Freie Universität Berlin

Institut für Mathematik und Informatik

Annotation und Interpretation von Varianten und Polymorphismen im humanen Genom

Marten Jäger

2019

Dissertation zur Erlangung des Grades
eines Doktors der Naturwissenschaften (Dr. rer. nat.)
am Fachbereich Mathematik und Informatik
der Freien Universität Berlin

1. Gutachter:	Prof. Dr. Peter N. Robinson
2. Gutachter:	Prof. Dr. Knut Reinert

Tag der Disputation: 27. Januar 2020

Selbstständigkeitserklärung

Ich erkläre gegenüber der Freien Universität Berlin, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe.

Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch bei keiner anderen Universität als Prüfungsleistung eingereicht und ist auch noch nicht veröffentlicht.

Berlin, den 10. September 2019

D

Vorwort

Die *Desoxyribonukleinsäure* (DNA), als Grundbaustein des Lebens auf der Erde wurde erstmals 1869 durch Friedrich Miescher in Form von Nuklein isoliert. Mieschers Schüler Richard Altmann erkannte zwanzig Jahre später die für die Namensgebung verantwortliche saure Eigenschaft (Altmann, 1889), wohingegen die Bedeutung der DNA für die Weitergabe erblicher Informationen erst 1943 durch Oswald Avery (Avery, MacLeod und McCarty, 1944) erfolgte.

Die Aufklärung der helikalen, doppelsträngigen Struktur der DNA durch Watson, Crick (J. D. Watson H. R. und Crick, 1953) und die um ihre Forschung betrogene Rosalind Franklin (J. Watson, 2012) bildete letztendlich die Grundlage, um die Kenntnisse der genetischen Vererbung, die durch Mendel (Mendel, 1866) begründet wurden, auch im molekularbiologischen und medizinischen Bereich zu untersuchen und interpretierbar zu machen.

Seit den siebziger Jahren des neunzehnten Jahrhunderts gab es erste Methoden zur DNA-Sequenzierung, wobei die nach Sanger (Sanger und Coulson, 1975) die verbreitetste war. Mit diesem Werkzeug in der Hand war es möglich, die Sequenzfolgen von ganzen Genen und kleineren Genomen zu sequenzieren. Zusammen mit den Kenntnissen der genetischen Vererbung bildeten diese Techniken die Grundlage der Genetik, wie wir sie heute kennen. Ein Bestreben bestand immer darin, die Methoden der DNA-Sequenzierung weiterzuentwickeln, um damit die Aufklärung der DNA-Sequenzen noch effizienter und dem medizinischen Bereich zugänglich zu machen.

Diese Ambitionen führten 1990 zur Begründung des *Humanen Genompro-*

jekts (HGP), einem internationalen Forschungsprojekt mit dem Ziel, die humane Referenzstruktur aufzuklären, beziehungsweise die Sequenzabfolge der 3,2 Milliarden Basenpaare der DNA des humanen Genoms zu bestimmen und alle Gene zu identifizieren. Ursprünglich noch mit veranschlagten Kosten von einem US-Dollar pro Basenpaar, konnte im Februar 2001 die erste „Arbeitsversion“ des humanen Genoms veröffentlicht werden und gilt seit 2003 offiziell als *entschlüsselt*.
Die rapide fallenden Sequenzierkosten pro Base in den letzten fünfzehn Jahren, haben den Preis für die Analyse eines Exoms, also die gezielte Anreicherung und Sequenzierung der bekannten, genkodierenden Abschnitte des Genoms, auf unter 500 € gesenkt. Damit ist *Whole Exome Sequencing* (WES) nicht mehr nur für die Forschung interessant, sondern in einen preislichen Bereich gerückt, der sie auch für die routinemäßige Diagnostik attraktiv macht. Durchschnittlich werden in einem Exom 20 000 bis 30 000 Varianten im Vergleich zum humanen Referenzgenom gefunden. Dies macht eine computergestützte Auswertung und Interpretation dieser Varianten notwendig. Der ersten Teil der Dissertation beschäftigt sich mit dieser Problematik. Zu diesem Zweck wurde eine Java Bibliothek namens *Jannovar* implementiert (Jäger u. a., 2014), welche zur Annotation der Varianten eingesetzt werden kann. Dies ermöglicht das Filtern und Priorisieren der gefundenen Varianten (Smedley, Jacobsen u. a., 2015) und durch die Automatisierung eine Erleichterung in der klinischen Arbeit. Folgende Arbeit wurde dazu veröffentlicht:

1. Marten Jäger, Kai Wang, Sebastian Bauer, Damian Smedley, Peter Krawitz, and Peter N Robinson. Jannovar: a java library for exome annotation. *Human mutation*, 35:548–555, May 2014.

In der Zwischenzeit (Frühjahr 2017) sind erstmals die reinen Sequenzierkosten für *Whole Genome Sequencing* (WGS) auf 1 000 € gefallen und die Bearbeitungszeit im Labor konnte auf wenige Tage reduziert werden, wodurch auch WGS für die klinische Diagnostik immer interessanter wird. Der nied-

rige Preis und die damit einhergehende Menge an sequenzierten Individuen zeigt deutlich, dass es keine simple „humane Referenz“ gibt, sondern dass wesentlich mehr (strukturelle) Variabilität im humanen Genom existiert, als zu Beginn des HGP angenommen. Diese Unterschiede werden insbesondere deutlich, wenn man Individuen aus unterschiedlichen Populationen betrachtet.
Das *Genome Reference Consortium* (GRC) veröffentlicht in unregelmäßigen Abständen Aktualisierungen des humanen Referenzgenoms. Seit der Version GRCh37 im Jahr 2009 wurde hierbei auch die strukturelle Variabilität in Sinne von alternativen Sequenzbereichen berücksichtigt. Der zweite Teil der Arbeit beschäftigt sich mit der Anwendung und Interpretation der graphenartigen Strukturrepräsentation im aktuellen humanen Referenzgenom, der Version GRCh38 von 2013. Darin untersuche ich, ob sich aus den genomischen Varianten Rückschlüsse auf die repräsentativste Referenz schließen lassen und welche Bedeutung diese im medizinisch interpretierbaren Kontext haben können. Folgende Arbeit wurde hierzu veröffentlicht:

2. Marten Jäger, Max Schubach, Tomasz Zemojtel, Knut Reinert, Deanna M Church, Peter N Robinson. Alternate-locus aware variant calling in whole genome sequencing. *Genome Medicine*, 8:130, 2016

Eine Auflistung der Projekte, welche die oben genannten Arbeiten anwenden und zu einem bedeutenden Bestandteil machen, ist in den folgenden Publikationen zu finden. Es handelt sich hierbei um kollaborative Projekte, bei denen mein Beitrag von geringerem Umfang war.

3. Tomasz Zemojtel, Sebastian Köhler, Luisa Mackenroth, Marten Jäger, Jochen Hecht, Peter Krawitz, Luitgard Graul-Neumann, Sandra Doelken, Nadja Ehmke, Malte Spielmann, Nancy Christine Oien, Michal R Schweiger, Ulrike Krüger, Götz Frommer, Björn Fischer, Uwe Kornak, Ricarda Flöttmann, Amin Ardeshirdavani, Yves Moreau, Suzanna E Lewis, Melissa Haendel, Damian Smedley, Denise Horn, Stefan Mundlos, and Peter N Robinson. Effective diagnosis of genetic disease by

computational phenotype analysis of the disease-associated genome. *Science translational medicine*, 6:252ra123, Sep 2014.

4. Damian Smedley, Julius O B Jacobsen, Marten Jäger, Sebastian Köhler, Manuel Holtgrewe, Max Schubach, Enrico Siragusa, Tomasz Zemojtel, Orion J Buske, Nicole L Washington, William P Bone, Melissa A Haendel, and Peter N Robinson. Next-generation diagnostics and disease-gene discovery with the exomiser. *Nature protocols*, 10:2004–2015, Dec 2015.

5. Damian Smedley, Max Schubach, Julius O B Jacobsen, Sebastian Köhler, Tomasz Zemojtel, Malte Spielmann, Marten Jäger, Harry Hochheiser, Nicole L Washington, Julie A McMurry, Melissa A Haendel, Christopher J Mungall, Suzanna E Lewis, Tudor Groza, Giorgio Valentini, and Peter N Robinson. A whole-genome analysis framework for effective identification of pathogenic regulatory variants in mendelian disease. *American journal of human genetics*, 99:595–606, Sep 2016.

Zusätzlich zu den in dieser Dissertation beleuchteten Arbeiten habe ich Beiträge zu weiteren Projekten geleistet, welche einen anderen Themenschwerpunkt hatten. Im Folgenden findet sich eine Liste der Projekte, welche abgeschlossen sind und in Fachzeitschriften veröffentlicht wurden.

6. Johannes Grünhagen, Raghu Bhushan, Elisa Degenkolbe, Marten Jäger, Petra Knaus, Stefan Mundlos, Peter N Robinson, and Claus-Eric Ott. MiR-497∼195 cluster microRNAs regulate osteoblast differentiation by targeting bmp signaling. *Journal of bone and mineral research : the official journal of the American Society for Bone and Mineral Research*, 30:796-808, May 2015.

7. Daniel M Ibrahim, Peter Hansen, Christian Rödelsperger, Asita C Stiege, Sandra C Doelken, Denise Horn, Marten Jäger, Catrin Janetzki, Peter Krawitz, Gundula Leschik, Florian Wagner, Till Scheuer,

Mareen Schmidt-von Kegler, Petra Seemann, Bernd Timmermann, Peter N Robinson, Stefan Mundlos, and Jochen Hecht. Distinct global shifts in genomic binding profiles of limb malformation-associated hoxd13 mutations. *Genome research*, 23:2091–2102, Dec 2013.

8. Tobias Penzkofer, Marten Jäger, Marek Figlerowicz, Richard Badge, Stefan Mundlos, Peter N Robinson, and Tomasz Zemojtel. L1Base 2: more retrotransposition-active LINE-1s, more mammalian genomes. *Nucleic Acids Research*, 45(D1):D68-D73, Jan 2017;

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einleitung

1.1 Hintergrund

Die Aufklärung der doppelsträngigen helikalen Struktur der DNA durch Francis Crick, James Watson und Rosalind Franklin (J. D. Watson H. R. und Crick, 1953; J. Watson, 2012) bildet die Grundlage für unser heutiges Verständnis des Aufbaus des humanen Genoms. Mit diesem Grundverständnis wurden Genetikern und Molekularbiologen neue Möglichkeiten zur näheren Erforschung der DNA und deren Funktionen eröffnet und sie bleiben weiterhin Ziel intensiver Forschung.

1.2 Aufbau des Genoms: DNA, RNA, Proteine

Ein eukaryotes Genom definiert sich durch die Gesamtheit aller DNA im Nukleus einer Zelle. Diese ist im humanen Genom in einem diploiden Satz aus 23 Chromosomenpaaren angeordnet, wobei sich das Paar für die Geschlechtschromosomen bei Frau (XX) und Mann (XY) unterscheidet. Eine Kopie für jedes Chromosomenpaar stammt jeweils von einem Elternteil. Jedes Chromosom besteht aus zwei komplementären Molekülen aus einer Abfolge von vier molekularen Bausteinen, den Nukleotiden (Adenin, Thymin,

Guanin und Cytosin). Die Nukleotide werden häufig entsprechend den Anfangsbuchstaben ihrer Namen abgekürzt. Durch die molekularen Adhäsionskräfte der Wasserstoffbrückenbindung zwischen den jeweils komplementären Nukleotiden der zwei Einzelstränge hybridisieren diese zu der charakteristischen Doppelhelix.

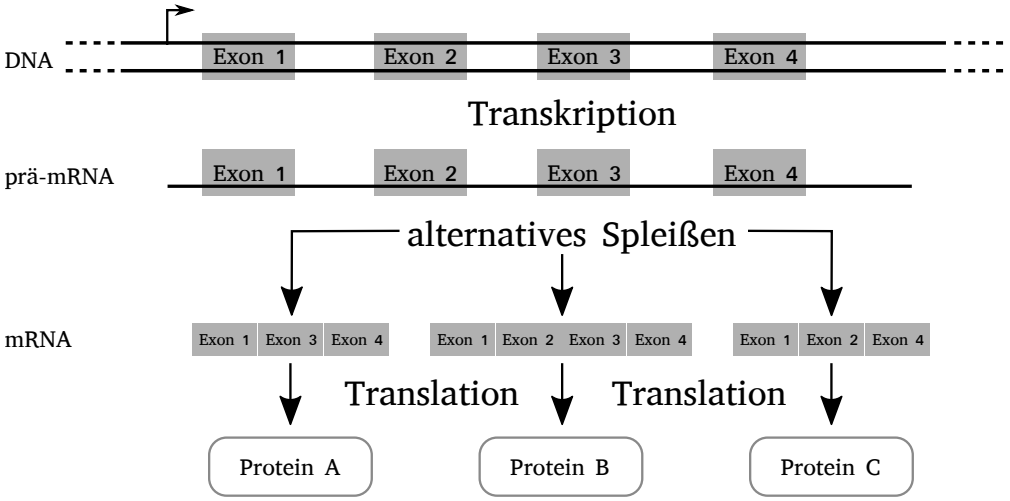


Abbildung 1.1: **Zentrales Dogma der Genexpression.** Abbildung nach Guttmacher und F. S. Collins (2002).

Molekularbiologisch wichtige funktionelle Einheiten auf der DNA sind die Gene. Die exonischen Abschnitte der Gene machen etwa zwei Prozent der gesamten humanen DNA aus. Das zentrale Dogma der Genexpression besagt, dass ein Gen translatiert wird, also eine einzelsträngige Kopie des kompletten gen-codierenden DNA-Abschnitts mit Ribonukleotiden angefertigt wird, die prä-RNA. Diese prä-RNA wird weiter prozessiert und die Introns, die nicht-codierenden Abschnitte des Gens, werden ausgeschnitten. Dieser Spleißen (engl.: *splicing*) genannte Schritt kann in alternativen Spleißprodukten resultieren (Blencowe, 2006). Die Reihenfolge der Exons, den protein-codierenden Abschnitten, ist stetig, jedoch können sie bei Spleißvarianten entfallen (siehe auch Abbildung 1.2). Hierdurch ist eine höhere

Anzahl von mRNA-Varianten möglich, als es Gene gibt. Die so entstandene mRNA dient anschließend dem Ribosom als Vorlage zur Synthese eines Proteins. Drei aufeinanderfolgende Ribonukleotide der mRNA bilden bei der Translation, der Übersetzung der mRNA in ein Protein, das jeweilige Codon für eine der 20 natürlichen Aminosäuren[1]. Es ist offensichtlich, dass $4^3 = 64$ Kombinationen deutlich mehr Möglichkeiten darstellen, als es Aminosäuren gibt. Die Erklärung hierfür liegt darin, dass eine Aminosäure durch mehrere Codons codiert sein kann. Dies betrifft vor allem das dritte Ribonukleotid, welches damit nicht so spezifisch ist und daher auch als Wobble[2]-Base betitelt wird. Zu den codierenden kommen noch spezielle Codons, welche den Translationsstart (`AUG`) und den Abbruch der Translation (`UAA`,`UAG`,`UGA`,) markieren. Eine Veränderung in der Nukleotidsequenzabfolge der DNA hat dementsprechend auch eine direkte Auswirkung auf die Aminosäuresequenz des prozessierten Proteins. Ein einfacher Austausch eines Nukleotids in der codierenden DNA-Sequenz hat demzufolge einen Austausch der Aminosäure zur Folge. Bei einer neutralen Veränderung (Wobble-Base) hat dies keine Auswirkung. Fehlen oder kommen Nukleotide hinzu, wird aus dem gewünschten Codonleseraster (Frame) in ein alternatives Leseraster (Frameshift) gewechselt.

1.3 DNA-Sequenzierung

Unter Desoxyribonukleinsäure (DNA)-Sequenzierung versteht man die Aufschlüsselung der genauen Abfolge der Nukleotide eines DNA-Moleküls, um die biologische Information, die durch die Abfolge der Nukleotide codiert ist, zu erhalten.

[1] Daher auch der Name Codon – für die Codierung der Aminosäure durch jeweils drei Ribonukleotide.

[2] wobble - Englisch für wackeln/schwanken.

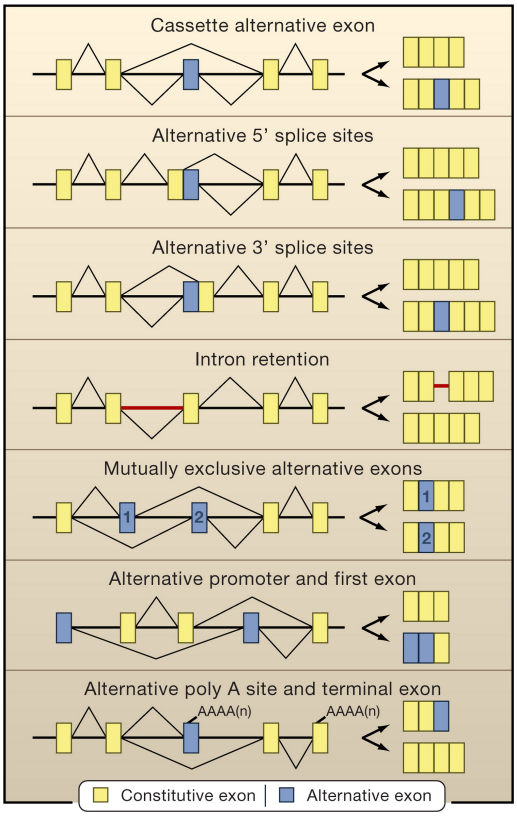


Abbildung 1.2: **Alternatives Spleißen.** Gezeigt werden mögliche alternative Spleißeprodukte. Abbildung übernommen aus Blencowe (2006).

1.3.1 Sequenzierung nach Sanger

Der britische Biochemiker Frederick Sanger, der zum damaligen Zeitpunkt schon Nobelpreisträger war, arbeitete seit 1975 zusammen mit A.R. Coulson an einer Methode zur Bestimmung der Basenabfolge in DNA-Molekülen. Diese enzymatische Methode basiert auf dem Kettenabbruchprinzip und wurde 1977 zusammen mit der erfolgreichen Sequenzierung des ersten Genoms, der Bakteriophage φ174 (auch als PhiX bekannt), veröffentlicht (Sanger und Coulson, 1975). Sie wird noch heute häufig als Positivkontrolle in zahlreichen Sequenzierlaboren eingesetzt. Diese Methode wird im Allgemeinen heutzutage einfach als Sanger-Sequenzierung bezeichnet. Diese

Sequenziermethode entwickelte sich zum *de facto* Standard in der DNA-Sequenzierung und wurde erst in der letzten Dekade durch die Sequenziermethoden der nächsten Generation (NGS) abgelöst.

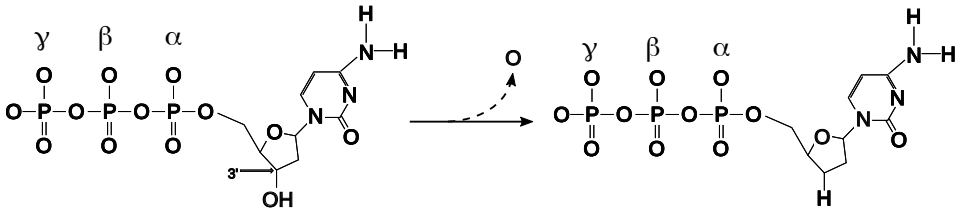


Abbildung 1.3: **Didesoxynukleotide.** Ein Didesoxyribonukleotid besitzt keine 3'-Hydroxylgruppe (-OH), weshalb es zum Abbruch der Kettenverlängerungsreaktion kommt. Diese Abbildung zeigt 2'-Desoxycytidintriphosphat (dCTP) und 2',3'-Didesoxycytidintriphosphat (ddCTP).

Das zugrundeliegende Kettenabbruchprinzip basiert auf einem simplen Ansatz, für dessen Durchführung wenige Voraussetzungen gehören. Die Wichtigste ist ein DNA-Molekül zum Sequenzieren, welches bei 95 °C denaturiert wird, um eine einzelsträngige DNA-Sequenz zu gewinnen. Hierfür werden eine DNA-Polymerase, vier unterschiedliche Typen von Desoxynukleotiden (dNTP) für die Kettenverlängerung und ebenso viele Didesoxynucleotide (ddNTP) für den Kettenabbruch benötigt. Didesoxynucleotide sind artifizielle DNA-Nukleotide, die sich im Grundaufbau von den Desoxynukleotiden nur darin unterscheiden, dass die 3'-Hydroxylgruppe (-OH) an der Ribose fehlt. In Abbildung 1.3 kann man diesen Unterschied anhand von Desoxycytidintriphosphat (dCTP) sehen.
Die Kettenverlängerung geschieht in vier parallelen Reaktionen, mit jeweils nur einer Art von ddNTPs pro Reaktion. Die Polymerase baut bei jedem Reaktionsschritt das entsprechend komplementäre dNTP zur DNA-Matrize ein. Dabei werden zufällig auch immer wieder ddNTPs, entsprechend ihres Verhältnisses zu den dNTPs, eingebaut. Durch die fehlende

3'-Hydroxylgruppe (-OH) ist es der Polymerase nicht mehr möglich ein weiteres Nukleotid anzuheften, sobald ein ddNTP Molekül eingebaut wird. Dies hat zur Folge, dass es zum Kettenverlängerungsabbruch kommt. Durch die massive Parallelisierung entstehen hierbei zahlreiche Fragmente unterschiedlicher Länge. Mit Hilfe einer Gelelektrophorese kann die Größe der Fragmente bestimmt werden. Die Fragmente aus den vier Reaktionen werden separat auf das Gel aufgetragen. Durch die Abfolge der Banden in den vier Spalten lässt sich die Abfolge der synthetisierten dNTPs ableiten. Das Beispiel in Abbildung 1.4 entspricht der Sequenzabfolge `TGATGCCAACGTA`.

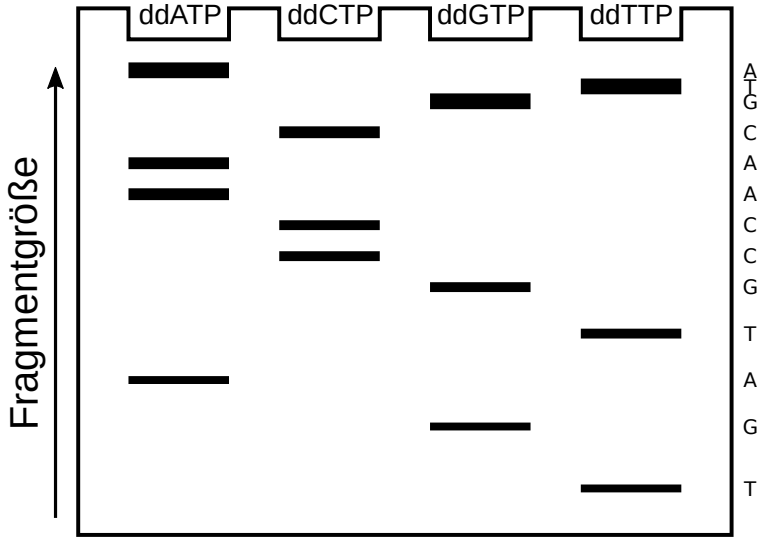


Abbildung 1.4: **Gelelektrophorese.** Die vier Reaktionen mit den verschiedenen ddNTPs werden einzeln auf ein Gel aufgetragen. Anhand der Anordnung der Banden kann man die Abfolge der Nukleotide ableiten.

Diese Methode durchlief weitere Optimierungen im Umgang und der Durchsatzdauer. Anfang der neunziger Jahre des zwanzigsten Jahrhunderts begann man, die radioaktive Markierung der ddNTPs durch Fluoreszenz-Farbstoffe zu ersetzen. Jedes der vier ddNTPs ist mit einem unterschiedlichen Farbstoff gekoppelt. Nicht nur, dass der Umgang mit den radioaktiven

Isotopen entfällt, diese Methode erlaubt es auch statt der vier separaten Reaktionen die vier ddNTPs in einem Gefäß zuzugeben. Mittels Kapillarelektrophorese werden die entstandenen Kettenabbruchprodukte aufgetrennt und die Farbstoffe der ddNTPs (am Ende eines jeden DNA-Fragments) mit Hilfe eines Lasers zur Fluoreszenz angeregt. Ein Detektor misst die Fluoresenzsignale und überträgt sie in ein Chromatogramm, welches direkt der Nukleotidsequenz der sequenzierten DNA-Matrize entspricht.

1.3.2 Sequenzierkosten

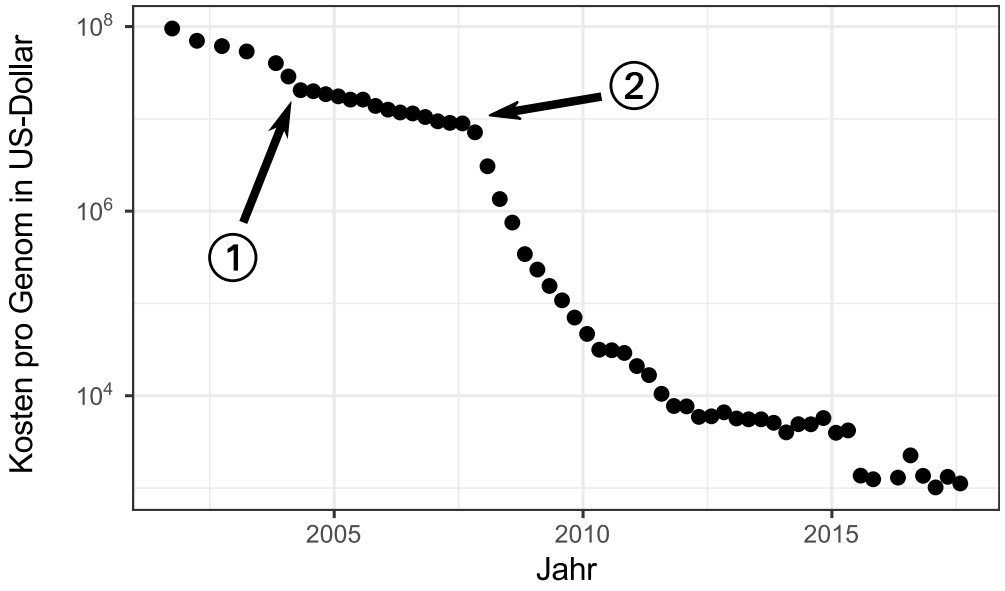


Abbildung 1.5: **Sequenzierkosten für ein humanes Genom.** Die Kosten sind in US-Dollar dargestellt. (1) Das NHGRI investiert 100 Mio. US-Dollar im Zuge des humanen Genomprojekts. (2) Einführung von NGS-Methoden.

Das Humangenomprojekt hatte in den neunziger Jahren des zwanzigsten Jahrhunderts zum Ziel, das humane Genom zu entschlüsseln. In den Anfangsjahren noch komplett von der traditionellen Sequenziermethode nach

Sanger getragen, lagen die ursprünglich veranschlagten Kosten bei einem *US-Dollar* (USD) pro entschlüsselter Base und dementsprechend bei rund drei Mrd. USD pro sequenziertem humanen Genom. Die ursprüngliche Methode der DNA-Sequenzierung nach Sanger durchlief mehrere Optimierungs- und Weiterentwicklungsschritte. Wenige Jahre vor Projektbeginn wurde 1987 von Applied Biosystems die erste automatisierte Sequenziermaschine (ABI 370) auf den Markt gebracht. Mit der Einführung der Kapillarelektrophorese und der Dye-Terminator-Methode konnte der Durchsatz und die Kosten für die Sequenzierung nochmals deutlich gesenkt werden. Zur Blütezeit der Sanger-Ära war eine moderne Maschine in der Lage, bis zu 400 000 Nukleotide (400 kb) pro Tag zu sequenzieren. Im Jahr 2000 wurde, durch die zehnjährige, anhaltende Kooperation von Forschungsinstituten aus verschiedenen Ländern, eine erste sogenannte „Arbeitsversion“ des Humanen Genoms veröffentlicht.

In der Computerindustrie gilt das Moore'sche Gesetz, welches besagt, dass sich die Rechenleistung etwa alle zwei Jahre verdoppelt. Dieses wird gerne auf andere Gebiete übertragen, wie zum Beispiel den Durchsatz und die Kosten bei der DNA-Sequenzierung. Beginnend mit dem Ende des Humangenomprojekts, konnte durch die industrialisierten Methoden in der Sequenzierung mit diesem Gesetz, zumindest bei den Kosten, Schritt gehalten werden (Abbildung 1.5). Im Jahre 2004 startete das *National Human Genome Research Institute* (NHGRI) eine Initiative und investierte mehr als 100 Mio. US-Dollar in die Forschung, um die Sequenzierkosten auf 1 000 US-Dollar pro Genom zu senken (Schloss, 2008). Dies war der Anstoß für die Entwicklung der Sequenziermethoden der zweiten Generation, wie der Pyrosequenzierung, Sequenzierung durch Hybridisierung, Ionen-Halbleiter-DNA-Sequenzierung und *Sequencing By Synthesis* (SBS). Mit dem Umstieg auf diese Methoden konnte ab 2007 der Sequenzierdurchsatz sprunghaft angehoben und das Moore'sche Gesetz weit hinter sich gelassen werden.

1.4 Humanes Referenzgenom

Das erste, vom Humangenomprojekt (HGP) veröffentlichte, humane Referenzgenom wurde noch durch eine Methode erstellt, die fast dreißig Jahre zuvor von Frederick Sanger (F. Sanger u. a., 1977) eingeführt wurde. Solch eine Assemblierung zu einer Genomreferenz besteht aus der DNA von zahlreichen anonymen Spendern und repräsentiert damit eine Mischung verschiedenster Haplotypen[3] von mehreren Individuen, statt dem Genom eines einzelnen Referenzindividuums. Die von dem *International Human Genome Sequencing Consortium* (IHGSC) veröffentlichte Referenz GRCh37 ist eine Kombination der genetischen Informationen von einem Dutzend Individuen. Diese stammten vornehmlich aus der Region um Buffalo, NY (Dudley und Konrad J. Karczewski, 2013) und stellen damit einen deutlichen Populationsbias in den Haplotypen dar, der die globale Vielfalt des humanen Genoms nicht wirklich widerspiegelt.

Für den Vergleich verschiedener Individuen und zur verlässlichen und vollständigen Interpretation von potentiell krankheitsrelevanten Abweichungen benötigt man eine möglichst repräsentative, fehlerfreie und vollständige Referenz. Die erste Arbeitsversion des humanen Genoms mit im Schnitt einem Sequenzierfehler alle 1 000 Basen enthielt noch rund 150 000 Lücken und entsprach nur zu 28% der ersten finalen Version. Bis zur Veröffentlichung des ersten Referenzgenoms im April 2003 konnte die Qualität deutlich erhöht werden, so dass es lediglich noch einen Sequenzierfehler pro 10 000 Basen für 99% des Genoms gab und die Anzahl der Lücken auf 400 reduziert werden konnte. Diese höhere Genauigkeit macht insbesondere für die Genomforschung einen signifikanten Unterschied aus, beispielsweise wenn nach krankheitsursächlichen genetischen Veränderungen gesucht wird, die nur eine oder wenige Basen betreffen.

[3]Haplotyp - Kurzform für haploider Genotyp, welcher eine Nukleotidsequenzvariante für identische chromosomale Abschnitte eines Genoms darstellt. Häufig ist diese für eine spezifische Population identisch oder zumindest innerhalb der Population sehr ähnlich und verbreitet.

Ursprünglich wurde angenommen, damit die größte Herausforderung – eine Blaupause des humanen Genoms als Referenz – gelöst zu haben. Es stellte sich jedoch schnell heraus, dass es deutlich mehr Variabilität im humanen Genom gibt (R. Li, Y. Li, Zheng u. a., 2010) als ursprünglich angenommen. Daraus entstand 2008 die Idee für das 1 000 Genom Projekt (Lander u. a., 2001), welches die Variabilität für verschiedene Populationen zu kartieren versucht und dafür 2 500 Genome sequenziert und analysiert hat.
Dieser Variabilität wurde 2013 mit dem aktuellen Referenzgenome GRCh38 Rechnung getragen, indem es für einige Regionen mit einer besonders markanten Populationsvariabilität, alternative Sequenzen anbietet.

1.5 Next Generation Sequencing

Im Laufe der Jahre wurden immer schnellere, günstigere und vor allem auch weniger arbeitsaufwendige Methoden der Hochdurchsatzsequenzierung / *High Throughput Sequencing* (HTS) entwickelt, die man unter dem Namen *Next Generation Sequencing* (NGS) zusammenfaßt. Teilweise werden sie, im Gegensatz zur Sanger-Sequenzierung, auch als Sequenziermethoden der zweiten Generation bezeichnet. All diese Methoden beruhen auf dem Ansatz, durch massive Parallelisierung Millionen von DNA-Fragmenten in einem Lauf zu sequenzieren.
In den Anfangsjahren gab es zahlreiche konkurrierende Technologien, wobei diese ständig rasante Fortschritte im Durchsatz, Qualität und Fragmentlänge durchliefen. Zu den bedeutendsten Mitstreitern gehörten der Roche 454 Sequenzierer, das Ion Proton™ System von ThermoFisher, das SOLiD Sequenziersystem von ABI und die Sequenzierer von Solexa, die später von Illumina übernommen wurden und auf Sequencing By Synthesis (SBS) basieren. Die Systeme mit der größten Marktdurchdringung sind heutzutage (2018) diejenigen von Illumina. Folglich dominieren diese auch in der humanen Genomsequenzierung.
In den letzten Jahren sind eine Reihe von ernstzunehmenden Konkurren-

ten von Sequenzierern der dritten Generation auf dem Markt erschienen, wie der PacBio Sequel (Single Molecule Real-Time Technology) (Wagner u. a., 2016), der BGIseq-500 (DNA Nanoball Sequenzierung) oder die Oxford Nanopore Technologie (Drmanac u. a., 2010). Der Vorteil einiger dieser Systeme besteht darin, dass keine Anreicherung durch eine Polymeraseketтenreaktion benötigt wird und damit theoretisch keine Limitierung für die Fragmentlänge mehr besteht.
Diese Systeme machen bisher jedoch nur einen sehr geringen Anteil aus. Im Folgenden wird deshalb exemplarisch Sequencing By Synthesis von Illumina ausführlicher erläutert.

1.5.1 Sequencing By Synthesis

Sequencing By Synthesis (SBS), häufig auch einfach Illumina-Sequenzierung genannt, ist die zur Zeit am häufigsten angewandte Sequenziermethode, wenn es darum geht, humane oder andere Vertebraten-DNA/RNA zu sequenzieren. SBS läßt sich in vier wichtige Abschnitte in der Prozessierung zusammenfassen:

- Aufbau einer DNA-Bibliothek
- Clusterbildung
- Sequenzierung
- Computer gestützte Datenanalyse

Aufbau einer DNA-Bibliothek

Der Aufbau einer DNA-Bibliothek ist der erste Schritt bei den Sequenziermethoden der zweiten und dritten Generation. Im Folgenden sollen die einzelnen Schritte genauer beleuchtet werden, aber im Grunde ist eine solche Bibliothek eine Sammlung von DNA-Fragmenten, die sequenzierfertig

aufbereitet sind. Hierfür wird die DNA fragmentiert, diese Fragmente anschließend mit Adaptern versehen und mit Hilfe einer *Polymerasekettenreaktion* (PCR) angereichert (Abbildung 1.6).

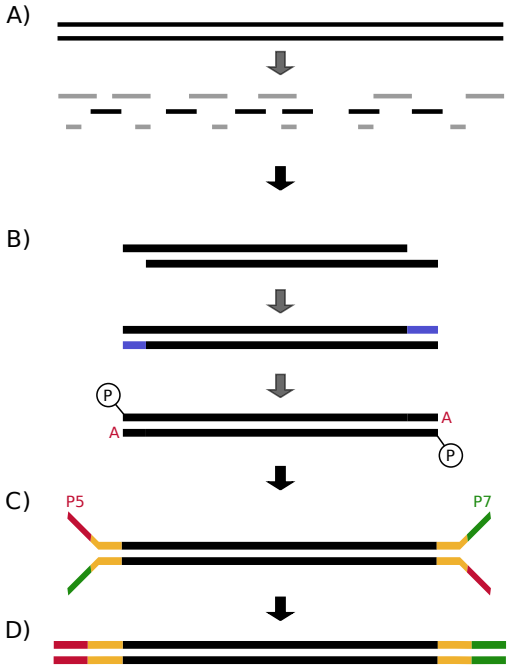


Abbildung 1.6: **Herstellung einer DNA-Bibliothek** A) Fragmentierung von DNA oder cDNA. Nur Fragmente einer bestimmten Größe werden ausgewählt. B) Reparatur der Fragmentenden (in blau), Phosphorylierung (P) der 5'-Enden und Hinzufügen einer Adeninbase (A) an die 3'-Enden, um die Ligation zwischen den Fragmenten zu unterbinden. C) An die Fragemente wird ein Adapter mittels komplementären dT-Überhang ligiert. D) Die fertigen Produkte können mittels PCR amplifiziert werden.

Fragmentierung der DNA Hierbei werden DNA/cDNA-Moleküle in kleinere Fragmente von wenigen hundert Basenpaaren runtergebrochen (Abbildung 1.6 A). Diese sollten im Idealfall eine in der Länge relativ homogene Fragmentmenge ergeben. Hierfür haben sich drei Methoden etabliert:

Sonifikation, enzymatische Fragmentierung und Nubilisation (Knierim u. a., 2011).
Eine weitere Methode ist die enzymatische Fragmentierung. Hierbei werden die DNA-Fragmente durch Enzyme aufgespalten. Im Gegensatz zu den anderen Methoden ist der Nachteil, dass die Enzyme nicht komplett zufällige Bruchpunkte erzeugen. Die Verteilung dieser ist jedoch ausreichend, um eine abdeckende, komplexe DNA-Bibliothek zu erhalten.
Eine alternative Methode ist die Nebulisation (Sambrook und Russell, 2006), bei der DNA mit Hochdruck durch ein kleines Loch gepresst, vernebelt und somit aufgebrochen wird. Die Fragmentgröße wird hierbei durch die Geschwindigkeit, den Druck, die Viskosität der Probenflüssigkeit und die Temperatur bestimmt.

Adapter Ligation Die Fragmentierung resultiert in inhomogenen Fragmenten mit einzelsträngigen 5'- und 3'-Überhängen. Bevor der Adapter ligiert werden kann, muss eine Endreparatur erfolgen, so dass stumpfe Fragmente (vollständige Doppelstränge) entstehen. Deren 5'-Enden werden phosphoryliert und an die 3'-Enden wird eine einzelne Adeninbase (dAMP) angehangen, um die Ligation untereinander zu verhindern. Dieser Schritt wird *dA-Tailing* genannt (Abbildung 1.6 B). Anschließend wird der Adapter mit Hilfe eines komplementären dT-Überhangs an beiden Enden des Fragments in einer "Y-Form" ligiert (Abbildung 1.6 C).

Polymerasekettenreaktion – klonale Amplifikation Entsprechend der angestrebten Sequenzierlänge und -methode werden die adapterligierten DNA-Fragmente ausgewählt und mittels Polymerasekettenreaktion (PCR) exponentiell angereichert.
In einem Thermocycler durchlaufen die DNA-Fragmente mehrere, der gewünschten Zielmenge entsprechende, Zyklen. Ein Zyklus besteht hierbei aus Denaturierung der doppelsträngigen DNA-Fragmente bei 90°C; die Primerhybridisierung geschieht bei 55 bis 65°C; die anschließende Polymerisa-

tion der Basen zum Komplementärstrang beginnt beim Primer und folgt der $5' \rightarrow 3'$-Richtung der DNA-Fragmentvorlage bei 68 bis 72°C.

Einflussfaktoren auf die Qualität der DNA-Bibliothek

Das Hauptziel bei der Erzeugung einer DNA-Bibliothek besteht darin, so viel Bias wie möglich zu vermeiden. Als Bias versteht man die systematische Verzerrung der Wahrheit (Ursprungsdaten) durch das Experimentdesign, wie zum Beispiel technische Schritte. Da gerade diese sich nicht komplett vermeiden lassen, sollte man verstehen wie sie entstehen können und wie man ihnen entgegenwirken oder sie möglichst gering halten kann.
Die Komplexität einer NGS DNA-Bibliothek kann den Bias des Experimentdesigns widerspiegeln. Idealerweise erhält man eine hochkomplexe Bibliothek, welche der Variabilität des Ausgangsmaterials entspricht und damit eine relativ gleichmäßige Abdeckung der Zielsequenz ermöglicht. Die technische Herausforderung besteht darin, dass jeder Amplifikationsschritt diese Variabilität reduzieren kann. Die Komplexität der Bibliothek kann anhand der Anzahl oder auch dem Prozentsatz der Duplikate[a] gemessen werden (Parkinson u. a., 2012). Ein Nachteil hierbei ist, dass zufällige Duplikate, also solche, die nicht durch die Amplifikation entstehen, mit der Erhöhung der erwünschten Abdeckung der Zielsequenz zunehmen. In einer idealen Welt, in der die Fragmentgrößen der Bibliothek immer gleich wären, könnte man theoretisch eine maximale Abdeckung – nach Entfernung von Duplikaten – entsprechende der Gesamtlänge der Reads erhalten. Abweichungen davon würden direkt auf eine Veränderung in der Sequenzlänge auf einem Allel hinweisen. Ein ähnliches Problem stellen repetitive Regionen dar. Hier können die Reads oft nicht eineindeutig auf die Referenz abgebildet werden. Dies läßt sich teilweise durch eine zufällige Platzierung der Reads kompensieren, die Qualität der so gewonnenen Information ist aber auch dementsprechend fraglich. Eine *de novo* Assemblierung dieser Regionen ist ohne Reads, welche die komplette Region überspannen, nahezu unmöglich.

[a] Doppelte Reads - also Reads, welche auf die gleiche Position des Referenzgenom abgebildet werden (Gilfillan u. a., 2012).

Clusterbildung

Die Bibliothek wird bei 95°C zu *einzelsträngigen DNA-Fragmenten* (ssDNA) aufgebrochen. Auf der Innenoberfläche der Flowcellkanäle gibt es einen dichten Teppich aus komplementären P5-Adaptoren, an welche die ssDNA-Moleküle binden. Von hier aus wird eine komplementäre Kopie der originalen ssDNA-Stränge synthetisiert. Anschließend wird die ssDNA denaturiert und abgewaschen, so dass nur noch die flowcellgebundene Kopie haften bleibt und als Vorlage für weitere Kopien dient. Dessen P7-Ende bindet an den entsprechenden Adaptor auf der Flowcell und von diesem ausgehend wird eine Kopie des Oligonukleotids synthetisiert. Das entstandene doppelsträngige Molekül wird denaturiert, so dass nun zwei komplementäre einzelsträngige Vorlagen an die Flowcell gebunden und für eine weitere Amplifikationsrunde verfügbar sind. Nach fünf bis sieben dieser Amplifikationsrunden werden die Oligonukleotide an den P5-Adaptoren abgespalten, so dass nur Moleküle in der Orientierung der ursprünglichen ssDNA für die Sequenzierung auf der Flowcell gebunden verbleiben. Dieser Vorgang wird als Brückenamplifikation bezeichnet, da die flowcellgebundenen Oligonukleotide bei der Amplifikation brückenartige Gebilde formen. Eine schematische Darstellung findet sich in Abbildung 1.7. Rund um die Bindungsstellen der ursprünglichen ssDNAs entstehen so Millionen von Clustern von homogenen Kopien der originalen Sequenz. Ein Cluster entspricht somit genau einem Read.

Sequenzierung

Im Sequenzierschritt wird anschließend die Abfolge der Nukleinsäuren ermittelt. Wie auch bei der Clusterbildung geschieht dies durch einen sich immer wiederholenden Schritt. Die vier am 3'-Ende blockierten und fluoreszenzmarkierten Desoxynukleotide (*Desoxynukleotid* (dNTP)) werden zusammen mit einem Primer und DNA-Polymerase auf die Flowcell gegeben, binden entsprechend der Oligonukleotidvorlage einzeln und werden anschlie-

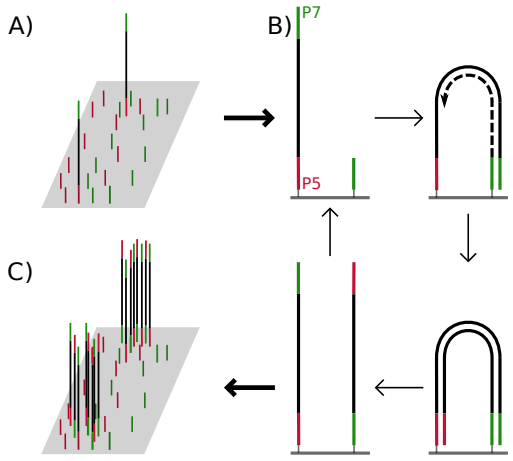


Abbildung 1.7: **Brückenamplifizierung** A) Einzelsträngige Fragmente der DNA-Bibliothek binden zufällig an den P5-komplementären Adaptoren auf der Flowcell. B) Die P7-Enden der Fragmente binden an die entsprechenden Adaptoren auf der Flowcell und bilden eine Brücke. Von den Adaptoren aus wird ein komplementärer Strang synthetisiert. Diese werden denaturiert, so dass jetzt zwei komplementäre Vorlagen an die Flowcell binden. C) Nach mehreren Wiederholungen der Brückenamplifikation haben sich an den ursprünglichen Bindungsstellen Cluster identischer Fragmente gebildet.

ßend mit Hilfe eines Lasers zur Emittierung ihres Fluoresenzsignals angeregt und anschließend das 3'-Ende freigegeben. Das Signal wird pro Cluster gemessen und die Basen für den entsprechenden Zyklus damit bestimmt. Dies wird entsprechend der gewünschten Sequenzlänge der Reads wiederholt – typischerweise 75 bis 250 Zyklen.

Computer gestützte Datenanalyse

Abhängig vom Aufbau des Experiments werden die Sequenzierdaten im letzten Schritt bioinformatisch analysiert. Typische Anwendungen sind ein Alignment der sequenzierten Reads an ein Referenz-Genom (z.B. WES, WGS, RNA-Seq) oder mit den Sequenzen wird ein *de novo* Assembly durchgeführt, um längere Contigs zu generieren (z.B. WGS, RNA-Seq).

Im Fall von Keimzellanalysen werden in der Regel Varianten von einem Patientengenom zu einem Referenzgenom gesucht. Hierfür wird ein referenzbasiertes Assembly verwendet, also die Abbildung aller individuellen sequenzierten NGS-Reads auf eine haploide Referenz. Im Anschluß folgt ein Variantencalling – der Bestimmung von Unterschieden zwischen dem Genom des Individuums und der Referenz – um diese Varianten in einem letzten Schritt zu interpretieren und bewerten.

Alignment

Das Ergebnis der NGS-Sequenzierung ist, unabhängig von der verwendeten Technik, eine Menge von hunderttausenden bis mehrere Millionen NGS-Reads. Für die Illumina SBS-Methode mit NGS-Reads einer Länge von 50 bp bis 150 bp, befinden sich diese im Bereich von 40 Mio. (WES) bis 900 Mio. (WGS). Die größte Herausforderung ist es, diese Reads auf der Referenzsequenz abzubilden, also ein globales Alignment zwischen allen Reads und der Referenz durchzuführen. Solch ein paarweises Alignment ist in linearer Zeit $O(ab)$ möglich, wobei a und b die Längen der beiden Sequenzen sind. Praktisch wird jedoch auf einem Index des Genoms gearbeitet, womit es in kurzer Zeit möglich ist, Kandidatenregionen (Seeds) für das Readalignment zu bestimmen und zu erweitern. Hierfür wurden verschiedene Algorithmen (SOAP (R. Li, Y. Li, Kristiansen u. a., 2008), Maq (H. Li, Ruan und Durbin, 2008), RazerS (Weese u. a., 2009)) entwickelt, welche Hashtabellen auf den Reads oder dem Referenzgenom für die Bestimmung der Seeds verwenden. Eine effizientere Möglichkeit für den Zugriff auf die Referenz bieten die neueren Alignmentprogramme, wie Bowtie (Langmead u. a., 2009), BWA (Li, Heng, 2013), SOAP2 (R. Li, Yu u. a., 2009), welche eine Burrows-Wheeler-Transformation auf das Referenzgenom anwenden. Damit kann eine noch effizientere Kompression der Indexstrukturen erfolgen. Somit ist es möglich, den Index (Suffix-Array, FM-Index, ...) komplett im Hauptspeicher zu halten, was einen noch schnelleren Zugriff ermöglicht. Eine nähere Erläuterung kann im Buch (P. N. Robinson, M.Piro und Jäger,

2018) gefunden werden.

Variantencalling

Die Bestimmung von Varianten (engl.: *Variantcalling*) spielt insbesondere auch bei der Keimzelluntersuchung eine wichtige Rolle. Das humane Genom ist bis auf die Geschlechtschromosomen diploid, das bedeutet an jeder Position wird im Vergleich zum Referenzgenom entweder die gleiche Base, eine völlig andere, oder zur Hälfte eine andere Base erwartet, was auf Heterozygotie hindeutet. Die Unterschiede des individuellen Genoms zur Referenz können häufig eine Erklärung für oder Hinweis auf den Phänotypen geben. Eine Herausforderung ist es, insbesondere bei einer niedrigen Abdeckung, die echten Varianten mit einer hohen Sensitivität vom Hintergrundrauschen zu trennen. Dieses kann durch verschiedene Faktoren verursacht werden. Typische Fehlerquellen sind PCR-Artefakte, Sequenzierfehler durch zum Beispiel überlagernde Cluster, Misalignment der NGS-Reads. Hierfür gibt es eine Reihe von Programmen wie SAMtools, Freebayes, Platypus, ... (Sandmann u. a., 2017), welche mittels statistischer Methoden oder heuristischer Ansätze versuchen, dem entgegenzuwirken. Ein momentan häufig eingesetztes Programm ist der Haplotypcaller von GATK (McKenna u. a., 2010), welcher auch in die Analysespipeline von Illumina integriert ist. Es implementiert ein lokales, paarweises Realignment für jedes Paar von lokal alignierten NGS-Reads und bestimmt die Varianten mit einem Maximumlikelihood Ansatz (siehe auch P. N. Robinson, M.Piro und Jäger (2018)).

1.6 Genetische Variationen

Varianten im Genom sind seit langer Zeit bekannt und so ist die Identifikation dieser, nach der Ermittlung der Gensequenzen durch die NGS-Sequenzierung, auch eine Triebfeder für die Entwicklung der Sequenziermethoden der ersten Generation (Sanger) und der folgenden High Throughput Sequencing Methoden gewesen. Mit der Fertigstellung des Humanen Refe-

renzgenoms stellte sich jedoch entgegen der ursprünglichen Annahme heraus, dass es weit mehr Varianz im humanen Genom gibt als angenommen. Die Arten und die Anzahl der Variationen zur Referenz unterscheiden häufig nicht zwischen zwei Individuen mit oder ohne klinische Auffälligkeiten und lassen sich anhand ihrer Länge grob in drei Klassen einteilen:

1. Chromosomenaberrationen, die so groß sind, dass sie mit einem Mikroskop im Karyogram identifiziert werden können (das Fehlen von ganzen Chromosomen oder Chromosomenarmen);

2. größere strukturelle Aberrationen, die sich mittels FISH unter einem Mikroskop detektieren lassen;

3. submikroskopische Aberrationen und Einzelbasenvarianten.

Die Häufigkeit der Aberrationen korreliert dabei mit deren Größe, beziehungsweise der Anzahl betroffener Basen. Kleinere Varianten kommen relativ häufig vor, wohingegen große sehr selten sind.

1.6.1 Einzelbasenvarianten – SNVs und MNVs

Unter Einzelbasenvarianten lassen sich Varianten zusammenfassen, die nur eine einzelne Base betreffen (SNVs). Hierbei läßt sich zwischen einer Insertion, einer Deletion und einer Substitution, dem Austausch einer einzelnen Base gegen eine andere, unterscheiden. In den ersten beiden Fällen würde es im codierenden Bereich zu einem Frameshift[4] bei der Translation kommen, der die Abfolge der Aminosäuren des Proteins verändert. Im nicht-codierenden, aber gegebenenfalls funktionalen Bereich würde es zur Zerstörung eines Bindungsmotivs kommen. Die Veränderung einer einzelnen Base kann ebenfalls zur Veränderung von Bindungsmotiven führen oder aber bei der Translation durch die Codonveränderung zum Austausch der eingebauten Aminosäure.

[4]Veränderung des Leserasters

Sind mehrere aufeinanderfolgende Basen betroffen, so spricht man von einer Multinukleotidvariante (MNV). Dies können kleinste Insertionen oder Deletionen sein, auch InDels genannt (Krawitz u. a., 2010), oder der Austausch von benachbarten Basen. Hierbei sind beliebig komplexe Kombinationen vorstellbar. Im unten stehenden Kasten „SNVs und MNVs“ sind Beispiele für Substitutionen, Deletionen und Insertionen jeweils für SNVs und MNVs gezeigt.

SNVs und MNVs

```
          Substitution   Deletion     Insertion
SNV
     REF: ...ACCACAG.....GACCATA......TAA GCC...
     ALT: ...ACCGCAG.....GAC-ATA......TAAAGCC...

MNV
     REF: ...ATTTCGAAG...ACCTTCAATT...TGACC   CAC...
     ALT: ...ATTAGTAAG...ACC----ATT...TGAGATGTCAC...
```

1.6.2 Chromosomenaberrationen

Chromosomenaberrationen als größte mögliche Variationen kommen relativ selten vor – etwa bei jedem 200. neugeborenen Kind. Die meisten sind mit dem Leben nicht vereinbar und führen oft frühzeitig zu einem Abort. Einige Chromosomenstörungen lassen jedoch die Geburt von lebensfähigen Kindern zu. Generell lassen sie sich in numerische und strukturelle Chromosomenaberrationen unterscheiden.

Numerische Chromosomenaberrationen

Die meisten humanen Chromosomenaberrationen sind numerisch, das heißt es gibt eine Veränderung in der Gesamtchromosomenanzahl, einzelne Chromosomen sind nicht mehr diploid vorhanden. Beispiele hierfür sind:

- Monosomien - das Fehlen einzelner Chromosomen, wie z.B. das Ullrich-Turner-Syndrom (Karyotyp 45,X)

- Trisomien - das Auftreten von zusätzlichen Chromosomen, wie beim Down-Syndom/Trisomie 21 (Karyotyp 47,XX+21/XY+21) oder dem Klinefelter-Syndrom (Karyotyp 47,XXY)

- Polyploidien - Vervielfältigung des kompletten Chromosomensatzes, wie bei Triploidien (Karyotyp 69,XXX)

Strukturelle Chromosomenaberrationen

Bei strukturellen Chromosomenveränderungen wird zwischen balancierten und unbalancierten Aberrationen unterscheiden. Balancierte Chromosomenveränderungen haben in der Regel keine bis wenige Auswirkungen für den Betroffenen, da sich die genetischen Informationen durch eine balancierte Translokation nur von einem auf ein anderes Chromosom übertragen haben. Dies führt jedoch in den Folgegenerationen häufig zu unbalancierten Chromosomenveränderungen, wenn während der meiotischen Teilung balancierte Chromosomenveränderungen ungleich auf die Keimzellen aufgeteilt werden. Bei der anschließenden Rekombination der elterlichen Chromosomen kommt es dann zu einer Imbalance der Chromosomen. Bekanntere Beispiele hierfür sind etwa das Cri-du-Chat-Syndrom, ausgelöst durch eine distale Deletion des kurzen Arms von Chromosom 5, oder auch das De-Grouchy-Syndrom Typ I/II, verursacht durch eine Deletion des kurzen/langen Arms von Chromosom 18.

1.7 Medizinische Notwendigkeit einer Interpretation von genomischen Varianten

Das erste Mal konnte im Jahre 1940 durch G. W. Beadle und Tatum (1941) gezeigt werden, dass eine Veränderung in der Nukleotidsequenz eines Gens

zu einem veränderten Phänotypen führt. Diese Beobachtung führte letzendlich zur *Ein-Gen-ein-Enzym-Hypothese* (George Wells Beadle, 1945; Tatum und G. W. Beadle, 1945), wonach in der Sequenz eines Gens die komplette Information zur Bildung eines bestimmten Enzyms enthalten ist. Erkrankungen, die durch eine Veränderungen in der genomischen Sequenz zu einem abnormalen Phänotypen führen, werden auch als Erbkrankheiten bezeichnet, da diese von einer Generation zur nächsten über die Gameten weitergegeben werden können. Man bezeichnet sie auch als genetische Erkrankungen oder Mendelsche Erkrankungen. Bei der Interpretation der genetischen Veränderungen sollte jedoch immer beachtet werden, dass die äußeren Faktoren ebenfalls einen deutlichen Einfluss auf die Ausprägung des möglichen Phänotyps haben. Ein Beispiel hierfür ist die Auswirkung von Unterernährung auf Kinder, welche dort zum Beispiel zu einer Entwicklungsverzögerung und verlangsamten Wachstum führen kann.

Kapitel 2

Jannovar: eine Java-Bibliothek zur Beurteilung von genomischen Varianten

Ein wichtiger Schritt in der Beurteilung von Varianten in den codierenden Bereichen (Exom) des Genoms ist es, Informationen zu Charakteristika dieser Position zusammenzutragen und zu verknüpfen. Diesen Schritt nennt man Annotation einer Variante. Dieses Kapitel behandelt die Java-Bibliothek *Jannovar*, welche im Hinblick auf die Annotation von Exomen entwickelt wurde. *Jannovar* enthält eine Beispielimplementation zur Annotation von genomischen Varianten in VCF-Dateien, wurde jedoch primär als Annotationsbibliothek für die Verwendung in größeren Projekten entworfen.

2.1 Einführung

Whole Exome Sequencing (WES) ist die gezielte Anreicherung und Sequenzierung von protein-codierenden Exons und einigen weiteren regulatorischen oder krankheitsrelevanten Bereichen im humanen Genom. Es ist damit eine mächtige und zudem auch kosteneffektive Methode zur Identifizierung von neuen krankheitsassoziierten Genen, die den Mendelschen Gesetzen folgen (Splinter u. a., 2018). Die kontinuierliche Weiterentwick-

lung in der Sequenzierung hin zum hoch parallelisierten Durchsatzverfahren, auch High Throughput Sequencing (HTS) genannt (Margulies u. a., 2005; Shendure u. a., 2005), führte 2010 zur ersten Identifikation einer genetischen Veränderung, die dem Mendelschen Vererbungsmuster folgt (Ng u. a., 2010). In der Tat ist Whole Exome Sequencing (WES) so effektiv, dass es immer häufiger auch im diagnostischen Umfeld eingesetzt wird (Bamshad u. a., 2011; Choi u. a., 2012; P. N. Robinson, Krawitz und Mundlos, 2011; Shendure, 2011). So wurden mittlerweile schon über 1 000 krankheitsrelevante Gene mit dieser Methode identifiziert und publiziert (Fernandez-Marmiesse, Gouveia und Couce, 2018; Rabbani u. a., 2012). Bedingt durch die stetig fallenden Sequenzierkosten – aktuell (2020) liegen diese Kosten bei 500 € – ist WES nicht mehr nur für die Forschung interessant, sondern zunehmend auch für die klinische Diagnostik in der Humangenetik. Daneben gewinnt es auch immer mehr an Bedeutung für zahlreiche weitere Fragestellungen in der Medizin und anderen Forschungsgebieten, wie der Evolotionsbiologie (Sullivan u. a., 2017).

Neben den Kosten sind und bleiben die Analyse, Interpretation und Auswertung der WES-Daten eine große Herausforderung. Je nach verwendeter Anreicherungsmethode und Zielregion[1] identifiziert man bis zu mehrere zehntausend Variationen zum Referenzgenom im Exom eines Individuums. Die Identifizierung der Varianten aus den Rohdaten ist von mehreren Schritten abhängig und kann sich durch die Wahl der verwendeten Methoden bzw. Programme zur Abbildung der Rohdaten (Mapping) und die anschließende Variantenbestimmung (Calling) deutlich unterscheiden (Laurie u. a., 2016; Warden u. a., 2014).

Ein wichtiger Schritt in der Interpretation, der durch das *Calling*[2] bestimm-

[1] Bereiche auf dem Genom, welche für die Sequenzierung angereichert werden. Im allgemeinen alle Exons der protein-codierenden Gene, mittlerweile aber auch zahlreiche nicht-codierende Gene und regulatorische Bereiche.

[2] Der Schritt in der bioinformatischen Analyse von NGS-Daten, bei dem anhand der Sequenz der überspannenden Reads zu einer genomischen Position Rückschlüsse auf das Vorhandensein einer möglichen Variante gezogen werden.

ten Varianten, ist die Annotation hinsichtlich ihres Effekts auf das enthaltene Gen oder Transkript, beziehungsweise die möglichen Auswirkungen während der Translation der prozessierten Transkripte. Für diesen Zweck wurden zahlreiche Programme entwickelt, wie ANNOVAR (K. Wang, M. Li und Hakonarson, 2010), VAT (Habegger u. a., 2012) oder auch der Variant Effect Predictor (VEP) vom Ensembl Projekt (Flicek u. a., 2013; McLaren u. a., 2010), um nur einige zu nennen. Diese Programme sind in ihrer Grundfunktion nicht zu bemängeln und jedes hat seine Vorteile, seien es Laufzeit, Annotationsumfang oder auch die Flexibilität der Datenbanken. Was allen gemeinsam fehlt ist die Möglichkeit auch als lokale Software-Bibliothek in größeren Frameworks eingesetzt zu werden.

2.2 Überblick

Der typische Anwendungsfall für Variantenannotationsprogramme, wie auch *Jannovar*, ist die Eingabe von Varianten im VCF-Format, deren Annotation und die anschließende Ausgabe einer annotierten VCF-Datei. Im Folgenden wird das VCF-Format erläutert und ein Überblick über die von *Jannovar* verwendeten Nomenklaturen zur Variantenannotation und Transkriptdatenbanken gegeben, bevor sich der nächste Abschnitt mit der effizienten Implementierung des Annotationsschritts in *Jannovar* befasst. Am Ende des Kapitels wird anhand eines Beispiels nochmals auf *Jannovar* als Software-Bibliothek eingegangen.

2.2.1 VCF-Format

Das Variant Calling Format (VCF) ist der *De-Facto*-Standard zur Repräsentation von Varianten im humanen Genom. Der Aufbau ist relativ simpel, unterliegt jedoch einer Reihe von Spezifikationen. Momentan liegt diese Spezifikation in der Version 4.3 vor (Danecek u. a., 2011)[3]. Eine VCF-Datei ist

[3]Diese (VCFv4.3.pdf) und die jeweils aktuelle Spezifikation kann unter https://samtools.github.io/hts-specs/ gefunden werden.

eine tabellarische Textdatei bestehend aus drei Abschnitten. Sie beginnt immer mit den Meta-Informationen, gefolgt von einem Kopfbereich (header), der die Spalten beschreibt, und dem eigentlichen Inhalt (body), welcher die eigentlichen variantenbeschreibenden Daten enthält (Abbildung 2.1).

Meta-Informationen

Zeilen mit Meta-Informationen werden stets mit einer Doppelraute (`##`) eingeleitet. Die erste Meta-Information und damit auch die erste Zeile in einer VCF-Datei ist immer die Angabe zum Dateiformat.

```
##fileformat=VCFv4.3
```

VCF-Format

```
##fileformat=VCFv4.3
##fileDate=20090805
##source=myImputationProgramV3.1
##reference=file:///seq/references/1000GenomesPilot-NCBI36.fasta
##contig=<ID=20,length=62435964,assembly=B36,md5=f126cdf8a6e0c7f379d618ff66beb2da,species="Homo sapiens">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership, build 129">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
##FILTER=<ID=q10,Description="Quality below 10">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
#CHROM POS     ID        REF    ALT     QUAL FILTER INFO                              FORMAT      NA00001
20     14370   rs6054257 G      A       29   PASS   DP=14;AF=0.5;DB;H2               GT:GQ:DP:HQ 1|0:48:8:51,51
20     17330   .         T      A       3    q10    DP=11;AF=0.017                   GT:GQ:DP:HQ 0|1:3:5:65,3
20     1110696 rs6040355 A      G,T     67   PASS   DP=10;AF=0.333,0.667;DB          GT:GQ:DP:HQ 1|2:21:6:23,27
```

Abbildung 2.1: Zu sehen ist das eingekürzte Beispiel einer VCF-Datei aus den VCFv4.3 Spezifikationen. Die Meta-Informationen werden durch doppelte Rauten (`##`) gekennzeichnet, gefolgt durch den *header* (`#`) und dem *body*, der in diesem Fall Bescheibungen zu drei Varianten auf Chromosom 20 enthält.

Variantenbeschreibung

Die Beschreibung der Varianten startet mit einer Beschreibung der Spalten. Die Anordnung und Bezeichnung ist für die ersten neun Spalten immer identisch:

Phred-score

Der Phred Qualitätswert Q definiert sich als logarithmisch abhängig von der (Base-calling) Fehlerwahrscheinlichkeit P.

$$Q = -10log_{10}P \tag{2.1}$$

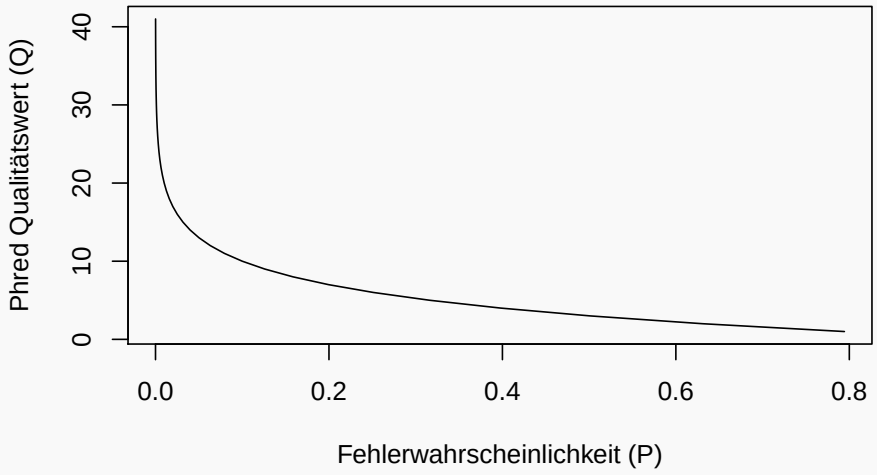


Abbildung 2.2: Beschreibung des Phred Qualitätswerts.

Tabellenspalten einer VCF-Datei

`#CHROM` `POS` `ID` `REF` `ALT` `QUAL` `FILTER` `INFO` `FORMAT` `...`

`CHROM` Die ID der Referenzsequenz. Häufig der Chromosomenname (`chr1, chr2, ...`).

`POS` Der Start der Variante in Bezug zu der Referenzsequenz.

`ID` Externe ID der Variante, falls in einer Datenbank bekannt, wie z.B. dbSNP (1000 Genomes Project Consortium u. a., 2010), Cosmic (Forbes

u. a., 2017), ...

`REF` Die betroffenen Referenzbase(n) aus dem Alphabet (`A,C,G,T,N`).

`ALT` Die alternativ gefundene Sequenz(en) bestehend aus dem Alphabet (`A,C,G,T,N,*`) oder einem Identifier (`ID`), der im *header* definiert wurde und eine Strukturvariantenklasse beschreibt.

`QUAL` Ein Qualitätswert als Phred-score (siehe Abbildung 2.2) für die Annahme der Alternative.

`FILTER` Filterstatus. `PASS`, wenn alle Filter passiert wurden, ansonsten eine Auflistung ohne Leerzeichen der nicht passierten Filter, getrennt durch ein Semikolon. Wurde keine Filter angewandt, so steht ein einfaches „.“.

`INFO*` Zusätzliche beschreibende Informationen. Die einzelnen Felder in der `INFO` Spalte werden durch ein Semikolon voneinander getrennt. Die Felder müssen in den Metainformationen am Anfang der Datei definiert werden und sind im folgenden Format:

```
<key>=<data>[,data]
```

`FORMAT*` Diese Spalte beschreibt, in welcher Reihenfolge die Genotypinformationen zu den einzelnen Einträgen für die Proben in den folgenden Spalten abgelegt sind. Die Anordnung der Informationen ist für alle Einträge in der VCF-Datei identisch. Ein Beispiel für solch ein Eintrag ist `GT:GQ:DP:HQ`. In der Regel steht der Eintrag zum Genotyp (`GT`) an erster Position. Alle Felder müssen in den Metainformationen definiert sein.

* Sowohl für die `INFO`- als auch `FORMAT`-Spalte gibt es eine Reihe von reservierten Feldern, die in den Spezifikationen zum entsprechenden Dateiformat nachgeschlagen werden können. Als Beispiel sei hier nochmals auf den Genotyp (`GT`) verwiesen.

2.2.2 Variantennormalisierung

Die Variantennormalisierung ist ein essentieller Schritt zur Vermeidung von redundanten und uneindeutigen Variantenbeschreibungen in Datenbanken. Betrachtet man die einzelnen Einträge in der Datenbank für Singlenukleotidpolymorphismen (dbSNP) (Sherry u. a., 2001; Smigielski u. a., 2000), so stößt man immer wieder auf lokale Anhäufungen von Varianten.
Im Folgenden ein Beispiel für zwei dbSNP Einträge: Eine Insertion von vier Basen `CTTT` zwischen den Basen an Position 16 537 622 und 16 537 623 auf Chromosom 22.

Variantennormalisierung - Teil 1

rs200449532:

```
TCAAAGACATGAATACAACCTAATGACTCCTTGTTCATCAAGA
                        ^
                        CTTT
```

rs4010175:

```
TCAAAGACATGAATACAACCTAATGACTCCTTGTTCATCAAGA
                        ^
                        TTCT
                        TTCG
```

Betrachtet man in diesem Fall für den dbSNP Eintrag *rs4010175* nur die Insertion `TTCT` und vernachlässigt für den Augenblick `TTCG`. Fügt man die alternativen Allele in die Originalsequenz ein, so erhält man exakt die gleichen Sequenzen:

Variantennormalisierung - Teil 2

```
TCAAAGACATGAATACAACCTTTCTAATGACTCCTTGTTCATCAAGA
TCAAAGACATGAATACAACCTTTCTAATGACTCCTTGTTCATCAAGA
```

Daraus lässt sich schlußfolgern, dass die beiden Einträge *rs200449532* und *rs4010175* identisch sind (zumindest für die Insertion von `CTTT` bzw. `TTCT`).

Durch die Wahl der Positionierung sind sie jedoch nicht mehr eineindeutig. In der Tat wäre es in diesem Fall gut, die Insertion von `TTCT` aus *rs4010175* mit dem Eintrag *rs200449532* zusammenzufassen und die Varianten damit zu normalisieren.
Die Normalisierung einer Variantenrepräsentation besteht aus zwei Teilen. Zum einen sollte sie immer sparsam in der Darstellung sein und des weiteren links-aligniert werden – in Abhängigkeit zur Variantenlänge und Position.

Sparsamkeit Nach dem Prinzip der Parsimonie soll eine höchstmögliche Sparsamkeit bei der Beschreibung angewendet werden. Im Kontext der Variantenrepräsentation bedeutet Sparsamkeit, eine Variante mit so wenig Nukleotiden wie möglich darzustellen. Dabei darf die Länge keines Allels auf 0 sinken.
Dies lässt sich am leichtesten mit Hilfe einer Multinukleotidvariante (MNV) veranschaulichen. Im folgenden Beispiel sieht man die Referenz und das alternative Allel einer MNV:

```
REF: CCCGATCCCC
ALT: CCCATGCCCC
```

Mögliche Repräsentationen dieser MNV wären:

```
                    VCF-Format
REF    CCCGATCCCC | POS REF    ALT    |
ALT1     CATG     | 4   CGAT   CATG   | nicht links-getrimmt
ALT2      ATGC    | 5   GATC   ATGC   | nicht rechts-getrimmt
ALT3     CATGC    | 4   CGATC  CATGC  | nicht getrimmt
ALT4      ATG     | 5   GAT    ATG    | normalisiert
```

Links-aligniert Eine Variante ist dann links aligniert, wenn ihr Startpunkt möglichst weit nach links verschoben ist, ohne ihren Informationsgehalt zu verlieren. Entsprechend den VCF-Spezifikationen dürfen Allele nicht leer sein (Danecek u. a., 2011).

Dies lässt sich anhand von einem kurzen Tandem-Repeat veranschaulichen:

```
REF: CGTATATATATAGAC
ALT: CGTATATATAGAC TA Deletion
```

```
                    VCF-Format
REF   CGTATATATATAGAC | POS REF    ALT |
REF           TA      | 11   TA   .    | nicht links-aligniert
ALT1          .       |                | ALT leer
REF         TAT       | 9    TAT  T    | nicht links-aligniert
ALT1        T         |                | sparsame Darstellung
REF     TAT           | 6    TAT  T    | normalisiert
ALT3    T             |                |
```

2.2.3 HGVS-Nomenklatur

Eine der wichtigsten Eigenschaften von *Jannovar* ist es, dass transkribierte Varianten in der im medizinischen und wissenschaftlichen Umfeld gebräuchlichen HGVS-Nomenklatur annotiert werden. Diese ist normalisiert und damit eindeutig, womit eine automatisierte Vergleichbarkeit von Varianten und deren Annotation gegeben ist.

Eines der erklärten Ziele der Human Genome Variation Society (HGVS) ist es, eine einheitliche Annotation für Variationen in DNA-, RNA- und Proteinsequenzen zu etablieren. Eine konsistente und einheitliche Beschreibung von Sequenzvarianten ist essentiell für einen vergleichbaren Austausch von Informationen und genomischen Analysen (Dunnen, 2017). Insbesondere für die DNA-Diagnostik ist eine standardisierte Beschreibung und der Austausch von bekannten Varianten kritisch. Obwohl der Ansatz zur systematischen Annotation ursprünglich im Jahre 2000 für humane Daten geschaffen wurde, wird er mittlerweile auch häufig für andere gut annotierte Spezies (Maus, Ratte, ...) verwendet und hat sich unter der Federführung der Human Genome Variation Society, dem *Human Variome Project* (HVP) und der *Human Genome Organization* (HUGO) zu einem international anerkannten Standard entwickelt.

In älteren Publikationen kann man häufig nicht eindeutige Beschreibungen für Varianten finden. Zur Veranschaulichung soll ein simples Beispiel dienen. Angenommen im Gen `ABC13` steht in der Referenz für die codierende Sequenz an Position 78 ein Adenosin (`A`). Bei der Sequenzierung des Gens mittels Sanger findet man nun an dieser Stelle ein Guanin (`G`). Eine häufig verwendete Beschreibung für solch eine Variante wäre `ABC13:A78G`. Anhand des verwendeten 1-Buchstabenkodes ist es nicht ersichtlich, dass es sich bei `A` und `G` um Nukleotide handelt. Diese Bezeichnung könnte ebenfalls für die Proteinsequenz gelten, d.h. in dem Polypeptid wurde an Position 78 ein Alanin (`A`) gegen ein Glycin (`G`) ausgetauscht. Ebenso ist der Verweis auf das Gen mittels des Gennamens uneindeutig, da weder die Version des Gens, noch die Referenzdatenbank eindeutig daraus hervorgeht. Diese Uneindeutigkeiten lassen keinen Rückschluss auf die tatsächlich verwendete Referenz zu. Aus diesem Grund ist die HGVS-Annotation eine vor allem im medizinischen und molekularbiologischen Bereich verwendete Annotation für chromosomale Veränderungen und deren direkte Auswirkungen im humanen Genom. Sie stellt heute den Standard zur Annotation von Varianten in ärztlichen Befunden sowie in Publikationen dar.
Die HGVS-Annotation gibt einige grundlegende Empfehlungen, die in *Jannovar* implementiert sind.

- Varianten sollen auf dem grundlegendsten Level (DNA) definiert werden. Zusätzlich kann man die Beschreibungen auf RNA- oder Proteinlevel geben.

- Varianten sollen in Relation zu einer allgemeingültigen (öffentlich zugänglichen) Referenzsequenz beschrieben werden.

- 3'Regel - für alle Annotationen gilt, dass die weitestmögliche 3'-Position in Relation zur Referenzsequenz verwendet wird. Dies gilt insbesondere auch für längere Wiederholungen von Residuen in der Referenz oder Tandem-Repeats. Sie ist für die Beschreibung auf allen Leveln

(Genom, Gen, Transkript, Protein) gültig und muss jeweils pro Level angewandt werden.

- Es sollen nur bestätigte HGNC (Gray u. a., 2015) Gensymbole verwendet werden.

Nummerierung von Varianten

Genomische Koordinaten (Chromosomen, Scaffolds, ...) werden mit einem kleinen `g` eingeleitet: `g.1`, `g.2`, `g.3` .... Die Nummerierung erfolgt durchgängig vom ersten bis zum letzten Nukleotid. Für nicht-codierende Sequenzen wird vom ersten Nukleotid des ersten Exons, wie in Abbildung 2.3 und Tabelle 2.1 gezeigt, innerhalb der kompletten Exonstruktur fortlaufend nummeriert (`n.123`). Die kompliziertere Struktur der protein-codierenden Transkripte spiegelt sich auch in der HGVS-Nummerierung wider. Codierende Sequenzen werden fortlaufend nummeriert (`c.1,c.2,...`). Dies erfolgt ab dem ersten Nukleotid des Startcodons (`ATG`/Methionin (`Met`)) bis zur letzten Position des Stopcodons (`TAG`, `TGA`, `TAA`). Die Positionen in der 5'UTR werden, im Gegensatz zu nicht-codierenden Transkripten, bei denen diese Zählweise schon vor dem ersten Nukleotid des ersten Exon genutzt wird, mit einem vorgestellten '`-`' hochgezählt. Das Gleiche gilt für die 3'UTR. Hier wird ein '`*`' vor die aufsteigenden Positionen gestellt. Nach dem letzten Exonnukleotid wird in der intragenischen Region mit einem vorgestellten '`*`' gezählt. Intronische Bereiche werden mit der letzten Exonposition, einem '`+`/'`-`, je nach Sichtweise zum nächsten Exon in `5'` oder `3'` Richtung und der Anzahl der Nukleotide zu diesem Exon angegeben: `n.70+11`, `c.51-9`. Nukleotide in Introns in der 5'UTR codierender Transkripte würden folgendermaßen nummeriert: `c.-12+1`, `c.-12+2`, ..., `c.-11-2`, `c.-11-1`. Solche in Introns in der 3' UTR: `c.*78+1`, `c.*78+2`, ..., `c.*79-2`, `c.*79-1`. Das Ganze ist, wie eingangs erwähnt, auch in Abbildung 2.3 und Tabelle 2.1 nochmals veranschaulicht.

Nukleotid Nummerierung nach HGVS

Genbereich		Nukleotid Nummerierung auf der Referenzsequenz			
		genomisch	***nicht-codierend***	***codierend***	**Protein**
5' flankierende Region		1 ... 1000	-1000 ... -1	-1020 ... -21	-
Exon1	5'UTR	1001 ... 1020	1 ... 20	-20 ... -1	-
	CDS	1021 ... 1070	21 ... 70	1 ... 50	1 ... 16 (17)
Intron1		1071 ... 1150	70+1 ... 70+80 71-80 ... 71-1	50+1 ... 50+80 51-80 ... 51-1	-
Exon2		1151 ... 1170	71 ... 90	51 ... 70	17 ... 23 (24)
Intron2		1171 ... 1200	90+1 ... 90+20 91-20 ... 91-1	70+1 ... 70+80 71-80 ... 71-1	-
Exon3	CDS	1201 ... 1226	91 ... 116	71 ... 96	24 ... 31
	3'UTR	1227 ... 1240	117 ... 130	*1 ... *13	-
3' flankierende Region		1241 ... 1400	*1 ... *160	*14 ... *174	-

Tabelle 2.1: Dies ist eine vereinfachte Adaption der Idee aus der Tabelle zum Abschnitt *Reference sequence DNA-level* von der Internetseite: `http://www.hgvs.org/mutnomen/examplesDNA.html`.
Es ist zu beachten, dass die Anzahl der Positionen im Protein um eins geringer ist, als die mögliche Anzahl der Codons in der CDS eines codierenden Gens, da das Stop-Codon nicht mitgezählt wird. Da Exons nicht in jedem Fall mit einem kompletten Codon abschließen, ist für diesen Fall die Nummer des Teilcodons in der Proteinspalte in Klammern angegeben. Für intronische Positionen sind beide möglichen Zählweisen angegeben, je nachdem ob vom Ende des davor lokalisierten Exons aufwärts oder vom Folgeexon rückwärts gezählt wird.

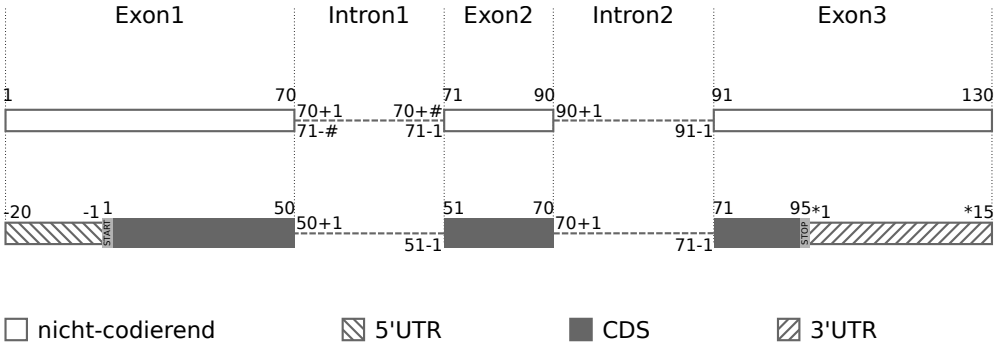


Abbildung 2.3: **Schematische Exon-Intron-Strukturdarstellung** für nicht-codierende und codierende Transkripte. Die Nummerierung entspricht den Vorgaben der HGVS-Nomenklatur (siehe auch Tabelle 2.1). # wird hier als Platzhalter verwendet und entspricht der Länge des jeweiligen Introns. Der besseren Übersichtlichkeit halber sind die Nummerierungsschemata in beide Richtungen nur im ersten Intron des nicht-codierenden Transkripts eingetragen.

Normalisierung

Anders als bei der Variantennormalisierung auf DNA-Ebene im VCF-Format, wo die Variante auf der genomischen Achse möglichst weit linksaligniert wird, werden diese entsprechend der HGVS-Nomenklatur rechtsaligniert bzw. 3' zur Ausrichtung des Transkripts. Dies alleine ist jedoch irreführend. Da sich die HGVS-Nomenklatur immer auf ein Referenztranskript und damit ein Gen bezieht, muss man die besondere Struktur der DNA beachten. Diese besteht aus einem gewundenen Doppelstrang, wobei etwa die Hälfte aller Gene in Leserichtung auf dem Referenzstrang und die andere Hälfte auf dem revers komplementären Gegenstrang codiert sind. Folglich entspricht für etwa die Hälfte der Varianten im VCF-Format (in Genen auf dem Gegenstrang) die Normalisierung auf DNA Ebene derjenigen in der HGVS-Nomenklatur.

Eine besondere Herausforderung stellt die Variantennormalisierung in der Proteinsequenz dar, da die Position nicht unbedingt mit der Position des

oder der betroffenen Codons übereinstimmen muss. In der folgenden Liste ist ein Beispiel aufgezeigt, wie sich die Positionen der DNA- und Proteinvariante im VCF-Format und in der HGVS-Nomenklatur unterscheiden können. Eine inframe-Deletion von drei Nukleinsäuren aus einer Folge von Prolin (Pro) codierenden Codons bedeutet für jedes Format eine andere Position. Für das VCF-Format (links-aligniert) liegt die Deletion (`CCC`) an Position 7-9 (Codon 3), für die cDNA im HGVS-Format an Position 18-20 (Codon 6/7) und für das Protein an Position 25-27 (Codon 9).

```
 Codon   1__2__3__4__5__6__7__8__9__10_11_
 Protein ArgSerProProProProProProProIleAsp

 REF     CGTAGTCCCCCCCCCCCCCACCACCAATAGAC
 ALT     CGTAGTCCCCCCCCCC---CACCACCAATAGAC

VCF-Format:
 REF                    CCCC                 nicht links-aligniert
 ALT                    C
 ALT          TCCC                           normalisiert
 ALT          T
missense_variant
HGVS-Format:
 cDNA:
 ALT     CGTAGTCCCCCCCCCCC---ACCACCAATAGAC   rechts-aligniert
 Protein:
 REF     ArgSerProProProProProProProIleAsp
 ALT     ArgSerProProProProProPro - IleAsp
```

Implementierung

Mutalizer (Wildeman u. a., 2008) gilt als Referenzimplementation für die teilweise sehr komplexe HGVS-Nomenklatur. Zahlreiche Programme wie Annovar (K. Wang, M. Li und Hakonarson, 2010) oder VEP (Flicek u. a., 2013; McLaren u. a., 2010) geben Annotationen im HGVS-Format aus, jedoch waren diese oft fehlerbehaftet. *Jannovar* ist eines der ersten Programme, welches die Umwandlung von Varianten vom VCF-Format in die HGVS-Nomenklatur entsprechend den Ausgaben von Mutalizer implementiert.

Variantenkategorien

Eine wichtige Funktion der Annotation von Varianten ist es, diese leichter zugänglich und weiterverarbeitbar zu machen. Ein Beispiel ist die Datenintegration in verschiedene Variantentypen zur Gruppierung und Beurteilung. Ebenso wichtig für die weitere Verarbeitung und Vergleichbarkeit ist es, die Dateninteroperabilität zu gewährleisten, also eine einheitliche und eindeutige Annotation zu verwenden. *Jannovar* verwendet zur rein technischen Beschreibung von Varianten die HGVS-Nomenklatur. Zusätzlich zu dieser Annotation werden Varianten entsprechend ihrer Lokalisation und den möglichen Auswirkungen während der Transkription, Prozessierung und Translation in verschiedene Kategorien unterteilt (siehe Tabelle 2.2 und Tabelle A1). Hierbei werden die Varianten von *Jannovar* mit den entsprechenden Termen aus der Sequence Ontology (SO) (Eilbeck und Lewis, 2004) annotiert. Jedem dieser Terme ist eine einzigartige Identifikationsnummer (Accessionnumber) zugeordnet, die aus `SO` für Sequence Ontology, einem Doppelpunkt ':' und einer siebenstelligen Nummer besteht. Nachfolgend ist exemplarisch der SO-Term zusammen mit der korrespondierenden SO-Accessionnumber für eine nicht-synonyme Variation (*engl.* missense variant) gezeigt:

`missense_variant SO:0000001`

Eine schematische Darstellung aller von *Jannovar* verwendeten SO-Terme, im Bezug zur Lokalisation auf den funktionellen Einheiten eines Transkripts, findet sich in Abbildung 2.4. Zusätzlich finden sich Beispiele für alle Variantenkategorien, die von *Jannovar* annotiert werden, im Anhang in Tabelle A1.

Intergenische Varianten

Für Varianten im intergenischen Bereich gibt es keine entsprechende HGVS-Annotation. *Jannovar* annotiert diese jedoch ebenfalls. Als Referenzpunkt

Übersicht der zur Annotation verwendeten SO-Terme

SO-Terme	**SO Accessionnumber**	**Möglicher Effekt**
start_lost	SO:0002012	High
stop_gained	SO:0001587	High
stop_lost	SO:0001578	High
complex_substitution	SO:1000005	High
mnv	SO:0002007	High
frameshift_elongation	SO:0001909	High
frameshift_truncation	SO:0001910	High
frameshift_variant	SO:0001589	High
splice_acceptor_variant	SO:0001574	High
splice_donor_variant	SO:0001575	High
missense_variant	SO:0001583	Moderate
inframe_insertion	SO:0001821	Moderate
disruptive_inframe_insertion	SO:0001824	Moderate
inframe_deletion	SO:0001822	Moderate
disruptive_inframe_deletion	SO:0001826	Moderate
synonymous_variant	SO:0001819	Low
non_coding_transcript_exon_variant	SO:0001792	Low
non_coding_transcript_intron_variant	SO:0001970	Low
5_prime_utr_exon_variant	SO:0002092	Low
3_prime_utr_exon_variant	SO:0002089	Low
5_prime_utr_intron_variant	SO:0002091	Low
3_prime_utr_intron_variant	SO:0002090	Low
stop_retained_variant	SO:0001567	Low
initiator_codon_variant	SO:0001582	Low
splice_region_variant	SO:0001630	Low
upstream_gene_variant	SO:0001631	Modifier
downstream_gene_variant	SO:0001632	Modifier
direct_tandem_duplication	SO:1000039	Modifier
intergenic_variant	SO:0001628	Modifier

Tabelle 2.2: Übersicht der SO-Terme, SO-Accessionnumbers und dem möglichen Effekt, wie sie von *Jannovar* zur Annotation von Varianten verwendet werden.

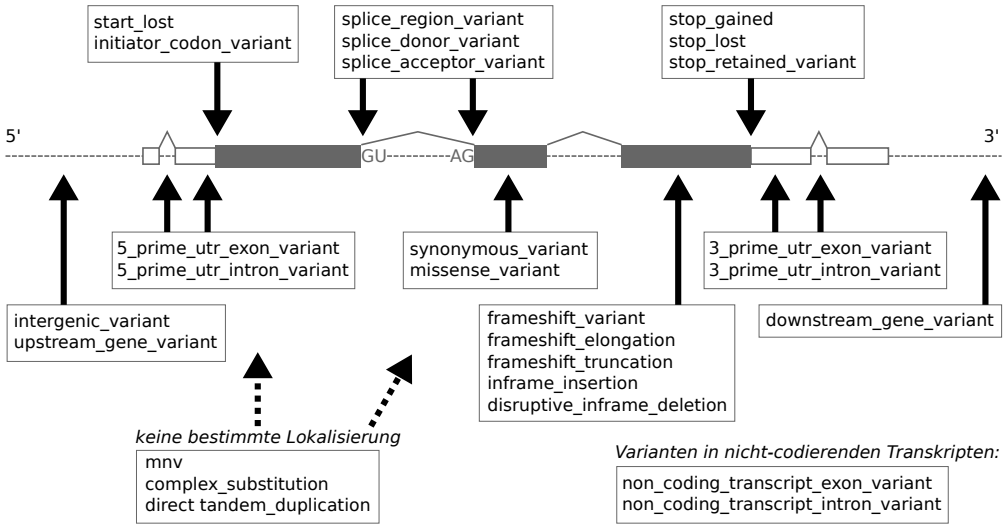


Abbildung 2.4: **Lokalisation der Varianten**. Lokalisation der von *Jannovar* mit SO-Termen annotierten Varianten. Exons sind als Blöcke dargestellt, welche durch Introns (gestrichelt und mit einem Dach die Exons verbindend) unterbrochen werden. In Grau ist der protein-codierende Bereich des Transkripts dargestellt, der weiße Bereich entspricht der UTR. Für das erste Intron im codierenden Bereich wurden Spleiß-Donor (`GU`) und -Akzeptor (`AG`) markiert.

gilt in diesem Fall der Abstand zum nächstgelegenen Gen. Dabei wird dasjenige prozessierte Transkript dieses Gens referenziert, welches am nächsten heranreicht. Zu diesem Referenzpunkt kommt die Annotation nach SO hinzu. Entsprechend Tabelle 2.2 gibt es drei mögliche SO-Variantentypen: `intergenic_variant`, `upstream_gene_variant` und `downstream_gene_variant`. Eine Differenzierung hängt von der Position zum Gen ab. Eine `intergenic_variant` kann vor oder hinter einem Gen liegen, ist von diesem jedoch mindestens 5 000 bp weit entfernt. Eine Variante innerhalb der 5 000 bp Bande wird entsprechend ihrer Position als `upstream_gene_variant`, für eine Lokalisation 5' vor einem Gen beziehungsweise `downstream_gene_variant` für 3' hinter einem Gen entsprechend der genomischen Achse annotiert.

UTR Varianten

Varianten können in der 5'UTR und 3'UTR sowohl in exonischen als auch intronischen Regionen auftreten. Wie eingangs erwähnt, wird für die 5'UTR die Position der Varianten als Abstand zum Start-Codon mit einem vorgestellten '−' angegeben. Die dbSNP Variante `rs3128113`, eine Transition von `A>G` an Position 935 459 auf Chromosom 1 entspricht einer `5_prime_utr_exon_variant` des Transkripts `NM_001142467.1` 106 Nukleotide vor dem Start-Codon. Die entsprechende Transkriptannotation ist: `NM_001142467.1:c.-106T>C`. Eine `5_prime_utr_intron_variant` im 5'UTR Intron würde zuätzlich mit dem Abstand zum nächsten Exon annotiert werden, die genomische Variante `chr1:g.11741304A>C` entsprechend zur Transkriptvariante `NM_001127325.1:c.-12-634T>G`. Analog werden die Annotationen in der 3'UTR gebildet, wobei hier der Abstand zum Stop-Codon mit einem vorgestellten `*` zur Nummerierung dient. Ein Beispiel für eine `3_prime_utr_exon_variant` wäre die genomische Variante `chr1:g900730G>A`, welche der Transkriptvariante `NM_198317.2:c.*159G>A` entspricht, und `3_prime_utr_intron_variant NM_005694.1:c.*5-154T>C` für eine intronische 3'UTR Variante steht.

CDS Varianten

Varianten im codierenden Bereich eines Gens können sich in den unterschiedlichsten funktionalen Bereichen befinden. Die HGVS-Nomenklatur beschreibt eine Substitution oder Deletion unabhängig von der Lokalisation. Für die SO-Terminologie ist die Position und mögliche Auswirkung auf die Translation von Bedeutung, so dass diese eine einfache Substitution je nach Lage ganz unterschiedlich annotiert werden kann. Die Breite der möglichen Auswirkungen reicht hier von in der Regel fast keinem Effekt (synonyme Variante) bis zu ernsthaften (Start-/Stop-Verlust) Effekten auf die Translation.

`synonymous_variant`

Diese wird auch stumme Variante genannt. Der Austausch eines Nukleotides innerhalb eines Codons führt nicht zur Änderung der codierenden Aminosäure. Eine interessante Variante ist `NM_015658.3:c.1843C>T`, bei der nicht die Wobble-Base betroffen ist. Als stumme Variante führt dies dennoch zu keinem Austausch der Aminosäure in der Translation, so dass die HGVS-Annotation `p.(=)` ist.

`missense_variant`

Diese nicht-synonyme Variante ist per Definition die Veränderung von mindestens einem Nukleotid, so dass es zu einem Austausch der codierenden Aminosäure kommt. Die Anzahl der Nukleotide wird dabei nicht verändert. Dies wird typischerweise durch die Transversion oder Transition einzelner Nukleotide verursacht. Ein typisches Beispiel für eine Transversion `chr1:g.883899T>G` in NA12878 ist der dbSNP Eintrag `rs72631890`. Dieser führt zum Austausch `c.1528A>C` im Transkript `NM_015658.3` von Gen NOC2L, resultierend im Aminosäureaustausch `p.(Asn510His)`.

`start_lost`

Die Substitution eines Nukleotids im Start-Codon hat zum Verlust dessen geführt und fatale Auswirkungen auf die Translation des Transkripts. Ohne das Start-Codon kann diese nicht initiiert werden und es wird kein Protein als Genprodukt produziert. Ein Beispiel hierfür ist die das Transkript `NM_015700.3` betreffende genomische Variante `chr2:69659126A>T`, welche als `missense_variant` (`c.2T>A`) das Start-Codon (`p.0?`) zerstört.

`initiator_codon_variant`

Eine Codon-Variante, welche mindestens ein Nukleotid des transkriptioninitiierenden Codons modifiziert.

`stop_lost`

Eine Variante innerhalb des Stop-Codons, die zum Verlust derselbi-

gen führt. Während der Tanslation kommt es zu keinem Abbruch, was wiederum zu einem verlängerten Protein mit einem unbestimmten Ende führt. So führt die Transition in `NM_001004689.1:c.937T>A` zum Verlust des Stop-Codons `p.(*313Argext*?)`. Es ist ungewiss, wann in diesem Fall ein funktionales Stop-Codon in der Sequenz auftaucht.

`stop_gain`

bezeichnet die Einführung eines zusätzlichen Stop-Codons in die codierende Sequenz eines Transkripts. Dies kann durch komplexe Varianten oder auch einfache SNVs verursacht werden. Die Substitution `NM_023013.2:c.314T>A` ist eine Nonsense-Variante und führt ein neues Stop-Codon `p.(Leu105*)` anstelle des Leucins an Position 105 ein. Da es zu diesem Codon keine entsprechende Aminosäure gibt, wird es auch Nonsense-Codon genannt.

`stop_retained_variant`

Für das translationale Stopsignal existieren drei (`UAA`, `UAG`, `UGA`) Codons, wobei es mehrere Möglichkeiten gibt, diese durch eine einzelne Substitution ineinander zu überführen. Eine `stop_retained_variant` ist demnach eine `synonymous_variant` innerhalb des Stop-Codons.

`mnv`

Eine Multinukleotidvariante ist ein mikro *InDel*[4] (Krawitz u. a., 2010). Sie betrifft eine kontinuierliche Folge von mindestens zwei Nukleotiden in der Referenz und wird durch ebenfalls mindestens zwei Nukleotide ersetzt. Sie wird als komplexes Rearrangement betrachtet, also der Kombination von elementaren Veränderungen (Substitution, Deletion, Duplikation, Insertion, Inversion, Translokation). Eine Deletion von fünf Nukleotiden aus der codierenden Sequenz an Position 78 bis 84, kombiniert mit der Insertion von drei Nukleotiden `ATT`, lässt sich nach HGVS-Nomenklatur als `c.78_84delinsATT` oder auch in der

[4]Kunstwort aus Insertion und Deletion

Langform `c.78_84delGCGCGinsATT` ausdrücken.

Frameshift Varianten

Wird in der codierende Sequenz kein Vielfaches von drei Nukleotiden deletiert oder eingefügt, wird der offene Leserahmen unterbrochen und es hat Auswirkungen auf die Translation der weiteren Codons. Die `frameshift_variant rs36013100` (`chr1:g.54605319G>GC`) ist eine einfache Insertion `NM_201546.3:c.1223_1224insG` auf Transkriptebene. In das Protein wird 45 Aminosäuren nach dem betroffenen Codon ein neues Stop-Codon (`*`) eingeführt: `p.(Met409Hisfs*45)`. Sind der Frameshift und das frühere Stop-Codon durch eine Deletion verursacht, so ist dies eine `frameshift_truncation`. Im umgekehrten Fall einer Insertion und einem relativ zur Referenz späteren Stop-Codon ist es eine `frameshift_elongation`.

Inframe Deletion

Eine Deletion, welche den ORF in der codierenden Sequenz erhält, kann entweder ein (`p.(Leu35del)`) oder mehrere Codons (`p.(Pro369_Gly394del)`) komplett entfernen (`inframe_deletion`). Beginnt die Deletion in einem Codon, so werden zwei Codons fusioniert und bei mehr als drei Nukleotiden werden ganze Codons deletiert (`disruptive_inframe_deletion`). Die Länge des entsprechenden Proteins verringert sich hierbei um die Anzahl der deletierten Codons.

Inframe Insertion

ORF erhaltende Insertionen können direkt an der Codongrenze Nukleotide insertieren (`inframe_insertion`). Duplikationen in der Nukleotidssequenz (`NM_012093.3:c.1603_1605dup`) führen hierbei auch zu Duplikationen im Protein (`p.(Ile535dup)`), andere Insertionen (`NM_001197234.2|c.42_43insTTCCTCCTC`) in der Regel auch zu Insertionen von neuen Aminosäuren (`p.(Ser14_Leu15insPheLeuLeu)`). Dabei gibt es Fälle, in denen die Codon-zu-Aminosäure-Zuordnung

zu einer Duplikation führen. Werden die Nukleotide innerhalb eines Codons eingefügt, so ist dies eine `disruptive_inframe_insertion`.

Intron Varianten

Wie im Abschnitt „Nummerierung von Varianten“ beschrieben, werden Varianten in Introns zwischen codierenden Exons (`coding_transcript_intron_variant`) entsprechend des Abstands zum am nächsten gelegenen Exon in 5’ (`NM_198317.2:c.1700+18G>C`) oder 3’ (`NM_015658.3:c.180-92C>T`) Leserichtung nummeriert.

Spleißvarianten

Protein-codierende Transkripte unterlaufen in der Regel einem Prozessionsschritt vor der Translation, bei dem die Introns (siehe auch Abbildung 2.3) aus dem Primärtranskript herausgeschnitten werden – dem sogenannten Spleißen. Dies geschieht an hoch konservierten Erkennungssequenzen, die den Übergang vom Exon zum Intron markieren. Beim Menschen findet man fast ausschließlich `GU-AG`-Introns, bei denen die Donor-Seite 5’ im Intron durch ein Guanin+Uracil eingeleitet wird und die letzten beiden Nukleotide des Introns 3’ Adenin+Guanin als Akzeptor dienen. Varianten in der unmittelbaren Umgebung der Spleißgrenzen sowohl im intronischen als auch exonischen Bereich werden Spleißvarianten genannt. Die Grenzen reichen hier vom dritten Nukleotid im Exon bis zum achten Nukleotid im Intron. Eine `splice_donor_variant` verändert mindestens eine der ersten beiden intronischen Nukleotide (Spleißdonor). Am anderen Ende des Introns betrifft eine `splice_acceptor_variant` die letzten beiden Nukleotide, den Spleißakzeptor. Eine Variante, die noch innerhalb der Spleißgrenzen liegt wird als `splice_region_variant` annotiert. Als Spleißgrenzen sind hier die ersten drei Nukleotide des Exons und die Nukleotide drei bis acht des Introns definiert.

Varianten in nicht-codierenden Transkripten

Im Gegensatz zu dem dreigeteilten funktionalen Aufbau (5'UTR, CDS, 3'UTR) von codierenden Transkripten gibt es bei nicht-codierenden Transkripten nur einen funktionalen Bereich. Aus diesem Grund wird für `non_coding_transcript_variant` (`SO:0001619`) Varianten die Nummerierung schon mit dem ersten transkribierten Nukleotid begonnen. Das `c` wird durch ein `n` für non-coding (nicht-codierend) ersetzt. Mögliche Variantentypen sind `non_coding_transcript_exon_variant` und `non_coding_transcript_intron_variant`, auch in Kombination mit `splice_region_variant`.

2.2.4 Transkripte und Datenbanken

Parallel mit dem Humangenomprojekt entstand das Bedürfnis, vorhandenes Wissen über Gene und Transkripte in Datenbanken festzuhalten. Insbesondere mit der Veröffentlichung des ersten humanen Genomreleases als Referenzpunkt zur einheitliche Beschreibung der Strukturen auf dem Genom entstanden mehrere Initiativen, um dies zu realisieren. So haben sich im Verlauf der letzten Dekaden eine Reihe unterschiedlicher Transkriptdatenbanken etabliert. Einige sind schon älteren Datums, aber die meisten entstanden um die Jahrtausendwende und damit bevor das erste offizielle humane Genomrelease veröffentlicht wurde.
Die wichtigsten und am häufigsten verwendeten Datenbanken, welche auch als Referenz für HGVS dienen können, sollen hier Erwähnung finden. Historisch gesehen haben alle diese Datenbanken ihre Daseinsberechtigung und eine sehr große gemeinsame Schnittmenge, unterscheiden sich aber durch die unterschiedlichen Ansätze sowohl im Umfang, als auch in ihrer Evidenz. *Jannovar* kann nativ diese drei Datenbanken zur Annotation verwenden.

Ensembl (Aken, Achuthan u. a., 2017; Aken, Ayling u. a., 2016) ist ein Gemeinschaftsprojekt des European Bioinformatics Institute (EMBI-EBI) und dem Wellcome Trust Sanger Institute (WTSI), angesiedelt in Hinxton

(Vereinigtes Königreich).

RefSeq ist die Reference Sequence Database (Referenzsequenz Datenbank) des NCBI (Nationales Zentrum für Biotechnologieinformation) und stammt aus Bethesda, MD (USA).

UCSC ist die Datenbank der University of California, Santa Cruz (UCSC), aus Santa Cruz, CA (USA).

Alle drei Datenbanken beziehen ihre Basisinformationen und Annotationen aus der International Nucleotide Sequence Database Collaboration (INSDC). Das INSDC ist ein Zusammenschluss von drei großen Nukleotid-Datenbanken, namentlich das European Nucleotide Archive (EBI), GenBank (NCBI) und DDBJ, und hat das Ziel, eine öffentliche Referenzdatenbank für DNA und RNA-Sequenzen bereitzustellen, die weltweit von Forschern aktualisiert und genutzt werden kann. Um den Nutzern Konsistenz zu bieten, führen alle INSDC-Datenbanken täglich untereinander eine Synchronisation aus. Da diese Referenzdatenbank eine Archivdatenbank ist, kann sie viele redundante Datensätze zu dem gleichen Locus (Gen oder auch Transkript) enthalten. Aus diesem Grund haben alle Transkriptdatenbanken ihre eigene Herangehensweise, die INSDC-Daten zu filtern und zusammenzuführen, um ein repräsentatives Referenz-Subset zu erhalten. Im Folgenden findet sich eine kurze Zusammenfassung für die drei erwähnten Datenbanken.

Ensembl/GENCODE[5] Ein Gen fasst alle Transkriptvarianten mit überlappender codierender Sequenz[6] zusammen. Die Transkriptannotationen entstammen entweder dem automatisierten Ensembl-Annotationsprozess, Havana/Vega[7] (J. L. Harrow u. a., 2014) oder dem Consensus Coding Se-

[5]GENCODE ist Teil des ENCODE-Projekts und hat zum Ziel eine „Enzyklopädie aller Gene und Genvarianten“ aufzubauen(J. Harrow, Denoeud u. a., 2006; J. Harrow, Frankish u. a., 2012).

[6]Es gibt einige wenige handkurierte Fusions-Gene, die trotz überlappender codierender Sequenz als separate Gene definiert sind.

[7]http://www.sanger.ac.uk/science/groups/vertebrate-annotation

quence (CCDS)-Projekt[8] (Pruitt u. a., 2009).

RefSeq Sequenzen sind nicht Teil der INSDC-Datenbank, sind aber von INSDC-Sequenzen (Genbank) abgeleitet und stellen einen nicht-redundanten, kurierten Datensatz dar, der unser heutiges Verständnis von bekannten Genen repräsentieren soll. Einige Einträge enthalten Sequenzinformationen (beschreibende Informationen, Publikationen, Eigenschaften), die nicht in einem einzelnen INSDC-Datensatz gefunden werden können, sondern aus mehreren zusammengestellt sind.

UCSC *known genes* (Hsu u. a., 2006) basieren auf den Proteinen in der Swiss-Prot/TrEMBL (UniProt) Datenbank und den dazu gehörigen Transkriptdaten in GenBank/INSDC. Im Sommer 2016 hat UCSC seine Datenbank angepasst und verwendet nun für GRCh38 ebenfalls GENCODE als Grundlage und ist damit zu Ensembl identisch[9].

GENCODE enhält hierbei die hochwertigsten Annotationen (Frankish u. a., 2015). Am Beispiel von UCSC wird deutlich, wie die Datensätze in den einzelnen Datenbanken vereinheitlicht werden. Dies hat nicht nur für die Bioinformatiker viele Vorteile, sondern ermöglicht allen Forschern, die mit diesen Datenbank arbeiten, eine einheitliche Referenz und damit Vergleichbarkeit ihrer Arbeit.

2.2.5 Transkripte als Intervalle – der Intervallbaum

Die erste und wichtigste algorithmische Aufgabe bei der genomischen Annotation ist die Identifizierung aller genomischer Einheiten (z.B. Gene, Transkripte, ...), die für die Fragestellung von Bedeutung sind und mit einer Variante überlappen. Dies ist insbesondere durch die sehr komplexe Struktur der Gene und der Transkripte (variable Exon und Intron Kombination) bei Eukaryoten eine Herausforderung (Abbildung 2.5). Überlappende Gene und Gene innerhalb intronischer Regionen eines anderen Gens sind eine zusätzliche Hürde bei der effizienten Bestimmung der betroffenen genomi-

[8] https://www.ncbi.nlm.nih.gov/CCDS/

[9] http://genome.ucsc.edu/blog/new-default-gene-set-on-grch38-gencode-basic-genes/

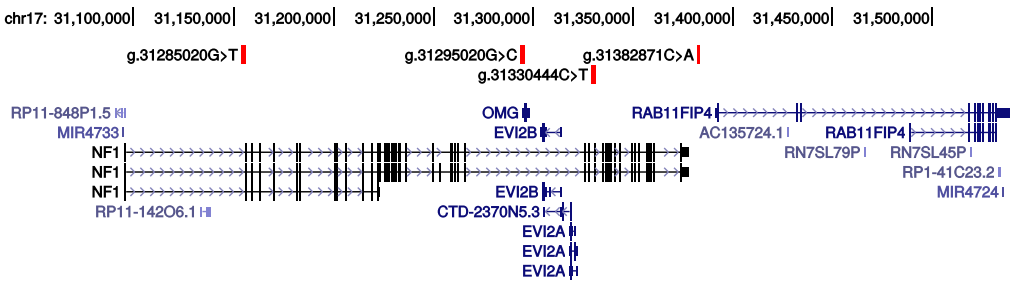


Abbildung 2.5: Gene und Transkripte über rund 480 000 bp entlang der genomischen Achse von Chromosom 17 aus dem GRCh38 Genomrelease. Transkripte werden als Intervalle mit unterschiedlichen Schraffierungen für die Exons (Balken) und Introns (Linie) angezeigt. Die Richtung der Transkripte (5' zu 3' UTR) beziehungsweise der codierende Strang für das Gen wird durch Pfeile in den Introns markiert. Rot markiert, oberhalb der Transkripte, sind vier Einzelbasenvarianten. Diese Abbildung veranschaulicht die Herausforderungen bei der Annotation von genomischen Varianten. Eine Variante kann mit mehreren Transkripten eines Gens oder auch mit mehreren Genen überlappen. Die Variante `chr17:g.31330444C>T` führt zum Beispiel zu einer nicht-synonymen Substitution in zahlreichen Transkriptvarianten (`ENST00000356175.7|ENST00000358273.8|ENST00000431387.8`) des NF1 Gens. Eine andere Variante `chr17:g.31295020G>C` überlappt mit zwei Genen und deren Transkiptvarianten (`OMG:ENST00000247271.4`, `NF1:ENST00000356175.7|ENST00000358273.8`). Diese Abbildung wurde mit Hilfe des UCSC Genome Browsers (Tyner u. a., 2017) erstellt.

schen Einheiten. Hierfür werden genomische Annotationen aus der VCF-Datei, wie zum Beispiel `chr11:g.1836518C>G`, in gen- bzw. transkriptbasierte Annotationen folgender Form umgewandelt: `SYT8:NM_001290332.1:c.655C>G:p.(Leu219Val)`.

Ein trivialer Ansatz wäre über alle Intervalle zu iterieren und auf eine Überlappung mit der Variante zu testen. Dies ließe sich in einer Laufzeit von $O(n)$, mit n gleich der Anzahl der Intervalle realisieren. Durch eine Sortierung der Start- und Endkoordinaten der Intervalle ließe sich dies noch beschleunigen. Im genomischen Kontext ist die Fragestellung jedoch komplizierter. Ausgehend von der Position der Variante werden neben den Start-

oder Endkoordinaten der überlappenden Transkripte gegebenenfalls auch die der am nächsten lokalisierten Transkripte gesucht. Da es, wie in Abbildung 2.5 gezeigt, sehr lange Gene (NF1) gibt, die weitere (OMG, EVI2A, ...) überspannen, welche nicht mit der Variante überlappen, kann man hieraus kein Abbruchkriterium für die Suche konstruieren. Ausgehend von der Variante `g.31295020G>C` würde man bei einer Suche entlang der genomischen Achse zuerst auf OMG, anschließend auf EVI2B, ... und erst ziemlich spät auf NF1 stoßen. Um die Suche nach allen überlappenden Features (bzw. Gene/Transkripte/...) mit einer bestimmten genomischen Position effektiv zu gestalten, eignet sich der Intervallbaum (Berg u. a., 2008). *Jannovar* verwendet diesen zur Implementation der Suche nach überlappenden Transkripten zu einer Variante.

2.3 Algorithmus

2.3.1 Aufbau des Intervallbaums

Ein Intervallbaum lässt sich, wie in Abbildung 2.6 und dem Pseudocode in Algorithmus 1 skizziert, in einer Laufzeit von $O(n \log n)$ aufbauen, wobei n die Anzahl der Intervalle $i \in \mathcal{I}$ (in diesem Fall Transkripte) ist (ebd.). Genomische Features in Form von Transkripten (A) lassen sich in Intervallen abstrahieren (B). Diese n Intervallpaare von Koordinaten, den genomischen Start- und Endpunkten der Transkripte, werden in einem ersten Schritt nummerisch sortiert. Mit Hilfe einer rekursiven Funktion (BUILD-NODE), die in Algorithmus 1 als Pseudocode dargestellt ist, werden nun die einzelnen Knoten und Blätter des Intervallbaums aufgebaut (C). Hierfür wird der Median (x_{median}) aller $2n$ Endpunkte der Intervalle in $\mathcal{I}$ berechnet. Die Menge aus Intervallen kann nun in drei Gruppen unterteilt werden, solche die mit x_{median} überlappen ($\mathcal{I}_{middle}$), jenen die auf der genomischen Achse komplett vor x_{median} liegen ($\mathcal{I}_{left}$) und denen, die hinter x_{median} liegen ($\mathcal{I}_{right}$). Im folgenden Schritt erzeugt man einen Knoten n (engl. *node*), aus dem der Intervallbaum aufgebaut wird. Dieser Knoten enthält die Intervalle

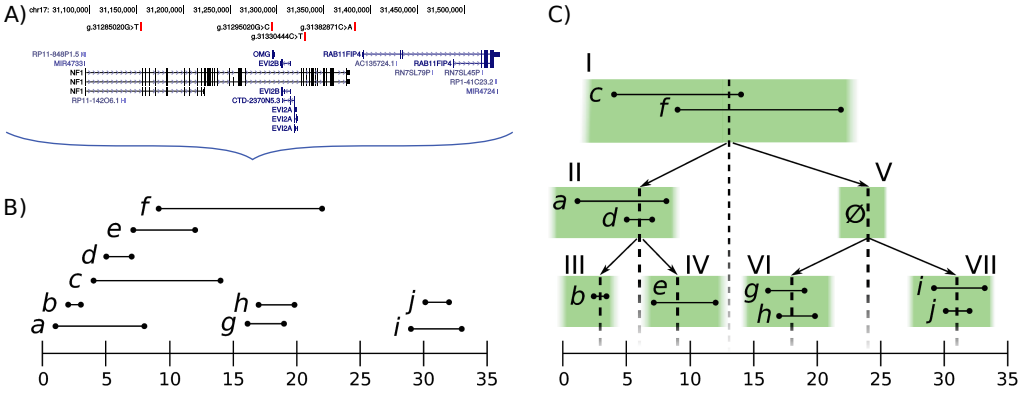


Abbildung 2.6: A) Ein Ausschnitt aus dem UCSC Genome Browser, der auch schon in Abbildung 2.5 zu sehen ist. B) Transkripte lassen sich abstrakt generalisiert darstellen. Sie werden dabei als durchgängige Intervalle von ihrem Start- bis zum Endpunkt angezeigt. C) Zeigt den Intervallbaum, der sich aus den Intervallen in B) aufbauen lässt. Die grünen Kästen stellen die Knoten und Blätter im Baum dar und können keinen (V), ein einzelnes (III,IV) oder mehrere (I,II,VI,VII) Transkriptintervalle enthalten.

aus $\mathcal{I}_{middle}$, wobei $\mathcal{I}_{middle}$ auch eine leere Menge sein kann und der Knoten damit ebenfalls leer wäre (siehe auch Knoten V in Abbildung 2.6). Um die Suche innerhalb des Knotens zu optimieren, werden die Startkoordinaten oder auch linken Endpunkte (LEP) entlang der genomischen Achse der enthaltenen Intervalle aufsteigend sortiert. Umgekehrt werden die Endkoordinaten oder rechten Endpunkte (REP) in absteigender Reihenfolge sortiert. Durch den rekursiven Aufruf der Knotenbaufunktion wird ein binärer Baum konstruiert.

Für die Implementation in *Jannovar* wurde für jedes Chromosom ein separater Intervallbaum gewählt. Alternativ ist es möglich einen einzelnen Baum aus der Gesamtsequenz aller Autosomen, Gonosomen und der mitochondrialen DNA aufzubauen. In diesem Fall müssen dann die relativen Koordinaten der Intervalle in die absoluten der Gesamtsequenz umgewandelt werden.

Algorithmus 1 Der Algorithmus zum Aufbau des Intervallbaums als Pseudocode. BUILD-NODE konstruiert einen Knoten des Baums und ruft sich rekursiv auf um Töchterknoten zu erzeugen. Hierzu werden die Intervalle in drei Gruppen unterteilt: I. komplett links vom Median der Start- und Endpunkte der Intervallen ($\mathcal{I}_{left}$), komplett rechts vom Median ($\mathcal{I}_{right}$) und denen, die mit dem Median überlappen ($\mathcal{I}_{middle}$). Die Intervalle links und rechts vom Median ($\mathcal{I}_{left}$ & $\mathcal{I}_{right}$) werden zur rekursiven Konstruktion der Töchterknoten verwendet, während die überlappenden Intervalle $\mathcal{I}_{middle}$ diesem Knoten zugeordnet werden.

BUILD-NODE($\mathcal{I}$)
1 Sort the set of $2n$ endpoints $\{e_1, e_2, \ldots, e_{2n}\}$ of the n intervals $i \in \mathcal{I}$
2 x_{median} = median($\{e_1, e_2, \ldots, e_{2n}\}$)
3 Divide all intervals $i \in \mathcal{I}$ into $\mathcal{I}_{\text{left}}$, $\mathcal{I}_{\text{middle}}$, and $\mathcal{I}_{\text{right}}$
4 Node n = new Node($\mathcal{I}_{\text{middle}}$) **// Construct new node**
5 n.median = x_{median}
6 n.LEP = sort $\mathcal{I}_{\text{middle}}$ by increasing left end point
7 n.REP = sort $\mathcal{I}_{\text{middle}}$ by decreasing right end point
8 n.left = BUILD-NODE($\mathcal{I}_{\text{left}}$)
9 n.right = BUILD-NODE($\mathcal{I}_{\text{right}}$)

2.3.2 Abfrage des Intervallbaums

Ebenso wie für den Aufbau des Baums wird für die Suche nach überlappenden Intervallen zu einer Anfrage $s = [s.lo, s.hi]$ eine rekursive Funktion verwendet.

Abfrage einer Einzelposition (SNV, Insertion) Nachfolgend wird von dem Fall ausgegangen, dass Start- ($s.lo$) und Endpunkt ($s.hi$) einer Anfrage s identisch sind [10]. Aus der Konstruktion des Intervallbaums wissen wir, dass jedes Intervall in einem Knoten n mit $n.median$ überlappt und die Intervalle in $n.LEP$ aufsteigend anhand der Startpunkte und in $n.REP$ absteigend entsprechend der Endpunkte sortiert sind. Für die Su-

[10] Dies wäre zum Beispiel bei SNVs und Insertionen der Fall.

Algorithmus 2 Der Algorithmus zur Suche im Intervallbaum als Pseudocode. SEARCH-NODE findet alle überlappenden Intervalle zu einer Suchanfrage s im Knoten n und ruft rekursiv die Suche in den Töchterknoten auf. Da per Definition alle Intervalle im Knoten n mit $n.median$ überlappen, ist die erste Anfrage trivial, erspart jedoch die Iteration über die sortierten Endpunktlisten.

```
SEARCH-NODE(n, s)
 1  // Search intervals {i_1, i_2, ...} stored in current node
 2  if s.lo == n.median || s.hi == n.median
 3        All intervals in n overlap s
 4  else
 5        if s.hi < n.median
 6              Search intervals in n.LEP until i_j.lo > s.lo for some j
 7        if s.lo > n.median
 8              Search intervals in n.REP until i_j.hi > s.hi for some j
 9  // Search in children nodes as appropriate
10  if s.hi < n.median
11        eliminate the right subtree beginning with n.right
12  else SEARCH-NODE(n.right, s)
13  if s.lo > n.median
14        eliminate the left subtree beginning with n.left
15  else SEARCH-NODE(n.left, s)
```

che in dem aktuellen Knoten wird geschaut, ob $s.lo == s.hi < n.median$ gilt. Denn es ist bekannt, dass alle Intervalle in n dann hinter s enden (siehe auch Abbildung 2.7). Ist dies der Fall, so müssen nur diejenigen Intervalle in $n.LEP$ bestimmt werden, die vor $s.hi$ liegen (Algorithmus 2 Zeile 2). Für den Fall $s.lo == s.hi > n.median$ muss man umgekehrt testen, welche Intervalle in $n.REP$ hinter $s.lo$ liegen (Algorithmus 2 Zeile 4). Sollte $s.lo == s.hi = n.median$ sein, so überlappen alle Intervalle in n mit dem Anfrageintervall s und $n.LEP$ bzw. $n.REP$ müssen nicht prozessiert werden. Die rekursive Suche in den Töchterknoten fällt ebenfalls weg. Im zweiten Schritt werden nun die Töchterknoten von n in Abhängigkeit der Position von $n.median$ zu $s.lo$ bzw. $s.hi$ prozessiert (siehe auch Algorith-

mus 2 Zeile 7 & 10 und Abbildung 2.7).

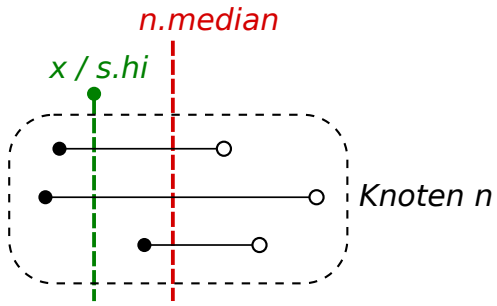


Abbildung 2.7: Alle Intervalle im Knoten n überlappen mit dem Median der Endpunkte $n.median$. Da alle Endpunkte der Intervalle dementsprechend hinter $n.median$ liegen, kann man für den Fall, dass man nur einen Punkt x kleiner als $n.median$ abfragt, sagen, dass alle Intervalle, die vor x in $n.LEP$ beginnen, mit diesem Punkt überlappen müssen. Dementsprechend gilt auch für ein gegebenes Intervall s, das in $s.hi$ endet und dieser Punkt kleiner als $n.median$ ist, dass alle Intervalle beginnend vor $s.hi$ mit dem Intervall s überlappen müssen.

Abfrage für ein Intervall (MNV, Deletion, SV) Sind Start- und Endpunkt einer Anfrage s unterschiedlich, dann gibt es drei Möglichkeiten für ein überlappendes Intervall r.

1. Start- oder Endpunkt von s liegen in r.
2. r umschließt das Anfrageintervall komplett.
3. s umschließt das Intervall r komplett.

Für Start- und Endpunkt müssen separate Suchen im Intervallbaum nach dem genannten Schema durchgeführt werden. Damit werden die in 1. & 2. beschriebenen Intervalle identifiziert. Hierbei muss beachten werden, dass für die 2. Suche Duplikate vermieden werden. Dies lässt sich mit Hilfe einer Markierung der identifizierten Intervalle bzw. Transkripte verhindern. Für

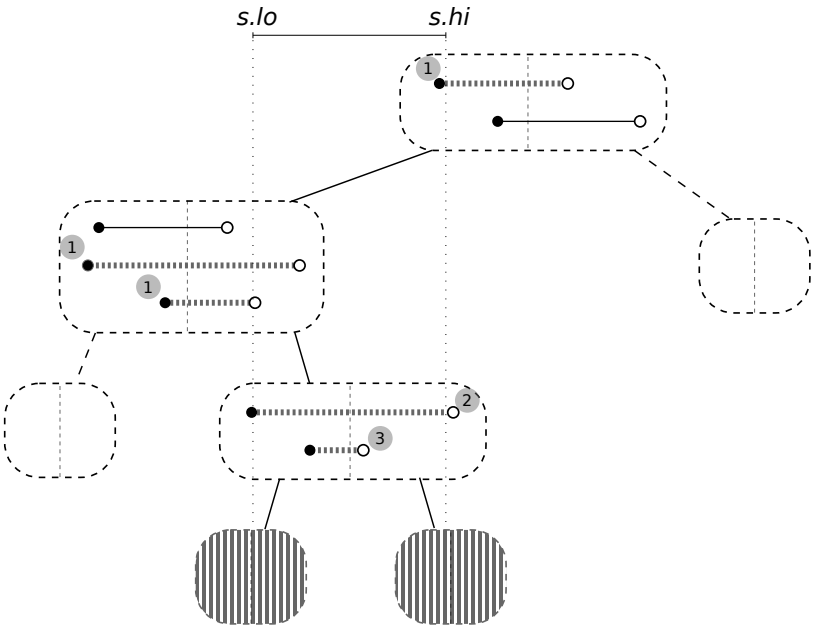


Abbildung 2.8: Schematische Darstellung eines Intervallbaums mit den Grenzen eines Abfrageintervalls (*s.lo*,*s.hi*). Gestrichelte Intervalle überlappen zumindest teilweise mit dem dargestellten Abfrageintervall (*s*). Bei der Abfrage von Intervallen statt einzelnen Punkten können folgende drei Möglichkeiten auftreten: (1) Start- oder Endpunkt des Abfrageintervalls überlappen mit Intervallen, (2) Start- und Endpunkt des Abfrageintervalls liegen beide innerhalb der Grenzen eines Intervalls oder (3) das Abfrageintervall umschließt ein Intervall komplett.

die Suche nach Intervallen, die wie in 3. komplett vom Anfrageintervall s umschlossen werden, lässt sich der Suchraum einschränken.

Wie in Abbildung 2.8 gezeigt, ist bekannt, dass alle Intervalle in den Kinderknoten zwischen dem am weitesten links liegenden Knoten, für den $s.lo \geq n.median$ gilt und jenem am weitesten rechts gelegenen Knoten mit $s.hi \leq n.median$, komplett vom Anfrageintervall s überlappt sein müssen. Im angewandten genomischen Kontext kommt dies eher selten vor. VCF-Dateien aus Exomanalysen enthalten in der Regel keine größeren (Struktur-) Varianten, sondern beschränken sich, technologiebedingt, auf SNVs und kleinere

(≤ 50) InDels.

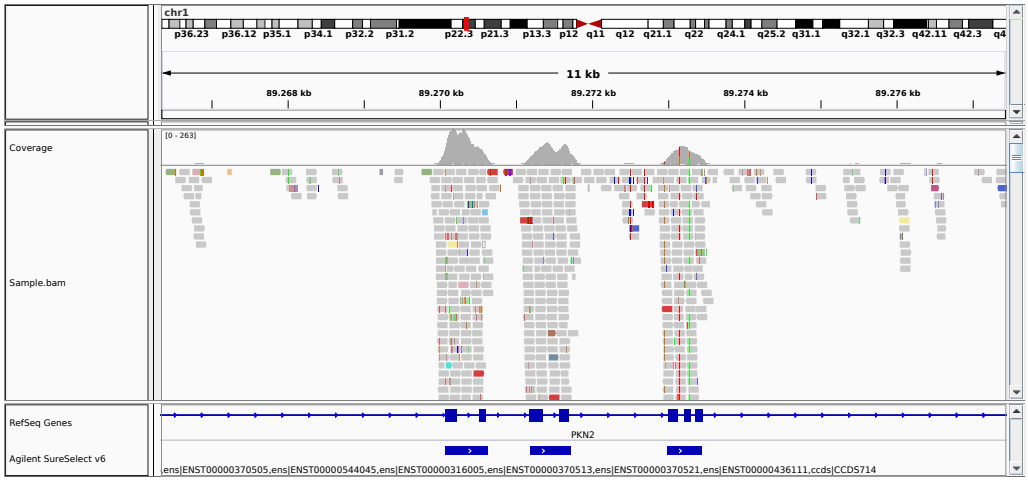


Abbildung 2.9: Gezeigt wird der Ausschnitt von sieben Exons auf Chromosom 1 im Bereich von 89 266 340-89 277 513 auf dem GRCh37 Genomrelease. Mittig sieht man die auf einem Illumina HiSeq4000 und mit BWA-MEM (Li, Heng, 2013) gemappen Reads für ein mit dem Agilent SureSelect Human All Exon V6 Targetkit angereichertes Exom. Unten sind die kumulierte Exon-Intron-Struktur für das Gen PKN2 aus der RefSeq Datenbank, wie auch die Platzierung der Sonden für die verwendete Anreicherung, dargestellt. Deutlich zu sehen ist, dass neben den Exons auch weite Teile der Introns abgedeckt und damit analysierbar sind. Diese Abbildung wurde mit Hilfe von IGV (J. T. Robinson u. a., 2011; Thorvaldsdóttir, J. T. Robinson und Mesirov, 2013) erstellt.

Erweiterte Implementierung der Intervallbaumsuche Bei der Sequenzierung des Exoms werden neben den Exons auch intronische und intergenische Bereiche angereichert und sequenziert. Bei der Auswertung liegen viele Varianten nicht innerhalb der Exons, sondern sind diesen vor- oder nachgelagert (siehe auch Abbildung 2.9). Für den Fall, dass bei der normalen Intervallbaumsuche keine überlappenden Transkripte identifiziert wurden, muss man auf eine Erweiterung der normalen Suchfunktion zurückgreifen, um die nächstgelegenen linken (5') und rechten (3') Nachbarn auf

der genomischen Achse zu bestimmen. Für eine Suchanfrage s, die mit keinem Intervall übereinstimmt, gilt es im allerersten Schritt zu schauen (das Gleiche gilt für alle Varianten), ob sie vor dem ersten oder hinter dem letzten Intervall liegt. Hierfür sortiert man in einem Vorverarbeitungsschritt die Start- beziehungsweise Endpunkte aller Intervalle analog zum Aufbau der Knoten im Intervallbaum. Mit diesen Listen ist es trivial nachzuschlagen, ob 1. s außerhalb der Intervallbaumkoordinaten liegt und eine Suche in diesem hinfällig ist und 2. lässt sich das nächstgelegene Transkript direkt ablesen. Theoretisch muss nicht einmal eine sortierte Liste der Transkripte angelegt werden, sondern es müssen nur die Extremwerte und die dazugehörigen Transkripte bekannt sein.
Liegt s nicht vor oder hinter den Intervallen des Baums, so endet eine Suche im Baum mit SEARCH-NODE in einem Blatt und alle Intervalle dieses Knotens liegen entweder vor oder hinter der Suchanfrage. Die Bestimmung der nächsten Intervalle ist nicht trivial und folgende Bedingungen müssen beachtet werden, um die korrekten Nachbarn zu bestimmen:

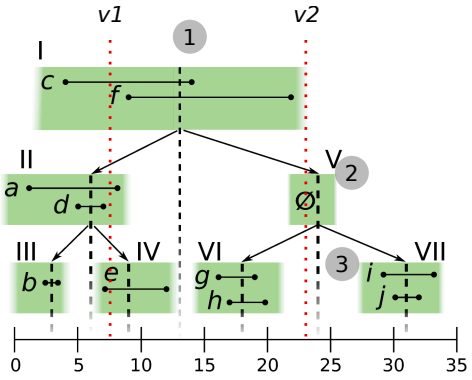
- Knoten können leer sein und keine Intervalle enthalten. (Abbildung 2.6 C Knoten V)
- Intervalle in Elternknoten können jene in Kinderknoten überspannen. (Vergleiche hierzu auch Abbildung 2.6 C. Der linke Endpunkt von Intervall a im Elternknoten II liegt noch vor dem von b des Tochterknotens III.)

Eine Kombination beider Bedingungen ist ebenso möglich, betrachtet man z.B. den rechten Endpunkt von Intervall f aus Knoten I (Abbildung 2.6 C) liegt dieser hinter jenen in Knoten VI. Um die nächsten Nachbarn einer solchen Suchanfrage s zu bestimmen, traversiert man erneut durch den Baum. Hier hilft wieder das Wissen, dass Intervalle von Töchterknoten nicht den Median des Elternknotens überschneiden können und s mit keinem Intervall des Baums überlappt. Daraus folgt, dass s zwischen den Intervallen eines Knotens und denen eines (Tochter-)Tochterknotens liegen muss. Man

initialisiert den besten linken Nachbarn (ln) bzw. rechten Nachbarn (rn) mit dem linkesten bzw. rechtesten Endpunkt aus den sortierten Listen des zuvor erwähnten Vorverarbeitungsschritts. Bei jedem besuchten Knoten aktualisiert man nun bis zum terminierenden Blatt die Nachbarn. Liegt die Suchanfrage s links von $n.median$, dann wird rn aktualisiert, wenn es ein Intervall gibt dessen linker Punkt vor rn liegt. Dementsprechend wird ln für den Fall, dass $s > n.median$ gilt, aktualisiert. Ein Beispiel für solch einen Fall wird in Abbildung 2.10 gezeigt.
Um alle Intervalle (Transkripte), die mit einem bestimmten Abfragepunkt/-intervall überlappen, zu bestimmen, benötigt es eine Laufzeit von $O(\log n + k)$ (Berg u. a., 2008)

2.3.3 Annotation von genomischen Varianten

Nachdem alle überlappenden Transkripte zu einer Variante bestimmt wurden, wird anhand einiger Regeln für jede identifizierte Position der mögliche Effekt auf das Transkript bestimmt (K. Wang, M. Li und Hakonarson, 2010). Der erste Schritt hierbei ist die Bestimmung der relativen Lage der Variante in der Exon/Intron-Struktur des Genes. Abhängig vom Typ (Substitution, Deletion, Insertion, MNV), den man aus den Informationen in der VCF-Datei ableiten kann (Danecek u. a., 2011), und der Lokalisation (codierende Sequenz, UTR, Intron) gibt es unterschiedliche Herangehensweisen, die genaue Kategorie und HGVS-Annotation zu bestimmen. Angenommen, man will die Auswirkung einer Deletion in der codierenden Region (CDS) bestimmen, so wird, ausgehend von der Distanz zum Translationsstart, die Position in der CDS bestimmt und welches Codon initial betroffen ist. Solange es keine in-frame Deletion ist, wird durch den Shift bei der Translation eine komplett veränderte und damit nicht mehr funktionale Proteinsequenz synthetisiert werden.
In Abbildung 2.11 ist der abstrahierte Pseudocode zur Annotation einer Deletion in *Jannovar* dargestellt. Dieser berücksichtigt noch nicht die Transkriptionsrichtung bzw. den Strang auf dem sich ein Gen befindet. Ein tiefe-



Schritt	Median	*ln*	*rn*
(1)	$23 > 13$	f.r $= 22$	$\emptyset$
(2)	$23 < 24$	f.r $= 22$	$\emptyset$
(3)	$23 > 18$	f.r $= 22$	i.l $= 26$

Abbildung 2.10: Beispiel für die Suche im Baum für zwei Positionen. Für die Abfrage eines überlappenden Punktes v1 kann der Standardalgorithmus für den Intervallbaum nach Berg u. a. (2008) verwendet werden. Beginnend in der Wurzel I kann direkt das erste überlappende Intervall c gefunden werden. Weiter durch den Baum traversierend, kann im Tochterknoten II das Intervall a und in dessen Tochter IV dann noch e identifiziert werden. Für die Abfrage der nächsten Nachbarn bei intergenischen Varianten ist eine Modifikation des Algorithmus notwendig. Als Beispiel soll v2 die Koordinate 23 haben. (1) Die Suche wird im Wurzelknoten begonnen. Da der Median (13) kleiner als 23 ist, wird der nächste linke Nachbar *ln* auf den am weitesten rechts liegenden Endpunkt (Intervall f mit f.r $= 22$) aus der Intervallliste von I gesetzt. (2) Nun wird im rechten Tochterknoten (V) geschaut ob sich *ln* oder *rn* aktualisieren lassen. Da V ein leerer Knoten ist und von einem kontinuierlichen, monotonen Koordinatensystem ausgegangen wird, wird im linken Tochterknoten (VI) geschaut, ob es ein weiteres Intervall mit der Endkoordinate 22 gibt. (3) Da der Median (24) von V größer als v2 ist, wird im rechten Tochterknoten nach dem am weitesten links gelegenen Endpunkt (Intervall i mit i.l $= 26$) geschaut und *rn* aktualisiert.

rer Einblick in die Implementation der Annotationen kann in der JavaDoc der *Jannovar*-Bibliothek erlangt werden.

Man sollte erwähnen, dass die HGVS-Annotation es nicht vorsieht, dass aus der chromosomalen Variante ein Rückschluß auf die Auswirkung in der Proteinsequenz abgeleitet wird. Dennoch ist es für die Interpretation und Bewertung/Beurteilung der Variante von großem Nutzen. Dies wird im All-

Pseudocode zur Annotation einer codierenden Deletion

```
ANNOTATE-DELETION(T, x, ref)
 1  cumlenintron = 0
 2  txstart = transcription start of transcript T
 3  cdsstart = coding sequence start of transcript T
 4  for i ∈ T.getExonCount()
 5        // Loop over all exons in transcript T
 6        if i > 0
 7              cumlenintron = T.getIntronLen(i)
 8        // no intron before first exon!
 9        if x ≥ T.getExonStart(i) and x ≤ T.getExonEnd(i)
10              rvarstart = x − txstart − cumlenintron + 1
11              break
12  mutpos = rvarstart − cdsstart + 1
13  return "c." + mutpos + "del" + ref
```

Abbildung 2.11: Der Pseudocode zeigt eine vereinfachte Idee zur Annotation einer Deletion von einer Base (ref) in der codierenden Region an der chromosomalen Position x. Um die kumulative Länge der Introns zu bestimmen, wird über alle Exons des betroffenen Transkripts $\mathcal{T}$ iteriert. Sobald das Exon erreicht ist, welches die Deletion enthält, kann von der chromosomalen Position x der Variante die Position des Transkriptionsstarts, sowie die kumulative Länge der Introns abgezogen werden, um die Position (rvarstart) der Variante im Transkript zu bestimmen. Um die Position im codierenden Bereich des Transkripts zu berechnen, wird von rvarstart noch der Translationsstart abgezogen. Die Funktion gibt die HGVS-Annotation für die cDNA des Transkripts zurück (z.B.: `c.204delT`). Eine dazugehörige Annotation für das korrespondierende Protein könnte z.B. `p.H86fs` sein.

gemeinen auch von allen Programmen zur Variantenannotation gemacht.

2.4 Stammbaumanalysen

Die moderaten Sequenzierkosten ermöglichen es mittlerweise, anstelle von nur dem betroffenen Individuum, ganze Familien zu sequenzieren. Gerade

PED-Format

Die ersten sechs Spalten einer PED-Datei:

```
#FamilienID  IndividumID  PaternalID  MaternalID  Geschlecht  Phänotyp
FAM01        MEM01        0           0           1           1
FAM01        MEM02        0           0           2           1
FAM01        MEM03        MEM01       MEM02       1           2
```

IDs sind alphanumerisch codiert und die Kombination aus FamilienID und IndividumID sollte eine Person eineindeutig definieren. Die IDs in den Spalten für die paternale und maternale ID beziehen sich auf die entsprechende IndividumID. Das Geschlecht ist numerisch und folgendermaßen definiert:

```
      0 unbekannt
      1 männlich
      2 weiblich
 andere unbekannt
```

Der Phänotyp wird ebenfalls numerisch codiert:

```
      0 unbekannt
      1 nicht betroffen
      2 betroffen
 andere unbekannt
```

Abbildung 2.12: Eine PED-Datei ist eine einfache Textdatei mit tabellarischen Aufbau, welche Spalten mittels Leerraum trennt (Leerzeichen oder Tabulator).

für die relativ trivialen Regeln der monogenen Erkrankungen nach mendelschen Gesetzen ist ein Filter nach bestimmten, für den Erbgang typischen Regeln einer linkage Analyse vorzuziehen.
Jannovar als Java-Bibliothek kann neben der Annotation von Varianten auch verwendet werden um solche Filter zu implementieren. Ein regelbasierter Stammbaum-Vererbungs-Filter ist in der Bibliothek implementiert. Mit diesem kann man – für ein klassisches Trio aus Kind, Vater und Mutter – Varianten auf die eingangs erwähnten mendelschen Erbgänge filtern und nur solche Varianten behalten, die für diesen Erbgang plausibel erscheinen. Das Programm verwendet als Eingabe eine multiVCF-Datei, mit den entsprechenden Varianten für mindestens das Trio (Mutter, Vater, betroffenes Kind), und eine PED-Datei (siehe Abbildung 2.12), welche das Geschlecht, das Eltern-Kind-Verhältnis und die Information, ob betroffen oder nicht, enthält. Die VCF-Datei wird importiert und es werden alle benötigten Informationen über die Abdeckung und den Genotyp zu einer Kandidatenvariante abgerufen. Im Code von *Jannovar* wird das folgendermaßen umgesetzt:

```
VCFReader vcfReader = new VCFReader(<VCF-File>);
ArrayList<Variant> variantList = vcfReader.getVariantList();
PedFileParser pedParser = new PedFileParser(<PED-File>);
Pedigree pedigree = pedParser.getPedigree(<Name>)
...
```

Im Folgenden finden sich die in *Jannovar* für die einzelnen Erbgänge implementierten Regeln wieder. Dabei werden Varianten in dem betroffenen Individuen auf die genannten Bedingungen überprüft:

Autosomal-dominant Für einen autosomal-dominanten Erbgang muss eine Variante von allen Betroffenen Individuen geteilt werden, wobei hier eine Heterozygotie ausreicht. Nicht betroffene Individuen dürfen diese Variante nicht tragen – weder homozygot noch heterozygot.

Autosomal-rezessiv Im autosomal-rezessiven Erbgang manifestiert sich der Phänotyp nur, wenn beide Allele die gleiche Variante tragen, die betroffene Person demnach homozygot für die Variante ist. Der Filter kontrolliert, dass alle betroffenen Individuen die Variante homozygot tragen und die Eltern zumindest heterozygot, da ansonsten die Allele nicht von beiden Eltern stammen können. Nicht betroffene Individuen dürfen die Variante nicht homozygot tragen.

X-Chromosomal-dominant Ein X-Chromosomal-dominanter Erbgang kann per Definition nur Varianten auf dem X-Chromosom betreffen. Alle betroffenen Individuen müssen die Variante auf mindestens einem Allel tragen, Männer hemizygot und in den PAR-Regionen[11] mindestens heterozygot und Frauen heterozygot. Nicht betroffene Individuen dürfen die Variante weder heterozygot noch homozygot aufweisen und mindestens eine betroffene Frau muss die Variante heterozygot tragen.

X-Chromosomal-rezessiv Der Filter für den X-Chromosomal-rezessiven Erbgang überprüft, dass eine Variante, im Fall eines betroffenen männlichen Kindes, nicht von einem nicht betroffenen Vater getragen wird und die Mutter die Variante heterozygot trägt. Bei einer betroffenen Tochter muss der Vater die Variante hemizygot tragen und betroffen sein, und die nicht betroffene Mutter die Variante heterozygot haben. Des Weiteren darf es im Stammbaum keine Person geben, welche die Variante auf allen Allelen trägt und nicht betroffen ist - homozygot für Frauen, hemizygot für Männer.

[11] PAR - pseudoautosomale Region – Region auf dem Y-Chromosom, welche eine homologe Entsprechung auf dem X-Chromosom hat.

2.5 Vergleich mit anderen Annotationsprogrammen

Die Laufzeit von *Jannovar* für die reine Annotation der Varianten wurde mit folgenden Programmen verglichen: ANNOVAR (K. Wang, M. Li und Hakonarson, 2010), VEP von Ensembl (Flicek u. a., 2013; McLaren u. a., 2010), AnnTools (Makarov u. a., 2012) und SnpEff (Cingolani u. a., 2012). Im UCSC Genome Browser wurde kürzlich das Programm *Variant Annotation Integrator* (VAI) zur Verfügung gestellt. Dieses annotiert Varianten ebenfalls funktionell und integriert diese mit den Features und genomischen Informationen aus dem UCSC Browser (Karolchik u. a., 2014). Leider ist dieses Programm nur als Onlineversion verfügbar, so dass es für den Vergleich nicht mit in Betracht gezogen werden konnte. Jedes der genannten Programme hat Eigenschaften, die es hervorhebt. Oft ist es der Annotationsumfang (Größe der Datenbank, Vereinbarungen mit nicht öffentlich zugänglichen Daten, ...) oder Zugänglichkeit (z.B. ein Webservice mit ansprechender Oberfläche). *Jannovar* sticht durch seine Flexibilität, seine Unterstützung für Stammbäume (siehe auch 2.4), die Parallelisierung der Variantenannotation und die Möglichkeit es als Softwarebibliothek zu verwenden, hervor.

Betrachtet man die mittlere Zeit, die benötigt wurde um VCF-Dateien mit 100 000 Varianten zu annotieren, so ist *Jannovar* das deutlich schnellste Programm (siehe Abbildung 2.13 (Jäger u. a., 2014)).

Im Januar 2015 wurde in einer kooperativen Arbeit von den Entwicklern von SnpEff, VEP und ANNOVAR ein Entwurf zur Variantenannotation im VCF-Format[12] veröffentlicht. Dieses Format wurde von allen Entwicklern, so auch *Jannovar*, als standardisiertes Ausgabeformat für ihre Annotationsprogramme umgesetzt.

[12] Variant annotations in VCF format:
http://snpeff.sourceforge.net/VCFannotationformat_v1.0.pdf

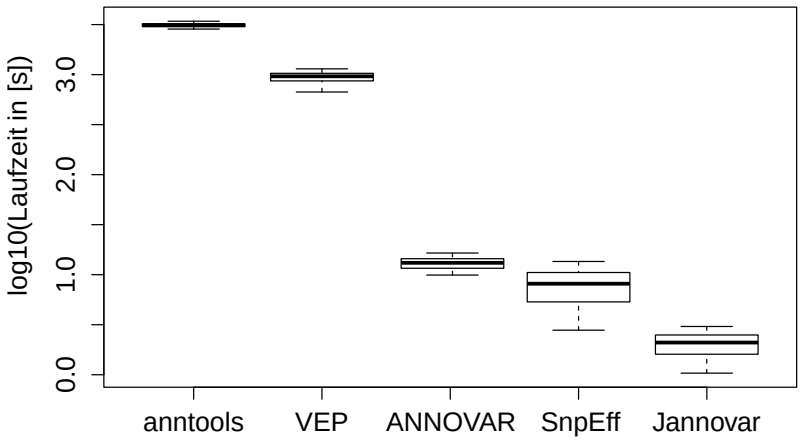


Abbildung 2.13: Diese Abbildung zeigt einen Vergleich der Laufzeiten zur Annotation von 100 VCF-Dateien mit jeweils 100 000 Varianten. Für alle Programme wurden die Features auf ein Minimum reduziert und falls vorhanden, die Annotation nach HGVS und auf SO-Terms aktiviert.

2.6 Ausblick und Anwendungsfälle

Jannovar scheint im ersten Augenblick nur ein weiteres Annotationstool zu sein, jedoch ist es im Hinblick auf Geschwindigkeit und Integrationsfähigkeit implementiert worden. Es ist das einzige Programm, das dahingehend entworfen wurde, als (Java-) Bibliothek in anderer Software Verwendung zu finden. Beispiele hierfür sind andere Exom- und Genomanalysetools wie Phenix (Zemojtel u. a., 2014), Exomiser (Smedley, Jacobsen u. a., 2015) und Genomiser (Smedley, Schubach u. a., 2016).

Jannovar kann auch als lokales Stand-Alone-Programm zur Annotation von Varianten im VCF-Format genutzt werden. Hierbei ist die Verwendung der Transkript-Datenbanken von Ensembl, UCSC und RefSeq schon vorimplementiert. Durch die laufzeitoptimierte Implementation der Suche mit Hilfe eines Intervallbaums, kann solch eine Annotation innerhalb von wenigen Sekunden auf einem einfachen „Desktop-Rechner“ ausgeführt werden (Ein

Exom z.B. innerhalb von ∼15 Sekunden.). Andere vergleichbare Programme benötigen dafür deutlich mehr Zeit (VEP 11.5min). Einige der kompetitiven Programme konzentrieren sich bei der Annotiation von Varianten auf das Exom und bieten zum Zeitpunkt der Veröffentlichung keine umfangreiche Annotation aller Varianten (z.B. UTR-Varianten, non-coding, intergenic) an. *Jannovar* hingegen liefert HGVS konforme Annotationen für alle Varianten.

Ein weiterer Vorteil ist, dass sich mit Hilfe von *Jannovar* sehr einfach auch Stammbaumanalysen und Filter umsetzen lassen. Eine Beispielimplementation ist in der Java-Bibliothek enthalten.

Durch die stetig sinkenden Kosten für NGS werden in absehbarer Zukunft immer häufiger Exome und auch Genome, insbesondere für seltene Erkrankungen und Krebs, sequenziert werden. Hierdurch wird der Bedarf nach Effizienz und Geschwindigkeit bei der Annotation immer mehr in den Vordergrund rücken.

Für die Abfrage der Transkripte wurde in *Jannovar* ein Intervallbaum gewählt. Dies ist eine der Datenstrukturen mit dem schnellstmöglichen Zugriff auf Intervalle, die mit einem bestimmten Punkt oder Intervall überlappen. Eine Alternative hierfür wäre ein Segmentbaum. Ein Segmentbaum ist jedoch auf die Abfrage von einzelnen Punkten optimiert und wäre nur optimal für den Fall, dass Variationen nur einzelne Basen betreffen, jedoch keine Intervalle, wie InDels oder MNVs.

Zum Aufbau eigener Annotationsprogramme oder Pipelines kann Jannovar von der GitHub-Seite `https://github.com/charite/jannovar` bezogen werden.

Kapitel 3

Alternative Locus Scaffolds – der Weg zum Graphengenom

In diesem Kapitel wird das Programm ASDPex beschrieben. ASDPex implementiert einen Algorithmus, der anhand des Musters der Varianten für die 178 Regionen mit alternativen Sequenzen im GRCh38 Genomassembly die wahrscheinlichste Kombination aus der primären Referenzsequenz und den verfügbaren alternativen Locus Scaffolds (*alt loci*) findet. Alternative Locus Scaffolds sind Sequenzen, welche eine alternative Repräsentation für einen bestimmten Bereich eines haploiden Genomassemblies darstellen[1].

3.1 Überblick

Das ursprüngliche humane Genomassembly hatte zum Ziel, ein haploides Konsensusgenom zu erschaffen – den sogenannten „Goldenen Pfad“ oder im englischen Original auch „golden path“ (Kent und Haussler, 2001; Lander u. a., 2001; Venter u. a., 2001). Diese haploide Repräsentation ermöglicht es leicht, Gene, Transkripte und andere genomische Merkmale, wie z.B. Varianten und deren eindeutige Annotationen, anhand der Koordinaten des „Goldenen Pfad“-Genoms, abzubilden. Basierend auf diesen eindeutigen Annotationen ist es möglich, Koordinaten zu vergleichen und Informationen

[1] https://www.ncbi.nlm.nih.gov/grc/help/definitions/

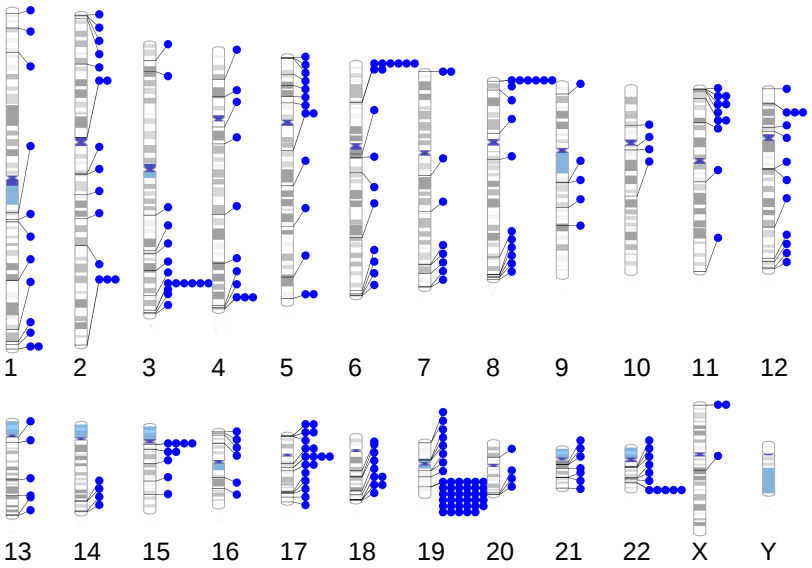


Abbildung 3.1: **Genomische Regionen mit *alt loci* Scaffolds** Das GRCh38 Referenzgenom enthält 178 Regionen mit einem oder mehreren *alt loci*. Die blauen Punkte geben die Position und jeweilige Anzahl der *alt loci* pro Region an. Dieses Ideogram wurde mit dem Programm PhenoGram (Wolfe u. a., 2013) erstellt.

über die genomischen Merkmale zu speichern und auszutauschen.

In den letzten Jahren zeigten sich durch den sprunghaften Anstieg der Anzahl sequenzierter humaner Genome und den Fortschritten in der Genomassemblierung die Nachteile dieser Repräsentation. Es stellte sich heraus, dass große strukturelle Variationen deutlich häufiger auftreten, als ursprünglich angenommen. Unter anderem gibt es populationsspezifische Regionen im Genom, die eine so hohe Allelvariabilität aufweisen, dass eine einzelne haploide Konsensussequenz als Referenz ungenügend ist (Church, Valerie A Schneider, Steinberg u. a., 2015).

Aus diesem Grunde fügte das Genome Reference Consortium (GRC) für ausgewählte Regionen, der ansonsten haploiden Genomreferenz, alternative Sequenzrepräsentationen hinzu. Dies geschah initial mit neun alterna-

tiven Sequenzen für drei Regionen in GRCh37. Mit der Veröffentlichung der großen Aktualisierung (GRCh38) für das humane Referenzgenom/Genomassembly im Dezember 2013 wurde dieser Aspekt noch stärker berücksichtigt. Im Folgenden findet sich eine Auflistung der wichtigsten Neuerungen der GRCh38 Veröffentlichung:

Centromere Die vorhergehenden Genomreferenzversionen repräsentierten die Centromere als mehrere Megabasen lange Lücken, die durch die ungefähre Anzahl von Basen durch *N*'s dargestellt wurden. Diese Lücken wurden in GRCh38 durch Sequenzen aus dem *HuRef genome*-Projekt (Levy u. a., 2007) ersetzt. Die Sequenzen stammen von einem einzelnen Individuum und wurden verwendet, um die Regionen um die Centromere zu modellieren (Miga u. a., 2014). Sie geben die ungefähre Anzahl von Sequenzwiederholungen (Repeats) und die Größenordung für jedes Centromer wieder und sind für die Abbildung von Reads auf eine Referenzsequenz (Read Alignment) wichtig.

Aktualisierung der Referenzsequenzen Studien, wie das 1000-Genome-Projekt, haben zahlreiche Basen und InDels in GRCh37 identifiziert, die in keinem Individuum der Studien gesehen wurden und höchstwahrscheinlich Assemblierungsfehler sind. Aus diesem Grunde wurden in GRCh38 mehrere tausend Basen angepasst, zahlreiche davon in codierenden Sequenzen. Zusätzlich dazu sind einige Regionen, die in GRCh37 schlicht falsch assembliert waren, korrigiert worden. Einige hochvariable Regionen wurden durch Sequenzen von Einzelindividuen ersetzt und über 100 Assemblierungslücken aktualisiert. Dies geschah entweder durch Auflösung oder eine deutliche Verkleinerung der Lücken (Valerie A. Schneider u. a., 2017).

Koordinaten Verbunden mit der Aktualisierung der Referenzsequenzen der einzelnen Chromosomen gab es eine Anpassung der Koordinaten. So hat sich die Gesamtlänge der Chromosomen und damit auch des humanen Genoms verändert (siehe auch Tabelle A2).

Variationen In GRCh38 wurden zahlreiche neue alternative Sequenzrepräsentationen für bestimmte Regionen hinzugefügt. Diese werden als alternative Loci Scaffolds (*alt loci*) bezeichnet. *Alt loci* sind selbstständige Sequenzen, für die die chromosomale Zuordnung in Form von einem Alignment zu den entsprechenden Referenzchromosomen bekannt ist. Alle *alt loci* besitzen sogenannte Ankersequenzen, die so identisch in den Referenzchromosomen wiedergefunden werden können, um die Qualität der Alignments sicherzustellen.

GRCh38 enthält 261 *alt loci* Scaffolds, die sich auf 178 Regionen verteilen. Von diesen waren 72 schon als NOVEL Patches in GRCh37 bekannt (siehe auch Abbildung 3.1 und Tabelle 3.1).

Bereich	Anzahl	Prozent [%]
< 100kb	27	15.2
$100 - 200$ kb	89	50.0
$200 - 500$ kb	43	24.1
500 kb – 2 Mb	13	7.3
> 2 Mb	6	3.4

Tabelle 3.1: **Größenverteilung der *alt loci*** für alle 178 Regionen des GRCh38 Referenzgenoms mit *alt loci*.

Mit der Aufnahme der alternativen Locus Scaffolds in GRCh38 führte das Genome Reference Consortium (GRC) eine graphenähnliche Repräsentation in Regionen mit bekannten komplexen Strukturvariationen ein. Auch wenn GRCh37 bereits drei Regionen mit insgesamt neun *alt loci* enthielt, eröffnet das neue Referenzgenom der Bioinformatik und allgemein der Genomforschung ganz neue Möglichkeiten, schafft aber auch neue Herausforderungen, welche mit den kommenden Versionen noch umfangreicher werden. Die Interpretation und die Analyse von Variationen kann und muss auf das neue komplexere Referenzgenom angepasst werden. Eine der wichtigsten technischen Herausforderungen ist die Adaption bestehender Formate und Programme an die Graphenrepräsentation. Die meisten wurden ursprünglich

mit Blick auf die haploide *golden path* Repräsentation des Referenzgenoms entwickelt und lassen sich nicht ohne weiteres adaptieren. Eine Struktur (Variante im Genom, NGS-Read, ...) läßt sich auf genau eine Position in der Referenz abbilden. Für das SAM-Format zur Abbildung von Sequenzalignments gab es erste Anpassungen. Es kann nun NGS-Reads repräsentieren, die sowohl auf das Primärassembly[2] (d.h. chr1 bis chr22, chrX, chrY, chrM), als auch auf eine zusätzliche Referenzsequenz (z.B. *alt locus*) abgebildet werden können. Für diese zusätzliche Abbildung ist ein zusätzlicher Eintrag in der SAM-Datei notwendig. Angepasste Alignmentprogramme, wie BWA-MEM (Li, Heng, 2013), markieren die Abbildung auf dem Primärassembly immer als die „repräsentative“ und die auf dem *alt locus* als die „zusätzliche“ Position oder auch (engl.) *supplementary* Alignment. Ein Beispiel für ein Read-Paar mit *supplementary* Alignment ist im Anhang in Abbildung A1 gezeigt. Eine entsprechende Erweiterung für VCF-Dateien, die es ermöglicht, Varianten von den gleichen NGS-Reads auf mehrere Position zu verweisen, fehlt bisher.
In der medizinischen Genomforschung geht es darum, die Varianten eines individuellen Genoms zu charakterisieren. Dies ist insbesondere bei einer diagnostischen Fragestellungen oder der Suche nach neuen krankheitsursächlichen Varianten in Genen von Bedeutung. In den hoch variablen Regionen, wie dem MHC Locus, die teilweise in mehreren tausend Positionen Varianten aufweisen, bietet sich nun die Möglichkeit mit den *alt loci*, die repräsentativste Sequenz zu bestimmen und damit die Anzahl der zu betrachtenden Varianten zu reduzieren und deren Plausibilität zu erhöhen.

3.1.1 Charakterisierung von *alt loci*

Die GRCh38 Referenz enthält 178 genomische Regionen mit einem oder mehreren *alt loci*. Insgesamt gibt es 261 *alt loci* Sequenzen. Deren Veror-

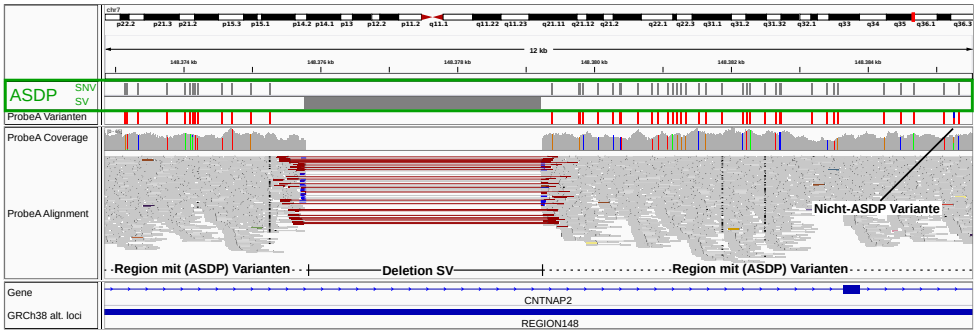
[2]Bei einer haploiden Genomreferenz entspricht das Primärassembly allen assemblierten Chromosomen, sowie den unlokalisierten und unplatzierten Sequenzen, welche zusammen ein nichtredundantes haploides Genom entsprechen. Dies schließt alle *alt loci* aus.

tung und Zuordnung zu den Regionen finden sich auf den entsprechenden Seiten zu den Genomreferenzen (bzw. deren Patches) auf dem FTP-Server des National Institute of Health (NIH)[3]. In den meisten Fällen (n=152) gibt es pro Region genau einen *alt locus*, jedoch auch fünf Regionen mit jeweils fünf oder mehr alternativen Sequenzen (CYP2D6 - 5; REGION151 / Mucin Region - 7; MHC - 8; LRC - 35). Die Größe der Regionen variiert zwischen 33 439 bis 5 081 216 nt, mit einem Mittelwert von 344 634 nt und einem Median von 169 569 nt. Die meisten Regionen besitzen also eine Länge zwischen 100 und 200 Kilobasen (Tabelle 3.1). Die kumulative Länge aller Regionen mit *alt loci* summiert sich auf 61 896 414 nt, was zwei Prozent der Länge des Primärassemblies (3 088 269 832nt) entspricht.

Die lokalen Sequenzsegmente der genomischen Regionen auf dem Primärassembly, die mit mindestens einem *alt locus* assoziiert sind, sollen der Einfachhalt halber als REF-HAP (Referenz-Haplotyp) und die entsprechenden Sequenzen der *alt loci* als ALT-HAP (Alternativ-Haplotyp) bezeichnet werden. Die Sequenzen der *alt loci* zeichnen sich dadurch aus, dass sie eine perfekte Übereinstimmung in den Ankersequenzen mit dem Primärassembly haben, jedoch mindestens eine variable Region aufweisen (siehe auch Abbildung A7). Angenommen in einem NGS sequenzierten Individuum liegt in einer der variablen Regionen heterozygot ein *alt locus* (ALT-HAP) vor und die Reads werden gegen das Primärassembly (REF-HAP) abgebildet. Dann erwartet man, dass sich heterozygote Varianten in genau denjenigen Positionen, in denen sich REF-HAP und ALT-HAP unterscheiden, zeigen. Dies können Unterschiede in einzelnen Basen oder auch strukturelle Variationen sein, die zu einem typischen Muster im Vergleich zwischen REF-HAP und ALT-HAP führen – einer Art *Fingerabdruck* für den *alt locus*. Ein Beispiel hierfür ist in Abbildung 3.2 gezeigt. Für Probe A liegt der ALT-HAP homozygot vor, wodurch es, nach dem Alignment und Calling der Variationen, zu einem typischen Variantenmuster für den REF-HAP kommt. Mit diesem Fingerabruck ist eine Bestimmung des wahrscheinlichsten *alt locus* mithilfe

[3] ftp://ftp.ncbi.nlm.nih.gov/genomes/H_sapiens/

A)



B)

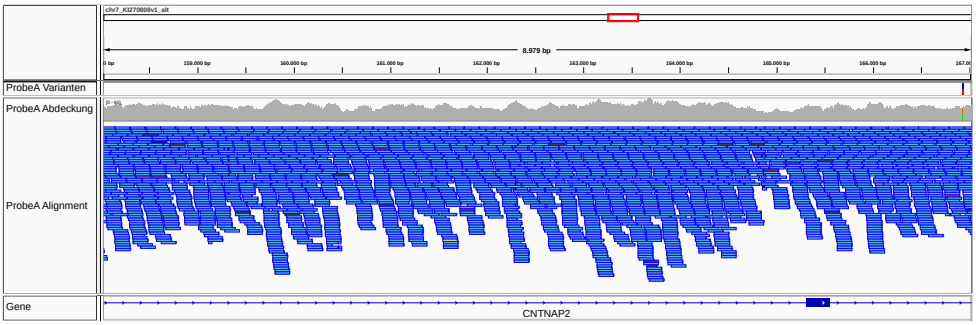


Abbildung 3.2: **Region 148**. Die beiden Abbildungen aus IGV (J. T. Robinson u. a., 2011) zeigen Alignments und Varianten für die *in-house* Probe A. A) Zahlreiche homozygote ASDP*-assoziierte Varianten, sowie eine mit dem *alt locus* `KI270808.1` assoziierte strukturelle Variante suggerieren, dass Probe A wahrscheinlich homozygot für `KI270808.1` statt für die REF-HAP Sequenz von `REGION148` ist. Zusätzlich zu den ASDP-assoziierten Varianten existiert eine heterozygote Variante (rechts). Von den 52 Varianten, welche mit ASDPs korrespondieren, sind 50 in dbSNP enthalten. B) Die korrespondierende Region auf `KI270808.1` wurde fehlerfrei aligniert. Einzig eine nicht-ASDP-assoziierte Variante wurde bestimmt. IGV markiert *supplementary* Reads in Blau (d.h. Reads, die auf das Primärassembly ebenso gut, wie auf einen *alt locus* mappen).
*ASDP - Alignable Scaffold-Discrepant Position. Siehe auch Definition auf Seite 91.

eines heuristischen Ansatzes möglich.

Definition alternativer Locus Scaffold (*alt locus*)

Ein alternativer Locus Scaffold (*alt locus*) ist eine Sequenz, die eine alternative Repräsentation einer genomischen Region darstellt. *Alt loci* werden für Regionen zur Verfügung gestellt, für die es eine hohe Variabilität zwischen den Populationen gibt und sind in die ansonsten haploide Repräsentation des Genoms eingebettet.

3.2 Alignments

Um den oben erwähnten Fingerabdruck für jeden *alt locus* zu der korrespondieren Region auf dem Primärassembly zu erhalten, benötigt man Alignments zwischen diesen. Das GRC hat mit dem GRCh38 Release Alignments für alle Regionen und alternativen Haplotypen für alle *alt loci* im GFF3-Format veröffentlich. Die Definition (Größe und Ausdehnung) der Regionen und die Bereiche der Alignments der *alt loci* zu den Regionen ist von der Anzahl und Positionierung der alternativen Haplotypen in der Region abhängig. Man kann die Regionen grob in fünf Gruppen untergliedern. Die erste Gruppe enthält Regionen mit nur einem *alt locus*, der sich über die gesamte Region ausdehnt (z.B. Region ADAM5 – Abbildung A2). In der zweiten Gruppe erstreckt sich der *alt locus* über die komplette Region, besitzt jedoch eine Insertion am Anfang des Alignments (z.B. REGION14 – Abbildung A3). Entsprechend gibt es eine dritte Gruppe von Regionen mit einer Insertion am Ende des Alignments (z.B. REGION142 – Abbildung A4). Die vierte Gruppe von Regionen beinhaltet mehrere *alt loci*, welche sich nahezu über die ganze Region ausstrecken (z.B. Region MHC – Abbildung A5). Die fünfte Gruppe besitzt auch mehrere *alt loci*, welche sich jedoch über die Region verteilen (z.B. REGION151 – Abbildung A6).

3.2.1 Das GFF3-Format

Das General Feature Format Version 3 (GFF3) ist ein tabellarisches Format zur Repräsentation von biologischen Features (genomische Einheiten, wie Exons etc.). Pro Zeile wird ein Feature mit Hilfe von folgenden neun Spalten beschrieben:

Spalte 1 - SequenzID Die einmalige ID der Referenz, an der das Koordinatensystem festgemacht wird. Sie enthält einfache Charakterzeichen (a-zA-Z0-9.:^*$!+_?-|) und darf keine Leerzeichen enthalten oder mit '>' beginnen.

Spalte 2 - Quelle Beschreibender Name für den Ursprung dieses Features. Typischerweise enthält es den Namen des Programms oder der Datenbank aus denen die Datei erzeugt wurde.

Spalte 3 - Typ Der Typ des Features. Dies muss ein Begriff aus der Sequence Ontology-Datenbank (Eilbeck und Lewis, 2004) oder eine SO-Accessionnumber sein.

Spalte 4 & 5 - Start & Ende Start- und Endposition auf der Referenzsequenz (SequenzID). Das Koordinatensystem ist 1-basiert und die Startkoordinate ist immer kleiner oder gleich der Endkoordinate.

Spalte 6 - Wert Der Wert des Features als Fließkommazahl.

Spalte 7 - Strang Der Strang des Features im Bezug auf die Referenz (SequenzID). '+' für den positiven Strang, '-' für den negativen Strang und '.' für stranglos. '?' kann für einen unbekannten Strang verwendet werden.

Spalte 8 - Phase Vom Typ Integer und wichtig für Features vom Typ „CDS„. Hier gibt die Phase an, wie weit das Feature zur Referenz im Codon im Frame versetzt ist. Der Wert (0, 1, 2) gibt an, wieviele

Basen 5' vom Feature entfernt werden müssen, um die Startposition des ersten vollständigen Codons zu erreichen.

Spalte 9 - Attribute Eine Liste von Eigenschaften des Features im Format 'Attribut=Wert'. Mehrere Attribute werden durch ein ';' voneinander getrennt.

Für die Liste der Attribute in Spalte 9 gibt es eine Reihe im GFF3-Format vordefinierter Tags[4]. Im Folgenden sind die zur Beschreibung eines Alignments notwendigen Attribute aufgelistet:

ID In der GFF3-Datei einmalige ID für das Feature.

Target Name beziehungsweise ID der Zielsequenz eines Alignments. Das Format ist '`ZielID Start Ende [Strang]`', wobei der Strang optional ist und '`+`' oder '`-`' sein kann.

Gap Das Alignment zwischen Feature und Zielsequenz. Das Alignment Format entspricht dem CIGAR-Format, das auch im SAM-Format (H. Li, Handsaker u. a., 2009) verwendet wird.

Im Folgenden sieht man einen Ausschnitt aus der vom GRC zur Verfügung gestellten Alignmentrepräsentation zwischen einer Region auf Chromosom 6 (`NC_000006.12`) und dem *alt locus* `HSCHR6_MHC_APD_CTG1` (`NT_167244.2`). Neben den oben genannten Attributen sind hier noch einige zusätzliche Tags mit Informationen und Werten zum Alignment enthalten.

GFF-Format

```
##gff-version 3
#!gff-spec-version 1.20
#!processor NCBI annotwriter
NC_000006.12   RefSeq   match   28734408   33367716   0   +   .   ID=aln0;Target=NT_167244.2 1 4672374 +;    \
align_id=1411;batch_id=13041;bit_score=1.12029;common_component=0;e_value=4.83307+11;expansion=1.96768;      \
filter_score=7;merge_aligner=1;merge_options=58;num_ident=2348261;num_mismatch=6441;pct_coverage=50.3963;    \
pct_identity_gap=33.7832;pct_identity_gapopen_only=99.688;pct_identity_ungap=99.7265;reciprocity=3;score=0;  \
Gap=M44558 D8 M13552 D2 M3960 D4 M2109 D2 M33 I2 M25649 I6 M18550 D1 M68504 I1 M25289 I44253 D44253 M3450    \
D1 M345 I1 M3095 I39 M16670 I1 M48609 D2 M2804 I29384 D29384 M14754 I7 M3825 D2 M7033 D8 M3588 D1 M19578 ...
```

[4]Eine Übersicht der Spalten und vordefinierter Attribute findet sich unter: `http://gmod.org/wiki/GFF3`

Der das Alignment beschreibende Abschnitt wird durch ein `Gap=` eingeleitet, worauf mehrere Blöcke aus `[M,I,D]` und einer Zahl (entsprechend der Länge des Blocks) folgen.

`M` steht für ein 'Match', also eine Region übereinstimmender Sequenz zwischen Referenz und *alt locus*. Diese Region kann Mismatches, jedoch keine Gaps aufweisen.

`I` steht für eine Insertion von zusätzlicher Sequenzinformation in den *alt locus*.

`D` steht für eine Deletion von Sequenz aus dem *alt locus*.

3.2.2 Suboptimale GRC-Alignments

Die vom GRC bereitgestellten Alignments beginnen und enden mit identischen Ankerregionen (siehe Abbildung A7). Eine manuelle Inspektion der Alignments mit besonderem Augenmerk auf die Blockenden zeigte, dass einige Bereiche, insbesondere am Übergang zu Insertionen und Deletionen, jedoch suboptimal sind und oft stark fragmentiert erscheinen. Sie sind häufig durch eine partielle Sequenzähnlichkeit in mehrere kleine (suboptimale) Alignmentblöcke aufgeteilt (siehe Abbildung A8). In Abbildung 3.3 kann man sehr deutlich sehen, dass sich die Qualität nach einem erneuten Alignment zwischen *alt locus* und korrespondierender Region auf dem Referenzchromosom deutlich verbessern läßt. Dieser Schritt ist von Bedeutung, um einen möglichst genauen und repräsentativen Fingerabdruck für jeden *alt locus* zu erhalten. Aus diesem Grund muss für alle vorhandenen Alignments zwischen REF-HAP und korrespondierenden ALT-HAPs ein erneutes paarweises Alignment durchgeführt werden, um einen optimalen Fingerabruck für die *alt loci* zu erhalten. Hierzu sollen die Informationen aus den GRC-Alignments im GFF3-Format genutzt werden.

A)

```
12650    .    :    .    :    .    :    .    :    .    :
     TTGTTGGGGTTGGGGTTGGTGGTGGTGGTGGTGGTGTGGTGTTGTTGGTG
     ||||||||||||||||||||||||||||||||||||||||||||||||||
     TTGTTGGGGTTGGGGTTGGTGGTGGTGGTGGTGGTGTGGTGTTGTTGGTG

12700    .    :    .    :    .    :    .    :    .    :
     TTAGTGTTGGTGTTCTGATGTGGTGTTGGTGGTGTTGGTGTTGGTGGTGT
     ||||||||||||||||||||||||||||||||||||||||||||||||||
     TTAGTGTTGGTGTTCTGATGTGGTGTTGGTGGTGTTGGTGTTGGTGGTGT

12750    .    :    .    :    .    :    .    :    .    :
     TGTTGGTGTGGTGGTGTTGGTGTGGTGTTGTTGGTGTTGGTGTTGGTGGT
     ||||||||||||||||||||||||||||||||||||||||||||||||||
     TGTTGGTGTGGTGGTGTTGGTGTGGTGTTGTTGGTGTTGGTGTTGGTGGT

12800    .    :    .    :    .    :    .    :    .    :
     GTTGGTGTTGGTGTGGTGTTGTTGGTGTTGGTGTGGTGTGATGATGGTGG
     ||||||||||||||||||||||||||||||||||||||||   | || |||
     GTTGGTGTTGGTGTGGTGTTGTTGGTGTTGGTGTGGTGTT--GGTGTTGG

12850    .    :    .    :    .    :    .    :    .    :
     TGGTATTGTGGTAATAGGATGGTGGTGGTAGTATGGTGGTGGT-------
     |||| |||  ||  | || || ||||| |  |  |||| ||||
     TGGTGTTGGTGTTGTTGG-TGCTGGTGTTGTT--GGTGTTGGTGTGGTGT




     ...




16100    .    :    .    :    .    :    .    :    .    :
     ----GGTGGTAACAGTAGGCATGGTGGTGATTACTATATAACGATGATAA
         ||||||||||||||||||||||||||||||||||||||||||||||||||
     TGGTGGTGGTAACAGTAGGCATGGTGGTGATTACTATATAACGATGATAA

16150    .    :    .    :    .    :    .    :    .    :
     CGTTCTTGTGGAGGTGACTGTGTCGATGACACACTTGTGTGTCTGATGCC
     ||||||||||||||||||||||||||||||||||||||||||||||||||
     CGTTCTTGTGGAGGTGACTGTGTCGATGACACACTTGTGTGTCTGATGCC
```

B)

```
12650    .    :    .    :    .    :    .    :    .    :
     TTGTTGGGGTTGGGGTTGGTGGTGGTGGTGGTGGTGTGGTGTTGTTGGTG
     ||||||||||||||||||||||||||||||||||||||||||||||||||
     TTGTTGGGGTTGGGGTTGGTGGTGGTGGTGGTGGTGTGGTGTTGTTGGTG

12700    .    :    .    :    .    :    .    :    .    :
     TTAGTGTTGGTGTTCTGATGTGGTGTTGGTGGTGTTGGTGTTGGTGGTGT
     ||||||||||||||||||||||||||||||||||||||||||||||||||
     TTAGTGTTGGTGTTCTGATGTGGTGTTGGTGGTGTTGGTGTTGGTGGTGT

12750    .    :    .    :    .    :    .    :    .    :
     TGTTGGTGTGGTGGTGTTGGTGTGGTGTTGTTGGTGTTGGTGTTGGTGGT
     ||||||||||||||||||||||||||||||||||||||||||||||||||
     TGTTGGTGTGGTGGTGTTGGTGTGGTGTTGTTGGTGTTGGTGTTGGTGGT

12800    .    :    .    :    .    :    .    :    .    :
     GTTGGTGTTGGTGTGGTGTTGTTGGTGTTGGTGTGGTGT----------
     ||||||||||||||||||||||||||||||||||||||
     GTTGGTGTTGGTGTGGTGTTGTTGGTGTTGGTGTGGTGTTGGTGTTGGTG



     ...



16000    .    :    .    :    .    :    .    :    .    :
     ------------------------------------------GTGATGA
                                                |||||||
     TGATGGTAGGACAGTGGTGGTGGTGATGATGGTGGTGGTGAGGGTGATGA


16050    .    :    .    :    .    :    .    :    .    :
     TGGTGGTGGTATTGTGGTAATAGGATGGTGGTGGTAGTATGGTGGTGGTG
     ||||||||||||||||||||||||||||||||||||||||||||||||||
     TGGTGGTGGTATTGTGGTAATAGGATGGTGGTGGTAGTATGGTGGTGGTG

16100    .    :    .    :    .    :    .    :    .    :
     GTGGTAACAGTAGGCATGGTGGTGATTACTATATAACGATGATAACGTTC
     ||||||||||||||||||||||||||||||||||||||||||||||||||
     GTGGTAACAGTAGGCATGGTGGTGATTACTATATAACGATGATAACGTTC

16150    .    :    .    :    .    :    .    :    .    :
     TTGTGGAGGTGACTGTGTCGATGACACACTTGTGTGTCTGATGCCAAGTG
     ||||||||||||||||||||||||||||||||||||||||||||||||||
     TTGTGGAGGTGACTGTGTCGATGACACACTTGTGTGTCTGATGCCAAGTG
```

Abbildung 3.3: **Alignment zwischen** `REGION155` (oben) **und** `chr8_KI270820v1_alt` (unten). A) Gezeigt ist der Ausschnitt des Alignments im Bereich einer größeren Insertion im *alt locus*, so wie es in der NCBI GFF3-Datei definiert ist. Deutlich zu erkennen ist das suboptimale Alignment ab Position 12 839. B) Der gleiche Bereich nach einem erneuten globalen Alignment (Needleman und Wunsch, 1970). Der lückenhafte Bereich ab Position 12 839 ist verschwunden und wurde durch einen zusammenhängenden passenden Block am 3'-Ende der Insertion ersetzt.

3.2.3 Banded-Chain-Alignment

Die Berechnung eines optimalen paarweisen globalen Alignments ist kostenintensiv in Bezug auf Speicher und Laufzeit. Im schlechtesten Fall steigen

die Kosten $O(n^2)$ quadratisch zur Sequenzlänge n. Ein typisches Beispiel für dynamische Programmierung aus der Bioinformatik ist der Needleman-Wunsch-Algorithmus (Needleman und Wunsch, 1970). Er wird häufig für ein optimales Alignment zwischen zwei Nukleotid- oder Aminosäuresequenzen eingesetzt. Dieser ist durch den quadratischen Speicherbedarf jedoch nicht für solch ein großes genomisches Sequenzalignment geeignet. Auch wenn es alternative Algorithmen, wie den Hirschberg-Algorithmus (Hirschberg, 1975), mit einem linearen Speicherplatzbedarf gibt, soll das Problem hier im quadratischen Fall dargestellt werden. Bei diesem Beispiel wird von einer equivalent langen Sequenz für `REGION25` und den *alt locus* ausgegangen. Eine Speicherabschätzung mit 32 bit Speicherplatzbedarf pro Integer ergibt für `REGION25` mit einer Länge von 5 161 414 nt dann:

$$\text{Matrix:} \quad 5\,161\,414 \times 5\,161\,414 \times 4\text{Byte} \approx \underline{106,6 \text{ Terabyte}}$$

Der Bedarf kann in naher Zukunft nur sehr unwahrscheinlich durch flüchtigen RAM-Speicher realisiert werden. Die Lösung für das Speicher- und Laufzeitproblem ist der „teile und herrsche“-Ansatz (im Englischen „divide and conquer“ genannt), ein in der Informatik gängiges Paradigma für den Entwurf von effizienten Algorithmen. Der Ansatz basiert auf dem Prinzip, ein unlösbares Problem solange rekursiv aufzuteilen, bis man mehrere einfachere und damit lösbare Teilprobleme erhält.
Eine effektive Umsetzung ist die „seed-and-extend“-Methode, welche wir hier Banded-Chain-Alignment (Brudno u. a., 2003) nennen wollen. Hierbei werden Seeds, also Regionen einer minimalen Größe mit einer nahezu perfekten Übereinstimmung zwischen zwei Sequenzen, bestimmt. Das Alignment beschränkt sich dann nur auf ein relativ schmales Band (engl. *band*) entlang der verketteten (Kette - engl. chain) Seeds und die Regionen zwischen den Enden der Seeds (siehe Abbildung 3.4 B). Dadurch läßt sich der Suchraum deutlich einschränken und aus einem quadratischen Problem läßt sich ein Problem viel geringerer Größe machen.
Im gegebenen Anwendungsfall stecken diese Seeds schon in den GFF3-

A) B)

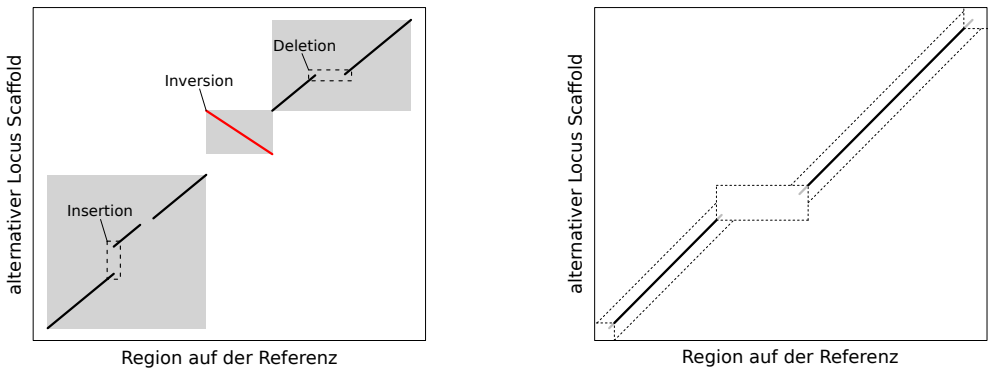


Abbildung 3.4: **Banded-Chain-Alignment**. A) Teile und herrsche. Die übereinstimmenden `M` Blöcke aus den GFF3-Alignments sind als schwarze und rote Linien dargestellt. Mittig gibt es eine große Inversion (rot), weshalb das ganze Alignment in drei größe Blöcke (grau hinterlegt) aufgeteilt wird. Für diese wird einzeln ein Banded-chain-Alignment durchgeführt. In der GFF3-Datei ist die Inversion als große Deletion gefolgt von einer vergleichbar großen Insertion der invertierten Sequenz dargestellt. B) Bestimmung der Seeds für das Banded-Chain-Alignment. Die Abbildung zeigt den dritten Alignmentblock aus Abbildung A). Die übereinstimmenden `M` Blöcke sind als dicke schwarze Linien mit grauen Enden dargestellt. Die grauen Enden sind entsprechend dem Algorithmus 3 abgeschnitten und der übrige schwarze Rest stellt die Seed für das Banded-Chain-Alignment (Brudno u. a., 2003) dar. Das finale Alignment ist auf den gestrichelten Bereich der Kästen und entlang der `M` Blöcke beschränkt.

Alignments. Da sich die *alt loci* auch durch Strukturvarianten in Form von Insertionen, Deletionen und Inversionen vom Primärassembly unterscheiden (siehe Abbildung 3.4 A), ist ein komplettes qualitatives Alignment häufig nicht für die kompletten Sequenzen möglich.

Um die Qualität des Alignments zu verbessern, wurde ein weiterer „divide and conquer“-Ansatz erdacht und implementiert. Die angewendete Strategie definiert das Gesamtalignment neu, indem es auf mehrere Teilprobleme oder

in diesem Fall Teilalignments runtergebrochen wird, welche auch Inversionen behandeln kann: I) Aufspalten der Alignments in Blöcke; II) Bestimmung der Seeds innerhalb der Blöcke; III) Behandlung von `'N'`-Blöcken.

I) Einige der GFF3-Dateien enthalten einen zweiten Alignmenteintrag. In allen diesen Fällen wird durch diesen zusätzlichen Eintrag eine Inversion definiert und es läßt sich im ersten Eintrag eine große Insertion (`I`), gefolgt von einer Deletion (`D`) finden. Da sich nur mit Matches, Insertionen und Deletionen keine Inversionen beschreiben lassen, wurde für Inversionen diese Abhilfe verwendet. In diesen Fällen wird das Alignment, wie in Abbildung 3.4 `A`) zu sehen, anhand der großen Insertions-/Deletionsblöcke in Teilprobleme aufgespalten und die Alignments mit der inversen Sequenz des *alt loci* Bereichs, separat durchgeführt.

II) Seeds sind durch lange `M` Blöcke definiert, welche eine große Sequenzähnlichkeit zwischen Referenz und *alt loci* aufweisen - es gibt nur übereinstimmende Basen und einige wenige Mismatches, jedoch keine InDels. Mismatches treten gehäuft an den Enden auf, aus diesem Grund wurden fünf Prozent, jedoch nicht mehr als 50 Nukleotide, von beiden Enden entfernt. Dies ermöglicht es dem Aligner diese gegebenenfalls neu zuzuordnen und wie in Abbildung 3.3 saubere Enden zu schaffen. Kandidaten `M` Blöcke, die nach dem Trimming der Enden kürzer als 50 nt waren, wurden verworfen.

III) Einige *alt loci* besitzen lange Bereiche von fortlaufenden `'N'`s (z.B. `KI270905.1`, `GL000258.2`, `GL383571.1`, ...), die in den GRC-Alignments als Matches zwischen REF-HAP und ALT-HAP gelten und damit in den `M` Blöcken auftreten. Für Folgen von mehr als zehn `'N'`s wurde ein Kandidaten-Seed an den Enden der `N`s in zwei neue Kandidaten-Seeds aufgeteilt, welche dann wiederum den oben genannten Kriterien entsprechen müssen.

Algorithmus 3 Der Algorithmus validiert Kandidatenseeds für die Verwendung im Banded-Chain-Alignment. Von jedem Ende des Eingabeseeds $\mathcal{M}$ werden 5% der Gesamtlänge, jedoch nicht mehr als 50 nt abgeschnitten. Anschließend wird auf die geforderte Minimallänge geprüft und Seeds < 50 nt werden verworfen. Das Ergebnis sind, in den chromosomalen Koordinaten (`ref_start`, `alt_start`), aktualisierte Seeds mit einer Länge von `seed_len`.

KandidatenSeedValidierung($\mathcal{M}$)

```
SVMIN ⇐ 50 // Minimallänge für Alignmentseed
TRIMFRAC ⇐ 0.05 // A
seed_len ⇐ length(M)
ref_start ⇐ start_position_in_ref_hap(M)
alt_start ⇐ start_position_in_alt_hap(M)
if seed_len * (1 − 2 * TRIMFRAC) < SVMIN
    return: null
CHOP ⇐ round(seed_len * TRIMFRAC) // zu entfernende Enden
if CHOP > SVMIN
    CHOP ⇐ SVMIN
ref_start ⇐ ref_start + CHOP
alt_start ⇐ alt_start + CHOP
seed_len ⇐ seed_len − (2 * CHOP)
return: endbereinigter Alignmentseed
```

Die hier genannten Vorverarbeitungs- und Filterschritte für die Kandidaten-Seeds sind in Algorithmus 3 zusammengefaßt. Das Ergebnis ist eine Liste von Seeds der Länge 50 nt oder größer, die als Eingabe für das Banded-Chain-Alignment Verwendung finden. In dieser Arbeit wurden 402 Alignmentblöcke identifiziert. Die mittlere Länge der Alignmentblöcke entspricht 248 928 nt in Bezug auf die REF-HAP Sequenzen.

Die C++ Bibliothek SeqAn (Döring u. a., 2008) in Version 2.0.1 wurde für die Implementierung des Filterschritts der Seeds sowie für die Anwendung des Banded-Chain-Alignment Algorithmus verwendet. Beide Schritte wurden in einem kleinen Programm (`regionalign2vcf`[5]) zusammengefaßt.

Die hier empfohlenen und für die folgenden Ergebnisse verwendeten Para-

[5] https://github.com/martenj/hg38altLociSelector

meterwerte für das Banded-Chain-Alignment sind:

```
match:          5
mismatch:      -2
gapextend:      0
gapopen:      -20
anchor bands: 10
```

3.3 Identifizierung von ASDPs

Die Alignments zwischen den Primärassemblies und den *alt loci* zeigen, dass sie große Bereiche enthalten, die sich stark ähneln, jedoch in einigen Positionen individuell unterscheiden. Diese wenigen Unterschiede in den Sequenzen, welche in ansonsten sequenzidentischen Regionen liegen, sollen von nun an als Alignable Scaffold-Discrepant Positions (ASDPs) referenziert werden. Zur Identifizierung dieser benötigt man qualitativ hochwertig Alignments zwischen jedem REF-HAP und allen dazugehörigen ALT-HAPs. Dies läßt sich mit dem in dieser Arbeit entwickelten Programm `regionalign2vcf` umsetzen (siehe Kapitel 3.2.3).

Parameterübersicht für `regionalign2vcf`

```
SYNOPSIS
DESCRIPTION
    -h, --help
          Displays this help message.
    -R, --region TEXT
          path to the region fastA file.
    -A, --altloci TEXT
          path to the alt loci fastA file.
    -V, --vcf TEXT
          path to the vcf file with the found differences.
    -S, --seed TEXT
          path to the file with the seed informations.
    -N, --aln TEXT
          path to the file with the final alignment.
    -o, --offset INT
          start of the region in the reference
    -a, --append
          append results to existing file
```

Als Eingabe benötigt es die Sequenzen für REF-HAP (`-R`) und ALT-HAP (`-A`) im FastA-Format sowie eine Liste von Seeds aus der GFF3-Datei (`-S`).

Die Ausgabe (`-V`) der Unterschiede erfolgt im VCF-Format (Abbildung 3.5 B). Neben den Abweichungen lassen sich die Alignments (Abbildung 3.5 A) im TXT-Format (`-N`) ausgeben. Mit dem Parameter (`-o`) läßt sich ein Offset für die Ausgabe im VCF-Format definieren. Dieser sollte idealerweise der Startposition des REF-HAPs auf dem Primärassemblies entsprechen, wodurch sich die Variante in Relation zur chromosomalen Position und nicht nur zu der Region angeben läßt. Standardmäßig gibt das Program eine Datei mit den Unterschieden für das Alignment eines REF-HAPs mit einem ALT-HAP aus. Die Ausgaben von mehreren Alignments lassen sich akkumulieren, indem man mit `-V` auf dieselbe Datei verweist. Für diesen Fall sollte man den `-o` Paramenter verwenden und die Startposition des REF-HAPs auf dem Primärassemblies angeben, um eine chromosomale Positionierung in der Ausgabe zu erhalten. Die Spalten der Ausgabedatei beinhalten, dem VCF-Format (`CHROM, POS, ID, REF, ALT, QUAL, FILTER, INFO, ...`) entsprechend, die folgenden Informationen:

- Das Chromosom des REF-HAP.
- Die Position auf dem Chromosom, falls mit dem Offset angegeben.
- Platzhalter für die ID – '.'.
- Die Referenznukleotide des REF-HAP.
- Die Veränderung im ALT-HAP im Vergleich zum REF-HAP.
- Platzhalter für die Qualität – '40'.
- Platzhalter für den Filter – `PASS`.
- Zusätzliche beschreibende Informationen. Schlüssel-Wert-Paare für die Attribute ALT-HAP (z.B. `AL=chr8_KI270822v1_alt`), REF-HAP (z.B. `RE=ADAM5`) und die Position im REF-HAP (z.B. `RP=69198`)

In Abbildung 3.5 B findet sich ein Beispiel für drei Alignmentunterschiede zwischen der Region `RE=ADAM5` und dessen *alt locus* `AL=chr8_KI270822v1_alt`.

A)

```
69100     .    :    .    :    .    :    .    :    .    :
      TTTTGGTAATAGTGTAGGGACCAGATTGCTGGTGGGAAAATTGGGGAAGG
      |||||||||||||||||||| ||||||||||||||||||||||||||||
      TTTTGGTAATAGTGTAGGGACGAGATTGCTGGTGGGAAAATTGGGGAAGG

69150     .    :    .    :    .    :    .    :    .    :
      AGGAATCAAATTTTAAGAGACTGTTCTAGTAATCAGGGTGAAAACTTAGA
      ||||||||   |||||||||||||||||||||||||||||||||||| |
      AGGAATCA---TTTAAGAGACTGTTCTAGTAATCAGGGTGAAAACTTATA
```

B)

#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO	...
chr8	39163439	.	C	G	40	PASS	AL=chr8_KI270822v1_alt;RE=ADAM5;RP=69121	...
chr8	39163475	.	AAAT	A	40	PASS	AL=chr8_KI270822v1_alt;RE=ADAM5;RP=69158	...
chr8	39163516	.	G	T	40	PASS	AL=chr8_KI270822v1_alt;RE=ADAM5;RP=69198	...

Abbildung 3.5: **VCF-Repräsentation ASDP-assoziierter Varianten**. Repräsentation von ASDP-assoziierten Varianten in einer VCF-Datei, wie von ASDPex zur Speicherung von ASDPs verwendet. A) Das Alignment zwischen primär (oben) und *alt locus* (unten) Scaffold zeigt Single-Nukleotid Mismatches an Position `69 122` und `69 199` und eine Deletion von drei Nukleotiden in `69 159-69 161`. B) Der Ausschnitt aus der VCF-Datei entspricht den Varianten im Panel A. ASDPex speichert die Felder `AL` (alternate locus), hier `chr8_KI270822v1_alt`, `RE` (Region), hier `ADAM5` und `RP` (Region Position) in der `INFO` Spalte (siehe auch Abschnitt *VCF-Format*).

Jede Position der so erzeugten Alignments wird auf Mismatches oder Gaps kontrolliert und diese werden, als Liste von Abweichungen der variablen *alt locus* Sequenz zur korrespondierenden Primärassemblyregion, in einer VCF-Datei ausgegeben. Diese Liste dient als Basis für die Definition des charak-

teristischen Fingerabdrucks aus ASDPs für jeden *alt locus*. ASDPs betreffen in den meisten Fällen einzelne Basen, jedoch gibt es auch Insertionen und Deletionen (InDels) unterschiedlichster Größe. Hier wird zwischen jenen unter 50 Basen und strukturellen ASDPs unterschieden. Der Cut-Off von 50 Basen wurde gewählt, da die meisten Varianten-Detektionsprogramme (z.B. FreeBayes (Garrison und Marth, 2012), GATK (McKenna u. a., 2010), ...) InDels nur zuverlässig callen können, wenn sie maximal der Hälfte der Länge der NGS-Reads entsprechen (R. Yang u. a., 2015). Eine typische Länge für NGS-Reads liegt bei 100 bis 150 Basen. Um den Fingerabdruck der *alt loci* mit den gecallten genomischen Varianten zu vergleichen, sollte der Größenbereich der Varianten möglichst einander entsprechen.

Dieser erste Schritt lieferte in dieser Arbeit eine Liste von 770 276 Kandidaten-ASDPs für Positionen, in denen sich die REF-HAP von ihren korrespondierenden ALT-HAP Sequenzen unterscheiden. Von den Kandidaten-ASDPs sind 768 316 Positionen SNVs, MNVs oder InDels bis zu einer Größe von 50 nt und die restlichen 1 960 sind strukturelle Unterschiede größer als 50 nt. Die Positionen entsprechen 661 805 einmaligen Positionen in den REF-HAP Sequenzen. Dies liegt darin begründet, dass REF-HAP Regionen teils mehrere *alt loci* besitzen und diese identische Unterschiede zum REF-HAP aufweisen können.

Es ist anzunehmen, dass NGS-Reads, die zu einem ALT-HAP gehören, eher auf die REF-HAP Sequenz aligniert werden, wenn es weniger Unterschiede zwischen den Sequenzen von ALT-HAP und REF-HAP gibt. Visuelle Inspektion der Alignments und die Auswertung der Kandidaten-ASDPs zeigen zwei Dinge: einerseits gibt es Regionen von mehreren tausend Nukleotiden, die komplett identisch sind oder sich nur in vereinzelten Positionen unterscheiden, andererseits solche, die so unterschiedlich sind, dass kein vernünftiges Alignment möglich ist. Dies ist auch in den Abbildungen 3.6 A-D gezeigt. Jeder dieser Abweichungen kann bei der Auswertung zu einer gecallten Variante führen, wenn Reads, die dem ALT-HAP entsprechen, fälschlicherweise auf die Sequenz des REF-HAPs abgebildet werden. Die

Wahrscheinlichkeit hierfür hängt von mehreren Faktoren ab, ein wichtiger ist die Sequenzähnlichkeit in der entsprechenden Region für das Alignment. Zu große Sequenzunterschiede würden nicht zur Abbildung der Reads vom ALT-HAP auf die Sequenz des REF-HAPs führen und damit auch nicht zu einem Variantencalling. Aus diesem Grund wurde die finale Liste von Kandidaten-ASDPs auf solche in Segmenten innerhalb der Alignments beschränkt, welche eine relativ hohe Qualität aufweisen, d.h. in denen sich die Sequenzen von ALT-HAP und REF-HAP nicht zu sehr unterscheiden und damit ein qualitativ hochwertiges, paarweises Alignment zulassen.

Die Güte der Alignments variiert entsprechend der Segmentsequenzen. Es gibt Bereiche mit hoher Sequenzähnlichkeit (Abbildung 3.6 A,B,C), die sich für den Fingerabdruck des *alt locus* nutzen lassen. Auf der anderen Seite gibt es Bereiche mit zerstückeltem Alignment (Abbildung 3.6 B,D), welche verworfen werden sollten, da sich das Muster so nicht in dem Alignment der NGS-Reads auf das Primärassembly wiederfinden lässt. Die ASDPs mit einer hohen Konfidenz werden über die Frequenz der gefundenen Kandidaten-ASDPs bestimmt. In dem Fenster um eine Abweichung darf es nur eine bestimmte Anzahl weiterer Unterschiede zwischen REF-HAP und ALT-HAP in Form von Mismatches und InDels geben. Dabei kann man sich an der Frequenz von Varianten in dbSNP orientieren. Die Varianten in dbSNP stellen, genau wie die *alt loci*, häufige Polymorphismen dar und basieren zum Teil auf den gleichen Daten. Dementsprechend sollten die ASDP-assoziierten Varianten in Relation zu diesen stehen, d.h. zahlreiche Varianten aus dbSNP in den Regionen mit *alt loci* sollten entsprechende Kandidaten-ASDPs haben.

Für die weitere Analyse wurde über die Liste der Kandidaten-ASDPs mit folgendem Filter iteriert: In einem Fenster von 50 Basen darf es in dem Alignment maximal zehn abweichende Positionen geben, ansonsten werden alle Kandidaten-ASDPs innerhalb des Fensters verworfen. Sowohl für die reinen ASDPs als auch für die dbSNP korrespondierenden ASDPs ist bei einer Fenstergröße von 50 nt eine Sättigung erreicht (siehe Abbildung 3.7 A und B). Die Überlappung von dbSNP-Einträgen mit ASDP-assoziierten

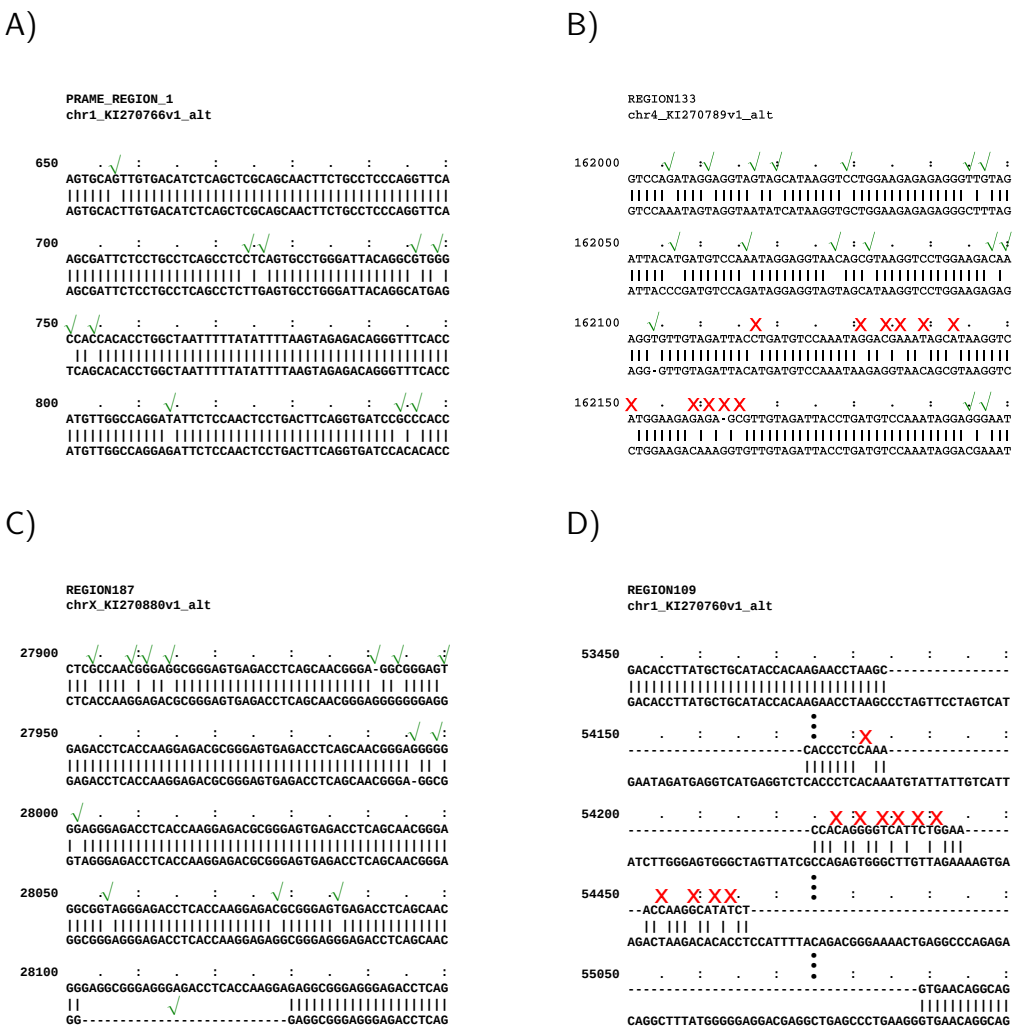


Abbildung 3.6: **Häufigkeit von ASDPs I**. Alignments enthalten Sequenzbereiche, welche zu einem großen Teil zwischen dem Primärassembly (oben) und dem *alt locus* (unten) identisch sind. Valide ASDPs sind als Abweichungen im Alignment zwischen REF-HAP und ALT-HAP definiert, für die es maximal zehn Positionen mit Abweichungen in einem fortlaufenden (sliding) Fenster von 50 Nukleotiden gibt – hier mit einem grünen Haken markiert. Die rot markierten Abweichungen erfüllen diese Anforderungen nicht. In A) & D) wurden keine ASDPs von dem sliding Fenster Ansatz rausgefiltert, wohingegen Bereiche mit geringer Sequenzähnlichkeit in B) zur Filterung von einigen Abweichungen führt. In D) führt eine Insertion im ALT-HAP zu einer größeren Anzahl von Abweichungen, welche die geforderten Kriterien nicht erfüllen.

Varianten ist im Verhältnis von 1 : 5 nahezu genauso hoch wie für 1 : 4 (Abbildung 3.7 A), jedoch ginge dies mit einer deutlich höheren Anzahl von verbleibenden Varianten einher (Abbildung 3.7 B), was sich als geringere Spezifität verstehen läßt. Die Performanz des Filters ist in Abbildung 3.6 gezeigt. Grüne Häckchen zeigen valide ASDPs (Unterschiede zwischen REF-HAP und ALT-HAP), wohingegen rote Kreuze diejenigen Kandidaten-ASDPs markieren, welche heraus gefiltert werden.

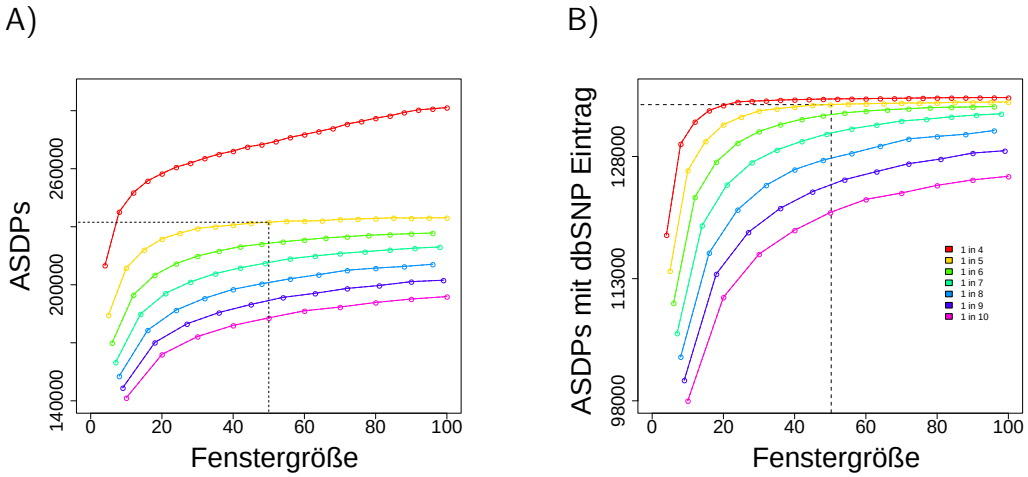


Abbildung 3.7: **Häufigkeit von ASDPs II**. Validierung der getesteten Fenstergrößen und Anzahl von Mismatches. Die gestrichelten Linien zeigen die final gewählten Grenzwerte an (Fenstergröße von 50 nt mit maximal zehn Mismatches). A) Der Effekt, den verschiedene Grenzwerte für die erlaubten Unterschiede zwischen REF-HAP und ALT-HAP zusammen mit der verwendeten Fenstergröße auf die Anzahl der validen ASDPs hat. B) Anzahl von ASDPs, die mit dbSNP Varianten, entsprechend der verschieden gewählten Grenzwerte für Mismatches und Fenstergröße, überlappen.

Nach dem Filtern der Kandidaten-ASDPs, mit den genannten Kriterien, verbleiben 232 333 Alignmentpositionen, welche letztendlich als ASDPs mit hoher Konfidenz bezeichnet werden können. 187 080 (80, 5%) von diesen ASDPs entsprechen SNPs und die restlichen Deletionen, Insertionen und MNVs der Größe 1 bis 50 nt (Tabelle 3.2). In vielen Fällen, wo eine Region

mehrere *alt loci* besitzt, finden sich ASDPs von verschiedenen ALT-HAPs auf derselben Position des Primärassemblies. Diese ASDPs können eine identisch abweichende Nukleotidsubstitution beschreiben. Von den 232 333 abweichenden Alignmentpositionen entsprechen 137 156 unterschiedlichen Positionen auf dem Primärassembly. Dies entspricht $\sim 2,2$ Variationen pro Kilobase REF-HAP Sequenz[6], die mit einer ASDP assoziiert sind.

ASDP-Kategorien	Anzahl	Prozent [%]
SNV	187 080	80,5
Deletion	15 955	6,9
Deletion (2 nt)	6 368	2,7
Deletion (3 nt)	2 413	1,0
Deletion (4–50 nt)	7 174	3,1
Insertion	15 286	6,6
Insertion (2 nt)	6 423	2,8
Insertion (3 nt)	2 224	1,0
Insertion (4–50 nt)	6 639	2,9
MNV	14 012	6,0
MNV (2 nt)	11 659	5,0
MNV (3 nt)	1 653	0,7
MNV (4–50 nt)	700	0,3

Tabelle 3.2: **Verteilung der ASDP-Kategorien**. Die 232 333 ASDPs mit hoher Konfidenz, welche aus dem Vergleich der Alignments zwischen *alt loci* und Primärassembly von dem Algorithmus und nach Anwendung der Qualitätsfilter übrig geblieben sind, verteilen sich auf rund 80% SNVs, jeweils 7% Insertionen und Deletionen und 6% MNVs.

In WGS-Daten kann das Muster der Verteilung der ASDP-assoziierten Varianten mit einem Fingerabdruck verglichen werden, der die Präsenz der ALT-HAP Sequenzen für die entsprechende Region, der REF-HAP Sequenz oder eine heterozygoten Kombination aus REF-HAP und ALT-HAPs widerspiegelt. Das bedeutet ASDPs sind mit charakteristischen Mustern von

[6]Die Gesamtlänge der REF-HAP Sequenzen beläuft sich auf 61 896 414 Basen über 178 Regionen verteilt.

gecallten Varianten auf dem REF-HAP assoziiert.
An diesem Punkt soll noch einmal auf die Abbildung der `REGION148` und deren *alt locus* vom Anfang des Kapitels verwiesen werden. Es ist ein Beispiel dafür, wie in einem *in-house* Genom (Patient des Instituts für Medizinische und Humangenetik der Charité Berlin) ein ALT-HAP das Calling der Varianten auf dem Primärassembly beeinflußt. In der Abbildung 3.2 `A` sieht man zahlreiche in `REGION148` homozygot gecallte ASDP-assoziierte Varianten und zusätzlich eine heterozygote nicht-ASDP-assoziierte Variante. Abbildung 3.2 `B` zeigt die korrespondierende Region auf dem *alt locus* `KI270808.1`. Nur die einzelne, heterozygote nicht-ASDP-assoziierte Variante wurde gecallt. Eine plausible Schlußfolgerung daraus ist, dass der sequenzierte Proband homozygot für `KI270808.1` ist. Desweiteren kann man annehmen, dass die homozygot gecallten Varianten in `REGION148` nicht echt sind, das sequenzierte Individuum in der Region auf Chromosom 7 nicht die Sequenz des Primärassemblies, sondern die des *alt locus* `KI270808.1`, hat. Dies impliziert, dass das Individuum nur die einzelne Variante in `KI270808.1` besitzt, alle anderen gecallten ASDP-assoziierten Varianten können als irrelevant interpretiert werden.

Definition ASDP

Das Akronym ASDP (Alignable Scaffold-Discrepant Position) bezieht sich auf eine divergente Position im Alignment zwischen der REF-HAP und der ALT-HAP Sequenz und nicht auf eine gecallte Variante. In dieser Arbeit wird jedoch gezeigt, dass viele Varianten im Whole Genome Sequencing mit ASDPs überlappen. Diese Varianten werden als *ASDP-assoziierte Varianten* betitelt.

3.4 ASDPex

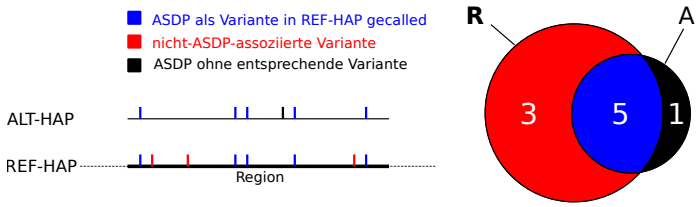
ASDPex – der ASDP-extraktions-Algorithmus – wurde in dieser Arbeit entwickelt um individuelle VCF-Dateien zu analysieren und darin die ASDP-

assoziierten Varianten zu markieren, um sie bei Bedarf für die Anschlußanalysen herauszufiltern. Dazu implementiert es eine Heuristik, welche die Verteilung der ASDP-assoziierten und anderer Varianten im REF-HAP und ALT-HAP betrachtet, um aus ihnen die wahrscheinlichste Kombination von Haplotypen für jede der 178 genomischen Regionen mit *alt loci* zu bestimmen.

Hierfür scannt ASDPex jede der 178 Regionen und vergleicht die ASDP-assoziierten Varianten aller dazugehörigen *alt loci* mit dem Primärassembly. Hierfür werden zwei Mengen betrachtet. $\mathcal{R}$ ist die Menge aller auf dem Referenzhaplotypen (REF-HAP) gefundenen Varianten. Dies umschließt ASDP-assoziierte Varianten wie auch alle weiteren gecallten Varianten in der entsprechenden Region. Die zweite Menge $\mathcal{A}$ ist die Menge aller ASDPs, welche mit dem jeweiligen *alt locus* assoziiert sind, also die Unterschiede zwischen REF-HAP und ALT-HAP im Alignment beschreiben. Besitzt eine sequenzierte Probe nicht den ALT-HAP, so erwartet man, in den Varianten auf dem REF-HAP wenige bis keine ASDP-assoziierten Varianten zu finden und würde für alle Varianten auf dem REF-HAP von richtig positiven Varianten ausgehen. Umgekehrt erwartet man, dass die meisten, wenn auch nicht notwendigerweise alle ASDPs eine korrespondierende ASDP-assoziierte Variante auf dem REF-HAP haben werden, wenn die Probe den ALT-HAP zumindest heterozygot vorliegen hat. Die restlichen Varianten, welche nicht-ASDP-assoziiert sind, und alle ASDPs, die nicht in der Referenzsequenz als Varianten auftauchen, werden vom Algorithmus als Residualvarianten `RV` der Probe definiert. Die Anzahl der Residualvarianten `RV` kann als symmetrische Mengendifferenz $\texttt{RV} = \mathcal{R} \triangle \mathcal{A}$ der Mengen $\mathcal{R}$ und $\mathcal{A}$ berechnet und zur Entscheidung herangezogen werden, welche Haplotypen am wahrscheinlichsten vorliegen. In Abbildung 3.8 ist dies auch schematisch dargestellt.

Daraus läßt sich ableiten, daß `RV` sich über die Mengen $\mathcal{R}$ und $\mathcal{A}$ folgendermaßen bestimmen läßt:

A



B

Wahrer Genotyp	Called gegen	
	REF-HAP	ALT-HAP
REF-HAP/REF-HAP	0/0	1/1
REF-HAP/ALT-HAP	0/1	0/1
ALT-HAP/ALT-HAP	1/1	0/0

Abbildung 3.8: **Überblick über den ASDPex Algorithmus**
(A) ASDPex vergleicht die Menge aller Varianten im REF-HAP ($\mathcal{R}$) mit der Menge der ASDPs assoziiert mit dem ALT-HAP ($\mathcal{A}$). In diesem Beispiel ist $|\mathcal{A}|$ (die Anzahl der ALT-HAP assoziierten ASDPs) 6 und $|\mathcal{R}|$ (die Gesamtzahl der in REF-HAP gecallten Varianten) 8. ASDPex definiert die Menge der Residualvarianten als die symmetrische Mengendifferenz zwischen $\mathcal{R}$ und $\mathcal{A}$ gleich $\texttt{RV} = \mathcal{R} \triangle \mathcal{A}$. Damit ist für dieses Beispiel $|\texttt{RV}| = 4$. Da hier $|\texttt{RV}| < |\mathcal{R}|$ gilt, schließt der Algorithmus auf die Präsenz des ALT-HAP. **(B)** Das Muster der gecallten Varianten unterscheidet sich entsprechend, ob der Proband homozygot oder heterozygot für einen oder zwei *alt loci* ist. Der Algorithmus betrachtet das Variantenmuster über die gesamte Länge der Region mit *alt loci*, um den wahrscheinlichsten Genotypen daraus zu schließen.

$$\texttt{RV} = \mathcal{R} \triangle \mathcal{A} = (\mathcal{R} \setminus \mathcal{A}) \cup (\mathcal{A} \setminus \mathcal{R}) = \mathcal{R} \cup \mathcal{A} \setminus (\mathcal{R} \cap \mathcal{A})$$

Hierbei ist $\mathcal{R} \setminus \mathcal{A}$ die Menge der nicht-ASDP-assoziierten Varianten, welche im Alignment gegen REF-HAP gecalled wurden, und $\mathcal{A} \setminus \mathcal{R}$ die Menge der ALT-HAP assoziierten ASDPs, welche nicht in REF-HAP gecalled wurden.

Unter der Annahme, dass der ALT-HAP tatsächlich vorliegt, kann dies entweder eine falsch-negative Variante sein (aufgrund schlechter Abdeckung etc.) oder, wie im Beispiel Abbildung 3.2, eine Variante in der REF-HAP Sequenz. Unter diesen Bedingungen ist es leicht zu sehen, dass die Anzahl der Residualvarianten $|\texttt{RV}| = |\mathcal{R}| + |\mathcal{A}| - 2 * |\mathcal{R} \cap \mathcal{A}|$ ist.

Die Idee des Algorithmus basiert auf der Annahme, dass derjenige Haplotyp am wahrscheinlichsten vorliegt, der die geringste Anzahl von Varianten hat. Es wird versucht, die Anzahl der Residualvarianten zu minimieren. Gilt $|\texttt{RV}| > |\mathcal{R}|$, dann nimmt ASDPex an, dass der *alt locus* nicht vorliegt. Diese Voraussetzung bedeutet, dass mehr Varianten auf dem ALT-HAP, als auf dem REF-HAP gecalled wurden. Für $|\texttt{RV}| = |\mathcal{R}|$ ist die Wahrscheinlichkeit gleich hoch, dass REF-HAP oder ALT-HAP vorliegen.

Für den Fall $|\texttt{RV}| < |\mathcal{R}|$ wären mehr Varianten mit REF-HAP assoziiert, als wenn der entsprechende ALT-HAP vorliegt. ASDPex würde in diesem Fall also die Präsenz des ALT-HAP vorhersagen.

Hat der Algorithmus einmal bestimmt, dass ein *alt locus* Haplotyp vorliegt, so versucht er anhand der Allelfrequenzen der vorliegenden ASDP-assoziierten Varianten zu bestimmen, ob er homozygot oder heterozygot vorliegt. Hierfür bestimmt er das Verhältnis zwischen den hetero- und homozygot vorliegenden Varianten, welche ASDPs entsprechen. Liegt der Anteil der homozygoten ASDP-assoziierten Varianten über einem bestimmten Grenzwert, so annotiert ASDPex den *alt locus* als homozygot, ansonsten heterozygot. Der für die hier gezeigten Ergebnisse verwendete Grenzwert wurde auf 90% festgelegt (siehe auch Algorithmus 4).

Ist die Region R mit mehr als einem *alt locus* assoziiert, so wird bestimmt, welcher dieser alternativen Scaffolds gegebenenfalls vorliegt. Hierfür wird die Anzahl der Residualvarianten RV für alle *alt loci* bestimmt. Der Haplotyp mit dem geringsten Wert für $|\texttt{RV}|$ ist der beste Kandidat. Liegt dieser heterozygot vor, so wird geschaut, ob für den zweitbesten ALT-HAP ebenfalls $|\texttt{RV}| < |\mathcal{R}|$ gilt.

Die so identifizierten ASDP-assoziierten Varianten werden von ASDPex in

Algorithmus 4 Der ASDPex Algorithmus bestimmt zu einer Region $\mathcal{R}$, mittels der Liste $\mathcal{A}$ der mit $\mathcal{R}$ assoziierten ASDPs und den Genotypdaten $\mathcal{G}$ der Probe (z.B. aus einer VCF-Datei), den wahrscheinlichsten haploiden Genotyp für Region $\mathcal{R}$ (homozygot REF-HAP, homozygot ALT-HAP$_1$, heterozygot REF-HAP/ALT-HAP$_1$ oder ALT-HAP$_1$/ALT-HAP$_2$). Die auf der Projektseite zur Verfügung gestellte Implementierung annotiert ASDP-assoziierte Varianten in einer VCF-Datei mit einer ASDP-Markierung, sobald der Genotyp nicht als homozygot REF-HAP definiert wurde.

```
ASDPex(R, A, G)
 1  A ⇐ G in R // Varianten in Region R
 2  B_1..n ⇐ A in R // ASDPs für ALT-HAP 1..n in Region R
 3  RV_1..n ⇐ A△B_1..n // Residualvarianten für ALT-HAP 1..n
 4  CUTOFF ⇐ 0.9 // Grenzwert für Homozygotie
 5  if |A| < |RV_1|
 6        Haplotyp ist homozygot REF-HAP
 7  else
 8        hom ⇐ count_homozygot_ASPD_associated_variants()
 9        N ⇐ count_all_ASPD_associated_variants()
10        if hom/N > CUTOFF
11              Haplotyp ist homozygot ALT-HAP_1
12        else
13              Haplotyp ist heterozygot
14              if n > 1 & |A| < |RV_2|
15                    Haplotyp ist heterozygot ALT-HAP_1/ALT-HAP_2
16              else Haplotyp ist heterozygot ALT-HAP_1/REF-HAP
17  return: wahrscheinlichster haploider Genotyp für Region R
```

der ausgegebenen VCF-Datei markiert. Der Eintrag in der `FILTER`-Spalte wird auf `ASDP` gesetzt und die `INFO`-Spalte erhält Informationen zu Region, *alt locus* und die Verteilung der Haplotypen (heterozygot/homozygot). In Abbildung 3.9 findet sich ein Auszug aus einer mit ASDPex annotierten VCF-Datei mit zwei ASDP-assoziierten Varianten.

Ein Schwachpunkt der Implementierung des Algorithmus ist, dass die Heuristik nur die Varianten auf dem kanonischen Primärassembly berücksichtigt, nicht jedoch diejenigen in den *alt loci*. Diese sollten theoretisch durch den paarweisen Alignmentvergleich zwischen Primärassembly und den *alt loci*

Mit ASDPex annotierte VCF-Datei

```
...
##FILTER=<ID=ASDP,Description="Filtered due to a more likely alternative scaffold">
##INFO=<ID=ALTGENOTYPE,Number=A,Type=String,Description="most likely alternate scaffold replacement
genotype">
##INFO=<ID=ALTLOCUS,Number=A,Type=String,Description="most likely alternate scaffold id replacement">
...
#CHROM  POS     ID    REF   ALT   QUAL     FILTER  INFO        FORMAT  SAMPLE
chr1    72126   .     G     C     1539.99  ASDP    ALTGENOTYPE=HOM_VAR;ALTLOCUS=chr1_KI270760v1_alt;...  GT  1/1
...
chr2    861114  .     C     T     620.70   ASDP    ALTGENOTYPE=HET;ALTLOCUS=chr2_GL383522v1_alt;...      GT  0/1
```

Abbildung 3.9: **Modifikationen einer VCF-Datei nach der Annotation mit ASDPex.** Oben sind die zusätzlichen VCF-header Einträge zu sehen, welche die neuen Schlagworte für die VCF-Einträge in den `FILTER` und `INFO` Spalten definieren. Der erste VCF-Eintrag ist eine homozygote Variante, welche in einem höchstwahrscheinlich ebenfalls homozygot vorliegenden *alt locus* liegt. Der zweite ist eine ASDP-assoziierte Variante in einer heterozygot annotierten Region.

abgedeckt sein, jedoch werden zum Beispiel die Varianten in Insertionen im *alt locus* nicht berücksichtigt.

Der vorhergesagte Genotyp für Probe A in Abbildung 3.2 wurde durch ASDPex als homozygot für den *alt locus* `KI270808.1` in `REGION148` vorhergesagt. Im Anhang finden sich in den Abbildungen A10–A12 weitere Beispiele, für die ASDPex entweder den *alt locus* Haplotyp heterozygot oder homozygot oder den REF-HAP homozygot vorhergesagt hat.

3.5 Validierung

Zur Validierung der ASDP-Definition und der Implementierung der Heuristik zur Haplotypvorhersage (ASDPex) sollen diese zu öffentlich verfügbaren Daten in Relation gesetzt werden und anschließend auf echten WGS-Daten getestet werden.

3.5.1 Alignment und Variantencalling

Die Prozessierung von WGS-Daten setzt sich aus den beiden Schritten *Alignment* auf ein Referenzgenom und dem anschließenden *Variantencalling* zusammen.

Bwakit[7] implementiert eine Pipeline für das Alignment von NGS-Reads auf die humanen Genomreleases GRCh37 und GRCh38. Dabei werden die Alignments gegen die Primärassemblies und die alternativen Locus Scaffolds separat berücksichtigt und Reads, welche mit einer hohen Qualität sowohl auf das Primärassembly als auch den *alt locus* mappen, werden auf diesem direkt als *supplementary* (zusätzliche) Alignments notiert. Damit wird verhindert, dass sie aufgrund der doppelten Positionierung eine schlechte Mappingqualität zugewiesen bekommen. Bwakit enthält ein Skript zum Download der Referenzsequenzen. Der Aufbau der bwakit Pipeline ist folgendermaßen. Im ersten Schritt werden mögliche Adapter mittels trimadapt[8] aus den Reads im FastQ-Format entfernt (`probe_R1.fastq.gz` & `probe_R2.fastq.gz`) und diese anschließend mit BWA-MEM (Li, Heng, 2013) gegen die Referenz aligniert. Das finale Alignment (`outfile`), im BAM-Format, entsteht durch die Sortierung der alignierten Reads durch Samtools (H. Li, 2011) und die Markierung von Duplikaten mittels SAMBLASTER (Faust und Hall, 2014). Das Program wurde auf einem Server mit 96 Kernen folgendermaßen aufgerufen:

```
run-bwakit -sd -t 96 -R <readgroup> -o <outfile>
-H hs38DH.fa probe_R1.fastq.gz probe_R2.fastq.gz
```

Die `readgroup` enthält Informationen zur Zuordnung der Probe zu einer ID, NGS-Bibliothek, ..., Sequenzierungs-Setup.

Auf den so generierten Alignments werden im Anschluß die Varianten (SNVs und InDels) mit FreeBayes (Garrison und Marth, 2012) identifiziert und

[7] https://github.com/lh3/bwa/tree/master/bwakit
[8] https://github.com/lh3/trimadap

mit vcflib[9] und vt (Tan, Abecasis und Kang, 2015) normalisiert (siehe Kapitel 2.2.2).

3.5.2 Datenquellen

Referenzgenome

Das erwähnte bwakit ermöglicht den Download der *hs37d5* und *hs38DH* Referenz. Die *hs37d5* Referenz setzt sich aus dem GRCh37.p13 Primärassembly, dem Epstein-Barr-Virus (EBV) Genom und den decoy Contigs[10], wie es vom 1000-Genome-Projekt (1000 Genomes Project Consortium u. a., 2010) Phase 3 verwendet wurde, zusammen. Die *hs38DH* Referenz besteht aus dem Primärassembly von GRCh38 inklusive der alternativen Locus Scaffolds, den decoy Contigs und zahlreichen HLA Alternativen.

Varianten und Annotationen

Der dbSNP-Release 146, welcher für GRCh37.p13 und GRCh38.p2 verfügbar ist und den verwendeten Genomreleaseversionen entspricht, wurde als VCF-Datei von der NCBI (NCBI Resource Coordinators, 2016) FTP-Seite für beide Genomreleases heruntergeladen. Für die Selektion von *common* (häufigen) Polymorphismen wurde die Definition von dbSNP übernommen[11] - eine Variante ist *common*, wenn sie mit einer MAF von $>= 0,01$ in mindestens einer der 1000-Genome-Projekt (1000 Genomes Project Consortium u. a., 2010) Phase 3 Populationen gefunden wurde und mindestens zwei Individuen aus zwei unterschiedlichen Familien das gleiche Minor Allel besitzen. Alle anderen polymorphen Einträge in dbSNP werden als selten betrachtet.

[9] Garrison E. Vcflib, a simple C++ library for parsing and manipulating VCF files. 2016. https://github.com/vcflib/vcflib.

[10] Das humane Referenzgenom ist immer noch unvollständig und es sind einige Sequenzen bekannt, die nicht im Genom platziert werden konnten. Diese Sequenzen werden decoy Contigs genannt und z.B. vom 1000-Genome-Projekt dazu verwendet, Reads nicht fälschlich auf das Primärassembly mappen zu lassen, sondern sie damit herauszufiltern.

[11] https://www.ncbi.nlm.nih.gov/variation/docs/glossary/

Annotationen für genomische Features (Exons und die codierende Sequenzen) für die Genomreleases GRCh37.p13 und GRCh38.p2 wurden ebenfalls von der NCBI FTP-Seite heruntergeladen. Die transkriptbasierten funktionellen Annotationen wurden mit Jannovar (Jäger u. a., 2014) in Version 0.16 durchgeführt.
Der Katalog für genomweite Assoziationsstudien (GWAS) (Welter u. a., 2014) stammt vom 1. Februar 2016. Er wurde von der Webseite[12] des European Bioinformatics Institute (EMBL-EBI) heruntergeladen.

1000-Genome-Projekt

Das 1000-Genome-Projekt stellt für insgesamt 27 Populationen aus fünf Superpopulationen (diese entsprechen den Ursprungs*sub*kontinenten Europa, Afrika, Ost-Asien, Süd-Asien und Amerika) genomische NGS-Daten zur Verfügung. In der Abbildung A9 findet sich dazu eine Übersicht. Es wurden für jeweils 30 Individuen von vier Populationen (FIN – Finnen aus Finland; LWK – Luhya aus Webuye, Kenya; CHB – Han Chinesen aus Beijing, China; PEL – Peruaner aus Lima, Peru) mit einer möglichst breiten geographischen Verteilung die WGS-Alignment Daten in Form von CRAM[13]-Dateien (Hsi-Yang Fritz u. a., 2011) runtergeladen. Mit Hilfe von Cramtools[14] wurden die ursprünglichen Rohdaten extrahiert und entsprechend der in Abschnitt 3.5.1 genannten Pipeline prozessiert.

In-house Daten

Für den auf aktuellen WGS-Daten basierenden Test standen 121 *in-house* Genome[15], mit einer 28–43-fachen mittleren Abdeckung (durchschnittlich 34-fach) und durchschnittlich 68% des Primärassemblies 30-fach abgedeckt zur Verfügung. Diese wurden von Macrogen (Seoul, Korea) auf einem Illumina HiSeq X-Ten System sequenziert. Gemäß Datenschutz und durch

[12] https://www.ebi.ac.uk/gwas/downloads
[13] CRAM3-Formatspezifikationen: https://samtools.github.io/hts-specs/CRAMv3.pdf
[14] https://github.com/enasequence/cramtools/
[15] Patienten des Instituts für Medizinische und Humangenetik der Charité Berlin

die Pseudonymisierung sind die Ethnien der Spender nicht bekannt, jedoch kann von einem europäischen und mediterranen Hintergrund ausgegangen werden. Auf alle 121 genomischen WGS-Proben wurde die oben genannte Pipeline zum Alignment und Variantencalling angewendet.

3.5.3 ASDPs als Polymorphismen

ASDPs stellen die Abweichungen der *alt loci* in den 178 Regionen zum Primärassembly dar. Genau wie diese als Polymorphismen ins GRCh38 Genomassembly aufgenommen wurden, sollten die Abweichungen (ASDPs) auch als Polymorphismen in den einschlägigen Variantendatenbanken für SNVs wiederzufinden sein.

ASDPs in dbSNP

Exemplarisch wurde nach SNPs und anderen kleinen Polymorphismen in dbSNP geschaut. dbSNP (b146) enthält 35 171 619 common SNP Einträge, von denen 826 612 (2, 35%) innerhalb der 178 Regionen mit *alt loci* liegen. Diese SNPs überlappen mit 75 138 ASDPs, was 71 653 eindeutigen Varianten auf dem REF-HAP entspricht[16]. Dies entspricht 32, 3% aller identifizierten ASDPs mit hoher Konfidenz.

ASDPs in GWAS

Die Frage, ob ASDPs auch klinisch von Bedeutung sein können, sollte anhand bekannter krankheitsassoziierter Varianten aus dem GWAS (Welter u. a., 2014) Katalog geklärt werden. Dieser enthält 18 130 SNPs (GWAS-Treffer), welche signifikant (P-Wert $<= 10^{-5}$) mit einer Krankheit oder einem Merkmal assoziiert sind.
791 (4, 36%) der GWAS-Varianten liegen innerhalb der 178 Regionen. Von diesen überlappen 437 mit ASDPs, wobei 360 dieser Treffer innerhalb der

[16]Für den Fall, dass es zu einem REF-HAP mehrere *alt loci* gibt, kann es vorkommen, dass die ASDPs der einzelnen *alt loci* identisch auf dem REF-HAP überlappen und damit mehrere ASDPs einer einzelnen Variante auf dem REF-HAP entsprechen.

MHC Region liegen. Ein Beispiel für ein ASDP, der mit einer GWAS-Variante assoziiert ist, ist in Abbildung 3.10 gezeigt. Insgesamt liegen 137 der 437 mit GWAS-Varianten überlappenden ASDPs in Bereichen, in denen es in den 25 davor und dahinter liegenden Basen im Alignment keine weiteren Abweichungen gab. Das deutet darauf hin, dass diese Varianten als Indikator für den *alt locus* dienen und nicht mit dem Phänotyp in Verbindung stehen. GWAS-Einträge stellen per Definition nur korrelative Zusammenhänge zwischen einem Allel mit der Variante und dem Phänotypen dar. Dazu werden Marker-SNPs verwendet, die das Genom möglichst gut abdecken und in der Regel nicht in codierenden, sondern intronischen und intergenischen Bereichen liegen (Pandey, 2010).

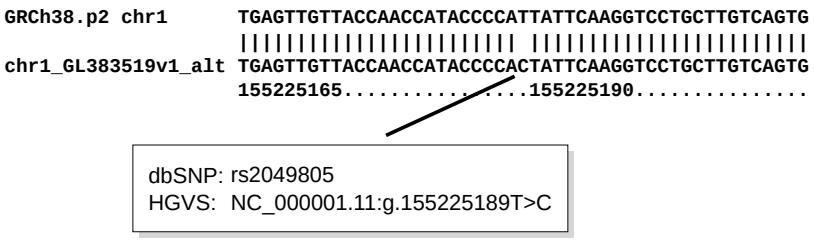


Abbildung 3.10: **rs2049805**. Der GWAS-Eintrag `rs2049805` entspricht einem ASDP, definiert durch ein Alignment zwischen Chromosome 1 des Primärassemblies (Region MTX1) und dem *alt locus* `GL383519.1`. Der gezeigte Ausschnitt zeigt ein perfektes Alignment über 49 Nukleotide mit Ausnahme der mittleren Position. `rs2049805` ist nach GWAS signifikant mit dem Blut-Harnstoff-Stickstoff Level in der ostasiatischen Population assoziiert (Okada u. a., 2012).

Populationsfrequenzen

Die Sequenzierdaten vom 1000-Genome-Projekt wurden verwendet, um die 178 Regionen auf einen populationsspezifischen Bias bezüglich der *alt loci* zu untersuchen. Für die vier Populationen (FIN, LWK, CHB, PEL) wurden jeweils 30 Genome runtergeladen und mit der beschriebenen Pipeline aus bwakit und Freebayes prozessiert. Die daraus resultierenden VCF-Dateien

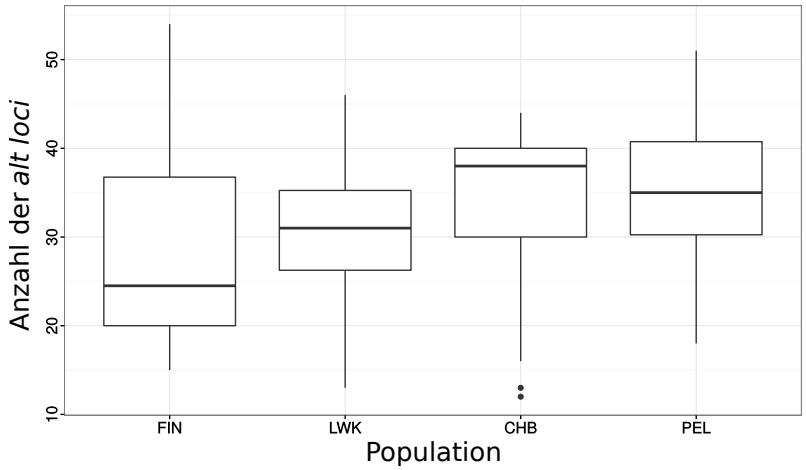


Abbildung 3.11: **Anzahl annotierter *alt loci* pro Population**. Für vier Populationen mit jeweils 30 Individuen wurde die Anzahl der annotierten *alt loci* gezählt. **FIN**: Finnen aus Finnland; **LWK**: Luhya aus Webuye, Kenya; **CHB**: Han Chinesen aus Beijing, China; **PEL**: Peruaner aus Lima, Peru.

wurden anschließend mit ASDPex annotiert. Die Daten wurden dann auf die Anzahl der annotierten *alt loci* und deren Verteilung untersucht (siehe auch Abbildung 3.11 und Tabelle 3.3). Auffällig hierbei ist, dass die peruanische Population die höchste Anzahl von *alt loci* aufweist. Es fällt auf, dass in dieser Population besonders viele *alt loci* vorhanden sind, welche in allen untersuchten Individuen der Population gefunden wurden. Im Gegensatz dazu wurde die niedrigste mittlere Rate an annotierten *alt loci* bei den europäischen und afrikanischen Individuen gefunden. Dies mag damit zusamenhängen, dass diese Populationen ausgiebig studiert wurden und damit einen Bias ins aktuelle Genomassembly bringen.

3.5.4 GRCh37 vs. GRCh38

Neben der Aufnahme von zahlreichen neuen *alt loci* in GRCh38 wurden, wie oben erwähnt, zahlreiche Sequenzkorrekturen und neue Sequenzen zum Schließen von Lücken im vorherigen Release (GRCh37) hinzugefügt. Die all-

alt locus	FIN	LWK	CHB	PEL
chr4_KI270787v1_alt			✓	✓
chr5_GL383531v1_alt	✓			
chr5_GL949742v1_alt			✓	
chr6_GL383533v1_alt				✓*
chr6_KI270801v1_alt		✓	✓	✓*
chr9_GL383542v1_alt		✓		
chr11_JH159136v1_alt		✓		✓*
chr13_KI270839v1_alt	✓		✓	✓*
chr14_KI270844v1_alt			✓	✓
chr15_GL383555v2_alt	✓*		✓	✓*
chr18_GL383570v1_alt			✓	

Tabelle 3.3: **Populationsspezifische *alt loci*.** Die Tabelle zeigt alle *alt loci*, die durch ASDPex in mindestens 90% aller Individuen einer Population als präsent vorhergesagt wurden. Es werden Daten für folgende Regionen und Populationen gezeigt: Europa - FIN *Finnen aus Finnland*, Afrika - LWK *Luhya aus Webuye, Kenya*, Asien - CHB *Han Chinesen aus Beijing, China* und Süd-America - PEL *Peruaner aus Lima, Peru*. *Alt loci*, welche in allen Individuen einer Population präsent waren, sind mit einem Stern (*) markiert.

gemeine Varianten Calling Performanz wurde an 121 *in-house* Genomen getestet, welche mit BWA-MEM (Li, Heng, 2013) aligniert und die Varianten anschließend mit Freebayes (Garrison und Marth, 2012) gecalled wurden. Bis auf die Referenzsequenz waren alle Prozessierungsschritte identisch (siehe auch Abschnitt 3.5.1). Die Analysen wurden auf die Primärassemblies der Chromosomen (Tabelle A2) beschränkt. Für beide Genomreleases war die Rate der alignierten Reads mit einem Mittelwert von 99.8% ausgesprochen gut. Dies weist darauf hin, dass die Referenzsequenzen repräsentativ sind. Reads, welche sowohl auf das Primärassembly als auch einen alternativen Haplotypen abgebildet werden können, werden für die *alt loci* als „supplementary mapped reads“ (siehe auch die SAM-Format Spezifikatio-

nen) markiert. Wie erwartet, sieht man hier einen deutlichen Unterschied zwischen den beiden Genomassemblies. In den Alignments auf GRCh38 gibt es 100 mal mehr „supplementary mapped reads“ als auf GRCh37. Dies läßt sich mit der erheblich gestiegenen Anzahl von *alt loci* von neun (GRCh37) auf 261 (GRCh38) erklären.

Die Anzahl der gecallten Varianten und deren Phred-Scores sind vergleichbar zwischen den Genomreleases (siehe Tabelle 3.4).

ASDP-assoziierte Varianten konnten sowohl in solchen Proben gefunden werden, für die ASDPex einen *alt locus* Haplotyp vorhergesagt hat, als auch in Regionen von Proben, wo das Primärassembly vorliegt. Die Dichte der ASDP-assoziierten Varianten für diese Regionen unterscheidet sich jedoch deutlich, falls für die Region der REF-HAP oder ALT-HAP vorhergesagt wurde (siehe Abbildung 3.12)

Für einen Vergleich der Regionen mit *alt loci* zwischen dem aktuellen und dem vorherigen Genomrelease, wurden deren Koordinaten von GRCh38 auf GRCh37 übertragen. Hierfür wurde das Batch Coordinate Conversion Tool von UCSC (Hinrichs u. a., 2006) verwendet. Dieses findet zu den gegebenen genomischen Koordinaten eines Genoms die entsprechenden Koordinaten in einem anderen Genomrelease. Die Zuordnung erfolgt anhand von Tabellen mit entsprechenden Koordinaten für beide Genomreleases. Es kann vorkommen, dass eine Urspungsregion dabei in mehrere Zielregionen aufgeteilt wird. Für die `REGION116` konnte keine Entsprechung in GRCh37 gefunden werden. Sie ist daraufhin für die Berechnungen ausgeschlossen worden und somit nicht in den Statistiken enthalten.

ASDPex wurde auf die VCF-Dateien aller 121 Genome angewandt. Von den 178 Regionen wurden im Mittel 51,8 ± 3,8 bestimmt, für die der ALT-HAP besser zu den Sequenzierdaten als zu den Primärassemblysequenzen passte. Im Durchschnitt wurden 7 863 ASDP-assoziierte Varianten pro Genom gefunden. Dies entspricht 6,51% aller in den 178 Regionen lokalisierten Varianten (siehe Tabelle 3.5). Die meisten der ASDP-assoziierten Varianten überlappen mit dbSNP-Einträgen, jedoch haben rund 13% der

	GRCh37			**GRCh38**		
Chr	*common*	*rare*	*phred*	*common*	*rare*	*phred*
1	299 500	44 659	503,42	291 704	67 825	473,83
2	247 585	106 528	506,54	243 400	118 069	492,1
3	268 021	27 426	503,76	263 993	37 993	492,99
4	290 080	29 908	515,41	285 405	38 729	507,33
5	235 462	30 615	498,58	231 747	40 734	485,57
6	252 705	28 084	495,02	246 545	32 588	487,11
7	220 543	29 437	488,95	216 669	41 248	475,69
8	204 823	24 509	499,70	200 845	28 696	490,15
9	162 202	30 413	475,09	159 119	40 916	466,46
10	194 385	23 572	508,74	190 658	38 694	494,18
11	197 412	21 722	522,84	194 132	34 192	498,04
12	184 477	20 608	502,78	175 990	36 799	483,94
13	151 271	14 856	530,31	148 870	31 651	494,65
14	124 790	17 181	503,15	122 524	17 919	495,75
15	112 085	18 239	505,95	109 741	21 648	493,46
16	116 224	18 069	487,40	113 589	23 210	473,36
17	102 300	15 796	479,64	99 074	31 563	452,99
18	111 958	12 552	516,80	110 349	22 279	485,89
19	84 416	13 688	456,51	82 875	16 750	455,35
20	79 709	10 781	486,09	78 562	33 999	475,40
21	55 211	14 300	525,23	53 052	19 975	513,27
22	50 242	9 418	455,99	48 961	22 151	445,27
Total	**4307761**			**4465432**		

Tabelle 3.4: **Anzahl der Varianten für die beiden Genomassemblies**. Gezeigt werden Mittelwert und Median der Phred-Scores der autosomalen Varianten pro Chromosom für GRCh37 und GRCh38. Spalten: *alle*: alle gefundenen Varianten; *common*: Varianten aufgelisted in den dbSNP *common_all_** Dateien; *rare*: Varianten, die nicht *common* sind. Die Mittelwerte der Varianten für Chromosom X sind 127 914 (GRCh37) und 132 177 (GRCh38). Für Chromosom Y konnte der Mittelwert der Varianten nicht bestimmt werden, da die Informationen zum Geschlecht der 121 *in-house* Genome nicht gegeben war. Beide Genomreleases enthalten dieselbe mitochondriale Referenz (`NC_012920.1`) mit jeweils 27 Varianten.

pro Genom gefundenen ASDP-assoziierten Varianten keine Entsprechung in dbSNP (siehe Abbildung 3.13). Für einige dieser Varianten wurde durch Jannovar ein hoher Effekt vorhergesagt (siehe Tabelle A3).

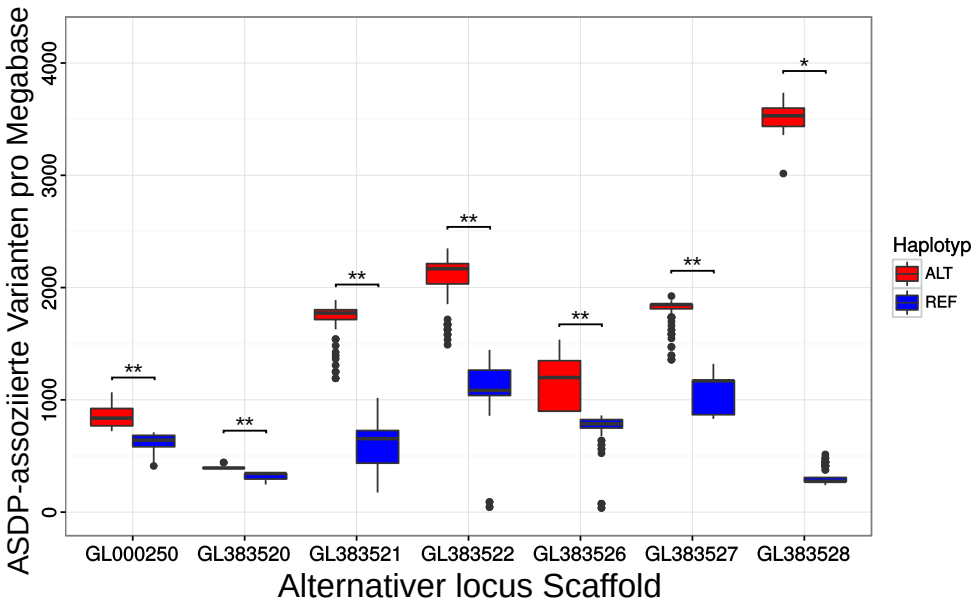


Abbildung 3.12: **Verteilung der ASDP-assoziierten Varianten auf den Primärassemblies.** Eine signifikant höhere Anzahl von ASDP-assoziierten Varianten kann auf dem Primärassembly gefunden werden, abhängig davon, ob der ASDPex Algorithmus die Region als REF-HAP oder ALT-HAP vorhergesagt hat. Die Abbildung zeigt die Anzahl der ASDP-assoziierten Varianten in sieben ausgewählten Regionen mit den entsprechenden *alt loci* für die 121 in-house Genome. Die Signifikanz der Unterschiede innerhalb der Gruppen wurde mit dem Mann-Whitney Test ermittelt: $* = p < 1 \times 10^{-8}$; $** = p < 1 \times 10^{-10}$

3.6 Schlusswort

Der Algorithmus wurde ASDPex genannt, da er dazu bestimmt ist, ASDP-assoziierte Varianten aus VCF-Dateien zu filtern oder diese in den Dateien zu markieren. Es kann für die Analyse der Varianten vorteilhaft sein, wenn diese weiter gefiltert und eindeutig ASDP-assoziierte Varianten ausgeschlossenen werden. Betrachtet man das Beispiel in Abbildung 3.2, so scheint es sinnvoll, die 52 in `REGION148` identifizierten Varianten zu entfernen und nur die eine verbliebene nicht-ASDP-assoziierte Variante beizubehalten, welche

Genomrelease	Anzahl Varianten total	Varianten pro Mb
GRCh37 canonical	114.023 ± 4 983	2 198,3 ± 207,6
GRCh38 canonical	120.807 ± 4 069	1 975,2 ± 66,5

Tabelle 3.5: **Reduzierung der Varianten durch ASDPex**. Gezeigt sind die Anzahl von Varianten in den *alt loci* enthaltenen Regionen. Für GRCh37 wurde ein *liftover*[17] der Regionen durchgeführt. `REGION116` wurde für beide Datensätze nicht betrachtet, da hierfür keine entsprechende Region in GRCh37 bestimmt werden konnte. Da sich die Gesamtlänge der Regionen für GRCh37 und GRCh38 unterscheiden, wurde neben der Gesamtanzahl der Varianten auch die Anzahl der Varianten pro Megabase (Mb) angegeben. Im Duchschnitt wurden pro Genom 7 863 ± 2 675 (6,5%) der Varianten als ASDP-assoziiert durch ASDPex annotiert. Dies entspricht einer Reduktion der Varianten pro Mb von 1 975,2 ± 66,5 auf 1 846,7 ± 71,6.

auch in `KI270808.1` gefunden wurde. Ein weiteres Beispiel ist die GWAS-Variante aus Abbildung 3.10, welche einem ASDP mit hoher Konfidenz entspricht.

In dieser Arbeit wurden alle Alignments mit BWA-MEM durchgeführt. Da ASDPex nur auf den Varianten der Primärassemblies arbeitet, kann man für dessen Funktionalität jedes Alignmentprogramm verwenden, welches NGS-Reads auf das Primärassembly mapped. Die Entscheidung BWA-MEM zu verwenden, war eine rein praktische und der Tatsache geschuldet, dass es mit bwa.kit eine Implementation gibt, welche die alternativen Locus Scaffolds und die zumindest duplizierten Ankersequenzen berücksichtigt. Damit verhindert es, dass Reads in den Regionen mit *alt loci* einen schlechteren Alignmentscore bekommen. Ein weiterer Vorteil ist, dass die NGS-Reads, die sowohl auf Primärassembly als auch einen *alt locus* mappen, auf den *alt loci* als *supplementary* Reads markiert werden. Dadurch lassen sich die Alignments auf den *alt loci* gut visualisieren.

Alle für die hier gezeigten Ergebnisse verwendeten Skripte und Algorithmen kann man auf der ASDPex GitHub-Seite unter `https://github.com/`

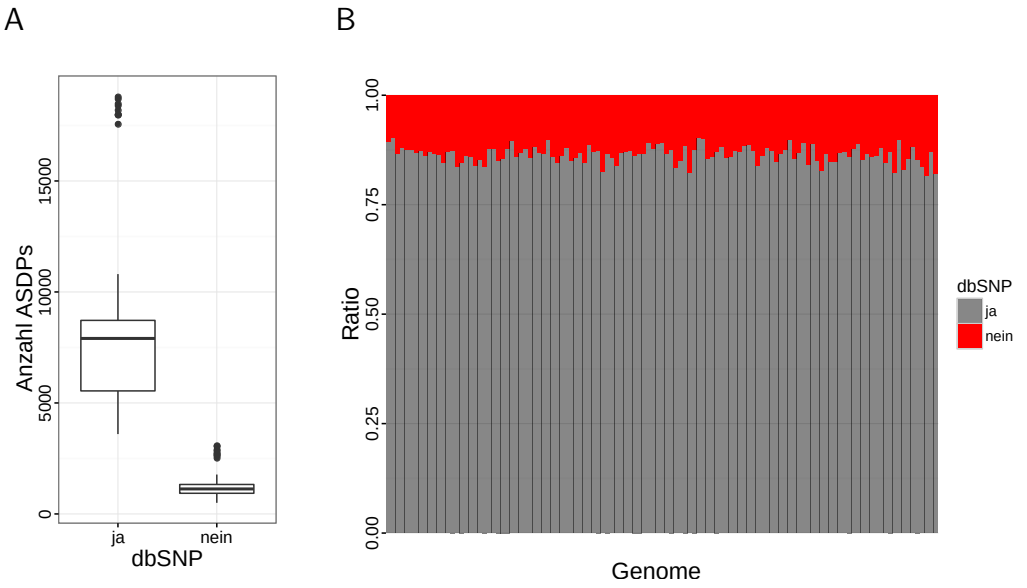


Abbildung 3.13: **In-house ASDPs mit dbSNP-Annotation**. A) Der Plot zeigt die Verteilung der ASDP-assoziierten Varianten in den *in-house* Genomen, die sich auch in dbSNP wiederfinden lassen. In B) sind für alle 121 Genome einzeln die Verhältnisse von ASDP-Varianten mit und ohne entsprechenden dbSNP-Eintrag gezeigt. Im Schnitt sind 13,48% der ASDP-assoziierten Varianten nicht in dbSNP enthalten. Der hohe Überlapp unterstreicht die Annahme, dass die Unterschiede in den *alt loci* zu den korrespondierenden Primärsequenzen als Polymorphismen in den entsprechenden SNV Datenbanken wiederzufinden sind.

`charite/asdpex` finden.

Kapitel 4

Diskussion

Der initiale Gedanke des HGP war es, ein Modell des humanen Genoms zu erstellen, welches einen einzelnen Pfad durch ein nicht-redundantes Referenzgenom, dem sogenannten *golden path*, verwendet. Das Modell passte auch sehr gut zu der ursprünglichen Annahme, dass SNVs den überwiegenden Anteil der Variationen ausmachen (Church, Valerie A Schneider, Graves u. a., 2011). An diesem Modell wurde bis zum humanen Genom Build 36 festgehalten. Es zeichnete sich jedoch schon deutlich früher ab, dass dies ein Eukaryontengenom zu stark vereinfacht, da es deutlich mehr populationsspezifische strukturelle Genomeigenschaften gibt (Feuk, Carson und Scherer, 2006), als ursprünglich angenommen. Dies zieht ebenfalls noch nicht die individuellen Varianten in Betracht.
Die Wahl der Referenz ist insbesondere bei der Exom- und Genomsequenzierung, welche neben der Forschung nun zunehmend auch in der Diagnostik Anwendung findet (Stark u. a., 2017; Taylor u. a., 2015; Veltman und Lupski, 2015), wichtig und sollte die Struktur des Patientengenoms möglichst genau abbilden (X. Yang u. a., 2019). Dies ist essentiell für die Interpretation von Patientendaten im klinischen Kontext, da diverse Varianten mit dem vereinfachten Modell als klinisch relevant erscheinen können, unter Berücksichtigung der Population jedoch klar wird, dass diese entsprechend dem ethnischen Hintergrund als Normvariante einzustufen sind. Diese Erkenntnis resultierte in der Erstellung eines sogenannten Graphengenoms,

welches in Kapitel 3 ausführlich beschrieben wurde. Die Idee dabei ist, für Regionen mit einer besonders häufigen populationsspezifischen Abweichung zum *golden path*, eine oder mehrere Alternativen anzubieten.
In diesem Zusammenhang ist es wichtig zu sehen, dass Innovationen und Fortschritte der NGS Technologien und Strategien sich in den letzten Jahren deutlich weiterentwickelt haben. Auf dem Markt erscheinen kontinuierlich neue Anbieter und Plattformen (MGI: MGI-SEQ 2000 – 2017; Thermo Fisher Scientific: Ion GeneStudio S5 – 2018; MGI: MGI-SEQ-T7 – 2018; Oxford NanoPore: PromethION – 2019, PacBio: Sequel II System – 2019). Diese Konkurrenz verhindert eine Stagnation in der Entwicklung und resultiert in der Regel in einer deutlichen Verbesserung der Qualität, Quantität und Sequenzierlänge, woraus sich neue Anwendungen für die NGS-Sequenzierung eröffnen.
Die wichtigste Konsequenz aus diesen technischen Fortschritten im NGS-Bereich ist ein deutlich höherer Sequenzierdurchsatz, da in der gleichen Zeit eine größere Anzahl von Megabasen sequenziert werden kann. Eine andere technische Weiterentwicklung ist die Erhöhung der Länge der sequenzierten NGS-Reads von ca. 2x150 bp auf mehrere Kilobasen, was die Assemblierung und Untersuchung von strukturellen Varianten begünstigt (Boldogkői u. a., 2019). Die in dieser Arbeit vorgestellten Programme und Ansätze sind aufgrund dieser Weiterentwicklungen auch in Zukunft äußerst relevant.

4.1 Jannovar

Die zuvor erwähnte Steigerung der Anzahl der sequenzierten Basen hat einen direkten Einfluss auf die Anforderungen an die Software, die diese Daten prozessieren. Modulare Softwarekomponenten – für die einzelnen Abschnitte einer NGS-Analyse Pipeline, wie die Annotation und Evaluierung der gefunden Varianten – haben das Potential, die Analyse und Auswertung in der Diagnostik zu vereinfachen und zu beschleunigen. Durch die Erhöhung des Sequenzierdurchsatzes sind die Programme angehalten,

möglichst schnell zu arbeiten, um der erhöhten Probenzahl gerecht zu werden. Häufig werden in der Diagnostik Multi-Gen Panel, also Sets von Genen, welche direkt mit einem Krankheitsbild assoziiert sind, verwendet. Durch die Beschleunigungen wären eine Ausweitung der Analysen auf größere Genmengen möglich, was ein umfassenderes Gesamtbild liefern kann. Das in Kapitel 2 vorgestellte Programm *Jannovar* ist für solche Anforderungen bestens vorbereitet. Die Laufzeit von *Jannovar* wird es auch in Zukunft erlauben, die Daten von tausenden Patienten in einer annehmbaren Zeit zu annotieren. Zusammen mit meinen Kooperationspartnern an der Charité, am Berlin Institute of Health, am Jackson Laboratory und der Genomics Community[1] werden wir auch in Zukunft *Jannovar* weiter verbessern und die Laufzeit für höhere Datendurchsätze optimieren. Ein Schritt hierbei ist die Ersetzung des sehr schnellen Intervallbaums durch ein Intervallarray[2].

4.2 ASDPex

In dieser Arbeit habe ich ASDPs eingeführt, welche die Unterschiede im Alignment von Sequenzabschnitten darstellen, die zu großen Teilen identisch zwischen Primärassembly und alternativem Locus sind. ASDPs sind typischerweise mit einem charakteristischen Muster von gecallten Varianten auf dem Primärassembly assoziiert, welche einem bestimmten *alt locus* entsprechen. Die durch die Pipeline in dieser Arbeit gefunden ASDP-assoziierten Varianten können in WGS Daten von unterschiedlichen Populationen gefunden werden (Abbildung 3.11 und Tabelle 3.3).

Eine möglichst präzise Kenntnis über den Aufbau des betrachteten Genoms ist für die Auswertung und Interpretation in der klinischen Diagnostik enorm wichtig (X. Yang u. a., 2019). Die Fortschritte in den Sequenziertechniken erlaubten mittels neuer Verfahren auch die Länge der NGS-Reads drastisch zu erhöhten. Dadurch ist eine verlässlichere Zuordnung der NGS-

[1] https://github.com/charite/jannovar/network/members
[2] https://github.com/charite/jannovar/releases/tag/v0.12

Reads zu einem bestimmten Pfad im Graphengenom möglich, zumal man zusätzlich die Informationen über kleinere Strukturvarianten für die Bestimmung verwenden kann. Dieser Ausblick motiviert dazu, eine Software zu entwickeln, um die Pfadinformation zum Referenzgenom in der klinischen Diagnostik verwenden zu können. Damit könnte die diagnostische Rate weiter erhöht werden.

Mit ASDPex wurde in dieser Arbeit solch eine Software vorgestellt (siehe Kapitel 3), welche dies auch für kurze Illumina NGS-Reads erreicht. ASDPex erlaubt es schon jetzt, anhand des Musters der ASDPs, eine Vorhersage zum wahrscheinlichsten Haplotypen in bestimmten Regionen zu treffen. Mit dieser Information können zahlreiche Varianten noch einfacher als Polymorphismen klassifiziert werden, die ohne Berücksichtigung der Pfadstruktur des Genoms als potentiell pathogen angesehen werden würden. Dies ist ein entscheidender Faktor in der klinischen Forschung und Diagnostik, da somit insbesondere der Zeitaufwand in der Analyse verringert werden kann.

Die Bestimmung von Varianten ist immer kontextabhängig. Was eine Variante ist, hängt vor allem von der verwendeten Referenz ab. Bei der Transition von GRCh37 auf GRCh38 wurden zum Beispiel in rund 10.000 genomischen Positionen die Basen angepasst, so dass einige Basen, die eine Variante in GRCh37 waren, nicht mehr als Variante in GRCh38 gefunden werden würden. In den meisten Fällen sind diese Varianten einfach Fehler, das heißt nicht die repräsentativste Variante in GRCh37 gewesen, welche in GRCh38 korrigiert wurden. Ein Beispiel hierfür ist die Base `chr15:48807637C`. Vergleicht man den RefSeq Eintrag für das FBN1-Gen (`NM_000138`) mit der genomischen Sequenz von GRCh37, würde es eine als pathogen vorhergesagt Variante `FBN1|NM_000138.3|c.1415G>A|p.Tyr472Cys` geben. Diese genomische Base wurde in GRCh38 zu einem `T` korrigiert. Das hat zur Folge, dass teilweise aufwändige Neuberechnungen stattfinden müssen um genomische Datenbanken, wie zum Beispiel die des 1000-Genome-Projekts (Zheng-Bradley u. a., 2017), zu aktualisieren.

Die repräsentativste Base in dem Genomrelease ist immer von der zugrundeliegenden Population und deren Subpopulationen abhängig. In diesem Sinne sind die in dieser Arbeit charakterisierten ASDP-assoziierten Varianten nicht falsch-positiv und lassen sich häufig in WGS Daten wiederfinden (siehe zum Beispiel Abbildung 3.12 und 3.13). Stattdessen kann die Verteilung der ASDP-assoziierten Varianten in den 178 Regionen mit einem Fingerabdruck verglichen werden, der das Vorhandensein einer der ALT-HAP-Sequenzen, der REF-HAP-Sequenz oder deren heterozygoter Kombination anzeigt. Die Verteilung der ASDP-assoziierten Varianten kann genutzt werden, um daraus abzuleiten, dass Variantenaufrufe gegen eine Region mit struktureller Variation im Primärassembly des GRCh38 als zweifelhaft zu betrachten sind. Veranschaulichen läßt sich dies mit der vorliegenden Probe in Abbildung 3.2. Sie weist mit hoher Wahrscheinlichkeit eher homozygot die ALT-HAP-Sequenz (`KI270808.1`) in diesem Abschnitt des Genoms auf. Die gefundenen Varianten, im Vergleich gegen das Primärassembly, können als zweifelhaft angesehen werden. Weitere Beispiele hierzu kann man im Anhang in den Abbildungen A10 und A11 finden.

Populationsspezifische Varianten sind ein wiederkehrendes Störsignal bei der Auswertung und Interpretation von genomischen Daten (Ameur u. a., 2018). Aus diesem Grund wird zum Beispiel bei der Entwurfsplanung einer GWAS-Studie versucht einen SNP-Chip zu wählen der, unter entsprechenden finanziellen Aspekten, möglichst gut den genomischen Hintergrund der Studie widerspiegelt (Ha, Freytag und Bickeboeller, 2014). Von Affymetrix werden hierfür populations-optimierte Arrays für die kaukasische, afrikanische und asiatische Population angeboten[3]. Im Abschnitt 3.5.3 habe ich gezeigt, daß *alt loci* als Polymorphismen insbesondere auch spezifisch für bestimmte Populationen auftreten (siehe z.B. Tabelle 3.3) und damit auch die dazugehörigen ASDP-assoziierten Varianten. Zur akkuraten Vorhersage von Varianten benötigt man genug genomische Information, um deren genaue Position zu lokalisieren. Oft ist diese Zuordnung mit Varianten-Calling Tech-

[3] Affymetrix Axiom™Genotyping Solution von Thermo Fischer Scientific

nologien, wie den SNP-Chips nicht möglich. Diese basiert auf der Erkennung von SNPs anhand von sequenzspezifischen Oligonukleotiden, entsprechend einer 50 bp langen genomischen Sequenz um einen bekannten SNP herum (siehe auch Abbildung 3.10). Von den 437 GWAS-Hits, welche mit ASDPs überlappen, haben 137 in dieser Sequenz nur die eine ASDP-assoziierte Variante. Dies wirft die Frage auf, ob diese Varianten nicht eher auf einen alternativen Haplotypen hinweisen statt auf einen Polymorphismus, der die Region auf dem Primärassembly markiert. Es ist bekannt, dass die meisten GWAS-Hits nicht selber kausativ sind, jedoch die Region des Haplotypen mit der einen oder den mehreren kausativen Varianten markieren. In dem genannten Fall sollte man bei der weiteren Analyse die Varianten auf dem *alt locus* betrachten, der mit den ASDP-assoziierten GWAS-Hits verbunden ist. Dies kann die Anzahl der fraglichen Varianten beträchtlich reduzieren (siehe Abbildung 3.2). Mit der Haplotypvorhersage von ASDPex läßt sich dies theoretisch auf alle 178 Regionen mit *alt loci* ausweiten. Affymetrix bietet nur die Unterscheidung für die drei Populationen, wohingegen hiermit eine viel feingranuliertere Auflösung für populationspezifische Regionen möglich ist. Ebenfalls gibt es deutlich mehr Populationen, die sich hiermit darstellen lassen und es ist möglich die spezifischen Regionen beliebig zu kombinieren.

Die Feststellung, dass eine einzelne Variante mit einem ASDP kolokalisiert ist, ist an sich kein Hinweis darauf, dass die Variante fälschlicherweise gecalled wurde oder eine falsch-positive Variante ist. In der Tat deuten die Analysen der ASDPs darauf hin, dass es im Primärassembly Polymorphismen gibt, deren alternative Allele Teilsequenz in einem *alt locus* entsprechen. ASDPex besitzt hierfür einen Schwellenwert `RV`, da zahlreiche ASDP-assoziierte Varianten in den Sequenzen der Primärassemblies gefunden wurden, diese jedoch nicht ausreichende Indizien für den ALT-HAP ergaben.

Daraus ergibt sich eine Einschränkung der gezeigten Studie. Es wurde nicht versucht, die Häufigkeit der Rekombination zwischen REF-HAP und ALT-HAP in der Bevölkerung zu analysieren. Die Rekombination zwischen den

verschiedenen Haplotypen in einer strukturell variablen genomischen Region kann ein Grund dafür sein, dass ASDP-assoziierte Varianten in Haplotypen gefunden werden können, die als REF-HAP vorhergesagt werden. Der implementierte Algorithmus basiert auf der vereinfachenden Annahme, dass die *alt loci* komplette Haplotypblöcke darstellen. Das Kopplungsungleichgewicht (Linkage disequilibrium) für Europäer liegt im Durchschnitt bei ∼60Kb und ist für die afrikanische Populationen noch geringer (Reich u. a., 2001). Die Länge der meisten Regionen mit einem *alt locus* ist häufig größer (siehe Tabelle 3.1), als diese Durchschnittslänge. Daher ist es wahrscheinlich, dass die *alt loci* als kompletter Block nicht in jedem Fall gültige Haplotypen sind. Dies sollte sich insbesondere bei Mischungen von Populationen beobachtet lassen. Isoliertere Populationen (z.B. PEL) können sich diese Eigenschaft eher erhalten (siehe auch Tabelle 3.3). Die Häufigkeit von Rekombinationsereignissen zwischen den Regionen auf dem Primärassembly und den entsprechenden *alt loci* sollte noch im Detail untersucht werden. Eine weitere Einschränkung der momentanen Implementierung der ASDPex Heuristik ist, dass sie nur SNPs und kleine InDels für die Berechnung von `RV` betrachtet. Die Bestimmung von ASDPs und den Varianten ist wiederum von der Genauigkeit und Qualität des Alignments, sowohl der NGS-Reads auf die Referenz, als auch zwischen Primärassembly und *alt loci*, abhängig.

4.3 Ausblick

In den vorangegangen Kapiteln dieser Arbeit habe ich gezeigt, wo momentan die Schwierigkeiten in der Auswertung und Interpretation von genomischen Daten im Allgemeinen und insbesondere im aktuellen Genomrelease liegen. Zum einen ist es schwer, Programme zu entwerfen, die möglichst modular und variable an die entsprechenden Bedürfnisse angepasst sind. Sie müssen dabei trotzdem eine konsistente Aussage und Ausgabe haben, wie *Jannovar* mit der Verwendung von Sequence Ontology und der HGVS-Nomenklatur. Auf der anderen Seite wurde gezeigt, wie man die Varianten

auch in den Kontext des aktuellen Genomreleases stellt und die gegebenen Informationen über die *alt loci* nutzt, um aus den Varianten Rückschlüsse auf die tatsächlichen Haplotypen zu schließen und damit Varianten auszufiltern. Obwohl der aktuelle Genomrelease GRCh38 schon seit fünf Jahren veröffentlicht ist, sind viele biologische und klinische Datenquellen noch nicht darauf angepasst und können dadurch nicht für medizinischen Interpretationspipelines verwendet werden. Zahlreiche Variantendatenbanken haben ihre Daten noch nicht auf GRCh38 adaptiert und es werden sogar immer noch klinisch relevante neue Daten(banken) veröffentlicht, die nicht für GRCh38, sondern nur den vorherigen Release GRCh37 verfügbar sind. Kürzlich wurde mit gnomAD-SV (R. L. Collins u. a., 2019) eine Datenbank für Strukturvarianten, basierend auf 15 000 individuellen Genomen, nur für GRCh37 veröffentlicht.

Teilweise wurde versucht, die Datenbanken mit einem *liftover*, also der Neuzuordung der bekannten Varianten auf das Koordinatensystem von GRCh38, zugänglich zu machen. Dies steht jedoch häufig mit den tausenden korrigierten Basen im Konflikt und kann im Umgang mit den Programmen einer Auswertepipeline sogar Probleme verursachen.

Ebenso habe ich die Schwierigkeiten aufgezeigt, die sich beim Calling von Varianten mit dem GRCh38 Referenzgenom Modell ergeben. Aktuelle Pipelines und Programme versuchen nicht, Variantencallings in Regionen des Genoms, die mit *alt loci* assoziiert sind, getrennt zu betrachten.

Das in dieser Arbeit vorgestellte Projekt ASDPex deutet auf das große Potenzial hin, die Ressourcen von graphenähnlichen Genomregionen vollständig in das Variantencalling zu integrieren. Die gezeigten Ergebnisse unterstreichen die Bedeutung der Entwicklung von Algorithmen, die eine vollständige Rekonstruktion des Genotyps in individuellen Genomen ermöglichen. Es ist vorstellbar, dass hierfür ein Variantencalling erforderlich ist, um daraus wiederum abzuleiten, welche Haplotypen (REF-HAP oder ALT-HAP) vorhanden sind. Im Anschluß können die Varianten dann entsprechend der Haplotypen ausgegeben werden. Die Community müsste sich dafür auf die

bestmögliche Darstellung dieser Ergebnisse im VCF-Format einigen. Letztendlich können neue Modelle zur Darstellung der Variation im Genom erforderlich sein (Zerbino u. a., 2013).

Der genaue Aufbau des Genoms ist für das Verständnis der genetischen Variation unerlässlich (Chaisson, Wilson und Eichler, 2015). Die GRCh38-Genom-Assemblierung war ein wichtiger Schritt bei der Entwicklung eines Modells, das die strukturellen Unterschiede in der menschlichen Population angemessen darstellen kann (Computational Pan-Genomics Consortium, 2018; Zerbino u. a., 2013). Es ist jedoch wahrscheinlich, dass es eine wesentlich höhere Anzahl von Regionen im menschlichen Genom gibt, die einen Grad an struktureller Variabilität aufweisen, der durch ein lineares Genommodell nicht ausreichend repräsentiert werden kann.

Die aktuellen Fortschritte der NGS-Technologien mit längeren Reads und schnellere, präzisere Algorithmen werden eine zunehmende Anzahl von variablen genomischen Regionen bestimmen können, die in zukünftige Zusammensetzungen des menschlichen Genoms integriert werden können (Ameur u. a., 2018; Berlin u. a., 2015; Chaisson, Wilson und Eichler, 2015; Huddleston u. a., 2014; Jain u. a., 2018; Shi u. a., 2016; Steinberg u. a., 2014; Sudmant u. a., 2015). Da unser Wissen und Verständnis über das menschliche Genom und seine Variation in der Population zunimmt, erscheint es wahrscheinlich, dass komplexere graphenbasierte Darstellungen des Genoms nützlich werden.

Zukünftig wird eine graphenbasierte alternative Darstellung des humanen Genomassemblies die populationsspezifischen Eigenarten auf eine intuitivere Weise darstellen (Church, Valerie A Schneider, Steinberg u. a., 2015; Computational Pan-Genomics Consortium, 2018). Dies kann nur mit einer Anpassung der existierenden Programme, (Datei-)Standards und Analysepipelines erfolgen und wird einigen Aufwand und Zeit in Anspruch nehmen.

Es gibt verschiedenen Ansätze die ursprünglich für lineare, haploide Referenzen entwickelten Mapper auf Graphengenome anzupassen. Bei den Vergleichen stach BWA-MEM hervor, was ein weiterer Grund war, dieses

Programm für die gezeigte Analyse zu verwenden. Zusätzlich zur Adaption der existierenden Programme, werden auch neue graphenbasierten Alignment Tools entwickelt (Duan u. a., 2019; Paten u. a., 2017). Im Jahre 2016 gründete sich zusätzlich das Pan-Genom Consortium, um ein humanes Pan-Genom zu erstellen, was möglichst viele populationsspezifische Variationen abbildet (Computational Pan-Genomics Consortium, 2018). Rakocevic u. a. (2019) veröffentlichten vor Kurzem eine effiziente Graphengenomrepresentation, welche 2 800 diploide Genome inkludiert und damit 12,6 Mio. SNPs und 4 Mio. InDels. Sie konnten zeigen, dass damit eine höhere Sensitivität beim Mapping der Reads erreicht werden, die Bestimmung der Varianten verbessert werden und Strukturvarianten akkurat unter diesem Einheitsmodell genotypisiert werden können.
Das GRC hat angekündigt vorerst nur Patches für das momentane Genomrelease zu veröffentlichen, solange es nicht geklärt ist, wie man die Variabilität am besten repräsentiert.

> We will continue to make these updates publicly available at regular intervals in the form of patch releases, but have decided to indefinitely postpone our next coordinate-changing update (GRCh39) while we evaluate new models and sequence content for the human reference assembly currently in development.[4]

Vorerst ist man auf die eingeschränkte Repräsentation von variablen Regionen in GRCh38 angewiesen. Alternativ wird nach weiteren Möglichkeiten geschaut, um den Populationsbias im humanen Genom gerecht zu werden. Hierbei wird auch die paternale und maternale (geographische) Abstammung des Patienten in Betracht gezogen. Eventuell muss dann bei einer Kombination der populationsspezifische Referenzgraph angepasst werden. Momentan wird der Populationsbias häufig durch lokale Genomprojekte kompensiert (Personal Genome Project: China - eine Studie mit mehr als 140 000 Individuen; Dänemark – 150 gesunde Probanden[5]; genome Map of

[4] https://www.ncbi.nlm.nih.gov/grc
[5] http://www.genomedenmark.dk/english/about/referencegenome/

poland – 5 000 Personen[6]; Genom Austria – 1 000 Personen[7]; Indigenous Australian reference genome[8]; Japanese reference genome – 3 500 Personen[9] (Nagasaki u. a., 2019) und viele weitere).

Ich würde für einen Wechsel auf GRCh38 aus mehreren Gründen plädieren. Die Darstellung ist repräsentativer und vollständiger. Neben der Korrektur von einzelnen Basen, wurden die Bereiche um die Centromere vervollständigt und es gab zahlreiche Relokationen von kompletten Sequenzabschnitten. Pan u. a. (2019) konnte zeigen, dass 5% der Varianten, welche in GRCh38 identifiziert wurden, nicht in GRCh37 platziert werden konnten. Dies liegt unter anderem an den 3,6 Megabasen an neuen zusätzlichen Sequenzen in den Primärassemblies, den *alt loci* und Scaffolds. Zusätzlich konnte unter anderem durch die Arbeit von Ameur u. a. (2018) gezeigt werden, dass lokale Genomprojekte fehlende Sequenzen in der GRCh38 Referenz einbinden und im Populationskontext deutlich genauere und spezifischere Variantenbestimmungen erlauben. Für eine aktuelle Analyse sollte diese verfügbare Informationen genutzt werden.

[6] http://www.ecbig.pl/page/genomic-map-of-poland/
[7] https://genomaustria.at/
[8] http://ncig.anu.edu.au/
[9] https://jrg.megabank.tohoku.ac.jp/en/

Zusammenfassung

Das Gesamtvolumen genomischer Sequenzierungsdaten nimmt, dank der Entwicklung der DNA-Hochdurchsatz-Sequenziertechniken, in den letzten Jahren in unglaublichem Tempo zu. Dies erweitert unser Wissen des humanen Genoms über dessen Aufbau, die Struktur und die räumliche Organisation. Die Erkenntnis um den komplexen Aufbau wird in naher Zukunft in die Interpretation von Variationen auch im klinischen Kontext einfließen müssen, birgt sie doch zahlreiche potentielle Möglichkeiten in der Diagnostik. Whole Genome Sequencing hat schon jetzt den Sprung aus den Forschungslaboren in die angewandte Diagnostik von Krankenhäusern geschafft und erlaubt damit die Einführung der Präzisionsmedizin für alle Patienten. Für eine optimale klinische Interpretation genomischer Varianten ist es wichtig, konsistente und passende Referenzen zu verwenden. Hierzu zählt neben der Auswahl des Referenzgenoms auch die verwendete Datenbank zur Annotierung von funktionalen Einheiten auf der DNA.
Diese Arbeit geht auf zwei wichtige Schritte auf dem Weg zum Einsatz des WGS im klinischen Alltag ein. Der erste Schritt beinhaltet, möglichst schnell die gefundenen Varianten zu genomischen Eigenschaften und Features (in Relation zu einer Referenz) zuzuordnen. Dies ist aufgrund der großen Datenmengen ein zunehmendes Problem geworden. Mit Jannovar wird hier eine Softwarebibliothek vorgestellt, welche hervorragend an diese Ansprüche angepasst ist. Die Bibliothek ist schnell, flexibel und kann leicht in Annotationspipelines und eigene Programme integriert werden. Die so annotierten und charakterisierten Veränderungen des Genotyps bilden eine Basis für die weitere Interpretation und Beurteilung durch andere Programme.
Die Repräsentation der Genomreferenz entwickelt sich hin zu einem Graphengenom, um die populationsspezifische Variabilität zumindest ansatzweise abzubilden. Diese kann einen enormen Einfluss auf die Interpretation von Varianten haben. Im zweiten Schritt geht es darum, diese populationsspezifische Komplexität zu erläutern. Mit ASDPex wird ein heuristischer Algorithmus vorgestellt, welcher für WGS-Daten eines Individuums das Auftreten von alternativen Haplotypsequenzen vorhersagt. Dafür verwendet es die Verteilung der Allelfrequenzen der individuellen Varianten und gleicht sie mit einer Art Fingerabdruck aus haplotypspezifischen Varianten ab. Das Wissen um die alternativen Sequenzen kann die Verlässlichkeit der klinischen Interpretation weiter verbessern.
Zukünftig wird es darum gehen, noch mehr Daten in die Varianteninterpretation zu integrieren, um noch mehr falsch positive/falsch negative Assoziationen zu verhindern und irrelevante Varianten herauszufiltern.

Danksagung

Es gibt eine ganze Anzahl an Menschen, denen mein Dank gilt. Die schwierigste Aufgabe besteht nun darin sie jetzt hier irgendwie in eine Reihe zu bringen. Das kann natürlich nur ohne Gewichtung geschehen, da mich jeder auf seine Weise bei der Anfertigung dieser Arbeit unterstützt hat.

In erste Linie gilt meine Dankbarkeit Prof. Dr. med. Peter N. Robinson, der mich vor Äonen von Jahren in seiner Gruppe aufgenommen hatte und mir damit letztendlich diese Arbeit ermöglicht hat. Ich denke immer noch gerne an die Zeit in der Arbeitsgruppe zurück. Damit einher gilt mein Dank dem gesamten Institut für Medizinische Genetik und Humangenetik und den vielen Mitarbeitern, die mich dort all die Jahre begleitet haben, insbesondere meinen ehemaligen Kollegen aus der Computational Biology Gruppe.

Mein besonderer Dank gilt Sebastian und Maria für Ihre Unterstützung und antreibenden Worte während des Verfassens der Arbeit. Sie haben viel ihrer private Zeit und Leberkäse während der Korrekturen investiert.

Bedanken möchte ich mich auch bei Marie für die Gespräche, das motivierende gemeinsame Arbeiten und für die aufmunternden Worte.

Nicht zu vergessen die Marathonsitzung zur finalen Korrektur meiner Schwester Katja. Was man nicht alles für seine Geschwister tut ...

Die entscheidende Unterstützung bei der Anfertigung dieser Arbeit habe ich durch meine Familie erhalten: Meine Eltern, die immer für mich da und ein Zufluchtsort waren, wenn ich mal eine Verschnaufpause brauchte. Jule, die mir den Rücken freigehalten hat und auf die ich mich immer verlassen konnte und wusste, dass es den Kindern auch gut geht.

Zum Schluss danke ich meinen beiden Kindern Johanna und Julius, die immer ein Lichtblick waren, sind und sein werden. Sie können mich den Stress leicht vergessen lassen und werden jetzt hoffentlich wieder mehr von mir haben.

Abkürzungsverzeichnis

ASDP Alignable Scaffold-Discrepant Position

BIH Berlin Institute of Health

bp Basenpaar

bzw. beziehungsweise

CCDS Consensus Coding Sequence

cDNA codierende DNA

CDS codierende Sequenz

dbSNP Datenbank für Singlenukleotidpolymorphismen

DDBJ DNA Data Bank of Japan

ddNTP Didesoxynukleotid

d.h. das heißt

DNA Desoxyribonukleinsäure

dNTP Desoxynukleotid

EBV Epstein-Barr-Virus

FISH Fluoreszenz-in-situ-Hybridisierung

GFF3 General Feature Format Version 3

GRC Genome Reference Consortium

GRCh37 Genome Reference Consortium human build release 37

GRCh38 Genome Reference Consortium human build release 38

GWAS Genom weite Assoziationsstudie

HGP Humangenomprojekt

HGNC HUGO Gene Nomenclature Committee

HGVS Human Genome Variation Society

HLA Human Leukocyte Antigen

HTS High Throughput Sequencing

HUGO Human Genome Organization

HVP Human Variome Project

IHGSC International Human Genome Sequencing Consortium

InDel Kunstwort aus Insertion und Deletion

INSDC International Nucleotide Sequence Database Collaboration

MAF minor allele frequency

Mb 1 Mb = 1.000.000 Basen = 1 Megabase

Mio. Million

MHC Haupthistokompatibilitätskomplex

MNV Multinukleotidvariante

Mrd. Milliarde

mRNA Boten-RNA

NCBI National Center for Biotechnology Information

NHGRI National Human Genome Research Institute

NIH National Institute of Health

NGS Next Generation Sequencing

nt Nukleotid

ORF offener Leserahmen

PCR Polymerasekettenreaktion

Read abgeleitete Sequenz von Basenpaaren, welches einem DNA-Fragment entspricht

Read Alignment Abbildung von Reads auf eine Referenzsequenz

RefSeq NCBI Reference Sequence Database

RNA-Seq RNA-Sequenzierung

RNA Ribonukleinsäure

SAM Sequence Alignment/Map

SBS Sequencing By Synthesis

SNV Einzelnukleotidvariation

SNP Einzelnukleotidpolymorphismus

SO Sequence Ontology

ssDNA einzelsträngige DNA

SV Strukturvariante

UCSC University of California Santa Cruz

USD US-Dollar

UTR untranslatierter Bereich

VCF Variant Calling Format

WES Whole Exome Sequencing

WGS Whole Genome Sequencing

z.B. zum Beispiel

Literatur

1000 Genomes Project Consortium u. a. (2010).
„A map of human genome variation from population-scale sequencing.“ eng. In: *Nature* 467.7319, S. 1061–1073.

Aken, Bronwen L., Premanand Achuthan u. a. (2017).
„Ensembl 2017.“ In: *Nucleic acids research* 45 (D1), S. D635–D642. ISSN: 1362-4962.

Aken, Bronwen L., Sarah Ayling u. a. (2016).
„The Ensembl gene annotation system.“ In: *Database : the journal of biological databases and curation* 2016. ISSN: 1758-0463.

Altmann, Richard (1889).
„Ueber Nucleinsäuren.“ In: *Archiv für Anatomie und Physiologie. Physiologische Abteilung. Leipzig*, S. 524–536.

Ameur, Adam u. a. (2018).
„De Novo Assembly of Two Swedish Genomes Reveals Missing Segments from the Human GRCh38 Reference and Improves Variant Calling of Population-Scale Sequencing Data.“ In: *Genes* 9 (10). ISSN: 2073-4425.

Avery, O., C. MacLeod und M. McCarty (1944).
„Studies on the chemical nature of the substance inducing transformation of pneumococcal types. Inductions of transformation by a desoxyribonucleic acid fraction isolated from pneumococcus type III“. In: *J Exp Med* 79.2, S. 137–158.

Bamshad, Michael J. u. a. (2011).
„Exome sequencing as a tool for Mendelian disease gene discovery.“ eng. In: *Nat Rev Genet* 12.11, S. 745–755.

Beadle, G. W. und E. L. Tatum (1941).
„Genetic Control of Biochemical Reactions in Neurospora.“ In: *Proceedings of the National Academy of Sciences of the United States of America* 27 (11), S. 499–506. ISSN: 0027-8424.

Beadle, George Wells (1945).

„Biochemical genetics.“ In: *Chemical reviews* 37.1, S. 15–96.

Berg, Mark de u. a. (2008).
Computational Geometry: Algorithms and Applications. 3rd ed. Santa Clara, CA, USA: Springer-Verlag TELOS. ISBN: 9783540779735.

Berlin, Konstantin u. a. (2015).
„Assembling large genomes with single-molecule sequencing and locality-sensitive hashing.“ eng. In: *Nat Biotechnol* 33.6, S. 623–630.

Blencowe, Benjamin J. (2006).
„Alternative splicing: new insights from global analyses.“ In: *Cell* 126 (1), S. 37–47. ISSN: 0092-8674.

Boldogkői, Zsolt u. a. (2019).
„Long-Read Sequencing - A Powerful Tool in Viral Transcriptome Research.“ In: *Trends in microbiology* 27 (7), S. 578–592. ISSN: 1878-4380.

Brudno, Michael u. a. (2003).
„LAGAN and Multi-LAGAN: efficient tools for large-scale multiple alignment of genomic DNA.“ In: *Genome research* 13 (4), S. 721–731. ISSN: 1088-9051.

Chaisson, Mark J P., Richard K. Wilson und Evan E. Eichler (2015).
„Genetic variation and the de novo assembly of human genomes.“ eng. In: *Nat Rev Genet* 16.11, S. 627–640.

Choi, Byung-Ok u. a. (2012).
„Exome sequencing is an efficient tool for genetic screening of Charcot-Marie-Tooth disease.“ In: *Human mutation* 33 (11), S. 1610–1615. ISSN: 1098-1004.

Church, Deanna M, Valerie A Schneider, Tina Graves u. a. (2011).
„Modernizing reference genome assemblies.“ In: *PLoS biology* 9 (7), e1001091. ISSN: 1545-7885.

Church, Deanna M, Valerie A Schneider, Karyn Meltz Steinberg u. a. (2015).
„Extending reference assembly models.“ In: *Genome biology* 16, S. 13. ISSN: 1474-760X.

Cingolani, Pablo u. a. (2012).
„Using Drosophila melanogaster as a Model for Genotoxic Chemical Mutational Studies with a New Program, SnpSift.“ In: *Frontiers in genetics* 3, S. 35. ISSN: 1664-8021.

Collins, Ryan L u. a. (2019).
„An open resource of structural variation for medical and population genetics“. In: *BioRxiv*, S. 578674.

Computational Pan-Genomics Consortium (2018).

„Computational pan-genomics: status, promises and challenges.“ In: *Briefings in bioinformatics* 19 (1), S. 118–135. ISSN: 1477-4054.

Danecek, Petr u. a. (2011).
„The variant call format and VCFtools.“ In: *Bioinformatics (Oxford, England)* 27 (15), S. 2156–2158. ISSN: 1367-4811.

Döring, Andreas u. a. (2008).
„SeqAn an efficient, generic C++ library for sequence analysis.“ eng. In: *BMC Bioinformatics* 9, S. 11.

Drmanac, Radoje u. a. (2010).
„Human genome sequencing using unchained base reads on self-assembling DNA nanoarrays.“ In: *Science (New York, N.Y.)* 327 (5961), S. 78–81. ISSN: 1095-9203.

Duan, Zhongqu u. a. (2019).
„HUPAN: a pan-genome analysis pipeline for human genomes.“ In: *Genome biology* 20 (1), S. 149. ISSN: 1474-760X.

Dudley, Joel T. und Konrad J. Karczewski (2013).
Exploring Personal Genomics. Oxford University Press.

Dunnen, Johan T. den (2017).
„Describing Sequence Variants Using HGVS Nomenclature.“ In: *Methods in molecular biology (Clifton, N.J.)* 1492, S. 243–251. ISSN: 1940-6029.

Eilbeck, Karen und Suzanna E. Lewis (2004).
„Sequence ontology annotation guide.“ eng. In: *Comp Funct Genomics* 5.8, S. 642–647.

Faust, Gregory G. und Ira M. Hall (2014).
„SAMBLASTER: fast duplicate marking and structural variant read extraction.“ eng. In: *Bioinformatics* 30.17, S. 2503–2505.

Fernandez-Marmiesse, Ana, Sofia Gouveia und Maria L. Couce (2018).
„NGS Technologies as a Turning Point in Rare Disease Research , Diagnosis and Treatment.“ In: *Current medicinal chemistry* 25 (3), S. 404–432. ISSN: 1875-533X.

Feuk, Lars, Andrew R. Carson und Stephen W. Scherer (2006).
„Structural variation in the human genome.“ In: *Nature reviews. Genetics* 7 (2), S. 85–97. ISSN: 1471-0056.

Flicek, Paul u. a. (2013).
„Ensembl 2013.“ eng. In: *Nucleic Acids Res* 41.Database issue, S. D48–D55.

Forbes, Simon A. u. a. (2017).

„COSMIC: somatic cancer genetics at high-resolution.“ In: *Nucleic acids research* 45 (D1), S. D777–D783. ISSN: 1362-4962.

Frankish, Adam u. a. (2015).
„Comparison of GENCODE and RefSeq gene annotation and the impact of reference geneset on variant effect prediction.“ In: *BMC genomics* 16 Suppl 8, S2. ISSN: 1471-2164.

Garrison, Erik und Gabor Marth (2012).
„Haplotype-based variant detection from short-read sequencing“. In: *ArXiv* 1207.3907.

Gilfillan, Gregor D. u. a. (2012).
„Limitations and possibilities of low cell number ChIP-seq.“ In: *BMC genomics* 13, S. 645. ISSN: 1471-2164.

Gray, Kristian A. u. a. (2015).
„Genenames.org: the HGNC resources in 2015.“ In: *Nucleic acids research* 43 (Database issue), S. D1079–D1085. ISSN: 1362-4962.

Guttmacher, Alan E. und Francis S. Collins (2002).
„Genomic medicine–a primer.“ In: *The New England journal of medicine* 347 (19), S. 1512–1520. ISSN: 1533-4406.

Ha, Ngoc-Thuy, Saskia Freytag und Heike Bickeboeller (2014).
„Coverage and efficiency in current SNP chips.“ In: *European journal of human genetics : EJHG* 22 (9), S. 1124–1130. ISSN: 1476-5438.

Habegger, Lukas u. a. (2012).
„VAT: a computational framework to functionally annotate variants in personal genomes within a cloud-computing environment.“ eng. In: *Bioinformatics* 28.17, S. 2267–2269.

Harrow, Jennifer L. u. a. (2014).
„The Vertebrate Genome Annotation browser 10 years on.“ In: *Nucleic acids research* 42 (Database issue), S. D771–D779. ISSN: 1362-4962.

Harrow, Jennifer, France Denoeud u. a. (2006).
„GENCODE: producing a reference annotation for ENCODE.“ In: *Genome biology* 7 Suppl 1, S4.1–S4.9. ISSN: 1474-760X.

Harrow, Jennifer, Adam Frankish u. a. (2012).
„GENCODE: the reference human genome annotation for The ENCODE Project.“ In: *Genome research* 22 (9), S. 1760–1774. ISSN: 1549-5469.

Hinrichs, A. S. u. a. (2006).
„The UCSC Genome Browser Database: update 2006.“ In: *Nucleic acids research* 34 (Database issue), S. D590–D598. ISSN: 1362-4962.

Hirschberg, Daniel S. (1975).

„A linear space algorithm for computing maximal common subsequences“. In: *Communications of the ACM* 18.6, S. 341–343.

Hsi-Yang Fritz, Markus u. a. (2011).
„Efficient storage of high throughput DNA sequencing data using reference-based compression.“ In: *Genome research* 21 (5), S. 734–740. ISSN: 1549-5469.

Hsu, Fan u. a. (2006).
„The UCSC Known Genes.“ eng. In: *Bioinformatics* 22.9, S. 1036–1046.

Huddleston, John u. a. (2014).
„Reconstructing complex regions of genomes using long-read sequencing technology.“ eng. In: *Genome Res* 24.4, S. 688–696.

Jäger, Marten u. a. (2014).
„Jannovar: a java library for exome annotation.“ In: *Human mutation* 35 (5), S. 548–555. ISSN: 1098-1004.

Jain, Miten u. a. (2018).
„Nanopore sequencing and assembly of a human genome with ultra-long reads.“ In: *Nature biotechnology* 36 (4), S. 338–345. ISSN: 1546-1696.

Karolchik, Donna u. a. (2014).
„The UCSC Genome Browser database: 2014 update.“ In: *Nucleic acids research* 42 (Database issue), S. D764–D770. ISSN: 1362-4962.

Kent, W. J. und D. Haussler (2001).
„Assembly of the working draft of the human genome with GigAssembler.“ eng. In: *Genome Res* 11.9, S. 1541–1548.

Knierim, Ellen u. a. (2011).
„Systematic comparison of three methods for fragmentation of long-range PCR products for next generation sequencing.“ In: *PloS one* 6 (11), e28240. ISSN: 1932-6203.

Krawitz, Peter u. a. (2010).
„Microindel detection in short-read sequence data.“ In: *Bioinformatics (Oxford, England)* 26 (6), S. 722–729. ISSN: 1367-4811.

Lander, E S u. a. (2001).
„Initial sequencing and analysis of the human genome.“ In: *Nature* 409 (6822), S. 860–921. ISSN: 0028-0836.

Langmead, Ben u. a. (2009).
„Ultrafast and memory-efficient alignment of short DNA sequences to the human genome.“ In: *Genome biology* 10 (3), R25. ISSN: 1474-760X.

Laurie, Steve u. a. (2016).

„From Wet-Lab to Variations: Concordance and Speed of Bioinformatics Pipelines for Whole Genome and Whole Exome Sequencing.“ In: *Human mutation* 37 (12), S. 1263–1271. ISSN: 1098-1004.

Levy, Samuel u. a. (2007).
„The diploid genome sequence of an individual human.“ In: *PLoS biology* 5 (10), e254. ISSN: 1545-7885.

Li, Heng (2013).
„Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM“. In: *arXiv*.

Li, Heng (2011).
„A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data.“ In: *Bioinformatics (Oxford, England)* 27 (21), S. 2987–2993. ISSN: 1367-4811.

Li, Heng, Bob Handsaker u. a. (2009).
„The Sequence Alignment/Map format and SAMtools.“ In: *Bioinformatics (Oxford, England)* 25 (16), S. 2078–2079. ISSN: 1367-4811.

Li, Heng, Jue Ruan und Richard Durbin (2008).
„Mapping short DNA sequencing reads and calling variants using mapping quality scores.“ In: *Genome research* 18 (11), S. 1851–1858. ISSN: 1088-9051.

Li, Ruiqiang, Yingrui Li, Karsten Kristiansen u. a. (2008).
„SOAP: short oligonucleotide alignment program.“ In: *Bioinformatics (Oxford, England)* 24 (5), S. 713–714. ISSN: 1367-4811.

Li, Ruiqiang, Yingrui Li, Hancheng Zheng u. a. (2010).
„Building the sequence map of the human pan-genome.“ In: *Nature biotechnology* 28 (1), S. 57–63. ISSN: 1546-1696.

Li, Ruiqiang, Chang Yu u. a. (2009).
„SOAP2: an improved ultrafast tool for short read alignment.“ In: *Bioinformatics (Oxford, England)* 25 (15), S. 1966–1967. ISSN: 1367-4811.

Makarov, Vladimir u. a. (2012).
„AnnTools: a comprehensive and versatile annotation toolkit for genomic variants.“ In: *Bioinformatics (Oxford, England)* 28 (5), S. 724–725. ISSN: 1367-4811.

Margulies, Marcel u. a. (2005).
„Genome sequencing in microfabricated high-density picolitre reactors.“ In: *Nature* 437 (7057), S. 376–380. ISSN: 1476-4687.

McKenna, Aaron u. a. (2010).

„The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data.“ eng. In: *Genome Res* 20.9, S. 1297–1303.

McLaren, William u. a. (2010).
„Deriving the consequences of genomic variants with the Ensembl API and SNP Effect Predictor.“ eng. In: *Bioinformatics* 26.16, S. 2069–2070.

Mendel, G. J. (1866).
„Versuche über Pflanzenhybriden“. In: *Verhandlungen des Naturforschenden Vereines in Brünn* 4, S. 3–47.

Miga, Karen H. u. a. (2014).
„Centromere reference models for human chromosomes X and Y satellite arrays.“ eng. In: *Genome Res* 24.4, S. 697–707.

Nagasaki, Masao u. a. (2019).
„Construction of JRG (Japanese reference genome) with single-molecule real-time sequencing.“ In: *Human genome variation* 6, S. 27. ISSN: 2054-345X.

NCBI Resource Coordinators (2016).
„Database resources of the National Center for Biotechnology Information.“ eng. In: *Nucleic Acids Res* 44.D1, S. D7–D19.

Needleman, S. B. und C. D. Wunsch (1970).
„A general method applicable to the search for similarities in the amino acid sequence of two proteins.“ In: *Journal of molecular biology* 48 (3), S. 443–453. ISSN: 0022-2836.

Ng, Sarah B. u. a. (2010).
„Exome sequencing identifies the cause of a mendelian disorder.“ eng. In: *Nat Genet* 42.1, S. 30–35.

Okada, Yukinori u. a. (2012).
„Meta-analysis identifies multiple loci associated with kidney function-related traits in east Asian populations.“ In: *Nature genetics* 44 (8), S. 904–909. ISSN: 1546-1718.

Pan, Bohu u. a. (2019).
„Similarities and differences between variants called with human reference genome HG19 or HG38.“ In: *BMC bioinformatics* 20 (Suppl 2), S. 101. ISSN: 1471-2105.

Pandey, Janardan P. (2010).
„Genomewide association studies and assessment of risk of disease.“ In: *The New England journal of medicine* 363 (21), 2076–7, author reply 2077. ISSN: 1533-4406.

Parkinson, Nicholas J. u. a. (2012).
„Preparation of high-quality next-generation sequencing libraries from picogram quantities of target DNA.“ In: *Genome research* 22 (1), S. 125–133. ISSN: 1549-5469.

Paten, Benedict u. a. (2017).
„Genome graphs and the evolution of genome inference.“ In: *Genome research* 27 (5), S. 665–676. ISSN: 1549-5469.

Pruitt, Kim D. u. a. (2009).
„The consensus coding sequence (CCDS) project: Identifying a common protein-coding gene set for the human and mouse genomes.“ In: *Genome research* 19 (7), S. 1316–1323. ISSN: 1088-9051.

Rabbani, Bahareh u. a. (2012).
„Next-generation sequencing: impact of exome sequencing in characterizing Mendelian disorders.“ eng. In: *J Hum Genet* 57.10, S. 621–632.

Rakocevic, Goran u. a. (2019).
„Fast and accurate genomic analyses using genome graphs.“ In: *Nature genetics* 51 (2), S. 354–362. ISSN: 1546-1718.

Reich, D. E. u. a. (2001).
„Linkage disequilibrium in the human genome.“ eng. In: *Nature* 411.6834, S. 199–204.

Robinson, James T. u. a. (2011).
„Integrative genomics viewer.“ eng. In: *Nat Biotechnol* 29.1, S. 24–26.

Robinson, Peter N., Peter Krawitz und Stefan Mundlos (2011).
„Strategies for exome and genome sequence data analysis in disease-gene discovery projects.“ In: *Clinical genetics* 80 (2), S. 127–132. ISSN: 1399-0004.

Robinson, Peter N., Rosario M.Piro und Marten Jäger (2018).
Computational Exome and Genome Analysis. Taylor und Francis Inc. ISBN: 978-1-4987-7598-4.

Sambrook, Joseph und David W. Russell (2006).
„Fragmentation of DNA by nebulization.“ In: *CSH protocols* 2006 (4).

Sandmann, Sarah u. a. (2017).
„Evaluating Variant Calling Tools for Non-Matched Next-Generation Sequencing Data.“ In: *Scientific reports* 7, S. 43169. ISSN: 2045-2322.

Sanger, F. u. a. (1977).
„Nucleotide sequence of bacteriophage phi X174 DNA.“ In: *Nature* 265 (5596), S. 687–695. ISSN: 0028-0836.

Sanger, F und A R Coulson (1975).

„A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase.“ In: *Journal of molecular biology* 94 (3), S. 441–448. ISSN: 0022-2836.

Schloss, Jeffery A. (2008).
„How to get genomes at one ten-thousandth the cost.“ In: *Nature biotechnology* 26 (10), S. 1113–1115. ISSN: 1546-1696.

Schneider, Valerie A. u. a. (2017).
„Evaluation of GRCh38 and de novo haploid genome assemblies demonstrates the enduring quality of the reference assembly.“ In: *Genome research* 27 (5), S. 849–864. ISSN: 1549-5469.

Shendure, Jay (2011).
„Next-generation human genetics.“ eng. In: *Genome Biol* 12.9, S. 408.

Shendure, Jay u. a. (2005).
„Accurate multiplex polony sequencing of an evolved bacterial genome.“ eng. In: *Science* 309.5741, S. 1728–1732.

Sherry, S. T. u. a. (2001).
„dbSNP: the NCBI database of genetic variation.“ In: *Nucleic acids research* 29 (1), S. 308–311. ISSN: 1362-4962.

Shi, Lingling u. a. (2016).
„Long-read sequencing and de novo assembly of a Chinese genome.“ eng. In: *Nat Commun* 7, S. 12065.

Smedley, Damian, Julius O. B. Jacobsen u. a. (2015).
„Next-generation diagnostics and disease-gene discovery with the Exomiser.“ In: *Nature protocols* 10 (12), S. 2004–2015. ISSN: 1750-2799.

Smedley, Damian, Max Schubach u. a. (2016).
„A Whole-Genome Analysis Framework for Effective Identification of Pathogenic Regulatory Variants in Mendelian Disease.“ In: *American journal of human genetics* 99 (3), S. 595–606. ISSN: 1537-6605.

Smigielski, E. M. u. a. (2000).
„dbSNP: a database of single nucleotide polymorphisms.“ In: *Nucleic acids research* 28 (1), S. 352–355. ISSN: 0305-1048.

Splinter, Kimberly u. a. (2018).
„Effect of Genetic Diagnosis on Patients with Previously Undiagnosed Disease.“ In: *The New England journal of medicine* 379 (22), S. 2131–2139. ISSN: 1533-4406.

Stark, Zornitza u. a. (2017).
„Prospective comparison of the cost-effectiveness of clinical whole-exome sequencing with that of usual care overwhelmingly supports early use

and reimbursement.“ In: *Genetics in medicine : official journal of the American College of Medical Genetics* 19 (8), S. 867–874. ISSN: 1530-0366.

Steinberg, Karyn Meltz u. a. (2014).
„Single haplotype assembly of the human genome from a hydatidiform mole.“ eng. In: *Genome Res* 24.12, S. 2066–2076.

Sudmant, Peter H u. a. (2015).
„An integrated map of structural variation in 2,504 human genomes.“ In: *Nature* 526 (7571), S. 75–81. ISSN: 1476-4687.

Sullivan, Alexis P. u. a. (2017).
„An evolutionary medicine perspective on Neandertal extinction.“ In: *Journal of human evolution* 108, S. 62–71. ISSN: 1095-8606.

Tan, Adrian, Goncalo R. Abecasis und Hyun Min Kang (2015).
„Unified representation of genetic variants.“ In: *Bioinformatics (Oxford, England)* 31 (13), S. 2202–2204. ISSN: 1367-4811.

Tatum, E. L. und G. W. Beadle (1945).
„Biochemical genetics of Neurospora“. In: *Annals of the Missouri Botanical Garden* 32.2, S. 125–129.

Taylor, Jenny C. u. a. (2015).
„Factors influencing success of clinical genome sequencing across a broad spectrum of disorders.“ eng. In: *Nat Genet* 47.7, S. 717–726.

Thorvaldsdóttir, Helga, James T. Robinson und Jill P. Mesirov (2013).
„Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration.“ In: *Briefings in bioinformatics* 14 (2), S. 178–192. ISSN: 1477-4054.

Tyner, Cath u. a. (2017).
„The UCSC Genome Browser database: 2017 update.“ In: *Nucleic acids research* 45 (D1), S. D626–D634. ISSN: 1362-4962.

Veltman, Joris A. und James R. Lupski (2015).
„From genes to genomes in the clinic.“ eng. In: *Genome Med* 7.1, S. 78.

Venter, J. C. u. a. (2001).
„The sequence of the human genome.“ eng. In: *Science* 291.5507, S. 1304–1351.

Wagner, Josef u. a. (2016).
„Evaluation of PacBio sequencing for full-length bacterial 16S rRNA gene classification.“ In: *BMC microbiology* 16 (1), S. 274. ISSN: 1471-2180.

Wang, Kai, Mingyao Li und Hakon Hakonarson (2010).

„ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data.“ eng. In: *Nucleic Acids Res* 38.16, e164.

Warden, Charles D. u. a. (2014).
„Detailed comparison of two popular variant calling packages for exome and targeted exon studies.“ In: *PeerJ* 2, e600.

Watson H. R., J. D. und F. H. Crick (1953).
„Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid.“ eng. In: *Nature* 171.4356, S. 737–738.

Watson, James (2012).
The double helix. Hachette UK.

Weese, David u. a. (2009).
„RazerS–fast read mapping with sensitivity control.“ In: *Genome research* 19 (9), S. 1646–1654. ISSN: 1549-5469.

Welter, Danielle u. a. (2014).
„The NHGRI GWAS Catalog, a curated resource of SNP-trait associations.“ eng. In: *Nucleic Acids Res* 42.Database issue, S. D1001–D1006.

Wildeman, Martin u. a. (2008).
„Improving sequence variant descriptions in mutation databases and literature using the Mutalyzer sequence variation nomenclature checker.“ In: *Human mutation* 29 (1), S. 6–13. ISSN: 1098-1004.

Wolfe, Daniel u. a. (2013).
„Visualizing genomic information across chromosomes with PhenoGram.“ eng. In: *BioData Min* 6.1, S. 18.

Yang, Rendong u. a. (2015).
„ScanIndel: a hybrid framework for indel detection via gapped alignment, split reads and de novo assembly.“ In: *Genome medicine* 7, S. 127. ISSN: 1756-994X.

Yang, Xiaofei u. a. (2019).
„One reference genome is not enough.“ In: *Genome biology* 20 (1), S. 104. ISSN: 1474-760X.

Zemojtel, Tomasz u. a. (2014).
„Effective diagnosis of genetic disease by computational phenotype analysis of the disease-associated genome.“ In: *Science translational medicine* 6 (252), 252ra123. ISSN: 1946-6242.

Zerbino, D. R. u. a. (2013).
„Representing and decomposing genomic structural variants as balanced integer flows on sequence graphs“. In: *ArXiv e-prints* 1303.5569.

Zheng-Bradley, Xiangqun u. a. (2017).

„Alignment of 1000 Genomes Project reads to reference assembly GRCh38.“ In: *GigaScience* 6 (7), S. 1–8. ISSN: 2047-217X.

Anhang

Tabelle A1: **Annotationsbeispiele**. Beispiele für die Annotation von Varianten für alle SO-Terme, welche *Jannovar* bekannt sind.

Begriff (SO)	**Beispiel**
start_lost	HRNR|NM_001009931.2|2/3|c.1del|p.0?
stop_gained	PRAMEF1|NM_023013.2|3/4|c.314T>A|p.(Leu105*)
stop_lost	OR2M3|NM_001004689.1|1/1|c.937T>A|p.(*313Argext*-313)
complex_substitution	
mnv	
frameshift_elongation	IFI44|XM_005270379.1|8/8|c.1295_1296insT|p.(Trp432Cysfs*13)
frameshift_truncation	CYP4B1|NM_000779.3|8/12|c.881_882del|p.(Asp294Glyfs*3)
frameshift_variant	MUC20|NM_001098516.|2/4|c.927del|p.(Ala309Alafs*40)
splice_acceptor_variant	TMEM216|NM_001173990.2|4/4|c.432-1G>C|p.?
splice_donor_variant	CGA|NM_001252383.1|3/4|c.181+2C>T|p.?
missense_variant	NOC2L|NM_015658.3|13/19|c.1528A>C|p.(Asn510His)
inframe_insertion	CPS1|NM_001122633.2|C2/39|c.15_16insTTC|p.(Ile5_Lys6insPhe)
disruptive_inframe_insertion	PCDHA4|NM_018907.2|1/4|c.209_210insACA|p.(Gly70_Arg71insHis)
inframe_deletion	DENND4B|NM_014856.2|18/28|c.2722_2730del|p.(Gln908_Gln910del)
disruptive_inframe_deletion	ZNF2|NM_001017396.1|4/4|c.345_347del|p.(Arg117del)
synonymous_variant	PANK4|NM_018216.1|3/19|c.393G>G|p.(=)
non_coding_transcript_exon_variant	WASH7P|NR_024540.1|11/11|n.1478G>A
non_coding_transcript_intron_variant	CROCCP2|NR_026752.1|6/6|n.917+74C>G
5_prime_utr_exon_variant	PHF13|NM_153812.2|1/4|c.-347C>G|p.(=)
3_prime_utr_exon_variant	PRDM16|NM_022114.3|17/17|c.*4240T>C|p.(=)
5_prime_utr_intron_variant	MAD2L2|NM_001127325.1|1/8|c.-12-634T>G|p.(=)
3_prime_utr_intron_variant	HTN1|NM_002159.2|5/5|c.*33+18G>A|p.(=)
stop_retained_variant	TMIGD2|NM_001169126.1|5/5|c.836G>A|p.(=)
initiator_codon_variant	
splice_region_variant	CROCC|NM_014675.3|17/36|c.2514+7A>C|p.?
upstream_gene_variant	NUDT17|NM_001012758.2|3176
downstream_gene_variant	NBPF10|NM_001039703.5|669
direct_tandem_duplication	MUC13|NM_033049.3|2/12|c.185_187dup|p.(Ser62dup)
intergenic_variant	MOB3C|NM_201403.2|Coding|25960

Chromosom	NCBI36	GRCh37	GRCh38
1	247 249 719	249 250 621	248 956 422
2	242 951 149	243 199 373	242 193 529
3	199 501 827	198 022 430	198 295 559
4	191 273 063	191 154 276	190 214 555
5	180 857 866	180 915 260	181 538 259
6	170 899 992	171 115 067	170 805 979
7	158 821 424	159 138 663	159 345 973
8	146 274 826	146 364 022	145 138 636
9	140 273 252	141 213 431	138 394 717
10	135 374 737	135 534 747	133 797 422
11	134 452 384	135 006 516	135 086 622
12	132 349 534	133 851 895	133 275 309
13	114 142 980	115 169 878	114 364 328
14	106 368 585	107 349 540	107 043 718
15	100 338 915	102 531 392	101 991 189
16	88 827 254	90 354 753	90 338 345
17	78 774 742	81 195 210	83 257 441
18	76 117 153	78 077 248	80 373 285
19	63 811 651	59 128 983	58 617 616
20	62 435 964	63 025 520	64 444 167
21	46 944 323	48 129 895	46 709 983
22	49 691 432	51 304 566	50 818 468
X	154 913 754	155 270 560	156 040 895
Y	57 772 954	59 373 566	57 227 415

Tabelle A2: **Chromosomenlängen**. Übersicht der Längen der Primärassemblies der *golden path* Chromosomen für die letzten drei humanen Referenzgenome.

Chrom	Position	Ref	Alt	Anzahl	*alt locus*	Info
chr2	131794984	A	AT	23/53	KI270768.1	HC2orf27B|NM_214461.2|c.808dup|p.(Ile270Asnfs*4)
chr4	68646933	T	TC	49/119	GL000257.2	UGT2B15|NM_001076.3|c.1763dup|p.(*588Trpext*11)
chr6	30949435	T	C	17/121	GL000250.2	DPCR1|NM_080870.3|c.970T>C|p.(*324Gln)
chr6	32828676	C	CCTCCACCCCA	5/7	GL000250.2	TAP2|NM_000544.3|c.2290_2291insTGGGGTGGAG|p.(Gly764Valfs*29)
chr15	30361719	G	A	1/32	GL383554.1	CHRFAM7A|NM_139320.1|c.1813C>T|p.(Gln605*)
chr15	32603707	C	T	1/65	GL383554.1	GOLGA8N|NM_001282494.1|c.1810C>T|p.(Gln604*)
chr17	37610127	A	AT	4/42	KI270857.1	DDX52|NM_001291476.1|16/16|c.5644dup|p.(Ile1882Asnfs*3)
chr19	54307242	C	CA	4/6	GL949746.1	LILRA5|NM_021250.3|c.1070dup|p.(Leu357Phefs*40)
chr19	54307242	C	CAA	2/3	GL949746.1	LILRA5|NM_021250.3|c.1069_1070dup|p.(Leu357Phefs*2)
chr22	23981036	A	G	4/11	KI270879.1	GSTT2|NR_126445.1|n.370+1A>G|

Tabelle A3: **ASDPs mit hohem Effekt und nicht in dbSNP gelistet**. Die Tabelle zeigt 10 ASDP-assoziierte Varianten, welche in der *in-house* Kohorte annotiert wurden und von *Jannovar* mit einem möglicherweise hohem Effekt vorhergesagt wurden. Die Spalte `Anzahl` sagt aus, wie oft die Variante als ASDP-assoziiert annotiert wurde.

A

```
2:1110:32373:30844    99   chr7   148380548   60   87M1I63M   =   148380991    594   ...
SA:Z:chr7_KI270808v1_alt,162211,+,151M,17,1;  ...
2:1110:32373:30844   147   chr7   148380991   60       151M   =   148380548   -594   ...
SA:Z:chr7_KI270808v1_alt,162655,-,151M,4,17;   ...
```

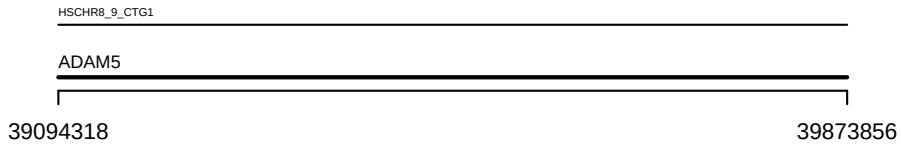
B

```
2:1110:32373:30844      2147    chr7_KI270808v1_alt     162211  60      151M  ...
2:1110:32373:30844      2195    chr7_KI270808v1_alt     162655  60      151M   ...
```

C

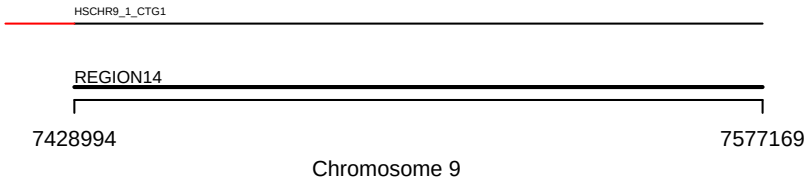
	Linkes Alignment	**Rechtes Alignment**
	Primäres Alignment	
Sample	ProbeA	ProbeA
Read length	151bp	151bp
Mapping Quality	MAPQ 60	MAPQ 60
Reference span	chr7:148.380.548-148.380.697 (+)	chr7:148.380.991-148.381.141 (-)
Cigar	87M1I63M	151M
Location	chr7:148.380.640	
Base	T @ QV 42	
Mate is mapped	yes	yes
Mate start	chr7:148.380.991 (-)	chr7:148.380.548 (+)
Insert size	594	-594
	First in pair	Second in pair
Pair orientation	F1R2	F1R2
	Supplementary Alignment	
Mapping Quality	MAPQ 60	MAPQ 60
Reference span	chr7_KI270808v1_alt:162.211-162.360 (+)	chr7_KI270808v1_alt:162.655-162.804 (-)
CIGAR	151M	151M
Mate is mapped	yes	yes
Mate start	chr7_KI270808v1_alt:162.655 (-)	chr7_KI270808v1_alt:162.211 (+)
Insert size	595	-595
	First in pair	Second in pair
Pair orientation	F1R2	F1R2

Abbildung A1: **SAM-Format Repräsentation der supplementary Reads.** Gezeigt werden die Einträge für das Alignment eines Read-Paars, welches eine ASDP-assoziierte homozygote Insertion aus Abbildung 3.2 überspannt. A) Verkürzte Repräsentation der Read-Alignments im SAM-Format. Hervorzuheben sind die sechste Spalte mit der CIGAR-Notation und die Info Spalte mit den supplementary Alignment Informationen (SA:Z). Der erste Mate-pair Read enthält die Insertion eines Ts, welche nicht im supplementary Alignment vorkommt. B) Die entsprechenden Einträge im SAM-Format für die Alignments auf dem *alt locus* und C) die kompletten Informationen zum primär und supplementary Alignment.



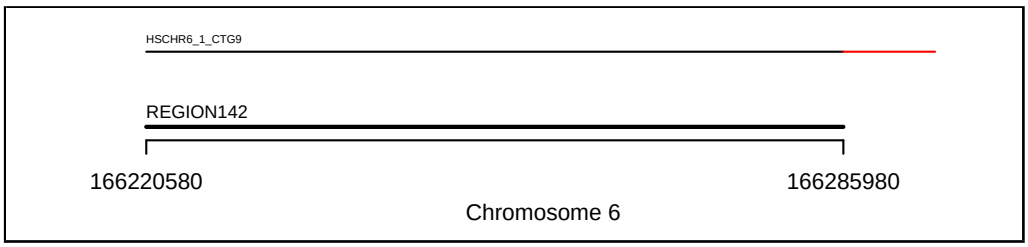
Feld	Wert
alt_asm_name	ALT_REF_LOCI_1
prim_asm_name	Primary Assembly
alt_scaf_name	HSCHR8_9_CTG1
alt_scaf_acc	NT_187577.1
parent_type	CHROMOSOME
parent_name	8
parent_acc	NC_000008.11
region_name	ADAM5
ori	+
alt_scaf_start	1
alt_scaf_stop	624492
parent_start	39094318
parent_stop	39873856
alt_start_tail	0
alt_stop_tail	0

Abbildung A2: **ADAM5 Region**. Die ADAM5 Region wird durch die Koordinaten eines einzelnen *alt locus* definiert. Die Region liegt auf Chromosom 8 (NC_000008.11) an Position 39 094 318–39 873 856 (779 539 Nukleotide). Die Länge des *alt locus* beträgt 624 492 Nukleotide. Die Tabelle zeigt die Werte aus der Datei alt_scaffold_placement.txt, welche den *alt locus* beschreiben. Siehe **Table A4** für eine Erklärung der einzelnen Felder der Tabelle.



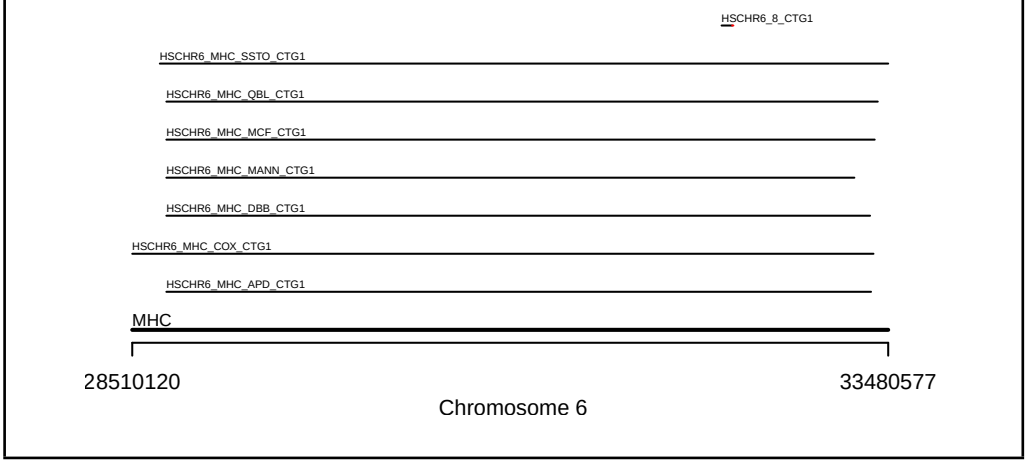
Field	Value
alt_asm_name	ALT_REF_LOCI_1
prim_asm_name	Primary Assembly
alt_scaf_name	HSCHR9_1_CTG1
alt_scaf_acc	NW_003315928.1
parent_type	CHROMOSOME
parent_name	9
parent_acc	NC_000009.12
region_name	REGION14
ori	+
alt_scaf_start	14845
alt_scaf_stop	162988
parent_start	7428994
parent_stop	7577169
alt_start_tail	14844
alt_stop_tail	0

Abbildung A3: **REGION14**. `REGION14` liegt auf Chromosom 9 (`NC_000009.12`) an Position 7 428 994–7 577 169 (148 176 Nukleotide). Das rote Segment ist eine Insertion am Anfang des Alignments des *alt locus* zu der Region. Siehe **Table A4** für eine Erklärung der einzelnen Felder der Tabelle.



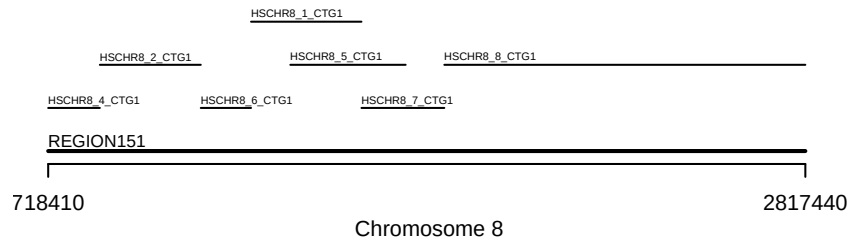
Field	Value
alt_asm_name	ALT_REF_LOCI_1
prim_asm_name	Primary Assembly
alt_scaf_name	HSCHR6_1_CTG9
alt_scaf_acc	NT_187557.1
parent_type	CHROMOSOME
parent_name	6
parent_acc	NC_000006.12
region_name	REGION142
ori	+
alt_scaf_start	1
alt_scaf_stop	66404
parent_start	166220580
parent_stop	166285980
alt_start_tail	0
alt_stop_tail	8601

Abbildung A4: **REGION142**. REGION142 liegt auf Chromosom 6 (NC_000006.12) an Position 166 220 580–166 285 980 (65 401 Nukleotide). Das rote Segment ist eine Insertion am Ende des Alignments des *alt locus* zu der Region. Siehe **Table A4** für eine Erklärung der einzelnen Felder der Tabelle.



Field	Value
alt_asm_name	ALT_REF_LOCI_1
prim_asm_name	Primary Assembly
alt_scaf_name	HSCHR6_MHC_APD_CTG1
alt_scaf_acc	NT_167244.2
parent_type	CHROMOSOME
parent_name	6
parent_acc	NC_000006.12
region_name	MHC
ori	+
alt_scaf_start	1
alt_scaf_stop	4672374
parent_start	28734408
parent_stop	33367716
alt_start_tail	0
alt_stop_tail	0

Abbildung A5: **MHC**. Die MHC Region liegt auf Chromosom 6 (NC_000006.12) an Position 28 510 120–33 480 577 (4 970 458 Nukleotide). Die Tabelle zeigt beispielhaft den korrespondierenden *alt locus* HSCHR6_MHC_APD_CTG1, welcher 672 374 Nukleotide lang ist. Für die Vergleiche wurden alle acht *alt loci* einzeln gegen die MHC Region aligniert. Siehe **Table A4** für eine Erklärung der einzelnen Felder der Tabelle.



Feld	Wert
alt_asm_name	ALT_REF_LOCI_1
prim_asm_name	Primary Assembly
alt_scaf_name	HSCHR8_4_CTG1
alt_scaf_acc	NT_187572.1
parent_type	CHROMOSOME
parent_name	8
parent_acc	NC_000008.11
region_name	REGION151
ori	+
alt_scaf_start	1
alt_scaf_stop	145606
parent_start	718410
parent_stop	861641
alt_start_tail	0
alt_stop_tail	0

Abbildung A6: **REGION151**. REGION151 liegt auf Chromosom 11 (NC_000008.11) an Position 718 410–2 817 440 (2 099 031 Nukleotide). Die Region wird durch sieben alternative Locus Scaffolds definiert, welche hier liegen. Die Tabelle zeigt den Eintrag für den *alt locus* HSCHR8_4_CTG1 zu REGION151, welche dort Position 718 410–861 641 (143 232 Nukleotide) auf Chromosome 8 entspricht und eine Länge von 145 606 Nukleotiden hat. Jeder der sieben *alt loci* wurde einzeln aligniert. Regionen, die nicht durch das Alignment abgedeckt sind, werden als hundertprozentig identisch zur Referenzsequenz betrachtet. Siehe **Table A4** für eine Erklärung der einzelnen Felder der Tabelle.

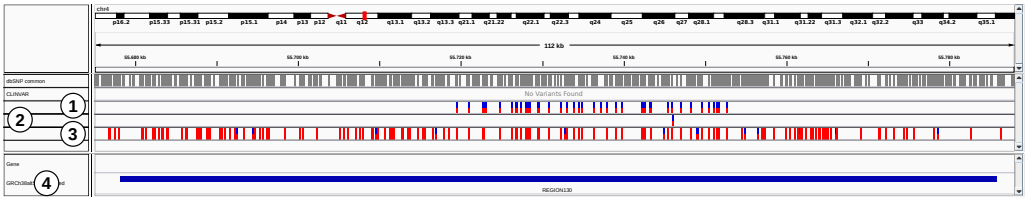


Abbildung A7: **Alignmentanker**. Das Alignment zwischen einer genomischen Region auf den Primärassembly und einem korrespondierenden *alt loci* beginnt und endet immer mit komplett identischen Ankersequenzen. In diesen kann man keine Abweichungungen finden und damit auch keine ASDPs. Die Analyses durch ASDPex beschränkt die Analyse der Varianten ③ in dieser Region auf den Bereich zwischen dem ersten und letzten ASDP. Die Abbildung zeigt einen IGV Screenshot von ④ REGION130 (Chromosom 4:55 678 095–55 785 754). Der ASDP enthaltenden Bereich ① & ② dieser Region beschränkt sich auf Chromosom 4:55 719 527–55 752 683.

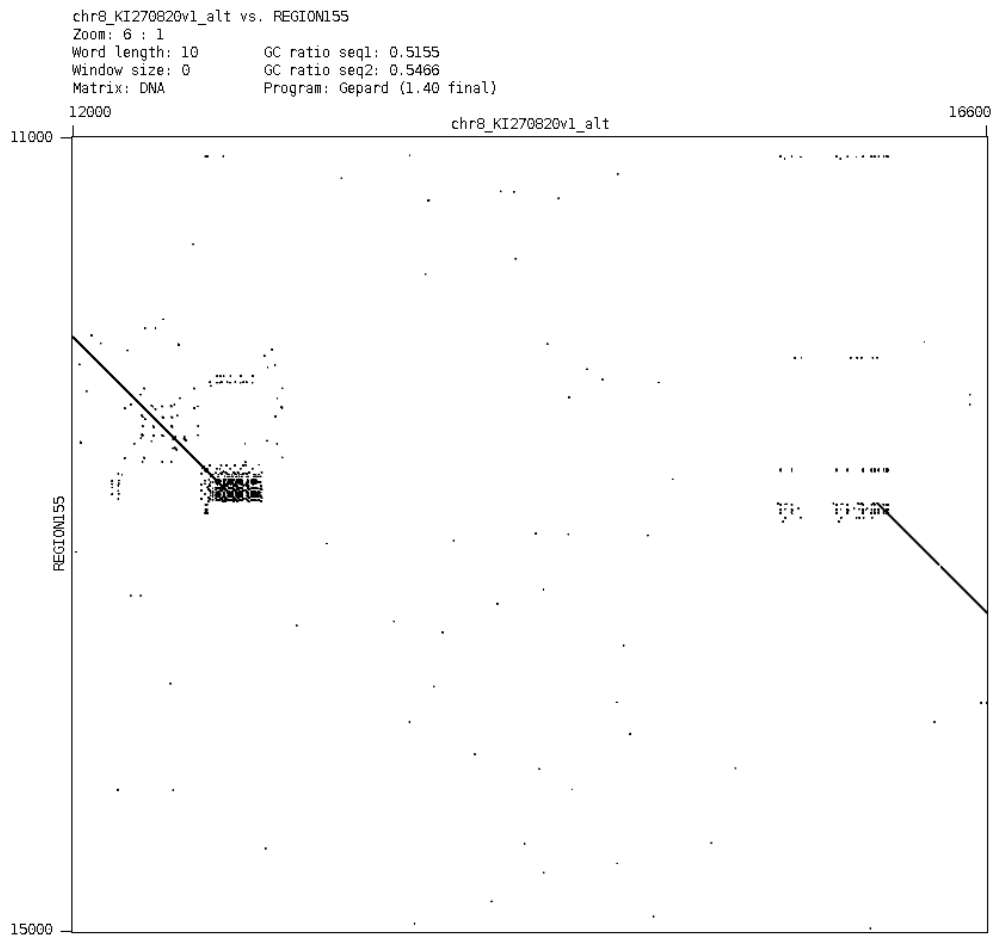


Abbildung A8: **DOT-Plot von `REGION155` vs.** `KI270820.1` mit einer Fenstergröße von 10 bp. Der Plot zeigt den Ausschnitt für die Basen 11 000–15 000 (`REGION155`) und 12 000–16 600 (`KI270820.1`), welche den Alignmentausschnitt in Abbildung 3.3 beinhalten. Deutlich zu erkennen sind die streuenden Ende an den Bruchpunkten der Insertion im alternativen Locus Scaffold.

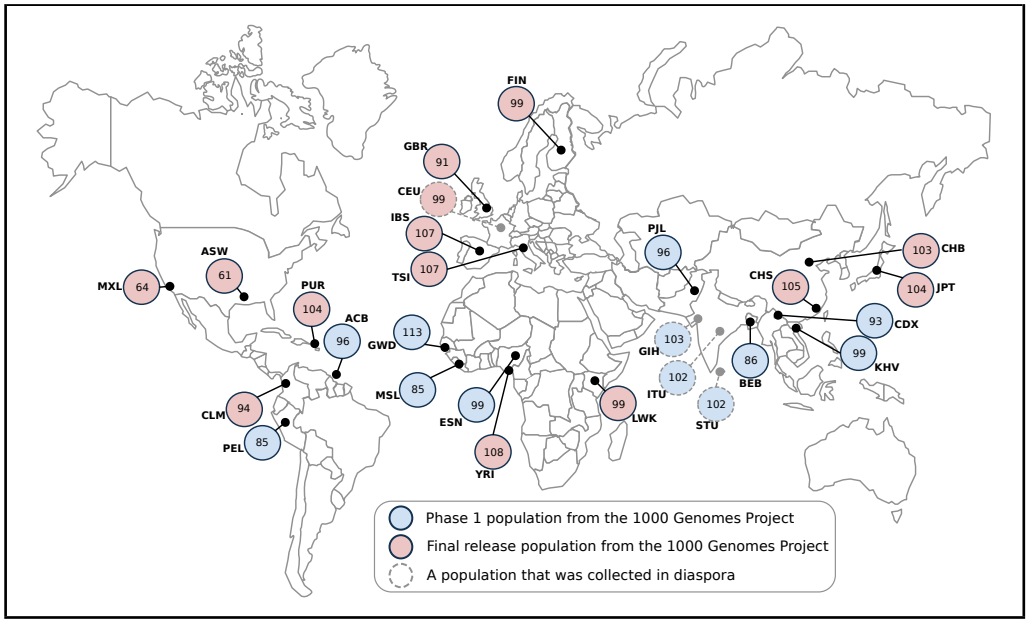
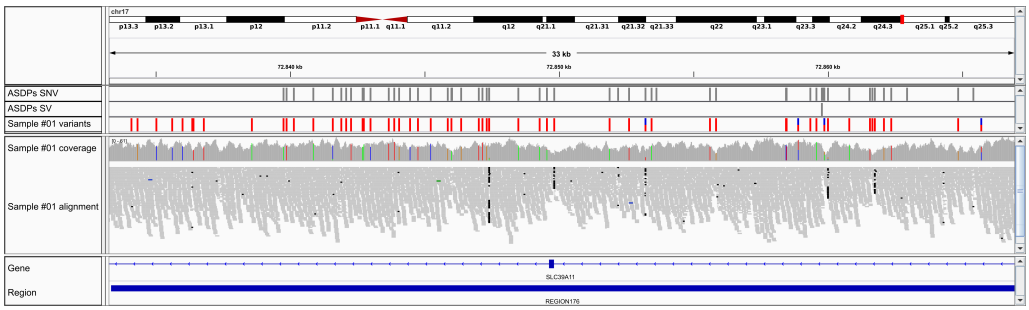


Abbildung A9: **1000-Genome-Projekt Populationen**. Übersicht über die Lokalisation der 27 Populationen des 1000-Genome-Projekts. In dieser Arbeit wurden die folgenden vier Populationen verwendet: FIN – Finnen aus Finland; LWK – Luhya aus Webuye, Kenya; CHB – Han Chinesen aus Beijing, China; PEL. Diese Abbildung wurde von der Projekthomepage `http://www.internationalgenome.org/` übernommen.

A)



B)

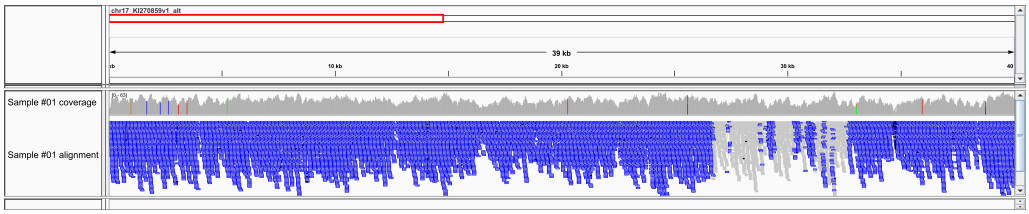


Abbildung A10: **REGION176 vs. KI270859.1**. A) Sample #01 zeigt zahlreiche homozygote ASDP-assoziierte Varianten in `REGION176` (72 833 239–72 866 965 auf Chromosom 17). B) Die korrespondierende Region auf dem *alt locus* `KI270859.1` zeigt deutlich weniger Unterschiede zwischen den alignierten Reads und dem ALT-HAP. Man kann annehmen, dass Sample #01 homozygot für `KI270859.1` vorliegt.

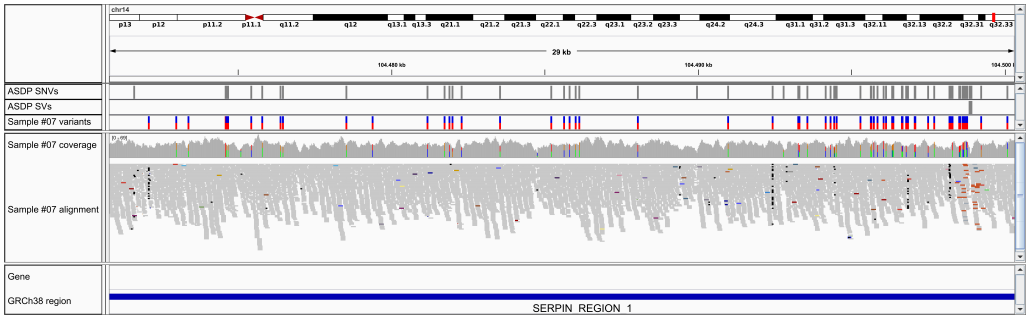
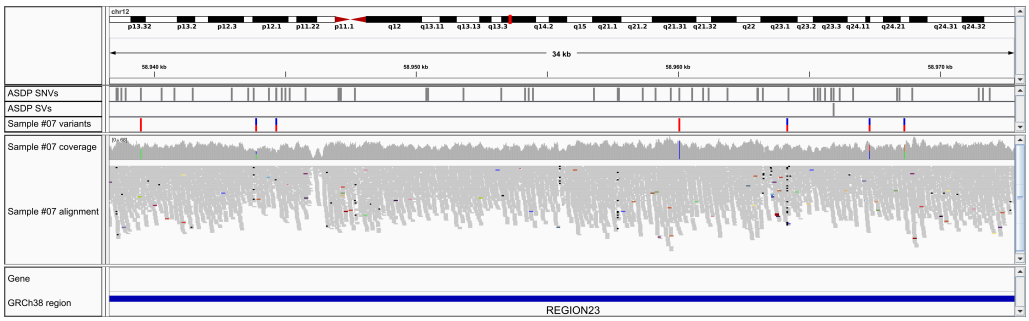


Abbildung A11: **SERPIN_REGION_1 vs. KI270845.1**. Sample #07 zeigt mehrere heterozygote ASDP-assoziierte Varianten, welche vom Alignment zwischen dem *alt locus* `KI270845.1` und der `SERPIN_REGION_1` (104 470 796–104 500 326 auf Chromosom 14) stammen. Man kann schlussfolgern, dass Sample #07 heterozygot für `KI270845.1` ist.

A)



B)

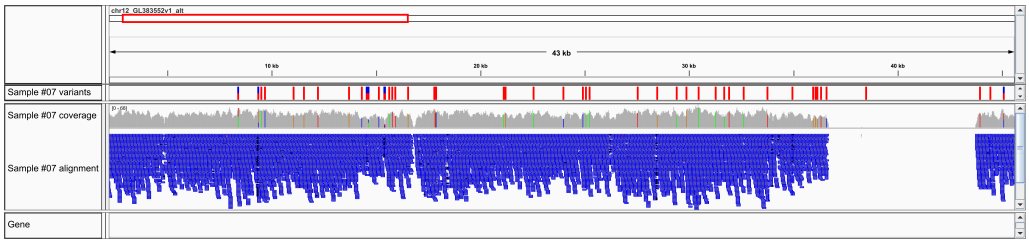


Abbildung A12: **REGION23 vs. GL383552.1**. Diese Abbildungen zeigen die (fehlende) Übereinstimmung der bekannten ASDPs und gecallten Varianten für eine zufällig ausgewählte Region, welche von ASDPex nicht als heterozygot oder homozygot für den ALT-HAP vorhergesagt wurde. A) Sample #07 besitzt kaum ASDP-assoziierte Varianten in `REGION23` (58 938 274–58 972 856 auf Chromosom 12). B) Die korrespondierende Region auf dem *alt locus* `GL383552.1` zeigt deutlich mehr Varianten. Man kann daraus schließen, dass der REF-HAP homozygot vorliegt.

Field	Example	Explanation
alt_asm_name	ALT_REF_LOCI_1	Alternativer Assemblyname
prim_asm_name	Primary Assembly	Primaryassemblyname
alt_scaf_name	HSCHR1_1_CTG3	Name des alternativen Locus Scaffold in der GenBank Datei referenziert von NT_187515.1
alt_scaf_acc	NT_187515.1	Accessionnumner des alternativen Locus Scaffold.
parent_type	CHROMOSOME	Type der "parentSequenz zu welcher der alternative Locus Scaffold zugeordnet ist.
parent_name	1	Der "parentName (i.e., chromosome 1).
parent_acc	NC_000001.11	Accessionnumber der "parentSequenz (entspricht den Informationen in der chr_accessions_GRCh38.p2) Datei
region_name	REGION108	Name der genomischen Region zu welcher der alternative Locus Scaffold zugeordnet ist (entspricht den Informationen in der genomic_regions_definitions.txt Datei; In diesem Beispiel ist spezifiziert, dass REGION108 auf NC_000001.11 lokalisiert ist, mit der Startposition 2 448 811 und der Stoppposition 2 791 270).
ori	+	"+", "-", oder "b".
alt_scaf_start	1	Startposition des Alignments in Bezug auf die Sequenz des alternativen Locus Scaffolds.
alt_scaf_stop	354444	Stopsposition des Alignments in Bezug auf die Sequenz des alternativen Locus Scaffolds. In diesem Beispiel umfasst das Alignment die gesamte Sequenz von NT_187515.1, was einer Länge von 354 444 bp entspricht.
parent_start	2448811	Startposition des Alignments in Bezug auf die Sequenz des "parent".
parent_stop	2791270	Stopposition des Alignments in Bezug auf die Sequenz des "parent". In diesem Beispiel entsprechen die Start- und Stoppositionen den Koordinaten der REGION108 in der genomic_regions_definitions.txt Datei.
alt_start_tail	0	Dies ist die Länge einer Insertion vor dem Anfang im Vergleich zur Referenzsequenz. Zum Beispiel hat NT_187517.1 einen alt_start_tail von 20 632 Nukleotiden und das Alignment mit der korrespondierenden Referenz beginnt in Position 20 633 der Sequence in NT_187517.1
alt_stop_tail	0	Analog zu alt_start_tail jedoch ist die Insertion am Ende des Scaffolds; siehe z.B. NT_187557.1.

Tabelle A4: **Spalten in der Positionstabelle für die alternativen Scaffolds**. Erläuterung der einzelnen Spalten in der alt_scaffold_placement.txt Datei. Diese beschreibt die Positionierung der *alt loci* auf den Primärassemblies der Chromosomen.