

Projectgroep 10

Document Analyse.docx

Datum 28-02-2018

Versie 1.0

Groepsleden Damian Bolwerk
Jonathan Feenstra
Fini De Gruyter
Lotte Houwen
Alex Janse

ORFpred

Open Reading Frame predictie tool



Versiebeheer

Versie	Datum	Auteur	Beschrijving
0.1	21-02-18	Jonathan Feenstra	Use cases 01 t/m 04 toegevoegd.
0.2	21-02-18	Fini De Gruyter	Eerste (niet-)functionele eisen toegevoegd.
0.3	22-02-18	Jonathan Feenstra, Fini De Gruyter	Use cases 05 t/m 07 toegevoegd en use case koppelingen ingevuld.
0.4	23-02-18	Jonathan Feenstra	Use case 08 toegevoegd.
0.5	23-02-18	Lotte Houwen	Eerste opzet inleiding toegevoegd.
0.6	24-02-18	Alex Janse	Systeemarchitectuur toegevoegd.
0.7	25-02-18	Jonathan Feenstra	(Niet-)functionele eisen toegevoegd en use cases 04 en 07 gewijzigd.
0.8	26-02-18	Fini De Gruyter	Begrippenlijst gemaakt, referenties toegevoegd en inleiding aangepast
0.9	27-02-18	Jonathan, Damian, Alex, Fini	Elkaars werk bekeken en aangepast/toevoegingen gemaakt. Use case diagram toegevoegd.
1.0	28-02-18	Iedereen	Laatste bewerkingen en ideeën uitgewerkt

Contents

Versiebeheer	2
1. Introductie	4
Aanleiding	4
Doel van dit document	4
Doel van de applicatie	5
Doelgroep	5
2. Requirements	6
Doel en focus	6
Functionele eisen	6
Niet-functionele eisen	7
3. Use cases	8
Doel van use cases	8
Actoren	8
Use Case 01 - Bestand openen	8
Use Case 02 - Header selecteren	9
Use Case 03 - ORF's voorspellen	10
Use Case 04 - ORF BLASTen	11
Use Case 05 - ORF opslaan in database	12
Use Case 06 - Sequentie uit database inladen	12
Use Case 07 - Sequentie verwijderen uit database	13
Use Case 08 - Minimale ORF lengte instellen	14
4. Use case diagram	15
Doel van de use case diagram	15
Actoren	15
Use case diagram	16
5. Systeemarchitectuur	17
Doel van de systeemarchitectuur	17
Tiers	17
Front-end tier	17
Middle tier	17
Back-end Tier	17
Overzicht van het geheel	18
Toelichting	18
6. Begrippenlijst	20
7. Bronvermelding	21
8. Bijlage	22
Bijlage 1	22

1. Introductie

Aanleiding

Een reading frame is een DNA sequentie opgedeeld in codons vanaf een bepaalde positie. Aangezien elk codon bestaat uit drie nucleotiden en DNA zowel van voren als naar achter kan worden gelezen, zijn er in totaal zes mogelijke reading frames. In een reading frame wordt vaak gezocht naar open reading frames (ORF's).

Een ORF kan in de moleculaire biologie een handig hulpmiddel zijn om meer te weten over de genen van een bepaald organisme (Brent, 2005). In eukaryoten is een ORF een gedeelte van een reading frame tussen twee stopcodons, wat mogelijk onderdeel is van een exon. ORF's zoeken in een genoom is om deze reden een van de manieren om meer te weten te komen over mogelijk coderend materiaal van het betreffende organisme. Als deze ORF's ook geannoteerd kunnen worden, kan dit veel informatie verschaffen over bijvoorbeeld de identiteit (soort) en functionaliteit (in vergelijking met soortgenoten) van het organisme (Stein, 2001). Daarnaast kunnen ORF's gebruikt worden in experimenteel onderzoek naar genfuncties, zoals knock-out experimenten en het meten van genexpressies onder verschillende omstandigheden (Pevsner, 2015). Tenslotte is het mogelijk om nieuwe genen te vinden door ORF's in een genoom te voorspellen, die mogelijk coderend zijn (Stein, 2001). Aangezien het voorspellen en annoteren van ORF's in een genoom handmatig veel te veel tijd zou kosten, is het van belang om dit geautomatiseerd te kunnen doen.

De tool die ontwikkeld moet worden (ORFpred) zal in staat zijn om zowel ORF's te voorspellen in willekeurige DNA sequenties (complete genomen of contigs) en deze daarnaast ook te annoteren door er een functie aan te verbinden met behulp van vergelijkende genoomanalyse. Dit zal de bioloog in staat stellen om snel meer te weten te komen over (nieuwe) sequenties die de bioloog verkregen heeft in experimenteel onderzoek.

Doel van dit document

In dit document zullen diverse use-cases worden weergegeven met als doel de opdrachtgever en de gebruiker de diverse opties van de applicatie te laten zien. Ook worden de functionele en niet-functionele eisen van de applicatie weergegeven zodat het voor de ontwerper duidelijk is aan welke eisen de applicatie moet voldoen. De opdrachtgever kan zo zien of eraan de opdracht is voldaan en de gebruiker kan in een oogopslag de mogelijkheden van de applicatie snel samenvatten.

Doel van de applicatie

Handmatig zoeken naar ORF's in een DNA sequentie is een ondoenlijke taak. Door gebruik te maken van bio-informatica tools zoals de Basic Local Alignment Search Tool (BLAST) (Altschul et al., 1990) kan het proces meer geautomatiseerd worden. Het doel van dit project is om een applicatie te ontwikkelen die de ORF's kan voorspellen op basis van een ingevoerde DNA sequentie.

De applicatie zal de DNA-sequentie opslaan in een database. Van de DNA-sequentie worden alle mogelijke ORF's voorspeld en eventueel opgeslagen in een database. De applicatie houdt hierbij rekening met de diverse reading frames voor de te voorspellen ORF's. Deze verschillende reading frames worden ook als aminozuursequenties gevisualiseerd in de applicatie. Per ORF zal er een BLAST search tool beschikbaar zijn. Deze resultaten worden gevisualiseerd in de applicatie en opgeslagen in de database.

Doelgroep

Deze applicatie is bedoeld voor biologen. Met name biologen in de vakgebieden moleculaire genetica of genomics. Geschikte data zijn bijvoorbeeld de novo sequenties voor het vinden en annoteren van genen of bestaande sequentiedata die nog onvolledig is in de gevonden genen en annotatie.

2. Requirements

Doel en focus

De requirements van een project geven aan wat de opdrachtgever en de developers verwachten van de te bouwen applicatie. Hierbij zijn de functionaliteiten van het programma omschreven in de "Functionele eisen" en de technische achtergrond in hoe dit te bewerkstelligen is omschreven in de "Niet-functionele eisen" (Warmer & Kleppe, 2015). Bij deze (niet-)functionele eisen zijn ook Use Cases gekoppeld (zie: 3. Use Case), die meer duidelijkheid verschaffen in hoe dit precies geïmplementeerd kan worden. Aan elke eis is een prioriteit en MoSCoW waarde verbonden, die aangeven hoe belangrijk bepaalde onderdelen zijn voor de uiteindelijke applicatie (Warmer & Kleppe, 2015). Deze geven ook de focus weer van het project. Hierbij is vooral de juiste ORF voorspelling van belang. Daarnaast zijn de visualisatie en opslag van data belangrijk punten en tenslotte is het gebruiksgemak belangrijk, zoals het hebben van een webapplicatie en de snelheid waarmee het programma werkt.

Functionele eisen

ID	Use case koppeling	Beschrijving	Prioriteit	MoSCoW
F1	UC_01	De ORF voorspeller accepteert een DNA sequentie in de volgende formats: FASTA, GenBank en FASTQ.	Hoog	M
F2	UC_02	De ORF voorspeller toont alle 6 de reading frames voor het voorspellen van ORF's.	Hoog	M
F3	UC_05	De ORF voorspeller slaat de DNA sequentie en gevonden ORF's op in een database.	Hoog	M
F4	UC_03	De ORF voorspeller voorspelt mogelijke ORF's op basis van stop codons en minimale ORF lengte.	Hoog	M
F5		De ORF voorspeller bevat een GUI die door de gebruiker kan worden geopend en gesloten.	Gemiddeld	S
F6	UC_05	De gebruiker kan via deze applicatie een BLAST zoekopdracht doen met de voorspelde ORF's.	Gemiddeld	S
F7	UC_06	De ORF voorspeller slaat de resultaten uit de BLAST search van een ORF op in een database.	Gemiddeld	S
F8	UC_01	De ORF voorspeller toont de DNA sequentie zodra het bestand geopend is.	Gemiddeld	S
F9	UC_02	De ORF voorspeller visualiseert de ingevoerde DNA sequentie, 6 reading frames en gevonden ORFs bij elkaar, zodat duidelijk de relatie te zien is tussen deze onderdelen.	Gemiddeld	S
F10	UC_03	De ORF voorspeller voorspelt pas de ORF's als de gebruiker aangeeft dat er voor de ingeladen sequentie begonnen kan worden aan de voorspelling.	Gemiddeld	S
F11	UC_08	In de ORF voorspeller kan de gebruiker een keuze geven over de minimale lengte van de ORF.	Gemiddeld	C
F12	UC_02	De ORF voorspeller visualiseert de oriëntatie van de voorspelde ORF's door de forward strand boven de reverse strand weer te geven.	Gemiddeld	C
F13	UC_02	De ORF reader visualiseert de sequenties voor alle 6 reading frames middels aminozuur coderingen en stopcodons.	Gemiddeld	C
F14	UC_02	De ORF voorspeller toont om welk stopcodon het gaat (TAA, TGA, GAT) met verschillende symbolen.	Laag	C
F15	UC_03	De gebruiker kan de kleur van de highlighting instellen.	Laag	W

Niet-functionele eisen

ID	Use case koppeling	Beschrijving	Prioriteit	MoSCoW
T1		De ORF voorspeller werkt op zowel Linux (UNIX) als Windows systemen.	Hoog	M
T2		De ORF voorspeller kan door een gebruiker worden geopend en worden afgesloten.	Gemiddeld	S
T3	UC_01	De gebruiker kan middels een bladerfunctie een bestand importeren.	Hoog	S
T4		Het programma is geschreven in Java van minimaal versie 1.8 en moet gegarandeerd kunnen draaien op JVM 1.8 of hoger.	Hoog	C
T5		Het aantal ORF's dat opgeslagen kan worden per run is minimaal 100.000.	Gemiddeld	C
T6		De ORF voorspeller heeft een snelle responstijd voor voorspellen ORFs, maximaal 5 minuten.	Gemiddeld	C
T7		De ORF voorspeller heeft een snelle responstijd voor het BLASTen van een ORF, onder normale omstandigheden maximaal 5 minuten (afhankelijk van server NCBI).	Gemiddeld	C
T8		De ORF voorspeller is een Maven project van minimale versie 3.5.1	Gemiddeld	C
T9		De ORF voorspeller moet ook kunnen draaien op niet-krachtige computers (dual core of hoger).	Laag	W
T10		Van de resultaten van de ORF voorspeller wordt een backup gemaakt elke keer als een gebruiker de voorspeller runt.	Laag	W
T11		De ORF voorspeller is een webapplicatie.	Laag	W
T12		De ORF voorspeller moet bij het opslaan van de locale data niet te veel opslagcapaciteit vragen. De gebruikte opslag van de gegenereerde data mag niet meer dan 2 GB zijn.	Laag	W

3. Use cases

Doel van use cases

Use cases beschrijven de wijze waarop een systeem door verschillende actoren kan worden gebruikt. Zo vormen deze een middel om de functionele eisen weer te geven die aan het systeem worden gesteld. Verder kunnen de use cases worden gebruikt als testcases en dienen als hulpmiddel bij de communicatie met gebruikers en opdrachtgevers (Warmer & Kleppe, 2015).

Actoren

Bij de use cases van ORFpred is de enige actor de bioloog die het programma gaat gebruiken. Deze bioloog heeft beschikking tot een bestand in FASTA, GenBank of FASTQ-format met DNA sequentie(s) waarin ORF's in voorspeld moeten worden.

Use Case 01 - Bestand openen

Naam	UC_01_BestandOpenen
Versie	1
Auteur	Jonathan Feenstra
Laatste wijziging	21-02-2018
Actoren	Bioloog
Korte beschrijving	Er wordt een bestand geselecteerd en geopend.
Aannames	De gebruiker heeft kennis over verschillende bestandsformaten.
Beschrijving	1a. De bioloog kiest voor 'Open' in het menu of drukt op [Ctrl+O]. 1b. Een venster opent waarin een bestand geselecteerd kan worden. 2a. De bioloog selecteert een bestand in FASTA, GenBank of FASTQ-format met DNA sequentie(s). Andere bestandsformaten veroorzaken een exceptie. 2b. Een drop-down menu wordt gevuld met alle headers uit het geselecteerde bestand en de eerste DNA sequentie wordt gevisualiseerd.
Alternatieve route	2a. De bioloog klikt op 'Cancel'. 2b. Het bestand selectie venster sluit.
Excepties	2b. Een pop-up verschijnt met een foutmelding en instructies voor de gebruiker om het probleem te verhelpen. 3a. De bioloog klikt op de 'OK' knop. 3b. De pop-up verdwijnt.
Resultaat	Het drop-down menu is gevuld met headers uit het geopende bestand en de eerste sequentie is gevisualiseerd.

Use Case 02 - Header selecteren

Naam	UC_02_HeaderSelecteren
Versie	1
Auteur	Jonathan Feenstra
Laatste wijziging	21-02-2018
Actoren	Bioloog
Korte beschrijving	Er wordt een header geselecteerd in het drop-down menu waarbij de sequentie wordt gevisualiseerd.
Aannames	Er is een bestand geopend met minimaal één header.
Beschrijving	1a. De bioloog selecteert een header in het drop-down menu. 1b. De bijbehorende sequentie wordt gevisualiseerd.
Alternatieve route	n.v.t.
Excepties	1b. Een pop-up verschijnt met een foutmelding en instructies voor de gebruiker om het probleem te verhelpen. 2a. De bioloog klikt op de 'OK' knop. 2b. De pop-up verdwijnt.
Resultaat	De sequentie bij de geselecteerde header is gevisualiseerd.

Use Case 03 - ORF's voorspellen

Naam	UC_03_ORFsVoorspellen
Versie	1
Auteur	Jonathan Feenstra
Laatste wijziging	24-02-2018
Actoren	Bioloog
Korte beschrijving	Er worden ORF's voorspeld en gehighlight voor de ingeladen sequentie.
Aannames	Er is een eukaryotische sequentie ingeladen.
Beschrijving	1a. De bioloog klikt op de 'Zoek ORF's' knop. 1b. Alle ORF's in de sequentie vanaf de ingestelde lengte (zie UC_08) worden voorspeld. De voorspelde ORF's worden gehighlight en aanklikbaar gemaakt om te kunnen BLASTen (zie UC_04).
Alternatieve route	n.v.t.
Excepties	1b. Een pop-up verschijnt met een foutmelding en instructies voor de gebruiker om het probleem te verhelpen. 2a. De bioloog klikt op de 'OK' knop. 2b. De pop-up verdwijnt.
Resultaat	Alle voorspelde ORF's in de sequentie vanaf de ingestelde lengte zijn gehighlight en aanklikbaar gemaakt.

Use Case 04 - ORF BLASTen

Naam	UC_04_ORFBLASTen
Versie	1
Auteur	Jonathan Feenstra
Laatste wijziging	26-02-2018
Actoren	Bioloog
Korte beschrijving	Een voorspeld ORF wordt geBLAST.
Aannames	Er is een bestand geopend, een sequentie geselecteerd en ORF's zijn gehighlight. Het geselecteerde ORF is al opgeslagen in de database.
Beschrijving	<p>1a. De bioloog klikt op een gehighlightte ORF.</p> <p>1b. Er opent een venster met BLAST instellingen.</p> <p>2a. De bioloog klikt op de 'BLAST' knop.</p> <p>2b. Het aangeklikte ORF wordt met de geselecteerde instellingen geBLAST.</p> <p>2c. De BLAST resultaten worden in een tabel weergegeven.</p> <p>3a. De bioloog klikt op 'Resultaten opslaan'.</p> <p>3b. De BLAST resultaten worden in de database opgeslagen bij het geBLASTE ORF.</p>
Alternatieve route	<p>2a. De bioloog klikt op 'Cancel'.</p> <p>2b. Het BLAST venster sluit.</p>
Excepties	<p>2b. Een pop-up verschijnt met een foutmelding en instructies voor de gebruiker om het probleem te verhelpen.</p> <p>3a. De bioloog klikt op de 'OK' knop.</p> <p>3b. De pop-up verdwijnt.</p>
Resultaat	De BLAST resultaten zijn weergegeven en eventueel opgeslagen.

Use Case 05 - ORF opslaan in database

Naam	UC_05_ORFOpslaan
Versie	1
Auteur	Jonathan Feenstra
Laatste wijziging	22-02-2018
Actoren	Bioloog
Korte beschrijving	De ORF's worden opgeslagen in de database.
Aannames	De ORF's zijn gevonden.
Beschrijving	1a. De bioloog kiest voor de optie 'ORF's opslaan in database' in het menu. 1b. Alle voorspelde ORF's worden in de database opgeslagen.
Alternatieve route	n.v.t.
Excepties	2b. Een pop-up verschijnt met een foutmelding en instructies voor de gebruiker om het probleem te verhelpen. 3a. De bioloog klikt op de 'OK' knop. 3b. De pop-up verdwijnt.
Resultaat	De ORF's is opgeslagen in de database.

Use Case 06 - Sequentie uit database inladen

Naam	UC_06_SequentiInladen
Versie	1
Auteur	Fini De Gruyter
Laatste wijziging	27-02-2018
Actoren	Bioloog
Korte beschrijving	De bioloog haalt een sequentie op uit de database.
Aannames	Er zijn sequenties in de database opgeslagen.
Beschrijving	1a. De bioloog kiest voor "Sequentie inladen" in het menu. 1b. De applicatie toont een drop-down met alle headers uit de database. 2a. De bioloog klikt op "Resultaten inladen" bij een header. 2b. De ORFpred laat de sequentie en de bijbehorende ORF's zien.
Alternatieve route	2a. De bioloog klikt op 'Cancel'.
Excepties	2b. Een pop-up verschijnt met een foutmelding en instructies voor de gebruiker om het probleem te verhelpen. 3a. De bioloog klikt op de 'OK' knop. 3b. De pop-up verdwijnt.
Resultaat	De ORFpred heeft de sequentie en de bijbehorende ORF's laten zien.

Use Case 07 - Sequentie verwijderen uit database

Naam	UC_07_SequentieVerwijderen
Versie	1
Auteur	Jonathan Feenstra
Laatste wijziging	27-02-2018
Actoren	Bioloog
Korte beschrijving	De bioloog verwijdert een sequentie met bijbehorende ORF's en BLAST resultaten uit de database.
Aannames	Er zijn sequenties in de database opgeslagen.
Beschrijving	<p>1a. De bioloog kiest voor "Sequentie inladen" in het menu.</p> <p>1b. De applicatie toont een drop-down menu met alle headers uit de database.</p> <p>2a. De bioloog selecteert een header.</p> <p>2b. De sequentie bij de geselecteerde header wordt gevisualiseerd, alle opgeslagen ORF's worden gehighlight.</p> <p>3a. De bioloog klikt op 'Verwijderen'.</p> <p>3b. De sequentie wordt uit de database verwijderd.</p>
Alternatieve route	<p>2a. De bioloog klikt op 'Cancel'.</p> <p>2b. Het opslaan venster sluit.</p>
Excepties	<p>2b. Een pop-up verschijnt met een foutmelding en instructies voor de gebruiker om het probleem te verhelpen.</p> <p>3a. De bioloog klikt op de 'OK' knop.</p> <p>3b. De pop-up verdwijnt.</p>
Resultaat	De sequentie, de bijbehorende ORF's en BLAST resultaten zijn uit de database verwijderd.

Use Case 08 - Minimale ORF lengte instellen

Naam	UC_08_ORFLengte
Versie	1
Auteur	Jonathan Feenstra
Laatste wijziging	23-02-2018
Actoren	Bioloog
Korte beschrijving	De bioloog stelt de minimale lengte in nucleotiden in waarbij ORF's worden voorspeld.
Aannames	Geen*.
Beschrijving	<p>1a. De bioloog kiest voor "ORF lengte instellen" in het menu.</p> <p>1b. Er verschijnt een venster waarin een minimale ORF lengte kan worden ingevoerd, dit is een positief geheel getal. De standaard lengte is 30 nucleotiden**.</p> <p>2a. De bioloog klikt op "Toepassen".</p> <p>2b. Bij het klikken op de 'Zoek ORF's knop' worden alleen nog ORF's vanaf de ingestelde lengte gehighlight.</p>
Alternatieve route	<p>2a. De bioloog klikt op 'Cancel'.</p> <p>2b. Het venster sluit.</p>
Excepties	<p>2b. Een pop-up verschijnt met een foutmelding en instructies voor de gebruiker om het probleem te verhelpen.</p> <p>3a. De bioloog klikt op de 'OK' knop.</p> <p>3b. De pop-up verdwijnt.</p>
Resultaat	Alleen ORF's vanaf de ingestelde lengte worden voorspeld en gehighlight bij het zoeken (zie UC_03).

* De minimale ORF lengte kan al voordat er een bestand is geopend worden ingesteld.

** Gebaseerd op de ORFfinder van de National Center for Biotechnology Information (NCBI) (2018).

4. Use case diagram

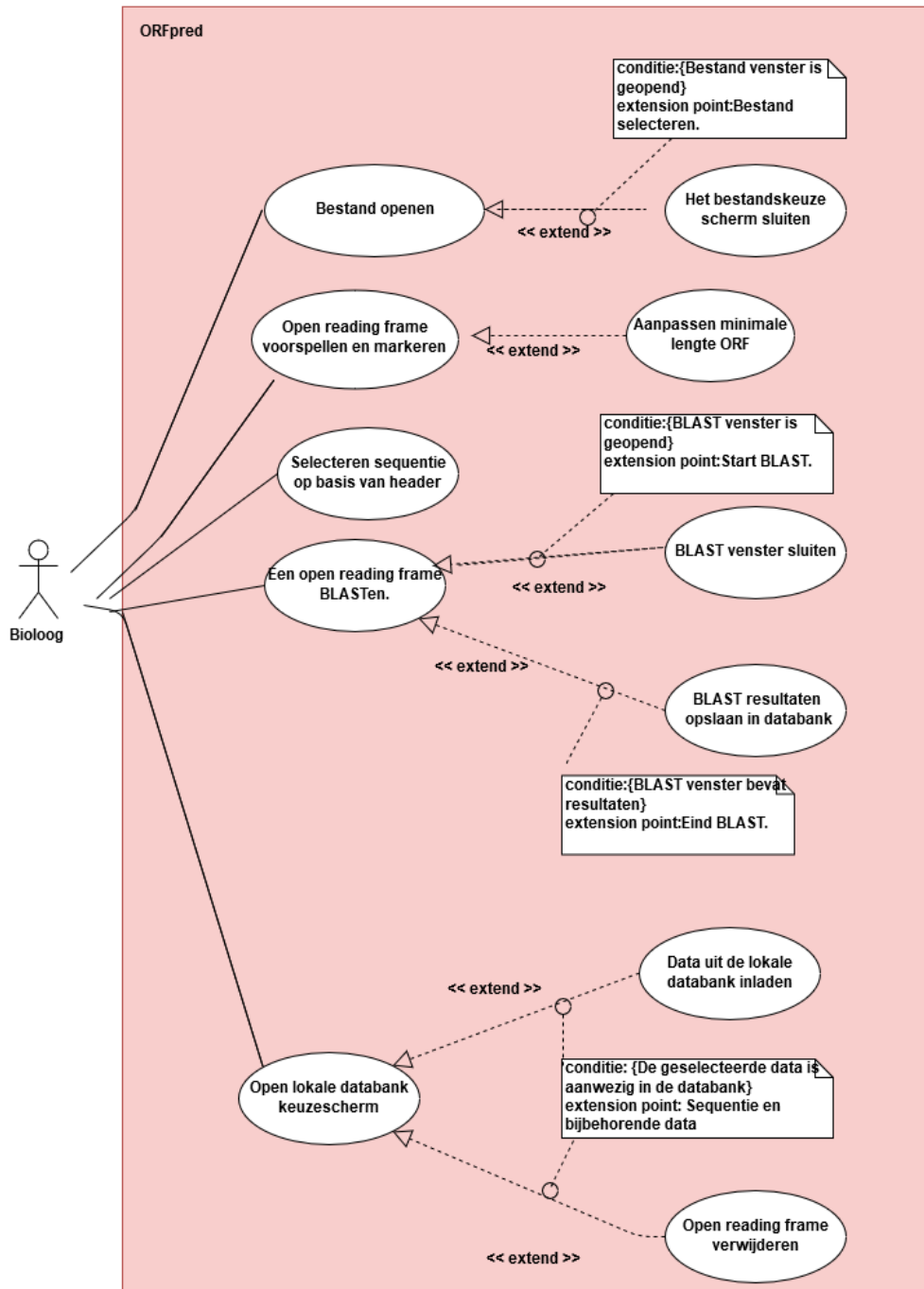
Doel van de use case diagram

Het doel van de use case diagram is het schematisch weergeven van de functionaliteiten van ORFpred en de verschillende actoren die daarvan gebruik kunnen maken. Het kan dienen als hulpmiddel bij de communicatie met gebruikers en opdrachtgevers (Warmer & Kleppe, 2015).

Actoren

Bij de use cases van ORFpred is de enige actor de bioloog die het programma gaat gebruiken. Deze bioloog heeft beschikking tot een bestand in FASTA, GenBank of FASTQ-format met DNA sequentie(s) waarin ORF's voorspeld kunnen worden.

Use case diagram



Figuur 1: Use case diagram van ORFpred. In de use case diagram is de actor weergegeven in de vorm van een stokfiguur, die betrokken is bij het gebruik van het programma. Ook is er weergegeven welke functionaliteiten (use cases) van het programma de actor kan raadplegen. De use cases hebben de vorm van een ovaal. Sommige use cases kunnen optioneel gebruikt worden binnen de eerste use case (de use case waar naar verwezen wordt). De relatie tussen deze use cases wordt aangegeven met extend. Bij de extend wordt ook aangegeven onder welke conditie de tweede use case uitgevoerd kan worden. De extension point geeft aan op welk punt de tweede use case gebruikt kan worden. Figuur is gemaakt met Draw.io, versie 8.2.3.

5. Systeemarchitectuur

Doel van de systeemarchitectuur

In figuur 2 staat de uitgebreide en in figuur 3 (bijlage 1) de versimpelde systeem architectuur. Met behulp van de systeemarchitectuur kunnen alle componenten en interacties worden weergegeven. Dit is handig voor de programmeur, systeembeheerder, databeheerder en projectleider om een goed overzicht te krijgen van het systeem.

Tiers

Front-end tier

De Front-end tier in figuur 2 beschrijft de systeem onderdelen die de gebruiker ziet en interactie mee kan hebben. Deze tier is belangrijk omdat het de interacties doorgeeft aan het Middle tier en de resultaten of foutmeldingen visualiseert. Ook is de Front-end tier de enige manier voor de gebruiker om de functionaliteit van het systeem aan te roepen. Voor dit systeem draait de Front-end tier op de Java Virtual Machine (JVM, minimale versie van 1.8) die weer draait op het besturingssysteem (operating system; O.S.) van de gebruiker. De Front-end tier wordt op het computerscherm van de gebruiker en door de gebruiker via hardware zoals computermuis, toetsenbord en aanraakscherm gevisualiseerd.

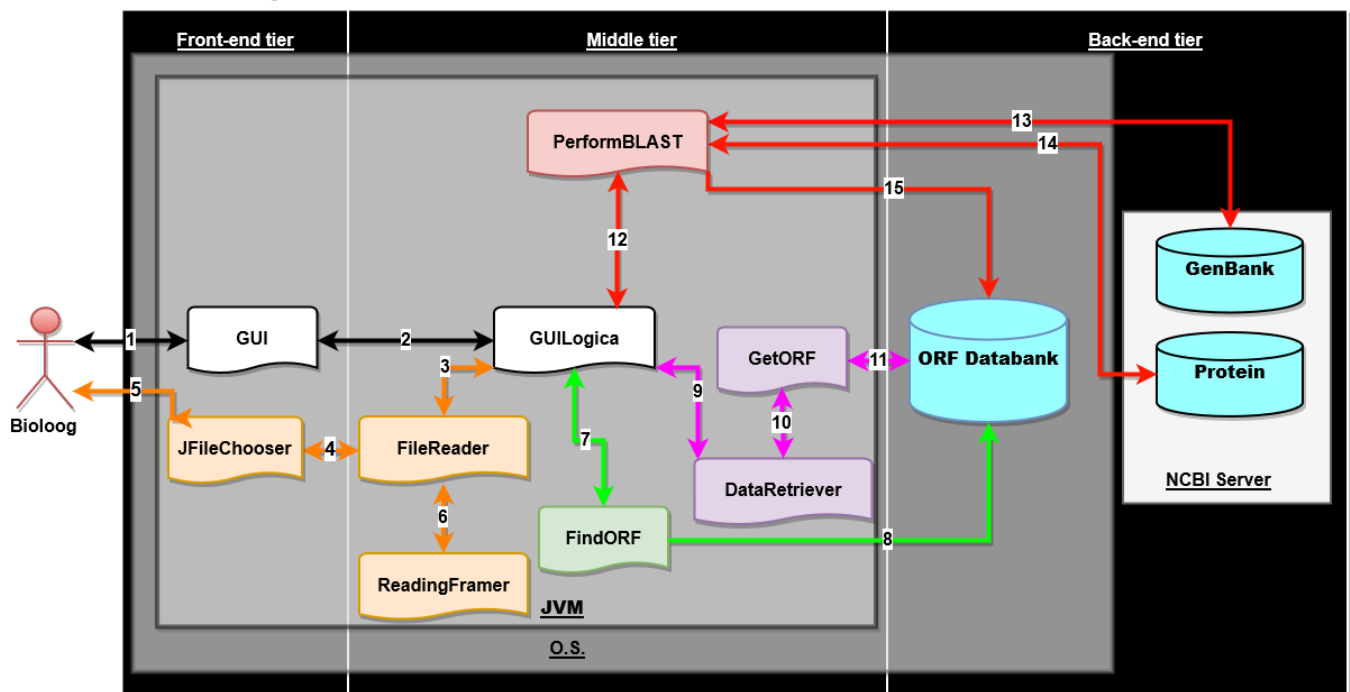
Middle tier

De Middle tier in figuur 2 beschrijft de systeem onderdelen die voor de werking (logica) van het systeem zorgen. De Middle tier is belangrijk omdat het interactie heeft met zowel de Front als de Back-end tier. Van de Front-end tier krijgt de Middle tier de gebruikers-input, vertaalt deze naar een opdracht en neemt (als het nodig is) contact op met de Back-end tier om gegevens op te halen of te versturen. Hierdoor zorgt de Middle tier voor de functionaliteit van de objecten waarmee de gebruiker interactie mee heeft. Voor dit systeem draait de Middle tier op de Java Virtual Machine (JVM) die weer draait op het besturingssysteem (operating system; O.S.) van de gebruiker.

Back-end Tier

De Back-end tier in figuur 2 beschrijft onderdelen zoals online en offline databanken. Deze tier is belangrijk omdat de Middle tier hier gegevens kan bewaren en ophalen. Voor dit systeem draait de Back-end tier op de besturingssysteem (operating system; O.S.) van de gebruiker en online op de server van de National Center for Biotechnology Information (NCBI).

Overzicht van het geheel



Figuur 2: Systeemarchitectuur van ORFpred. In de architectuur staan de verschillende tiers weergegeven in zwarte rechthoeken met witte randen. De naam van elke tier is bovenaan met wit gekleurde en onderstreepte letters weergegeven. De gebruiker van ORFpred is weergegeven als een rood stokfiguur. De systemen waarop ORFpred gaat draaien zijn weergegeven als transparante grijze vlakken. Hierbij overlapt het vlak van de besturingsysteem (operating system; O.S.) elke tier en de Java Virtual Machine (JVM, minimale versie van 1.8) alleen de Front-end tier en de Middle tier. De online National Center for Biotechnology Information (NCBI) server is weergegeven in een niet transparant grijs vlak in de Back-end tier. De databanken zijn weergegeven met blauwe cilinders in de Back-end tier. De verschillende Java classes die binnen de Front-end en Middle tier gebruikt worden zijn weergegeven in gekleurde rechthoeken met een golvende onderkant. Classes die uit Java libraries komen zijn niet weergegeven met als uitzondering JFileChooser aangezien deze een grafische gebruikersomgeving (GUI) laat zien aan de gebruiker. De kleuren zorgen voor een onderscheid van verschillende classes die (buiten de GUILogica class om) direct met elkaar in verbinding staan. De kleur zelf is arbitrair. De verbindingen tussen de verschillende onderdelen van het systeem zijn weergegeven met gekleurde en genummerde pijlen. De nummers voor de pijlen zijn arbitrair en dienen alleen als verwijzingen om de verschillende relaties uit te leggen voor de analyse van ORFpred. Figuur is gemaakt met behulp van Draw.io, versie 8.2.3.

Toelichting

Er zijn verschillende paden die binnen het systeem genomen kunnen worden (figuur 2). Aan het begin van elke sessie met ORFpred zal de bioloog het programma starten en krijgt vervolgens de grafische gebruikersomgeving (GUI) te zien (pijl 1, naar links) op het beeldscherm. Vervolgens kan de bioloog met behulp van de GUI een keuze maken welke functionaliteit van het systeem gebruikt wordt (pijl 1, naar rechts). Dit veroorzaakt een interactie met de GUILogica class (pijl 2, naar rechts). Afhankelijk van de gekozen functionaliteit zal de GUILogica interactie aangaan met de direct verbonden classes.

Als er gekozen is om een bestand te laden (zie Use Case 01) dan wordt de FileReader aangeroepen (pijl 3, beneden). FileReader roept vervolgens `javax.swing.JFileChooser` aan (Oracle, 2018) (pijl 4, links) om een GUI te laten zien (pijl 5, links) waarin de bioloog een FASTA, GenBank of FASTQ bestand kan selecteren met een DNA sequentie (pijl 5, rechts). De locatie van het gekozen bestand wordt vervolgens doorgegeven aan de FileReader (pijl 4, rechts). Deze kan worden gebruikt om het bestand op te halen, af

te lezen, controleren en de inhoud doorte sturen naar de ReadingFramer (pijl 6, beneden). Deze gaat van de DNA sequentie de 6 reading frames bepalen met de bijbehorende aminozuursequenties. De aminozuursequenties van de 6 reading frames worden geretourneerd via de FileReader class (pijl 6, omhoog) naar de GUILogica class (pijl 3, omhoog). De GUILogica zorgt ervoor dat de DNA en aminozuursequenties worden weergegeven in de GUI en zo zichtbaar is voor de gebruiker (pijl 1 en 2, links).

Als de bioloog kiest om van de reading frames de open reading frames (ORF's) te bepalen (zie Use Case 03) dan wordt de FindORF class aangeroepen (pijl 7, beneden). De FindORF gaat van de aminozuursequenties de ORF's bepalen met de opgegeven minimale lengte (Use Case 08). Deze worden met bijbehorende gegevens opgeslagen op een (lokale) databank (Use Case 05) (pijl 8) en geretourneerd naar de GUILogica (pijl 7, omhoog). De GUILogica zorgt er voor dat de gevonden ORF's worden gehighlight en klikbaar worden gemaakt voor de gebruiker (pijl 1 en 2, links).

Als de bioloog op een ORF klikt, wordt de PerformBLAST class aangeroepen (pijl 12, omhoog). De PerformBLAST zorgt ervoor dat de DNA sequentie van de geklikte ORF wordt opgestuurd naar de NCBI server. Daar wordt de sequentie geBLAST tegen de GenBank of Proteïn databank (pijl 13 en 14, rechts). De resultaten worden geretourneerd naar de PerformBLAST (pijl 13 en 14, links) en opgeslagen in de (lokale) ORF databank bij de bijbehorende ORF (pijl 15). Tevens worden de resultaten gestuurd naar de GUILogica (pijl 12, beneden), die ervoor zorgt dat de resultaten weergegeven worden in de GUI (pijl 1 en 2, links).

Als de bioloog gegevens wilt ophalen die zich al in de databank bevinden, wordt de DataRetriever class aangeroepen (pijl 9, beneden). De DataRetriever kan met behulp van een zoekopdracht van een FASTA, GenBank of FASTQ header de bijbehorende gegevens ophalen (pijl 11). Deze gegevens worden geretourneerd naar de GUILogica (pijl 9, links) die de resultaten weergeeft in de GUI (pijl 1 en 2, links). Ook wordt de DataRetriever aangeroepen als er gegevens uit de databank verwijderd moeten worden.

Voor elk voorbeeld geldt dat als er ergens een error optreedt of een niet logische resultaat wordt gevonden deze wordt opgevangen en vermeld aan de gebruiker. De mogelijke errors zijn vermeld in het Use Case hoofdstuk.

6. Begrippenlijst

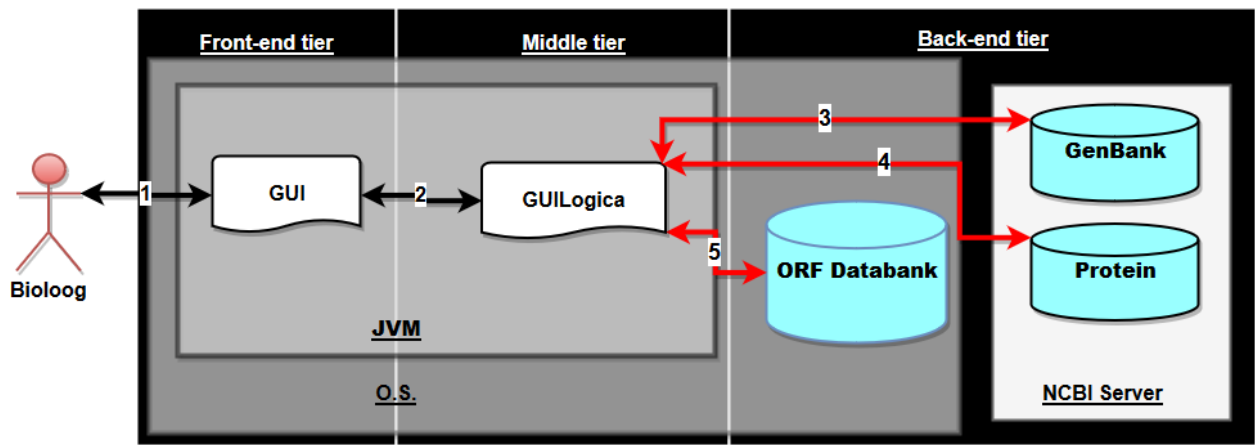
Begrip	Betekenis
ORF	Staat voor: Open Reading Frame. Een van de 6 mogelijke leesramen waarin een mRNA mogelijk vertaald kan worden in een eiwit.
FASTA	Bestandsformaat met headers beginnend met '>' en daaronder de sequentie. FASTA slaat de biologische sequentie op en in de header wordt informatie over de betreffende sequentie opgeslagen
FASTQ	Bestandsformaat met headers beginnend met '@', daaronder de sequentie, daaronder een '+' en vervolgens de kwaliteitsscore in ASCII formaat. FASTQ slaat zowel de biologische sequentie als de corresponderende kwaliteitsscores op van elke nucleotide.
GenBank	Bestandsformaat van de NIH genetic sequence database, dat behalve de sequenties ook annotatie bevat. er zijn vaste woorden aan het begin van een regel die bepaalde informatie weergeeft. Elke entry begint altijd met het woord 'LOCUS', de sequentie staat altijd achter 'ORIGIN' en het einde van de entry wordt aangegeven met '//'.
BLAST	Staat voor: Basic Local Alignment Search Tool. Een algoritme om nucleotide of eiwit sequenties met een sequentie database te vergelijken en de statistische significantie te berekenen (Altschul et al., 1990).
Maven	Een tool die gebruikt kan worden om Java projecten te beheren (Miller et al., 2010).
MoSCoW-coderingen.	M staat voor "Must have", S staat voor "Should have", C staat voor "Could have" en W staat voor "Won't have". Het is een manier om prioriteiten te stellen aan het ontwikkelingsproces van de software. De (niet-)functionele eisen kunnen met deze codering worden ingedeeld in verschillende groepen.
JVM	Staat voor: Java Virtual Machine. Een platformafhankelijke omgeving voor het uitvoeren van Java bytecode.

7. Bronvermelding

- Altschul, S.F., Gish, W., Miller, W., Myers, E.W. & Lipman, D.J. (1990). Basic local alignment search tool. *J. Mol. Biol.* 215:403-410.
- Brent, M. R. (2005). Genome annotation past, present, and future: how to define an ORF at each locus. *Genome research*, 15(12), 1777-1786.
- Draw.io (Versie 8.2.3). Desktop diagramming application by the technology leaders in web diagramming. Northampton UK: Atlassian Confluence.
- Miller, F.P., Vandome, A.F. & McBrewster J. (2010). *Apache Maven*. Alpha Press.
- Oracle (2018) *Package javax.swing* [online document]. Geraadpleegd op 24-02-2018 van https://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html#package_description
- National Center for Biotechnology Information (NCBI). (z.d.). *ORFfinder*. Geraadpleegd op 26-02-2018 van <https://www.ncbi.nlm.nih.gov/orffinder/>
- Pevsner, J. (2015). *Bioinformatics and Functional Genomics third edition*. John Wiley & Sons, Inc.
- Stein, L. (2001). Genome annotation: from sequence to biology. *Nature reviews genetics*, 2(7), 493.
- Warmer, J. & Kleppe, A. (2015). *Praktisch UML* (5th ed.). Amsterdam: Pearson Benelux.

8. Bijlage

Bijlage 1



Figuur 3. Versimpelde weergave van de systeemarchitectuur van ORFpred. In de architectuur staan de verschillende tiers weergegeven in zwarte rechthoeken met witte randen. De naam van elke tier is bovenaan met wit gekleurde en onderstreepte letters weergegeven. De gebruiker van ORFpred is weergegeven als een rood stokfiguur. De systemen waarop ORFpred gaat draaien zijn weergegeven als transparante grijze vlakken. Hierbij overlapt het vlak van de besturingssysteem (operating system; O.S.) elke tier en de Java Virtual Machine (JVM, minimale versie van 1.8) alleen de Front-end tier en de Middle tier. De online National Center for Biotechnology Information (NCBI) server is weergegeven in een niet transparant grijs vlak in de Back-end tier. De databanken zijn weergegeven met blauwe cilinders in de Back-end tier. De verbindingen tussen de verschillende onderdelen van het systeem zijn weergegeven met gekleurde en genummerde pijlen. De nummers voor de pijlen zijn arbitrair en dienen alleen als verwijzingen om de verschillende relaties uit te leggen voor de analyse van ORFpred. Figuur is gemaakt met behulp van Draw.io, versie 8.2.3.