Verteiltes Genom Browsing

Grobspezifikation

Malte Kruse, Ben Schumacher

9. November 2015

Inhaltsverzeichnis

1 Einleitung

Dies ist die Grobspezifikation für das Semesterprojekt zur verteilten Echtzeitrecherche in Genomdaten. Hierbei umfasst der Begriff "Genomdaten":

Chromosomen enthalten das Erbgut eines Menschen. Sie enthalten somit die DNA, auf welcher die einzelnen Gene abgebildet sind. Verschiedene Lebewesen haben unterschiedlich viele Chromosomen und Gene.
Für dieses Projekt beziehen wir uns ausschließlich auf die 24 Chromosomen des Menschen. Hier sind die zwei Geschlechtschromosomen X und Y bereits enthalten.

Gene bestehen aus einer Sequenz von Basen, welche ein zu synthetisierendes Protein kodieren. Jedes Gen kodiert genau ein Protein. Jedes Gen kann auf Grund seiner Genkoordinaten gefunden werden.

Genkoordinaten geben an, auf welchem Abschnitt des Chromosoms ein Gen liegt. Hierbei werden die untere und die obere Grenze angegeben.

Basenpaare ergeben sich durch die Doppelhelix-Struktur der DNA.
Es gibt vier Basen:

1. Adenin (A)
2. Cytosin (C)
3. Guanin (G)
4. Thymin (T)

von denen jeweils zwei Gegenstücke zueinander sind. Diese Paare sind A - T und C - G. Es folgt also, dass nur diese Basenpaare in der Helix vorhanden sind.
Spricht man von MBp, so sind dies Megabasenpaare, hierbei entspricht
1Mbp = 1000000 Bp (Basenpaare).

Mutationen sind Abweichungen in der Basensequenz eines Gens zum Referenzgenom. Das heißt:
Unterscheidet sich an einer spezifischen Stelle des Gens eines Probanden eine Base zu der im Referenzgenom verzeichneten, so ist dies eine Punktmutation.

Das Referenzgenom wird alle drei Monate, also einmal pro Quartal, neu festgelegt. An Hand dieses Genoms werden alle weiteren Sequenzierungen von Basenpaaren überprüft und Mutationen ermittelt.

Im folgenden Dokument wird die genaue Aufgabenstellung sowie deren Umsetzung erläutert.

2 Aufgabenstellung und Anforderungen

2.1 Aufgabenstellung

Es soll ein System entwickelt werden, welches in Echtzeit Genomdaten durchsucht und die grafische Auswertung übernimmt.

2.2 Anforderungen

- Es werden vier Datenbanken fest integriert:
  - HGMD[1]
  - dbSNP[2]
  - 1000 Genomes Project[3]
  - The Cancer Genome Atlas: Lung and Colorectal Cancer[4]

  Weiterhin wird das System für weitere (eigene) Datenquellen erweiterbar sein.
- Folgende Informationen werden bereitgestellt:
  - Metadaten: Datenquelle, Zeitpunkt des Downloads
  - Genkoordinate der Mutation (ggf. Abschnitt)
  - Mutierte Basensequenz (auch Referenzgenom muss bekannt sein)
  - (relative) Häufigkeit der Mutationen
  - Quellspezifische Sample-Attribute (Krankheit, Geschlecht, etc.)
- Eine Anfrage wird in <1 Sekunde durchlaufen.
  Zulässige Anfragen sind:
  - Intervallanfragen:
    - Anfrage mit: Chromosom, linke und rechte Grenze (Genkoordinaten)
    - Systemkorrektur: Intervall >10Mbp verkleinern auf 10Mbp
    - Ergebnis: Anzahl der Mutationen in dem Abschnitt
  - Genanfragen:
    - Anfrage mit: Name eines humanen Gens
    - Systemkorrektur: Namensvorschläge, wenn Genname nicht vollständig bekannt
    - Ergebnis: Vom System bestimmter Genabschnitt

[1] http://www.hgmd.cf.ac.uk/ac/index.php
[2] http://www.ncbi.nlm.nih.gov/SNP/
[3] http://www.1000genomes.org/
[4] http://cancergenome.nih.gov/

Folgende Filter werden zur Suchoptimierung angeboten:

  - Einschänkung auf Quelle
  - Einschänkung auf relative Häufigkeit
  - Gleichheit quellspezifischer Attribute

- Die Darstellung erfolgt:
  - je Datenquelle
  - mit einem max 10Mbp großen Ausschnitt des Chromosoms
  - mit Referenzgenom zum Vergleich bei kleinen Abschnitten
  - mit Anzeige der Basenpaare und Markierung von Mutationen in kleinen Abschnitten ($\leq$ 200Bp)
  - mit Andeutung der Verteilung der Häufigkeit von Mutationen in großen Abschnitten (> 200Bp)

  Zudem bietet das Interface interaktiv Möglichkeiten:

  - Zoomfunktion, um den dargestellten Bereich zu verkleinern oder zu vergrößern
  - Srcollen, um sich horizontal auf dem Gen fortzubewegen
  - Konfigurationsmöglichkeiten zur Darstellung und anderer Systembereiche

- Das System wird unter Mozilla Firefox laufen und auf bestehenden Webservern leicht einzurichten sein

3 Beispiele zur Nutzung: Use Cases

3.1 Use Case 1

Ein Benutzer möchte herausfinden, welche Mutationen im Bereich von 144MB – 154MB auf dem Chromosom 7 auftreten können. Hierfür wählt er die 1000 Genomes Projekt-Datenbank aus.
Das ganze Chromosom in einem eigenen Anzeigebereich (Lane) im Vergleich zum Referenzgenom angezeigt. Er erkennt durch eine Markierung, dass im Bereich von 150MB-152MB gehäuft Mutationen auftreten können. Da er diesen Bereich nicht detailliert einsehen kann, zoomt der Benutzer herein. Die Basenpaare, sowohl in der 1000 Genomes-Lane als auch vom Referenzgenom, sind immer detaillierter zu erkennen, bis er genau einsehen kann, welche Mutationen auftreten können. Bei erreichen der maximale Zoomstufe, sind alle Basenpaare des Chromosoms 7 und des Referenzgenoms zu erkennen und er kann die auftretenden Mutationen betrachten.

3.2 Use Case 2

Ein Benutzer möchte herausfinden, welche Mutationen bei einer bestimmten relativen Häufigkeit im Bereich eines Gens auftreten. Da er sich sehr für den colorectalen Bereich interessiert, wählt er The Cancer Genome Atlas (TCGA) aus. Da der Benutzer nicht den genauen Namen des Gens kennt und sich bei der Eingabe irrt, bekommt er keine Basenpaare angezeigt, sondern Vorschläge, welches Gen er gesucht haben könnte. Auf Grund der Vorschläge erinnert sich der Benutzer und wählt das korrekte Gen.
Es erscheint eine Lane, welche die Daten für den Bereich darstellt und der Benutzer erkennt die Bereiche, in welchen Mutationen auftreten. Zusätzlich werden ihm die relativen Häufigkeiten der Mutationen angezeigt.

3.3 Use Case 3

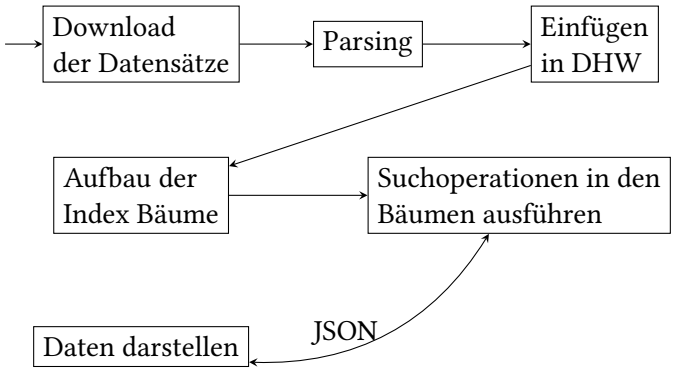
Ein Benutzer möchte sich darüber informieren, mit welcher Häufigkeit in einem bestimmten Bereich eines bestimmten Gens Mutationen auftreten können. Hierbei wählt er die HGMD mit einem Bereich von 140MB – 155MB. Da der Bereich jedoch zu groß ist, wird ihm nur der Bereich von 140MB-150MB dargestellt.
Er erkennt in diesem Bereich, dass sehr wenig Mutationen auftreten. Ihn interessiert jedoch ebenfalls, ob bei Krebspatienten höhere Mutationsraten existieren. Dazu führt er die gleiche Suche noch einmal aus, wählt jedoch zusätzlich TCGA für Lungenkrebs und TCGA für Colorectalkrebs.
Als Ergebnis werden ihm die drei Lanes untereinander angezeigt und er kann die Häufigkeit von Mutationen in den Bereichen vergleichen.

4 Umsetzung

4.1 Vorgehensweise des Systems

Damit dem System Daten zur Verfügung stehen, müssen die vier Hauptdatenbanken bei Programmstart zuerst aus dem Internet heruntergeladen werden.
Nachdem der Download erfolgreich war, werden die Datenbanken in ein einheitliches Format überführt. Ist die Vereinheitlichung vollzogen, können nun die humanen Datensätze herausgefiltert und in das Datawarehouse (DWH) eingespeichert werden.
Sind alle Daten verarbeitet und somit das DWH vollständig, stehen die Daten zum Aufbau des verteilten Index zur Verfügung.
Wurde der verteilte Index fertig gebaut, kann der Benutzer nun Anfragen an das System absenden. Die Anfragen werden vom Userinterface (GUI) an die Middleware geschickt. Hier werden die Anfragen validiert und gegebenenfalls korrigiert. Sind die Anfragen korrekt, werden diese über den verteilten Index ausgeführt und die Ergebnisse an das GUI zurückgegeben. Hat das GUI die Ergebnisdaten empfangen, werden diese grafisch aufbereitet. Nun werden die vom Benutzer angefragten Informationen grafisch dargestellt.

Folgendes Kommunikationsdiagramm verdeutlicht hierbei den Ablauf:



4.2 Arbeitsbreiche

Zur Umsetzung des Systems müssen, wie aus *4.1 Vorgehensweise des Systems* hervorgeht, drei Hauptbereiche abgedeckt werden. Diese sind:

Integration: Zuständigkeitsbereiche der Integration sind die Aufbereitung der Datenquellen und die Bereitstellung der aufbereiteten Datensätze in einer eigens entworfenen Datenbank.

Middleware: Die Middleware ist für die Kommunikation zwischen Datenbank und Frontend zuständig. Um die vom Frontend an die Middleware gesendeten Anfragen effizient bearbeiten zu können, befasst sich die Middleware mit dem Entwurf eines verteilten Index.

Dieser ermöglicht die zeiteffiziente Suche in den Daten der Datenbank und somit eine zeiteffiziente Rückgabe der Ergebnisse an das GUI.

Frontend: Das Frontend ist für die grafische Aufbereitung der Ergebnisdaten, sowie für das GUI zur Bedienung des Systems zuständig.

Im folgenden werden die Teilbereiche, die Grundzüge ihrer Realisierung, sowie die zu Verwendung findenden Technologien erläutert. Des Weiteren werden offene Fragen und Risiken der einzelnen Realisierungen angesprochen.

5 Integration

5.1 Aufgabe und Systemarchitektur

5.1.1 Aufgabe

Die Integration stellt das DWH zur Verfügung. Hierzu müssen zuerst die zu integrierenden Datenbanken aus dem Internet heruntergeladen werden. Danach erfolgt eine Aufbereitung der Daten in ein einheitliches Format. Ist dies geschehen, werden die Datensätze über einen Parser in das DWH eingespeichert.

Bevor jedoch diese Schritte realisiert werden können, muss sich die Integration mit der manuellen Analyse der Datensätze beschäftigen. Hieraus ergibt sich ein Datenbankschema, in das alle zu integrierenden Datenbanken überführt werden müssen.

5.1.2 Architektur

Die Realisierung erfolgt durch materialisierte Integration in drei Schritten:

1. Zuerst werden die Datenbanken heruntergeladen und die Datensätze vereinheitlicht.
2. Nachdem dies erfolgt ist, läuft über alle Datensätze ein Parser, der die benötigten Informationen und Metadaten herausliest und in unser Datensatzformat überführt.
3. Die Datensätze werden in das DWH geschrieben.

5.1.3 Technologien

Der Download, sowie der Parser werden voraussichtlich in Java realisiert.
Für die Datenbank wird verwendet.

5.2 Risikoanalyse

Wird ein ungeeignetes DWH entworfen, das bedeutet die ungeeignet Datenbankmanagementsoftware oder ein unpassendes Datenbankschema verwendet, kann dies zu vielen Problematiken führen. Hierunter versteht man unter anderem zeiteffiziente Zugriffsprobleme, sowie mangelnde Erweiterbarkeit des System oder ähnliche Probleme.
Um dieses Risiko zu vermeiden, werden die Datensätze sorgfältig analysiert und eine geeigneter erster DWH-Entwurf erstellt. Der Entwurf wird ebenfalls sorgfältig überprüft und bei Problemen verändert. Ist der endgültige Entwurf fertig, wird die Datenbank umgesetzt.

Es besteht ebenfalls das Risiko, dass neue Datenquellen nur schwer oder falsch eingebunden werden können. Diesem versuchen wir entgegen zu wirken, indem wir einen allgemeinen, modifizierbaren Parser entwickeln wollen, der über die Datensätze läuft. Außerdem wird die Integration neuer Datenquellen möglichst einfach gehalten und gut dokumentiert werden.

Weiterhin können Probleme mit der Speicherplatzkapazität entstehen. Da noch keine genauen

Datengrößen bekannt sind, ist dieses Risiko noch nicht einzuschätzen.
Sollte dieser Fall eintreten, muss abgeschätzt werden, wie wir damit umgehen können.

Es besteht das Risiko, dass die Datenbanken unterschiedliche Referenzgenome verwenden. In diesem Fall ist unklar, wie damit umgegangen wird.

5.3 Offene Fragen

- TCGA stellt für Projekte ohne Forschungszweck nur Metadaten zur Verfügung, keine Genomdaten. Gibt es eine Alternative oder kann die Bioinformatik "Forschungszwecke" geltend machen? [5]
- Wie sollen verfahren werden, wenn unterschiedliche Referenzgenome verwendet werden?

5.4 Schnittstellen

Die Schnitstelle zur Middleware wird eine Datenbank darstellen, welche die Daten zur weiteren Verwendung bereitstellt.

[5] Seite 12: http://cancergenome.nih.gov/PublishedContent/Files/pdfs/TCGA%20Human%20Subjects%20Protection%20and%20Data%20Access%20Policies%20Rev_2014-01-16.pdf

6 Middleware

6.1 Aufgabe und Systemarchitektur

6.1.1 Aufgabe

Die Aufgabe der Middleware umfasst den Entwurf eines verteilten Index, sowie der Suche auf der Datenbank.
Die Daten müssen also aus der Datenbank geladen werden und ein verteilter Index aufgebaut werden. Dieser Index soll hierbei eine zeiteffiziente Suchstruktur darstellen, die in Echtzeit die Datensätze durchsucht.
Des Weiteren nimmt die Middleware die Anfragen des Benutzers aus dem Frontend entgegen. Wurde eine Suchanfrage beendet, gibt die Middleware die gefundenen Datensätze an das Frontend zurück.

6.1.2 Architektur

Der verteilte Index baut für jede Datenbank einen eigenständigen Intervalltree auf. Ist dies geschehen, werden diese jeweils vollständig auf die vier VMs verteilt.
Ist dies geschehen, kann das Frontend Anfragen an die Middleware senden. Diese werden nun über die entsprechenden Intervalltrees abgearbeitet und eine Ergebnisliste wird berechnet. Hierbei werden zunächst nur die angefragten Informationen berücksichtigt. Diese Ergebnisliste wird dann in Form von JSON-Objekten an das Frontend zurückgegeben.
Nachdem diese Ergebnisse bereitgestellt wurden, wird die Middleware weitere, wahrscheinlich demnächst benötigte, Informationen vorberechnen und zwischenspeichern. Dazu gehören unter anderem weitere Zoomstufen. Hierdurch soll eine bessere zeiteffizienz bei Anfragen gewährleistet werden, sollten diese angefordert werden.

6.1.3 Technologien

Für den verteilten Index wird Java verwendet werden. Die Laufzeitumgebung der Middleware wird ein Tomcat-Server darstellen.

6.2 Risikoanalyse

Sollte nur ein Teil der Quellen angefragt werden, sind nicht alle VMs in Benutzung, sondern nur diese, deren Intervalltrees benötigt werden. Damit auch hier ein verteiler Ansatz besteht, wird dementsprechend eine Ergebnisliste erstellt, welche hinterher auf alle VMs aufgeteilt wird, bevor weitere Filter angewandt werden.

Weiterhin besteht ein Risiko bei der Größe der Intervalltrees. Sind diese zu groß, können unter Umständen nicht alle Daten vorgehalten werden und es muss auf der Datenbank selbst gearbeitet werden. Dies wirkt sich negativ auf die Zeiteffizienz von Suchanfragen aus.
Diese Problematik soll dahingehend Umgangen werden, in dem gut durchdachte Intervalltrees aufgebaut werden. Sollte die Datenbank nicht komplett geladen werden können, muss sich eine Swappingmethode überlegt werden.

6.3 Offene Fragen

- Wie werden Genpositionen dargestellt (Stelle, Intervall)?
- Sollen auch Basensequenzen dargestellt werden oder nur positionelle Daten?

6.4 Schnittstellen

Die Middleware stellt eine Schnittstelle zur Verfügung, welche es dem Frontend ermöglicht, Anfragen an die Middleware zu senden. Über diese Schnittstelle werden hinterher JSON-Objekte an das Frontend zurückgegeben.

7 Frontend

7.1 Aufgabe und Systemarchitektur

7.1.1 Aufgabe

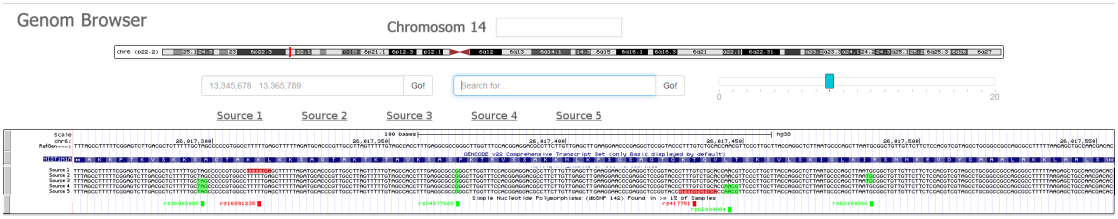
Das Frontend nimmt die Daten der Middleware entgegen und bereitet diese grafisch auf. Neben der Darstellung der Daten ist sie auch für die Konfiguration und die Bedienung des Systems durch den Benutzer zuständig.

7.1.2 Architektur

Die Architektur der GUI lässt sich an folgenden Mock-Ups verdeutlichen, welche nur der Anschauung dienen:

Dieses Mock-Up stellt eine vereinfachte Ansicht der Genomdaten dar. Diese wird bei niedrigeren Zoomstufen dargestellt.
Oben zu sehen ist das gerade dargestellte Chromosom mit seinen Banden. Hierbei stellt die rote Markierung die Position dar, an der man sich gerade befindet.
Unter der Chromosomdarstellung befinden sich die Elemente zur Benutzersuche. Links zu sehen ist das Suchfenster für Intervalle und Genkoordinaten, in der Mitte das für Gene und rechts befindet sich der Zoomstufenregler.
Im unteren Bereich des Bildes sind die einzelnen Lanes zu erkennen. Oben befindet sich das Referenzgenom. Darunter sind die Ergebnisse der Suchanfragen an die verschiedenen Datenbanken zu erkennen. Hierbei stellen die Markierungen die Positionen der Mutationen dar.

Wenn näher herangezoomt wird, sind ab einem gewissen Punkt detailiertere Genabschnitte zu sehen. Folgendes Mock-Up verdeutlicht dies:



Es sind nun die einzelnen Basen erkennbar, sowie die Mutationen im Detail ersichtlich.

7.1.3 Technologien

Für die Realisierung des Frontends werden XHTML, CSS und JavaScript verwendet. Eine Verwendung von PHP ist nicht ausgeschlossen.
Weiterhin werden folgende JavaScript-Bibliotheken benutzt:

- Flot.js
- React.js

Für den Entwurf der Mock-Ups wird Photoshop verwendet.

7.2 Risikoanalyse

Ein Risiko stellt hier die falsche Benutzung des GUI dar. Dem kann entgegengewirkt werden, in dem falsche und unzulässige Eingaben korrigiert, verhindert bzw. abgefangen werden.
Außerdem wird das GUI intuitiv und möglichst übersichtlich aufgebaut werden.

Weiterhin kann es zu Problemen führen, wenn zu viele Daten in der Ergebnisliste zurückgegeben werden, zum Beispiel bei geringen Zoomstufen viele Mutationsinformationen. Hier muss das GUI viel zur Darstellung berechnen, was somit zeitaufwändig sein kann.
Dieser Problematik wollen wir mit dem Ansatz begegnen, dass zunächst die unmittelbar benötigten Informationen von der Middleware zurückgegeben werden und dann weitere Informationen später nachgereicht werden, wenn diese schließlich auch benötigt werden.

7.3 Offene Fragen

- Was soll die GUI für Konfigurationsmöglichkeiten bieten?
- Sind spezielle Bedienelemente gewünscht?

7.4 Schnittstellen

Das Frontend stellt die Benutzerschnittstelle dar.