MutationTaster

ein web-basiertes Computerprogramm zur Bewertung des Krankheitspotentials von DNA-Mutationen

Dissertation

zur Erlangung des akademischen Grades
des Doktors der Naturwissenschaften (Dr. rer. nat.)

eingereicht im Fachbereich Biologie, Chemie, Pharmazie
der Freien Universität Berlin

vorgelegt von

Jana Marie Schwarz

Mai 2013

Angefertigt zwischen April 2009 und Mai 2013 unter der Leitung von Prof. Dr. Markus Schülke-Gerstenfeld in der Klinik für Pädiatrie mit Schwerpunkt Neurologie und dem NeuroCure Clinical Research Center (NCRC) - Charité Universitätsmedizin Berlin.

Erster Gutachter:
Prof. Dr. rer. nat. Stephan Sigrist
Fachbereich Biologie, Chemie, Pharmazie
Institut für Biologie, Freie Universität Berlin
Takustr. 3, 14195 Berlin

Zweiter Gutachter:
Prof. Dr. med. Markus Schülke-Gerstenfeld
Klinik für Pädiatrie m. S. Neurologie, Campus Virchow Klinikum
Charité - Universitätsmedizin Berlin
Augustenburger Platz 1, 13353 Berlin

Disputation am 13. August 2013

Danksagungen

Mein besonderer Dank gilt meinem Kollegen und Betreuer der Doktorarbeit, Dr. rer. medic. Dominik Seelow. Dank ihm habe ich mich auf ein mir bis vor einigen Jahren völlig fremdes Terrain, die Bioinformatik, oder - wie wir lieber sagen - die „angewandte Informatik“ getraut - und festgestellt, dass es Spaß macht! Darüber hinaus, und das ist noch viel wichtiger, hat er mich immer wieder ermutigt, IT-Probleme selber zu lösen, und mir so viel Selbstvertrauen gegeben. Ich danke ihm für die hervorragende Betreuung, seine Geduld in Programmier- und Datenbankfragen und für die gute Zusammenarbeit.

Meinem Gutachter und Doktorvater Prof. Dr. Markus Schülke gilt ebenfalls mein besonderer Dank. Er hat mir nicht nur die Anfertigung der Doktorarbeit ermöglicht, sondern auch sehr schnell das Gefühl gegeben, ein vollwertiges und wichtiges Mitglied der Arbeitsgruppe zu sein. Ich danke ihm für sein Vertrauen.

Ganz herzlich danken möchte ich auch meinem ehemaligen Kommilitonen Tobias Winter. Mit ihm zusammen entstand die allererste, sehr frühe Version von MutationTaster (die damals noch nicht so hieß), und ohne ihn hätte ich vermutlich nicht so viel Freude an dem Projekt entwickelt.

Frank Schacherer von der Firma BIOBASE ermöglichte uns durch eine Kooperation den Zugriff auf Inhalte der Mutationsdatenbank HGMD - vielen Dank dafür.

Prof. Dr. Stephan Sigrist möchte ich dafür danken, dass er sich bereit erklärt hat, meine Dissertation zu begutachten.

Allen anderen Mitgliedern der AG Schülke bin ich dankbar für ihre Hilfe in diversen Laborfragen und für ein freundliches und angenehmes Arbeitsklima.

Ein großes Dankeschön geht an die mir unbekannte(n) Person(en) an der Charité, die durch die Realisierung der KidsMobil Kinderbetreuung für Mitarbeiter eine großartige und enorm hilfreiche Form der Unterstützung zur Vereinbarkeit von Beruf und Familie geschaffen haben.

Meiner Familie und meinen Freunden, insbesondere Martin, gilt ebenfalls mein Dank für die stetige Unterstützung in großen wie in kleinen Dingen.

Inhaltsverzeichnis

Vorwort

Obwohl es für viele Fachwörter aus Biologie, Medizin oder Informatik deutsche Begriffe gibt, werden die meisten von ihnen im Alltag nur selten benutzt. Sie sind folglich im deutschen Sprachraum auch unter Wissenschaftlern teils unbekannt. Ich bemühe mich in der vorliegenden Arbeit, deutsche Fachbegriffe zu verwenden, sofern es gebräuchliche gibt. Ich bitte allerdings um Verständnis, dass ich in einigen Fällen auf die gängigeren englischen Varianten zurückgreife (z.B. *missense* Mutation statt *Fehlsinnmutation*). Für diese Fälle werde ich bei der ersten Benutzung eines englischen Fachbegriffs die deutsche Entsprechung bzw. Übersetzung in Klammern angeben, im Folgenden jedoch nicht weiter verwenden. Ich verwende außerdem aus Gründen der besseren Lesbarkeit das generische Maskulinum, welches sowohl die weibliche als auch die männliche Form umfasst (z.B. *der Benutzer* statt *der Benutzer, die Benutzerin*). Nicht allgemein verständliche Fachbegriffe werden im Glossar im Anhang erläutert und bei ihrem ersten Auftreten mit einem Sternchen (*) markiert.
Das Thema dieser Dissertation, die Software *MutationTaster*, basiert im Wesentlichen auf einer Methode, deren prinzipielle Funktionalität ich bereits im Jahr 2008 in einer dieser Dissertation vorangehenden Masterarbeit zeigen konnte. Die damals entstandene frühe Version von MutationTaster unterscheidet sich in wesentlichen Aspekten von der jetzt verfügbaren und hier vorgestellten Version, auch wenn das endgültige Ziel, nämlich die Bewertung des Krankheitspotentials einer DNA-Variante, beiden gemein ist. Die Idee dazu hatte Dominik Seelow, der mich im dritten Modul meines Masterstudiums der Molekularen Medizin in einem mit meinem damaligen Kommilitonen Tobias Winter gemeinsam absolvierten Praktikum betreute. Unsere Aufgabe war es, ein System zur einheitlichen Bewertung von DNA-Sequenzveränderungen zu entwickeln. Im Praktikum entstand eine recht umständlich zu bedienende, jedoch in Ansätzen vielversprechende „Gebrauchsanweisung“ für den Umgang mit Mutationen. Das Projekt gefiel mir, in meiner Masterarbeit wurde somit aus einer besseren „Gebrauchsanweisung“ ein kleines Computerprogramm namens *MutationTaster*. Der wesentliche Fortschritt bestand zu dem Zeitpunkt in einer Validierung mit einem größeren Testset sowie einer durch Automatisierung von Prozessen verbesserten Benutzbarkeit. Positive Rückmeldungen, das Gefühl, dass MutationTaster nach wie vor „unfertig“ war, und die Vermutung, dass zeiteffiziente und automatische Lösungen mit dem zunehmenden Erfolg des *Next Generation Sequencing* gebraucht würden, bewegten mich dazu, auch meine Promotion diesem Thema zu widmen. Das Ziel war diesmal, eine für die Öffentlichkeit und für den Routineeinsatz geeignete, performante und benutzerfreundliche Software zu erschaffen. Deren Entstehung wird in der hier vorliegenden Arbeit beschrieben. Dominik Seelow, dessen Expertise in der Softwareentwicklung und Informatik die meine bei Weitem übersteigt, hat neben der Betreuung der Doktorarbeit außerdem die Implementierung eines Bayes Klassifikators in MutationTaster übernommen und die Datenbankstruktur optimiert.

1 Einleitung

1.1 Monogene Erkrankungen

Monogene Erkrankungen sind Krankheiten, die durch einen von Geburt an bestehenden Defekt in einem einzelnen Gen hervorgerufen werden. Im menschlichen Organismus gibt es aufgrund des doppelten Chromosomensatzes von 2n,XY = 46 (22 Autosomen sowie die geschlechtsspezifischen Chromosomen X und Y) von fast jedem Gen zwei Kopien, die nicht zwingend identisch sind. Der Fachbegriff für eine alternative Form eines Gens oder genetischen Markers ist *Allel**. Wenn beide Kopien, d.h. beide Allele, identisch sind, spricht man von *Homozygotie**, bei zwei unterschiedlichen Allelen von *Heterozygotie**. Die meisten monogenen Krankheiten werden vererbt. Vorfahren geben Krankheitsallele typischerweise gemäß den Mendel'schen Regeln an nachfolgende Generationen weiter, weshalb sie auch Mendel'sche Krankheiten genannt werden. Wenn bereits eine defekte Kopie eines Gens zur Manifestation der Krankheit führt, handelt es sich in der Regel um eine dominant vererbte Krankheit. Das Krankheitsmerkmal liegt hier heterozygot vor. Bei vollständiger *Penetranz** erkrankt jeder Träger des Merkmals. Führen erst zwei fehlerhafte Kopien eines Gens zur Krankheit (Homozygotie) nennt man dies rezessive Vererbung. Es kann auch passieren, dass zwei verschiedene, heterozygote Mutationen zur Manifestation einer eigentlich rezessiv vererbten Krankheit führen. In diesem Fall beruht die Merkmalsausprägung nicht auf Homozygotie der gleichen Mutation, sondern auf unterschiedlichen Mutationen in beiden Kopien des gleichen Gens (*Compound-Heterozygotie**). Der Genort (*Locus**), also die genaue chromosomale Positionsangabe, spielt auch eine Rolle: Liegt ein Gendefekt auf einem Autosom, also auf einem der Chromosomen 1-22, spricht man von autosomal-dominanter oder autosomal-rezessiver Vererbung. Bei einer Frau kann auch ein Gendefekt auf dem X-Chromosom dominant oder rezessiv vererbt werden (X-chromosomal dominant / X-chromosomal rezessiv). Da ein Mann jeweils nur ein X- und ein Y-Chromosom hat, wird der *Genotyp** als hemizygot bezeichnet. Man spricht nur von X- bzw. Y-chromosomaler Vererbung. Fehler in mitochondrial kodierten Genen stellen einen Sonderfall dar: Eine mitochondrial vererbte Krankheit mag auf den ersten Blick einen autosomalen Erbgang aufweisen, da beide Geschlechter betroffen sein können, jedoch gibt der Vater die Krankheit so gut wie nie an seine Nachkommen weiter. Menschliche mitochondriale DNA wird in der Regel ausschließlich maternal, d.h. über die weibliche Eizelle, vererbt. Monogene Erkrankungen können alle vorgestellten Erbgänge aufweisen [1], die meisten werden aber rezessiv vererbt. Der Grund dafür ist, dass rezessiv vererbte Krankheitsmerkmale weniger stark der Selektion unterliegen als dominant vererbte Krankheitsmerkmale, bei denen in der Regel jeder Merkmalsträger erkrankt. Je nachdem in welchem Alter solch eine unter Umständen schwere Krankheit ausbricht, besteht eine geringere Chance auf eine Vererbung an Nachkommen, da diese vermutlich gar nicht erst gezeugt werden können. Die Chance auf Propagation ist bei vergleichbar schweren, rezessiv vererbten

Krankheiten also höher. Die Sichelzellanämie ist ein Beispiel für eine autosomal-rezessiv vererbte, monogene Krankheit. Durch den Austausch eines einzelnen Nukleotids in der proteinkodierenden Region des β-Globin Gens auf Chromosom 11 werden die physikalischen Eigenschaften des Hämoglobins so verändert, dass es bei Sauerstoffmangel aggregiert und dadurch die Erythrozyten sichelförmig verformt werden [2]. Weitere Beispiele für monogene Erkrankungen sind die autosomal-dominant vererbte Neurofibromatose, die Mukoviszidose (autosomal-rezessiv) oder die X-chromosomal vererbte Muskeldystrophie Duchenne.
Die Datenbank OMIM (*O*nline *M*endelian *I*nheritance in *M*an) [3] enthält Informationen zu allen bekannten Mendel'schen Krankheiten sowie assoziierten Genen. Aktuell werden dort mehr als 7000 Einträge zu vermuteten oder nachgewiesenen monogenen Erkrankungen verzeichnet, wobei die molekulargenetische Ursache nur ungefähr für die Hälfte davon bekannt ist. Viele monogene Erkrankungen sind für sich betrachtet und im Gegensatz zu komplexen Erkrankungen wie Diabetes oder Herz-Kreislauferkrankungen selten, in ihrer Gesamtheit jedoch betreffen sie aber sehr viele Menschen (die WHO spricht in diesem Zusammenhang von Millionen Betroffener weltweit [4]). Weiterhin mehren sich die Hinweise, dass einige bislang als relativ häufig angesehene und in ihrer molekularen Ursachen komplexe Krankheiten wie Autismus, Schizophrenie oder mentale Retardierung besser beschrieben werden könnten als eine sehr heterogene Gruppe seltener monogener Erkrankungen [5], von denen die Mehrzahl jedoch noch unbekannt ist.
Früher war die Erforschung der Ursache einer genetischen Erkrankung ein sehr arbeitsintensiver, mehrstufiger Prozess bestehend aus Kopplungsanalyse, Kandidatengensuche, Sequenzierung einzelner Gene (bzw. deren Exons) in Patienten und gesunden Kontrollen sowie gegebenenfalls abschließenden, funktionalen Tests. Eine neue Sequenzierungstechnologie, das sogenannte *Next Generation Sequencing** (NGS), hat in den letzten Jahren allerdings die Erforschung der Ursachen genetisch bedingter Erkrankungen revolutioniert. Mit dieser Hochdurchsatztechnologie ist es möglich, einzelne große genomische Regionen, das komplette Exom oder sogar das gesamte Genom auf einmal zu sequenzieren. Im Januar 2010 wurde der erste erfolgreiche Einsatz der *Exomsequenzierung**, also der Sequenzierung der kompletten kodierenden Bereiche des Genoms zur Identifikation einer Krankheitsmutation, publiziert [6]. Seitdem steigt die Zahl derartiger Veröffentlichungen rapide an. Bislang konnten mehr als 70 monogene Erkrankungen mit dieser Methode aufgeklärt werden [7]. Für Patienten bedeutet das Wissen um die molekulare Ursache ihrer Erkrankung zunächst die Möglichkeit einer gesicherten Diagnose und die Vermeidung zahlloser unnötiger und belastender Untersuchungen oder unwirksamer Therapien. Auch eine genetische Beratung bezüglich Familienplanung und Überträgerstatus wird ermöglicht, wenn Krankheitsgen und -mutation bekannt sind. Nicht zuletzt ermöglicht deren Identifikation möglicherweise auf lange Sicht die Entwicklung einer besseren, gezielteren Therapie.

1.2 Mutation *versus* Polymorphismus

Eine Mutation ist eine dauerhafte Veränderung des Erbgutes einer Zelle [8]. Diese kann harmlose, schädliche oder sogar vorteilhafte Folgen haben. In der Biologie allgemein spricht man bereits von einer Mutation, wenn sich besagte Veränderung in der DNA einer Zelle findet, ohne zu berücksichtigen, welchen Effekt diese für den Organismus hat [9]. Im Sprachgebrauch der Molekulargenetiker wird der Begriff Mutation häufig als Synonym für eine DNA-Veränderung mit negativem, d.h. krankheitsverursachendem Effekt benutzt [8][10]. Für die generelle Beschreibung vieler biologischer bzw. genetischer Prozesse oder Phänomene wird der Begriff Mutation jedoch ebenfalls verwendet, ohne dass etwas über deren Folgen bekannt wäre. So wird beispielsweise der Begriff *missense* Mutation oft benutzt, um zum Ausdruck zu bringen, dass eine DNA-Variante die Aminosäuresequenz des kodierten Proteins verändert. Rückschlüsse auf die Pathogenität der DNA-Variante kann man in diesem Zusammenhang aus der bloßen Verwendung des Wortes Mutation nicht schließen.
Bei einem Polymorphismus handelt es sich um eine genetische Variation innerhalb einer Population [11]. Viele Wissenschaftler verlangen von einem Polymorphismus, dass mindestens zwei Allele mit einer Frequenz von wenigstens 0,01 in einer Population vorkommen [9][12], andere bezeichnen alle DNA-Veränderungen ohne offensichtliche schädliche Folgen als Polymorphismus und benutzen den Begriff somit als Synonym für „harmlose Veränderung“. Eine besondere Form des Polymorphismus ist der SNP (*single nucleotide polymorphism**, Einzelbasenpolymorphismus), eine DNA-Variante, die nur ein einzelnes Nukleotid betrifft. Die bekannte Online-Ressource dbSNP [13][14], die „weltgrößte Datenbank für DNA-Variationen“
[15] benutzt als Einschlusskriterium für Varianten die eher lose Definition eines SNPs als eine „genetische Variation“, ohne gesondertes Augenmerk auf Allelfrequenzen zu richten. Folgerichtig enthält sie sowohl harmlose als auch krankheitsverursachende [16], und sowohl häufige als auch selten auftretende DNA-Veränderungen. Neben SNPs sind dort außerdem auch andere DNA-Veränderungen, die keine Einzelbasenaustausche sind, gespeichert.
Es wird deutlich, wie unpräzise die Begriffe SNP und Mutation verwendet werden. Der für die vorliegende Dissertation wichtigste Aspekt einer DNA-Veränderung ist deren krankheitsverursachendes Potential. Zur zweifelsfreien und klaren Darstellung des Inhalts dieser Arbeit werde ich von krankheitsverursachenden, für den Organismus schädlichen DNA-Varianten immer als *Krankheitsmutation* sprechen und den Begriff *Polymorphismus* im Sinne von harmlosen DNA-Veränderungen verwenden. Die Begriffe „DNA-Variante“, „DNA-Veränderung“ oder „Mutation“ lassen keine Rückschlüsse auf deren Pathogenität zu. Vor allem in der Einleitung werde ich den Ausdruck Mutation verwenden, weil er im Kontext der zu erläuternden Sachverhalte gebräuchlicher ist als „DNA-Variante“ oder „DNA-Veränderung“. Im weiteren Verlauf der Arbeit werde ich für nicht klassifizierte DNA-Veränderungen, wann immer möglich, die von mir präferierten Begriffe „DNA-Veränderung“ oder „DNA-Variante“ be-

nutzen.

1.3 Klassifikation von Mutationen

Mutationen können hinsichtlich ihrer Ursache, Größe, sowie ihrer Folge für den Organismus sehr unterschiedlich sein. Eine grobe Einteilung nach Größe ergibt drei Arten von Mutationen: (1) die strukturellen Variationen, auch strukturelle Chromosomenabberationen genannt, bei denen Teile einzelner Chromosomen verändert sind, z.B. durch *CNVs** (*C*opy *N*umber *V*ariation, Kopienzahlvariation), Inversion, Translokation, Deletion oder Insertion; (2) die numerischen Chromosomenaberration, bei denen komplette Chromosomen fehlen oder vermehrt vorhanden sind, und (3) Mutationen mit kleinerer Ausdehnung, die nur eines oder mehrere Nukleotide innerhalb oder außerhalb eines Gens betreffen. Für diese Arbeit relevant sind ausschließlich die Mutationen mit kleinerer Ausdehnung, weshalb ich im Folgenden eine feinere Einteilung eben dieser vornehmen werde.

1.3.1 Einteilung nach Art der Veränderung

Eine Mutation kann grundsätzlich eine oder mehrere Basen betreffen. Wird eine einzelne Base durch eine andere ersetzt (ein sogenannter Einzelbasenaustausch), spricht man auch von einer Punktmutation. Wenn eine oder mehrere Basen komplett fehlen, spricht man von einer Deletion, sind eine oder mehrere Basen zusätzlich vorhanden von einer Insertion. Manchmal treten beide Phänomene gekoppelt auf, das heißt, dass gleichzeitig Basen fehlen und neu hinzugefügt wurden. Solche Veränderungen werden häufig in ihrer Gesamtheit betrachtet und folglich als InDel (zusammengesetzt aus *In*sertion und *Del*etion) bezeichnet.

1.3.2 Einteilung nach Effekt auf das Protein

Viele Gene enthalten die Information zur Bildung ihrer Genprodukte, der Proteine. Ein Protein ist eine Kette von Aminosäuren. Der genetische Code, also die in der Basensequenz der DNA verschlüsselt vorliegende Information zur Bildung einer Aminosäuresequenz, ist redundant: Es gibt 64 Codons, von denen 61 für nur 20 Aminosäuren kodieren. In menschlichen Körperzellen sind drei der Codons, nämlich TAA, TAG und TGA, sogenannte Stopcodons (für Mitochondrien gelten andere Regeln). Viele Aminosäuren werden durch mehr als ein Codon kodiert, die Unterschiede liegen meistens in der dritten Base. Das hat zur Folge, dass einige DNA-Veränderungen trotz einer Veränderung des ursprünglichen Codons nicht zu einer Änderung der Aminosäuresequenz führen - sie sind still bzw. *synonym**. Im Gegensatz dazu stehen die *nicht-synonymen** DNA-Veränderungen, bei denen der Austausch einzelner Nukleotide auch zu einem Aminosäureaustausch führt. Obwohl synonyme DNA-Varianten nicht direkt die Aminosäuresequenz abwandeln, können sie wichtige Regulationssequenzen (z.B. für die Transkription, das Spleißen der mRNA oder für die Effizienz der Translation)

verändern und somit indirekt die Struktur und / oder Menge des resultierenden Proteins beeinflussen. Die biologischen Grundlagen von Spleißen, Transkriptions- und Translationsregulation sind jedoch äußerst komplex und noch lange nicht vollständig aufgeklärt, so dass eine mögliche Beeinträchtigung durch eine stille Mutation meist nicht ohne aufwendige experimentelle Laboruntersuchungen (z.B. durch die Analyse der *cDNA**) gesichert werden kann.
In der Klasse der nicht-synonymen Veränderungen kann man wiederum unterscheiden zwischen sogenannten *missense* (Fehlsinn) Mutationen und *nonsense* (Unsinn) Mutationen. *Missense* bedeutet, dass anstelle der regulären Aminosäure nun eine andere Aminosäure tritt, welche die ursprüngliche Proteinfunktion beeinträchtigen kann. Bei einer *nonsense* Mutation wird durch die DNA-Veränderung keine Aminosäure mehr kodiert, sondern es entsteht ein Stopcodon. Die Aminosäuresequenz bricht vorzeitig ab, das Protein ist unvollständig oder wird durch RNA-Überwachungsmechanismen (*nonsense-mediated mRNA decay, NMD**) degradiert und fehlt somit komplett [17]. *Frameshift* Mutationen (das Leseraster verändernde Mutationen) entstehen durch das Fehlen oder zusätzliche Vorhandensein von Nukleotiden, deren Anzahl nicht drei oder ein Vielfaches davon ist. Abgesehen davon, dass ab der *frameshift* Mutation zunächst eine völlig neue Aminosäuresequenz generiert wird, führen die meisten *frameshift* Mutationen zur Entstehung eines verfrühten Stopcodons. Ein derart verkürztes Protein wird anschließend möglicherweise durch NMD-Mechanismen degradiert. *Frameshift* Veränderungen können aber nicht nur durch die Deletion oder Insertion von Nukleotiden entstehen, sondern auch durch Spleißmutationen, die das Überspringen von Exons (*exon skipping*), das Beibehalten von Introns (*intron retention*) oder die Entstehung komplett neuer Spleißstellen und damit ebenfalls eine Veränderung des Leserasters nach sich ziehen.

1.3.3 Einteilung nach Folgen für den Organismus

Durch die unterschiedlichen Effekte von Mutationen auf die Proteinmenge, -sequenz und -struktur können sich verschiedene Folgen für den Organismus ergeben. Grundsätzlich kann man hier unterscheiden zwischen *loss-of-function* (LOF) Mutationen (Funktionsverlustmutationen) und *gain-of-function* (GOF) Mutationen (Funktionsgewinnmutationen). Bei LOF Mutationen hat das betroffene Gen nur noch eine eingeschränkte oder fast gar keine Funktion mehr und führt unter Umständen zur Erkrankung des Merkmalsträgers (z.B. klassische Phenylketonurie [18]). Wenn überhaupt keine normale Genaktivität mehr vorhanden ist, spricht man von einer *Nullmutation* (*null mutation*). LOF Mutationen sind meistens rezessiv, in vielen Fällen funktioniert der menschliche Körper noch, so lange wenigstens eine intakte Kopie des Gens vorhanden ist (Heterozygotie). Sobald aber die zweite Kopie des Gens ebenfalls beschädigt ist (Homozygotie), kann es durch das völlige Fehlen des Proteins zu Schäden für den Organismus kommen. Es kann allerdings auch passieren, dass 50% des Genprodukts für eine normale Funktion nicht ausreichen, so dass LOF Mutationen einen

dominanten *Phänotyp** aufweisen (*Haploinsuffizienz**). Dies ist beispielsweise häufiger der Fall bei Krankheitsmutationen in Genen für Transkriptionsfaktoren [19]. Ein weiterer Sonderfall ist ein *dominant-negativer** Effekt, bei dem das Produkt eines mutierten Allels mit dem Produkt des intakten Allels interferiert und somit die normale Genaktivität blockiert (*gain-of-pathologic-function*). Bei GOF Mutationen erwirbt das Protein eine neue, abnorme Funktion. Einige Tumorerkrankungen entstehen, weil GOF Mutationen zur Aktivierung eines Proto-Onkogens führen (z.B. [20]). Sie werden meist dominant vererbt, denn der abnorme Funktionsgewinn des Proteins kommt auch bereits bei nur einem mutierten Allel zum Tragen. Der Gendefekt wird nicht durch eine noch intakte Kopie des Gens verhindert. Es liegt dann faktisch ein Funktionsverlust vor. Die Einteilung in Funktionsverlust- oder Funktionsgewinnmutationen ist nicht immer einfach, die Bestimmung des Erbganges kann hierbei hilfreich sein [21][22].

1.4 Bewertung des Krankheitspotentials von DNA-Veränderungen

Mit dem routinemäßigen Einsatz der DNA-Sequenzierung zur Erforschung der Ursachen genetischer Krankheiten rückte eine wichtige Frage immer mehr in den Fokus der Wissenschaftler und Ärzte: Wie können gefundene DNA-Varianten als Ursache für bestimmte Krankheiten identifiziert und bestätigt werden? Der endgültige Beweis liegt dann vor, wenn in Laborexperimenten nachgewiesen werden kann, dass eine bestimmte DNA-Veränderung sich pathologisch auf die Funktion von Zellen und dem Organismus auswirkt oder die Zellfunktion durch Einbringung des gesunden Gens wieder hergestellt werden kann (*rescue* Experiment; „Rettungs“experiment. Jedoch sind Laboruntersuchungen sehr zeitaufwendig und teuer, weshalb gefundene DNA-Veränderungen in der Regel vorab auf ihr Krankheitspotential überprüft werden. Zwei verschiedene Methoden werden im Folgenden vorgestellt.

1.4.1 Manuelle Bewertung

Zur manuellen Bewertung des Krankheitspotentials von DNA-Varianten im Zusammenhang mit einer monogenen Erkrankung stehen verschiedene biologische / medizinische Datenbanken sowie Computerprogramme für unterschiedliche Analysen zur Verfügung. Ein häufiger Ausgangspunkt ist die Suche in der Datenbank dbSNP [13], um auszuschließen, dass die fragliche Variante ein bereits bekannter und harmloser Polymorphismus ist. OMIM [3] oder ClinVar (siehe Kapitel 2.2.1) können konsultiert werden, um zu überprüfen, ob die Variante bereits als Ursache einer Krankheit identifiziert worden ist. Ist die Variante bislang unbekannt, wird die Natur der DNA-Veränderung genauer untersucht. Mögliche Hinweise auf deren Pathogenität liefern zum Beispiel folgende Aspekte:

- Führt die DNA-Veränderung zu einem Aminosäureaustausch?
- Ist die veränderte Aminosäure evolutionär konserviert?

- Ist ein eventueller Aminosäureaustausch in einer funktionellen Proteindomäne (z.B. Transmembrandomäne, Bindungsstelle für Interaktionspartner o.ä.) lokalisiert?
- Könnte die DNA-Veränderung das Spleißmuster verändern?
- Beeinflusst die Variante DNA-Regulationsmechanismen (z.B. *Kozak**-Sequenz oder Adenylierungssignal)?

Diese und weitere Fragestellungen können helfen, das Krankheitspotential einer DNA-Veränderung einzuschätzen, allerdings ist für ihre Beantwortung jeweils eine andere Datenbank oder ein anderes Computerprogramm nötig. Tabelle 1 listet mögliche Online-Ressourcen (Datenbanken und Computerprogramme) auf. Die Evaluation bereits einer einzigen DNA-Veränderung ist auf diese Art und Weise sehr zeitaufwendig. Für die unter Umständen mehreren tausend Varianten, die mit neuen Sequenzierungstechnologien auf einen Schlag gefunden werden, ist sie *de facto* unmöglich.

Typ	Ressource	Referenz
Allgemeine DB / Polymorphismen	dbSNP	[13]
Allgemeine DB / Genom	Ensembl	[23]
Allgemeine DB / Genom	Entrez Gene	[24]
Literatur-DB	PubMed	[25]
Gen-spezifische DB	CFMDB	[26]
Krankheits-DB	OMIM	[3]
Krankheits-DB	HGMD	[27]
Computerprogramm / Splice Site Vorhersage	NNSplice	[28]
Computerprogramm / Alignment	BLAST	[29]
Computerprogramm / Poly(A)-Signal Vorhersage	polyadq	[30]

Tabelle 1: Beispielhafte Zusammenstellung von Datenbanken und Computerprogrammen, die für die manuelle Charakterisierung einer DNA-Veränderung benutzt werden können. DB = Datenbank

1.4.2 Automatische Bewertung

Der große Erfolg der NGS-Technologie ist ermutigend, jedoch stellt die Analyse der aus Exom- oder sogar kompletter Genomsequenzierungen hervorgehenden Datenmengen Wissenschaftler und Ärzte vor eine große Herausforderung. Die Frage nach einer zeitsparenden Möglichkeit zur Bewertung des Krankheitspotentials von DNA-Veränderungen wird durch die Vielzahl der gefundenen Varianten noch dringlicher [31]. Die Anzahl der in einer kompletten Genomsequenzierung gefundenen DNA-Veränderungen beläuft sich auf etwa drei bis vier Millionen [32][33][34]; in unseren eigenen Studien werden pro Exom etwa 50.000 bis

100.000 Varianten gefunden (unveröffentlichte Daten). Zunächst ist nichts über diese Varianten bekannt. Sie können bereits bekannt oder auch neu sein, sie können harmlos oder krankheitsverursachend sein. Während sich die erste Frage mit Hilfe von dbSNP klären lässt (die aktuelle dbSNP Version 137 enthält mehr als 50 Millionen Einträge zu verschiedenen DNA-Varianten [35]), ist eine Antwort auf die zweite Frage deutlich schwieriger. Natürlich ist es nicht sinnvoll, 50.000 bis 100.000 Varianten experimentell auf ihr Krankheitspotential hin zu untersuchen, auch eine manuelle *in silico* Evaluation ist für so eine große Zahl von Varianten nicht denkbar. Die Zahl der interessanten Veränderungen muss also mit Hilfe automatischer Filter- und Priorisierungsstrategien verringert werden.
Es gibt Computerprogramme, die einige der in 1.4.1 angeführten Fragestellungen aufgreifen, und eine automatische Bewertung des Krankheitspotentials von DNA-Varianten anbieten (z.B. PolyPhen [36], SIFT [37], Panther [38]). Besonders häufig wird die evolutionäre Konservierung von Aminosäuren als Kriterium für eine etwaige Pathogenität einer DNA-Veränderung herangezogen [39], aber auch die Betrachtung struktureller und funktioneller Aspekte eines Proteins kann sinnvoll sein [40][41].

1.4.3 Hintergrund und Entstehung von MutationTaster

MutationTaster ist ein web-basiertes Computerprogramm zur automatischen Bewertung des krankheitsverursachenden Potentials von DNA-Sequenzveränderungen.
Zwar gab es zum Zeitpunkt der Entstehung der ersten Version von MutationTaster im Jahre 2008 bereits Computerprogramme zur automatischen Bewertung des Krankheitspotentials von Sequenzveränderungen. Allerdings benutzten all diese Programme ausschließlich die Aminosäuresequenz, um den Einfluss von DNA-Veränderungen auf die Proteinfunktion vorherzusagen (z.B. PolyPhen [36], SIFT [37], und andere [42]-[45]). Effekte auf DNA-Ebene, wie beispielsweise eine Reduktion der mRNA-Stabilität und -Menge oder Entstehung und Verlust von Spleißstellen wurden nicht berücksichtigt. Außerdem konnten die existierenden Programme nur einfache Aminosäureaustausche und keine Insertionen oder Deletionen verarbeiten. Von einigen Programmen wurden Download-Versionen angeboten, die nach einer (nicht immer einfachen) lokalen Installation auch eine hohe Anzahl von Varianten in annehmbarer Geschwindigkeit analysieren konnten. Die für die Nutzung über das Internet zur Verfügung stehenden Versionen waren allerdings für die Analyse größerer Datenmengen ungeeignet, da sie viel zu langsam waren.
MutationTaster muss nicht heruntergeladen werden. Die aktuell verfügbare und in dieser Arbeit vorgestellte Version von MutationTaster bietet die Möglichkeit, DNA-Sequenzveränderungen einzeln oder aber im *Batch Modus** zu analysieren. Die Vorhersage basiert auf einer Reihe von durchgeführten Tests, die sich mit den direkten und indirekten Auswirkungen der DNA-Mutation auf das kodierte Protein beschäftigen. So wird beispielsweise überprüft, ob die Veränderungen in der DNA-Sequenz überhaupt zu einer

Veränderung in der Aminosäuresequenz führt, und falls ja, ob die betroffene Aminosäure Teil einer funktionellen Domäne des Proteins ist. Auch der Konservierungsgrad der betroffenen DNA-Base(n) sowie Aminosäure(n) wird untersucht. Indirekte Auswirkungen auf das Protein, etwa durch fehlerhaftes Spleißen, werden ebenfalls untersucht. Die Summe aller Testergebnisse benutzt MutationTaster, um eine Entscheidung über das krankheitsverursachende Potential der fraglichen DNA-Mutation zu treffen. Diese Entscheidung kann entweder *polymorphism*, d.h. wahrscheinlich harmlos, oder *disease causing*, d.h. wahrscheinlich schädlich, lauten.
Mittlerweile haben die Entwickler zwei der am häufigsten genutzten Programme, SIFT und PolyPhen, neue Versionen veröffentlicht [46][47], die nun ebenfalls den gestiegenen Leistungsanforderungen des NGS-Zeitalters genügen und zum Teil auch mit erweiterter Funktionalität aufwarten: Beide bieten jetzt einen Eingabemodus auf DNA-Ebene und im *Batch* Modus an und SIFT wurde um das Programm PROVEAN [48] ergänzt, welches zusätzlich zu Punktmutationen auch Insertionen / Deletionen, und InDels, die das Leseraster nicht verändern, analysieren kann.

1.4.4 Die erste Version von MutationTaster

MutationTaster entstand aus der Idee, eine Art Leitfaden zu entwickeln, der die einzelnen Arbeitsschritte zur Bewertung des krankheitsverursachenden Potentials von DNA-Varianten vereinheitlichen sollte. Es galt, eine Vorlage zu entwickeln, die alle nötigen Schritte zur Mutationsanalyse in sinnvoller Reihenfolge enthielt, um sicherzugehen, dass keine Krankheitsmutation „übersehen“ wird, und um zu verhindern, dass wertvolle Arbeitszeit auf die genauere Analyse von bereits bekannten, harmlosen Polymorphismen verwendet wird.
Der in einem Praktikum entwickelte erste Algorithmus von MutationTaster war ein Entscheidungsbaum, durch den der Benutzer Schritt für Schritt anhand einfacher, sich automatisch nacheinander öffnender *HTML-Seiten** geführt wurde. Für jede einzelne Station (Entscheidungspunkt) des Entscheidungsbaumes wurde eine eigene HTML-Seite erstellt. Die Entscheidungspunkte waren entweder Fragen, die mit ja oder nein beantwortet werden konnten (Entscheidungsknoten, z.B. *Ist die DNA-Variante in dbSNP verzeichnet?* oder *Führt der DNA-Austausch auch zu einem Aminosäure-Austausch?*), oder Fragen, deren Antwort ein Zahlenwert war (Evaluierungsknoten, z.B. *Was ist der Grad der Konservierung der ausgetauschten Aminosäure?*). Bei Entscheidungsknoten wurde abhängig von der Antwort eine neue HTML-Seite mit der nächsten logischen Frage geöffnet, bei Evaluierungsknoten erschien die nächste Frage unabhängig vom eingetragenen Zahlenwert. Die Benutzer mussten zur Beantwortung der einzelnen Fragen zum Teil eigenständig im Internet die entsprechenden Datenbanken oder Programme konsultieren und gegebenenfalls mit Hilfe einer eigens für diesen Zweck verfassten Anleitung einen Zahlenwert für die Antwort ermitteln. Am Ende wurde eine Vorhersage zum Krankheitspotential der untersuchten DNA-Variante angezeigt,

welche auf den gemachten Antworten und den miteinander verrechneten eingegebenen Zahlenwerten basierte. Diese Version war semi-automatisch. Da wir im Praktikum mit einem sehr kleinen Testset (25 harmlose und 25 krankheitsverursachende DNA-Varianten) die generelle Funktionalität der Methode gezeigt zeigen konnten, beschloss ich, das Thema weiter zu verfolgen und widmete ihm meine Masterarbeit. In dieser sollte das zugrunde liegende Programm stärker automatisiert und die Performanz an einem deutlich größeren Testset geprüft werden. Im Internet verfügbare Datenbanken oder Analyseprogramme musste der Benutzer bislang selbstständig aufrufen, und Ergebnisse seiner Recherche von Hand in die HTML-Seite einfügen. Dies war relativ zeitaufwendig, weshalb wir planten, das Abfragen benötigter Daten aus unterschiedlichen Quellen und deren anschließende Analyse automatisch im Hintergrund zu erledigen. Der Benutzer sollte eine Eingabemaske vorfinden, in die er die zu untersuchende DNA-Variante eingeben konnte, um nach einem Klick auf einen Abschicken-Schaltknopf anschließend direkt die Vorhersage zu sehen. In meiner Masterarbeit konnte ich ein Programm präsentieren, welches mit mehreren hundert Krankheitsmutationen und häufigen Polymorphismen getestet worden war, und das über eine einfache Eingabemaske bedient werden konnte.

1.4.5 Weiterentwicklung von MutationTaster und Ziel dieser Arbeit

Im Gespräch mit Ärzten und Wissenschaftlern wurde ein größeres Interesse an MutationTaster deutlich. Allerdings waren trotz der bereits gemachten Fortschritte weiterhin maßgebliche Verbesserungen nötig. Die größte Herausforderung bestand darin, das bisherige Punkte-System zur Vorhersage des Krankheitspotentials zu verbessern. Die einzelnen Testergebnisse flossen gewichtet ein, was die Optimierung schon zuvor erschwert und verkompliziert hatte. Deshalb sollte das Punktesystem komplett abgeschafft und die Vorhersage künftig einem Bayes Klassifikator überlassen werden. Ein Bayes Klassifikator ist eine bekannte Methode, die gute Klassifizierungsergebnisse liefern kann und nicht aufwendig optimiert werden muss, da aus den vorhandenen Daten automatisch das beste Modell gebildet wird. Voraussetzung ist allerdings eine genügend große Menge bekannter Polymorphismen und Krankheitsmutationen, da der Klassifikator zunächst trainiert und später mit anderen Testfällen validiert werden muss. Auch die Benutzerfreundlichkeit sollte gesteigert werden, da viele Kollegen die Eingabe über die Position als sehr umständlich empfanden. Außerdem sollte die Möglichkeit geschaffen werden, große Datenmengen, wie sie bei NGS-Projekten entstehen, in einer möglichst kurzen Zeit zu analysieren. Speziell für die Analyse von NGS-Daten und den darin zahlreich vorhandenen intronischen oder synonymen DNA-Veränderungen waren bislang allerdings nur wenige Tests vorhanden. Aus diesen Überlegungen ergaben sich folgende Arbeitsschritte für die Weiterentwicklung von MutationTaster:

- Erweiterung der Testpalette insbesondere für stille Mutationen
- automatische Integration aller Ergebnisse mit Hilfe eines Bayes Klassifikators zur Vorher-

sage des Krankheitspotentials, dazu im Vorfeld Erstellung geeigneter, genügend großer Datenreihen mit bekannten harmlosen Polymorphismen und krankheitsverursachenden Mutationen

- Verwendung dieser Daten zum Training des Klassifikators
- Verwendung dieser Daten zur (Kreuz-)Validierung des Klassifikators
- Verbesserung der Benutzerfreundlichkeit durch unterschiedliche Eingabe-Modi für die Einzelabfrage
- Entwicklung eines web-basierten *Batch*-Modus für die simultane Analyse von NGS-Ergebnissen

Das Ziel dieser Arbeit war es daher, die bestehende, rudimentäre Version von MutationTaster zu einem benutzerfreundlichen, verlässlichen, schnellen und vielfältig einsetzbaren Computerprogramm zur Vorhersage der Pathogenität von DNA-Varianten zu entwickeln.

2 Datenintegration

Zur Vorhersage des Krankheitspotentials einer unbekannten DNA-Veränderung, benötigt MutationTaster Informationen zum betroffenen Gen und zur DNA-Sequenzveränderung. Dazu muss die Software viele verschiedene Datenquellen nutzen. In diesem Kapitel werde ich beschreiben, welche Datenquellen in MutationTaster integriert worden sind, und wie dies geschah.

2.1 Einleitung zur Informationsintegration

Exom- oder Genomsequenzierungen sind dank des technischen Fortschrittes in den letzten Jahren immer mehr zu einem Standardinstrument der Identifizierung ursächlicher Mutationen von monogenen Erkrankungen geworden. Auch in anderen Bereichen wird die Technik mittlerweile erfolgreich eingesetzt: In der Vergleichenden Genomik (*comparative genomics*) trägt sie beispielsweise zum besseren Verständnis der Evolution von Pathogenitätsmechanismen von Krankheitserregern bei [49][50] und durch *de novo* Sequenzierung genetisch bislang unbekannter Organismen werden wichtige Erkenntnisse zu Modellorganismen gewonnen [51][52][53]. Die Folge dieser Entwicklung ist ein beständiger Anstieg sehr heterogener, wissenschaftlicher Daten in digitaler Form. Sie sind gespeichert und zum Teil öffentlich verfügbar in verschiedenen biomedizinischen Datenbanken oder können über Computerprogramme genutzt werden. Eine Suche im Internet nach einer bestimmten Information endet typischerweise mit vielen verschiedenen geöffneten Browserfenstern, bei denen man leicht den Überblick verliert. Um die vorhandenen Daten effizient nutzen zu können, ist es nötig, sie zusammenzuführen.
Den Vorgang der Zusammenführung von Daten aus unterschiedlichen, oft heterogenen Quellen in eine gemeinsame, einheitliche Datenstruktur nennt man *Informationsintegration* [54].
Grundsätzlich kann man zwei verschiedene Arten der Integration unterscheiden [55]:

- **Materialisierte oder physische Integration:**
  Daten aus verschiedenen Quellen werden in eine gemeinsame Zielstruktur überführt und dauerhaft in einer zentralen Datenbank gespeichert. Die Daten in der ursprünglichen Quelle bleiben erhalten, die lokal materialisierten Daten können für die Anfragebearbeitung abgerufen werden. Die gängigste Form der physischen Integration ist das *Data-Warehouse* (Datenlager, z.B. Ensembl [23]).
- **Virtuelle oder logische Integration:**
  Die Daten verbleiben in ihren ursprünglichen, verschiedenen Quellen und die Integration erfolgt direkt während der Datenabfrage, z.B. über das Internet. Die Daten werden also nur während der Anfragebearbeitung von der ursprünglichen Quelle zum lokalen Integrationssystem transferiert und anschließend über kurz oder lang wieder gelöscht. Bei jeder neuen Anfrage findet eine erneute Integration statt. Dieses Prinzip wird bei

verschiedenen Integrationsformen angewendet, unter anderem vom *Föderierten Informationssystem* (z.B. IBM DiscoveryLink [56]).

Da virtuell integrierte Daten nicht lokal gespeichert werden, ist das Problem der Beschaffung geeigneter Hardware mit genügend Speicherplatz an den Betreiber der ursprünglichen Datenquelle ausgelagert, ebenso die Verantwortlichkeit über Aktualität und Integrität der Daten. Dies kann allerdings auch ein Nachteil sein, wenn der Betreiber der Originaldaten das Datenformat und/oder die Struktur ändert. Derartige Änderungen im Datenformat sind in der Regel nicht vorhersehbar, was zu Problemen bei der Datenabfrage und zu einem Datenausfall führen kann. Auch aus anderen Gründen (Internetverbindung, Hardware Stabilität der externen Quelle) ist die ständige Verfügbarkeit von virtuell integrierten Daten nicht gewährleistet. Ein weiterer Nachteil sind die längeren Antwortzeiten durch die Datenübertragung. Ebenfalls sehr aufwendig gestaltet sich der Vergleich von virtuell integrierten Daten aus zwei verschiedenen Tabellen, da eine sehr große Datenmenge über eine Netzwerkverbindung transportiert und für den Vergleich im Speicher gehalten werden müsste.

Abgesehen davon, dass physisch integrierte Daten in eigener Verantwortung aktualisiert und gewartet werden müssen, bringt das Speichern von Daten in einer eigenen, zentralen Datenbank einige Vorteile mit sich: Durch entsprechende Maßnahmen kann die Stabilität des Speichersystems sichergestellt werden. Die Wartung der Daten kann geplant werden. Daten können besser auf ihre Integrität überprüft werden, als dies der Fall ist, wenn sie in entfernten Quellen gespeichert sind. Außerdem kann die Struktur einer eigenen, zentralen Datenbank den eigenen Bedürfnissen angepasst werden. So garantiert beispielsweise das Abfrageoptimierte *Star-Schema* sehr kurze Antwortzeiten [57]. Dabei werden mehrere Dimensionstabellen sternförmig um eine zentrale Faktentabelle angeordnet. Die Faktentabelle enthält typischerweise viele Zeilen mit Entitäten (die Fakten), die jedoch in dieser Tabelle nicht genauer beschrieben werden (sie enthält also weniger Spalten). Genauer beschrieben werden die Fakten in den Dimensionstabellen. Diese bestehen im Vergleich zu den Faktentabellen aus weniger Einträgen, die aber unter Umständen viele Attribute zur Charakterisierung der Fakten aufweisen können. Fast alle Dimensionstabellen haben eine direkte Verbindung zur Faktentabelle, wodurch einfache, schnelle Abfragen ermöglicht werden.

Auch bei Computerprogrammen bietet die lokale Installation Vorteile gegenüber der dynamischen Echtzeit-Abfrage. Normalerweise geht der Zugriff auf eine lokal installierte Software deutlich schneller vonstatten als die Abfrage von Ressourcen aus dem Internet. Letzteres kann außerdem durch eine instabile Internetverbindung oder die temporäre Abschaltung der Online-Ressource zu Problemen führen.

2.2 Integrierte Daten und Computerprogramme

Der weitaus größte Teil der für MutationTaster nötigen Daten ist physisch integriert, um Abfragezeiten so kurz wie möglich zu halten und größtmögliche Programmstabilität zu

gewährleisten. Neben Datenbanken nutzt MutationTaster auch die Ausgabe von externen, jedoch lokal installierten, Computerprogrammen als Informationsquelle. Im Folgenden werde ich die in MutationTaster integrierten Datenquellen kurz beschreiben. Details zu den Inhalten und Verknüpfungen der einzelnen, in der MutationTaster Datenbank enthaltenen Tabellen, werden in Abbildung 1 auf S. 25 dargestellt.

2.2.1 Datenbanken

ENCODE Projekt http://genome.ucsc.edu/ENCODE/

Das ENCODE (*Enc*yclopedia *O*f *D*NA *E*lements) Projekt [58] wurde im September 2003 vom National Human Genome Research Institute (NHGRI) mit dem Ziel gestartet, alle funktionalen Elemente des menschlichen Genoms zu identifizieren. Die Daten, welche mit Hilfe akademischer, staatlicher und privater Forschungseinrichtungen generiert wurden, können zum Beispiel über den UCSC [59] *Genome Browser* eingesehen werden. Die Ergebnisse des Projekts sind divers und komplex. In ihrer Gesamtheit lassen sie den Schluss zu, dass mit 80,4% der Großteil des humanen Genoms in wenigstens einem Zelltyp auf wenigstens eine Art und Weise an biochemischen RNA- und/oder chromatin-assoziierten Vorgängen involviert ist, entweder indem ein Protein kodiert wird, oder eine Beteiligung an der Genregulation stattfindet [58]. Die Autoren interpretieren diesen Umstand so, dass also mindestens 80% des menschlichen Genoms funktional, und nicht wie zuvor oft gedacht, funktionslos ist. Es wurden verschiedene „Encode Elemente“ definiert, unter anderem in RNA umgeschriebene Elemente, Histonmodifikationen, Chromatin-Status und Transkriptionsfaktorbindungsstellen [58]. MutationTaster integriert über Ensembl (s.u.) bereitgestellte Daten des ENCODE Projekts.

Ensembl http://www.ensembl.org

Ensembl [23] ist ein zentrales, öffentliches, über das Internet zugängliches Datenlager von Genomdaten. Seit der ersten Veröffentlichung im Jahr 2000 hat es sich zu einer viel genutzten und wichtigen Ressource von Genominformationen entwickelt. Das Kernstück ist die Datenbank *Ensembl Genes*, die derzeit Genomdaten von mehr als 70 verschiedenen Spezies umfasst, wobei der Informationsgehalt zwischen den einzelnen Datensätzen variiert und für die am häufigsten abgerufenen Spezies Mensch, Maus, Ratte und Zebrafisch am größten ist [23]. Der Ensembl *Genome Browser* ermöglicht einen einfachen und bequemen Datenzugriff. Üblicherweise werden *gDNA**-, cDNA-, und *CDS**-Sequenzen, Informationen zu Exon-Intron-Struktur, alternativen *Transkripten** und den entsprechenden Aminosäuresequenzen sowie Hyperlinks zu externen Datenquellen (z.B. InterPro [60], UniProt [61] oder Pfam [62]) bereitgestellt. Zu einem Gen werden sämtliche bekannten Transkripte zusammen mit der CDS angegeben. Die Transkripte können aus unterschiedlichen Quellen stammen: der automatischen Ensembl Annotationspipeline für proteinkodierende Gene (Ensembl *genebuild*

pipeline), aus dem HAVANA/Vega Set mit manuell kuratierten Transkripten oder aus dem *consensus CDS* (CCDS) Projekt, welches sich zum Ziel gesetzt hat, einen Kerndatensatz von konsistent und umfassend annotierter Protein-kodierender Regionen zu identifizieren.
Für 19 häufig abgefragte Spezies steht mit *Ensembl Variation* zusätzlich eine Sammlung genetischer Varianten aus unterschiedlichen Quellen (z.B. dbSNP [13]) zur Verfügung. *Ensembl Regulation* enthält Informationen zu regulatorischen Elementen im Genom des Menschen und der Maus und integriert dazu Daten unter anderem aus dem ENCODE Projekt [58] und aus der Transkriptionsfaktordatenbank JASPAR [63].
Neben dem besonders für Einzelabfragen geeigneten Ensembl *Genome Browser* kann Ensembl auf weiteren Wegen genutzt werden. Um große Datenmengen auf einmal abzurufen kann die Ensembl MySQL Datenbank unter anderem über eine Ensembl Perl *API** (*a*pplication *p*rogramming *i*nterface; Programmierschnittstelle) oder über einen MySQL *Client** abgerufen werden. Die Datenbankinhalte sind außerdem auf einem *FTP** *Server** zugänglich. Sowohl kleinere als auch große Datenmengen sind außerdem über den Webservice BioMart [64] zugänglich. BioMart kann entweder über Schaltflächen im BioMart *web interface** bedient, oder durch eine XML-Syntax direkt aus einem anderen Computerprogramm abgefragt werden. Der Benutzer kann durch Auswählen diverser Attribute und Filter sehr gezielt die gewünschten Daten in verschiedene Dateiformate (z.B. CSV oder TSV) exportieren lassen.
MutationTaster integriert die Datenbanken *Ensembl Genes* und *Ensembl Regulation*. Für die Integration von Ensembl Daten in MutationTaster wurden Daten sowohl über den FTP Server als auch über BioMart bezogen. Letzteres wird dynamisch abgefragt, um DNA- und Proteinsequenzen, die bislang noch nicht in der MutationTaster Datenbank gespeichert sind, zu beziehen. Aus Speicherplatzgründen werden die Sequenzen nicht bereits im Voraus in die Datenbank geschrieben, sondern erst nach einer initialen Nutzung durch MutationTaster. Durch *Ensembl Regulation* hat MutationTaster Zugriff auf derzeit 677 verschiedene Regulationselemente aus den folgenden zwölf Klassen: *Histone*, *Polymerase*, *Segmentation State*, *Enhancer*, *Association Locus*, *Search Region*, *Regulatory Feature*, *Regulatory Motif*, *Pseudo*, *Open Chromatin*, *Transcription Factor Complex* und *Transcription Factor*.

Das 1000-Genom-Projekt

http://www.1000genomes.org/

Das Ziel des 1000-Genom-Projekts (*1000 Genomes Project*) ist es, möglichst viele der „häufigen“ genetischen Varianten zu identifizieren, die mit einer Frequenz von mindestens 1% in der Bevölkerung vorkommen [65]. Im Juni 2009 konnte das Pilotprojekt abgeschlossen werden, 2010 wurden erste Ergebnisse publiziert [66]. Das Pilotprojekt war in mehrere Phasen unterteilt, in denen unterschiedliche Strategien verfolgt wurden: (1) komplette Genomsequenzierung mit niedriger Abdeckung von 176 Proben (*pilot 1 - low coverage*); (2) komplette Genomsequenzierung mit 20 bis 60-facher Abdeckung von zwei Mutter-Vater-Kind Trios (*pilot 2 - trios*; (3) Sequenzierung von 1000 Gen-Regionen in 697 Proben mit 50-facher Ab-

deckung (*pilot 3 - gene regions*).
Für das Hauptprojekt wurden 1.092 Proben aus 14 Populationen sequenziert [32]. Dabei wurden die zuvor in der Pilot-Phase etablierten Methoden angewendet, nämlich komplette Genomsequenzierungen mit niedriger Abdeckung (4x) in Kombination mit höher abgedeckten Exomsequenzierungen. Insgesamt konnten so vermutlich etwa 98% der SNPs, die mit einer Frequenz von 1% in einer Bevölkerung vorkommen, katalogisiert werden [32]. Die bislang verfügbaren Daten sind im *VCF-Format** (*V*ariant *C*all *F*ormat) [67] frei zugänglich. Das VCF-Format bietet Informationen zu den einzelnen Allelen einer Variante, Allelfrequenzen, den absoluten Zahlen der Allele und den Genotypen. In MutationTaster wurden Daten aus dem 1000-Genom-Projekt verwendet, um den Klassifikator zu trainieren, der die Vorhersage generiert. Außerdem werden unbekannte, zu analysierende Varianten immer mit den Daten des 1000-Genom-Projekts verglichen, um so die Anfrage-Variante gegebenenfalls sofort als harmlosen Polymorphismus zu identifizieren.

dbSNP und ClinVar www.ncbi.nlm.nih.gov/snp und www.ncbi.nlm.nih.gov/clinvar/

dbSNP (NCBI Short Genetic Variations (SNV) database) [13][14] ist eine vom NCBI (*N*ational *C*enter for *B*iotechnology *I*nformation) gegründete, frei zugängliche Datenbank für Nukleotidvarianten. Der Verbund aus vielen Spezies-spezifischen Einzeldatenbanken enthält in der aktuellen Version 137 insgesamt mehr als 130 Millionen [68] Einzelbasenaustausche, InDels (*deletion insertion polymorphisms*, DIPs) und *Mikrosatelliten** sowie mehr als eine Billion Genotypen aus HapMap und anderen großangelegten Genotypisierungsprojekten. Die einzelnen Einträge werden, je nach Datenlage, mit Zusatzinformationen abgespeichert. Zusätzliche Attribute sind beispielsweise die Allelfrequenzen und -zahlen, die Allelherkunft (unbekannt, Körper- oder Keimzelle, beides), die Allelfrequenzen, die klinische Relevanz sowie der Validierungsstatus der Variante (z.B. „suspekt“). Obwohl in dbSNP prinzipiell jede „genetische Variation“ gespeichert wird, ohne dass bestimmte Allelfrequenzen als Voraussetzung verlangt werden, findet sich für viele Einträge ein Hinweis zur *minor allele frequency** (MAF): Sie bezeichnet die Frequenz des zweithäufigsten Allels in einer Bevölkerung (welches, wenn es nur zwei Allele gibt, das *minor*, also das seltenere Allel ist). Als „Referenzbevölkerung“ wird unter anderem auf das 1000-Genom-Projekt [32] zurückgegriffen, das Daten zu 1.092 Individuen aus der ganzen Welt bereitstellt. Die Angabe einer MAF erleichtert die Unterscheidung zwischen häufigen, meist harmlosen und seltenen, eventuell krankheitsverursachenden Varianten. Das Speichern von letzteren ist in dbSNP erstmals seit 2008 explizit gewollt [16]. Zur Kennzeichnung der klinischen Relevanz von DNA-Varianten können diese verschiedenen Kategorien zugeordnet werden, unter anderem *non-pathogenic* (nicht pathogen), *probable-non-pathogenic* (wahrscheinlich nicht pathogen), *probable-pathogenic* (wahrscheinlich pathogen) und *pathogenic* (pathogen).
dbSNP Daten können sowohl über ein *web interface* direkt angeschaut als auch über einen

FTP Server als Textdateien heruntergeladen werden. Alle gespeicherten Varianten mit Kenntnis über deren klinische Relevanz (in ihrer Gesamtheit als NCBI *ClinVar*, von *Clinical Variation* bezeichnet) stehen als gesonderte VCF-Datei zum Download bereit. Das *web interface* für ClinVar befindet sich derzeit im Aufbau, der Zugang zu einzelnen ClinVar-Einträgen erfolgt deshalb über die Suchmaske von dbSNP. In MutationTaster werden normale dbSNP Einträge und ClinVar-Einträge (also Krankheitsmutationen aus dbSNP) gesondert behandelt.

Entrez Gene http://www.ncbi.nlm.nih.gov/gene

Entrez Gene ist eine vom NCBI betriebene Genom-Datenbank, die Inhalte aus verschiedenen Einzeldatenbanken zu einer umfassenden, bequem abzufragenden Informationsquelle verknüpft. Bei einer über das Entrez Gene *web interface* gestarteten Suche nach einem bestimmten Gen erhält man die gesammelten Informationen aus verschiedenen Datenbanken, z.B. dbSNP [13], OMIM [3], PubMed [69] oder GO [70]. Ein FTP Server stellt alle in Entrez Gene enthaltenen Datensätze als *flat files** (z.B. reines Textformat) bereit. Für viele der in MutationTaster genutzten Ensembl-Transkripte kann eine Verknüpfung zu einem entsprechenden Entrez Gene Transkript erfolgen. Sofern es eine eindeutige Verknüpfung gibt, wird ein *Hyperlink** zu Entrez Gene auf der MutationTaster Ergebnisseite angezeigt und das Entrez Gene Transkript kann auch für eine MutationTaster Einzelabfrage über das MutationTaster *web interface* genutzt werden. MutationTaster nutzt Entrez Gene außerdem, um auf Ensembl basierende Anfragepositionen mit dem entsprechenden Chromosom und ein Ensembl Gen mit dem entsprechenden HGNC (*H*UGO *G*ene *N*omenclature *C*ommittee) Gensymbol und Proteindomänen zu verknüpfen. Allerdings kann die Verknüpfung zwischen Ensembl und Entrez nicht immer (eindeutig) hergestellt werden, manchmal gibt es für einen Ensembl Eintrag gar keinen oder mehrere entsprechende Entrez Einträge. In erster Linie erfolgt die Zuordnung von Ensembl zu Entrez über eine von Entrez bereitgestellte Verknüpfung zu einer Ensembl ID. Falls diese nicht zur Verfügung steht, wird geprüft, ob das HGNC Symbol in Entrez und Ensembl übereinstimmt, und gegebenenfalls eine Verknüpfung angelegt. Die von Ensembl bereitgestellte Verknüfung zu Entrez IDs wird nicht verwendet, da diese auf Sequenzhomologie basiert und nicht immer verlässlich ist.

HapMap http://hapmap.ncbi.nlm.nih.gov/

Das Internationale HapMap Projekt [71] wurde im Jahr 2002 mit dem Ziel gegründet, chromosomale Regionen aufzudecken, in denen unterschiedliche Personen die gleichen genetischen Varianten (*Haplotyp** Blöcke) aufweisen. Die Kombination mehrerer, gemeinsam vererbter Allele auf demselben Chromosom wird als Haplotyp bezeichnet. Hierzu wurde die DNA von 1.184 Individuen aus elf verschiedenen Populationen genotypisiert und die Genotypen und Allelfrequenzen katalogisiert. HapMap enthält mehr als eine Million SNPs und mehr als 800

CNVs.
Die Daten aus dem HapMap Projekt sind über eine online Suchmaske sowie über HapMart, eine Anwendung basierend auf der BioMart [64] Schnittstelle, verfügbar. Ähnlich wie bei BioMart können verschiedene Filtereinstellungen vorgenommen und unterschiedliche Ausgabeformate gewählt werden. Komplette Datensätze (z.B. alle verfügbaren Genotypen in einer Datei) können ebenfalls im Textformat heruntergeladen werden. MutationTaster integriert HapMap SNPs mit den entsprechenden Genotyphäufigkeiten.

Grantham Matrix
Die in der Grantham Substitutionsmatrix [72] enthaltenen Werte stellen die biochemische Distanz zwischen zwei beliebigen Aminosäuren dar. Die Matrix kann benutzt werden, um zu bewerten wie ähnlich oder unähnlich sich zwei Aminosäuren sind. Grantham konnte belegen, dass die von ihm genutzten Kriterien zur Erstellung der Matrix, nämlich Komposition, Polarität und Molekulargewicht der jeweiligen Aminosäure, in ihrer Gesamtheit besser mit tatsächlich beobachtbaren Aminosäureaustauschfrequenzen korrelieren als die bis dato oft herangezogenen Basenaustauschraten in den zugrunde liegenden Codons [72]. Viele Jahre später wurde die BLOSUM (BLOcks SUbstitution Matrix) [73] entwickelt, eine Substitutionsmatrix, welche die Wahrscheinlichkeiten für alle möglichen Aminosäureaustausche widerspiegelt. In die BLOSUM fließen jedoch nicht primär die physiko-chemischen Eigenschaften der verschiedenen Aminosäuren ein, sondern die tatsächlich beobachteten relativen Aminosäurefrequenzen und deren Austauschraten.
Die Grantham-Werte reichen von null bis 215, je höher der Wert, desto unähnlicher sind sich die alte und neue Aminosäure. Deletionen oder Insertionen sind in der Grantham-Matrix nicht enthalten. An den Klassifikator werden die Grantham Werte nicht übergeben, weil sich in unseren Tests gezeigt hat, dass sie ein schlechtes Merkmal zur Unterscheidung von Krankheitsmutationen und Polymorphismen sind. Statt des Grantham Wertes wird dem Klassifikator nur der Aminosäureaustausch an sich, ohne eine Bewertung des Schweregrades, übermittelt. In dem Ergebnis von einer MutationTaster Analyse wird für DNA-Varianten, die einen Aminosäureaustausch nach sich ziehen, der entsprechende Grantham Wert nur zu Informationszwecken angezeigt. Dies erleichtert dem Benutzer die Einschätzung darüber, wie ähnlich oder unähnlich sich die ausgetauschten Aminosäuren sind.

phyloP und phastCons
phyloP und phastCons sind Computerprogramme, zur Bestimmung der evolutionären Konservierung eines gegebenen Nukleotids. Als Bewertungsgrundlage liegt beiden ein multiples Alignment von DNA-Sequenzen zugrunde. phastCons basiert auf einem *Hidden Markov Model** (HMM), das abschätzt, mit welcher Wahrscheinlichkeit ein bestimmtes Nukleotid zu einem konservierten Element gehört [74]. Dazu betrachtet phastCons nicht nur das multiple

Alignment an der zu analysierenden Position, sondern auch die benachbarten Positionen. Das Ergebnis ist ein Wert zwischen 0 und 1, wobei 1 die größte Wahrscheinlichkeit repräsentiert, dass die fragliche Position unter negativer Selektion steht, also konserviert ist.
phyloP [75] (*phylogenetic P-value*) ist eine Kombination aus vier verschiedenen statistischen phylogenetischen Tests zur Detektion von Sequenzen, die sich entweder langsamer oder schneller entwickeln, als es unter der Nullhypothese einer neutralen Evolution zu erwarten wäre. Die absoluten und in MutationTaster genutzten Werte sind der negative dekadische Logarithmus der P-Werte. phyloP kann also nicht nur Konservierung messen (d.h. langsamere Evolution als erwartet) sondern auch „Nicht-Konservierung“ bzw. Beschleunigung, d.h. schnellere Evolution als erwartet. Je niedriger der phyloP Wert, desto weniger ist eine Position konserviert und umgekehrt. phyloP betrachtet nur individuelle Spalten des multiplen Alignments, ohne Effekte benachbarter Positionen zu berücksichtigen.
MutationTaster benutzt vorberechnete phyloP und phastCons Werte, die auf einem multiplen Alignment der Genome von 46 Vertebratenspezies basieren und lokal in der Datenbank gespeichert sind.

UniProtKB www.uniprot.org/

Die Proteindatenbank UniProtKB [76] ist eine gemeinsame Arbeit des European Bioinformatics Institute (EBI), des Swiss Institute of Bioinformatics (SIB) und der Protein Information Resource (PIR). UniProtKB (UniProt *K*nowledge*b*ase) besteht aus der manuell annotierten und kuratierten Proteindatenbank Swiss-Prot und dem automatisch annotierten Teil TrEMBL und enthält Proteinsequenzen mit Annotationen, Hinweise zu Proteinfunktion, eventueller Krankheitsrelevanz und Publikationen, sowie gegebenenfalls Daten zur Taxonomie, Verknüpfungen zu Ontologien (z.B. GO [70]), Interaktionsdatenbanken (z.B. STRING [77]) oder Strukturdatenbanken (z.B. EMBL-EBI *Protein Data Bank in Europe*, PDBe [78]). Die positionsbasierte Sequenzannotation weist strukturelle oder funktionelle Proteindomänen, wie z.B. post-translationale Modifikationen, Bindungsstellen, das aktive Zentrum eines Enzyms oder Regionen mit Sekundärstrukturen aus. Für die Integration in MutationTaster wurden Protein IDs und zugehörige Sequenzannotationen aus einer vom UniProtKB FTP Server heruntergeladenen Textdatei extrahiert. Die Verknüpfung von Ensembl Genen zu den jeweiligen Protein IDs erfolgt über die NCBI Gene ID.

Human Gene Mutation Database http://www.hgmd.cf.ac.uk/ac/index.php

Die *H*uman *G*ene *M*utation *D*atabase (HGMD) [27] enthält eine nicht-redundante Sammlung von DNA-Mutationen, die für genetische Erkrankungen verantwortlich sind, sowie krankheitsassoziierter oder funktioneller Polymorphismen.
Von Einzelbasenaustauschen in kodierenden, regulatorischen oder Spleißregionen über kleinere und größere InDels bis hin zu komplexen Aberrationen sind in HGMD verschiedene Arten von

Varianten vertreten. Stille Mutationen in der kodierenden Region werden nur gespeichert, wenn gezeigt wurde, dass sie das Spleißen stören, oder eine anderweitige direkte Assoziation mit einer Krankheit besteht. Somatische sowie mitochondriale Mutationen sind derzeit nicht enthalten.
HGMD enthält nur solche Mutationen, deren Krankheitsrelevanz durch die entsprechende Publikation oder eine persönliche Rücksprache mit deren Autoren eindeutig belegt ist [27]. Als Evidenz für eine krankheitsverursachende Mutation wird unter anderem die Erfüllung eines oder mehrerer der folgenden Kriterien erwartet: Lokalisierung der Mutation in einer bekannten strukturellen oder funktionellen Proteindomäne, Lokalisation der Mutation in einer evolutionär hochkonservierten Region, *in vitro* Demonstration von reduzierter Genexpression / Stabilität oder Aktivität des Proteinproduktes / aberrantem Spleißing durch die Mutation, *in vitro* Nachweis der gleichen aberranten Proteineigenschaften wie sie *in vivo* beobachtet werden oder *rescue* (Rettung oder Umkehr) des pathologischen Phänotyps durch Reparatur oder Transfektion des Krankheitsgens *in vitro*.
Trotz sorgfältiger Kuration der Mutationen muss man dennoch davon ausgehen, dass einige der in HGMD gespeicherten Krankheitsmutationen in Wahrheit nicht die Ursache für eine genetische Erkrankung sind [79]. Mit Hilfe der Daten aus dem 1000-Genom-Projekt wurden Ende 2012 33 DNA-Varianten wegen unzureichender Krankheitsevidenz aus HGMD entfernt und viele weitere re-klassifziert (siehe auch Kapitel 8.1.1).
Die Datenaquise erfolgt mittels einer Kombination aus automatischer und manueller Literaturrecherche in PubMed [69] und Lokus-spezifischen Datenbanken (LSDBs).
Jeder Eintrag in HGMD wird mit einem der folgenden vier *tags* (Anhängsel, in der Informatik: Kennzeichnung eines Datensatzes) versehen:

- **DM** (*disease-causing mutations*)**:** Mutationen, die in der Literatur als krankheitsverursachend beschrieben wurden
- **DP** (*disease-associated polymorphisms*)**:** Polymorphismen, denen eine signifikante ($p<0.5$) Assoziation mit bestimmten Krankheiten nachgewiesen werden konnte, die jedoch noch nicht auf ihre Funktion in einer Krankheit untersucht worden sind
- **DFP** (*disease-associated polymorphisms with additional supporting functional evidence*)**:** Zusätzlich zur statistisch signifikanten Krankheitsassoziation ($p<0.5$) wurden funktionelle Untersuchungen angestellt, deren Ergebnisse die Krankheitsassoziation stützen
- **FP** (*in vitro/laboratory or in vivo functional polymorphisms*)**:** Beeinflussen nachgewiesenermaßen die Genstruktur, -funktion oder -expression, ohne dass jedoch eine Krankheitsassoziation berichtet wurde.

Krankheitsmutationen werden ausschließlich als *DM* gekennzeichnet, die drei anderen *tags* werden für Varianten mit nicht eindeutig belegter Krankheitsevidenz benutzt. HGMD existiert in zwei Formen: die kostenpflichtige, professionelle Version (*HGMD Professional*) en-

thält alle verfügbaren Mutationen; die freie, öffentliche Version (*HGMD Public*) enthält sehr viel weniger und nur ältere Mutationen. Für das Training und die Validierung des MutationTaster Bayes Klassifikators wurden alle geeigneten HGMD *Professional* Einträge mit eindeutig belegter Krankheitsrelevanz und dem HGDM-internen Vermerk *DM* verwendet (siehe auch Kapitel 5.1.2).

2.2.2 Computerprogramme

BLAST / *bl2seq* http://blast.ncbi.nlm.nih.gov/Blast.cgi

MutationTaster benutzt eine lokale Installation von *bl2seq* [80] (*bl*ast *2 seq*uences), um den Status der evolutionären Konservierung von Nukleotiden oder Aminosäuren zu untersuchen. *bl2seq* ist eine spezielle BLAST (*B*asic *L*ocal *A*lignment *S*earch *T*ool) [29] Variante zum Alignieren zweier Sequenzen.
BLAST wird in der Regel verwendet, um in einer kompletten Sequenzdatenbank homologe bzw. ähnliche Sequenzen zu finden. Falls aber zwei bereits bekannte Sequenzen miteinander verglichen werden sollen, bietet *bl2seq* sich als Alternative an. Es nutzt den geschwindigkeitsoptimierten BLAST Algorithmus für ein paarweises Alignment zweier zuvor spezifizierter Sequenzen, ohne dass ein Umweg über die (zeit-)aufwendige Datenbanksuche nötig ist.

NNSplice / Spleißstellen Vorhersage http://www.fruitfly.org/seq_tools/splice.html

NNSplice ist eine Software, die ein neuronales Netz zur Vorhersage von Spleißstellen nutzt. NNSplice wurde bereits im Jahr 1997 publiziert [28], in der Folge jedoch immer wieder mit einer steigenden Anzahl von Sequenzen trainiert und getestet. Zuletzt passierte dies im Jahre 2008 [81] (siehe Tabelle 2):

Typ	Sequenzen Training	Sequenzen Test
Akzeptor echt	1116	208
Donor echt	1116	208
Akzeptor falsch	4672	881
Donor falsch	4140	782

Tabelle 2: Zahl der Sequenzen mit echten und unechten (falschen) Intron-Exon-Grenzen für Training und Validierung der in MutationTaster integrierten Version von NNSplice.

NNSplice sagt neben Konsensus-Spleißstellen (diese müssen am 5' Ende ein GT-Dinukleotid aufweisen und am 3' Ende ein AG-Dinukleotid) auch Spleißstellen vorher, die dieser Regel nicht entsprechen. Das Programm gibt für jede vorhergesagte Spleißstelle einen Wahrscheinlichkeitswert zwischen 0 und 1 aus, je näher der Wert an 1 ist, desto wahrscheinlicher handelt es sich um eine echte Spleißstelle. Sogenannte *exonic splice enhancers** (ESE) und *exonic*

*splice silencers** (ESS) werden nicht erkannt.
MutationTaster benutzt eine lokal installierte Version von NNSplice, um Spleißstellen in der Wildtyp- sowie in der varianten Sequenz vorherzusagen und eventuelle Unterschiede zu detektieren. Die lokale Version der Software bietet mehr Optionen als die Web-Version und die besten Einstellungen für den Einsatz in MutationTaster haben wir mit einem (kleinen) Datensatz von annotierten Intron-Exon-Grenzen ermittelt.

polyadq http://rulai.cshl.org/tools/polyadq/polyadq_form.html
polyadq [30] ist ein Computerprogramm zur Vorhersage von Polyadenylierungssignalen (Poly(A)-Signal, PAS). Von den Autoren der Software wurde für Vorhersagen in den letzten zwei Exons von Genen eine Sensitivität von 64,1% und eine Spezifität von 83,3% angegeben. MutationTaster benutzt eine lokal installierte Version von polyadq, um Wildtyp- und variante DNA-Sequenz auf den eventuellen Verlust des Poly(A)-Signals zu überprüfen, sofern die zu analysierende Variante in der 3'UTR eines Gens lokalisiert ist.

2.3 Datenspeicherung

2.3.1 Einleitung zur Datenspeicherung

Die Verwendung eines Datenbanksystems zum Speichern von Daten bringt einige Vorteile mit sich: (1) Transaktionskontrolle, d.h. dass einzelne Zugriffe auf verschiedene Tabellen, die jedoch zu einem logischen Vorgang gehören, entweder ganz oder gar nicht durchgeführt werden; (2) Geschwindigkeit, weil Datenbanksysteme für die Verwaltung sehr großer Datenmengen optimiert sind und deshalb insbesondere Verknüpfungen verschiedener Tabellen zeiteffizient durchgeführt werden können; (3) Mehrbenutzerfähigkeit, d.h. dass mehrere Benutzer gleichzeitig auf die Datenbank zugreifen und Transkationen vornehmen können, und evenutelle Datenkonflikte vom Datenbanksystem abgefangen werden und (4) standardisierte Schnittstellen zur Datenabfrage.
In einer Datenbank können sich Inhalte verschiedener Tabellen aufeinander beziehen. Ein Eintrag in einer Tabelle sollte sich dabei nur auf Inhalte einer anderen Tabelle beziehen, die tatsächlich existieren und eindeutig sind. Dieser Zustand wird referenzielle Integrität genannt. Um sie zu gewährleisten, werden sogenannte *Fremdschlüssel* angelegt: Ein Fremdschlüssel ist ein Verweis aus einer Tabelle heraus auf ein eindeutiges Tupel (Zeile) in einer anderen Tabelle [82]. Dieses eindeutige Tupel in der anderen Tabelle muss dort in Form eines eindeutigen Schlüssels definiert sein. Dieser kann ein einzelnes Attribut (Spalte) oder eine Kombination aus mehreren Attributen umfassen, deren Wert bzw. Wertekombination für ein Tupel eindeutig ist [83][84]. Zusätzlich zum *Primärschlüssel* können aber auch noch weitere eindeutige Schlüssel (sogenannte *unique keys*) definiert werden. Ein Beispiel soll die Verwendung von Primär- und Fremdschlüsseln erläutern: In einer Firmendatenbank gibt es die zwei

Tabellen *Mitarbeiter* und *Arbeitszeiten*. In der Tabelle *Arbeitszeiten* werden für jeden Mitarbeiter die geleisteten Arbeitsstunden erfasst. Diese Tabelle verweist auf die Mitarbeiter, die in der Tabelle *Mitarbeiter* gespeichert sind. Es könnte nun passieren, dass ein Mitarbeiter kündigt, und deshalb aus der Tabelle *Mitarbeiter* gelöscht wird. In der Tabelle *Arbeitszeiten* sind seine Arbeitszeiten aber nach wie vor erfasst, es entsteht ein Verweis auf einen Eintrag, der in der Tabelle *Mitarbeiter* nicht mehr existiert. Durch das Anlegen eines Fremdschlüssels in der Tabelle *Arbeitszeiten*, der sich auf den Wert des Primärschlüssels in der Tabelle *Mitarbeiter* bezieht, ist das Löschen eines Eintrags aus der Tabelle *Mitarbeiter* nun nicht mehr ohne weiteres möglich - zunächst muss der entsprechende Eintrag aus der Tabelle *Arbeitszeiten* entfernt werden, der auf den Eintrag in der Tabelle *Mitarbeiter* verweist. Erst danach kann der Eintrag in der Tabelle *Mitarbeiter* gelöscht werden. Durch die Verwendung von Primär- und Fremdschlüsseln wird die referenzielle Integrität gewahrt, also sichergestellt, dass Verweise auf andere Tabellen nicht ins Leere oder auf mehrere Tupel zeigen.
Sogenannte *Datenbankindizes* werden verwendet, um einen schnellen Zugriff auf Daten zu ermöglichen. Ein Index ist eine geordnete Liste der Werte eines oder mehrerer Attribute einer Datenbanktabelle. Zu jedem Indexeintrag wird ein Verweis auf das entsprechende Tupel in der Datenbanktabelle gespeichert. Durch die Sortierung des Index' können so sehr schnell die passenden Einträge des Index und folglich der Datenbanktabelle abgerufen werden, ohne dass dafür die ganze Datenbanktabelle durchsucht werden muss.

2.3.2 Datenbankstruktur

Die in MutationTaster physikalisch integrierten Daten sind in einer gemeinsamen Datenbank, jedoch verschiedenen logischen Bereichen (Schemata), gespeichert. Abbildung 1 zeigt die Schemata der Datenbankstruktur. Ein Teil der Daten in der Datenbank wird von mehreren Applikationen genutzt. Dies sind die Daten im Bereich *public* (in der Abbildung die blauen Tabellen ohne Prefix), welche initial für GeneDistiller [85], eine ebenfalls von uns entwickelte Web-Applikation, integriert wurden. Der größte Teil der für MutationTaster benötigten Daten stammt aus *Ensembl Genes* (grüne Tabellen in der Abbildung). Neben diesem Ensembl-Datenbereich gibt es noch den zusätzlichen Bereich *Ensembl Regulations*, dieser enthält Daten zu regulatorischen Elementen im Genom (orangene Tabellen). Der Datenbereich *mute* (gelbe Tabellen) beinhaltet Daten, die ausschließlich von MutationTaster genutzt werden, aber nicht über Ensembl bezogen werden können. Der Bereich *hm* (rote Tabelle in der Abbildung) wird vor allem für das Computerprogramm HomozygosityMapper [86] verwendet, aber MutationTaster greift ebenfalls auf Daten aus einer Tabelle daraus zurück. Die Tabelle 12 im Anhang gibt einen Überblick über die tatsächlichen Inhalte der einzelnen Datenbanktabellen. Obwohl die physikalisch integrierten Daten in verschiedenen logischen Bereichen gespeichert sind, werde ich der Einfachheit halber im Anschluss an dieses Kapitel im Allgemeinen von *Datenbank* sprechen, auch wenn tatsächlich die Rede von einzelnen Datenbankbereichen ist.

2.3.3 Aktualisierung

Die verschiedenen Datenbankbereiche werden in unregelmäßigen Abständen aktualisiert (Update). Wie häufig ein Update tatsächlich durchgeführt wird, hängt davon ab, wie stark sich die benutzten Datenquellen verändern. Das Schema *public* wird im Zuge der GeneDistiller Wartung regelmäßig mehrmals pro Jahr aktualisiert. Die umfangreichste MutationTaster Datenquelle, Ensembl, bietet im Durchschnitt alle drei Monate eine neue Version an. Diese werden jedoch nicht alle in die MutationTaster Datenbank eingespielt, sondern nur jede dritte bis vierte, weil jede Aktualisierung ein neues Training des Bayes Klassifikators erfordert (s.u.). Durch das Speichern der Ensembl Daten in einem separaten Schema und das Beibehalten alter Schemata kann MutationTaster theoretisch nach einer Aktualisierung noch mit älteren Ensembl Versionen laufen. Andere Daten, wie z.B. phyloP und phastCons Werte müssen erst aktualisiert werden, wenn eine neue Genomversion veröffentlicht wird. Aktuell basiert MutationTaster auf der Genomversion GRCh37 (*G*enome *R*eference *C*onsortium *H*uman genome build 37) bzw. hg19 (*h*uman *g*enome version 19).
Generell muss man beim Update zwischen Daten unterschieden, nach deren Aktualisierung der in MutationTaster integrierte Bayes Klassifikator neu trainiert werden muss, und Daten, die dies nicht erforderlich machen. Zur Vorhersage des Krankheitspotentials einer Veränderung nutzt der Klassifikator die Frequenz des Auftretens bestimmter Merkmale in harmlosen Polymorphismen und Krankheitsmutationen (siehe Kapitel 4.2. Durch die bessere Datengrundlage erhöhen sich die Zahlen vieler Merkmale (z.B. der regulatorischen Elemente) laufend, damit steigt auch deren Frequenz im Genom. Dies führt dazu, dass es auch in der Gruppe der Polymorphismen zu einem Anstieg der Zahl der Merkmale kommt. Wird nun eine Analyse auf Grundlage deutlich niedrigerer Frequenzen dieser Merkmale in der Gruppe der Polymorphismen durchgeführt, steigt die Wahrscheinlichkeit, daß der Klassifikator die Veränderung aufgrund der gefundenen Merkmale fälschlich als krankheitsverursachend einstuft. Dies betrifft insbesondere die regulatotischen Elemente sowie die Proteindomänen. Aktualisierungen von der dbSNP oder ClinVar hingegen können jederzeit ohne erneutes Training durchgeführt werden. Diese Informationen werden vom Klassifikator nicht zur Vorhersage benutzt sondern nur zu Informationszwecken auf der Ergebnisseite angezeigt. Zwar gibt es automatische Vorhersagen für Polymorphismen und Krankheitsmutationen basierend auf entsprechenden Informationen in den Datenbank Tabellen *TGP* und *ClinVar*, diese automatischen Vorhersagen werden aber ohne Hilfe des Klassifikators gemacht.
Das Einspielen neuer Daten erfolgt semi-automatisch mit Hilfe eines *Perl-Skriptes**, das jedoch vor und während des Update-Vorgangs unter Umständen an geänderte Datenformate angepasst werden muss. Vorher müssen außerdem die aktuellen Daten als *flat files* heruntergeladen und im richtigen Verzeichnis entpackt werden. Da sich URL, Dateiname und Dateiformat regelmäßig ändern, kann dieser Vorgang leider nicht automatisiert werden.

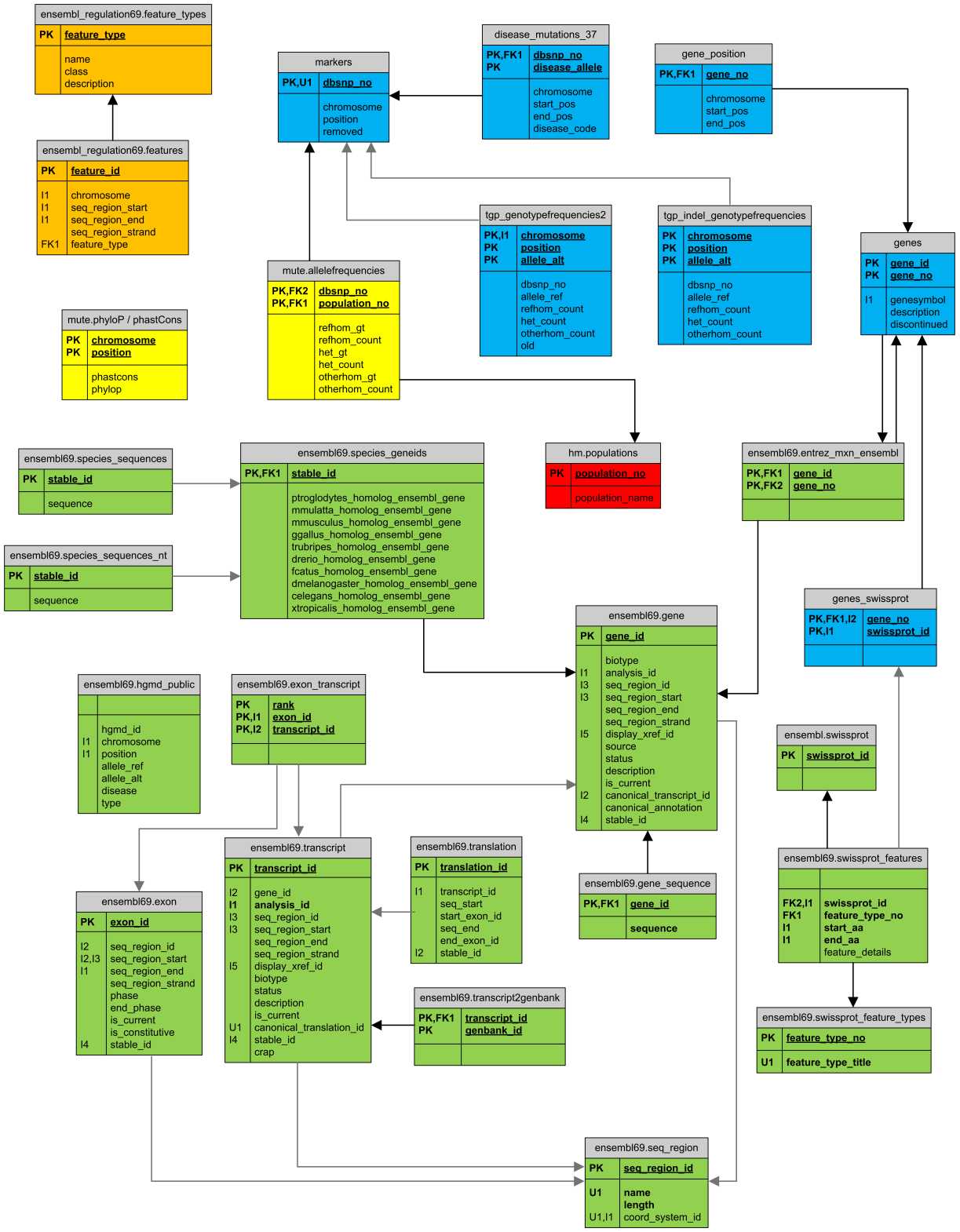


Abbildung 1: Von MutationTaster genutzte Datenbank Tabellen und ihre Verknüpfungen. PK = *primary key* = Primärschlüssel; FK = *foreign key* = Fremdschlüssel; U = *unique key* = eindeutiger Schlüssel; I = Index; schwarzer Pfeil = expliziter Fremdschlüssel; grauer Pfeil = impliziter Fremdschlüssel

3 Das MutationTaster Computerprogramm

In diesem Kapitel werde ich MutationTaster und dessen Benutzung vorstellen. Dazu werde ich zunächst einen kurze technische Einführung geben und anschließend einen typischen MutationTaster Aufruf sowie die Abläufe innerhalb der Software Schritt für Schritt nachvollziehen.

3.1 Technische Erläuterungen

Die Strukturierung der verschiedenen Software-Komponenten des MutationTaster Computerprogramms entspricht der sogenannten 3-Schichten-Architektur (*three tier architecture*).

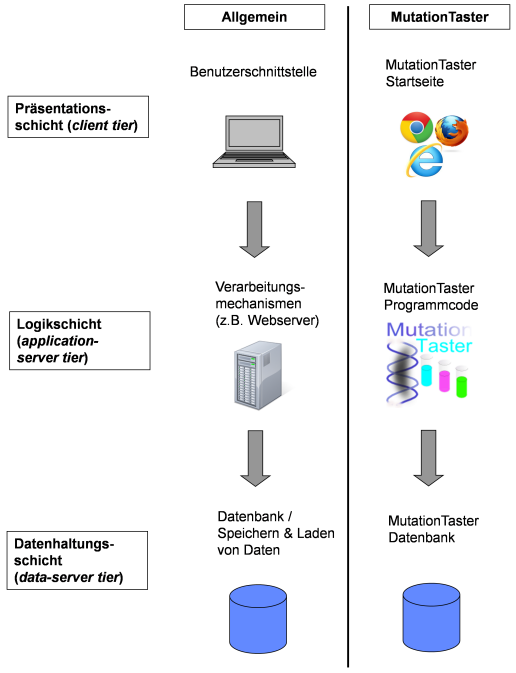


Abbildung 2: 3-Schichten-Architektur und ihre Entsprechung in den einzelnen Programmkomponenten von MutationTaster.

Diese besteht typischerweise aus den folgenden drei Schichten [87]:

1. **Präsentationsschicht**: Diese wird auch als *Front End* bezeichnet und enthält typischerweise die Benutzerschnittstellen. Client-seitige Software ist oft ein Webbrowser.
2. **Logikschicht:** Diese wird auch *application-server tier* oder *middle tier* genannt und beinhaltet Mechanismen zur Verarbeitung der Benutzeranfragen und zur Ergebnisrückgabe.
3. **Datenhaltungsschicht:** Diese auch als *data-server tier* oder *Back End* bezeichnete Schicht ist verantwortlich für das Speichern und Laden von Daten und enthält häufig eine Datenbank.

MutationTaster wurde in der Programmiersprache Perl geschrieben. Ein zentrales *Perl Modul** (MutationTaster.pm) enthält alle Funktionen, die während eines Programmlaufs aufgerufen werden können und greift gegebenenfalls auf die Datenbank zu. In der 3-Schichten-Architektur entspricht dies dem *Back End*. Die Vorteile der 3-Schichten-Architektur liegen für den Benutzer darin, dass er außer einem Webbrowser keine zusätzliche Software benötigt. Für Softwareentwickler bedeutet ein in der 3-Schichten-Architektur angelegtes Computerprogramm in der Regel weniger Wartungsarbeit als dies bei Distributionslösungen der Fall ist, weil sie die volle Kontrolle über alle Daten und alle Funktionen haben und Akutalisierungen nur in der Web-Version der Software vornehmen müssen. Sobald verschiedene Versionen einer Software zum Download bereit stehen, muss man mit einem nicht unerheblichen Zeitaufwand für die Beantwortung von Benutzerfragen

zum Beispiel zu Installationsproblemen oder Softwarefehlern rechnen. Unter anderem aus diesen Gründen steht MutationTaster nicht als lokal zu installierende Download-Version zur Verfügung. Die Startseite von MutationTaster (das *Front End*) wurde im HTML-Format

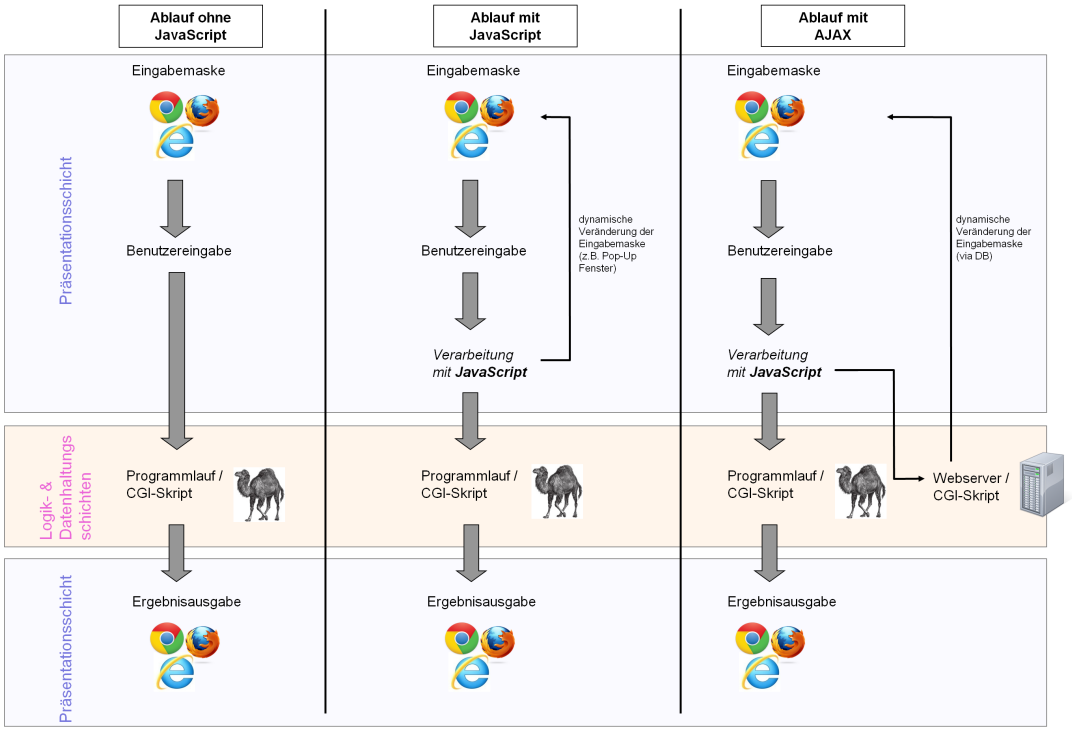


Abbildung 3: Schematische Darstellung der Funktionsweise von JavaScript und AJAX.

angelegt und benutzt JavaScript und *AJAX** (*A*synchronous *J*avaScript *a*nd *X*ML), um Informationen dynamisch einzublenden. Eine AJAX Funktion ruft dazu ein *CGI**-Skript (*C*ommon *G*ateway *I*nterface) auf. CGI ist ein Standardprotokoll, das für den Datenaustausch zwischen einer HTML-Webseite und einer nachgeschalteten Software (in diesem Fall das MutationTaster Perl Modul) benutzt werden kann [88]. Das CGI-Skript, das in der 3-Schichten-Architektur die Logikschicht repräsentiert, liest die bislang gemachten Benutzereingaben aus der HTML-Seite aus und greift auf bestimmte Funktionen des Perl Moduls zu, um die gewünschten Informationen zu erhalten. Diese werden anschließend in die HTML-Seite eingebaut, ohne dass sie neu geladen werden muss. Sobald der Benutzer seine Anfrage abgeschickt hat, wird wieder das *CGI-Skript* aufgerufen. Es liest erneut die Benutzereingaben aus der HTML-Seite aus und ruft dann nacheinander alle für die zu analysierende Variante sinnvollen Funktionen aus dem zentralen Perl Modul auf. Zum Schluss wird eine Funktion aufgerufen, die eine statische HTML-Ergebnisseite erstellt.

3.2 Benutzereingaben

MutationTaster ist über eine einfache HTML-Seite (Abbildung 4) im Internet erreichbar. Die Internetadresse[1] verweist auf die Startseite für Einzelabfragen im Transkript-basierten Modus. Daneben gibt es weitere Möglichkeiten, MutationTaster ohne Angabe eines Transkripts zu benutzen, die ich in Kapitel 6 genauer vorstellen werde.
Benutzereingaben erfolgen über das Ausfüllen von Textfeldern und das Anklicken von Schaltflächen. Folgende Benutzereingaben sind obligatorisch:

- Ensembl Tranksript ID (ENST)
- Sequenztyp (CDS, cDNA oder gDNA)
- Sequenz-Schnipsel mit der in eckigen Klammern spezifizierten Variante (Einzelbasenaustausch oder InDel)

  oder
- Position und neue Base (Einzelbasenaustausch)

  oder
- letzte normale Position vor der Variante und erste normale Position nach der Variante mit fehlenden / zusätzlichen Basen (InDel)

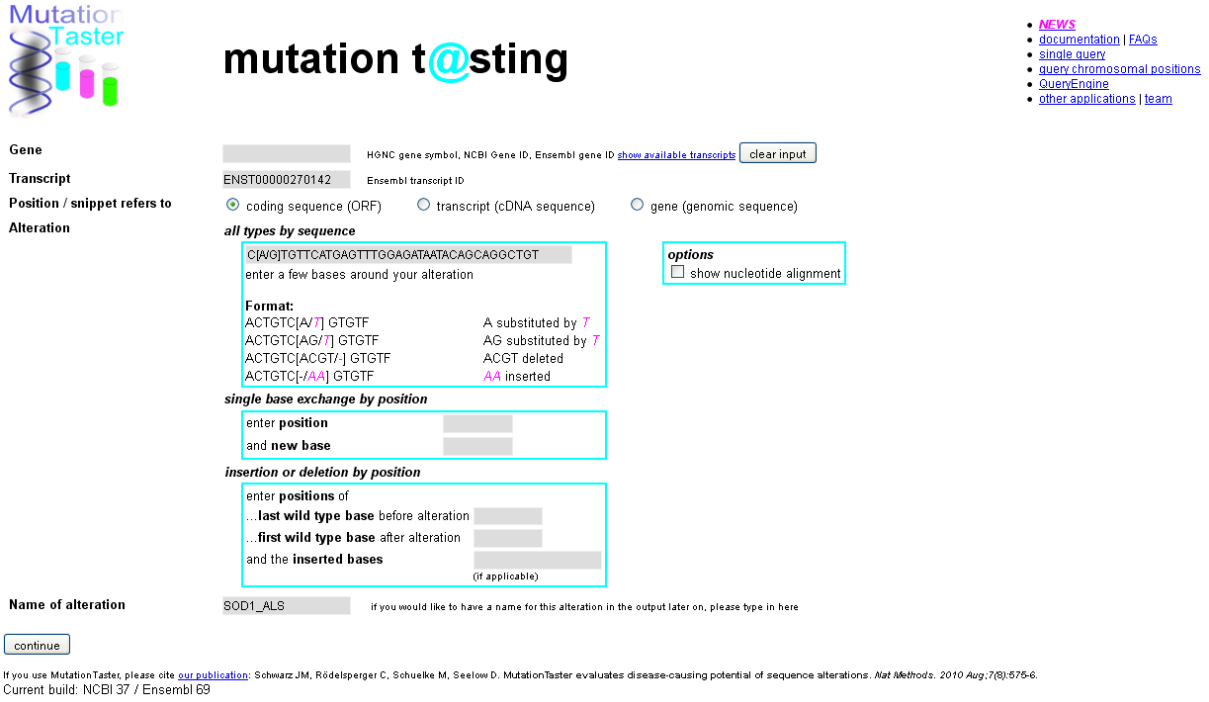


Abbildung 4: MutationTaster Startseite für Transkript-basierte Einzelabfragen.

Die Angabe des Transkripts ist hier zwingend notwendig, da MutationTaster eine DNA-Veränderung in einem bestimmten, dem spezifizierten, Transkript analysiert. Die Angabe

[1] `http://www.mutationtaster.org`

eines Gennamens ist nicht nötig, wenn direkt ein Transkript spezifiziert wird. Falls der Benutzer zunächst ein Gen einträgt, wird nach der Eingabe mittels AJAX eine Liste der verfügbaren Ensembl Transkripte angezeigt, von denen eines ausgewählt werden kann. Nach Auswahl eines Transkripts muss der Benutzer außerdem festlegen, ob sich seine Eingaben auf die CDS, cDNA oder gDNA beziehen sollen - dies ist besonders wichtig, wenn die Eingabe über die Position erfolgt oder wenn eine Variante sich über eine Intron-Exon-Grenze erstreckt (in diesem Fall muss *gDNA* gewählt werden).
Die Variante selbst kann auf verschiedenen Wegen eingegeben werden: Die einfachste Möglichkeit besteht in einem Sequenzschnipsel im „dbSNP Format“, der die Variante in eckigen Klammern enthält. Der Sequenzschnipsel CATGTTCATG[A/G]GTTTGGAGATAATACA enthält zum Beispiel die Information, dass ein A durch ein G ersetzt worden ist. Formatierungshinweise für die Eingabe von Einzelbasenaustauschen, Insertionen, Deletionen und InDels befinden sich auf der Startseite direkt unter dem Textfeld zur Eingabe des Sequenzschnipsels. Diese Art der Eingabe ist komfortabel, weil Sequenzschnipsel im von MutationTaster verlangten Format ohnehin als Ergebnis einer Sanger Sequenzierung vorliegen. Sie ist außerdem auch verlässlich, weil sie unabhängig ist von Positionen, die sich mit verschiedenen Datenbank Versionen durchaus ändern können. Der Sequenzschnipsel wird von MutationTaster mit der DNA-Sequenz des festgelegten Transkripts verglichen und die Position der Variante automatisch berechnet.
Alternativ kann die Variante auch direkt über die Position und das ausgetauschte Nukleotid (oder die ausgetauschten Nukleotide) spezifiziert werden. Wenn das Transkript über eine Schaltfläche ausgewählt wurde, und die Varianteneingabe über das Positionsfeld erfolgt, wird mittels AJAX das veränderte Nukleotid im Sequenzkontext angezeigt, sobald man außerhalb des Positionsfeldes klickt. So kann der Benutzer die von ihm gemachten Eingaben überprüfen.

Weitere Angaben können optional gemacht werden:

- Gen (*HGNC Symbol*, *NCBI Gen ID* oder *Ensembl Gen ID* (ENSG)
- Projektname
- Option zur Anzeige der Konservierung auf der DNA Ebene

MutationTaster überprüft die Benutzereingaben während und nach der Eingabe. Wird zum Beispiel sowohl das Positionsfeld für einen Einzelbasenaustausch als auch das Positionsfeld für InDels ausgefüllt, informiert ein sich automatisch öffnendes Fenster den Benutzer über die nicht eindeutige Eingabe und gibt einen Hinweis zur Korrektur. Andere Sachverhalte werden erst nach dem Abschicken überprüft, z.B. ob überhaupt etwas eingetragen wurde, ob sich Referenz- und alternative Base wirklich unterscheiden oder ob der Sequenzschnipsel ausschließlich die erlaubten Zeichen A, G, T und C enthält. Mit einem Klick auf den *continue*-Schaltknopf wird die Anfrage an MutationTaster abgeschickt, und MutationTaster beginnt,

die Variante zu analysieren. Eine vereinfachte Übersicht über die einzelnen Schritte während eines Programmlaufes zeigen die Abbildungen 5 bis 7.

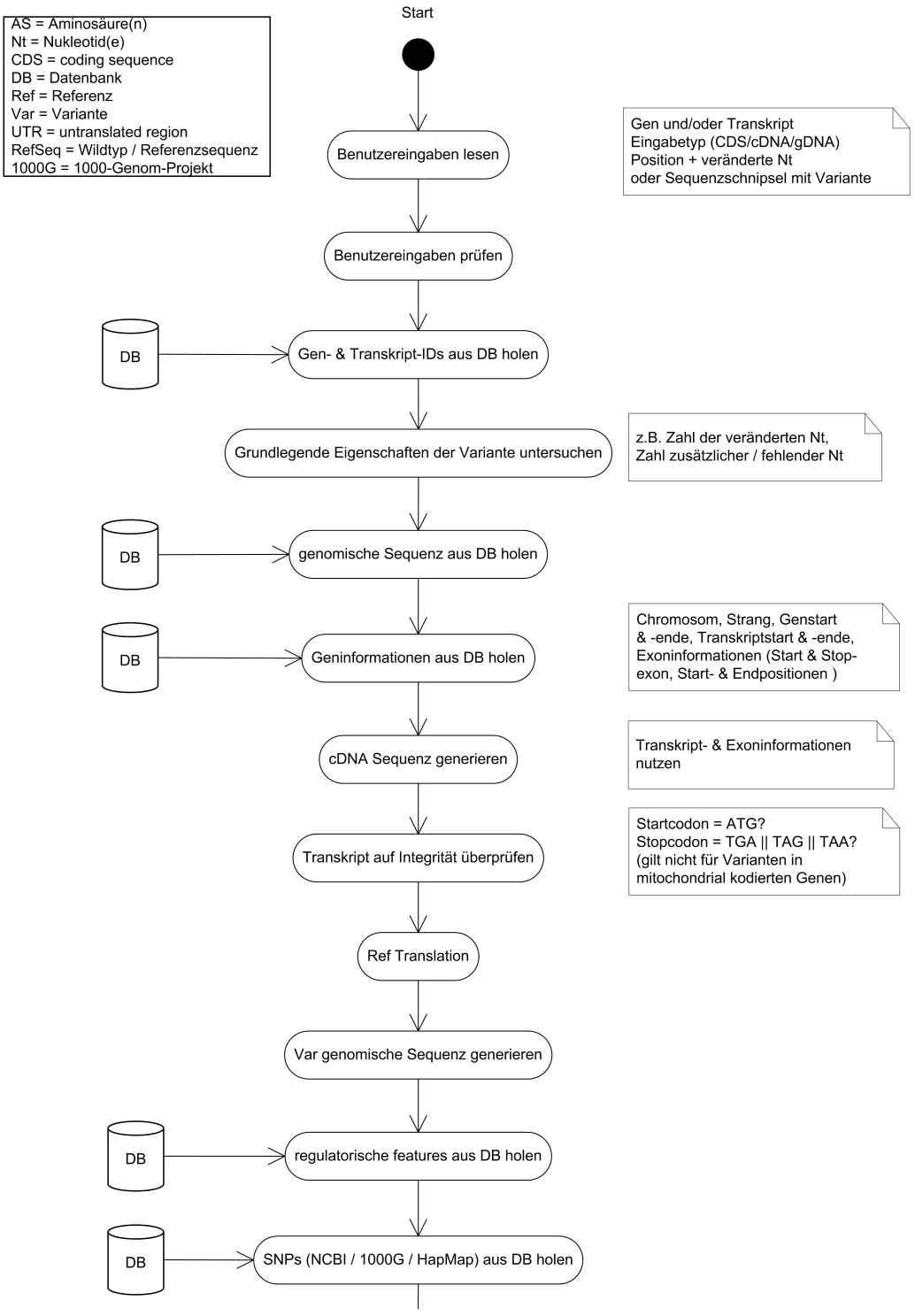


Abbildung 5: Vereinfachter Programmablauf von MutationTaster, Teil 1.

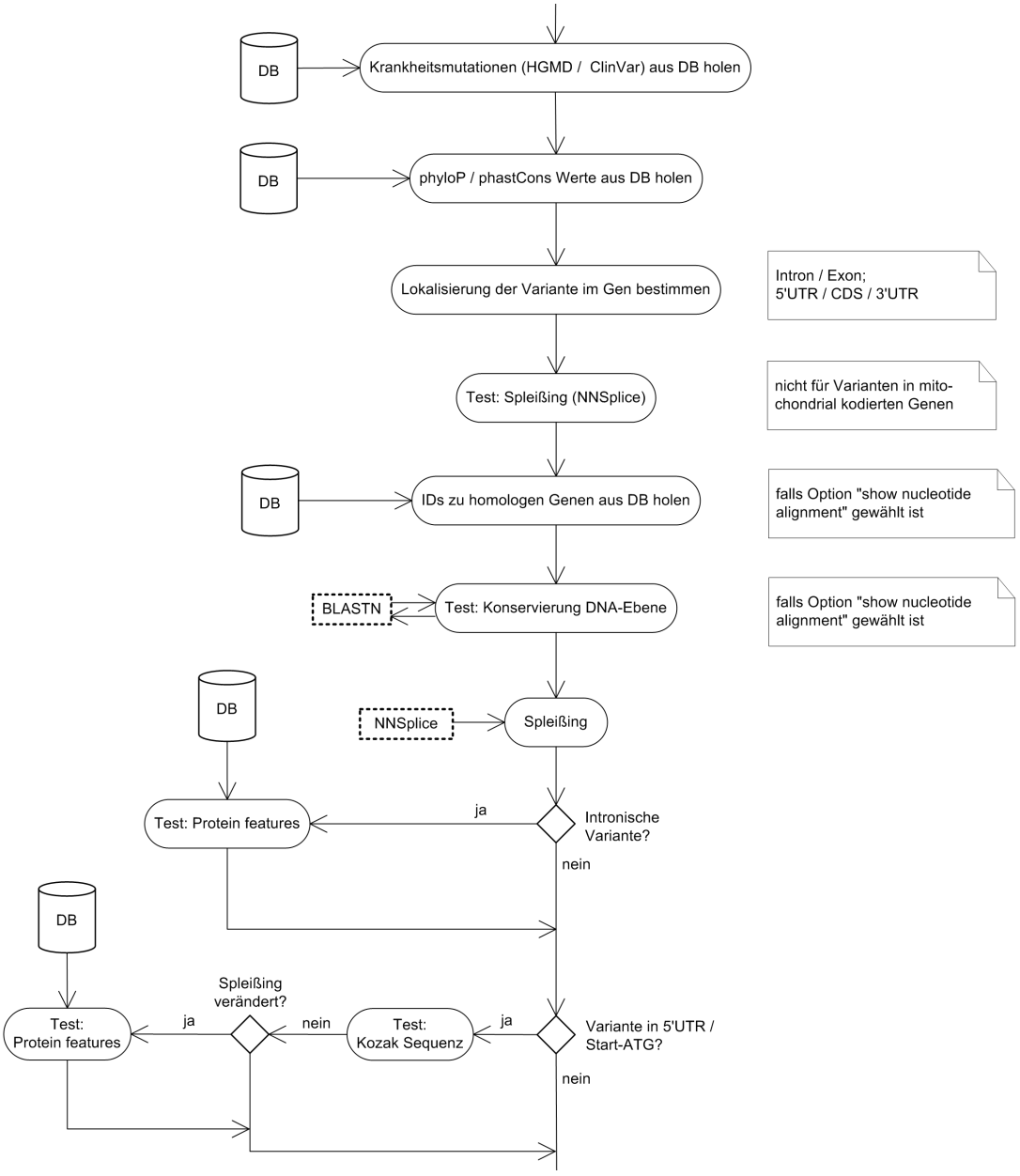


Abbildung 6: Vereinfachter Programmablauf von MutationTaster, Teil 2.

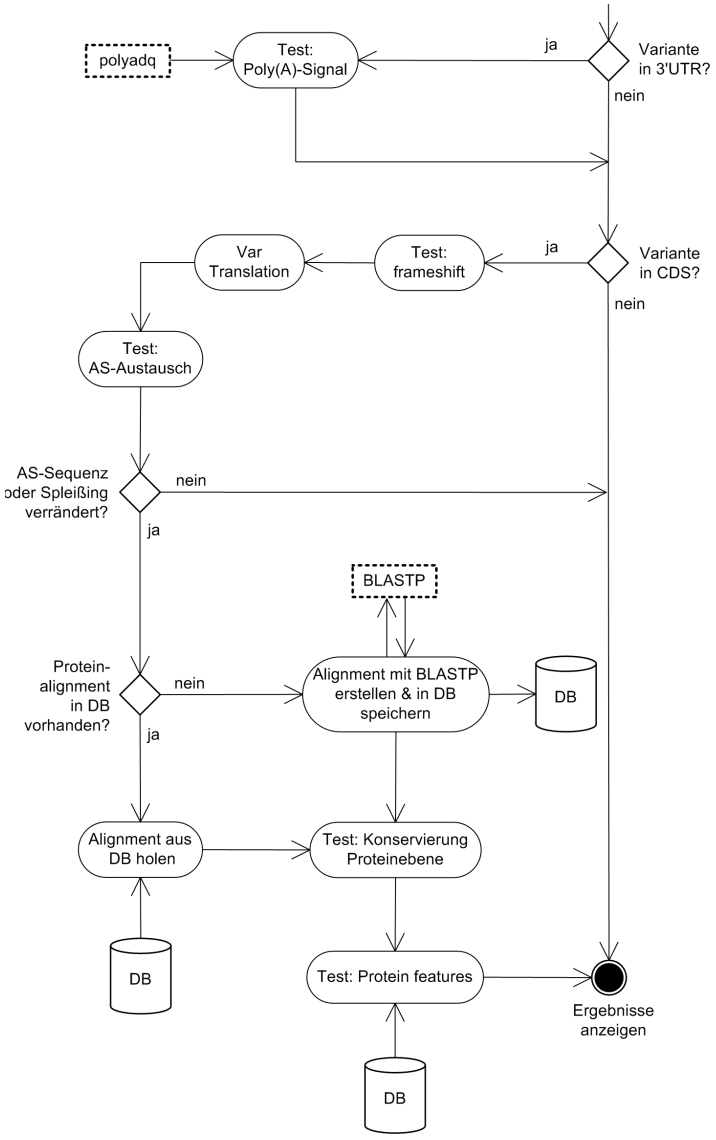


Abbildung 7: Vereinfachter Programmablauf von MutationTaster, Teil 3.

3.3 Programmablauf und Tests

Zur Bewertung des Krankheitspotentials einer DNA-Variante führt MutationTaster verschiedene Berechnungen, Tests und Analysen durch. Während einige allgemeine Tests für alle Varianten, unabhängig von ihrer Natur (z.B. Aminosäureaustausch *vs.* synonyme Veränderung der DNA-Sequenz) und Lokalisation (Intron *vs.* Exon, 5'UTR, CDS oder 3'UTR) vorgesehen sind, kommen andere spezielle Tests nur für bestimme Varianten zum Einsatz (z.B. Poly(A)-Signal Überprüfung nur für Veränderungen, die in der 3'UTR liegen). Im Folgenden werde ich zunächst die allgemeinen und anschließend die speziellen Tests genauer erläutern.

3.3.1 allgemeine Tests

phyloP / phastCons

Für alle Varianten wird die Konservierung auf der DNA-Ebene anhand von phyloP und phastCons Werten (siehe Kapitel 3.3) überprüft. phyloP und phastCons Werte basieren auf dem multiplen Alignment von 46 Vertebratenspezies. Dem Benutzer wird aber nicht das Alignment selbst angezeigt, sondern nur die phyloP / phastCons Werte für die von MutationTaster zu analysierende(n) Position(en) der DNA plus jeweils eine angrenzende Position *up-* und *downstream.*

Konservierung auf DNA-Ebene

Falls ein Benutzer dennoch am Alignment selber interessiert ist, kann er auf der MutationTaster-Startseite die Option *show nucleotide alignment* aktivieren. In der Folge wird MutationTaster für ein Maximum von zehn durch uns ausgewählte Spezies (für Details zur Auswahl siehe Kapitel 3.3.2, Konservierung auf Proteinebene) mit Hilfe von BLASTN (nucleotide BLAST) [29] und *bl2seq* [80] ein lokales Alignment rund um die veränderte DNA Basen ausführen und die Ergebnisse anzeigen. Diese zehn Spezies sind: Schimpanse (*P. troglodytes*), Rhesusaffe (*M. mulatta*), Maus (*M. musculus*), Katze (*F. catus*), Huhn (*G. gallus*), Krallenfrosch (*X. tropicalis*), Kugelfisch (*T. rubripes*), Zebrafisch *D. rerio*), Fruchtfliege (*D. melanogaster*) und der Fadenwurm (*C. elegans*). Über die Ensembl Gene ID der zu analysierenden Variante und eine Homologietabelle sucht MutationTaster zunächst nach homologen Sequenzen in den zehn Spezies. Die gefundenen Sequenzen (es gibt sie nicht immer in allen zehn Spezies) werden dann mit *bl2seq* jeweils paarweise mit der menschlichen Wildtypsequenz verglichen. Die Ausgabe von *bl2seq* wird anschließend analysiert und der Konservierungsstatus an der varianten Position determiniert. Folgende Stati sind möglich: *all identical*, alle Nukleotide an betrachteten Positionen sind komplett identisch; *partly identical*, die betrachteten Positionen sind teils identisch, teils nicht (kann nur zutreffen, falls die Variante sich über mehrere Nukleotide erstreckt); *not conserved*, die betrachtete(n) Position(en)

weisen komplett unterschiedliche Nukleotide auf. Angezeigt werden maximal 24 Nukleotide, dabei wird die veränderte Position mittig platziert und durch Fettdruck hervorgehoben. Aus Geschwindigkeitsgründen ist die Durchführung des Alignments auf Nukleotidebene nicht voreingestellt, da die evolutionäre Konservierung auf DNA-Ebene bereits durch die phyloP / phastCons Werte betrachtet wird. Außerdem ist ein isoliert betrachtetes Alignment auf DNA Ebene im Vergleich zu einem Proteinalignment weniger aussagekräftig, unter anderem deshalb weil der genetische Code redundant ist, und gleiche oder unterschiedliche Nukleotide an bestimmten Positionen nicht zwingend Rückschlüsse auf deren funktionelle Bedeutung zulassen.

Spleißanalyse

Unter „Spleißen“ versteht man in der Biologie das Herausschneiden von Introns aus der prä-mRNA und das anschließende Zusammenfügen der verbleibenden Exons zu einem reifen mRNA Molekül [89]. An diesem im Zellkern oft bereits parallel zur Transkription stattfindenden, zweistufigen und streng reguliertem Prozess sind eine Vielzahl verschiedener Moleküle beteiligt [90]. Elementar ist dabei die Erkennung der Spleißstellen, also der Nukleotidsequenzen an den Grenzen zwischen Exons und Introns. Diese sind hoch konserviert, die meisten Introns starten mit einem GT- und enden mit einem AG-Dinukleotid [89]. Allerdings tritt eine bestimmte Sequenz von nur zwei Basen Länge auch rein zufällig viel zu häufig auf, als dass sie als Markierung für Anfang und Ende eines Introns ausreichend wäre. Erst im Kontext mit benachbarten, ebenfalls konservierten Sequenzbereichen werden die AG / GT Dinukleotide zur Spleißstellen-Konsensussequenz [89]. Am 3’ Ende eines typischen Introns befindet sich ein konserviertes Pyrimidin an Position -3 und ab Position -5 ein bis zu zehn Basenpaare langes Polypyrimidin-Stück (poly(Y)) [91]. Außerdem spielt die typischerweise 11 bis 40 Nukleotide *upstream* der 3’ Introngrenze lokalisierte sogenannte *branch site* eine wichtige Rolle in der Initiierung des Spleißingprozesses [89]. Durch Veränderungen der verschiedenen an der Intron-Erkennung beteiligten Sequenzmuster kann es zu fehlerhaftem Spleißen kommen. Es wird eine andere als die ursprünglich vorgesehene mRNA translatiert und folglich ein verändertes Protein synthetisiert. Dessen Funktion kann dadurch eingeschränkt oder komplett gestört sein und somit zur Ausbildung einer Krankheit führen (z.B. [92]). Aber nicht nur die Zerstörung einer Spleißstelle kann schwerwiegende Folgen für den Organismus haben, sondern auch die Aktivierung sogenannter „kryptischer“, also normalerweise unbenutzter Spleißstellen. Dies kann ebenfalls die Herstellung fehlerhafter Proteine bedingen. In MutationTaster werden deshalb alle übermittelten Sequenzvarianten auf eventuelle Störungen der Spleißerkennungssequenzen untersucht. Ausgeführt wird diese Untersuchung durch das Computerprogramm NNSplice [28], welches jeweils einen Sequenzschnipsel von 30 Nukleotiden *up-* und *downstream* der veränderten Position von Wildtyp- und Variantensequenz übermittelt bekommt und Spleißstellen und ihre Stärke vorhersagt. MutationTaster vergleicht dann die Ergebnisse

von NNSplice und listet eventuelle Unterschiede auf. Folgende Szenarien sind möglich: 1) existierende Spleißstelle wurde stärker (*increased*) oder 2) schwächer (*decreased*), 3) zusätzliche Spleißstelle ist entstanden (*gained*) oder 4) existierende Spleißstelle ist verloren gegangen (*lost*). MutationTaster determiniert die Position der Spleißveränderung relativ zur Intron-Exon-Grenze und falls ein Verlust oder eine Reduzierung der Stärke einer Spleißstelle direkt an einer annotierten Intron-Exon-Grenze auftritt, wird dies als sicherer Verlust gewertet. Der Verlust oder die reduzierte Stärke von entfernteren, im jeweiligen Transkript ohnehin nicht genutzten Spleißstellen wird ignoriert, der Gewinn einer neuen Spleißstelle wird immer angezeigt, sobald der von NNSplice berechnete Konfidenzwert größer als 0.3 ist. Eine Verstärkung einer existierenden Spleißstelle wird angezeigt, sofern sich der Konfidenzwert um mindestens 10% erhöht hat. Die Analyse von Spleißstellen wird nicht für Varianten in mitochondrialen Genen durchgeführt, weil dort kein Spleißen erfolgt.

dbSNP / 1000 Genomes Project / HapMap / ClinVar

MutationTaster überprüft für alle übermittelten Varianten, ob diese bereits bekannt und in gängigen Datenbanken wie dbSNP, HapMap, dem 1000-Genom-Projekt, ClinVar oder der öffentlichen Version von HGMD gelistet sind.

Die Daten von *HGMD Public* wurden über Ensembl bezogen und beinhalten lediglich Positionen und nicht die Krankheits*allele*. MutationTaster überprüft für eine Variante, ob sie an einer Position lokalisiert ist, für die in *HGMD Public* eine Krankheitsmutation annotiert ist. Falls dies der Fall ist, sieht der Benutzer auf der Ergebnisseite den Hinweis, dass an der Position „seiner“ DNA-Variante in *HGMD Public* eine Krankheitsmutation gelistet ist, jedoch nicht genau welche. Über die HGMD ID und einen direkten Hyperlink zum Eintrag in *HGMD Public* können, sofern gewünscht, zusätzliche Informationen von der HGMD Webseite eingeholt werden. Aus rechtlichen Gründen darf MutationTaster die Krankheitsallele und den Phänotyp nicht anzeigen.

Für das 1000-Genom-Projekt, HapMap und ClinVar sind für alle Einträge auch die Allele verfügbar, hier kann also nicht nur anhand der Position eine eventuelle Übereinstimmung vermutet, sondern dank der Allele die tatsächliche Übereinstimmung bestätigt werden. Für die dbSNP Einträge waren zum Zeitpunkt der Datenbeschaffung keine Informationen zur Strangorientierung verfügbar, weshalb bei den gespeicherten Allelen teils unklar ist, worauf sie sich beziehen. Hier wird also nur darauf hingewiesen, dass die Position in dbSNP gelistet ist, und ein Hyperlink zum entsprechenden Eintrag angezeigt. Ob ein in dbSNP annotiertes Allel mit dem Allel der zu analysierenden Variante übereinstimmt, kann der Benutzer auf der dbSNP Webseite nachprüfen, die durch einen Hyperlink direkt angesteuert werden kann. Manche dbSNP Einträge sind ebenfalls in HapMap enthalten und werden somit zusammen mit Genotyp- und Allelfrequenzen gespeichert. In diesem Fall kann man mit Sicherheit sagen, ob die von MutationTaster zu analysierende Variante mit einem Eintrag in dbSNP / HapMap

übereinstimmt oder nicht. Wenn jeder der möglichen Genotypen (Wildtyp homozygot = AA, heterozygot = AB, Variante homozygot = BB) in mindestens einer der HapMap Populationen gefunden wurde, betrachtet MutationTaster die Variante als harmlos und vergibt automatisch die Vorhersage *polymorphism*.
Wenn der Vergleich mit ClinVar oder dem 1000-Genom-Projekt eine Übereinstimmung ergibt, wird dies ebenfalls angezeigt. Folgende Ergebnisse sind für Daten aus dem 1000-Genom-Projekt möglich: (1) die Variante wurde in mehr als vier Fällen homozygot im 1000-Genom-Projekt gefunden, genaue Anzahl der Fälle BB; (2) die Variante wurde mehr als vier Mal heterozygot im 1000-Genom-Projekt gefunden, Anzahl der gefundenen Fälle AB (gegebenenfalls Anzahl der Fälle für BB); (3) die Variante wurde in weniger als vier Fällen homo- oder heterozygot im 1000-Genom-Projekt gefunden, Anzahl der gefundenen Fälle für AB und BB. Falls eine Variante mehr als vier Mal homozygot im 1000-Genom-Projekt auftaucht, vergibt MutationTaster automatisch die Vorhersage *polymorphism*.
Folgende Ergebnisse sind für den Vergleich mit ClinVar möglich: (1) bekannte Krankheitsmutation in ClinVar, „pathogen“ und (2) bekannte Krankheitsmutation in ClinVar, „wahrscheinlich pathogen“. Die Übereinstimmung mit einer in ClinVar als „pathogen“oder „wahrscheinlich pathogen“ deklarierten Variante führt zur automatischen Vorhersage *disease causing*.
Wenn eine Vorhersage automatisch aufgrund existierender Übereinstimmung mit Daten in HapMap, dem 1000-Genom-Projekt oder ClinVar erfolgt ist, wird auf diesen Umstand hingewiesen. Im Falle einer automatischen (nicht vom Klassifikator gemachten) Vorhersage wird der Bayes Klassifikator zwar über alle Testergebnisse informiert, generiert aber einen Wahrscheinlichkeitswert, der sich auf die automatische Vorhersage bezieht. Ein Wahrscheinlichkeitswert von weniger als 0,5 bedeutet, dass der Bayes Klassifikator von sich aus die entgegengesetzte Vorhersage gemacht hätte. Falls eine Variante zugleich in ClinVar und mehr als vier Mal homozygot im 1000-Genom-Projekt auftaucht, erhält die Präsenz im 1000-Genom-Projekt die höhere Priorität und die Variante wird automatisch als *polymorphism* deklariert.

Regulatorische Elemente

Für alle zu analysierenden Varianten wird überprüft, ob sie ein in *Ensembl Regulation* annotiertes, regulatorisches Element beeinflussen könnten (z.B. Promotor-assoziierte Regulationselemente, DNase1 Hypersensitivitäts-, Histonmodifikations- oder Transkriptionsfaktorbindungsstellen, siehe auch Kapitel 2.2.1). Falls eine Variante innerhalb eines solchen Elements lokalisiert ist, nutzt der Bayes Klassifikator die möglicherweise daraus resultierende Fehlfunktion des regulatorischen Elements zur Vorhersage und auf der Ergebnisseite erscheint eine entsprechende Information für den Benutzer. Es gibt mehr als 600 verschiedene regulatorischen Element-Typen, die in zwölf verschiedene Klassen eingeteilt sind (siehe auch Kapitel 2.2.1. Aus Gründen der Einfachheit wird dem Bayes Klassifikator lediglich die Klassenzugehörigkeit übermittelt und auch nur diese fließt in die Vorhersage ein (siehe dazu auch Kapitel

8.2.1). Dem Benutzer werden alle Details zu einem gefundenen Element angezeigt (Klassenzugehörigkeit, Name des Elements und Beschreibung).

3.3.2 spezielle Tests

Nicht alle Tests sind für alle Varianten sinnvoll. MutationTaster sammelt deshalb zunächst grundsätzliche Informationen über die Natur der Variante und teilt sie dann in eine bestimmte Kategorie (Intron *vs.* Exon, 3'UTR *vs.* CDS *vs.* 5'UTR, nicht-synonym *vs.* synonym) ein. Für die unterschiedlichen Kategorien sind unterschiedliche Tests vorgesehen.

3.3.2.1 5'UTR

Als 5'UTR wird der Bereich am Beginn eines Gens bezeichnet, der zwar in mRNA umgeschrieben, jedoch nicht in eine Aminosäuresequenz übersetzt wird. Er beginnt mit dem ersten Nukleotid der mRNA und endet mit dem letzten Nukleotid vor dem Start-ATG. Obwohl die 5'UTR keine Aminosäuren kodiert, spielt sie dennoch eine maßgebliche Rolle für die Proteinsynthese: Sie enthält wichtige Elemente zur Regulation der Translationseffizienz [93], beeinflussende Faktoren sind z.B. die 5'-Cap-Struktur, lokale Sekundärstrukturen der mRNA, multiple ORFs (*o*pen *r*eading *f*rame, Leseraster zum Ablesen der mRNA), multiple uAUGs (*u*pstream AUGs, vorgelagerte ATG/AUG Codons), IRESs (*i*nternal *r*ibosome *e*ntry *s*ites) sowie die Positionierung des Start-ATGs in einem mehr oder weniger optimalen Kozak-Kontext (siehe unten). Der letzte Punkt ist bislang der einzige, der von MutationTaster zur Analyse von Varianten in der 5'UTR genutzt wird.

Kozak-Sequenz

Die Kozak-Sequenz (gccRccAUGG; R = Purin) erstreckt sich rund um das Startcodon (ATG bzw. AUG) und spielt eine wichtige Rolle in der Translationsinitiation. Neben dem ATG an den Positionen +1,+2 und +3 ist ein Purin an Position -3 sowie ein G an Position +4 hoch konserviert [94][95]. Ein „starkes" ATG liegt vor, wenn die konservierten Positionen -3 und +4 der Konsensussequenz entsprechen und führt zu einer effizienteren Translation als ein ATG, das nicht den konservierten Sequenzkontext aufweist und als „schwaches" ATG bezeichnet werden kann [96]. DNA-Mutationen, die ursprünglich starke Startcodons abschwächen, können krankheitsverursachend sein [97][98]. MutationTaster überprüft für in der 5'UTR oder den ersten vier Nukleotiden der CDS lokalisierte Varianten, ob durch Veränderung der Kozak-Sequenz ein vormals starkes ATG abgeschwächt wird.

3.3.2.2 Kodierende Sequenz

DNA-Varianten in der kodierenden Sequenz (CDS) eines Gens können die Codons, welche für Aminosäuren kodieren, verändern. MutationTaster führt für in der CDS lokalisierte Varianten Tests aus, die sich auf die Veränderung der Aminosäuresequenz beziehen.

Aminosäureaustausch und *frameshift*

Für Varianten, die ganz oder teilweise in der CDS eines Gens lokalisiert sind, wird überprüft, ob die Veränderung auf DNA-Ebene die ursprünglich kodierte Aminosäuresequenz verändert. Dazu translatiert MutationTaster die Wildtyp- und variante CDS und vergleicht anschließend, ob die beiden Aminosäuresequenzen sich unterscheiden. Eventuelle Veränderungen der Aminosäuresequenz werden in der HGVS (*H*uman *G*enome *V*ariation *Society*) Nomenklatur dargestellt. Eine Insertion / Deletion eines Nicht-Vielfachen von drei führt zu einer Verschiebung des Leserasters (*frameshift*). Dies wird von MutationTaster registriert und zusätzlich zur Annotation in der HGVS Nomenklatur (z.B. S49Gfs87*) separat als *frameshift: yes* oder *frameshift: no* auf der Ergebnisseite notiert.
Für jeden einzelnen Aminosäureaustausch wird ein den Schweregrad des Austausches reflektierender Zahlenwert angezeigt. Dieser wird aus der Grantham Matrix [72] (siehe Kapitel 2.2.1) entnommen, die basierend auf den physiko-chemischen Eigenschaften zweier Aminosäuren ihre Ähnlich- bzw. Unähnlichkeit beziffert. Die Anzeige des Grantham-Wertes dient allerdings lediglich Informationszwecken, der Bayes Klassifikator nutzt ihn nicht für die Generierung seiner Vorhersage, sondern betrachtet nur den oder die isolierten Aminosäureaustausche, ohne die Bewertung durch die Grantham-Matrix zu berücksichtigen.
Falls eine zu analysierende Variante das Start-ATG zerstört, sucht MutationTaster nach dem nächsten auftretenden ATG und informiert den Benutzer darüber, wie viele Aminosäuren am Beginn des Proteins fehlen würden, und ob durch das neue ATG das alte Leseraster erhalten bliebe (*in-frame*) oder verschoben würde (*out-of-frame*).

Konservierung auf Protein-Ebene

Falls eine Variante zu einem Aminosäureaustausch führt, wird der Konservierungsstatus der entsprechenden Position(en) auf Proteinebene überprüft. MutationTaster benutzt *bl2seq* [80] mit BLASTP, um das Wildtyp-Protein mit bis zu zehn verschiedenen Homologen anderer Spezies zu vergleichen. Die zehn Spezies stellen mit dem Schimpansen (*P. troglodytes*), dem Rhesusaffen (*M. mulatta*), der Maus (*M. musculus*), Katze (*F. catus*), Huhn (*G. gallus*), Krallenfrosch (*X. tropicalis*), dem Kugelfisch (*T. rubripes*), dem Zebrafisch *D. rerio*), der Fruchtfliege (*D. melanogaster*) und dem Fadenwurm (*C. elegans*) einerseits eine evolutionär breit gefächerte Auswahl dar und umfassen andererseits einige der oft genutzten Modellorganismen. In Optimierungstests konnten wir durch die Einbindung von mehr als zehn Spezies keine verbesserte Vorhersagequalität erreichen, die Geschwindigkeit von MutationTaster wurde allerdings deutlich langsamer. Aus Zeitgründen ist die Konservierungsanalyse daher bewusst auf die oben genannten, phylogenetisch breit gefächerten zehn Spezies beschränkt.
Da die Analyse mit *bl2seq* / BLASTP zeitaufwendig ist, werden die Proteinalignments in der MutationTaster Datenbank gespeichert, wenn sie zum ersten Mal benötigt werden. Wird anschließend erneut eine Variante in dem gleichen Gen analysiert, benutzt MutationTaster

zur Überprüfung der Konservierung an der fraglichen Position die bereits in der Datenbank vorhandenen Alignments. Dies ist möglich, da für die Konservierungsanalyse nur die veränderte Position der Wildtypsequenz betrachtet wird, und nicht die Veränderung selber.
Das Ergebnis des Konservierungstests und die analysierte(n) Position(en) im Sequenzkontext werden für jede Spezies einzeln angezeigt. Tabelle 3 zeigt, welche Ergebnisse möglich sind.

Ergebnis	Voraussetzung
all identical	identische Aminosäure in der humanen und der homologen Sequenz
conserved	ähnliche Aminosäure in der humanen und homologen Sequenz
not conserved	unterschiedliche Aminosäure in der humanen und homologen Sequenz
no homologue	keine dem Gen entsprechende homologe Aminosäuresequenz verfügbar
no alignment	humane und homologe Sequenz sind sich im fraglichen Sequenzabschnitt so unähnlich, dass *bl2seq* sie nicht alignieren kann

Tabelle 3: Mögliche Ergebnisse der Konservierungsanalyse auf Proteinebene.

Proteindomänen

Der Austausch nur einer einzigen Aminosäure kann fatale Folgen für ein Protein haben, wenn diese Aminosäure eine elementare Rolle für die Funktion des Proteins spielt. Wenn beispielsweise die relativ kleine Aminosäure Glycin in der Biegung einer alpha-Helix durch ein Prolin ersetzt wird, wird die Sekundärstruktur des Proteins nicht mehr korrekt ausgebildet. In der Folge kann z.B. die Interaktion mit einem Bindungspartner gestört sein und das Protein daran hindern, seine ursprüngliche Funktion auszuüben. Falls MutationTaster feststellt, dass eine zu analysierende Variante einen Aminosäureaustausch oder Spleißveränderungen nach sich zieht, wird geprüft, ob die veränderte(n) Aminosäure(n) in einer in SwissProt annotierten, funktionellen oder strukturell distinkten Proteindomäne liegen. Dies können unter anderem posttranslationelle Modifikationen an einzelnen Aminosäuren (z.B. Phosphorylierung an einem Tyrosin), einfache Proteindomänen (z.B. Signalpeptide oder Transmembrandomänen) oder komplexere Proteindomänen (z.B. SH2 Domäne, Ras-GAP Domäne, Stellen zur Ausbildung von Disulfidbrücken) sein. Im Falle einer *frameshift* Variante werden alle Proteindomänen *downstream* der Variante als verloren gewertet. Bei Spleißveränderungen werden Proteindomänen *downstream* des irregulären Spleißing-Events dem Benutzer als möglicherweise verloren (*might get lost*) angezeigt, aber vom Bayes Klassifikator nicht für die Vorhersage genutzt.

Stopcodon

Generell kann man davon ausgehen, dass eine *nonsense*-Mutation, also eine DNA-Variante, die ein verfrühtes Stopcodon (*premature termination codon*, PTC) bewirkt, dramatische Folgen für das Protein hat. Ein verfrühtes Stopcodon kann entweder direkt durch einen Base-

naustausch (z.B. TAC > TAG) oder indirekt durch einen *frameshift* entstehen. Im letzteren Fall wird die Aminosäuresequenz ab der Änderung mehr oder weniger komplett verändert. Die Chance auf ein zufällig auftretendes Stopcodon liegt bei knapp 5% pro Codon (drei der 64 Codons sind Stopcodons) und addiert sich nach 100 Codons auf 99%. Das heißt, wenn ein *frameshift* nicht ganz am Ende eines Proteins auftritt, bewirkt er früher oder später fast immer ein vorzeitiges Stopcodon. Wenn eine mRNA mit einem vorzeitigen Stopcodon translatiert wird, resultiert dies theoretisch in einem verkürzten Protein mit einer in der Regel veränderten Funktion. Im besten Falle ist das Protein einfach funktionslos, im schlimmsten Fall kann es aber auch einen negativen Funktionsgewinn haben. Bestimmten Regeln folgend findet daher eine Degradierung von irregulären mRNAs mit einem vorzeitigen Stopcodon statt, wodurch negative Folgen für den Organismus vermieden werden. Diesen Mechanismus nennt man *n*onsense-*m*ediated mRNA *d*ecay (NMD), er tritt unter anderem dann auf, wenn ein vorzeitiges Stopcodon mehr als 50-55 Nukleotide *upstream* zur letzten Intron-Exon-Grenze auftritt (die sogenannte *50-54 nt boundary rule*) [99].
MutationTaster bestimmt ob und wie stark sich durch eine Aminosäureveränderung die ursprüngliche Länge des Proteins verändert. Tabelle 4 zeigt, welche Ergebnisse möglich sind:

Ergebnis	Voraussetzung
NMD	*50-54 nt boundary rule* erfüllt, in diesem Fall vergibt MutationTaster automatisch die Vorhersage *disease causing*
strongly truncated, might cause NMD; stark verkürzt, könnte NMD verursachen	Protein um mehr 10% verkürzt, aber *50-54 nt boundary rule* nicht erfüllt
slightly truncated, might cause NMD; geringfügig verkürzt, könnte NMD verursachen	weniger als 10% der originalen Proteinlänge fehlen, *50-54 nt boundary rule* nicht erfüllt
prolonged; verlängert	originales Stopcodon außer Kraft gesetzt, Translation wird erst später beendet als vorgesehen
protein of normal length; normale Proteinlänge	keine Veränderung der Proteinlänge

Tabelle 4: Mögliche Ergebnisse der Untersuchung zur Lokalisation des Stopcodons

3.3.2.3 3'UTR

Die 3'UTR befindet sich am hinteren Ende eines Genes, genauer gesagt ist es der Bereich zwischen Stopcodon und dem Ende des letzten Exons. In der 3'UTR befinden sich verschiedene regulatorische Elemente [100].

Polyadenylierungs-Signal

Die am 3' Ende der mRNA angehängte Poly(A)-Sequenz spielt nicht nur eine Rolle in der Translationsinitiierung [101], sondern beeinflusst auch die mRNA-Stabilität [102]. Der

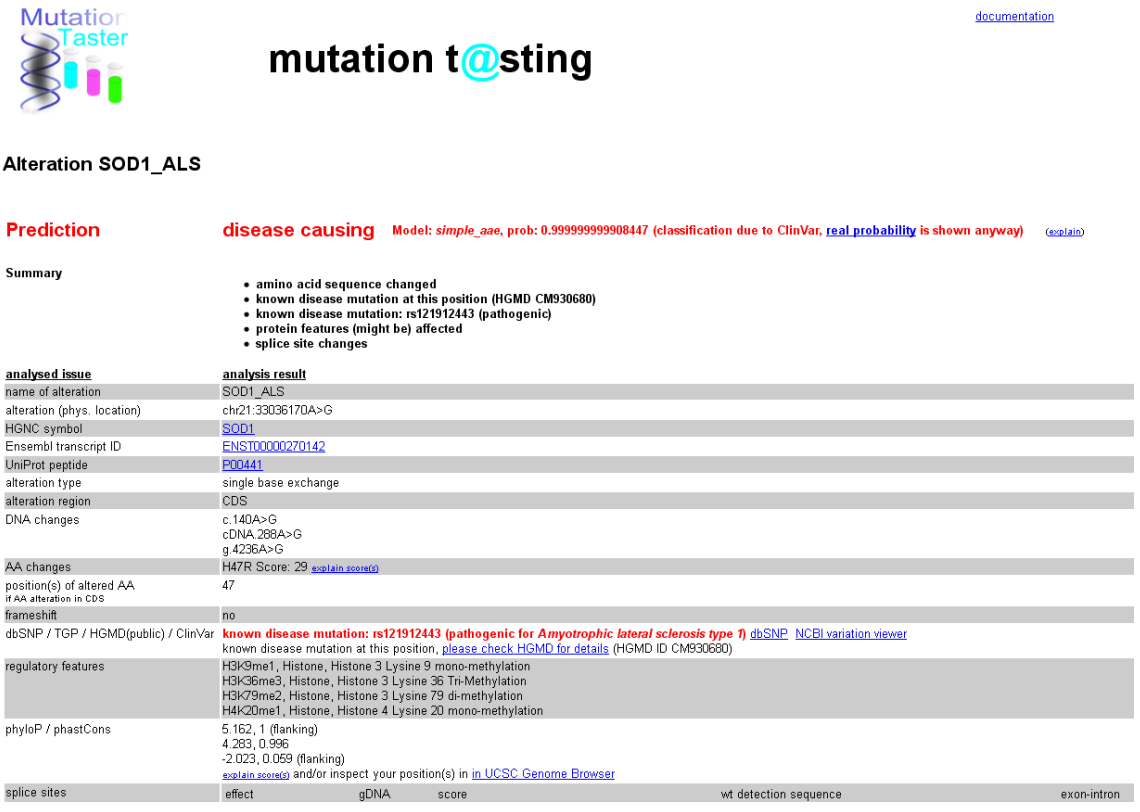
Mechanismus der Polyadenylierung ist relativ gut untersucht. In den meisten Fällen ist das Poly(A)-Signal (PAS) eines der beiden Hexamere AATAAA oder ATTAAA [30].
Es ist bekannt, dass Mutationen in der 3'UTR, die das PAS stören, krankheitsverursachend sein können [103][104]. Für DNA-Veränderungen in der 3'UTR eines Gens überprüft MutationTaster deshalb, ob durch sie das PAS gestört wird. Dazu wird die Wildtyp- und variante cDNA-Sequenz an das lokal installierte Programm polyadq [30] übermittelt, das für beide Sequenzen eventuelle PAS vorhersagt. Die Ergebnisse werden von MutationTaster verglichen, und falls durch die zu analysierende DNA-Variante ein PAS verloren gegangen ist, wird dies dem Klassifikator mitgeteilt und dem Benutzer auf der Ergebnisseite angezeigt.

3.4 Geschwindigkeitsoptimierung

Die Laufzeit eines Programmes zur Evaluierung von DNA-Sequenzvarianten ist im Zeitalter von NGS-Projekten von großer Relevanz. Deshalb wurden verschiedene Aspekte optimiert, um so die Laufzeit eines einzelnen MutationTaster Aufrufes soweit wie möglich zu reduzieren. Um Ansatzpunkte zur Optimierung zu finden, wurde für jede einzelne Subroutine im MutationTaster Programmcode die Start- und die Endzeit in eine Datei geschrieben. So konnten wir einzelne Subroutinen, die besonders viel Laufzeit benötigten, identifizieren. Fehlende *Datenbankindizes** sowie der Aufruf externer Programme waren die häufigste Ursache für längere Laufzeiten einzelner Subroutinen. Während wir im ersteren Fall durch Anlegen passender Indizes relativ leicht die Laufzeit der einzelnen Subroutine verkürzen konnten, ist dies beim Aufruf externer Programme schwieriger. Der Aufruf von *bl2seq* bzw. BLASTP während der Konservierungsanalyse dauert sehr lange (etwa 350 ms), das resultierende Alignment ist aber elementar. Da allerdings für die Konservierungsanalyse nur das Wildtyp-Protein mit geeigneten Homologen verglichen werden muss, ist dieser Vorgang von der eigentlichen Variante unabhängig. Somit kann das Ergebnis des Alignments, sobald es einmal ausgeführt wurde, in der Datenbank gespeichert und bei der Analyse einer anderen Variante im gleichen Gen aus der Datenbank genommen werden. Dadurch wird ein erneuter, zeitaufwendiger Aufruf von *bl2seq* / BLASTP umgangen, die Verwendung des gespeicherten Alignments dauert im Durchschnitt nur 7 ms. Durch das Speichern aller Konfigurationsdateien und externer Programme auf einer *RAM-Disk** (eine virtuelle Festplatte im Arbeitsspeicher, wodurch echte, zeitaufwendige, Festplattenzugriffe entfallen können) und die Benutzung von *mod_perl** zur Ausführung des MutationTaster Programmcodes konnten wir die Geschwindigkeit ebenfalls enorm erhöhen. mod_perl ist ein Modul für den Apache Webserver, das dafür sorgt, dass einmal kompilierter Code im Speicher verbleibt. Dadurch ist der Code wesentlich schneller verfügbar, als dies durch eine ständige Neu-Kompilierung mit dem normalen Perl-Interpreter möglich wäre [105].

3.5 Programmausgaben

Die Ergebnisse einer MutationTaster Analyse werden im HTML-Format ausgegeben. Zuoberst wird das vorhergesagte Krankheitspotential der analysierten DNA-Variante (entweder *polymorphism* in grün oder *disease causing* in rot) zusammen mit dem Konfidenz- oder Wahrscheinlichkeitswert (*probability value*) der Vorhersage angezeigt. Darunter folgt in Fettschrift aufgelistet eine Zusammenfassung der wichtigsten Ergebnisse (z.B. *amino acid sequence changes*, *protein features affected*). Anschließend werden in tabellarischer Anordnung alle durchgeführten Tests und Berechnungen (links) sowie die Ergebnisse (rechts) aufgeführt. Abbildung 8 zeigt den oberen Teil einer typischen MutationTaster Ergebnisseite.



documentation

mutation t@sting

Alteration SOD1_ALS

Prediction **disease causing** Model: *simple_aae*, prob: 0.999999999908447 (classification due to ClinVar, real probability is shown anyway) (explain)

Summary

- amino acid sequence changed
- known disease mutation at this position (HGMD CM930680)
- known disease mutation: rs121912443 (pathogenic)
- protein features (might be) affected
- splice site changes

analysed issue	analysis result
name of alteration	SOD1_ALS
alteration (phys. location)	chr21:33036170A>G
HGNC symbol	SOD1
Ensembl transcript ID	ENST00000270142
UniProt peptide	P00441
alteration type	single base exchange
alteration region	CDS
DNA changes	c.140A>G cDNA.288A>G g.4236A>G
AA changes	H47R Score: 29 explain score(s)
position(s) of altered AA if AA alteration in CDS	47
frameshift	no
dbSNP / TGP / HGMD(public) / ClinVar	known disease mutation: rs121912443 (pathogenic for *Amyotrophic lateral sclerosis type 1*) dbSNP NCBI variation viewer known disease mutation at this position, please check HGMD for details (HGMD ID CM930680)
regulatory features	H3K9me1, Histone, Histone 3 Lysine 9 mono-methylation H3K36me3, Histone, Histone 3 Lysine 36 Tri-Methylation H3K79me2, Histone, Histone 3 Lysine 79 di-methylation H4K20me1, Histone, Histone 4 Lysine 20 mono-methylation
phyloP / phastCons	5.162, 1 (flanking) 4.283, 0.996 -2.023, 0.059 (flanking) explain score(s) and/or inspect your position(s) in in UCSC Genome Browser
splice sites	effect gDNA score wt detection sequence exon-intron

Abbildung 8: MutationTaster Ergebnisseite (nur der obere Teil ist dargestellt).

4 Der Bewertungsprozess in MutationTaster

MutationTaster bewertet das wahrscheinliche Krankheitspotential von DNA-Sequenzveränderungen mit Hilfe eines Bayes Klassifikators. Dabei dienen die Ergebnisse der in Kapitel 3.3 beschriebenen Tests und Berechnungen dem Klassifikator als Grundlage für die Entscheidung, ob eine DNA-Veränderung eher als harmlose oder eher als eine krankheitsverursachende Variante eingestuft wird. In diesem Kapitel werde ich den Bewertungsprozess in MutationTaster genauer erläutern.

4.1 Computergestützte Klassifizierungsverfahren

Eine Entscheidung ist die Wahl zwischen zwei oder mehreren Alternativen [106]. Wenn eine Entscheidung aus rationalen Gründen getroffen wird, dann beruht sie oft auf bereits vorhandenen Wertmaßstäben und der Berücksichtigung eines angestrebten Zieles. Ein Mensch, der eine rationale Entscheidung trifft, rekapituliert Wissen, das im Zusammenhang mit der zu treffenden Entscheidung hilfreich sein könnte, wägt verschiedene Möglichkeiten gegeneinander ab und entscheidet sich schließlich für die Variante, die ihm aufgrund seiner Überlegungen als die beste erscheint. In bestimmten Situationen kann es sinnvoll sein, rationale Entscheidungen mit Hilfe eines Computers zu automatisieren: Dies nennt man computergestützte oder automatische Klassifizierung. Die automatische Klassifizierung ist ein Teilgebiet des „maschinellen Lernens“, das heißt der künstlichen Generierung von Wissen aus Erfahrung [107].
Maschinelles Lernen wiederum kann als ein Teilgebiet der „Künstlichen Intelligenz“ betrachtet werden, die sich im speziellen mit der Entwicklung maschinengestützter Intelligenz beschäftigt. Dabei geht es um das Verständnis intelligenten Verhaltens und dessen künstlicher Imitation zu Zwecken der Automatisierung [108]. Ein künstlich intelligentes System sollte in der Lage sein, sich möglichst viel Wissen anzueignen und dieses zur Beantwortung von Fragen und zur Lösung bestimmter Probleme zu nutzen [108].
Das intelligente Verhalten einer Maschine bzw. eines Computerprogramms basiert auf und resultiert aus (1) angeeignetem Wissen, (2) bereits gemachten Erfahrungen, (3) vorgegebenen Zielen und (4) Beobachtungen [108]. Dabei ist das angeeignete Wissen ein wichtiger Punkt, in dem Mensch und Maschine sich unterscheiden. Menschen besitzen und benutzen in der Regel selbst für einfache Aufgaben sehr viel von ihrem im Laufe der Jahre gesammelten Wissen. Wenn es zum Beispiel um das Erkennen von Gesichtern, medizinische Diagnosen, oder komplexe Zusammenhänge geht, ist der Mensch dem Computer überlegen. Letzterer hingegen ist sehr gut darin, weniger komplexe, aber arbeitsaufwendige Aufgaben schnell und sicher zu erledigen, z.B. Daten zu sortieren oder zu suchen. Deshalb gibt es auch halbautomatische Verfahren zur computergestützten Klassifizierung: Die Kandidatengensuchmaschine GeneDistiller [85] beispielsweise vereint die Vorteile eines Computers (schnelle Durch-

suchung, Berechnung und Darstellung großer Datenmengen) mit dem umfassenden Fachwissen eines Menschen (der z.B. komplexe Sachverhalte gedanklich verknüpfen und miteinander in Verbindung bringen kann).
Eine automatische Klassifizierung bedingt, ein Modell zu entwickeln, welches unterschiedliche Datenklassen beschreibt und unterscheidet, und dieses Modell zu nutzen, um die Klassenzugehörigkeit unbekannter Objekte vorherzusagen [109]. Aus einem vorangehenden Training mit Objekten bekannter Klassenzugehörigkeit wird das Modell entwickelt. Das Modell kann unterschiedliche Formen annehmen: beispielsweise die eines Entscheidungsbaum, der sich einfach in Klassifizierungsregeln übersetzen lässt, die eines neuronalen Netzes, einer *Support Vector Machine* oder eines Bayes Klassifikators [109].

4.2 Bayes Klassifikator - Grundlagen

Ein Bayes Klassifikator ist eine auf dem Bayes Theorem basierende Anwendung zur Vorhersage von Klassenzugehörigkeiten [110]. Das Bayes Theorem [111] beschreibt die Berechnung bedingter Wahrscheinlichkeiten, das heißt, die Wahrscheinlichkeit für ein Ereignis B, unter der Bedingung, dass Ereignis A eingetreten ist, $P(A|B)$. Der Bayes Klassifikator nutzt dies, um ein Objekt der Klasse zuzuordnen, der es am wahrscheinlichsten angehört, es also zu klassifizieren. Dazu muss der Klassifikator zunächst mit Daten bekannter Klassenzugehörigkeit trainiert werden, sogenannte Trainingsfälle. Dieser werden systematisch auf bestimmte Eigenschaften, d.h. auf Zustände bestimmter Attribute, untersucht, und deren Frequenzen in den einzelnen Klassen gespeichert. Einen unbekannten Fall, also einen Testfall, kann nun ebenfalls auf die Zustände der in den Testfällen gesehenen Attribute untersucht, und dadurch klassifiziert werden.
Ein Beispiel für den Einsatz von Bayes Klassifikatoren sind Spam Filter für E-Mails. In Spam E-Mails werden bestimmte, charakteristische Wörter besonders häufig verwendet, z.B. *medication* oder *buy now*. Das Wort MutationTaster beispielsweise kommt vermutlich äußerst selten in Spam Mails vor. Der Bayes Klassifikator, der initial mit Spam E-Mails und normalen E-Mails trainiert wurde, kann anhand der Frequenz bestimmter Wörter in einer E-Mail diese als Spam oder Nicht-Spam klassifizieren.
Ein *naiver* Bayes Klassifikator wird deshalb als naiv bezeichnet, weil davon ausgegangen wird, dass alle Attribute voneinander unabhängig sind. Man nimmt also an, dass das Auftreten oder nicht Auftreten eines bestimmten Zustandes eines Attributs unabhängig ist vom Auftreten des Zustandes eines anderen Attributs [110]. Obwohl dies in der Realität meist nicht gegeben ist, bietet der naive Bayes Klassifikator dennoch bei erfreulich einfacher Handhabung robuste und gute Klassfizierungsergebnisse. Zwar müsste zumindest theoretisch ein differenziertes, ausgeklügelteres System (z.B. ein Neuronales Netz), das die verschiedenen Variablen im richtigen Zusammenhang berücksichtigt, besser sein als ein naiver Bayes Klassifikator. Dies ist allerdings nicht immer der Fall, wie diverse Vergleiche von Bayes Klassifikatoren

mit Enscheidungsbaumklassifikatoren oder Neuronalen Netzen zeigen konnten: Oft ist die Performanz der verschiedenen Methoden tatsächlich vergleichbar [110]. Zahlreiche Studien belegen außerdem, dass naive Bayes Klassifikatoren trotz ihrer „Schwäche“ ausgeklügelten, hochentwickelten Systemen sogar teils überlegen sind [112]. Sehr detaillierte Modelle haben ein größeres Risiko fehlerbehaftet zu sein, ein Problem das durch das einfache Modell des naiven Bayes Klassifikators minimiert wird [112]. Bayes Klassifikatoren werden deshalb auch eingesetzt um kompliziertere künstlich intelligente Systeme zu testen: Wenn ein solches System schlechter ist als ein Bayes Klassifikator, dann muss es überarbeitet werden.
Neben ihrer guten Erkennungsrate bieten naive Bayes Klassifikatoren den weiteren Vorteil, dass sie sehr schnell sind. Wenn der Klassifikator trainiert wurde, muss er anschließend nicht mehr auf die kompletten Trainingsdaten zugreifen, sondern nur noch auf die Frequenzen der Zustände ihrer Attribute, auf das sogenannte Modell. Kritische Punkte beim Aufsetzen eines Bayes Klassifikators sind die Erstellung eines genügend großen, ausgewogenen Trainingssets sowie die Wahl geeigneter Attribute, die vom Klassifikator als Diskriminatoren genutzt werden können.

4.3 Bayes Klassifikator - Anwendung in MutationTaster

In MutationTaster wird ein Bayes Klassifikator eingesetzt, um eine zu analysierende Variante entweder der Klasse *polymorphism* oder der Klasse *disease causing* zuzuordnen. Hierzu haben wir ein publiziertes Perl Modul (*AI::NaiveBayes1*) genutzt. Die einzelnen zur Vorhersage benutzten Attribute werden zwar als unabhängig voneinander betrachtet, der in MutationTaster integrierte Klassifikator ist aber in der Realität nicht komplett naiv. Tatsächlich wird die Entscheidung in einigen Situationen nicht ausschließlich basierend auf dem Modell getroffen, sondern auch unter Berücksichtigung zusätzlicher Informationen.
Der Klassifikator wurde mit bekannten Krankheitsmutationen und harmlosen, in der Bevölkerung häufig auftretenden Polymorphismen, trainiert. Alle Trainingsfälle wurden den jeweils geeigneten, in Kapitel 3.3 beschriebenen, Tests unterworfen und die Ergebnisse zusammen mit der Klassenzugehörigkeit an den Klassifikator übermittelt. Dieser speichert in der Regel die Tests als Attribute und die Ergebnisse als Zustände. In einigen Fällen werden zu einem Test mehrere Attribute gespeichert (z.B. phyloP / phastCons). Daraus entsteht am Ende das Modell, aus dem sich beispielsweise ergibt, dass für den Test *Proteinlänge* das Ergebnis *verfrühtes Stopcodon mit Folge NMD* in 79% der Krankheitsfälle aber nur in 0.7% der harmlosen Fälle aufgetreten ist (*complex_ aae* Modell). Wenn der Klassifikator dann in einem unbekannten Fall, also einer vom Benutzer übermittelten Variante, für das Attribut *Proteinlänge* den Zustand *NMD* beobachtet, wird dies die Wahrscheinlichkeit für eine Krankheitsmutation erhöhen. Für die endgültige Vorhersage ausschlaggebend ist, ob die betrachten Zustände aller untersuchter Attribute in ihrer Gesamtheit mehr für einen harmlosen Polymorphismus sprechen oder mehr für eine Krankheitsmutation.

Die meisten bekannten Polymorphismen sind synonym, belassen die Aminosäuresequenz also unverändert, während viele der bekannten Krankheitsmutationen teils drastische Veränderungen für das Protein nach sich ziehen (siehe Übersicht der Trainingsdaten in Tabelle 5). Die bloße Existenz eines Aminosäureaustausches könnte für den Klassifikator daher ein Hinweis auf eine eventuelle Krankheitsmutation sein. Um dies zu umgehen, gibt es in MutationTaster nicht einen, sondern insgesamt drei verschiedenen Klassifikatoren, bzw. Modelle, für unterschiedliche Arten von Varianten. MutationTaster bestimmt bei einer zu analysierenden Variante zunächst, welcher Art sie ist, und benutzt automatisch das passende Vorhersagemodell. Die drei Modelle wurden jeweils nach der Art der Veränderung benannt, die sie analysieren, dabei steht das *aae* im Namen für *a*mino *a*cid *e*xchange (Aminosäureaustausch). Die drei Modelle sind:

1. **without_aae**: Dieses Modell analysiert Varianten, die keinen Aminosäureaustausch bewirken. Dies sind entweder intronische Veränderungen, exonische Veränderung in den UTRs oder exonische Veränderungen, die aufgrund der Redundanz des genetischen Codes nicht zu einer Veränderung der kodierten Aminosäure führen.

2. **simple_aae**: Modell für Varianten, durch die eine einzelne Aminosäure durch eine andere ersetzt wird. Der Aminosäureaustausch wird als einzelnes Attribut an den Klassifikator übergeben.

3. **complex_aae**: Modell für komplexere Veränderungen, die die Aminosäuresequenz stärker als durch einen einzelnen Aminosäureaustausch verändern. Dies sind zum Beispiel *frameshift* Varianten, verschobene Startcodons oder verfrühte Stopcodons. Alle beobachteten Aminosäureaustausche werden zusammen mit dem Status (erfüllt oder nicht erfüllt) als einzelne Attribute an den Klassifikator übergeben.

Jedes Modell ist an den jeweils spezifischen Kontext angepasst und erhält nur solche Attribute als Parameter, die für die jeweils zu analysierenden Varianten auch beobachtet werden können (z.B. wird das Attribut *Proteinlänge* im Modell *without_aae* nicht zur Klassifizierung benutzt).

5 Training, Optimierung und Validierung

In diesem Kapitel werde ich darlegen, wie der Bayes Klassifikator trainiert, optimiert und validiert wurde, und welche Aspekte bei der Zusammenstellung der Trainingsdaten zu beachten waren. Außerdem werde ich die Durchführung und das Ergebnis eines Vergleichs von MutationTaster mit ähnlichen Vorhersageprogrammen beschreiben.

5.1 Training und Trainingsdaten

Damit der in MutationTaster integrierte Bayes Klassifikator unbekannte DNA-Veränderungen als harmlos oder krankheitsverursachend klassifizieren kann, muss er zunächst Informationen über typische Eigenschaften harmloser und krankheitsverursachender Varianten bekommen: Er muss mit Varianten mit bekanntem Krankheitspotential trainiert werden. Diese bekannten Varianten werden in ihrer Gesamtheit als Trainingsdatensatz bezeichnet. Ihre Zugehörigkeit zu einer bestimmten Klasse (*polymorphism* oder *disease causing*) ist bekannt.
Die spätere Vorhersagequalität des Klassifikators hängt maßgeblich von den gewählten Trainingsdaten ab. Idealerweise besteht ein Trainingsdatensatz aus vielen verschiedenen Fällen für eine jeweilige Klasse und außerdem aus möglichst vielen Fällen insgesamt. Dies maximiert die Wahrscheinlichkeit, dass sowohl häufige als auch seltene Fälle enthalten sind. Außerdem sollten die Fälle der verschiedenen Klassen zu gleichen Anteilen enthalten sein. Einen solchen Trainingsdatensatz mit Fällen bekannter Klassenzugehörigkeit zu erstellen, kann unter Umständen sehr schwierig sein.
In dem konkreten Fall für MutationTaster musste gewährleistet sein, dass möglichst kein Testfall der falschen Klasse zugeordnet wird, d.h. in der Klasse der harmlosen Varianten sollte sich möglichst keine krankheitsverursachende Variante befinden und umgekehrt. Außerdem sollte der Trainingsdatensatz in seiner Zusammensetzung ausgewogen sein, d.h. die unterschiedlichen Arten von DNA-Varianten zu ungefähr gleichen Anteilen enthalten. Die Wahrscheinlichkeit, dass ein bestimmtes Ergebnis vorkommt, eine Variante also entweder harmlos oder krankheitsverursachend ist, nennt man *a priori* Wahrscheinlichkeit. Bei unausgewogenen Datensätzen, wenn der Klassifikator z.B. mit mehr Polymorphismen als Krankheitsmutationen trainiert würde, wäre die *a priori* Wahrscheinlichkeit für Polymorphismen höher als für Krankheitsmutationen. Der Klassifikator könnte fälschlicherweise davon ausgehen, dass das Ergebnis „Krankheitsmutation“ an sich eher unwahrscheinlich ist.
Da MutationTaster nicht nur mit einem sondern mit drei verschiedenen Modellen (*without_aae*, *simple_aae*, *complex_aae*) arbeitet, wurden für die einzelnen Modelle auch separate Trainingsdatensätze angelegt: einer mit bekannten Krankheitsmutationen (*disease causing, dc*) und einer mit harmlosen Polymorphismen (*not disease causing, ndc*). Diese zwei Datensätze wurden wiederum in drei weitere Datensätze unterteilt, um die drei verschiedenen Vorhersagemodelle *without_aae*, *simple_aae* und *complex_aae* zu trainieren. Dafür haben

wir für die Datensätze für das *without_ aae* Modell ausschließlich synonyme bzw. intronische Krankheitsmutationen und Polymorphismen genutzt, für das *simple_ aae* Modell nur solche Trainingsfälle gewählt, die einen einfachen Aminosäureaustausch verursachen, und für das *complex_ aae* Modell kompliziertere Aminosäureaustausche. Damit ergeben sich insgesamt sechs verschiedene Datensätze für das Training (siehe auch Tabelle 5 für eine Übersicht über die gewonnenen Trainingsdaten):

- without_aae_ndc
- without_aae_dc
- simple_aae_ndc
- simple_aae_dc
- complex_aae_ndc
- complex_aae_dc

Der kritische Punkt bei der Erstellung der Trainingsdatensätze ist die Einteilung in die später vorherzusagenden Kategorien dc (*disease causing*) und ndc (*not disease causing*, d.h. *polymorphism*), weshalb ich die Einschlusskriterien unter 5.1.1 und 5.1.2 genauer erläutern werde.

5.1.1 Polymorphismen

Harmlose Polymorphismen für das Trainingsset ndc wurden aus den Daten des 1000-Genom-Projekts [66] (siehe Kapitel 2.2.1) gewonnen. Die Daten wurden im VCF-Format bezogen und in der MutationTaster Datenbank gespeichert. Einschlusskriterien für das MutationTaster ndc Trainingsset waren die folgenden:

- Anzahl Referenzallel homozygot > 50 (SNPs) bzw. > 20 (InDels)
- Anzahl Referenzallel / alternatives Allel heterozygot $> 50 / 20$
- Anzahl alternatives Allel homozygot $> 50 / 20$
- Variante ist in einem bekannten Gen lokalisiert

Wenn für eine Variante sowohl das Referenz- also auch das alternative Allel im homo- und heterozygoten Zustand in mehr als 50 bzw. 20 Proben detektiert wurden, kann man davon ausgehen, dass diese Variante nicht die Ursache für eine seltene, monogene Erkrankung ist. Sie ist somit für das ndc Trainingsset geeignet. Alle geeigneten Trainingsfälle wurden anschließend einer der drei Kategorien *without_ aae*, *simple_ aae* und *complex_ aae* zugeordnet und in den jeweiligen Trainingsdatensatz aufgenommen.

5.1.2 Krankheitsmutationen

Krankheitsmutationen für das Trainingsset dc wurden aus der kommerziellen Version der HGMD [27] (*HGMD Professional*, siehe Kapitel 2.2.1) bezogen. HGMD enthält verschiedene Mutationen, unter anderem *missense*-, *nonsense*- und Spleißmutationen, kleinere und größere

Deletionen, Insertionen und InDels. Jede Mutation ist mit einer Kennzeichnung (*tag*) versehen, welche die Krankheitsrelevanz reflektiert. Für MutationTaster wurden alle Mutationen benutzt, die als *DM*, d.h. Krankheitsmutation, gekennzeichnet und in einem bekannten Gen lokalisiert sind. Jede geeignete Mutation wurde einer der drei Kategorien *without_ aae*, *simple_ aae* und *complex_ aae* zugeordnet und in das jeweilige Trainingsset aufgenommen.

	without_aae		simple_aae		complex_aae	
Anzahl dc	122.238	(1,76%)	151.542	(87,84%)	123.213	(98,13%)
Anzahl ndc	6.807.269	(98,23%)	20.967	(12,15%)	2.340	(1,86%)
$\sum$ dc + ndc	6.929.507	(100%)	172.509	(100%)	125.553	(100%)

Tabelle 5: Für das Training des Bayes Klassifikators benutzte Datensätze. dc = *disease causing* = Krankheitsmutation; ndc = *not disease causing* = harmloser Polymorphismus; without_aae, simple_aae, complex_aae = Namen der drei verschiedenen Klassifikationsmodelle.

5.2 Optimierung

5.2.1 Verwendete Formeln

Zur Optimierung und Validierung des Klassifikators werden verschiedene statistische Berechnungen und Methoden herangezogen. Die Klassifikationsgüte wird üblicherweise mit der *accuracy* (Genauigkeit) angegeben [113]. Die *accuracy* $\left(\frac{TP+TN}{TP+TN+FP+FN}\right)$ist der Prozentsatz richtig klassifizierter Objekte [113], also die Erkennungsrate [114]. Dazu muss also die Anzahl an richtig positiven (*true positive*, TP), richtig negativen (*true negative*, TN), falsch positiven (*false positive*, FP) und falsch negativen (*false negative*, FN) Fällen bestimmt werden. Falsch positive Ergebnisse werden auch als Typ I Fehler (*type I error*), falsch negative auch als Typ II Fehler (*type II error*) bezeichnet. Außerdem ist es sinnvoll, zusätzlich zur *accuracy* die Sensitivität $\left(\frac{TP}{TP+FN}\right)$ und Spezifität $\left(\frac{TN}{TN+FP}\right)$ zu bestimmen. Die Sensitivität spiegelt den Anteil richtig positiver aus allen tatsächlich positiven Fällen wieder, die Spezifität gibt den Anteil der korrekt als negativ klassifizierten Objekten gemessen an allen in Wirklichkeit negativen Fällen an. Auch der positive Vorhersagewert (*positive prediction value*, PPV, oder *precision*), der den Anteil der richtig positiven von allen positiv vorhergesagten Ergebnissen darstellt $\left(\frac{TP}{TP+FP}\right)$ oder der negative Vorhersagewert (*negative prediction value*, NPV), der den Anteil richtig negativer an allen negativ klassifizierten Ergebnissen reflektiert $\left(\frac{TN}{TN+FN}\right)$ wird häufig zusätzlich zur *accuracy* angegeben.

5.2.2 Vorgehensweise zur Optimierung

Nach Fertigstellung der Trainingsdatensätze muss eine Vorauswahl der Attribute getroffen werden, die dem Klassifikator als Parameter übergeben werden. Diese Vorauswahl wird im Rahmen der Optimierung verfeinert. Generell muss man bedenken, dass es nicht sinnvoll ist, Attribute zu verarbeiten, die sich zwischen den einzelnen Klassen nur sehr wenig unterscheiden, also wenig diskriminant sind. Weiterhin sollten bestimmte Attribute aufbereitet werden, bevor sie an den Klassifikator übermittelt werden. Da das von uns verwendetet Perl Modul leider keine ordinalskalierten Werte verarbeiten kann, müssen diese dem Klassifikator entweder als verschiedene Klassen oder aber als normalverteilte Werte übergeben werden.
Um eine gute Parameterauswahl und -aufbereitung zu finden, wurden mehrere Parameterkombinationen und Aufbereitungsmöglichkeiten ausprobiert, indem der Klassifikator jeweils mit den verschiedenen Kombinationen trainiert und anschließend die Vorhersagequalität verglichen wurde. Zur Bestimmung der Vorhersagequalität wird der Klassifikator nach der Trainingsrunde erneut mit den bereits bekannten Trainingsfällen konfrontiert, ihm deren Klassenzugehörigkeit diesmal jedoch nicht mitteilt. Er muss nun sein gelerntes Wissen anwenden.
Für das Training und die Optimierung der Vorhersagemodelle in MutationTaster wurden unter anderem folgende Parameter optimiert:

- Schwellenwerte für die Berücksichtigung von vorhergesagten verlorenen / neugewonnen Spleißstellen: 0,3 / 0,6 / 0,9
- vorhergesagte relative Veränderung (stärker / schwächer) von Spleißstellen: Klassen / Normalverteilung
- Betrachtung der Spleißveränderung in ihrer Gesamtheit: additiv / absolut
- *a priori* Wahrscheinlichkeiten für dc / ndc: nicht identisch / identisch
- Aminosäureaustausch: tatsächlicher Austausch / Grantham-Wert

Es wurde von allen Modellen das robusteste gewählt, um zu vermeiden, dass zwar für eine statistische Berechnung ein sehr guter Wert erzielt wird, für eine andere aber ein sehr schlechter. Statt dessen sollte die Güte des Klassifikators, bezogen auf verschiedene Berechnungen, möglichst ausgewogen sein. Dazu wurden für alle drei Modell-Typen (*without_ aae*, *simple_ aae* und *complex_ aae*) aus allen getesteten Modellen (Parameterkombinationen) das Modell mit dem schlechtesten Wert für Sensitivität oder Spezifität ermittelt und der Parameterkombination zugeordnet. Aus diesen „schlechtesten“ Modellen wurde wiederum das Modell mit dem besten Wert für Sensitivität oder Spezifität gewählt. Das Modell der Wahl ist also nicht das Modell, welches die beste Sensitivität oder Spezifität aufweist, sondern das Modell, welches die am wenigsten schlechte Sensitivität oder Spezifität aufweist. Tabelle 6 zeigt die Ergebnisse der in der Optimierung als am robustesten identifizierten Modelle.

	without_aae		simple_aae		complex_aae	
dc	122.238	(1,76%)	151.542	(87,84%)	123.213	(98,13%)
ndc	6.807.269	(98,23%)	20.967	(12,15%)	2.340	(1,86%)
$\sum$ dc + ndc	6.929.507	(100%)	172.509	(100%)	125.553	(100%)
ndc richtig	6.533.390	(96,0%)	18.328	(87,4%)	2.010	(85,9%)
ndc falsch	273.879	(4,0%)	2.639	(12,6%)	330	(14,1%)
dc richtig	108.141	(88,5%)	135.839	(89,6%)	116.453	(94,5%)
dc falsch	14.097	(11,5%)	15.703	(10,4%)	6.760	(5,5%)
$\sum$ richtig	6.641.531	(95,8%)	154.167	(89,4%)	118.463	(94,4%)
$\sum$ falsch	287.976	(4,2%)	18.342	(10,6%)	7.090	(5,6%)

Tabelle 6: Ergebnisse von Training und Optimierung aller drei MutationTaster Vorhersagemodelle. dc = *disease causing* = Krankheitsmutation; ndc = *not disease causing* = harmloser Polymorphismus; dc richtig = korrekt als Krankheitsmutation klassifizierte Krankheitsmutation; dc falsch = fälschlich als Polymorphismus klassifizierte Krankheitsmutation; ndc richtig = korrekt als harmloser Polymorphismus klassifizierter harmloser Polymorphismus; ndc falsch = fälschlich als Krankheitsmutation klassifizierter harmloser Polymorphismus; without_aae, simple_aae, complex_aae = Namen der drei verschiedenen Klassifikationsmodelle.

5.3 Validierung

Nachdem die Optimierung zufriedenstellende Ergebnisse lieferte, wurde geprüft, wie gut das Modell unbekannte Fälle klassifizieren kann.

Häufig werden für die endgültige Validierung eines Klassifikators die Werte für die in Kapitel 5.2.1 beschriebenen statistischen Berechnungen zur Beschreibung der Vorhersagegüte mit einem unabhängigen Testdatensatz ermittelt, der nicht zum Trainieren eingesetzt wurde. Während für die Optimierung jeweils die gleichen Fälle für Training und anschließende Evaluierung benutzt werden können, soll für die Validierung sichergestellt werden, dass der Klassifikator mit unterschiedlichen Fällen trainiert und getestet wird [113]. In der Regel ist jedoch bereits die Zusammenstellung einer genügend großen Anzahl von Trainingsfällen sehr aufwendig und oft fehlen die Ressourcen, um weitere Fälle für einen komplett neuen Testdatensatz zu finden. Außerdem werden gerne so viele der Fälle wie möglich für das Training des Klassifikators genutzt, um so die Vorhersagequalität zu maximieren. Dieser Konflikt kann durch eine Kreuzvalidierung gelöst werden. Bei einer k-fachen Kreuzvalidierung werden dafür die Trainingsdaten in k möglichst gleich große Teilmengen $T_1, ..., T_k$ aufgeteilt und anschließend k Runden gestartet, bei denen die Daten der jeweils i-ten Teilmenge T_i als Testdaten und die Daten der verbleibenden *k-1* Teilmengen als Trainingsdaten verwendet werden. Für die k Einzeldurchläufe können die in Kapitel 5.2.1 beschriebene statistischen Mittel berechnet und der Durchschnitt für ein Gesamtergebnis ermittelt werden.

Obwohl bei einem Bayes Klassifikator eine Kreuzvalidierung nicht zwingend notwendig ist,

da im Modell lediglich Frequenzen und nicht Merkmalskombinationen der einzelnen Trainingsfälle gespeichert werden, wurde auch der MutationTaster Bayes Klassifikator mit einem unabhängigen Testdatensatz validiert. Für die 5-fache Kreuzvalidierung haben wir aus den vorhandenen Datensätzen jeweils eine bestimmte Anzahl von Fällen (4.000 für *simple_ aae* und *without_ aae*, 400 für *complex_ aae*; Polymorphismen und Krankheitsmutationen zu gleichen Teilen) zufällig entfernt, und den Klassifikator mit den verbliebenen Fällen trainiert. Mit den zuvor entfernten Fällen wurde anschließend die Vorhersagequalität validiert. Dieser Vorgang wurde insgesamt fünf Mal wiederholt. Aus den *accuracy* Werten der fünf einzelnen Runden kann ein Durchschnittswert gebildet werden. Der Vorteil dieser Methode ist, dass so jeder Fall theoretisch sowohl als Training- als auch als Testfall genutzt werden kann, aber kein Fall für Training und Validierung gleichzeitig verwendet wird. Der Nachteil ist, dass der Trainings- und Validierungsprozess mehrere Male wiederholt werden muss, was unter Umständen sehr zeitaufwendig sein kann.
Genotyp- und Allelfrequenzen von Polymorphismen wurden für die Kreuzvalidierung nicht als Parameter für die Klassifikation genutzt, da nach diesen Kriterien das Trainingsset zusammengestellt worden ist. Die im Internet verfügbare Version von MutationTaster nutzt diese Informationen jedoch sehr wohl, um harmlose Varianten zu identifizieren.
Das Ergebnis der 5-fachen Kreuzvalidierung der drei einzelnen Vorhersagemodelle wird in Abbildung 9 gezeigt. Mit dem *without_ aae* Modell konnten wir eine *accuracy* von 92,2%, mit dem *simple_ aae* Modell eine *accuracy* von 88,6% und mit dem *complex_ aae* Modell eine *accuracy* von 90,7% erreichen. Die durchschnittliche *accuracy* liegt somit bei 90,5%. Tabelle 7 zeigt die Ergebnisse der Kreuzvalidierung im Detail.

	without_aae	simple_aae	complex_aae
NPV (%)	95,7 ± 0,3	87,7 ± 0,8	86,9 ± 3,2
PPV (%)	88,8 ± 0,6	89,5 ± 0,5	94,4 ± 0,4
Sensitivität (%)	95,4 ± 0,4	87,9 ± 0,7	87,9 ± 2,6
Spezifität (%)	89,5 ± 0,5	89,3 ± 0,4	93,9 ± 0,5
accuracy (%)	92,2 ± 0,4	88,6 ± 0,4	90,7 ± 1,7

Tabelle 7: Ergebnisse der 5x Kreuzvalidierung aller drei MutationTaster Vorhersagemodelle. NPV = *negative prediction value* / negativer Vorhersagewert; PPV = *positive prediction value* / positiver Vorhersagewert; simple_aae, without_aae, complex_aae = Namen der drei verschiedenen Klassifikationsmodelle.

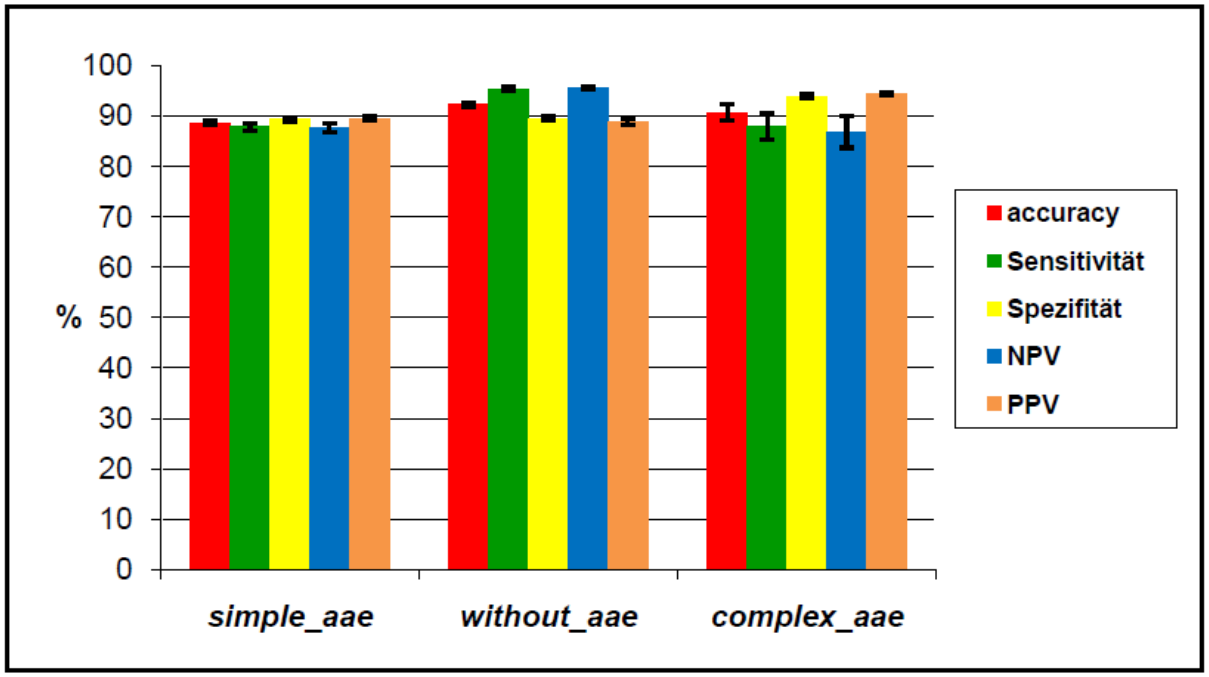


Abbildung 9: Ergebnisse der 5x-Kreuzvalidierung aller drei MutationTaster Vorhersagemodelle. NPV = *negative prediction value* / negativer Vorhersagewert; PPV = *positive prediction value* / positiver Vorhersagewert; simple_aae, without_aae, complex_aae = Namen der drei verschiedenen Klassifikationsmodelle.

5.4 Vergleich von MutationTaster mit ähnlichen Programmen

Um den Nutzen von MutationTaster besser einordnen zu können, habe ich die Software mit ähnlicher, bereits verfügbarer Vorhersagesoftware, verglichen. Zu den etabliertesten und bekanntesten Programmen zählen PolyPhen-2 [46] und SIFT [47], die jedoch keine Nukleotidaustausche, sondern nur Aminosäureaustausche bewerten können. Daher konnte ich auch nur eines der drei MutationTaster Vorhersagemodelle, das *simple_aae* Modell, im Vergleich mit diesen Programmen testen. SIFT und PolyPhen-2 können außerdem nur einfache Aminosäureaustausche und keine InDels prozessieren. PROVEAN [48] ist eine Weiterentwicklung von SIFT und kann InDels, die das Leseraster nicht verändern, analysieren.
Ich habe daher zwei Testdatensätze mit jeweils 3.600 einfachen Aminosäureaustauschen (1.800 harmlose Polymorphismen aus dem 1000-Genom-Projekt und 1.800 Krankheitsmutationen einmal aus HGMD *Professional* und einmal aus *ClinVar*) an die Programme PolyPhen-2 (HumVar und HumDiv Modell), SIFT, PROVEAN und MutationTaster geschickt. Ausgehend von den Aminosäureaustauschen aus dem Testdatensatz wurden für alle Programme die Anfragen auf DNA-Ebene gestellt (Chromosom, Position, Referenzallel, alternatives Allel). Die Option, eine Variante in mehreren Transkripten zu analysieren, wurde für alle Programme aktiviert. Aus den Ergebnissen (teilweise gab es mehrere Transkripte und somit mehrere Vorhersagen für eine Variante) wurden dann diejenigen extrahiert, die eine Vorher-

sage für exakt den Aminosäureaustausch aus dem Testset enthielten. Nicht alle Testfälle konnten in allen Programmen im gleichen Transkript bearbeitet werden, weshalb jeweils 1.300 Polymorphismen und 1.300 Mutationen aus den insgesamt 2.957 (HGMD) bzw. 2814 (ClinVar) in allen Programmen vorhergesagten Varianten ausgewählt wurden. Die Ergebnisse für diese 2.600 Testfälle wurden verglichen (siehe Tabelle 8). Da MutationTaster eine automatische Vorhersage für bekannte harmlose Polymorphismen aus dem 1000-Genom-Projekt und für bekannte Krankheitsmutationen aus NCBI ClinVar anzeigt, habe ich für den Vergleich mit den anderen Programmen nicht die automatische Vorhersage berücksichtigt, sondern die tatsächliche, vom Klassifikator gemachte Vorhersage, die durch den Wahrscheinlichkeitswert (*probability value*) reflektiert wird. Eine automatische Vorhersage mit einem Wahrscheinlichkeitswert kleiner als 0,5 bedeutet, dass sich der Bayes Klassifikator für das Gegenteil der automatischen Vorhersage entschieden hätte (siehe auch Kapitel 3.3).

Programm	$\sum$	tp	tn	fp	fn	NPV	PPV	Sensitivität	Spezifität	accuracy
1000G und HGMD										
PPH2-var	2600	1036	1159	141	264	81,4%	88,0%	79,7%	89,2%	**84,4%**
PPH2-div	2600	1120	1076	224	180	85,7%	83,3%	86,2%	82,8%	**84,5%**
PROVEAN	2600	1030	1145	155	270	80,9%	87,0%	79,2%	88,2%	**83,7%**
SIFT	2600	1079	1123	177	221	83,6%	85,9%	83,0%	86,4%	**84,7%**
MT	2600	1157	1132	168	143	88,8%	87,3%	89,0%	87,1%	**88,0%**
1000G und ClinVar										
PPH2-var	2600	1108	1159	141	192	85,8%	88,7%	85,2%	89,2%	**87,2%**
PPH2-div	2600	1175	1076	224	125	89,6%	84,0%	90,4%	82,8%	**86,6%**
PROVEAN	2600	1096	1146	154	204	84,9%	87,7%	84,3%	88,2%	**86,2%**
SIFT	2600	1136	1123	177	164	87,3%	86,5%	87,4%	86,4%	**86,9%**
MT	2600	1213	1132	168	87	92,9%	87,8%	93,3%	87,1%	**90,2%**

Tabelle 8: Vergleich von MutationTaster mit anderen Vorhersageprogrammen. PPH2-var = PolyPhen-2 HumVar Modell; PPH2-div = PolyPhen-2 HumDiv Modell; MT = MutationTaster; NPV = *negative prediction value* / negativer Vorhersagewert; PPV = *positive prediction value* / positiver Vorhersagewert; 1000G und HGMD = Testfälle stammen aus dem 1000-Genom-Projekt und aus HGMD *Professional*; 1000G und ClinVar = Testfälle stammen aus dem 1000-Genom-Projekt und ClinVar. Es wurden die identischen 1000G Testfälle für beide Vergleiche genutzt.

Für beide Vergleichsdatensätze konnte MutationTaster die meisten Fälle richtig vorhersagen. Bei Verwendung des Datensatzes mit Krankheitsmutationen aus HGMD erreichte MutationTaster eine *accuracy* von 88,0%, gefolgt von SIFT (84,7%), (PolyPhen-2 HumDiv (84,5%), PolyPhen-2 HumVar (84,4%) und PROVEAN (83,7%). Für den Datensatz

mit Krankheitsmutationen aus ClinVar erreichte MutationTaster eine *accuracy* von 90,2%, gefolgt von PolyPhen-2 HumVar (87,2%), SIFT (86,9%), PolyPhen-2 HumDiv (86,6%) und PROVEAN (86,2%). Es fällt auf, dass bei der Vorhersage der Krankheitsmutationen aus ClinVar alle Programme besser abschneiden als bei der Vorhersage der Krankheitsmutationen aus HGMD. Das legt die Vermutung nahe, dass Krankheitsmutationen in ClinVar einen eindeutigeren Effekt auf das Protein haben und somit besser zu klassifizieren sind. Dies könnte daran liegen, dass die Datenressource ClinVar noch relativ jung ist, und zunächst die klar belegten und besonders eindeutigen Krankheitsmutationen eingeschlossen worden sind, während weniger klare Fälle bislang noch nicht dort gespeichert werden.

6 MutationTaster Zusatzprogramme

In Kapitel 3 wird ein typischer MutationTaster Programmlauf nachvollzogen. Startpunkt ist dabei das MutationTaster *web interface* für Einzelabfragen, in dem eine Variante immer bezüglich eines bestimmten Transkripts spezifiziert werden muss. Es gibt jedoch noch weitere Möglichkeiten, DNA-Varianten mit MutationTaster zu analysieren, ohne dass bereits im Voraus ein bestimmtes Transkript festgelegt werden muss. Diese Möglichkeiten werde ich in diesem Kapitel vorstellen.

6.1 Einzelabfrage über die chromosomale Position

Ein zusätzliches *web interface* bietet die Möglichkeit, Varianten in Bezug auf das Chromosom und die chromosomale Position einzugeben (siehe Abbildung 10).
Der Vorteil ist, dass die Variante nicht nur in einem, im Vorfeld festgelegten Transkript, sondern in allen möglichen, die chromosomale Position überspannenden, Transkripten analysiert wird. MutationTaster prüft über die Angabe der chromosomalen Position automatisch, welche Gene und Transkripte die Variante betrifft, berechnet die entsprechenden Transkript-spezifischen Positionen und führt für jedes einzelne die übliche MutationTaster Analyse durch. Die Ergebnisseite im HTML-Format beginnt mit einer Tabelle, in der alle analysierten Transkripte mit ihren Vorhersagen zusammengefasst sind. Anschließend folgen untereinander, nach absteigendem Krankheitspotential sortiert, die üblichen, detaillierten MutationTaster Ergebnisseiten (siehe Abbildung 11).

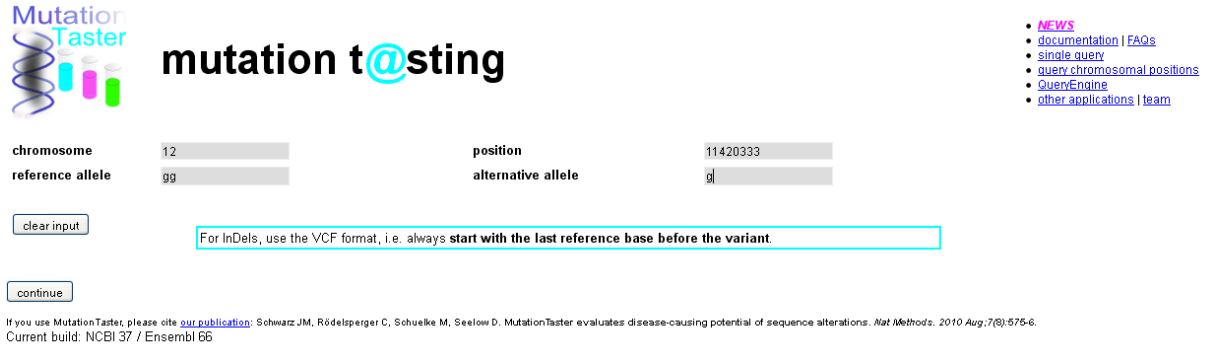
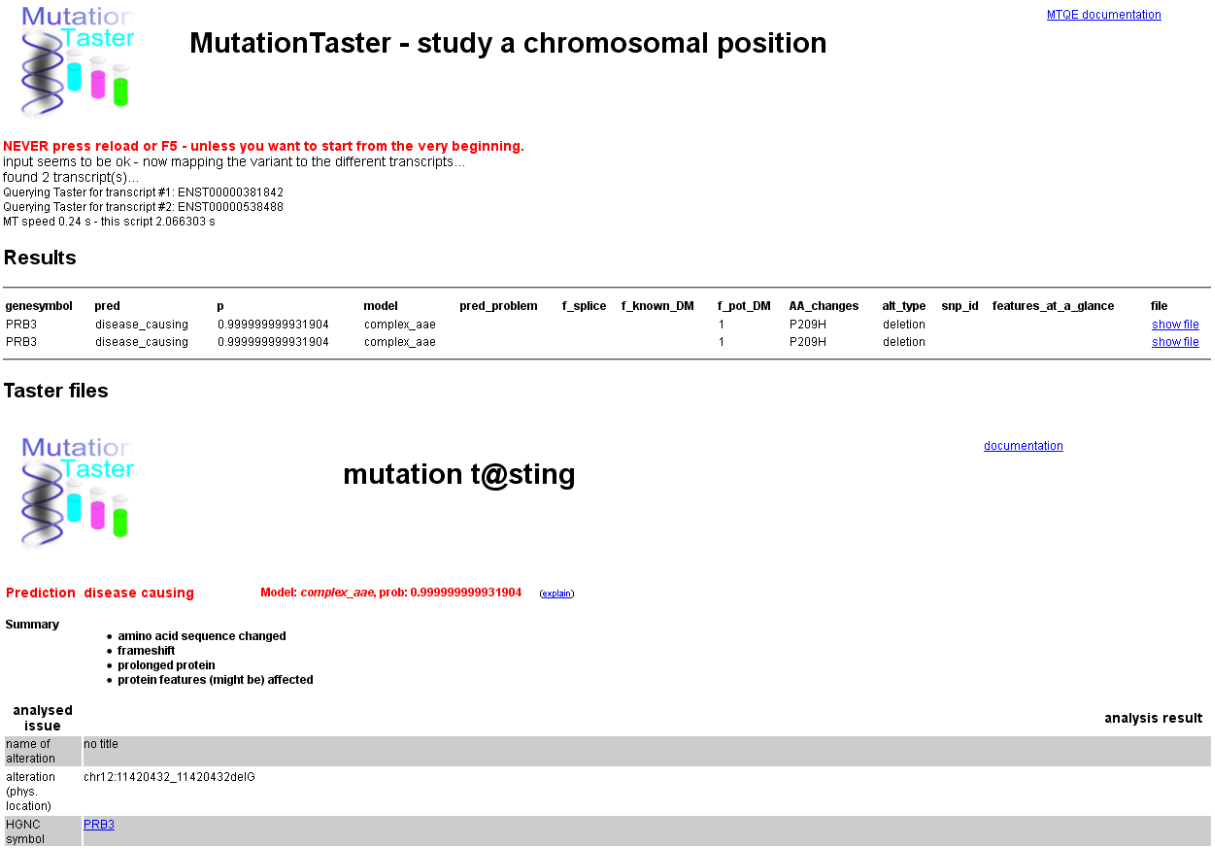


Abbildung 10: MutationTaster Startseite für Einzelabfragen basierend auf einer chromosomalen Position.



MTQE documentation

MutationTaster - study a chromosomal position

NEVER press reload or F5 - unless you want to start from the very beginning.
input seems to be ok - now mapping the variant to the different transcripts...
found 2 transcript(s)...
Querying Taster for transcript #1: ENST00000381842
Querying Taster for transcript #2: ENST00000538488
MT speed 0.24 s - this script 2.066303 s

Results

genesymbol	pred	p	model	pred_problem	f_splice	f_known_DM	f_pot_DM	AA_changes	alt_type	snp_id	features_at_a_glance	file
PRB3	disease_causing	0.999999999931904	complex_aae				1	P209H	deletion			show file
PRB3	disease_causing	0.999999999931904	complex_aae				1	P209H	deletion			show file

Taster files

documentation

mutation t@sting

Prediction disease causing **Model: *complex_aae*, prob: 0.999999999931904** (explain)

Summary

- amino acid sequence changed
- frameshift
- prolonged protein
- protein features (might be) affected

analysed issue	analysis result
name of alteration	no title
alteration (phys. location)	chr12:11420432_11420432delG
HGNC symbol	PRB3

Abbildung 11: MutationTaster Ergebnisseite für die Analyse einer Variante mit chromosomaler Position in unterschiedlichen Transkripten (nur der obere Teil ist dargestellt).

6.2 MutationTaster Query Engine

Ein NGS-Projekt ist ein mehrstufiger Prozess aus (1) der Sequenzierung (dabei entstehen viele einzelne kurze DNA-Sequenzen, die sogenannten *reads*), (2) der Alignierung der *reads* an eine Referenzsequenz um so eine regional zusammenhängende DNA-Sequenz der untersuchten Probe zu erhalten, (3) der Variantensuche (dazu werden das Alignment der *reads* und eine Referenzsequenz auf Unterschiede untersucht) und (4) der Variantenbewertung. Das Ergebnis von Schritt 3, der Variantensuche, ist häufig eine Datei im VCF-Format, die hunderttausende Varianten (eine Position pro Zeile) enthalten kann, die anschließend in Schritt 4 genauer untersucht werden müssen. Es wäre sehr unpraktisch und zeitaufwendig, wenn all diese Varianten von Hand in das MutationTaster *web interface* für Einzelabfragen mit chromosomaler Position (siehe Kapitel 6.1) eingetippt werden müssten. Um eine bequeme und simultane Analyse der NGS-Ergebnisse im VCF-Format zu ermöglichen, habe ich die MutationTaster QueryEngine (MTQE) entwickelt. Die QueryEngine ist ein System, das eine VCF-Datei einlesen, und deren Inhalte zeilenweise prozessieren kann. Ein in die MTQE integrierter *Job Scheduler**, eine Software zum Steuern von Aufgaben (Jobs), sorgt nach festgelegten Kriterien für eine sinnvolle Reihenfolge und Parallelisierung der einzelnen

Aufgaben. Die Startseite der QueryEngine bietet unter anderem die in der folgenden Tabelle 9 dargestellten Filteroptionen und Einstellungen (siehe auch Abbildung 13):

Filter	
	nur homozygote Varianten
	nur heterozygote Varianten
	nur exonische Varianten (plus X bp vom Intron)
	nur Varianten auf benutzerdefiniertem Chromosom
	nur Varianten in benutzerdefinierte(r/n) Region(en)
	nur Varianten außerhalb von benutzerdefinierte(r/n) Region(en)
	homozygote Varianten aus 1000G ausschließen (benutzerdefinierter Schwellenwert)
	heterozygote Varianten aus 1000G ausschließen (benutzerdefinierter Schwellenwert)
	nur Varianten, die mindestens X-fach abgedeckt sind
Option	
	Kombination mehrerer benachbarter Varianten zu einer einzigen Variante
	keine HTML Dateien speichern (schneller)

Tabelle 9: Filtermöglichkeiten und Analyseoptionen in der MutationTaster QueryEngine. 1000G = 1000-Genom-Projekt

Der Benutzer kann eine VCF-Datei auf seinem Rechner zur Verarbeitung auswählen, Filtermöglichkeiten und Analyseoptionen einstellen und über einen Klick auf *Submit* die Anfrage absenden. Wenn gewünscht, wird er daraufhin per E-Mail über seine Anfrage und die Internetadresse zu seinem Projekt informiert. Die sich als nächstes öffnende HTML-Seite wird automatisch aktualisiert und informiert den Benutzer über den Fortschritt seiner Anfrage. Über einen Hyperlink gelangt er zu einer Statusanzeige über den Auslastungsgrad der MTQE. Diese liest zunächst die VCF-Datei Zeile für Zeile ein und überprüft jede einzelne Variante. Bestimmte Varianten werden aussortiert, zum Beispiel solche, die doppelt vorhanden sind, im falschen Format vorliegen oder bei denen Referenz- und alternatives Allel identisch sind. Auch Varianten, die unter die vom Benutzer definierte Mindestabdeckung fallen, werden herausgefiltert. Die aussortierten Varianten werden in eine Datei geschrieben, die der Benutzer am Ende herunterladen kann. Alle anderen Varianten werden in der speziellen MTQE-Datenbank gespeichert.
In dieser Datenbank werden für jedes Projekt (also für jede einzelne Benutzeranfrage) eigene Tabellen angelegt (siehe Abbildung 12 für die Datenbankstruktur), die abhängig von der Anzahl der Varianten in der VCF-Datei unter Umständen recht groß werden können (mehrere Millionen Zeilen). Die Anfragen in separaten Tabellen zu speichern, hat den Vorteil, dass Datenbankabfragen schneller sind, als wenn alle Projekte in einer Tabelle gespeichert würden, weil so zeitaufwendige Suchen und insbesondere komplizierte Verknüpfungen über große

Tabellen entfallen. Außerdem ist die Wartung und Archivierung separater Tabellen wesentlich einfacher. Da das Anlegen von Fremdschlüsseln den Schreibvorgang in eine Tabelle verlangsamt, wurden keine expliziten Fremdschlüssel angelegt. Eine Verletzung der referenziellen Integrität ist ausgeschlossen, da die Daten vor dem Speichern durch die Applikation überpürft und später nicht aktualisiert oder verändert werden. Zudem wird durch die Transkationskontrolle (siehe Kapitel 2.3.1) sichergestellt, dass nur komplette Datensätze gespeichert werden.

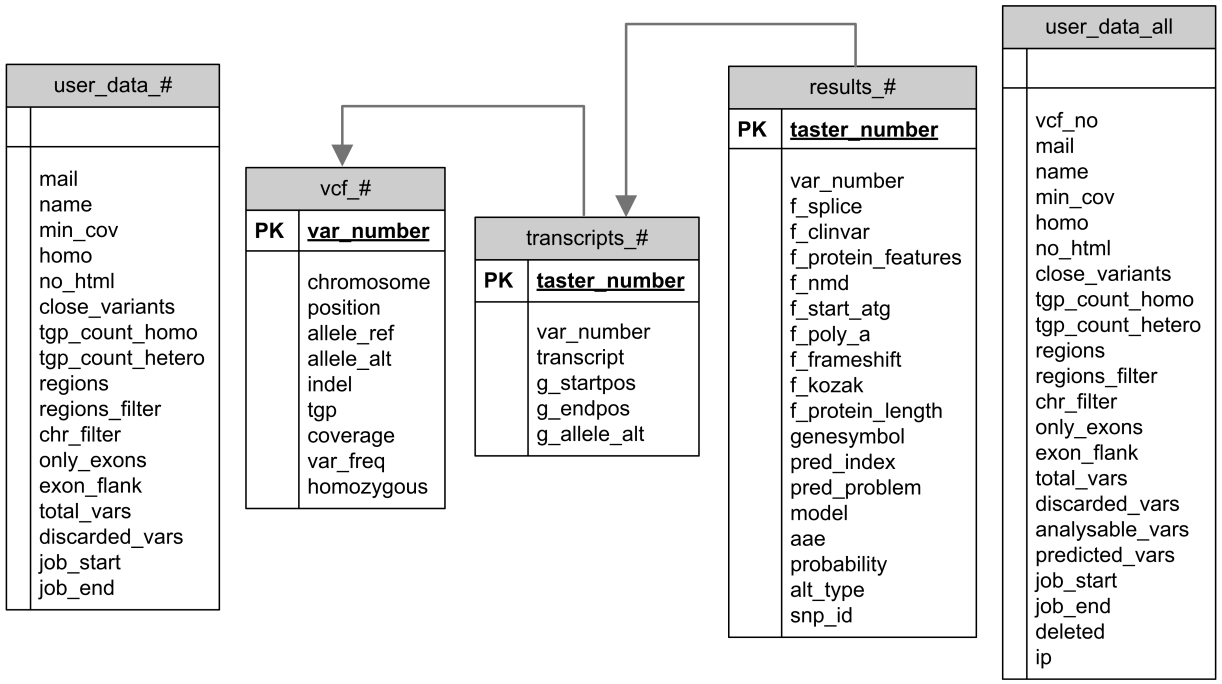


Abbildung 12: Datenbanktabellen der MutationTaster QueryEngine.

Die in der Datenbank gespeicherten Varianten werden nun genutzt, um die der Positionsannotation zugrunde liegende Genomversion zu überprüfen. Die MTQE kann nur Varianten verarbeiten, die sich auf die derzeit aktuelle Genomversion GRCh37 beziehen. Wenn die Varianten in der VCF-Datei nicht GRCh37 entsprechen, wird der Prozess abgebrochen und der Benutzer mit einer entsprechenden Fehlermeldung informiert.
Falls die Genomversion aktuell ist, werden als nächstes die Varianten aus der VCF-Datei auf Übereinstimmung mit bekannten Varianten aus dem 1000-Genom-Projekt überprüft und gemäß den vom Benutzer vorgenommenen Filtereinstellungen unter Umständen herausgefiltert. Anschließend sucht MutationTaster für jede einzelne Variante nach durch sie betroffenen Genen und Transkripten und berechnet die jeweiligen Transkript-spezifische Positionen. Hier werden entsprechend der Benutzereinstellungen gegebenenfalls weitere Varianten herausgefiltert, z.B. solche die außerhalb einer benutzerdefinierten Region liegen. Durch die Zuordnung der Varianten zu verschiedenen Transkripten ist die Anzahl der resultierenden MutationTaster Analysen in der Regel deutlich höher als die ursprüngliche Zahl der

Varianten in der VCF-Datei. Das liegt daran, dass für viele Gene mehrere Transkripte existieren, und eine Variante somit in Bezug auf mehrere Transkripte analysiert werden muss. Es werden jedoch nicht alle in Ensembl annotierten Transkripte verwendet. Falls es für ein Gen ein oder mehrere protein-kodierende und andere Transkripte gibt, werden die protein-kodierenden Transkripte zur Analyse herangezogen. Nur wenn für ein Gen kein protein-kodierendes Transkript existiert, werden andere Transkripte in Betracht gezogen, wobei viele Transkript-Typen von vorneherein von der Analyse ausgeschlossen werden (z.B. *nonsense_ mediated_ decay-, ambiguous_ orf-* oder Pseudogen-Transkripte). Die durchzuführ-

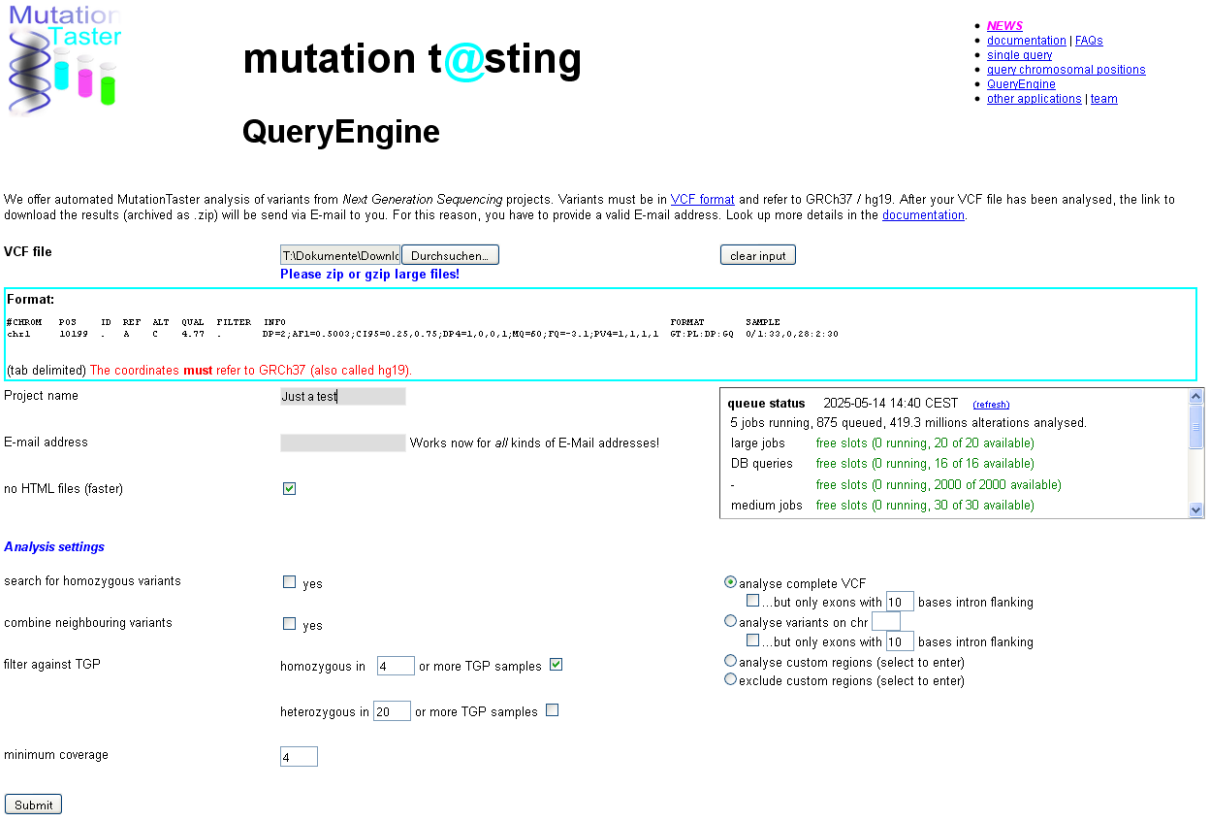


Abbildung 13: MutationTaster QueryEngine Startseite.

enden MutationTaster Analysen werden in größere Pakete (Jobs) gebündelt (je nach Gesamtanzahl der Varianten in der VCF-Datei zwischen zehn und 25.000 pro Paket). Der *Job Scheduler* sorgt nun für eine parallele Abarbeitung der Jobs, so dass viele Analysen gleichzeitig ausgeführt werden können. Die Kontrolle und sequentielle Abarbeitung der Jobs erfolgt nach bestimmten Kriterien. Je nach Größe der ursprünglichen VCF-Datei erhalten die Pakete unterschiedliche Prioritäten: je größer die VCF-Datei, desto geringer die Priorität. Die Priorität entscheidet darüber, in welche Schlange die Jobs eingereiht werden: Es gibt eine Schlange für sehr große VCF-Dateien (mehr als zwei Millionen Zeilen, geringste Priorität), für große VCF-Dateien (mehr als 10.000 Zeilen), für normale VCF-Dateien (mehr als 500 Zeilen) und für kleine VCF-Dateien (weniger als 500 Zeilen, höchste Priorität). In der

Schlange mit der geringsten Priorität werden Jobs mit sehr vielen einzelnen Analysen eingereiht (bis zu 50.000 Analysen pro Job), in der Schlange mit der höchsten Priorität landen Jobs mit wenigen Analysen (bis zu 100). Kleinere VCF-Dateien sollen bevorzugt prozessiert werden, damit Benutzer, die nur wenige Varianten analysieren möchten, nicht auf die Fertigstellung sehr großer Anfragen warten müssen. Deshalb werden in der Schlange für sehr große Anfragen nur maximal fünf Prozesse gleichzeitig ausgeführt, während in der Schlange für kleine Anfragen maximal 30 Prozesse parallelisiert werden. Kleinere Anfragen werden somit unabhängig von größeren Anfragen abgearbeitet. Für Datenbankabfragen gibt es ebenfalls eine separate Schlange, in der bis zu acht Prozesse parallelisiert werden können. Manchmal gibt MutationTaster für eine Analyseanfrage keine Ergebnisse aus (z.B. aufgrund eines Paketverlusts während der Datenübertragung), weshalb ein Job nach dessen Fertigstellung daraufhin überprüft wird, ob für alle zu analysierenden Varianten Ergebnisse vorliegen. Falls dies nicht der Fall ist, werden fehlende Analysen maximal vier Mal wiederholt.
Die MutationTaster Ergebnisse werden in eine eigene Tabelle in der MTQE Datenbank geschrieben und später nach benutzerdefinierten Einstellungen in tabellarischer Form ausgegeben. Auf der Ergebnisseite (siehe Abbildung 14) links platziert befindet sich eine Zusammenfassung des Auftrags und der Ergebnisse, die unter anderem über folgendes informiert:

- Zahl der Varianten im VCF
- Zahl der vorgefilterten Varianten
- Zahl der analysierbaren Varianten
- Zahl der nach 1000-Genom-Projekt gefilterten Varianten
- Zahl der nach benutzerdefinierter Region gefilterten Varianten
- Zahl der analysierten Varianten
- Zahl der resultierenden MutationTaster Fälle
- Benutzereinstellungen (z.B. Filteroptionen)

Auf der Ergebnisseite (siehe Abbildung 14) rechts befinden sich Optionen zum Filtern und Sortieren der Ergebnisse. So können die Ergebnisse beispielsweise geordnet nach vorhergesagtem Krankheitspotential, nach dem verwendeten Klassifikationsmodell, nach Chromosom, Position oder Genname angezeigt werden. Es ist möglich, synonyme Varianten, bekannte Polymorphismen (homozygoter Genotyp liegt mehr als vier Mal im 1000-Genom-Projekt vor) oder alle Ergebnisse mit der Vorhersage *polymorphism* auszublenden. Die Filteroptionen können miteinander kombiniert werden. Die (eventuell sortierten und gefilterten) Ergebnisse können entweder direkt im Web-Browser angeschaut werden, oder im *TSV-Format** als *Zip-Archiv** heruntergeladen werden. Außerdem werden drei verschiedene Optionen zum direkten Aufruf unserer Kandidatengensuchmaschine GeneDistiller [85] mit den von MutationTaster analysierten Varianten angeboten: Aufruf mit (1) allen als *disease causing* klassifizierten Varianten; (2) allen nicht-synonymen, als *disease causing* klassifizierten Varianten und (3) allen nicht-synonymen Varianten (außer bekannten Polymorphismen aus dem 1000-Genom-

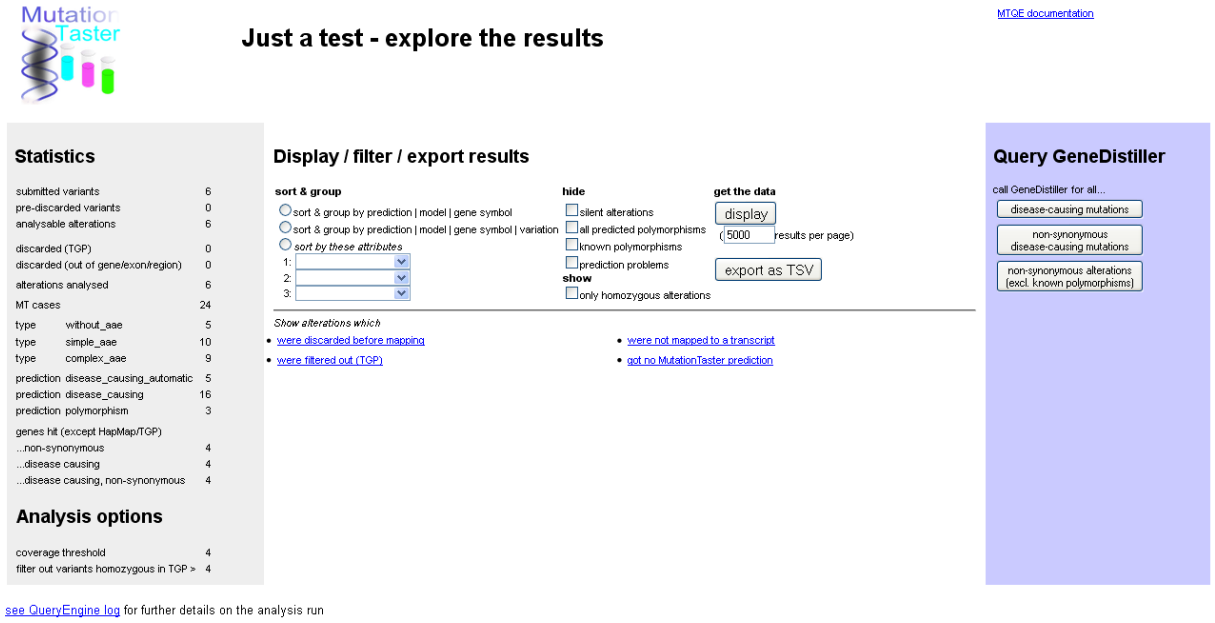


Abbildung 14: MutationTaster QueryEngine Ergebnisseite.

Projekt). Die QueryEngine kann etwa 500.000 Varianten pro Stunde analysieren. Abbildung 15 zeigt in vereinfachter Form die Abläufe während eines QueryEngine Laufs.

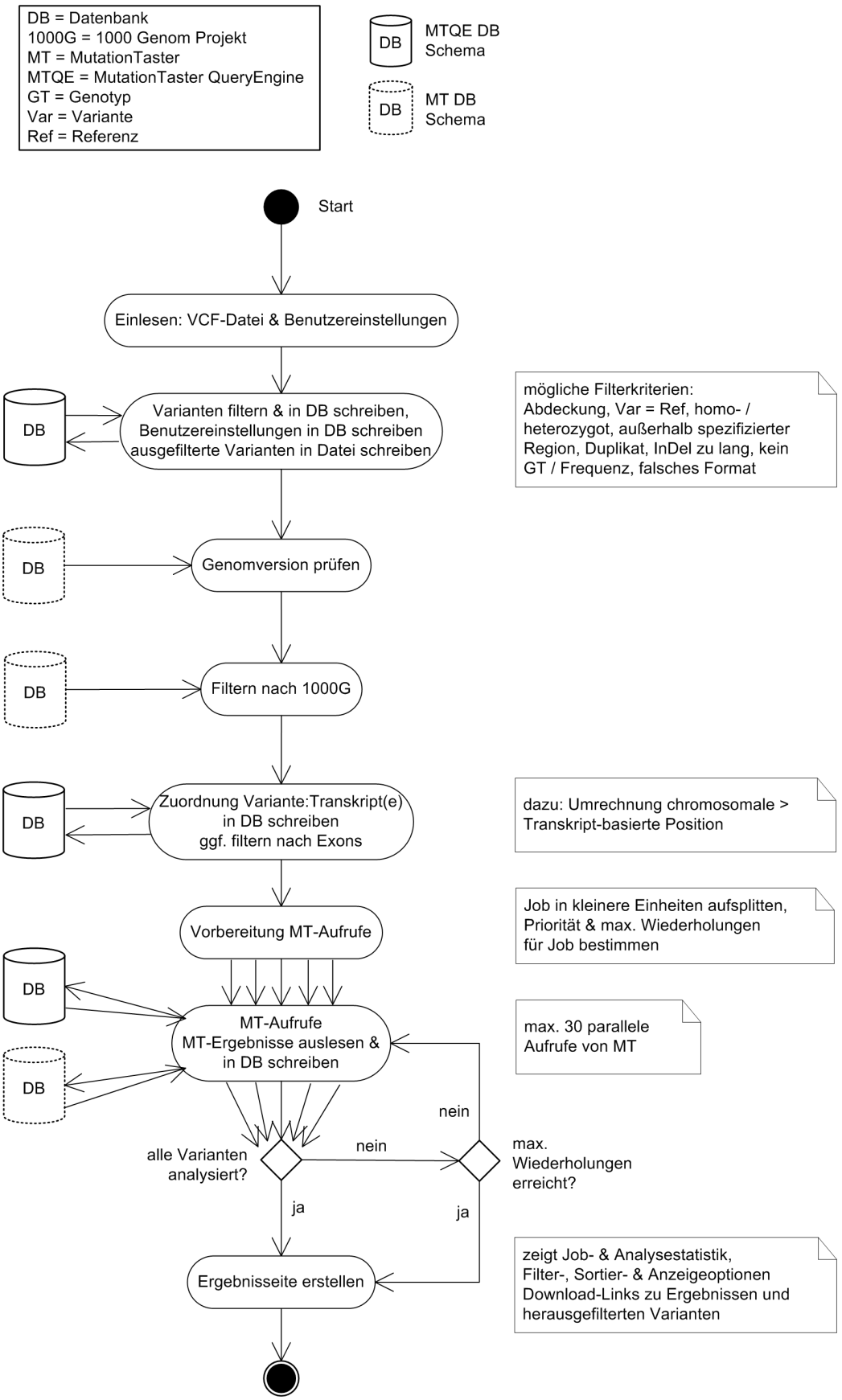


Abbildung 15: Vereinfachte Darstellung der Abläufe während eines einzelnen MutationTaster QueryEngine Laufs.

7 Implementierung

7.1 Hardware

Alle MutationTaster Anwendungen laufen unter Linux Fedora Core 14 auf einem Intel Xeon 24 CPU Server mit 72 GB RAM. Als Webserver dient Apache 2.2.9 mit mod_perl und als RDBMS (*r*elational *d*ata*b*ase *m*anagement *s*ystem) PostgreSQL 8.4.

7.2 Software-Entwicklung

MutationTaster ist eine iterativ entwickelte Software. Das bedeutet, dass die Entwicklung nicht gemäß einem von Anfang detailliert geplanten und festgelegtem Konzept folgte, sondern vielmehr im Laufe der Zeit immer wieder den Bedürfnissen der Benutzer und den durch technische Fortschritte geänderten Anforderungen angepasst wurde.

7.2.1 MutationTaster

Das MutationTaster Computerprogramm wurde in der Programmiersprache Perl geschrieben. Ein zentrales Modul (MutationTaster.pm) enthält alle relevanten Subroutinen bzw. Funktionen. Zum Teil wurden bereits existierende, frei verfügbare Perl Module eingebunden: für den für die Bewertung benutzten naiven Bayes Klassifikator *AI::NaiveBayes1*, für die Interaktion mit der Datenbank *DBI*, für die Laufzeitoptimierung *Time::HiRes*, für die interne Speicherung und das Lesen der Modelle *FreezeThaw* und für das Auslesen der Benutzereingaben aus der HTML-Startseite *CGI*. Module, die nicht über die Paketverwaltung des Betriebssystems installiert werden konnten, wurden über CPAN (*C*omprehensive *P*erl *A*rchive *N*etwork)[2] bezogen.

7.2.2 MutationTaster QueryEngine

Die MutationTaster QueryEngine wurde ebenfalls in Perl geschrieben. Das *Job-Scheduling* (Zeitplanerstellung) wird über die in das Betriebssystem integrierte, frei verfügbare Software TORQUE Resource Manager[3] abgewickelt. Es gibt ein zentrales Perl Modul *MTQE.pm* sowie eine Reihe einzelner Perl-Skripte, die über Shell-Skripte nacheinander aufgerufen und aus denen Jobs an TORQUE übermittelt werden. Für das Auslesen der Benutzereingaben aus der HTML-Startseite wird das Perl Modul *CGI* verwendet, für die Kommunikation mit TORQUE das Perl Modul *PBS::Client*.

[2] `www.cpan.org`
[3] `http://www.adaptivecomputing.com/products/open-source/torque/`

7.2.3 Webschnittstellen

Die zentrale MutationTaster Startseite `http://www.mutationtaster.org` ist eine statische HTML-Seite. Dynamische Daten (die Anzeige verfügbarer Transkripte und des Sequenzschnipsels) werden über AJAX und Perl / CGI (siehe Kapitel 3.1) eingebunden. Die HTML-Ergebnisseite wird von einer Funktion im zentralen MutationTaster Perl Modul dynamisch erstellt. Die Webschnittstellen wurden mit Firefox entwickelt und für die jeweils aktuelle Firefox-Version optimiert, jedoch auch mit Internet Explorer 8 und Google Chrome unter Windows XP, Mac OS X, CentOS 6 und Linux Ubuntu getestet.
Über eine auf der zentralen MutationTaster Startseite platzierte Navigationsleiste sind die ebenfalls statischen HTML-Startseiten der MutationTaster Zusatzprogramme (Einzelabfragen über die chromosomale Position, siehe Kapitel 6.1, und die MutationTaster QueryEngine, siehe Kapitel 6.2) erreichbar. Deren HTML-Ergebnisseiten werden ebenfalls jeweils dynamisch mit Perl erzeugt.

8 Diskussion

8.1 Auswahl und Zusammenstellung der Trainingsdaten

Die Vorhersagequalität eines Bayes Klassifikators hängt wesentlich von den zum Training benutzten Daten ab. Um einen Bayes Klassifikator für die Vorhersage des Krankheitspotentials von DNA-Sequenzveränderungen zu erstellen, benötigt man für das Training einerseits bekannte Mutationen und andererseits harmlose Polymorphismen.

8.1.1 Auswahl der Trainingsdaten

Polymorphismen

Die Polymorphismen für das Training des MutationTaster Klassifikators wurden aus den Daten des 1000-Genom-Projekts [32] gewonnen. Die meisten monogenen Erkrankungen werden durch extrem seltene, hoch penetrante Allele (Allelfrequenz $<< 1\%$) verursacht [115]. Man kann generell davon ausgehen, dass seltene Varianten eher eine Rolle in der Entstehung von Krankheiten spielen, als häufige [116]. Erklären lässt sich dieser Umstand damit, dass schädliche Varianten stärker der natürlichen Selektion unterliegen, und deshalb in der Gruppe der hoch-frequenten Allele unterrepräsentiert sind. Trotzdem sollte die oft genannte Grenze von einer *minor allele frequency* von unter 1% nicht zu starr betrachtet werden. So werden einige wenige monogene Erkrankungen von Mutationen verursacht, die in mehr als 1% der generellen Bevölkerung zu finden sind. In diesen Fällen nimmt man an, dass das gehäufte Auftreten das Ergebnis eines Selektionsvorteils heterozygoter Träger der Mutation sein könnte. Der Defekt im β-Globin Gen bei Patienten mit Sichelzellanämie verringert zum Beispiel im heterozygoten Zustand das Risiko, an Malaria zu erkranken [117].

Für die Trainingsdaten ist es wichtig, möglichst nur sicher klassifizierte Fälle einzuschließen. Andererseits sollte auch eine genügend große Anzahl an Fällen enthalten sein. Für die angestrebte Größenordnung (mindestens 1000 Fälle pro Einzeldatensatz) ist es nicht möglich, eventuelle Krankheitsassoziationen manuell zu überprüfen. Daher habe ich als pauschales Einschlusskriterium die Präsenz der Variante in allen drei Genotypen (AA, AB, BB) in mindestens 20 (InDels) bzw. in mindestens 50 (SNPs) der im 1000-Genom-Projekt analysierten Proben definiert.

Die Qualität der Daten aus dem 1000-Genom-Projekt ist gut: Der erwartete Anteil falsch positiver Funde gemessen an allen Funden (*false discovery rate, FDR*) wurde von deren Autoren mit verschiedenen Techniken bestimmt: Für einfache Varianten lag diese zwischen 1,6 und 1,8%, für größere Deletionen bei 2,1% [32]. Die *accuracy* individueller Genotyp-Bestimmungen liegt bei 99% für häufige SNPs und bei 95% für seltene SNPs mit einer Frequenz von 0.5%. Das heißt, in je mehr Proben eine Variante detektiert wurde, desto sicherer sind die Daten. Die Allelfrequenz alleine sagt allerdings noch nichts über das Auftreten der verschiedenen Genotypen aus. Da gerade für seltene, homozygot-rezessiv vererbte, monogene

Erkrankungen der homozygot-variante Genotyp (BB) von Belang ist, wurden für die MutationTaster Trainingssets die Anzahl der einzelnen Genotypen (nicht die Allelfrequenzen) als Einschlusskriterium verwendet. Der gewählte Mindestwert von 20 bzw. 50 reflektiert einen Kompromiss zwischen größtmöglicher Sicherheit, dass die Variante harmlos ist, und dem Wunsch, möglichst viele Varianten einzuschließen.

Krankheitsmutationen

Für den Trainingsdatensatz mit Krankheitsmutationen wurden als solche gekennzeichnete (Kategorie *DM*, siehe Kapitel 2.2.1) Einträge aus der professionellen Version von HGMD [27] verwendet. Obwohl Varianten vor ihrer Aufnahme in die Datenbank auf ihr Krankheitspotential überprüft werden, kann man nicht zu 100% ausschließen, dass sich harmlose Polymorphismen darin befinden, die irrtümlicherweise für krankheitsverursachend gehalten wurden (persönliche Kommunikation mit Frank Schacherer, CEO BIOBASE GmbH). Ende das Jahres 2012 wurden aus dem Datenbestand der HGMD mehr als 500 Einträge aus der Kategorie *DM* überprüft, die mit einer Frequenz von mehr als 1% bzw. mehr als 5% ebenfalls im 1000-Genom-Projekt gelistet werden. 33 Einträge wurden daraufhin aus der HGMD entfernt, da ihr Krankheitspotential durch die Originalpublikationen nicht eindeutig belegt war. Viele weitere Varianten wurden re-klassifiziert oder mit zusätzlichen Kommentaren oder Referenzen versehen (persönliche Kommunikation mit Peter Stenson, Erstautor der aktuellen HGMD Publikation). Verschiedene Trainings- und Validierungsrunden des MutationTaster Klassifikators mit Daten, die einmal aus der Zeit vor dem Abgleich mit dem 1000-Genom-Projekt stammten, und einmal aus der Zeit danach, zeigten eine Verbesserung in der Vorhersagequalität. Die Bereinigung des Datensets hat vor allem die Vorhersage komplexer Varianten verbessert, was die Vermutung nahe legt, dass sich im unbereinigten Datenset wahrscheinlich einige falsch klassifizierte Varianten befunden haben. Diese Erfahrung unterstreicht, wie sehr die Performanz des Klassifikators von der Qualität der Trainingsdaten beeinflusst wird.
Höchstwahrscheinlich wird es immer wieder vorkommen wird, dass eine Variante gleichzeitig als Krankheitsmutation (z.B. in NCBI ClinVar) gelistet ist und mehr als vier Mal in homozygoter Form im 1000-Genom-Projekt gefunden wurde. In solchen Fällen kommt MutationTaster in einen Konflikt, weil theoretisch zwei sich widersprechende automatische Vorhersagen generiert werden müssten: Die Präsenz einer Variante als Krankheitsmutation in NCBI ClinVar sorgt für die automatische Vorhersage *disease causing*. Die Präsenz einer Variante im homozygoten Status in mehr als vier Fällen im 1000-Genom-Projekt erzeugt die automatische Vorhersage *polymorphism*. MutationTaster wird bei einem derartigen Konflikt immer den Daten aus dem 1000-Genom-Projekt das größere Vertrauen schenken, und die Variante als *polymorphism* klassifizieren. Der Benutzer wird aber auf die widersprüchliche Datenlage hingewiesen.

8.1.2 Die Zusammenstellung der Trainingsdatensätze

Generell ist es erstrebenswert, dass die Datensätze zum Trainieren eines Bayes Klassifikators ausgewogen zusammengestellt sind. Beispielsweise sollte die Anzahl synonymer Varianten im Datensatz für Polymorphismen und im Datensatz für Krankheitsmutationen etwa gleich sein. In der Realität lässt sich dies leider nicht immer erreichen (siehe Tabelle 6 in Kapitel 5.1.2). Synonyme Varianten zum Beispiel werden viel seltener auf eventuelle Krankheitsassoziationen hin untersucht, als kodierende Varianten, weil bei letzteren die Kausalitäten einfacher nachzuweisen sind. So sind zwar 11% der Krankheitsmutationen in HGMD Varianten, welche das Spleißmuster zerstören; Varianten in anderen Regionen des Introns werden aber oft vernachlässigt, weil ihre Beteiligung an einer Krankheit schwerer experimentell zu belegen ist [118]. 55% der Krankheitsmutationen in HGMD sind *missense*- oder *nonsense* Mutationen. Obwohl die Anzahl gefundener Krankheitsmutationen dank neuer Sequenzierungstechnologien generell steigt, liegt bei Ansätzen wie dem *Whole Exome Sequencing* der Fokus von vorneherein auf den Exons und somit vor allem auf kodierenden Varianten. Es werden zwar auch relativ viele nicht-kodierende oder synonyme Varianten detektiert - diese werden aber wegen mangelnder Analysemöglichkeiten (sowohl *in silico* als auch experimentell) meist vernachlässigt. Im Gegensatz zu nicht-synonymen Varianten ist die Mehrzahl der vielen nichtkodierende Varianten wahrscheinlich tatsächlich harmlos: Die Chance, dass tief im Intron ein Nukleotidaustausch das Spleißen oder die Proteinexpression negativ beeinflusst, ist geringer als im Exon, wo meistens jedes einzelne Nukleotid eine unmittelbare Funktion hat, nämlich die Vervollständigung eines Codons zur Kodierung einer bestimmten Aminosäure. Die synonymen Austausche reichern sich aufgrund eines geringeren Selektionsdruckes an, weshalb es in dieser Klasse insgesamt sehr viel mehr Varianten gibt, von denen die harmlosen über- und die krankheitsverursachenden Varianten unterrepräsentiert sind.
Das Gegenteil ist bei den komplexen Austauschen der Fall: Hier enthält der Trainingsdatensatz sehr viel mehr komplexe krankheitsverursachende Mutationen als komplexe harmlose Polymorphismen. Das liegt zum Teil daran, dass komplexe Aminosäureveränderungen einfacher zu untersuchen sind, da sich die Konsequenzen unmittelbar erschließen. Viele der Wissenschaftler auf der Suche nach der genetischen Ursache einer Krankheit unterziehen komplexe Aminosäureaustausche viel eher einer genaueren Analyse als synonyme Varianten. Nicht-synonyme bzw. komplexe Varianten werden also bevorzugt untersucht, weshalb auch viele davon bekannt und charakterisiert sind. Studien oder Untersuchungen, in denen die ursächliche Mutation nicht identifiziert werden konnte, werden außerdem in der Regel nicht veröffentlicht (*publication bias*), was die Datenlage weiter verzerrt. Es ist allerdings nicht auszuschließen, dass einige der komplexen, als Krankheitsursache publizierten Mutationen, fälschlicherweise als solche klassifiziert wurden.
Harmlose komplexe Varianten sind seltener und lassen sich oft dadurch erklären, dass das Genprodukt für den Organismus nicht essentiell ist. Ein Beispiel ist die häufigste Ursache

für Blutgruppe 0: durch die Deletion eines Nukleotids wird das Leseraster verschoben und es kommt zu einem verfrühten Stopcodon [119]. Für das Protein ist das fatal, es wird nämlich nicht exprimiert, den Organismus jedoch macht es nicht krank. Solche oder ähnliche Fälle sind für einen Bayes Klassifikator extrem schwer einzuordnen, da Informationen über die Bedeutung des Genprodukts für den Organismus nicht vorliegen und daher auch nicht in seine Vorhersage einfließen können (siehe Kapitel 8.7 für unsere Pläne, wie dies in Zukunft geändert werden könnte). Durch das Training mit harmlosen, komplexen Varianten können aber andere Merkmale zur Unterscheidung herangezogen werden, weshalb es wichtig ist, dass der Trainingsdatensatz möglichst viele Varianten dieser Art enthält.

8.2 Tests

8.2.1 Implementierte Tests

8.2.1.1 Regulatorische Elemente

MutationTaster hat Zugriff auf mehr als 600 regulatorische Elemente aus *Ensembl Regulation*, die in 12 verschiedene Klassen eingeteilt sind. Wenn eine Variante in einem solchen Element lokalisiert ist, wird dem Benutzer dies zusammen mit Details zum regulatorischen Element (Klassenzugehörigkeit, Name, Beschreibung) angezeigt. Dem Bayes Klassifikator wird aber nur die Klassenzugehörigkeit übermittelt. Die Übergabe des tatsächlichen Elements an den Klassifikator hätte das Modell sehr kompliziert gemacht, da dies mehr als 600 zusätzliche Parameter bedeutet hätte. In einem zukünftigen Modell könnte diese sehr grobe Einteilung in Klassen aufgehoben und die einzelnen Elemente genauer darauf untersucht werden, welche am Besten zur Diskrimination zwischen harmlosen und krankheitsverursachenden Varianten geeignet sind. Problematisch bei der Betrachtung der regulatorischen Elemente ist die Tatsache, dass diese oft sehr weit von dem Gen, auf das sie wirken, entfernt liegen. Das heißt, wenn eine Variante in einem Gen in einem regulatorischen Element lokalisiert ist, heißt dies nicht unbedingt, dass dieses regulatorische Element auch tatsächlich das Gen beeinflusst, in dem es liegt. Hier müsste also eine genauere Unterscheidung der Elemente in Hinsicht auf ihre Gen-Assoziation vorgenommen werden.
In der Elementenklasse *Regulatory Feature* sind unter anderem Promotor-assoziierte sowie Gen-assoziierte Elemente enthalten. Erstere sind in der Region um die Transkriptionsstartstelle überrepräsentiert, letztere im Rest des Gens. Bei diesen Elementen können wir zunächst davon ausgehen, dass sie auf das unmittelbar angrenzende oder sie umschließende Gen wirken. Im aktuell verwendeten Klassifikationsmodell ist P(regulatory feature | DM), also die Wahrscheinlichkeit, dass eine Krankheitsmutation in einem *Regulatory Feature* liegt gleich 0,18. P(regulatory feature | SNP), also die Wahrscheinlichkeit, dass ein harmloser Polymorphismus ein *Regulatory Feature* beeinflusst, ist gleich 0,08. Das bedeutet, dass sich die Elementenklasse hier als Diskriminationsmerkmal eignet. Jedoch enthält die Elementenklasse

nicht nur Promotor- und Gen-assoziierte Elemente, sondern auch noch andere, nicht direkt mit Genen assoziierte Elemente. Hier wäre es lohnenswert, die einzelnen Elemente in dieser Klasse auf ihr diskriminatives Potential zu untersuchen, da man möglicherweise durch Übergabe des detaillierten Elemententyps an den Klassifikator noch eine bessere Unterscheidung erreichen kann.

8.2.1.2 Spleißing

MutationTaster nutzt zur Spleißanalyse das Programm NNSplice. Es stammt aus dem Jahr 1997 und ist damit relativ alt. Es wurde jedoch in unregelmäßigen Abständen aktualisiert, zuletzt im Jahr 2008 (siehe Kapitel 2.2.2). NNSplice kann weder ESEs (*exonic splice enhancers*), ESSs (*exonic splice silencers*) noch die *branch point* Sequenz detektieren. In einem 2008 publizierten Vergleich von sechs Programmen zur Spleißstellenvorhersage schnitt NNSplice als das zweitbeste ab [120]. Das in diesem Vergleich beste Programm, ASSA (*A*utomated *S*plice *S*ite *Analysis*) [121], war leider zum Zeitpunkt der Entstehung der hier vorgestellten Version von MutationTaster (zwischen 2009 und 2012) nicht mehr verfügbar.
NNSplice erkannte in einem von uns durchgeführten Test mit 3599 bekannten Spleißmutationen in 78% (2785) der Fälle eine Veränderung der Spleißstelle. In dem 2008 publizierten Vergleich wurde die Sensitivität von NNSplice auf 74% beziffert. Problematischer als falsch negativ vorhergesagte Spleißstellen ist in MutationTaster die relativ hohe Anzahl von Typ I Fehlern, also falsch positiven Ergebnissen. Falsch positiv heißt in diesem Fall, dass entweder a) eine Veränderung einer Spleißstelle angezeigt wird, die überhaupt nicht existiert, oder b) dass die Entstehung einer neuen, kryptischen Spleißstelle angezeigt wird. Während man letzteres Problem bioinformatisch nicht lösen kann, weil es keine Möglichkeit gibt, die vorhergesagte Entstehung einer neuen Spleißstelle *in silico* nachzukontrollieren, kann man die ersteren Fälle, also die vorhergesagte Veränderung von „unechten“ Spleißstellen eingrenzen. Sobald MutationTaster von NNSplice die Information bekommt, dass eine Spleißstelle durch eine Mutation verloren geht oder abgeschwächt wird, überprüft die Software anhand der in der Datenbank vorliegenden Exoninformationen, ob diese Spleißstelle eine tatsächlich genutzte ist. Falls dies nicht der Fall ist, wird der Verlust oder die Abschwächung der Spleißstelle ignoriert. Da die Autoren von NNSplice keine Empfehlungen zur Interpretation der NNSplice Konfidenzwerte angeben, wurden die für den Einsatz in MutationTaster optimalen, von uns genutzten Schwellenwerte in einem eigenen Test ermittelt. Von den als neu entstanden vorhergesagten Spleißstellen werden nur solche angezeigt, deren von NNSplice ausgegebener Konfidenzwert 0.3 übersteigt, und nur solche Verstärkungen einer bestehenden Spleißstelle angezeigt, deren Differenz mindestens 10% beträgt.
Erst kürzlich präsentierten die Autoren von ASSA einen Nachfolger ihrer Software, die mittlerweile ASSEDA heißt [122]. Es ist denkbar, ASSEDA und NNSplice einem direkten Vergleich zu unterziehen und gegebenenfalls die (zusätzliche) Integration von ASSEDA in Er-

wägung zu ziehen, sofern die Software gute Ergebnisse erzielt und eine lokal installierbare Version frei verfügbar ist. Die Abfrage einer web-basierten Version kommt aus Gründen der Geschwindigkeits- und Stabilitätsoptimierung für die Benutzung in MutationTaster nicht in Frage. Generell gehen unsere Überlegungen zur Optimierung der Spleißstellenanalyse in MutationTaster jedoch in die Richtung, einen eigenen, den Bedürfnissen von MutationTaster angepassten, Algorithmus zu entwickeln. Hierbei würden wir neben der Qualität der Ergebnisse auch einen besonderen Schwerpunkt auf der Geschwindigkeit des Programmes legen. Gerade in Anbetracht der Tatsache, dass mit den durch NGS-Technologien produzierten immensen Datenmengen die Analyse von intronischen Varianten immer mehr an Bedeutung gewinnen wird, scheint die Weiterentwicklung der Spleißanalyse in einem gesonderten Projekt sinnvoll.

8.2.1.3 Alignment für Konservierungsanalyse

Ein Alignment dient dem Vergleich zweier Sequenzen (z.B. DNA- oder Aminosäuresequenzen), die auf ihre Ähnlichkeit untersucht werden sollen. Bei einem globalen Alignment werden die beiden Sequenzen in Gänze, also von Anfang bis Ende, miteinander verglichen. Die verschiedenen Möglichkeiten, die beiden Sequenzen miteinander zu alignieren, werden mit einem Punktesystem bewertet. Unähnlichkeiten oder Lücken werden durch Punktabzug „bestraft“. Ein globales Alignment ist sinnvoll, wenn zwei Sequenzen sich in ihrer Zusammensetzung und Länge vermutlich ähneln (z.B. homologe Gensequenzen) [123]. Der erste Algorithmus für ein globales Alignment zweier Sequenzen wurde 1970 von Needleman und Wunsch vorgestellt [124]. Zum Vergleich zweier Sequenzen, die größere Unterschiede aufweisen, ist das lokale Alignment besser geeignet. Basierend auf dem Algorithmus von Needleman und Wunsch entwickelten Smith und Waterman [125] eine Variante, die zwei Sequenzen auf einzelne, sich maximal ähnelnde Teilabschnitte, also *lokale Alignments*, überprüft, und unähnliche Sequenzabschnitte ignoriert. Für Lücken, die am Anfang oder Ende eines lokalen Alignments auftreten, wird keine Lückenstrafe vergeben, so dass mehrere einzelne Alignments eine bessere Bewertung bekommen, als ein globales Alignment mit Lücken und Unähnlichkeiten.
Obwohl im Rahmen der Konservierungsanalyse zwei homologe, sich vermutlich ähnelnde, Sequenzen miteinander verglichen werden (jeweils die menschliche Sequenz mit einem Homolog aus einer anderen Spezies), wird mit BLAST [29] ein Algorithmus für lokale Alignments eingesetzt. Der Grund dafür ist, dass MutationTaster für die Bewertung des Konservierungsgrades der durch die zu analysierende Variante veränderten Position(en) keine Kenntnis über das globale Alignment des gesamten Gens oder Proteins benötigt. Es reicht, wenn der Bereich rund um die zu analysierende Variante aligniert wird. Ein Alignment dieses Bereiches ließe sich selbstverständlich auch aus den Ergebnissen eines globalen Alignments extrahieren, dieses wäre umfassender, jedoch auch viel zeitaufwendiger. BLAST ist zudem nicht nur schneller als übliche Algorithmen für globale Alignments, sondern auch anderen

Programmen für lokale Alignments in der Geschwindigkeit überlegen. Erreicht wird dies dadurch, dass BLAST sich während der Suche nach dem optimalen lokalen Alignments auf sogenannte *high-scoring segment pairs* (Teilabschnitte mit besonders hohem Score) beschränkt, und andere lokale Alignments mit weniger guter Bewertung ignoriert [29].

8.2.1.4 Aminosäureaustausch

Zur Bewertung der Schwere eines Aminosäureaustausches übergibt MutationTaster dem Bayes Klassifikator den konkreten Austausch, also die Information, dass beispielsweise ein Leucin gegen ein Isoleucin ausgetauscht wurde. Zu Informationszwecken wird dem Benutzer auf der Ergebnisseite zusätzlich zum Aminosäureaustausch auch dessen Bewertung basierend auf einer von Grantham entworfenen Substitutionsmatrix angezeigt [72]. Grundsätzlich wäre es denkbar, dem Klassifikator nicht nur den eigentlichen Austausch, sondern ebenfalls die Bewertung durch die Grantham-Matrix als Parameter zu übergeben. Dies wurde im Rahmen der Optimierung von MutationTaster auch untersucht, jedoch war die Erkennungsrate des Klassifikators besser, wenn er statt des Grantham-Wertes den eigentlichen Austausch mitgeteilt bekam. So kann ausgehend von den Trainingsdaten eine eigene Substitutionsmatrix (wenn auch nicht in der klassischen Form einer Matrix) erstellt werden, deren Werte die Häufigkeit der möglichen Aminosäureaustausche in harmlosen und Krankheitsverursachenden Varianten wiederspiegeln. Dies ist für den Zweck von MutationTaster deutlich besser geeignet, als beispielsweise die Werte einer Matrix zur Ähnlichkeit von Aminosäuren (Grantham-Matrix) oder zur beobachteten evolutionären Austauschrate in Proteinen (BLOSUM).

8.2.1.5 Stopcodon

DNA-Veränderungen können in einem verfrühten Stopcodon resultieren. Unter bestimmten Umständen wird eine mRNA mit einem verfrühten Stopcodon degradiert, wodurch negativen Folgen für den Organismus vorgebeugt werden kann (siehe auch Kapitel 3.3.2). Dies konnte zum Beispiel für eine *nonsense* Mutation im β-Globin Gen gezeigt werden, die die Produktion eines trunkierten Proteins zur Folge hat. Die normalerweise rezessiv vererbte Krankheit *Thalassaemia intermedia* tritt in diesem Fall aufgrund schädlicher dominant-negativer Effekte des trunkierten Proteins bereits auf, wenn die Mutation heterozygot vorliegt [126]. Ein Schutzmechanismus des Organismus vor irregulär verkürzter mRNA ist die Degradierung selbiger durch sogenannten *nonsense-mediated mRNA decay* (NMD) [127]. Die Regeln, nach denen ein reguläres von einem verfrühten Stopcodon unterschieden wird, sind noch nicht vollständig aufgeklärt. Am besten belegt ist die sogenannte *50-54 nt boundary rule*, die besagt, dass ein Stopcodon, das mehr als 50-54 Nukleotide *upstream* von der letzten Exon-Exon-Grenze auftritt, zu NMD führt [99][128]. Es gibt jedoch mRNAs mit PTCs, die dieses Kriterium erfüllen und nicht durch NMD degradiert werden, genauso wie es mRNAs gibt, die auch ohne vorliegende Bedingungen zur Erfüllung dieser Regel wegen eines PTCs abgebaut

werden [129]. Als alternative Determinanten des NMD-Mechanismus werden unter anderem die Länge der 3'UTR [130] und die Nähe des PTC zum Start-ATG [131][132] diskutiert. Die Umstände, unter denen diese Kriterien greifen sind allerdings bislang längst nicht so gut verstanden, die wie die *50-54 nt boundary rule*, und konnten daher nicht im MutationTaster Computerprogramm operationalisiert werden. In MutationTaster werden deshalb generell *nonsense* Varianten, welche letztere Regel erfüllen als Ziel von NMD deklariert, und automatisch als *disease causing* klassifiziert. Da man aber bei alle anderen *nonsense* Varianten ebenfalls nicht ausgeschließen kann, dass durch alternative Mechanismen die verkürzte mRNA durch NMD degradiert wird, werden diese als *mögliches* NMD-Ziel bewertet.

8.2.2 Mögliche zusätzliche Tests

Obwohl MutationTaster auf eine breite Palette von Tests zurückgreifen kann, gibt es einige Tests, die bislang nicht integriert sind, aber durchaus hilfreich wären. Vor allem für Varianten in der 3'- und 5'UTR stehen nur wenige Tests zur Verfügung. Dabei beherbergen die untranslatierten Bereiche der mRNA wichtige Sequenzelemente die unter anderem Stabilität und Lokalisation der mRNA sowie die Translationseffizienz regulieren [133][134]. Wird die Sequenz eines solchen regulatorischen Elements verändert, kann das krankheitsverursachend sein (z.B. [135][136][137]). Auch die Entstehung neuer regulatorischer Elemente, z.B. eines zusätzlichen Leserasters *upstream* des originalen Start-ATGs (*uATG*), kann an der Entstehung von Krankheiten beteiligt sein (z.B. [138]). Folgende zusätzliche Tests für Varianten in den UTRs wären denkbar und mit absehbarem Aufwand umzusetzen: 1) Suche nach bekannten miRNA Bindungsstellen und Überprüfung, ob diese durch eine Variante verändert werden; 2) Überprüfung auf die Entstehung von einem *uATG*.
Ein Aspekt, der von MutationTaster gänzlich vernachlässigt wird, ist die Frage, ob eine Veränderung der Gensequenz die Sekundär- und Tertiärstruktur des entsprechenden Proteins verändert. Es gibt Vorhersageprogramme, die Strukturdaten integrieren, so z.B. SNPs3D [139] oder auch PolyPhen-2 [46]. Während für fast alle Proteine Sequenzdaten verfügbar sind, ist der Anteil der Proteine mit verfügbaren Strukturdaten deutlich geringer. Die Autoren von SNPs3D konnten von insgesamt gut 10.200 Testfällen für 92% eine Vorhersage basierend auf Sequenzinformationen generieren, jedoch nur für 37% eine Vorhersage basierend auf Strukturinformationen treffen. Den Autoren von PolyPhen standen von 11.152 nicht-synonymen Einzelbasenaustauschen lediglich für 1.092 die entsprechenden Proteinstrukturmodelle zur Verfügung [36]. Sie haben außerdem festgestellt, dass in dem von ihnen untersuchten Datenset die durch multiple Sequenz-Alignments gewonnenen Informationen das entscheidende Kriterium für die Vorhersage waren und sie auch für Veränderungen in Proteinen ohne bekannte 3D-Struktur verlässliche Vorhersagen treffen konnten [36].
Wegen der unbefriedigenden Datenlage habe ich auf die Implementierung eines Proteinstrukturtests in MutationTaster verzichtet. Für das Training des Bayes Klassifikators sind

sehr viele Trainingsfälle nötig. Wenn das Vorhandensein eines 3D-Proteinmodells Voraussetzung zum Einschluss in die Trainingsdatensätze gewesen wäre, hätte sich die Anzahl geeigneter Trainingsfälle drastisch reduziert. Abgesehen davon, dass die Auswirkung einer DNA-Sequenzvariante auf die 3D-Struktur des entsprechenden Proteins ohnehin nur sehr schwer vorhersehbar ist, würde eine derartige Analyse die Laufzeit eines einzelnen MutationTaster Laufes vermutlich erheblich verlängern.

8.3 Der Einfluss der einzelnen Tests auf die Klassifikation

Generell ist es nicht möglich, genau festzustellen, warum der Bayes Klassifikator in einem bestimmten Fall eine bestimmte Entscheidung getroffen hat. Es lässt sich nicht identifizieren, welches einzelne Testergebnis welchen Einfluss auf die Vorhersage hatte. Das dem Klassifikator zugrunde liegende Modell gibt jedoch Aufschluss über die bedingten Wahrscheinlichkeiten für die einzelnen Parameter. Dort steht beispielsweise, dass der Klassifikator anhand der Trainingsdaten die Wahrscheinlichkeit für den Verlust des Protein *features* „Helix“ im Zusammenhang mit einer Krankheitsmutation bei 0,45 sieht, während sie im Zusammenhang mit einem Polymorphismus nur bei 0,15 liegt (Modell *complex_aae*). Man kann also die bedingten Wahrscheinlichkeiten im Modell betrachten um einen Eindruck davon zu bekommen, wie stark sich die einzelnen Parameter in ihren Zuständen zwischen Krankheitsmutation und Polymorphismus unterscheiden.

8.4 Die Verwendung eines Bayes Klassifikators in MutationTaster

Die Entscheidung, in MutationTaster einen Bayes Klassifikator zu implementieren, war im Wesentlichen davon beeinflusst, dass naive Bayes Klassifikatoren einerseits bekanntermaßen robuste Ergebnisse erzielen [112] und andererseits recht einfach zu implementieren sind. Bayes Klassifikatoren benötigen außer dem Modell kaum Konfigurationsdateien. Für viele Programmiersprachen gibt es Bayes Klassifikatoren als vorgefertigtes Modul, so auch in Perl, der Programmiersprache in der MutationTaster geschrieben wurde. Durch die unabhängige Betrachtung aller Variablen wird das Risiko des *overfitting**, der Überanpassung des Modells an einen bestimmten Datensatz, reduziert. Natürlich haben wir initial auch andere Klassifikationsmöglichkeiten in Betracht gezogen (z.B. ein neuronales Netz), als erstes aber der Bayes Klassifikator getestet. Die Ergebnisse waren so überzeugend, dass wir auf die Erprobung weiterer Klassifizierungsmethoden verzichtet haben. Stattdessen haben wir direkt mit der Ausarbeitung des Bayes Klassifikators begonnen. Das für die Implementierung von MutationTaster verwendete Perl Modul hat den Nachteil, dass es ordinalskalierte Werte (wie sie z.B. durch phyloP und phastCons anfallen) nicht als solche interpretieren kann. Sie müssen entweder als normalverteilt betrachtet werden (was sie in der Regel nicht sind) oder in Klassen unterteilt werden (was den Informationsgehalt mindern kann). In einer eventuellen

zukünftigen Version von MutationTaster sollte daher eine alternative Bayes Klassifikator Implementierung, möglicherweise in einer anderen Programmiersprache, integriert werden, so dass ordinalskalierte Werte direkt verarbeiten werden können.
Außerdem könnte die Optimierung des Bayes Klassifikators noch ausführlicher gestaltet werden, als dies bislang geschah. Es wäre z.B. möglich, mit einem Minimum an Parametern zu starten und nach und nach zusätzliche hinzufügen. Ebenso kann zu Anfang der komplette Satz an möglichen Parametern genutzt und anschließend können Schritt für Schritt welche entfernt werden. Auf diese Weise können unnötige oder störende Parameter identifiziert werden. Dieses Verfahren kann sehr zeitaufwendig werden, wenn ein einzelner Trainingslauf lange dauert. Deshalb haben wir bislang die zeitsparendere Möglichkeit der Optimierung, nämlich die direkte Inspektion des Modells, vorgenommen. Parameter, die in verschiedenen Klassen ein ähnliches Gewicht haben, können entfernt, und die Performanz anschließend erneut bestimmt werden. Diese manuelle Methode ist jedoch nicht so gründlich wie ein systematisches Iterieren durch alle möglichen Parameterkombinationen.

8.5 Nutzen von MutationTaster zur Analyse von NGS-Daten

MutationTaster kann genutzt werden, um in NGS-Projekten gefundene Sequenzvarianten gemäß ihres Krankheitspotentials zu priorisieren und die Zahl der interessanten Varianten drastisch zu reduzieren. Bei einem *Whole Exome Sequencing* (WES) Projekt werden typischerweise mehrere zehn- bis hunderttausend Varianten detektiert und z.B. im VCF-Format gespeichert. Die MutationTaster QueryEngine kann Varianten im standardmäßigen VCF-Format prozessieren. In vielen Fällen kann ein Großteil der detektierten Varianten von vorneherein als Krankheitsursache ausgeschlossen werden. Wenn beispielsweise die genetische Ursache für eine seltene, in einer konsanguinen Familie homozygot-rezessiv vererbte Krankheit gesucht wird, können alle heterozygoten sowie alle mehrfach im 1000-Genom-Projekt im homozygoten Status detektierten Varianten aussortiert werden. Die QueryEngine bietet derartige Filtermöglichkeiten an. Meist kann man so schon vor der eigentlichen Analyse durch MutationTaster die Zahl der interessanten Varianten um bis zu 50% reduzieren. Das verkürzt den Zeitaufwand sowohl für die eigentliche Analyse als auch für die spätere Auswertung der Ergebnisse. Die Ergebnisse können nach verschiedenen Aspekten sortiert werden. Meistens wird man sich die Varianten sortiert nach der Schwere des vorhergesagten Krankheitspotentials anzeigen lassen. Als weitere Sortieroption kann die Art des Austausches gewählt werden: Varianten, die einen komplexen oder einfachen Aminosäureaustausch nach sich ziehen, stehen meist mehr im Fokus des Interesses als synonyme oder intronische Veränderungen. Üblicherweise ist die große Mehrheit der Veränderungen (70-85%) intronisch oder synonym, während sich lediglich um die 0,2% der Varianten als komplexe Veränderungen und weitere 2% als einfache Aminosäureaustausche darstellen. Das sind zum Teil zwar immer noch hunderte

bis tausende von Einzelergebnissen, die manuell inspiziert werden müssen. Gegenüber den zehn- oder hunderttausenden, die initial detektiert werden, ist dies jedoch eine erhebliche Arbeitsentlastung. Selbstverständlich besteht die Gefahr, dass durch rigide Filtermethoden die eigentlich krankheitsverursachende Variante übersehen wird. Ein mögliches Vorgehen wäre, initial strenge Filtereinstellungen nach und nach zu lockern, falls sich im Pool der jeweils in Frage kommenden Varianten kein guter Kandidat findet. Bei einer homozygot-rezessiv vererbten Krankheit in einer konsanguinen Familie könnten beispielsweise zunächst alle heterozygoten und als *polymorphism* vorhergesagten Veränderungen ausgeschlossen werden. Wenn unter den verbliebenen homozygoten Veränderungen kein guter Kandidat ist, kann die Suche auf compound-heterozygote, als *disease causing* klassifizierte Varianten ausgeweitet werden. In einem typischen WES-Projekt sind die meisten der als *polymorphism* klassifizierten Varianten synonym oder intronisch. Es kann passieren, dass zum Schluss all diese potentiell als Krankheitsursache in Frage kommen, weil alle anderen Varianten ausgeschlossen worden sind. In dieser Situation bringt auch der Einsatz von MutationTaster wahrscheinlich keinen weiteren Vorteil und man muss damit rechnen, dass die Krankheitsmutation unerkannt bleibt.

8.6 Vergleich mit anderen Vorhersageprogrammen

Im Vergleich mit anderen Vorhersageprogrammen schneidet MutationTaster bei der Bewertung einfacher Aminosäureaustausche signifikant besser ab. MutationTaster integriert mehr Daten und greift auf mehr Tests zurück als SIFT, PROVEAN und PolyPhen-2, was ein Grund für die bessere Vorhersagequalität sein kann. Anders als andere Programme, führt MutationTaster für einen Aminosäureaustausch ebenfalls Tests auf DNA-Ebene durch. Der Konservierungsstatus beispielsweise wird nicht nur für die betroffene(n) Aminosäure(n) bestimmt, sondern zusätzlich für das veränderte Nukleotid oder die veränderten Nukleotide. Zwar greift PolyPhen-2 im Gegensatz zu MutationTaster wiederum auf zusätzliche Tests zur Proteinstruktur zurück, jedoch stehen die nötigen Strukturinformationen nur für einen geringen Teil aller bekannten Protein zur Verfügung und die Aussagekraft dieser Tests ist eher gering [36].
Die rasche Verfügbarkeit der für eine Vorhersage nötigen Daten ist neben der Vorhersagequalität ein weiterer Aspekt der über die Nützlichkeit eines Vorhersageprogrammes entscheidet. Generell sind speziellere Daten, wie z.B. Proteinstrukturinformationen, oder für bestimmte Proteine auch Daten zur Sequenzhomologie, tendenziell in geringerem Maße vorhanden als allgemeine Informationen zu Gen und Protein. Die Vorhersage von PolyPhen-2 basiert in großem Maße auf dem Vergleich von homologen Proteinsequenzen. Da ein derartiges Alignment aber für einige Proteine (v.a. solche, die häufige Wiederholungen eines bestimmten Sequenzmusters enthalten) nicht erstellt werden kann, können Varianten in diesen Proteinen oft nicht klassifiziert werden.

In einem älteren Vergleich von MutationTaster mit den Programmen PolyPhen [36], PolyPhen-2 [46], Pmut [140], SNAP [141] und Panther [38] wurde nicht nur die Qualität der Vorhersage für analysierte Varianten gemessen, sondern auch untersucht, wie viele Varianten insgesamt prozessiert werden konnten. Außer MutationTaster konnte keines der anderen Programme für alle Varianten des Testsets eine Vorhersage generieren, zum Teil blieb eine erhebliche Anzahl der Varianten ohne Vorhersage (siehe Tabelle 10). Zwar stammten die Testfälle aus den ursprünglichen MutationTaster Trainingsdatensätzen, so dass es sich hier ausschließlich um Varianten in Genen bzw. Transkripten handelte, die MutationTaster auch vorhersagen konnte. Jedoch zeigte sich auch in anderen Studien, dass MutationTaster im Vergleich mit anderen Vorhersageprogrammen die wenigsten Varianten unklassifiziert ließ (z.B. [142]).

Programm	n (analysierte Fälle)					*accuracy* (%)		
	TP	**TN**	**FP**	**FN**	**ND**	**analysierte Fälle**	**gemeinsame Fälle**	**alle Fälle**
PPH	728	789	206	272	5	76,0	75,8	75,8
PPH2-var	773	666	211	134	216	80,7	81,9	72,0
PPH2-div	776	655	222	131	216	80,2	81,3	71,5
SNAP	789	403	362	185	261	68,5	68,3	59,6
Panther	510	196	503	181	610	50,8	53,4	35,3
Pmut	581	720	270	418	11	65,4	62,0	65,0
MT	859	855	145	141	0	85,7	86,1	85,7

Tabelle 10: Vergleich von MutationTaster mit anderen Vorhersageprogrammen aus dem Jahr 2010. PPH = PolyPhen; PPH2-var = PolyPhen-2 HumVar Modell; PPH2-div = PolyPhen-2 HumDiv Modell; MT = MutationTaster; TP = richtig positiv; TN = richtig negativ; FP = falsch positiv; FN = falsch negativ; ND = nicht determiniert (nicht vorhergesagt); *accuracy* analysierte Fälle = (TP + TN) / (TP + TN + FP + FN); *accuracy* gemeinsame Fälle = (TP + TN) / (TP + TN + FP + FN), unter Berücksichtigung nur derjenigen Fälle, die in allen Programmen vorhergesagt werden konnten; *accuracy* alle Fälle = (TP + TN) / (TP + TN + FP + FN + ND). An alle Programme wurden die identischen 1.000 harmlosen und 1.000 krankheitsverursachenden Aminosäureaustausche versendet. MutationTaster Vorhersagen wurden mit dem *simple_aae* Modell generiert.

8.7 Ausblick

MutationTaster ist in der hier vorgestellten Form dafür geeignet, das Krankheitspotential von Varianten, die in bekannten Genen lokalisiert sind, vorherzusagen. Es ist wünschenswert, die Funktion von MutationTaster auf die Analyse extragenischer Varianten auszuweiten. Obwohl *Whole Genome Sequencing* (WGS) bereits vereinzelt eingesetzt wird, um die Ursache genetisch bedingter Erkrankungen zu klären, ist das *Whole Exome Sequencing* nach wie vor das Standardvorgehen. Bei WGS werden im Vergleich zu WES viel mehr Varianten

detektiert. Das ist einerseits gewünscht, denn auch DNA-Sequenzveränderungen außerhalb von Genen können krankheitsverursachend sein. Andererseits ist die Analyse der in WGS Projekten gefundenen, extragenischen Varianten bezüglich deren Krankheitspotential extrem schwierig. Es gibt aktuell noch keine Computerprogramme, die diese Varianten umfassend bezüglich ihrer Auswirkungen für den Organismus untersuchen können. Das liegt daran, dass in den letzten Jahrzehnten der Fokus zunächst auf den proteinkodierenden Abschnitten einzelner Gene, und mit dem Erfolg des WES auf dem gesamten Exom lag und liegt. Erfahrungen, wie man Daten außerhalb bekannter Gene behandeln sollte, um größtmöglichen Nutzen aus ihnen zu ziehen, liegen allenfalls vereinzelt vor. Mit dem im Oktober 2012 vorgestellten Ergebnissen des ENCODE Projekts [58] könnte sich dieser Umstand aber ändern. Zu den „Encode Elementen“ gehören unter anderem Histonmodifikationen (z.B. Methylierung oder Acetylierung), Informationen über den Chromatin-Status, Transkriptionsfaktorbindungsstellen und in RNA umgeschriebene Elemente [58]. Zwar werden diese Daten teilweise (über die Einbindung von *Ensembl Regulation* schon von MutationTaster genutzt, allerdings wurden im Vorfeld keine genauere Analyse der Daten und ihres diskriminativen Potentials zwischen harmlosen und krankheitsverursachenden Varianten vorgenommen. Wir vermuten, dass die ENCODE Daten wesentlich mehr Potential bergen als wir bislang ausschöpfen. Wie genau die Daten von MutationTaster besser genutzt und als Grundlage für die Vorhersage des Krankheitspotentials von extragenischen DNA-Sequenzvarianten genutzt werden müssen, bleibt zu untersuchen. Die Frage, ob eine Variante in einem ENCODE-annotierten Element lokalisiert ist, lässt sich bioinformatisch einfach lösen. Daten zu Transkriptionsfaktorbindungsstellen würden sich wahrscheinlich nutzen lassen, indem man untersucht, ob eine Variante ein in der Wildtypsequenz vorhandenes Bindungsmotiv abschwächt. Dafür müsste natürlich eine entsprechende Software integriert werden, die die Stärke der Motive analysiert. Die Entwicklung eines Vorhersagemodells mit Hilfe eines Bayes Klassifikators ist jedoch nur dann möglich, wenn es entsprechende Trainingsdaten gibt, d.h. Krankheitsmutationen in regulatorischen Elementen, von denen bekannt ist, dass sie die Genregulation stören, sowie ebenfalls in ENCODE-Elementen lokalisierte Polymorphismen, die jedoch für den Organismus harmlos sind. Falls es keine geeigneten Trainingsdaten gibt, ist es dennoch erstrebenswert, die ENCODE Daten einzubinden, teilweise jedoch vielleicht zunächst nur auf einer informativen Basis. Selbst das hätte für viele Forscher schon den Vorteil, dass sie sich nicht für jede gefundene Variante einzeln durch den Dschungel der ENCODE Daten schlagen müssten.

Eine weitere mögliche und sinnvolle Erweiterung von MutationTaster wäre die Einbindung von Phänotypinformationen vorzugsweise über die Kandidatengensuchmaschine GeneDistiller [85]. Informationen zu Krankheitssymptomen, betroffenen Geweben oder Interaktionspartnern könnten auf diese Weise genutzt werden, um einige Varianten von vorneherein auszuschließen und andere auf der Prioritätenliste weiter oben anzuordnen. Die technische

Umsetzung dieses Vorhabens wäre relativ unkompliziert, da MutationTaster und GeneDistiller bereits jetzt über einige Schnittpunkte verfügen, und für Arbeitsgruppen-interne Zwecke bereits eine (wenn auch noch nicht vollautomatisierte) Verknüpfung der beiden Programme realisiert wurde.
Ebenfalls für die nähere Zukunft geplant ist die Verbesserung des Spleiß-Tests in MutationTaster entweder durch die Implementierung eines zusätzlichen Vorhersageprogramms oder durch Entwicklung eines neuen, eigenen Programms zur Spleißing-Analyse. Außerdem soll das Modul für den bisherige Bayes Klassifikator, der keine ordinalskalierten Werte verarbeiten kann, durch ein anderes, diese Funktionalität aufweisendes, Modul ersetzt werden.

9 Zusammenfassung

Der Erfolg neuer Sequenzierungstechnologien und deren routinemäßiger Einsatz zur Aufklärung genetischer Krankheiten hat dazu geführt, dass sehr viel mehr Sequenzvarianten detektiert werden als früher. Die gefundenen Varianten müssen genauer auf ihr Krankheitspotential untersucht und priorisiert werden. Diese Aufgabe ist manuell nicht zu bewältigen, weshalb diverse *in silico* Verfahren zur Sequenzvariantenanalyse entwickelt wurden, die jedoch meist nur nicht-synonyme Varianten untersuchen können. In *Whole Exome Sequencing* Projekten ist ein Großteil der gefundenen Varianten aber intronisch oder synonym. Die Mutationsdatenbank HGMD (Human Gene Mutation Database) enthält zur Zeit gut 134.000 Mutationen, von denen nur etwa 55% in die Kategorien *missense* oder *nonsense* fallen.
MutationTaster ist ein web-basiertes Computerprogramm zur Vorhersage des Krankheitspotentials von DNA-Sequenzvarianten. Es kann sowohl intronische als auch exonische synonyme und nicht-synonyme Einzelbasenaustausche und InDels (<12bp) analysieren. In diversen Tests werden unter anderem der Grad der evolutionären Konservierung auf Protein-Ebene und DNA-Ebene, die Auswirkungen einer DNA-Veränderung auf das Protein (z.B. ein *frameshift* oder Aminosäureaustausch), und der mögliche Einfluss auf bekannte Proteindomänen untersucht. Die Genotypen aus dem 1000-Genom-Projekt (1000G) und HapMap werden genutzt, um harmlose Polymorphismen zu identifizieren, mit Hilfe der Daten von NCBI ClinVar werden bekannte Krankheitsmutationen identifiziert. Die externen, lokal installierten Computerprogramme NNSplice und polyadq detektieren Veränderungen im Spleißmuster und Poly(A)-Signal. Basierend auf den Testergebnissen prognostiziert der integrierte Bayes Klassifikator das Krankheitspotential der fraglichen Sequenzveränderung.
Der Klassifikator wurde mit mehr als 6.000.000 harmlosen Polymorphismen aus dem 1000G und mehr als 100.000 Krankheitsmutationen aus HGMD trainiert. Die anschließende Kreuzvalidierung ergab Durchschnittswerte für *accuracy*, Sensitivität und Spezifität von 90,5%, 90,5% und 90,9%. Im direkten Vergleich mit ähnlichen Vorhersageprogrammen (PolyPhen-2, SIFT und PROVEAN) schnitt MutationTaster bei der Vorhersage von 1.300 Polymorphismen und 1.300 bekannten Krankheitsmutationen mit einer *accuracy* von 88,0% am besten ab (PolyPhen-2 84,5%, SIFT 84,7%, PROVEAN 83,7%). Einzelanfragen an MutationTaster erfolgen über chromosomale Positionen oder Transkript-basiert über intuitiv zu bedienende Webschnittstellen. Für die schnelle und bequeme Analyse von NGS-Ergebnissen im VCF-Format steht eine speziell für diesen Zweck entwickelte *QueryEngine* zur Verfügung. Im benutzerfreundlichen *web interface* können eine VCF-Datei hochgeladen und diverse Filteroptionen eingestellt werden. Die Varianten werden parallel analysiert (500.000 Veränderungen / Stunde) und die Ergebnisse können anschließend im Browser sortiert, gefiltert und inspiziert oder auch heruntergeladen werden. MutationTaster ist frei verfügbar unter `http://www.mutationtaster.org`.

10 Abstract

The advent of Next Generation Sequencing (NGS) has led to a dramatically increased demand for *in silico* solutions that predict the disease-causing potential of the DNA variants identified. Most of the huge number of variants discovered by deep sequencing projects are either synonymous or intronic. In the past, these have often been neglected because their potential functional impact at the protein level tends to be less obvious than that of missense or nonsense variants. Accordingly, most of the tools available to predict the biophysical and clinical consequences of DNA sequence alterations have focused on the latter. However, only 55% of the 134,000 disease mutations currently listed by the Human Gene Mutation Database (HGMD), fall into this category.

The web-based mutation prediction tool, MutationTaster, analyses DNA sequence alterations and, uniquely, has incorporated tests for synonymous as well as for non-coding variants. It is able to analyse intronic and exonic single base exchanges and InDels up to 12 bp. MutationTaster evaluates evolutionary conservation via phyloP/phastCons and searches for regulatory features such as histone- or transcription factor-binding sites. Various data sources have been integrated. Genotypes from HapMap and the 1000 Genomes Project are used to identify neutral polymorphisms, and NCBI ClinVar to disclose known disease mutations. External software for splice site analysis (NNSplice) and poly(A)-signal analysis (polyadq) is locally installed and integrated. All the test results are passed on to the integrated Bayes classifier, which finally generates the prediction. The classifier was trained with a large set of single base pair exchanges and short InDels (up to 12 bp), comprising more than 6,000,000 confirmed polymorphisms from the 1000 Genomes Project and more than 100,000 known pathogenic mutations from HGMD. Cross-validation revealed a mean accuracy of 90.5%, with a mean sensitivity of 90.5% and a mean specificity of 90.9%. In a direct comparison using 1,300 known polymorphisms and 1,300 known disease mutations, MutationTaster displayed an accuracy of 88.0%, thereby proving superior to PolyPhen-2 (84.5%), SIFT (84.7%) and PROVEAN (83.7%).

Single queries to MutationTaster referring to chromosomal- or transcript-based positions can be submitted via intuitive web interfaces. To facilitate NGS data analysis, I developed the MutationTaster QueryEngine, a rapid and user-friendly web-based solution to directly analyse NGS variant files on our server. The web interface is easy to use, enabling geneticists to analyse their data without the help of IT specialists. After selecting a VCF file to upload, filters for coverage, homozygosity, known polymorphisms and genomic regions can be applied. The alterations are processed in highly parallel fashion (allowing the analysis of about 500,000 variants per hour) and the results can be downloaded after a reasonable time period or filtered, sorted and browsed in a web interface. MutationTaster is freely available at `http://www.mutationtaster.org`.

Literatur

[1] Strachan T, Read A: Human Molecular Genetics, Garland Science, 4. Edition, Kapitel 3.2, S.64.

[2] Ingram VM: A specific chemical difference between the globins of normal human and sickle-cell anemia haemoglobin. Nature 1956;178: 792-794.

[3] Amberger J, Bocchini CA, Scott AF, Hamosh A: McKusick's Online Mendelian Inheritance in Man (OMIM). Nucleic Acids Res. 2009 Jan;37(Database issue)

[4] WHO: http://apps.who.int/genomics/public/geneticdiseases/en/index2.html (22.11.2012)

[5] McClellan J, King MC: Genetic heterogeneity in human disease. Cell. 2010 Apr 16;141(2):210-7.

[6] Ng SB, Buckingham KJ, Lee C, Bigham AW, Tabor HK, Dent KM, Huff CD, Shannon PT, Jabs EW, Nickerson DA, Shendure J, Bamshad MJ: Exome sequencing identifies the cause of a mendelian disorder. Nat Genet. 2010 Jan;42(1):30-5.

[7] Doherty D, Bamshad MJ: Exome sequencing to find rare variants causing neurologic diseases. Neurology. 2012 Jul 31;79(5):396-7.

[8] Strachan T, Read A: Human Molecular Genetics. Wiley-Liss 2. Edition, 1999.

[9] Cotton RG, Scriver CR: Proof of disease causing mutation. Hum Mutat. 1998;12(1):1-3.

[10] Condit CM, Achter PJ, Lauer I, Sefcovic E: The changing meanings of „mutation": A contextualized study of public discourse. Hum Mutat. 2002 Jan;19(1):69-75.

[11] Oxford Wörterbuch online: http://oxforddictionaries.com/definition/english/polymorphism (27.11.2012)

[12] Brookes AJ: The essence of SNPs. Gene. 1999 Jul 8;234(2):177-86. Review.

[13] Sherry ST, Ward MH, Kholodov M, Baker J, Phan L, Smigielski EM, Sirotkin K: dbSNP: the NCBI database of genetic variation. Nucleic Acids Res. 2001 Jan 1;29(1):308-11.

[14] Database of Single Nucleotide Polymorphisms (dbSNP). Bethesda (MD): National Center for Biotechnology Information, National Library of Medicine. (dbSNP Build ID: 137) http://www.ncbi.nlm.nih.gov/SNP/ (16.11.2012)

[15] SNP FAQ Archive: http://www.ncbi.nlm.nih.gov/books/NBK44469/ (16.11.2012)

[16] SNP FAQ Archive: `http://www.ncbi.nlm.nih.gov/books/NBK44447/#Content.if_dbsnp_is_a_database_containin` (16.11.2012)

[17] Strachan T, Read A: Human Molecular Genetics, Garland Science, 4. Edition, Kapitel 13.3, S.418.

[18] Blau N, van Spronsen FJ, Levy HL: Phenylketonuria. Lancet. 2010 Oct 23;376(9750):1417-27.

[19] Seidman JG, Seidman C: Transcription factor haploinsufficiency: when half a loaf is not enough. J Clin Invest. 2002 Feb;109(4):451-5.

[20] Kralovics R, Passamonti F, Buser AS, Teo SS, Tiedt R, Passweg JR, Tichelli A, Cazzola M, Skoda RC: A gain-of-function mutation of JAK2 in myeloproliferative disorders. N Engl J Med. 2005 Apr 28;352(17):1779-90.

[21] Strachan T, Read A: Human Molecular Genetics, Garland Science, 4. Edition, Kapitel 13.4, S.429.

[22] Alberts B et al.: Molecular Biology of the Cell, Garland Science, 4. Edition, Kapitel 8, S.527

[23] Flicek P, Amode MR, Barrell D, Beal K, Brent S, Carvalho-Silva D, Clapham P, Coates G, Fairley S, Fitzgerald S, Gil L, Gordon L, Hendrix M, Hourlier T, Johnson N, Kähäri AK, Keefe D, Keenan S, Kinsella R, Komorowska M, Koscielny G, Kulesha E, Larsson P, Longden I, McLaren W, Muffato M, Overduin B, Pignatelli M, Pritchard B, Riat HS, Ritchie GR, Ruffier M, Schuster M, Sobral D, Tang YA, Taylor K, Trevanion S, Vandrovcova J, White S, Wilson M, Wilder SP, Aken BL, Birney E, Cunningham F, Dunham I, Durbin R, Fernández-Suarez XM, Harrow J, Herrero J, Hubbard TJ, Parker A, Proctor G, Spudich G, Vogel J, Yates A, Zadissa A, Searle SM: Ensembl 2012. Nucleic Acids Res. 2012 Jan;40(Database issue):D84-90.

[24] Schuler GD, Epstein JA, Ohkawa H, Kans JA: Entrez: molecular biology database and retrieval system. Methods Enzymol. 1996;266:141-62.

[25] McEntyre J, Lipman D: PubMed: bridging the information gap. Canadian Medical Association Journal (2001);164,1317–1319.

[26] CFMD: `http://www.genet.sickkids.on.ca/cftr/app` (16.11.2012)

[27] Stenson PD, Mort M, Ball EV, Howells K, Phillips AD, Thomas NS, Cooper DN: The Human Gene Mutation Database: 2008 update. Genome Med. 2009 Jan 22;1(1):13.

[28] Berkeley Drosophila Genome Project; Reese MG, Eeckman FH, Kulp D, Haussler D: Improved Splice Site Detection in Genie. J Comp Biol 19974;(3), 311-23. `http://fruitfly.org/seq_tools/splice.html` (16.11.2012)

[29] Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ: Basic local alignment search tool. J. Mol. Biol. (1990) 215:403-410.

[30] Tabaska JE, Zhang MQ: Detection of polyadenylation signals in human DNA sequences. Gene 1999;231: 77-86.

[31] Cooper GM, Shendure J: Needles in stacks of needles: finding disease-causal variants in a wealth of genomic data. Nat Rev Genet. 2011 Aug 18;12(9):628-40.

[32] 1000 Genomes Project Consortium, Abecasis GR, Auton A, Brooks LD, DePristo MA, Durbin RM, Handsaker RE, Kang HM, Marth GT, McVean GA: An integrated map of genetic variation from 1,092 human genomes. Nature. 2012 Nov 1;491(7422):56-65.

[33] Levy S, Sutton G, Ng PC, Feuk L, Halpern AL, Walenz BP, Axelrod N, Huang J, Kirkness EF, Denisov G, Lin Y, MacDonald JR, Pang AW, Shago M, Stockwell TB, Tsiamouri A, Bafna V, Bansal V, Kravitz SA, Busam DA, Beeson KY, McIntosh TC, Remington KA, Abril JF, Gill J, Borman J, Rogers YH, Frazier ME, Scherer SW, Strausberg RL, Venter JC: The diploid genome sequence of an individual human. PLoS Biol. 2007 Sep 4;5(10):e254.

[34] Wheeler DA, Srinivasan M, Egholm M, Shen Y, Chen L, McGuire A, He W, Chen YJ, Makhijani V, Roth GT, Gomes X, Tartaro K, Niazi F, Turcotte CL, Irzyk GP, Lupski JR, Chinault C, Song XZ, Liu Y, Yuan Y, Nazareth L, Qin X, Muzny DM, Margulies M, Weinstock GM, Gibbs RA, Rothberg JM: The complete genome of an individual by massively parallel DNA sequencing. Nature. 2008 Apr 17;452(7189):872-6.

[35] dbSNP: `http://www.ncbi.nlm.nih.gov/projects/SNP/snp_summary.cgi?view+summary=vi` (25.01.2013)

[36] Ramensky V, Bork P, Sunyaev S: Human non-synonymous SNPs: server and survey. Nucleic Acids Res. 2002 Sep 1;30(17):3894-900.

[37] Ng PC, Henikoff S: SIFT: Predicting amino acid changes that affect protein function. Nucleic Acids Res. 2003 Jul 1;31(13):3812-4.

[38] Mi H, Lazareva-Ulitsky B, Loo R, Kejariwal A, Vandergriff J, Rabkin S, Guo N, Muruganujan A, Doremieux O, Campbell MJ, Kitano H, Thomas PD: The PANTHER database of protein families, subfamilies, functions and pathways. Nucleic Acids Res. 2005 Jan 1;33(Database issue):D284-8.

[39] Miller MP, Kumar S: Understanding human disease mutations through the use of interspecific genetic variation. Hum Mol Genet. 2001 Oct 1;10(21):2319-28.

[40] Sunyaev S, Ramensky V, Bork P: Towards a structural basis of human non-synonymous single nucleotide polymorphisms. Trends Genet. 2000 May;16(5):198-200.

[41] Wang Z, Moult J: SNPs, protein structure, and disease. Hum Mutat. 2001 Apr;17(4):263-70.

[42] Ng PC, Henikoff S: Predicting deleterious amino acid substitutions. Genome Res. 2001;11:863–74.

[43] Chasman D, Adams RM. 2001. Predicting the functional consequences of non-synonymous single nucleotide polymorphisms: structure-based assessment of amino acid variation. J. Mol. Biol. 307:683–706.

[44] del Sol Mesa A, Pazos F, Valencia A: Automatic methods for predicting functionally important residues. J. Mol. Biol. 2003;326:1289–302.

[45] Krishnan VG, Westhead DR: A comparative study of machine-learning methods to predict the effects of single nucleotide polymorphisms on protein function. Bioinformatics 2003;19:2199–209.

[46] Adzhubei IA, Schmidt S, Peshkin L, Ramensky VE, Gerasimova A, Bork P, Kondrashov AS, Sunyaev SR: A method and server for predicting damaging missense mutations. Nat. Methods. 2010;7:248–249.

[47] Kumar P, Henikoff S, Ng PC: Predicting the effects of coding non-synonymous variants on protein function using the SIFT algorithm. Nat Protoc. 2009;4(7):1073-81.

[48] Choi Y, Sims GE, Murphy S, Miller JR, Chan AP: Predicting the Functional Effect of Amino Acid Substitutions and Indels. PLoS One. 2012;7(10):e46688.

[49] Supply P, Marceau M, Mangenot S, Roche D, Rouanet C, Khanna V, Majlessi L, Criscuolo A, Tap J, Pawlik A, Fiette L, Orgeur M, Fabre M, Parmentier C, Frigui W, Simeone R, Boritsch EC, Debrie AS, Willery E, Walker D, Quail MA, Ma L, Bouchier C, Salvignol G, Sayes F, Cascioferro A, Seemann T, Barbe V, Locht C, Gutierrez MC, Leclerc C, Bentley SD, Stinear TP, Brisse S, Médigue C, Parkhill J, Cruveiller S, Brosch R: Genomic analysis of smooth tubercle bacilli provides insights into ancestry and pathoadaptation of Mycobacterium tuberculosis. Nat Genet. 2013 Feb;45(2):172-9.

[50] Sahl JW, Gillece JD, Schupp JM, Waddell VG, Driebe EM, Engelthaler DM, Keim P: Evolution of a pathogen: a comparative genomics analysis identifies a genetic pathway to pathogenesis in Acinetobacter. PLoS One. 2013;8(1):e54287.

[51] Nowrousian M, Stajich JE, Chu M, Engh I, Espagne E, Halliday K, Kamerewerd J, Kempken F, Knab B, Kuo HC, Osiewacz HD, Pöggeler S, Read ND, Seiler S, Smith KM, Zickler D, Kück U, Freitag M: De novo assembly of a 40 Mb eukaryotic genome from short sequence reads: Sordaria macrospora, a model organism for fungal morphogenesis. PLoS Genet. 2010 Apr 8;6(4):e1000891.

[52] Young ND, Jex AR, Li B, Liu S, Yang L, Xiong Z, Li Y, Cantacessi C, Hall RS, Xu X, Chen F, Wu X, Zerlotini A, Oliveira G, Hofmann A, Zhang G, Fang X, Kang Y, Campbell BE, Loukas A, Ranganathan S, Rollinson D, Rinaldi G, Brindley PJ, Yang H, Wang J, Wang J, Gasser RB.: Whole-genome sequence of Schistosoma haematobium. Nat Genet. 2012 Jan 15;44(2):221-5.

[53] Jex AR, Liu S, Li B, Young ND, Hall RS, Li Y, Yang L, Zeng N, Xu X, Xiong Z, Chen F, Wu X, Zhang G, Fang X, Kang Y, Anderson GA, Harris TW, Campbell BE, Vlaminck J, Wang T, Cantacessi C, Schwarz EM, Ranganathan S, Geldhof P, Nejsum P, Sternberg PW, Yang H, Wang J, Wang J, Gasser RB: Ascaris suum draft genome. Nature. 2011 Oct 26;479(7374):529-33.

[54] Leser U, Naumann F: Informationsintegration. dpunkt.verlag 1.Auflage 2007. Kapitel 1, S.86.

[55] Leser U, Naumann F: Informationsintegration. dpunkt.verlag 1.Auflage 2007. Kapitel 1, S.4.

[56] Haas LM, Schwarz PM, Kodali P, Kotlar E, Rice JE, Swope WC: DiscoveryLink: A system for integrated access to life sciences data sources. IBM Systems Journal Volume 40,2001, Number 2.

[57] Nußdorfer R: Star Schema - Teil 1: Modellierung von Dimensionstabellen. Datenbank-Fokus 1998 11;16-23.

[58] ENCODE Project Consortium: An integrated encyclopedia of DNA elements in the human genome. Nature. 2012 Sep 6;489(7414):57-74.

[59] Meyer LR, Zweig AS, Hinrichs AS, Karolchik D, Kuhn RM, Wong M, Sloan CA, Rosenbloom KR, Roe G, Rhead B, Raney BJ, Pohl A, Malladi VS, Li CH, Lee BT, Learned K, Kirkup V, Hsu F, Heitner S, Harte RA, Haeussler M, Guruvadoo L, Goldman M, Giardine BM, Fujita PA, Dreszer TR, Diekhans M, Cline MS, Clawson H, Barber GP, Haussler D, and Kent WJ. The UCSC Genome Browser database: extensions and updates 2013. Nucleic Acids Res. 2013 Jan 1;41(D1):D64-9.

[60] Hunter S, Apweiler R, Attwood TK, Bairoch A, Bateman A, Binns D, Bork P, Das U, Daugherty L, Duquenne L, Finn RD, Gough J, Haft D, Hulo N, Kahn D, Kelly E, Laugraud

A, Letunic I, Lonsdale D, Lopez R, Madera M, Maslen J, McAnulla C, McDowall J, Mistry J, Mitchell A, Mulder N, Natale D, Orengo C, Quinn AF, Selengut JD, Sigrist CJ, Thimma M, Thomas PD, Valentin F, Wilson D, Wu CH, Yeats C: InterPro: the integrative protein signature database. Nucleic Acids Res. 2009 Jan;37(Database issue):D211-5.

[61] UniProt Consortium: The Universal Protein Resource (UniProt) in 2010. Nucleic Acids Res. 2010 Jan;38(Database issue):D142-8.

[62] Punta M, Coggill PC, Eberhardt RY, Mistry J, Tate J, Boursnell C, Pang N, Forslund K, Ceric G, Clements J, Heger A, Holm L, Sonnhammer EL, Eddy SR, Bateman A, Finn RD: The Pfam protein families database. Nucleic Acids Res. 2012 Jan;40(Database issue):D290-301.

[63] Portales-Casamar E, Thongjuea S, Kwon AT, Arenillas D, Zhao X, Valen E, Yusuf D, Lenhard B, Wasserman WW, Sandelin A: JASPAR 2010: the greatly expanded open-access database of transcription factor binding profiles. Nucleic Acids Res. 2010 Jan;38(Database issue):D105-10.

[64] Kinsella RJ, Kähäri A, Haider S, Zamora J, Proctor G, Spudich G, Almeida-King J, Staines D, Derwent P, Kerhornou A, Kersey P, Flicek P: Ensembl BioMarts: a hub for data retrieval across taxonomic space. Database (Oxford). 2011 Jul 23;2011:bar030.

[65] 1000-Genom-Projekt: `http://www.1000genomes.org` (18.03.2013)

[66] 1000 Genomes Project Consortium, Abecasis GR, Altshuler D, Auton A, Brooks LD, Durbin RM, Gibbs RA, Hurles ME, McVean GA: A map of human genome variation from population-scale sequencing. Nature 2010 Oct 28;467(7319):1061-73.

[67] Beschreibung VCF Format: `http://www.1000genomes.org/wiki/Analysis/Variant%20Call%20Format/vcf-variant-call-format-version-41` (01.05.2013)

[68] dbSNP: `http://www.ncbi.nlm.nih.gov/projects/SNP/snp_summary.cgi?view+summary=view+summary&build_id=137` (12.05.2013)

[69] McEntyre J, Lipman D: PubMed: bridging the information gap. Canadian Medical Association Journal (2001);164,1317–1319.

[70] The Gene Ontology Consortium: Gene ontology: tool for the unification of biology. Nat. Genet. May 2000;25(1):25-9.

[71] The International HapMap 3 Consortium: Integrating common and rare genetic variation in diverse human populations. Nature. 2010 Sep 2;467(7311):52-8.

[72] Grantham R: Amino acid difference formula to help explain protein evolution. Science 1974 185: 862-864.

[73] Henikoff S, Henikoff JG: Amino acid substitution matrices from protein blocks. Proc Natl Acad Sci U S A. 1992 Nov 15;89(22):10915-9.

[74] Siepel A, Bejerano G, Pedersen JS, Hinrichs AS, Hou M, Rosenbloom K, Clawson H, Spieth J, Hillier LW, Richards S, Weinstock GM, Wilson RK, Gibbs RA, Kent WJ, Miller W, Haussler D: Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes. Genome Res. 2005 Aug;15(8):1034-50.

[75] Pollard KS, Hubisz MJ, Rosenbloom KR, Siepel A: Detection of nonneutral substitution rates on mammalian phylogenies. Genome Res. 2010 Jan;20(1):110-21.

[76] Magrane M, Consortium U: UniProt Knowledgebase: a hub of integrated protein data. Database (Oxford). 2011 Mar 29;2011:bar009.

[77] Jensen LJ, Kuhn M, Stark M, Chaffron S, Creevey C, Muller J, Doerks T, Julien P, Roth A, Simonovic M, Bork P, von Mering C: STRING 8–a global view on proteins and their functional interactions in 630 organisms. Nucleic Acids Res. 2009 Jan;37(Database issue):D412-6.

[78] Velankar S, Alhroub Y, Best C, Caboche S, Conroy MJ, Dana JM, Fernandez Montecelo MA, van Ginkel G, Golovin A, Gore SP, Gutmanas A, Haslam P, Hendrickx PM, Heuson E, Hirshberg M, John M, Lagerstedt I, Mir S, Newman LE, Oldfield TJ, Patwardhan A, Rinaldi L, Sahni G, Sanz-García E, Sen S, Slowley R, Suarez-Uruena A, Swaminathan GJ, Symmons MF, Vranken WF, Wainwright M, Kleywegt GJ: PDBe: Protein Data Bank in Europe. Nucleic Acids Res. 2012 Jan;40(Database issue):D445-52.

[79] HGMD: `http://www.hgmd.cf.ac.uk/docs/new_back.html` (22.02.2013)

[80] Tatusova TA, Madden TL: Blast 2 sequences - a new tool for comparing protein and nucleotide sequences. FEMS Microbiol Lett. 1999;174:247-250

[81] NNSplice Datensätze: `http://www.fruitfly.org/sequence/human-datasets.html` (25.02.2013)

[82] Wikipedia: `http://en.wikipedia.org/wiki/Foreign_key` (10.05.2013)

[83] Wikipedia: `https://de.wikipedia.org/wiki/Schl%C3%BCssel_%28Datenbank%29` (10.05.2013)

[84] Descartes D, Bunce T: Programmierung mit Perl DBI. O'Reilly, 1. Auflage 2001, Kap. 3, S.67.

[85] Seelow D, Schwarz JM, Schuelke M: GeneDistiller–distilling candidate genes from linkage intervals. PLoS ONE. 2008;3(12):e3874.

[86] Seelow D, Schuelke M: HomozygosityMapper2012–bridging the gap between homozygosity mapping and deep sequencing. Nucleic Acids Res. 2012 Jul;40(Web Server issue):W516-20.

[87] Wikipedia: `https://de.wikipedia.org/wiki/Schichtenarchitektur` (27.04.2013)

[88] Herrmann E: CGI Programming with Perl in a week. Sams.net Publishing, 2. Edition (1999), Kapitel 1, S.5.

[89] Strachan T, Read A: Human Molecular Genetics, Garland Science, 4. Edition, Kapitel 1.4, S.16.

[90] Montes M, Becerra S, Sánchez-Álvarez M, Suñé C: Functional coupling of transcription and splicing. Gene. 2012 Jun 15;501(2):104-17.

[91] Moore MJ: Intron recognition comes of AGe. Nat Struct Biol. 2000 Jan;7(1):14-6.

[92] Baralle D, Baralle M: Splicing in action: assessing disease causing sequence changes. J Med Genet. 2005 Oct;42(10):737-48.

[93] Chatterjee S, Pal JK: Role of 5'- and 3'-untranslated regions of mRNAs in human diseases. Biol Cell. 2009 May;101(5):251-62.

[94] Kozak M: Point mutations define a sequence flanking the AUG initiator codon that modulates translation by eukaryotic ribosomes. Cell. 1986 Jan 31;44(2):283-92.

[95] Kozak M: At least six nucleotides preceding the AUG initiator codon enhance translation in mammalian cells.J Mol Biol. 1987 Aug 20;196(4):947-50.

[96] Kozak M: Initiation of translation in prokaryotes and eukaryotes. Gene 1999; 234:187–208.

[97] Morlé F, Lopez B, Henni T, Godet J: alpha-Thalassaemia associated with the deletion of two nucleotides at position -2 and -3 preceding the AUG codon. EMBO J. 1985 May;4(5):1245-50.

[98] C S Choong, C A Quigley, F S French, E M Wilson: A novel missense mutation in the amino-terminal domain of the human androgen receptor gene in a family with partial androgen insensitivity syndrome causes reduced efficiency of protein translation. J Clin Invest. 1996 September 15; 98(6): 1423–1431.

[99] Nagy E, Maquat LE: A rule for termination-codon position within intron-containing genes: when nonsense affects RNA abundance.Trends Biochem Sci. 1998 Jun;23(6):198-9.

[100] Chen XS, Brown CM: Computational identification of new structured cis-regulatory elements in the 3'-untranslated region of human protein coding genes. Nucleic Acids Res. 2012 Oct;40(18):8862-73.

[101] Sachs AB, Sarnow P, Hentze MW: Starting at the beginning, middle, and end: translation initiation in eukaryotes. Cell. 1997 Jun 13;89(6):831-8.

[102] Beelman CA, Parker R: Cell: Degradation of mRNA in eukaryotes. Cell. 1995 Apr 21;81(2):179-83.

[103] Bennett et al.: A rare polyadenylation signal mutation of the FOXP3 gene (AAUAAA–>AAUGAA) leads to the IPEX syndrome. Immunogenetics 2001 Aug, 53(6):435-9

[104] Higgs et al.: Alpha-thalassaemia caused by a polyadenylation signal mutation. Nature 1983, Vol. 306, No. 5941, pp. 398-400

[105] Bekman S, Cholet E: Practical mod_perl. O'Reilly 2003.

[106] Wikipedia: `http://de.wikipedia.org/wiki/Entscheidung` (28.04.2013)

[107] Wikipedia: `http://de.wikipedia.org/wiki/Maschinelles_Lernen` (28.04.2013)

[108] Poole D, Mackworth A, Goebel R: Computational Intelligence: A Logical Approach. New York: Oxford University Press. (1998) Kapitel 1, S.2.

[109] Han J, Kamber M: Data Mining Concepts and Techniques. Morgan Kaufmann, Second Edition, Kapitel 1, S.21-27.

[110] Han J, Kamber M: Data Mining Concepts and Techniques. Morgan Kaufmann, Second Edition, Kapitel 6, S.310-315.

[111] Bayes T: An essay towards solving a problem in the doctrine of chances. Philosophical Transactions of the Royal Society of London. 1763; 53:370–418.

[112] Hand DJ, Yu KM: Idiot's Bayes - Not so stupid after all? International Statistical Review 69, 385-398 (2001).

[113] Han J, Kamber M: Data Mining Concepts and Techniques. Morgan Kaufmann, Second Edition, Kapitel 6, S.288.

[114] Han J, Kamber M: Data Mining Concepts and Techniques. Morgan Kaufmann, Second Edition, Kapitel 6, S.360-362.

[115] Antonarakis SE, Chakravarti A, Cohen JC, Hardy J: Mendelian disorders and multifactorial traits: the big divide or one for all? Nat Rev Genet 2010, 11:380-384

[116] Marth GT, Yu F, Indap AR, Garimella K, Gravel S, Leong WF, Tyler-Smith C, Bainbridge M, Blackwell T, Zheng-Bradley X, Chen Y, Challis D, Clarke L, Ball EV, Cibulskis K, Cooper DN, Fulton B, Hartl C, Koboldt D, Muzny D, Smith R, Sougnez C, Stewart C, Ward A, Yu J, Xue Y, Altshuler D, Bustamante CD, Clark AG, Daly M, DePristo M, Flicek P, Gabriel S, Mardis E, Palotie A, Gibbs R; 1000 Genomes Project: The functional spectrum of low-frequency coding variation. Genome Biol. 2011 Sep 14;12(9):R84.

[117] Ferreira A, Marguti I, Bechmann I, Jeney V, Chora A, Palha NR, Rebelo S, Henri A, Beuzard Y, Soares MP: Sickle hemoglobin confers tolerance to Plasmodium infection. Cell. 2011 Apr 29;145(3):398-409.

[118] Cooper DN: Functional intronic polymorphisms: Buried treasure awaiting discovery within our genes. Hum Genomics. 2010 Jun;4(5):284-8.

[119] Yamamoto F, Clausen H, White T, Marken J, Hakomori S: Molecular genetic basis of the histo-blood group ABO system. Nature. 1990 May 17;345(6272):229-33.

[120] Houdayer C, Dehainault C, Mattler C, Michaux D, Caux-Moncoutier V, Pagès-Berhouet S, d'Enghien CD, Laugé A, Castera L, Gauthier-Villars M, Stoppa-Lyonnet D: Evaluation of in silico splice tools for decision-making in molecular diagnosis. Hum Mutat. 2008 Jul;29(7):975-82.

[121] Nalla VK, Rogan PK: Hum Mutat. Automated splicing mutation analysis by information theory. 2005 Apr;25(4):334-42.

[122] Mucaki EJ, Shirley BC, Rogan PK: Prediction of Mutant mRNA Splice Isoforms by Information Theory-Based Exon Definition. Hum Mutat. 2013 Apr;34(4):557-65.

[123] Mullan L: Pairwise sequence alignment–it's all about us! Brief Bioinform. 2006 Mar;7(1):113-5.

[124] Needleman SB, Wunsch CD: A general method applicable to the search for similarities in the amino acid sequence of two proteins. J. of Molecular Biology,1970, 48.

[125] Smith TF, Waterman MS: Identification of common molecular subsequences. J Mol Biol. 1981 Mar 25;147(1):195-7.

[126] Thein SL: Dominant beta thalassaemia: molecular basis and pathophysiology. Br J Haematol. 1992 Mar;80(3):273-7.

[127] Maquat LE, Carmichael GG: Quality control of mRNA function. Cell 104, 173–176 (2001).

[128] Zhang J, Sun X, Qian Y, Maquat LE: Intron function in the nonsense-mediated decay of beta-globin mRNA: indications that pre-mRNA splicing in the nucleus can influence mRNA translation in the cytoplasm. RNA. 1998 Jul;4(7):801-15.

[129] Chang YF, Imam JS, Wilkinson MF: The nonsense-mediated decay RNA surveillance pathway. Annu Rev Biochem. 2007;76:51-74.

[130] Bühler M, Steiner S, Mohn F, Paillusson A, Mühlemann O: EJC-independent degradation of nonsense immunoglobulin-mu mRNA depends on 3' UTR length. Nat Struct Mol Biol. 2006 May;13(5):462-4.

[131] Inácio A, Silva AL, Pinto J, Ji X, Morgado A, Almeida F, Faustino P, Lavinha J, Liebhaber SA, Romão L: Nonsense mutations in close proximity to the initiation codon fail to trigger full nonsense-mediated mRNA decay. J Biol Chem. 2004 Jul 30;279(31):32170-80.

[132] Silva AL, Ribeiro P, Inácio A, Liebhaber SA, Romão L: Proximity of the poly(A)-binding protein to a premature termination codon inhibits mammalian nonsense-mediated mRNA decay. RNA. 2008 Mar;14(3):563-76.

[133] Gray NK, Wickens M: Control of translation initiation in animals. Annu Rev Cell Dev Biol. 1998;14:399-458.

[134] Matoulkova E, Michalova E, Vojtesek B, Hrstka R: The role of the 3' untranslated region in post-transcriptional regulation of protein expression in mammalian cells. RNA Biol. 2012 May;9(5):563-76.

[135] Murphy SM, Polke J, Manji H, Blake J, Reiniger L, Sweeney M, Houlden H, Brandner S, Reilly MM: A novel mutation in the nerve-specific 5'UTR of the GJB1 gene causes X-linked Charcot-Marie-Tooth disease. J Peripher Nerv Syst. 2011 Mar;16(1):65-70.

[136] Larocque D, Pilotte J, Chen T, Cloutier F, Massie B, Pedraza L, Couture R, Lasko P, Almazan G, Richard S: Nuclear retention of MBP mRNAs in the quaking viable mice. Neuron. 2002 Dec 5;36(5):815-29.

[137] López de Silanes I, Quesada MP, Esteller M: Aberrant regulation of messenger RNA 3'-untranslated region in human cancer. Cell Oncol. 2007;29(1):1-17.

[138] Lukowski SW, Bombieri C, Trezise AE: Disrupted post-transcriptional regulation of the cystic fibrosis transmembrane conductance regulator (CFTR) by a 5'UTR mutation is associated with a CFTR-related disease. Hum Mutat. 2011 Oct;32(10):E2266-82.

[139] Yue P, Moult J: Identification and analysis of deleterious human SNPs. J Mol Biol. 2006 Mar 10;356(5):1263-74.

[140] Ferrer-Costa C, Gelpí JL, Zamakola L, Parraga I, de la Cruz X, Orozco M: PMUT: a web-based tool for the annotation of pathological mutations on proteins. Bioinformatics. 2005 Jul 15;21(14):3176-8.

[141] Bromberg Y, Yachdav G, Rost B: SNAP predicts effect of mutations on protein function. Bioinformatics. 2008 Oct 15;24(20):2397-8.

[142] Liu X, Jian X, Boerwinkle E: dbNSFP: a lightweight database of human nonsynonymous SNPs and their functional predictions. Hum Mutat. 2011 Aug;32(8):894-9.

[143] Wikipedia: `http://de.wikipedia.org/wiki/Ajax_(Programmierung)` (27.04.2013)

[144] Wikipedia: `http://de.wikipedia.org/wiki/Programmierschnittstelle` (26.04.2013)

[145] Wikipedia: `http://de.wikipedia.org/wiki/Stapelverarbeitung` (01.05.2013)

[146] Wikipedia: `http://en.wikipedia.org/wiki/RNA_splicing` (02.05.2013)

[147] Wikipedia: `http://de.wikipedia.org/wiki/Common_Gateway_Interface` (27.04.2013)

[148] Wikipedia: `http://de.wikipedia.org/wiki/Heterozygotie` (10.05.2013)

[149] Wikipedia: `http://en.wikipedia.org/wiki/Database_index` (01.05.2013)

[150] Wikipedia: `http://en.wikipedia.org/wiki/Exonic_splicing_enhancer` (02.05.2013)

[151] Wikipedia: `http://en.wikipedia.org/wiki/Exonic_splicing_silencer` (02.05.2013)

[152] Wikipedia: `http://de.wikipedia.org/wiki/Haploinsuffizienz` (10.05.2013)

[153] Wikipedia: `http://de.wikipedia.org/wiki/Hidden_Markov_Model` (02.05.2013)

[154] Wikipedia: `http://de.wikipedia.org/wiki/Hypertext_Markup_Language` (26.04.2013)

[155] Wikipedia: `http://de.wikipedia.org/wiki/Hyperlink` (26.04.2013)

[156] Wikipedia: `http://de.wikipedia.org/wiki/Hypertext` (26.04.2013)

[157] Wikipedia: `http://de.wikipedia.org/wiki/Kozak-Sequenz` (27.05.2013)

[158] Wikipedia: `http://de.wikipedia.org/wiki/MRNA` (01.05.2013)

[159] Wikipedia: `http://de.wikipedia.org/wiki/Penetranz_%28Genetik%29` (10.05.2013)

[160] Wikipedia: `http://de.wikipedia.org/wiki/Url` (01.05.2013)

[161] Wikipedia: `http://de.wikipedia.org/wiki/Webschnittstelle` (26.04.2013)

Publikationen

Knierim E, **Schwarz JM**, Schuelke M, Seelow D: CNVinspector: a web-based tool for the interactive evaluation of copy number variations in single patients and in cohorts. J Med Genet. 2013;0:1–7 (im Druck). *impact factor* 6,365.

Knierim E, Lucke B, **Schwarz JM**, Schuelke M, Seelow D: Systematic comparison of three methods for fragmentation of long-range PCR products for next generation sequencing. PLoS One. 2011;6(11):e28240. *impact factor* 4,092.

Schwarz JM, Rödelsperger C, Schuelke M, Seelow D: MutationTaster evaluates disease-causing potential of sequence alterations. Nat Methods. 2010 Aug;7(8):575-6. *impact factor* 19,276.

Seelow D, **Schwarz JM**, Schuelke M: GeneDistiller–distilling candidate genes from linkage intervals. PLoS One. 2008;3(12):e3874. *impact factor* 4,092.

Lebenslauf

Mein Lebenslauf wird aus datenschutzrechtlichen Gründen in der elektronischen Version meiner Arbeit nicht veröffentlicht.

Anhang

Abkürzungsverzeichnis

API	*application programming interface*; Programmierschnittstelle
AS	Aminosäure(n)
bzw.	beziehungsweise
DB	Datenbank
CDS	*coding sequence*; entspricht: kodierender Bereich der mRNA
dc	*disease - causing*; hier: krankheitsverursachende Mutation
d.h.	das heißt
DIP	InDel (deletion and insertion) polymorphism; InDel-Polymorphismus
ENSG	Ensembl Gen ID
ENST	Ensembl Transkript ID
GOF	*gain of function*; Funktionsgewinn
HGMD	Human Gene Mutation Database
HGNC	HUGO Gene Nomenclature Committee
InDel	Kombination aus Insertion und Deletion
LOF	*loss of funtion*; Funktionsverlust
MAF	*minor allele frequency*; Frequenz des zweithäufigsten Allels in einer Bevölkerung
mRNA	*messenger ribonucleic acid*; Boten-RNA
nd	nicht determinierbar
ndc	*not disease - causing*; hier: harmloser Polymorphismus
NGS	Next Generation Sequencing (neue Sequenzierungstechnologie)
NMD	*nonsense-mediated (mRNA) decay*; Abbau der mRNA aufgrund eines Stopcodons
nsSNP	*non-synonymous SNP*; nicht-synonymer SNP
o.ä.	oder ähnliches / oder ähnlichem / oder ähnlichen
poly(A)-Signal	Polyadenylierungssignal
PTC	*premature termination codon*; verfrühtes Stopcodon;
SNP	*single nucleotide polymorphism*; Einzelnukleotid Polymorphismus
TSS	*transcriptional start site*; Transkriptionsstartstelle
UTR	*untranslated region*; nicht translatierter Bereich der mRNA
VCF	*variant call format*; Standarddateiformat für NGS-Ergebnisse
WES	*Whole Exome Sequencing*; Exomsequenzierung
WGS	*Whole Genome Sequencing*; Genomsequenzierung
z.B.	zum Beispiel

Glossar

AJAX	von engl. *A*synchronous *J*avaScript *a*nd *X*ML; Konzept der asynchronen Datenübertragung zwischen einem Browser und dem Server. Dieses ermöglicht es, HTTP-Anfragen durchzuführen, während eine HTML-Seite angezeigt wird, und die Seite zu verändern, ohne sie komplett neu zu laden [143]
Allel	eine alternative Ausprägung eines Gens an einem bestimmten Locus
API	von engl. *application programming interface*; Programmierschnittstelle; von einem Softwaresystem bereitgestelltes Programm, das anderer Programmen die Anbindung an das System ermöglicht [144]
Batch Modus	Stapelverarbeitung, auch Batchverarbeitung genannt, ist ein Begriff aus der Datenverarbeitung und bezeichnet die automatische, vollständige und meist sequenzielle Verarbeitung der in einem Eingabemedium bereitgestellten Menge an Aufgaben oder Daten durch einen Computer [145]
branch point	für das Spleißen wichtiges Adenosin, das sich in einem bestimmten Sequenzkontext im Intron *upstream* vom Polypyrimidintrakt befindet [146]
cDNA	*complementary DNA*; DNA, die komplementär zur mRNA nach dem Spleißen ist
CDS	*coding sequence*; kodierender Bereich der mRNA
CGI	*Common Gateway Interface*; Standard für den Datenaustausch zwischen einem Webserver und weiterer Software, die Anfragen bearbeitet [147]
Client	deutsch: Kunde, Dienstnutzer. Wird hier im Kontext von Netzwerkanfragen benutzt und bezeichnet die Partei, welche eine Anfrage an einen Server stellt, der einen bestimmten Dienst bereitstellt. Dies kann ein Webbrowser oder ein anderer Computer sein.
CNV	bezeichnet eine abweichende Anzahl von Kopien eines bestimmten DNA-Abschnitts im Genom
Compound-Heterozygotie	Unterschiedliche Mutationen in beiden Kopien des gleichen Gens führen zur Merkmalsausprägung [148]
Datenbankindex	kurz Index, (im Plural „Indexe“ oder „Indizes“), ist eine von der Datenstruktur getrennte Indexstruktur in einer Datenbank, die die Suche und das Sortieren nach bestimmten Feldern beschleunigt [149]
dominant	ein dominant vererbtes Merkmal tritt bereits dann in Erscheinung, wenn nur eines der beiden Allele in der für das Merkmal entscheidenden Form vorliegt, sich also bereits im heterozygoten Zustand durchsetzt.

dominant-negativ	Von einem dominant-negativen Effekt spricht man dann, wenn eine DNA-Veränderung ein Genprodukt so verändert, dass das veränderte Genprodukt das normale Wildtyp-Genprodukt negativ beeinflusst.
Exomsequenzierung	Sequenzierung aller kodierenden Bereiche des menschlichen Genoms auf einmal, kann realisiert werden mit Hilfe neuer Sequenzierungstechnologien, dem *Next Generation Sequencing*
exonic splice enhancer	DNA Sequenzmotif in einem Exon, das sich verstärkend auf der Spleißen von RNA auswirkt [150]
exonic splice silencer	Sequenzmotif in einem Exon, das Spleißen der RNA abschwächt oder verhindert [151]
flat file	Datei in einem flachen, unkomplizierten Format, z.B. dem Textformat .txt
FTP	*file transfer protocol*; Netzwerkprotokoll zur Dateiübertragung
gDNA	von Gen-DNA; vollständige DNA-Sequenz eines Gens
Genotyp	in der Genetik: vollständiger Satz von Genen in einem Organismus.
Haplotyp	Kombination mehrerer, gemeinsam vererbter Allele auf dem selben Chromosom
Haploinsuffizienz	Haploinsuffizienz tritt dann auf, wenn ein diploider Organismus nur noch über eine intakte Kopie eines Gens verfügt (z.B. durch eine Mutation) und diese eine Kopie nicht ausreicht, um die normale Funktion des Gens im Organismus aufrecht zu erhalten. Haploinsuffizienz ist verantwortlich für einige autosomal dominante Krankheiten [152]
Heterozygotie	Mischerbigkeit in Bezug auf ein genetisches Merkmal, d.h. die zwei Allele an einem bestimmten Genort haben unterschiedliche Ausprägungen
Hidden Markov Model	stochastisches, durch unbeobachtete Zustände modelliertes System (Modell), das z.B. zur Mustererkennung oder Vorhersage genutzt werden kann [153]
Homozygotie	Reinerbigkeit in Bezug auf ein genetisches Merkmal, d.h. die zwei Allele an einem bestimmten Genort haben die gleiche Ausprägung
HTML	Hypertext Markup Language; textbasierte Auszeichnungssprache zur Strukturierung von Inhalten wie Texten, Bildern und Hyperlinks in Internet-Dokumenten [154]
Hyperlink	Verknüpfung, Verbindung; elektronischer Querverweis in einem Hypertext, der funktional einen Sprung an eine andere Stelle innerhalb desselben oder zu einem anderen elektronischen Dokument ausführt [155]

Hypertext	Text, der Querverweise (Hyperlinks) zu anderen Hypertext-Knoten enthält, und in einer Auszeichnungssprache (z.B. HTML) geschrieben ist [156]
InDel	*Insertion and Deletion*; DNA-Veränderung, bei der zwei oder mehr Nukleotide fehlen und durch zwei oder mehr andere Nukleotide ersetzt worden sind.
Job Scheduler	Software, die festlegt, welche Prozesse wann laufen und wie viel Prozessorzeit erhalten
Kozak-Sequenz	Nukleinbasen-Sequenz in der mRNA eukaryotischer Lebewesen, die einen Konsens aus den am häufigsten vorkommenden Nukleinbasen in unmittelbarer Nähe des Startcodons (ATG bzw. AUG) darstellt. Oft wird die Basenfolge gccRccATGG genannt, wobei es Unterschiede zwischen verschiedenen Gruppen von Eukaryoten gibt [157]. Neben dem ATG an den Positionen +1,+2 und +3 ist ein Purin (R) an Position -3 sowie ein G an Position +4 hoch konserviert.
Locus	physische Position eines Gens im Genom
MAF	*minor allele frequency*; Frequenz des zweithäufigsten Allels in einer Bevölkerung (welches, wenn es nur zwei Allele insgesamt gibt, das *minor*, also *seltenere* Allel ist)
Mikrosatellit	kurze, nicht kodierende DNA-Sequenzen, die sich im Genom eines Organismus oft wiederholen.
mod_perl	Modul für den Apache Webserver, das dafür sorgt, dass einmal kompilierter Code im Speicher verbleibt. Dadurch ist der Code schneller verfügbar, als dies durch eine ständige Neu-Kompilierung mit dem normalen Perl-Interpreter möglich ist.
mRNA	*messenger ribonucleic acid*; RNA-Transkript eines zu einem Gen gehörigen Teilabschnitts der DNA [158]
Next Generation Sequencing	Hochdurchsatz-Technologie(n) zur parallelen Sequenzierung großer genomischer Regionen, des kompletten Exoms oder gesamten Genoms.
nicht synonym	hier: DNA-Veränderung in der kodierenden Sequenz eines Gens, bei der der Austausch eines oder mehrerer Nukleotide auch zu einem Aminosäureaustausch führt.
NMD	*nonsense-mediated mRNA decay*; Kontrollmechanismus in eukaryotischen Zellen, der verfrühte Stopcodons (*nonsense*-Mutationen) in der mRNA erkennt und durch Degradierung der mRNA die Expression von verkürzten und möglicherweise schädlichen Proteinen verhindert.
overfitting	Überanpassung des Modells an einen bestimmten Datensatz

Penetranz	hier: Wahrscheinlichkeit, mit der ein bestimmter Genotyp zur Ausbildung eines bestimmten Phänotyps führt [159]
Perl-Modul	Datei mit der Endung *.pm*, die in der Programmiersprache Perl geschriebene Funktionen erhält. Andere Perl-Skripte (s.u.) können durch Einbinden des Moduls auf alle dort vorhandenen Funktionen zugreifen.
Perl-Skript	in der Programmiersprache Perl geschriebenes Computerprogramm (Dateiendung *.pl*)
Phänotyp	in der Genetik: die Gesamtheit aller beobachtbaren Merkmale eines Organismus (morphologische, physiologische und psychologische Eigenschaften).
Primärschlüssel	dient in einer relationalen Datenbank dazu, die Tupel (Datensätze) einer Relation (Tabelle) eindeutig zu identifizieren. Ein Primärschlüssel kann eine einzelne Spalte oder eine Spaltengruppe sein, und wird so ausgewählt, dass jede Tabellenzeile über den Werten dieser Spalte oder Spaltengruppe eine einmalige Wertekombination hat [83]
RAM-Disk	virtuelle Festplatte im Arbeitsspeicher, durch deren Einbindung echte, zeitaufwendige, Festplattenzugriffe entfallen können
rezessiv	ein rezessiv vererbtes Merkmal tritt nur in Erscheinung, wenn beide Allele in der gleichen Form, also homozygot, vorliegen.
Server	Programm, das einen Dienst (Service) anbietet
SNP	*Single Nucleotide Polymorphism*; DNA-Veränderung, die ein einzelnes Nukleotid betrifft
synonym	hier: DNA-Veränderung in der kodierenden Sequenz eines Gens, die jedoch nicht die Aminosäuresequenz des entsprechenden Proteins ändert.
Transkript	anhand der Vorlage einer DNA-Gensequenz synthetisiertes RNA-Molekül. Ein Gen kann in verschiedene, sogenannte *alternative* Transkripte umgeschrieben werden
URL	Bezeichnungsstandard für Netzwerkressourcen [160]
VCF	*variant call format*; Standarddateiformat für in NGS-Projekten gefundene DNA-Veränderungen
web interface	Webschnittstelle; Schnittstelle zu einem System, die über das Hypertext Transfer Protocol (HTTP) angesprochen werden kann; z.B. eine grafische Benutzeroberfläche (GUI), über die ein Benutzer mit Hilfe eines Webbrowsers mit dem System interagieren kann [161].

Tabellen

Datentyp	Datenquelle	Ref.
Ensembl Transkript- und Gen ID, Genpositionen (Start, Ende, Strangorientierung), Transkripte (Gen, Start, Ende), Exons (Transkript, Start, Ende), Translation (erstes und letztes Exon, TSS)	Ensembl via DB	[23]
homologe Gene in anderen Spezies	Ensembl via BioMart oder DB	[23]
homologe DNA- und Proteinsequenzen aus anderen Spezies	Ensembl via BioMart oder DB	[23]
NCBI Gen ID, HGNC Symbol, Chromosom, chrom. Position	NCBI Entrez Gene via DB	[24]
SNPs mit chrom. Position, build 37	dbSNP via DB	[13]
SNPs mit chrom. Position und Genotypfrequenzen	HapMap via DB	[71]
Polymorphismen mit Allelfrequenzen und Genotypen für Trainingsset	1000G via DB	[66]
Krankheitsmutationen für Trainingsset	HGMD via DB	[27]
SwissProt-ID, *protein features*	SwissProt / UniProtKB via DB	[76]
Werte für Aminosäureaustausche	Grantham Matrix via Textdatei	[72]
Spleißstellenvorhersage	NNSplice	[28]
Poly(A)-Signal Vorhersage	polyadq	[30]

Tabelle 11: Von MutationTaster benutzte Datenquellen und -typen. DB = Datenbank; 1000G = 1000-Genom-Projekt.

Tabellenname	Inhalt
ensembl69.entrez_mxn_ensembl	Verknüpfung von Ensembl Gen ID (intern) zu NCBI Gen-Nummer(n)
ensembl69.exon	einzelne Exons mit Exon-IDs (intern) und Exoninformationen
ensembl69.exon_transcript	Verknüpfung von Exon-IDs (intern) mit Transkripten
ensembl69.gene	einzelne Gene mit Gen IDs (intern) und Geninformationen
ensembl69.gene_sequence	Gen IDs (intern) und Gensequenzen
ensembl69.hgmd_public	HGMD-Einträge mit HGMD-ID und Informationen
ensembl69.pblast	Alignments von Proteinsequenzen anderer Spezies an humane Proteinsequenzen
ensembl69.seq_region	IDs einzelner genomischer Regionen mit Informationen (z.B. Chromosom)
ensembl69.species_geneids	Gen IDs (stabil) von homologen Genen aus anderen Spezies
ensembl69.species_sequences	Gen IDs (stabil) von homologen Genen aus anderen Spezies und Proteinsequenzen
ensembl69.species_sequences_nt	Gen IDs (stabil) von homologen Genen aus anderen Spezies und Gensequenzen
ensembl69.swissprot	SwissProt ID
ensembl69.swissprot_feature_types	alle in SwissProt annotierten Typen von Protein *features* mit *feature type* ID und Beschreibung
ensembl69.swissprot_features	alle in SwissProt annotierten Protein *features* mit SwissProt-ID, Position und *feature type* ID
ensembl69.transcript	Transkript ID (intern) und Transkriptinformationen
ensembl69.transcript2genbank	Verknüpfung von Transkript ID zu Genbank ID
ensembl69.translation	Translations ID und Informationen
ensembl_regulations69.feature_types	alle in *Ensembl regulations* annotierten Typen von regulatorischen *features* mit *feature type* Klassenzugehörigkeit und Beschreibung
ensembl_regulations69.features	alle in *Ensembl regulations* annotierten regulatorischen *features* mit chromosomaler Start- und Endposition und Typen-Zuordnung

(Fortsetzung folgt auf nächster Seite)

disease_mutations_37	Einträge aus NCBI ClinVar (u.a. bekannte Krankheitsmutationen)
genes	Ensembl Gen ID (intern) und NCBI Gen Nummer mit Geninformationen (z.B. HGNC Symbol)
genes_swissprot	Verknüpfung von NCBI Gen Nummer mit SwissProt ID
gene_position	Ensembl Gen ID (intern) und NCBI Gen Nummer mit Geninformationen (z.B. Chromosom)
markers	dbSNP Einträge
tgp_indel_genotype_frequencies	Einträge aus dem 1000G (InDels) mit Informationen
tgp_genotype_frequencies2	Einträge aus dem 1000G (SNPs) mit Informationen
mute.allelefrequencies	Einträge aus dbSNP mit Informationen (z.B. Allelfrequenzen
mute.phyloP_phastCons	phyloP und phastCons Werte für einzelne chromosomale Positionen
hm.populations	verschiedene Populationen, auf die sich dbSNP Allelfrequenzen beziehen können

Tabelle 12: Von MutationTaster genutzte Datenbanktabellen und ihre Inhalte. intern = von Ensembl intern verwendete IDs zur Identifizierung einzelner Daten; stabil = im Ensembl *web interface* angezeigte IDs zur Identifizierung von Daten (z.B. Ensembl Gen ID $ENSG00000104177$ oder Transkript ID $ENST00000324324$).

Selbständigkeitserklärung

Hiermit versichere ich, die vorliegende Dissertation eigenständig und ausschließlich unter Verwendung der angegebenen Hilfsmittel, angefertigt zu haben. Alle öffentlichen Quellen sind als solche kenntlich gemacht. Die vorliegende Arbeit ist in dieser oder anderer Form zuvor nicht als Prüfungsarbeit zur Begutachtung vorgelegt worden.

Berlin, den 30.05.2013 _____

Jana Marie Schwarz