatin restructuring, and their association with transcriptional regulation and disease," *PLoS genetics*, vol. 11, no. 12, 2015.

[16] J. Wolff, V. Bhardwaj, S. Nothjunge, G. Richard, G. Renschler, R. Gilsbach, T. Manke, R. Backofen, F. Ramírez, and B. A. Grüning, "Galaxy hicexplorer: a web server for reproducible hi-c data analysis, quality control and visualization," *Nucleic acids research*, vol. 46, no. W1, pp. W11–W16, 2018.