**Technische Documentatie**

Sanne Post en Ruben Kampf

*26-06-2019*

**Inhoud**

[1. Inleiding 3](#_Toc12483610)

[2. Hoe de applicatie werkt 4](#_Toc12483611)

[3. BLAST Python script 10](#_Toc12483612)

[4. Database 11](#_Toc12483613)

[5. Software architectuur 13](#_Toc12483614)

[6. Testscripts en resultaten 14](#_Toc12483615)

[7. Literatuurlijst 17](#_Toc12483616)

# Inleiding

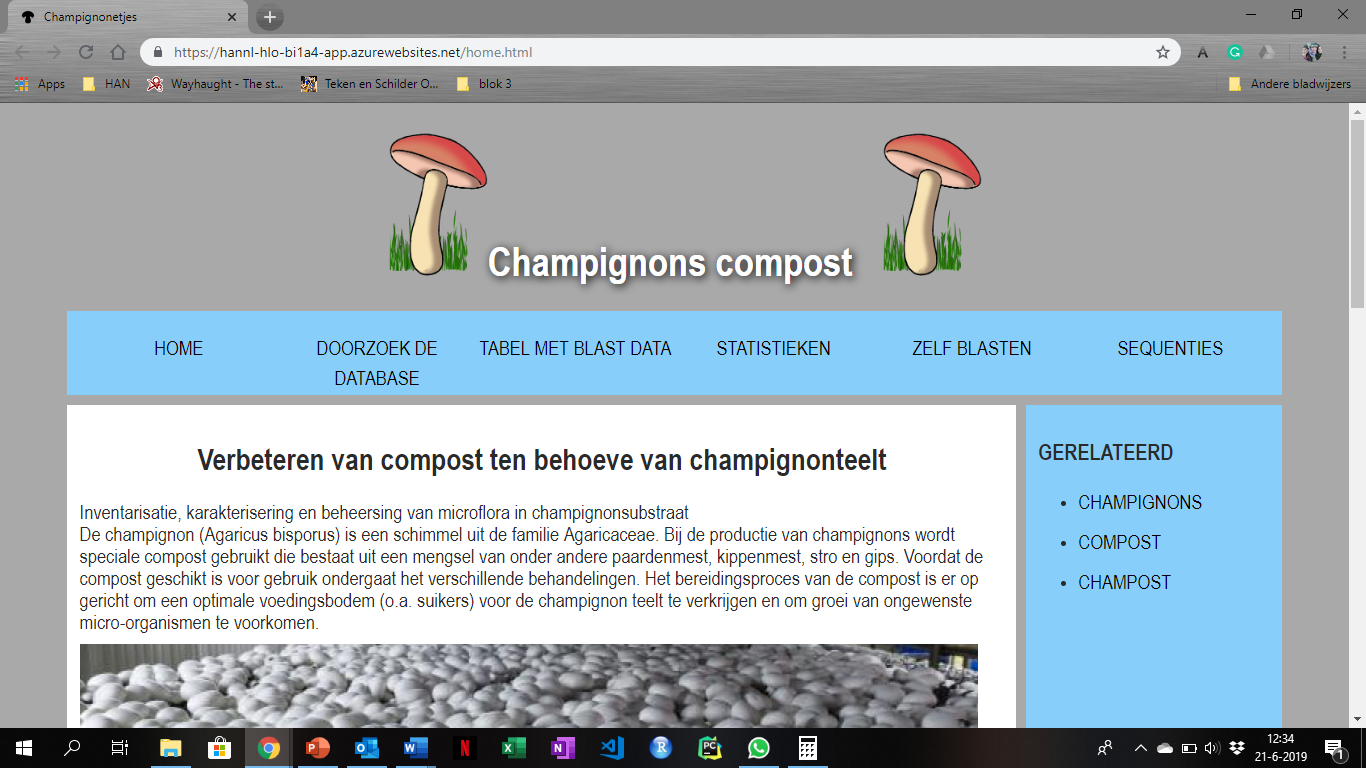
Om de samenstelling van de microflora van de compost van champignons zo danig te kunnen aanpassen dat het optimaal is, is er aan het HAN Biocentre gevraagd om de samenstelling van de microflora te bestuderen. Hiervoor is er een metagenomics onderzoek uitgevoerd. Het DNA is hierna gesequenced is met behulp van paired-end illumina sequencing. De reads zijn opgeslagen in FastQ format. De data bestaat uit een header, sequentie en een score geschreven in ASCII-code. De reads zijn hierna automatisch aligned m.b.v. Basic Local Alignment Search Tool (BLAST) tegen de nr database. Resultaten die zijn verkregen uit deze BLAST zijn vervolgens automatisch opgeslagen in een database. De database is verbonden aan een webapplicatie ([https://hannl-hlo-bi1a4-app.azurewebsites.netl](https://hannl-hlo-bi1a4-app.azurewebsites.net/home.html)).

De webapplicatie is gebouwd om biologische data uit een database te weergeven, met als doel dat de data eenvoudig en overzichtelijk toegankelijk is voor biologen. In deze documentatie zal het script van de applicatie, en de werking daarvan, beschreven worden.

# Hoe de applicatie werkt

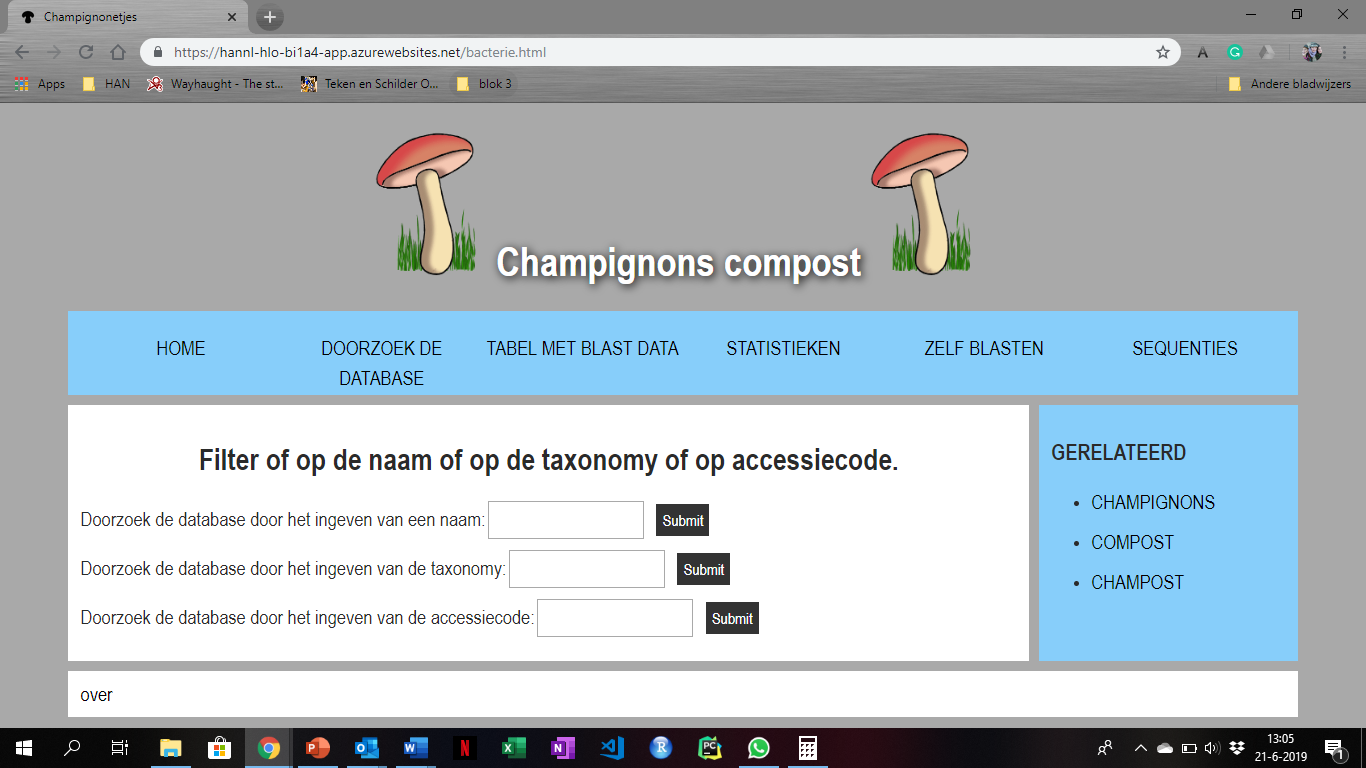
De webapplicatie is gebouwd om biologische data uit een database te weergeven, met als doel dat de data eenvoudig en overzichtelijk toegankelijk is voor biologen. Om de applicatie te raadplegen kun je klikken op de volgende url: <https://hannl-hlo-bi1a4-app.azurewebsites.net/>

Figuur 1 is wat je ziet wanneer je op de url klikt. Op deze pagina kan aan de bovenkant op verschillende kopjes gedrukt worden (bijv. Doorzoek de database, Tabel met blast data, etc.). Deze zullen je leiden naar andere pagina’s te gaan. Bij het klikken op ‘Home’ krijgt u tevens onderstaande pagina te zien.

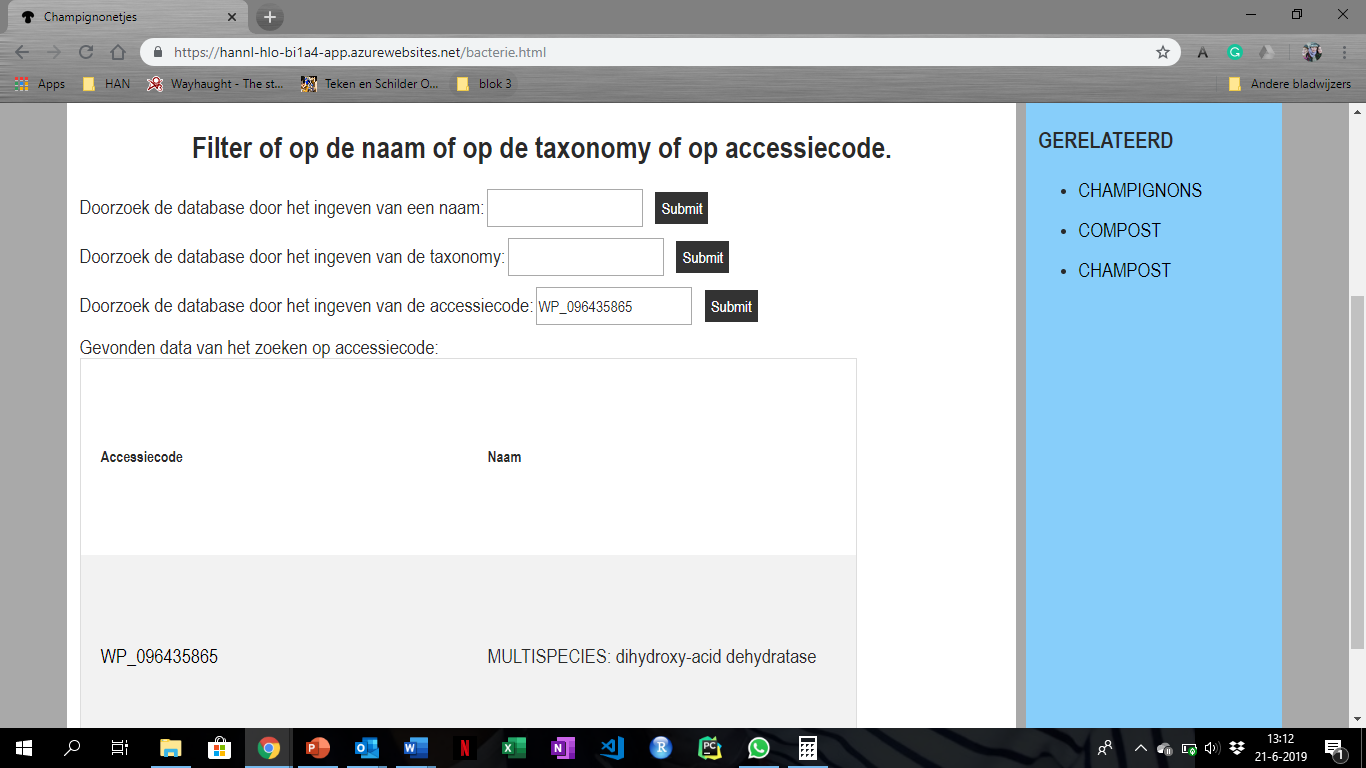


*Figuur 1 Homepage. Eigen werk van auteur.*

Bij het klikken op ‘Doorzoek de database’ zal de volgende pagina worden weergegeven als te zien in figuur 2. Op de ‘Doorzoek de database’ pagina kan er gezocht worden op een naam, een keyword in de taxonomy of op een accessiecode. Om resultaten te krijgen moet er op de submit knop gedrukt worden. In figuur 3 is te zien wat je krijgt wanneer je op de accessie code filtert.

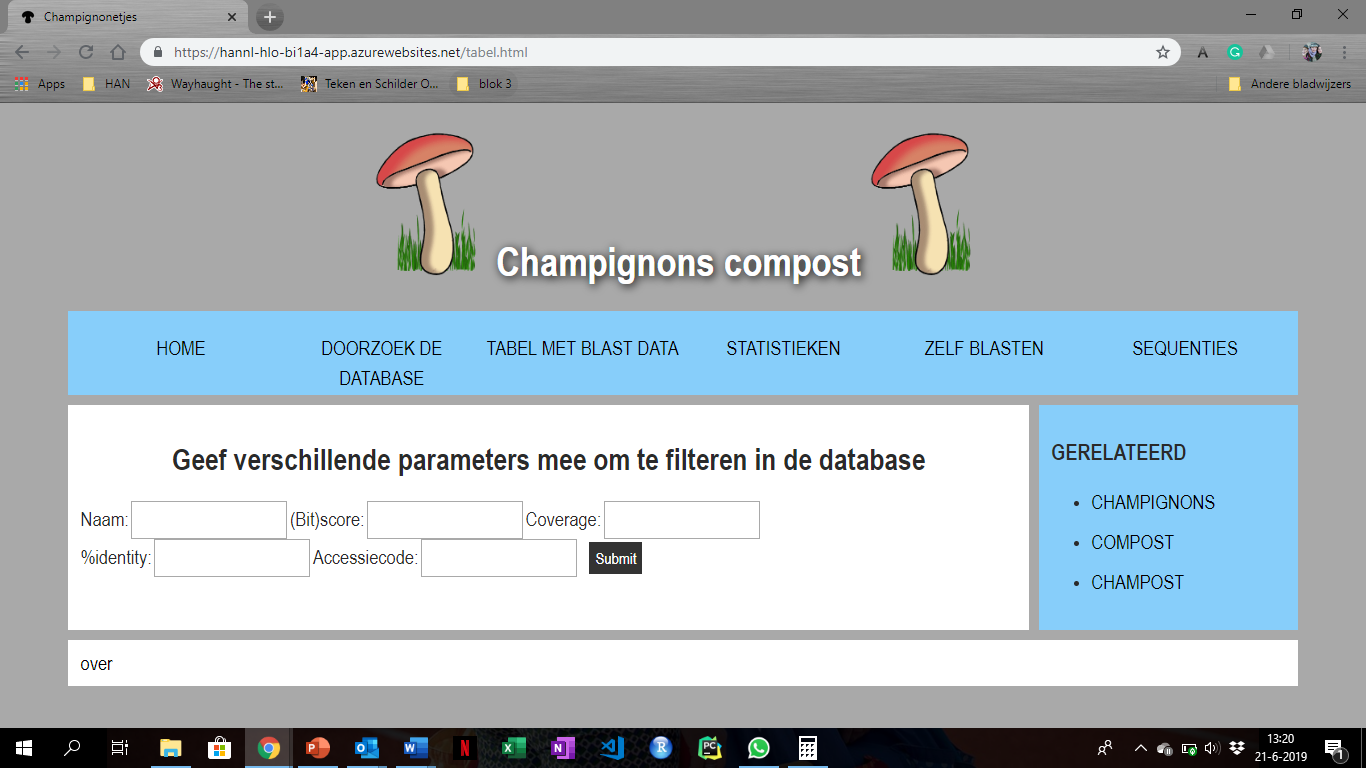


*Figuur 2 pagina ‘Doorzoek de database’. Eigen werk van auteur.*



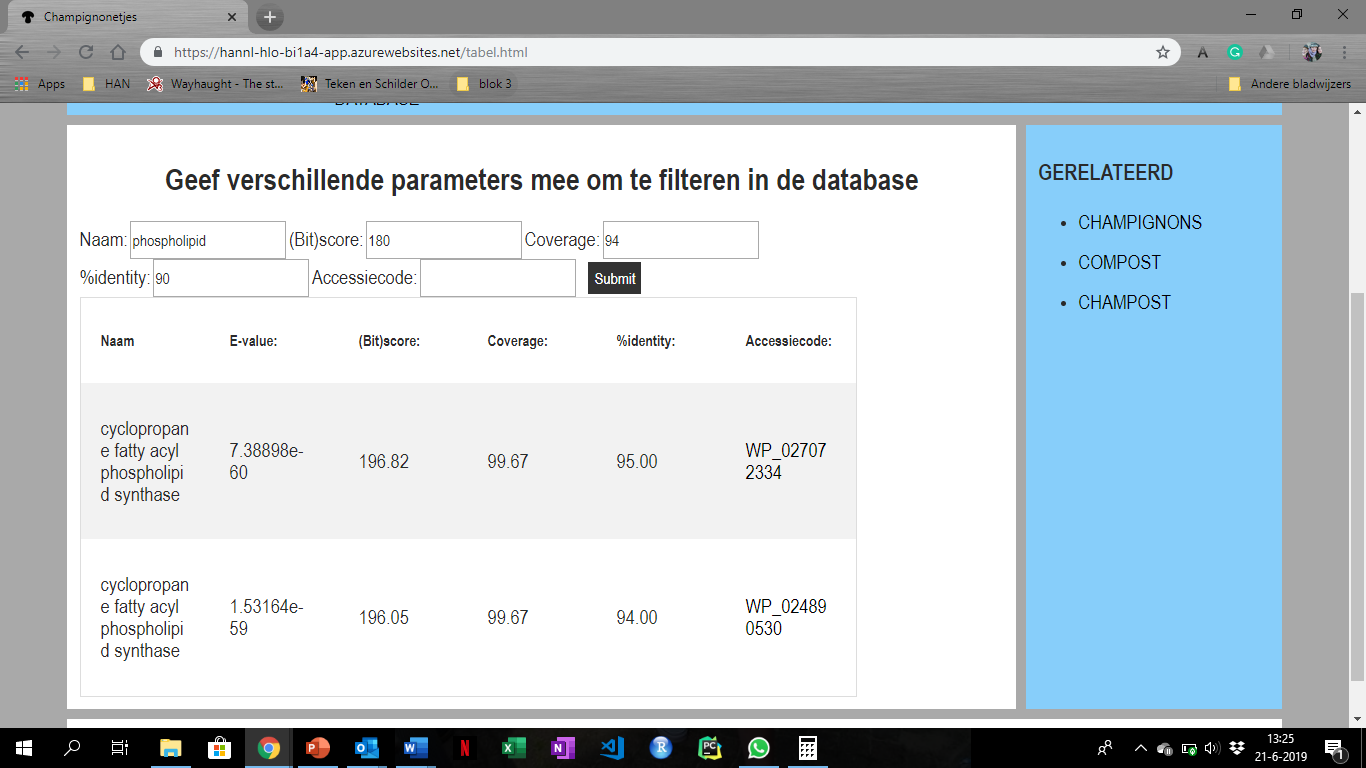
*Figuur 3 resultaten na filtering op accessiecode. Eigen werk van auteur.*

Bij het klikken op ‘Tabel met blast data’ krijgt u figuur 4 te zien. Deze tabel geeft de mogelijkheid om de informatie van de BLAST-resultaten raad te plegen. In de verschillende zoekvelden kan er op verschillende dingen gefilterd worden, zoals naam, bitscore, coverage, %identity en accessiecode. Wanneer vervolgens de gewenste gegevens zijn ingevuld, kunt u op ‘Submit’ klikken om de resultaten op te vragen, niet al de velden hoeven ingevuld te zijn.



*Figuur 4 pagina ‘Tabel met blast data’. Eigen werk van auteur.*

In figuur 5 is te zien wat er wordt weergegeven bij de zoekopdracht naam: ‘phospholipid’, (Bit)score: ‘180’, coverage: ‘94’, %identity: ‘90’ en accessiecode: ‘’. In de tabel zijn meerdere kolommen met blast informatie te zien; Naam, E-value, (Bit)score, Coverage, %identity en Accessiecode.

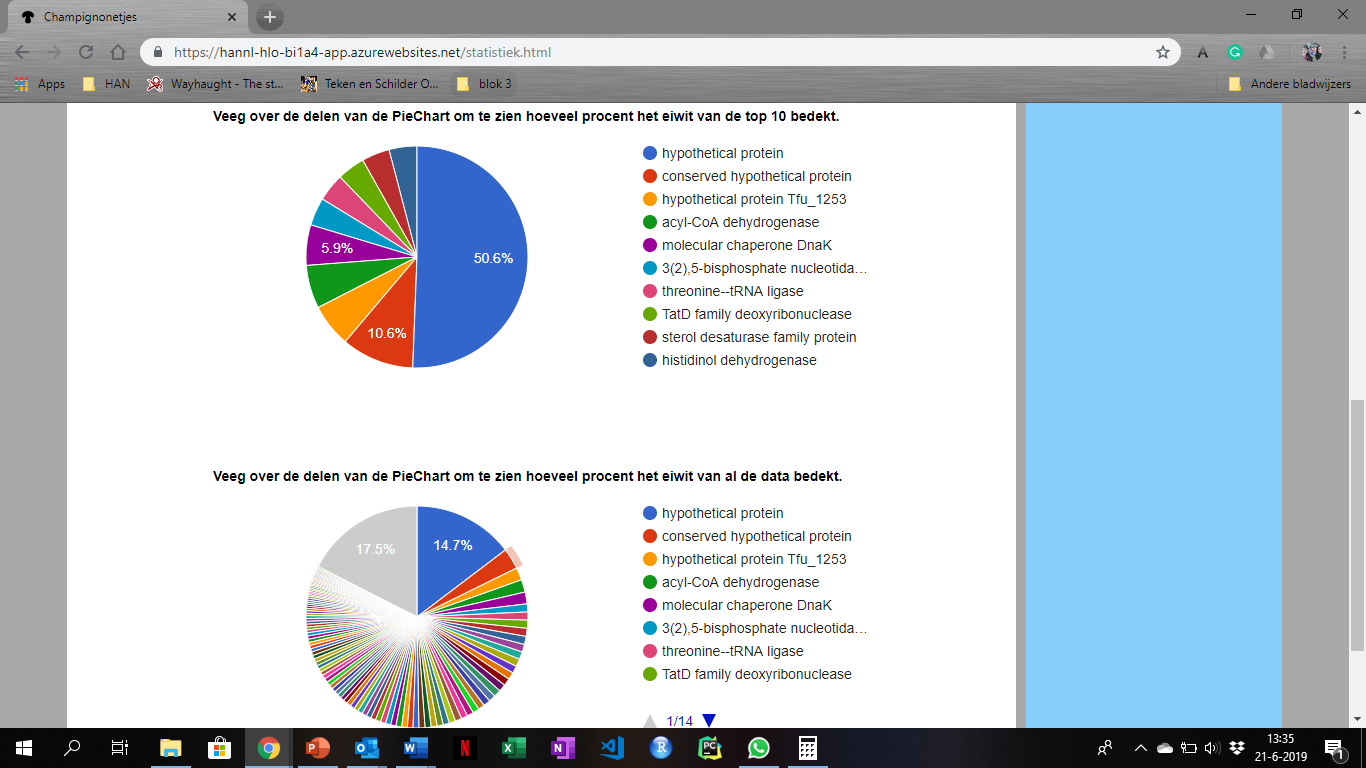


*Figuur 5 resultaten tabel van zoekopdracht: naam: ‘phospholipid’, (Bit)score: ‘180’, coverage: ‘94’, %identity: ‘90’ en accessiecode: ‘’. Eigen werk van auteur.*

Bij het klikken op ‘Statistieken’ krijgt u figuur 6 en figuur 7 te zien. Op deze pagina krijg je 3 PieCharts te zien de eerste laat de top 5 van de meest voorkomende eiwitten zien en hoeveel procent dit eiwit van de top 5 bedekt. De tweede laat de top 10 van de meest voorkomende eiwitten zien en hoeveel procent dit eiwit van de top 10 bedekt. De derde laat zien hoeveel procent het eiwit van al de data bedekt.

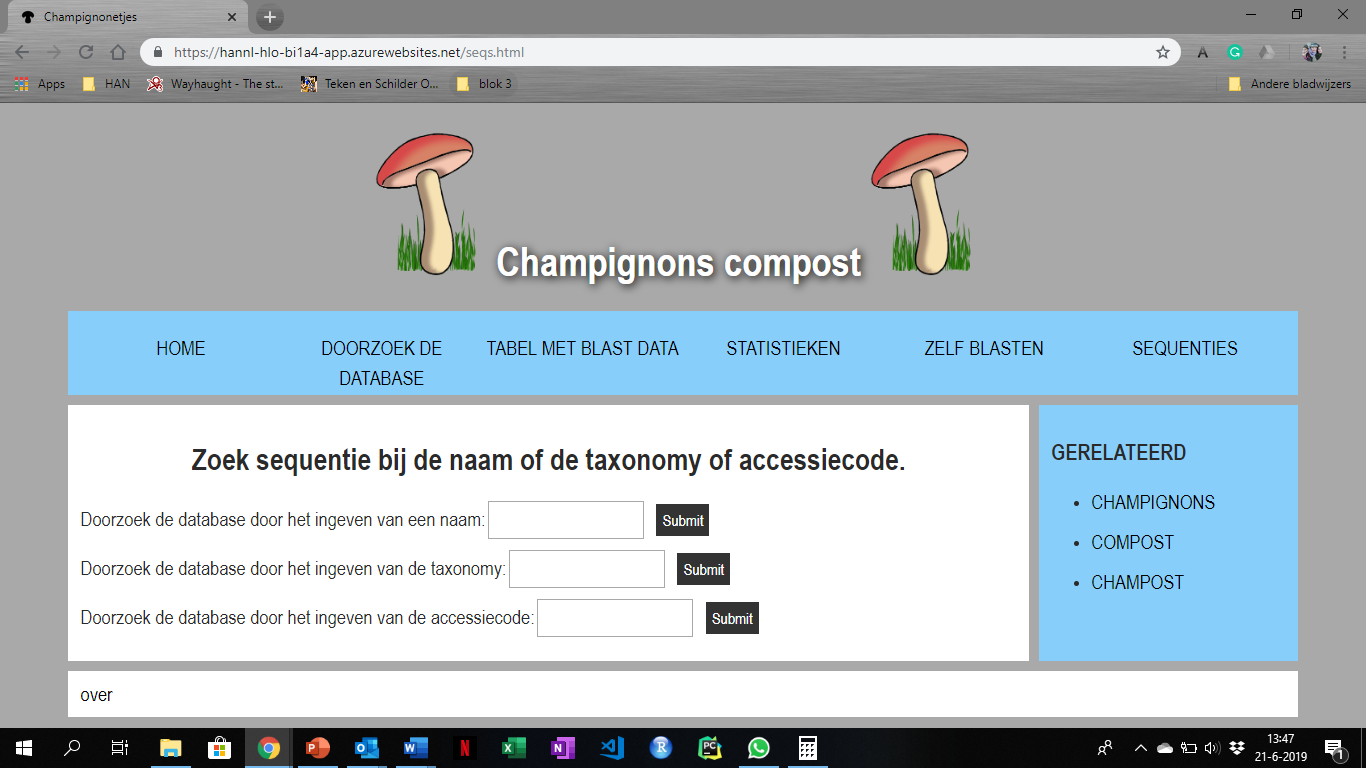


*Figuur 6 pagina 'Statistiek'. Eigen werk van auteur.*

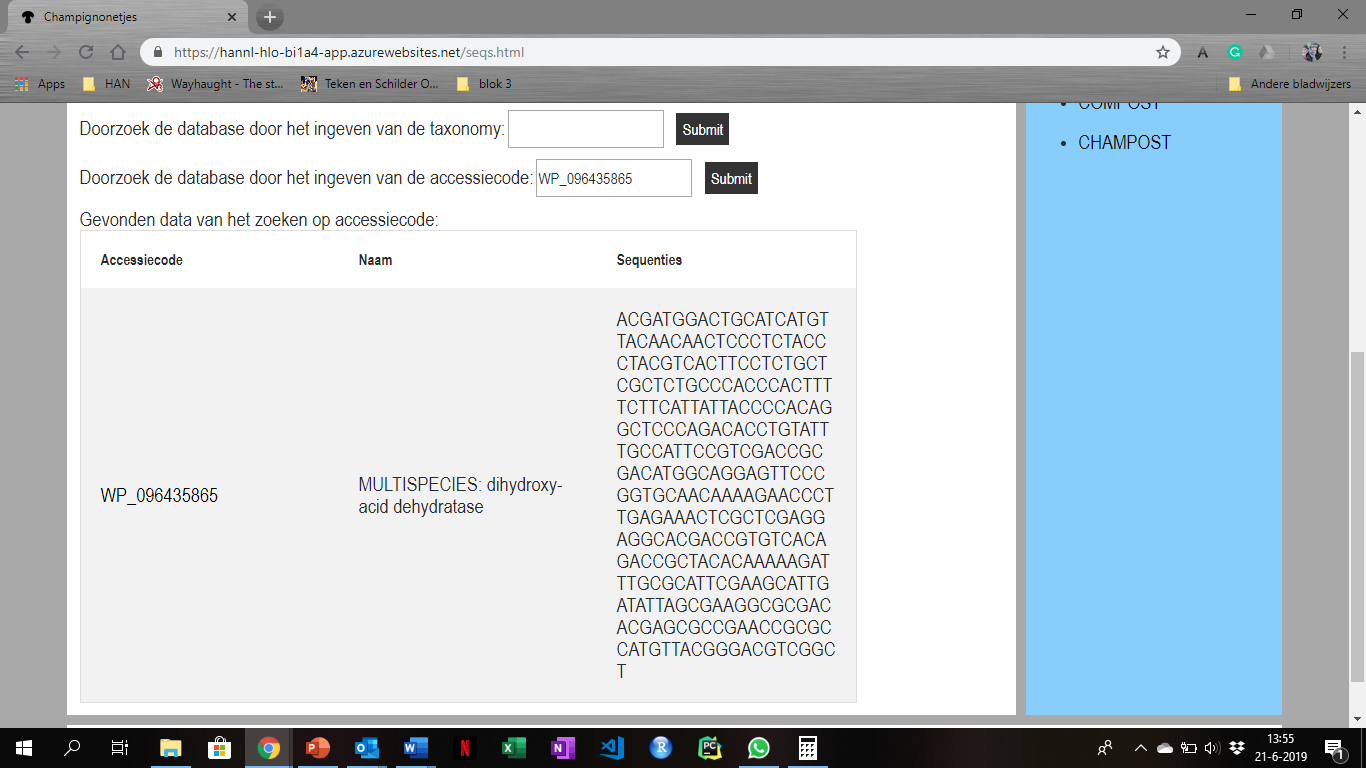


*Figuur 7 rest van de pagina 'Statistiek'. Eigen werk van auteur.*

Bij het klikken op ‘Sequenties’ zal de volgende pagina worden weergegeven als te zien in figuur 8. Op de ‘Sequenties’ pagina kan er gezocht worden op een naam, een keyword in de taxonomy of op een accessiecode. Om resultaten te krijgen moet er op de submit knop gedrukt worden. In figuur 9 is te zien wat je krijgt wanneer je op de accessie code filtert.



*Figuur 8 pagina Sequentie‘’. Eigen werk van auteur.*

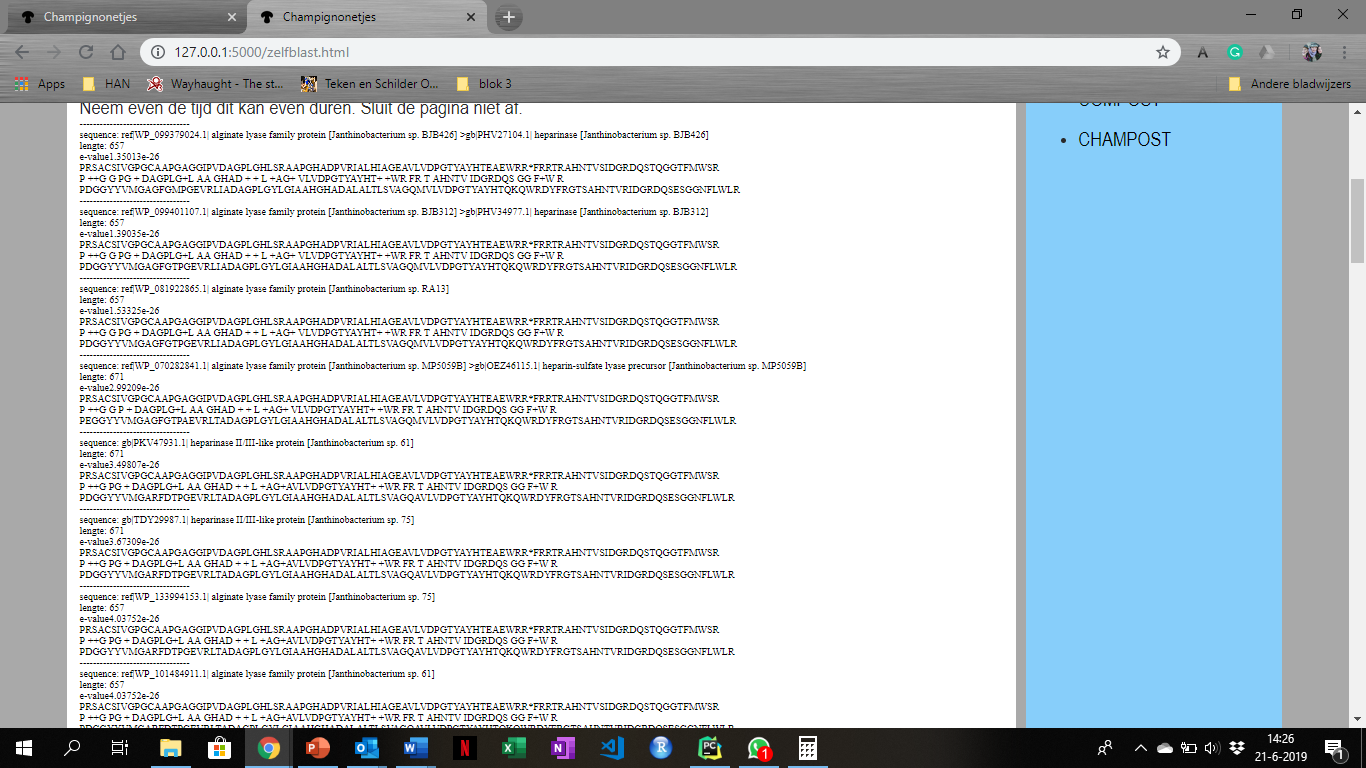


*Figuur 9 resultaten van filteren op de accessiecode. Eigen werk van auteur.*

De laatste pagina is raad te plegen door op ‘Zelf blasten’ te klikken. De pagina wordt dan weergegeven als te zien in figuur 10. Op deze pagina kan een DNA-sequentie ingevoerd worden. Deze zal dan vervolgens doormiddel van BLASTx geblast worden. De resultaten worden vervolgens weergegeven zoals in figuur 11. ‘Sequence’ staat voor de header van de gesequencede sequentie, ‘lengte’ voor de lengte van het alignement en ‘e-value’ voor e-value van het alignement. Hieronder staat ook de score van het alignement. Deze resultaten zullen niet worden toegevoegd aan de database, dit is om te voorkomen dat er vervuilde data in de database terecht komt.



*Figuur 10 pagina 'Zelf blasten'. Eigen werk van auteur.*



*Figuur 11 resultaten van blast.*

# BLAST Python script

Script: [BLAST.py](https://github.com/sannepost2001/Eind2/blob/master/BLAST.py)Gebruikte dataset: [Course4\_dataset\_v03.tsv](https://github.com/sannepost2001/Eind2/blob/master/Course4_dataset_v03.tsv)

Voor het automatiseren van het BLASTen van de test dataset is gekozen voor een Python script geschreven in Python 3.6 gebruikmakende van de Biopython module Bio, de sleep functie van de time module, de Blast class de NCBIWWW functie, en de NCBIXML functie uit Biopython, en ook vanuit Biopython Entrez en SearchIO.

Het script is opgebouwd uit verscheidene functies en 2 classes. Verder is het programma niet object geheel object oriented geschreven en maakt het gebruik van een vast pad omdat voor het vullen van de database de volgorde van handelen erg van belang is.

Het script is opgebouwd om de testdataset geautomatiseerd te BLASTen en is daardoor niet verder bedoeld om andere sequenties of bestanden te gebruiken, eventueel zou het script hier wel naar omgeschreven kunnen worden wanneer er delen van de main functie omgeschreven worden.

Verder zijn de 2 gemaakte classes:

**Database en BLASTer**Waarbij Database gebruikt wordt om alle informatie door te sluizen naar de Microsoft Azure webserver gehoste MySQL database.  
Database bevat de volgende functies:

**Functie taxonomy:**  
Haalt de taxonomy van een BLAST-resultaat op d.m.v. de Entrez Class, en retourneert een lijst die in volgorde de gevonden taxonomy van het resultaat bevat.

**Functie save\_header:**  
Insert de huidige header in de original tabel in de database

**Functie save\_sequence:**Insert de huidige sequentie, de huidige read, en de FASTQ-score in de sequence tabel in de database en verbindt deze met de correcte header uit de original database.

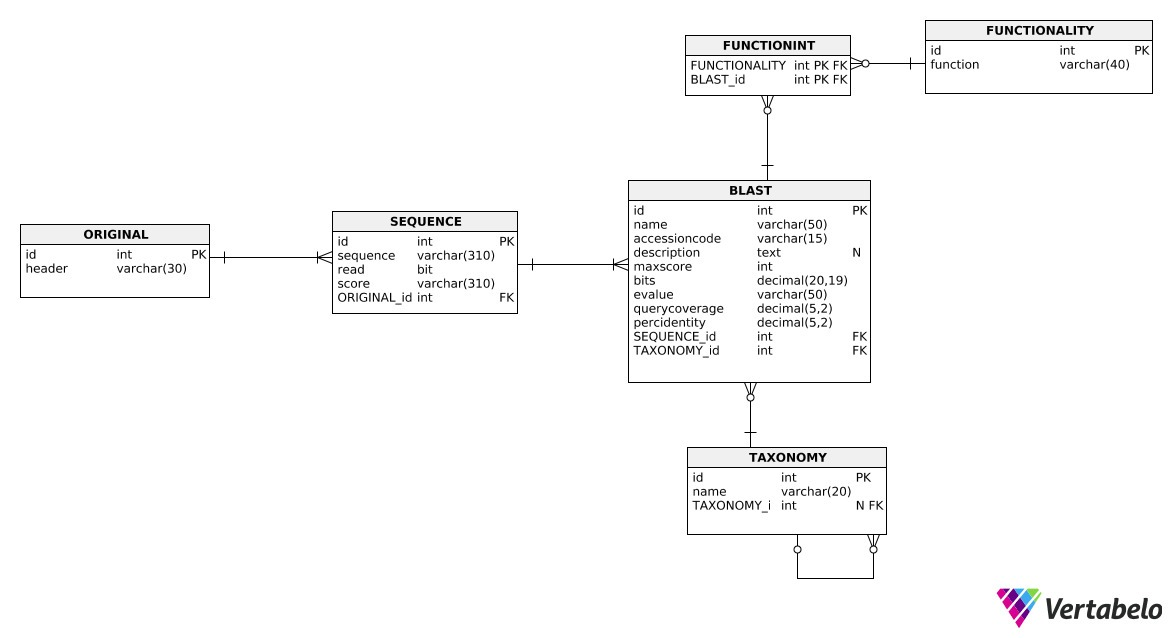
**Functie save\_blast:**  
Verzamelfunctie die alle voorgenoemde functies uitvoerd en deze data structureel aanroept om ge-insert te worden evenals benodigde functies uitvoert om delen van de informatie te verzamelen of om te zetten naar bruikbare vorm. En dit tot slot insert in de blast tabel in de database, en tegelijkertijd koppelt aan de correcte sequentie en de correcte taxonomy koppelt.

**Functie save\_functionality:**Ongebruikte incomplete functie, mocht hiervoor vraag zijn kan deze functie uitgebreid worden om de functionaliteit van een gevonden eiwit te achterhalen en op te slaan in de database.

En BLASTer wie de BLAST van de toegevoerde sequentie uitvoert.

De BLASTer class bevat 1 functie:  
blast, Deze functie controleert of er al een xml-bestand bestaat voor de huidige te BLASTen sequentie en wanneer deze niet bestaat voert deze een BLASTx uit en creëert een xml-bestand waarin de resultaten worden opgeslagen, de functie retourneert daarna een boolean False of True afhankelijk of het bestand al bestond of aangemaakt moest worden.

# Database



*Figuur 12 Technisch ERD MySQL database*

Voor het opslaan van de data is er een MySQL [Widenius et al, 2002] database gegenereerd. Deze database bestaat uit zes tabellen: original, sequence, blast, taxonomy, functionint en functionality. Figuur 12 laat het technisch Entitiy Relationship Diagram (ERD) van de gegenereerde database zien.

Voor het opslaan van de data is er een MySQL [Widenius et al, 2002] database gegenereerd. Deze database bestaat uit zes tabellen: original, sequence, blast, taxonomy, functionint en functionality. Figuur 12 laat het technisch Entitiy Relationship Diagram (ERD) van de gegenereerde database zien. Momenteel is er nog geen gebruik gemaakt van de functionint en functionality tabellen, hiervoor is gekozen omdat het integreren van de benodigde functies niet binnen de deadline verwerkt konden worden.

Alle keuzes omtrent relaties in de Database zijn gemaakt op de beide gemak in gebruik en de eigenschappen van alle individuele onderdelen van de verzamelde informatie.

**Original tabel:**

Verbonden met de sequence tabel d.m.v. een op veel relatie, deze verbindt een header met de 2 bijbehorende sequenties ofwel reads in de sequentie tabel, dmv een auto-incrementing column genaamd id. De tabel bevat 2 columns, primary-id: id een integer die auto-incrementing is en column: header een varchar van maximaal 30 karakters.

**Sequence tabel:**

Verbonden met de blast tabel d.m.v. een een op veel relatie, deze verbindt een de resultaten in de blast tabel aan de bijbehorende sequentie door een auto-incrementing column genaamd id.

De tabel zelf bevat 5 columns, primary-id: id een integer die auto-incrementing is, en columns: sequence een varchar van maximaal 310 karakters waarin de nucleotide sequentie uit de testdata instaat, read een bit om aan te geven of de sequentie read 1 of 2 is, score een varchar van maximaal 310 karakters waarin de FASTQ-score is opgeslagen, en foreign-key ORIGINAL\_id die de header aan de sequentie koppelt.

**Blast tabel:**

Verbonden met de taxonomy en functionint tabel, taxonomy verbonden via een Veel of null op een relatie dit verbindt een gevonden taxonomy aan het bijbehorende BLAST-resultaat, en function in verbonden via een een op veel of null relatie dit verbindt eventueel gevonden functionaliteit(en) aan een BLAST-resultaat.

De tabel zelf bevat 11 columns, primary-id: id een integer die auto-incrementing is, en columns: Name een varchar van maximaal 50 karakters waarin de naam van het gevonden eiwit van het BLAST-resultaat is opgeslagen.

Accessioncode een varchar van maximaal 15 karakters waarin de accessiecode van het BLAST-resultaat is opgeslagen.

Description nen text waarin de volledige beschrijving van het BLAST-resultaat is opgeslagen.

Maxscore een int waarin maxscore van het BLAST-resultaat is opgeslagen.

Bits een decimal verdeling (20,19) waarin de behaalde bit score is opgeslagen.

Evalue een varchar van 50 waarin de e-value van het BLAST-resultaat is opgeslagen.

Querycoverage een decimal verdeling (5,2) waarin de query coverage van het BLAST-resultaat is opgeslagen.

Percentidentity een decimal verdeling (5,2) waarin de percentage identity van het BLAST-resultaat is opgeslagen.

En foreign keys: SEQUENCE\_id en TAXONOMY\_id die respectiefelijk de sequence- en taxonomy tabel aan een BLAST-resultaat linken.

**Taxonomy tabel:**

Verbonden met de taxonomy tabel d.m.v. null of een op null of veel relatie dit verbindt de taxonomy tabel op zichzelf zodat deze data non-redundant opgeslagen kan worden.

De tabel zelf bevat 3 columns, primary-id: id een integer die auto-incrementing is, en column: name een varchar van maximaal 20 karakters waarin een taxonomische naam is opgeslagen, en foreign key TAXONOMY\_id die de taxonomy tabel opzichzelf linkt zodat er een ketting aan namen ontstaat.

Van de volgende 2 tabellen wordt huidig nog geen gebruik gemaakt, ze bestaan enkel als mogelijkheid tot uitbreiding van de functionaliteit van de Database.

**Functionality tabel:**

Verbonden met de functionint intertabel d.m.v. een een op veel of null relatie, dit kan worden gebruikt om 1 of meer functionalities aan 1 of meer BLAST-resultaten te koppelen.

De tabel zelf bevat 2 columns, primary-id: id een integer die auto-incrementing is, en columns: function een varchar van maximaal 40 karakters waarin een functie van eiwitten opgeslagen kan worden.

**Functionint intertabel:**

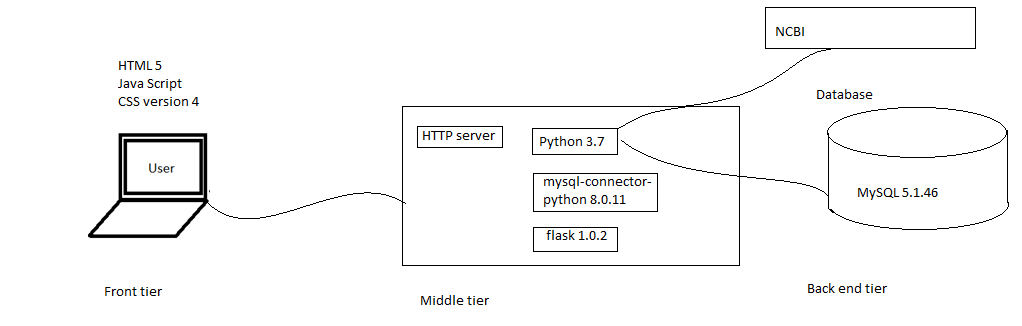
Tabel die 2 Foreign keys accepteert en deze samen bevat als een unieke primary key, deze kan gebruikt worden om de relaties tussen de blast tabel en de functionality tabel weer te geven en op te slaan.

# Software architectuur

De gebruiker wordt connected aan de website (<https://hannl-hlo-bi1a4-app.azurewebsites.net/>), door middel van een HyperText Transfer Protocol Secure (HTTPS) server deze wordt gehost op Microsoft Azure [Copeland, et al. 2015]. Hieruit kunnen meerdere functies in de website aangeroepen worden.

Vanuit een Python 3.7 script wordt de applicatie (app.py) aangeroepen. Hierbij wordt gebruik gemaakt van Flask 1.0.2 [Miguel Grinberg, 2014]. Door het gebruik van Flask kan er een HTML-pagina geretourneerd worden en de python functionaliteit wordt behouden. Het Python script zal een van de meerdere HTML-pagina’s retourneren, dit ligt aan wat er via de website is opgevraagd. Een van de HTML-codes bevat ook een deel JavaScript, dit wordt gebruikt voor het genereren van de PieCharts. Ook hebben al de HTML-pagina’s een verwijzingen naar een CSS-file. De code van app.py is te vinden via <https://github.com/sannepost2001/Eind2>.

In de Python code wordt gebruik gemaakt van de mysql-connector-python 8.0.11 [Widenius et al, 2002] om een connectie te maken met de MySQL 5.1.46 database met daarin al de gegevens van het blasten. Ook is er een connectie tussenn de National Center for Biotechnology Information (NCBI) database [O’Leary et al. 2016] en de Python code, met behulp van de BioPython 1.37 module [Cock PA et al., 200]. Om op de website te kunnen blasten is er een functie van BioPython gebruikt.



*Figuur 13 Software architectuur van de applicatie.*

# Testscripts en resultaten

De website is toegankelijk via deze link, de site bestaat uit 6 pagina’s met elk hun eigen functie, op elke pagina vindt u een header met links naar de overige pagina’s en een zij-balk met links naar wat extra achtergrondinformatie.

Alle functies zijn getest op 11 juni 2019.

*Tabel 1 Werking van ‘Doorzoek de database’*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Stap** | **Instructie** | **Verwacht resultaat** | **Resultaat** | **Opmerking** |
| 1 | Open de applicatie | De applicatie wordt geopend in een gekozen browser | Werkt | Gebruikte browser: Google Chrome Versie 75.0.3770.100 (Officiële build) (64-bits) |
| 2 | Klik op ‘Doorzoek de database’ | De applicatie geeft het pagina van ‘Doorzoek de database’ weer | Werkt |  |
| 3 | Klik op ‘submit met een ingevulde zoekterm | Geeft de data die bij deze zoekterm hoort weer | Werkt |  |
| 4 | Klik op de bovenste rij en sorteer op deze manier op de verschillende kolommen | Er wordt gesorteerd op een kolom. | Werkt |  |

*Tabel 2 Werking van '* *Tabel met blast data’*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Stap** | **Instructie** | **Verwacht resultaat** | **Resultaat** | **Opmerking** |
| 1 | Open de applicatie | De applicatie wordt geopend in een gekozen browser | Werkt | Gebruikte browser: Google Chrome Versie 75.0.3770.100 (Officiële build) (64-bits) |
| 2 | Klik op ‘Tabel met blast data’ | De applicatie geeft het pagina van ‘Tabel met blast data’ weer | Werkt |  |
| 3 | Klik op ‘submit’ na het invullen van de zoekterm in de zoekregel | Geeft de data die bij deze zoekterm hoort weer | Werkt | Enige nadeel is dat er niet te filteren is op de e-value |
| 4 | Klik op de bovenste rij en sorteer op deze manier op de verschillende kolommen | Er wordt gesorteerd op een kolom. | Werkt |  |

*Tabel 3 Werking van 'Statistiek'*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Stap** | **Instructie** | **Verwacht resultaat** | **Resultaat** | **Opmerking** |
| 1 | Open de applicatie | De applicatie wordt geopend in een gekozen browser | Werkt | Gebruikte browser: Google Chrome Versie 75.0.3770.100 (Officiële build) (64-bits) |
| 2 | Klik op ‘Statistiek’ | De applicatie geeft het pagina van ‘Statistiek’ weer | Werkt |  |

*Tabel 4 Werking van 'Zelf blasten'*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Stap** | **Instructie** | **Verwacht resultaat** | **Resultaat** | **Opmerking** |
| 1 | Open de applicatie | De applicatie wordt geopend in een gekozen browser | Werkt | Gebruikte browser: Google Chrome Versie 75.0.3770.100 (Officiële build) (64-bits) |
| 2 | Klik op ‘Statistiek’ | De applicatie geeft het pagina van ‘Statistiek’ weer | Werkt |  |
| 3 | Geef een DNA sequentie in klip op ‘submit’ | De sequentie wordt geblast en de resultaten worden getoond | Werkt niet op Azure maar wel als je de applicatie lokaal opent |  |

*Tabel 5 Werking van ‘Sequenties’*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Stap** | **Instructie** | **Verwacht resultaat** | **Resultaat** | **Opmerking** |
| 1 | Open de applicatie | De applicatie wordt geopend in een gekozen browser | Werkt | Gebruikte browser: Google Chrome Versie Versie 75.0.3770.100 (Officiële build) (64-bits) |
| 2 | Klik op ‘Sequenties’ | De applicatie geeft het pagina van ‘Doorzoek de database’ weer | Werkt |  |
| 3 | Klik op ‘submit’ na het invullen van de zoekterm in de zoekregel | Geeft de data die bij deze zoekterm hoort weer | Werkt |  |
| 4 | Klik op de bovenste rij en sorteer op deze manier op de verschillende kolommen | Er wordt gesorteerd op een kolom. | Werkt |  |

# Literatuurlijst

* Cock PA, Antao T, Chang JT, Chapman BA, Cox CJ, Dalke A, Friedberg I, Hamelryck T, Kauff F, Wilczynski B and de Hoon MJL (2009) Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics, 25, 1422-1423
* Copeland M, Soh J, Puca A, Manning M, Gollob D (2015). *Microsoft Azure: Planning, Deploying, and Managing Your Data Center in the Cloud*. Berkely, CA, USA: Apress.
* Miguel Grinberg (2014), Flask Web Development: Developing Web Applications with Python, O'Reilly Media, Inc. Geraadpleegd op 2 juni 2019, van https://doc.lagout.org/programmation/python/Flask%20Web%20Development\_%20Developing%20Web%20Applications%20with%20Python%20%5BGrinberg%202014-05-18%5D.pdf
* O'Leary NA, Wright MW, Brister JR, Ciufo S, Haddad D, McVeigh R, Rajput B, Robbertse B, Smith-White B, Ako-Adjei D, Astashyn A, Badretdin A, Bao Y, Blinkova O, Brover V, Chetvernin V, Choi J, Cox E, Ermolaeva O, Farrell CM, Goldfarb T, Gupta T, Haft D, Hatcher E, Hlavina W, Joardar VS, Kodali VK, Li W, Maglott D, Masterson P, McGarvey KM, Murphy MR, O'Neill K, Pujar S, Rangwala SH, Rausch D, Riddick LD, Schoch C, Shkeda A, Storz SS, Sun H, Thibaud-Nissen F, Tolstoy I, Tully RE, Vatsan AR, Wallin C, Webb D, Wu W, Landrum MJ, Kimchi A, Tatusova T, DiCuccio M, Kitts P, Murphy TD, Pruitt KD. Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation. *Nucleic Acids Res*. 2016 Jan 4;44(D1): D733-45 PubMed
* Widenius M, Axmark D, DuBois P (2002). *Mysql Reference Manual.* Sebastopol, CA, USA: O’Reilly & Associates, Inc.