

1 Integration

1.1 Integrationsprozess

1.1.1 Ablauf der Integration

Die Software wird standardm g zwei Quellen integriert haben: die dbSNP und das 1000GenomeProject. Diese werden, ber ein Skript gesteuert, heruntergeladen. Wenn dieser Part abgeschlossen ist, wird ein lokaler Parser (spr mehr zu den Parsern und ihrer Arbeitsweise) alle relevanten Daten aus den heruntergeladenen Dateien extrahieren und sie in einem, fr den globalen Parser akzeptablen Format bereitstellen. Der globale Parser wird diese Dateien dann in Datense staffeln und diese dann in das Data Ware House einfgn um sie dann so der Middleware fr die Abfragen bereitzustellen.

Der lokale Parser wird bentigt, da es kein allgemeingltiges Standardformat gibt, in dem die Dateien abgespeichert werden. Die Formate sind auch keine, von dem DWH anerkannten, Dateiformate, die integriert werden knnen. Hinzu kommt, dass Sehr viele weitere Daten, die nicht notwendig sind, in den Dateien vorhanden sind und somit nur die bentigten Daten ausgelesen werden mssen.

Wenn der User weitere Quellen integrieren mchte, bentigt er zwei Dinge dafr: Ein Downloadskript, welches von der gewnschten Quelle die Dateien herunterl, und ein lokaler Parser, der die heruntergeladenen Dateien danach fr den globalen Parser bereitstellt.

1.1.2 Quellenauswahl

Unsere derzeitig benutzten Quellen werden die dbSNP und das 1000GenomeProject sein. Die HGMD werden wir nicht benutzen knnen, da diese Quelle ein Entgelt zur Benutzung der Daten verlangt, was unser Budget bersteigt. Die TCGA Datenbank werden wir vorerst nicht nicht beachten, da uns derzeit noch nicht klar ist, ob deren Daten verwendbar sind. Zu einem spren Zeitpunkt, wenn die Integration von dbSNP und 1000GenomeProject abgeschlossen ist, werden wir uns mit TCGA befassen und, sollten die Daten erreichbar und verwendbar sein, werden wir diese nehmen um den Erweiterbarkeitsprozess zu verdeutlichen.

1.1.3 Attributauswahl und -mapping

Unsere verwendeten Datenbanken stellen uns die Mutationsdaten in .vcf-Dateien bereit. Die Metadaten sind in separaten .txt-Dateien abgespeichert. Fr die Metadaten werden uns die Daten Gender und Population bereitgestellt. Fr die Mutation sind die relevanten Daten das Chromosom, in dem die Mutation auftaucht, die Position der Mutation im jeweiligen Chromosom und die vollstige Mutationssequenz. Mit diesen Daten werden wir auch im Weiteren arbeiten. Die Referenzgenome werden in einer separaten Datei abgespeichert, da diese sich nie verern, und somit ein schnellerer Zugriff auf die Daten gewleistet wird, und auch ein schnellerer Aufbau der Datenbank selber.

1.1.4 Mengengerst

Der derzeitige Stand des benötigten Speichers belgt sich auf einige hundert (aber weniger als 500) Gigabyte. Diese Angabe gilt nur für dbSNP und 1000GenomeProject, weitere Quellen werden dementsprechend mehr Speicher benötigen. Weitere Mengenangaben können wir derzeit noch nicht genau machen, diese werden im weiteren Verlauf konkreter und können dann gemacht werden.

1.1.5 Inputfile-Format

Referenzgenomname:Name des Referenzgenoms Bsp: GRCh38

Quelle:hier die Quelle angeben

\$\$

SampleID:hier Samplename

Genkoordinaten:Angabe der Koordinaten

Mutationssequenz:Sequenz

\$\$

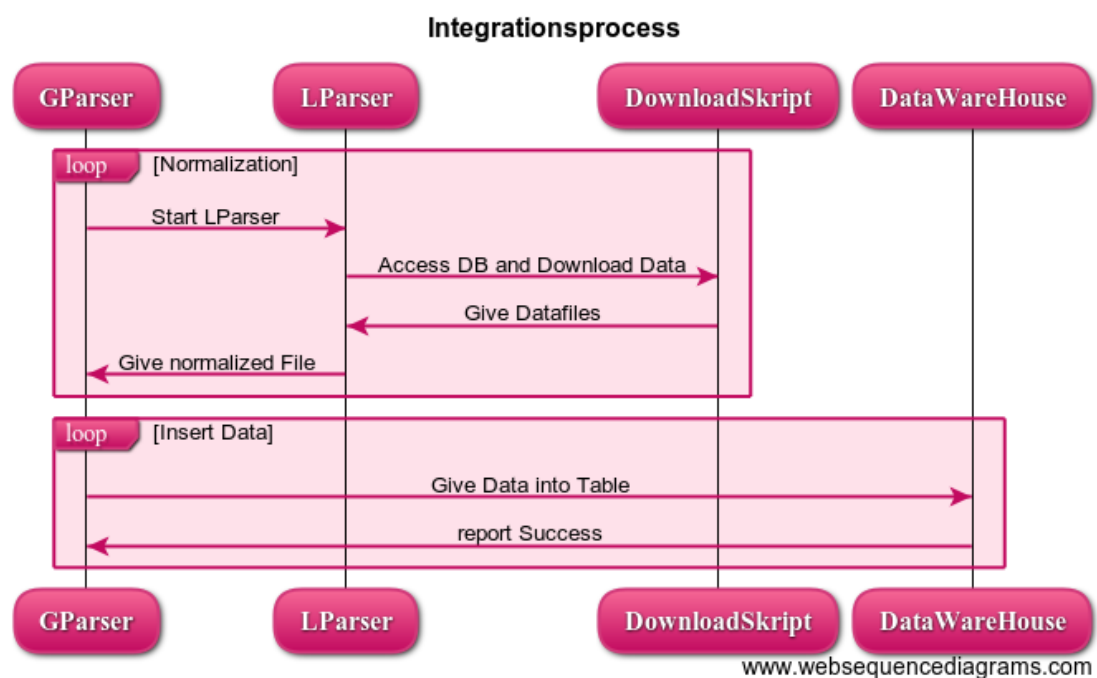
SampleID:hier SampleID aus der Datenbank

Gender:m oder f

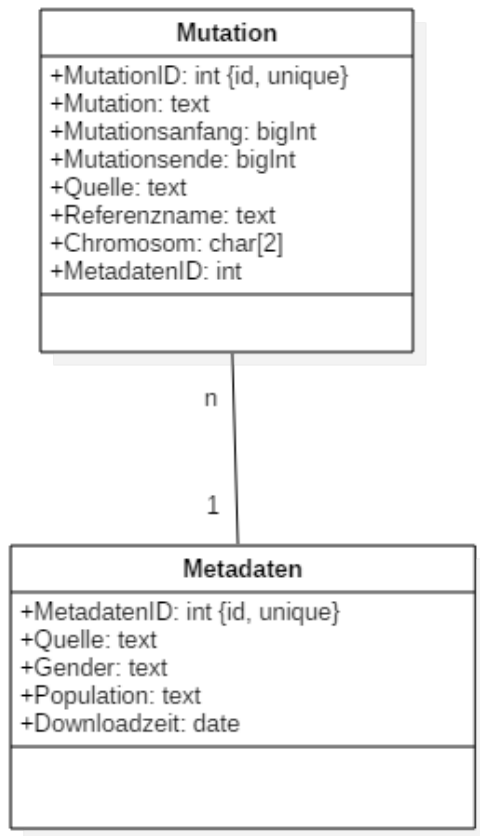
Population:drei Buchstaben bsp: GBR

EOF

1.1.6 Sequenzdiagramm



1.2 Datenbankentwurf

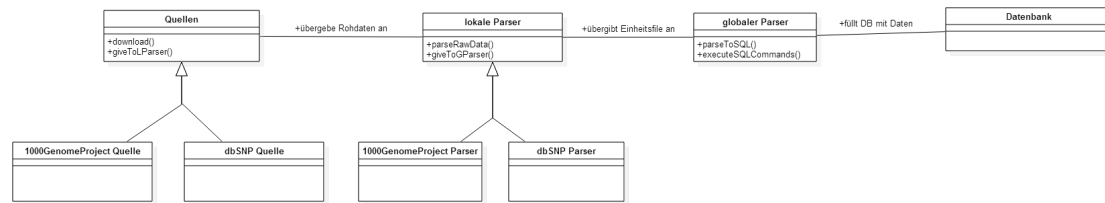


1.3 Entwurf des Parsers

Der lokale Parser wird auf die jeweilige Quelle zugeschnitten sein. Er wird die vorher heruntergeladenen Dateien entpacken, entschlüsseln und danach die relevanten Daten aus den Dateien herauslesen und in einem einheitlichen Format für den globalen Parser abspeichern.

Der globale Parser wird die Einheitsdateien der lokalen Parser nehmen, die beinhalten den Daten in einzelne Datensätze aufteilen und diese dann in der Datenbank abspeichern und sie somit der Middleware bereitstellen.

1.4 Klassendigramm



1.5 Schnittstellenspezifikation

1.5.1 Schnittstelle: Integration - Middleware

Die Schnittstelle zwischen der Integration und der Middleware ist die, im Modell sichtbare, Datenbank. Sie ist der Ort, an dem die Integration die Daten bereitstellt und von wo die Middleware sich die Daten für die Anfragen abholt.

1.5.2 Schnittstelle: Integration - Benutzer

Die Schnittstelle zwischen der Integration und dem Benutzer ist das Hinzufügen neuer Quellen. Der Nutzer wird angehalten sein, zu wissen, wie seine neue Quelle aufgebaut ist, da er selber ein Downloadskript o.ä. schreiben muss, sowie einen lokalen Parser. Diese werden an den entsprechenden Stellen im Programmcode eingefügt. Die lokalen Parser werden durch ein Interface vereinheitlicht.

1.6 Tests

1.6.1 Unit-Tests

Konkrete Tests konnten wir bisher nicht durchführen, jedoch gibt es einige Dinge zu testen: Es muss getestet werden, ob der lokale Parser arbeitet wie gewünscht, also mindestens 2 Testfälle für ihn: Bei einem, für ihn korrekten Inputfile, muss er ein entsprechend richtiges Outputfile für den globalen Parser erstellen. Sollte er ein Inputfile parsen, was nicht für ihn gedacht ist, soll er das Inputfile verwerfen oder eine Fehlermeldung ausgeben, aber auf alle Fälle das Outputfile nicht mit diesem Input erweitern. Natürlich kann das Inputfile auf verschiedene Weisen korrupt sein, was dort mehrere Testfälle notwendig macht.

Der globale Parser muss auf ähnliche Weisen getestet werden, jedoch kann man bei ihm als Voraussetzung annehmen, dass die lokalen Parser korrekt arbeiten und auch ein korrektes Outputfile erstellt haben. Somit müsste nur überprüft werden, dass der globale Parser die Daten korrekt ausliest und korrekt in die Datenbank einfügt.

Korrektes Inputfile:

Referenzgenomname:GRCh38

Quelle:1000Genom

\$\$ SampleID:HG0094

Genkoordinaten:6:19:19
Mutationssequenz: AGTCTAGTA
\$\$ SempelID:HG0094
Gender:m
Population:GRB
Download:01:01:2001

Fehlerhaftes Inputfile

Referenzgenomname:KeinFehlerMglich
Quelle:KeinFehlerMglich
\$
SapelID:KeinFehlerMglich
Genkoordinaten:67:21:15
Mutationsequenz:ATCERROR
\$\$
SampelID:KeinFehlerMglich?
Gender:h
Population:XXXX
Download:64:64:2045