

Technische Documentatie

KARAKTERISEREN VAN COMPOST

EVANDER VAN WOLFSWINKEL, RICK SCHOENMAKER EN
VALERIE VERHALLE

Inhoud

Inleiding	2
Werking applicatie.....	3
BLAST PROGRAMMA STANDALONE ('BLASTEN.py')	3
FLASK PROGRAMMA Webapplicatie ('app.py').....	4
Database.....	5
Navigatie.....	6
Software Architectuur	9
Gebruikte Modules:.....	10
Testscripts.....	11
Verwijzingen	13

Inleiding

Om de effecten van de samenstelling van de microflora in compost op de opbrengst van champignons te kunnen bestuderen moet de kwalitatieve en kwantitatieve samenstelling van de microflora in de verschillende fases van het composteringsproces betrouwbaar kunnen worden bepaald. Bij het gebruik van NGS data is daarvoor een juiste annotatie van de reads van groot belang. Voor het ontwikkelen en testen van een betrouwbare annotatie methode is een specifieke dataset gegenereerd. Op verschillende momenten tijdens een commercieel composteringsproces is een compost sample genomen en daaruit is het DNA geïsoleerd. Het DNA is vervolgens gesequenced door het bedrijf BaseClear met behulp van de Illumina MiSeq technologie als paired-end reads (per sample ongeveer 2500000 reads). Voor het opzetten en testen van een annotatie-pipeline is een kleine test set reads genomen (~100 sequenties). Deze reads zijn middels de gemaakte webapplicatie en een BLAST search van een annotatie voorzien.

Deze technische documentatie is geschreven voor technici die betrokken zijn bij het eindproduct of project, om zo de applicatie beter te begrijpen en er ook mee leren om te gaan. Het is niet alleen van belang om met de applicatie om te kunnen gaan maar ook om te begrijpen hoe de applicatie is opgebouwd, wat de denkwijze achter bepaalde stappen is en hoe de applicatie technisch in elkaar steekt. Verder heeft deze technische documentatie nut voor biologen die meer informatie zouden willen over het biologische aspect van het onderzoek, zoals hoe BLAST is verwerkt in dit project.

Biologen zullen voornamelijk gebruik maken van de applicatie om sequenties te blasten met de door de projectgroep vastgestelde parameters. Mochten deze sequenties interessante resultaten geven, dan zouden de resultaten achteraf aan de database toegevoegd kunnen worden. Deze database kan men inzien via de website van de applicatie.

Werking applicatie

BLAST PROGRAMMA STANDALONE ('BLASTEN.py')

In figuur 1 zijn de gebruikte modules weergegeven, waaronder de vanuit Biopython library 'Blast' geïmporteerde modules NCBIWWW (Peter J. A. Cock, Module NCBIWWW, 2018) en NCBIXML (Peter J. A. Cock, Module Bio.Blast.NCBIXML, 2018). De NCBIWWW module maakt een connectie met

```
from Bio.Blast import NCBIWWW
from Bio.Blast import NCBIXML
from Bio import Entrez
import mysql.connector
```

Figuur 1 Gebruikte modules in 'BLASTEN.py'.
NCBIWWW, NCBIXML, Entrez en een mysql.connector.

de NCBI BLAST server over het internet waarna de functie 'qblast' wordt gebruikt om de parameters in te stellen, zoals in figuur 2 te zien is. Na het blasten wordt er een xml file gegenereerd met hierin de resultaten van de blast van alle gevonden hits. De NCBIXML module kan het verkregen xml file parsen om er zo de benodigde informatie uit te halen.

```
result_handle = NCBIWWW.qblast('blastx', 'nr', record, gapcosts='10 1',
                                expect='0.0001', filter=True,
                                matrix_name='BLOSUM62',
                                word_size='6', alignments='10',
                                descriptions='10', perc_ident='25')
```

Figuur 2 De module NCBIWWW met de functie 'qblast' waarmee parameters ingesteld kunnen worden voor het BLASTEN.

```
handle = Entrez.efetch(db='protein',
                        id=alignment.accession,
                        retmode='xml', rettype='gb')
record = Entrez.parse(handle, 'genbank')
```

Figuur 3 Door het meegeven van een accessiecode kan de module Entrez de taxonomie van een hit ophalen

Verder is er vanuit de BioPython library de module Entrez (Peter J. A. Cock, Package Entrez, 2018) geïmporteerd. Deze module maakt het mogelijk om op een makkelijke manier de taxonomie van een bepaalde hit op te vragen. Entrez doet dit d.m.v. een accessiecode, zoals in figuur 3 weergegeven.

Als laatste is er een mysql.connector (Oracle Corporation and/or its affiliates , 2018) gebruikt om te kunnen connecteren aan de MYSQL database. Deze module is ook gebruikt om gegevens uit het xml file naar de SQL database te itereren. In figuur 4 is een voorbeeldje gegeven van hoe deze module gebruikt is.

```
for index in range(len(header)):
    # index = index.replace('>', '').replace('\n', '')
    print(header[index])
    print(seq[index])
    print(ascii[index])

    sqlheader = ("insert into Sequentie(header, sequentie, ascii_score) values ('{}', '{}', '{}')".format(
        header[index], seq[index], ascii[index]))
    cursor.execute(sqlheader)

conn.commit()
conn.close()
cursor.close()
```

Figuur 4 Voorbeeld van hoe de module mysql.connector is gebruikt. 'Header' is een lijst met alle headers, 'seq' is een lijst met alle sequenties en 'ascii' is een lijst met alle ascii-scores.

FLASK PROGRAMMA Webapplicatie ('app.py')

Om van het Flask framework gebruik te maken is de module van de Flask library geïmporteerd samen met de Jinja `render_template` module, `request` en `redirect`. De Jinja `render_template` functie maakt het mogelijk om binnen een HTML pagina python code te itereren via FLASK (Grinberg, 2014). Op deze

```
from flask import Flask, render_template, request, redirect
from Bio.Blast import NCBIWWW
from Bio.Blast import NCBIXML
from Bio import Entrez
import mysql.connector
import matplotlib.pyplot as plt
from io import BytesIO
import base64
```

Figuur 5 Gebruikte modules: Flask, render_template, request, redirect, NCBIWWW, NCBIXML, Entrez, mysql.connector, matplotlib.pyplot, BytesIO en base64

manier kunnen de resultaten van een blast doorgegeven worden naar de desbetreffende HTML pagina, een voorbeeldje hiervan is te zien in figuur 6. Met de Flask import request wordt een opgegeven fasta file uit HTML form opgeslagen in een variabele. Import redirect kan python code laten 'terugkeren' naar een specifieke `@app.route`. De modules NCBIWWW, NCBIXML, Entrez en de mysql.connector worden op dezelfde manier gebruikt als in het 'BLASTEN.py' bestand.

De gebruiker kan data uit de database opvragen. Om deze data te visualiseren is er gebruik gemaakt van de module matplotlib (John Hunter, 2018). Om het plotje netjes weer te geven op de site is deze eerst in FLASK omgezet naar een png formaat met behulp van de module BytesIO en verder gecodeerd via base64, te zien in figuur 8.

```
return render_template('databaseresult.html', plotimage= resultplot.decode('utf8'))
```

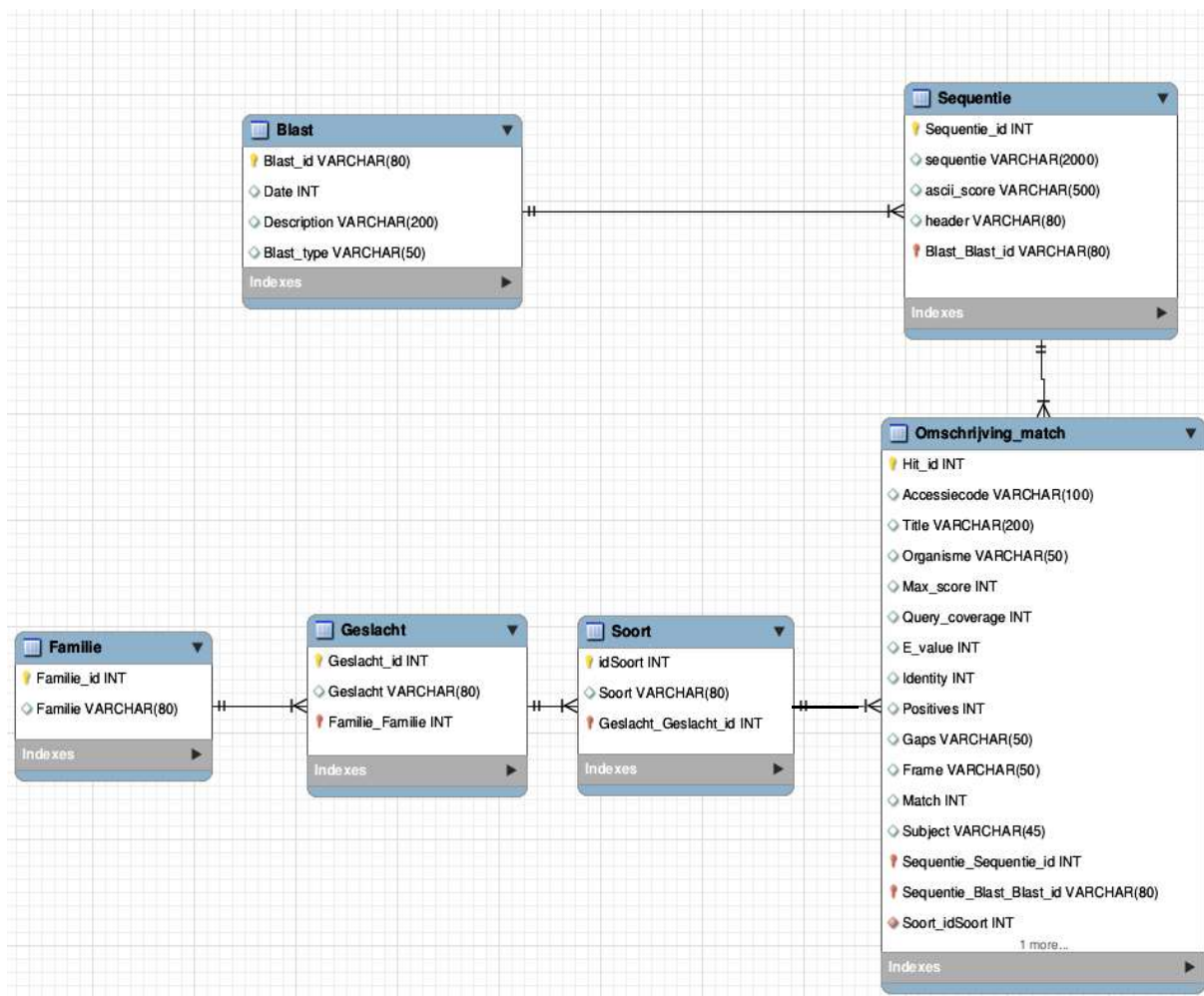
Figuur 6 De resultaten van een opgegeven zoekwoord worden hier via `render_template` doorgegeven naar de pagina: `databaseresult.html`

```
figfile = BytesIO()
plt.savefig(figfile, format='png')
figfile.seek(0)
figdata_png = base64.b64encode(figfile.getvalue())
resultplot = figdata_png
```

Figuur 7 Een matplotlib result (figfile) wordt hier omgezet naar een png format om vervolgens met de module `base64` weergegeven kan worden op de html pagina.

Database

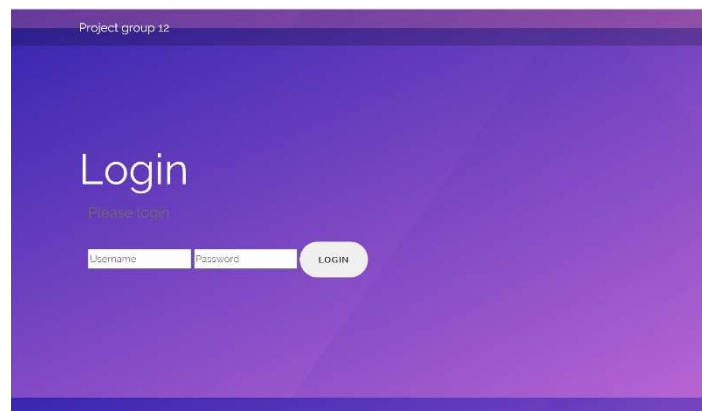
Zoals te zien is in Figuur 8 bestaat de database uit 6 tabellen. Tussen de tabellen Blast en Sequentie zit een 1 op veel relatie want 1 blast kan meerdere sequenties tegelijk van annotatie voorzien. Elke sequentie kan door middel van zijn blast_id gevonden worden. De één op veel relatie tussen de tabel Sequentie en de tabel Omschrijving_match is noodzakelijk want voor elke sequentie worden maximaal 10 hits opgeslagen. Belangrijke informatie over de hits worden opgeslagen in de tabel Omschrijving_match, het gaat hier om rijen zoals het percentage identity, de E-value van een alignment, accessiescore en positives. Deze gegevens zijn gelinkt aan de sequentie met een unieke sequentie_id. Hiermee is altijd terug te vinden welke gegevens bij welke sequentie horen. De database is door deze connecties flexibel uit te breiden. Verder is de database voorzien van een hiërarchie. De hiërarchie is terug te zien in de tabellen, Familie, Geslacht en Soort. 1 familie bevat meerdere geslachten en 1 geslacht bevat op zijn beurt weer verschillende soorten. Door een hiërarchie in de database te verwerken kan de taxonomie van een hit makkelijk achterhaald worden.



Figuur 8 ERD, SQL database

Navigatie

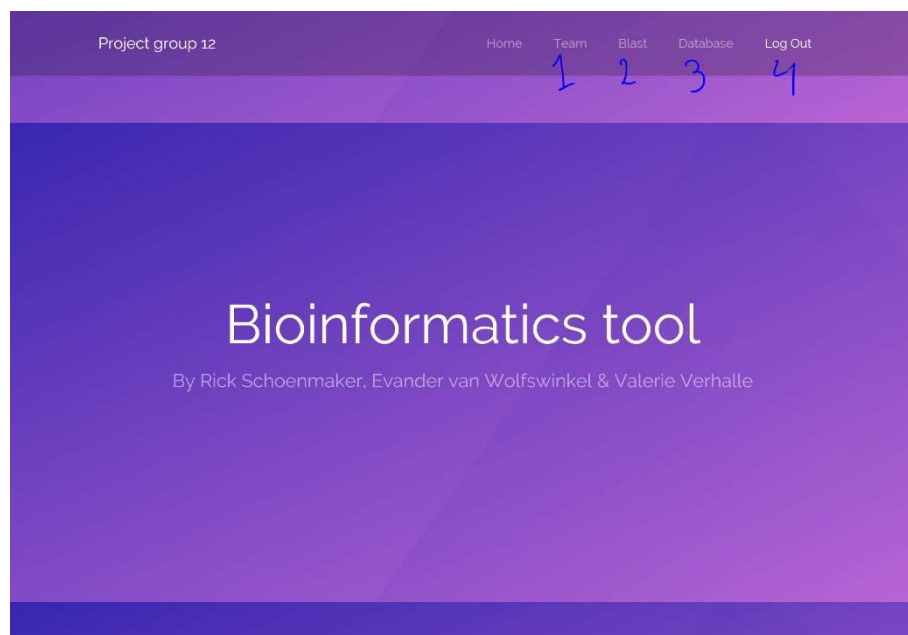
Bij het opstarten van de applicatie zal het volgende scherm weergegeven worden:



Figuur 9 Inlogscherf

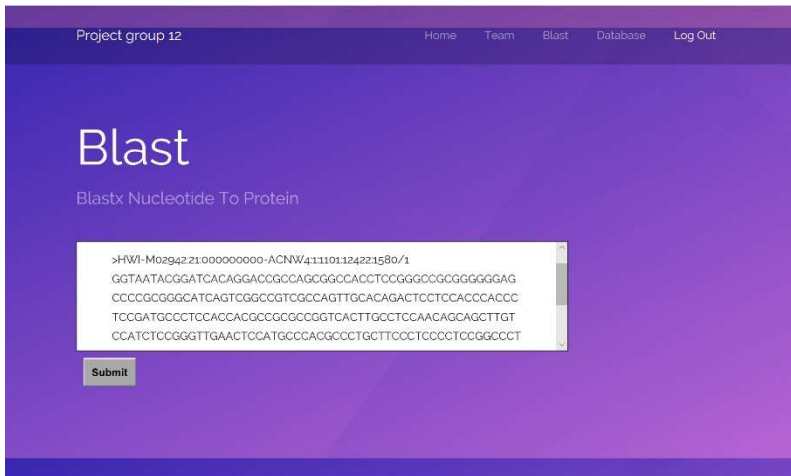
Eenmaal ingelogd kom je op de homepage (figuur 10) vanuit hier kun je alle kanten op (gebruikersnaam en wachtwoord staan in de README).

1. Team: Hier is informatie te vinden over de mensen die de applicatie gebouwd hebben.
2. Blast: Hier kun je sequenties blasten met de door de projectgroep vastgestelde parameters (welke dit zijn staan aangegeven op de Blast-pagina).
3. Database: Op deze pagina kun je de SQL database doorzoeken door een zoekwoord op te geven.
4. Log Out: log jezelf weer uit van de site.



Figuur 10 homepage

In figuur 11 is de Blast-pagina weergegeven met een voorbeeldsequentie als input. Als output wordt er een result-pagina weergegeven (figuur 12) en als de gebruiker de resultaten naar de database wil updaten, volgt er een nieuwe database is up to date pagina.



Project group 12 Home Team Blast Database Log Out

Blast

Blastx Nucleotide To Protein

```
>HWI-M02942.21.000000000-ACNW4.11101124221580/1
GGTAATACGGATCACAGGACCGCCAGCGGCCACCTCCGGGCGCGGGGGGAG
CCCCGCGGGCATCAGTCGGCCGTCGCCAGTTGCACAGACTCCTCCACCCACCC
TCCGATGCCCTCCACCAGCCGCGCGGTCACTTGCTCGAACAGCAGCTTGT
CCATCTCCGGTTGAACTCATGCCCAAGCCCTGCTTCCCTCCCTCCGCGCCT
```

Submit

Figuur 11 Blast-pagina met voorbeeldsequentie

Parameters

Blast-type: Blastx

Database: Non-redundant protein sequences database

Scorematrix: BLOSUM62

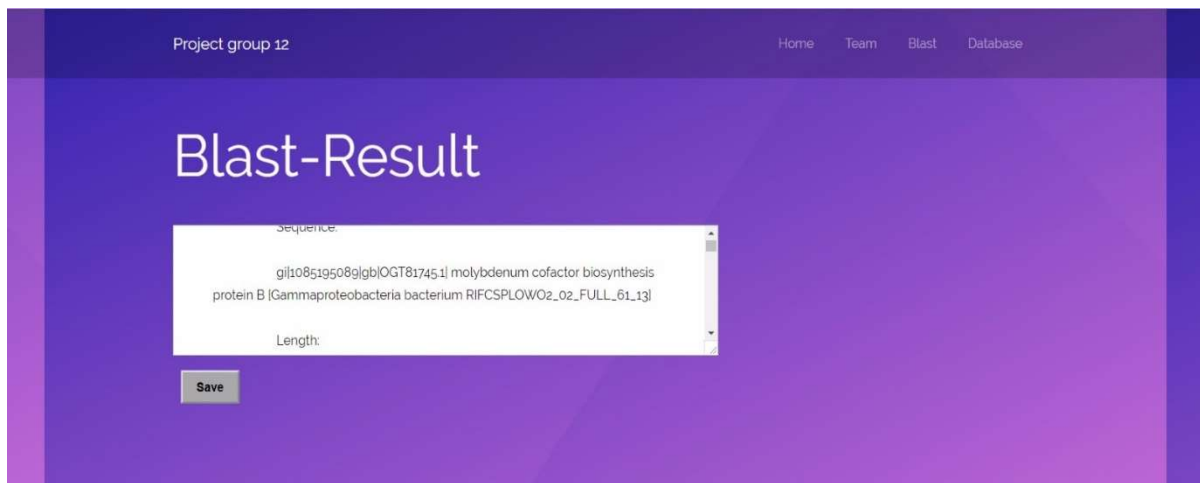
Gapcost: Opening a gap:-10, for every position in gap:-1

Exclusions: Eukaryotes excluded

Word size: 6

Low complexity filter: Active

Figuur 12 Gebruikte parameters.



Project group 12 Home Team Blast Database

Blast-Result

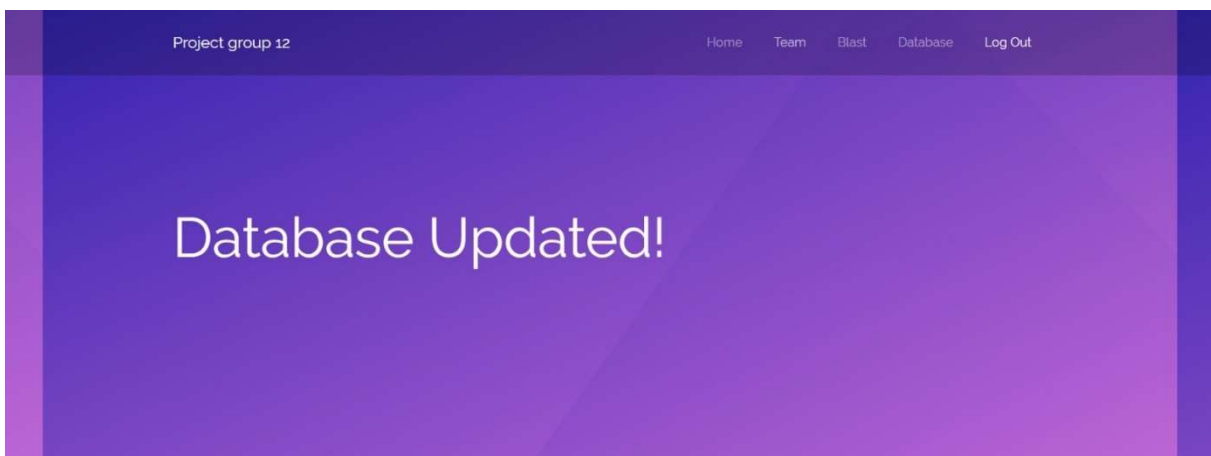
Sequence:

```
gij108519508gjb|OGT81745.1| molybdenum cofactor biosynthesis
protein B |Gammaproteobacteria bacterium RIFCSPLOW02_02_FULL_61_13|
```

Length:

Save

Figuur 13 Blast-result page



Project group 12 Home Team Blast Database Log Out

Database Updated!

Figuur 14 Database is updated result-page

In de volgende afbeelding is voorgedaan hoe je een zoekwoord op moet geven om door de database te kunnen zoeken. Het programma gaat op zoek naar organismen met dit zoekwoord in hun omschrijving en geeft dan in een grafiek weer hoe vaak dit zoekwoord bij elk organisme voorkomt (figuur 16).

Project group 12

[Home](#)[Team](#)[Blast](#)[Database](#)[Log Out](#)

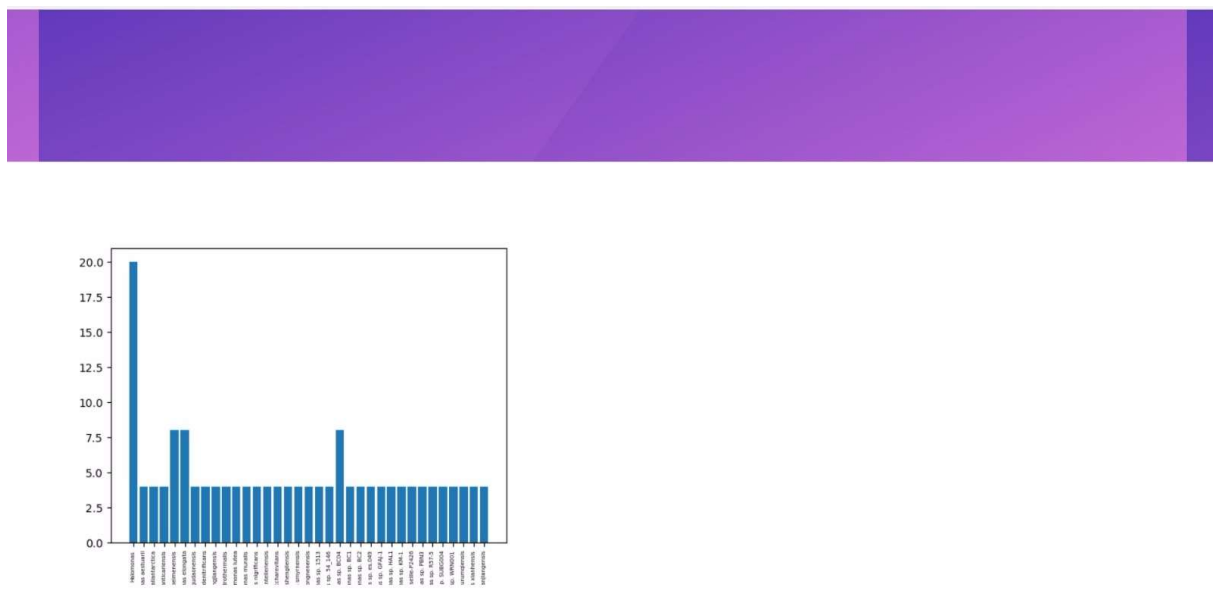
Database

Identified Prokaryotes in Mushroom Compost

Search organism count for Description

Search

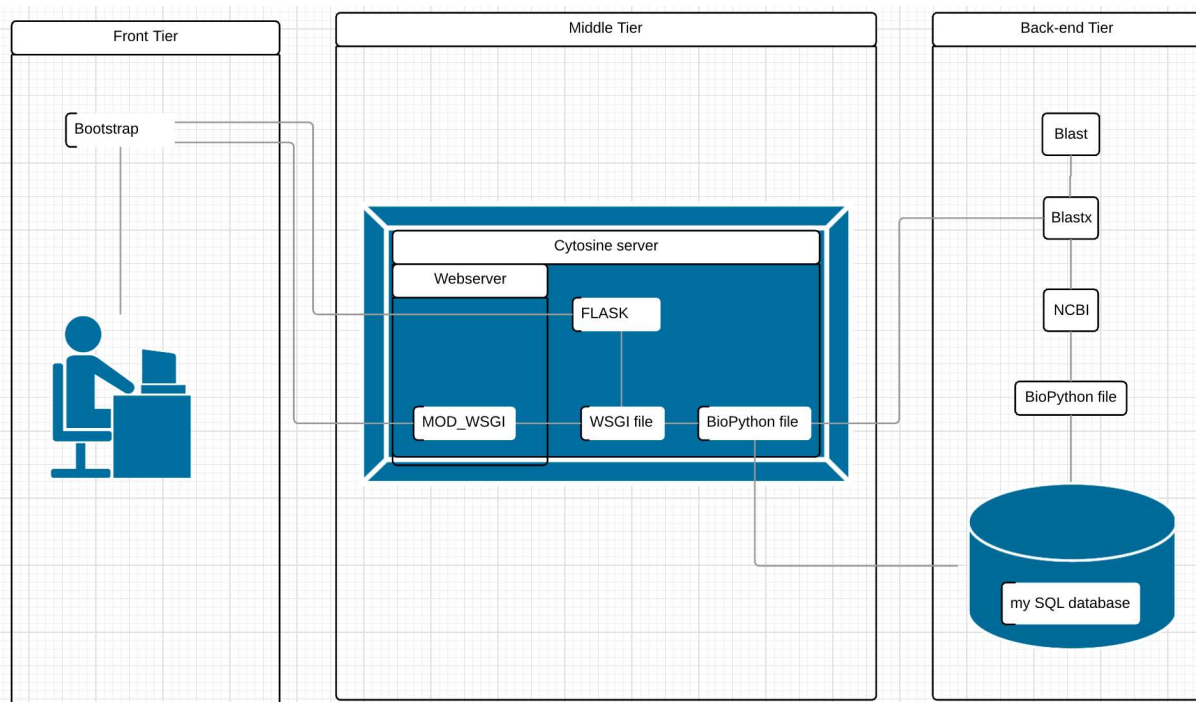
Figuur 15 Database pagina, geef een zoekwoord op



Figuur 16 Database result page

Software Architectuur

In Figuur 17 is de applicatie architectuur te zien, met in de Front Tier de gebruiker van de applicatie die de site aanroept, de site wordt weergegeven met behulp van Bootstrap. De applicatie draait op de cytosine server (dit is in de praktijk niet gelukt). Binnen deze cytosine server wordt de webserver aangeroepen met hierin de MOD_WSGI module. Deze module roept het WSGI file aan dat op de cytosine server staat. Vanuit dit WSGI file kan de webapplicatie FLASK aangeroepen worden of een BioPython (Peter J. A. Cock, Biopython: freely available Python tools for computational molecular biology and bioinformatics, 1st edition, march 2009) file. Vanuit de webapplicatie FLASK kunnen andere HTML pagina's binnen de site worden aangeroepen ook deze worden weergegeven met behulp van Bootstrap. Het BioPython file op de Cytosine server wordt aangeroepen wanneer de gebruiker de SQL database wil raadplegen of als de gebruiker via de site wil wil blasten met blastx. Het tweede BioPython file (in de Back-end Tier, BLASTEN.py) is gebruikt om de lege SQL database te vullen, dit is gedaan met 200 sequenties.



Figuur 17 ERD, Applicatie architectuur

Gebruikte Modules:

Package	Version	Latest version
Flask	0.12.2	1.0.2
Flask-Admin	1.5.1	1.5.1
biopython	1.71	1.71
matplotlib	2.2.2	2.2.2
matplotlib-colorbar	0.3.5	0.3.5
matplotlib-subsets	1.0	1.0
mysql-connector	2.1.6	2.1.6
mysql-connector-python	8.0.11	8.0.11
Jinja 2	2.10	2.10
Base64image	0.5.1	0.5.1
Bytesio	0.1	0.1

Testscripts

De functionaliteit van de webapplicatie met de verwachte resultaten, getest 16-6-2018.

Opstarten applicatie en gebruik maken van de blast functie, en deze resultaten opslaan

Stap #	Instructie	Verwacht resultaat	Feitelijk resultaat	Opmerking
1	Open 'app.py'	CMD scherm opent en een localhost link verschijnt (http://127.0.0.1:5000/)	Werkt	Open app.py in de werkmap van de applicatie
2	Kopieer link in webbrowser navigatie	Login scherm toont zich	Werkt	
3	Login met juiste login gegevens	Home scherm toont zich	Werkt	
4	Gebruik navigatie en klik op 'blast'	Blast scherm toont zich	Werkt	
5	Voer een fasta sequentie in het tekstvak in en klik op de 'submit' knop	Blast-result scherm toont zich met resultaten in het tekstvak	Werkt	Moet niet leeg zijn, en een werkende juiste fasta sequentie opgeven.
6	Klik op de 'save' knop	Zodra de resultaten in de database zijn geplaatst zal de applicatie teruggaan naar het blast scherm	Werkt	

Ophalen van de database (vanuit het homescherm)

Stap #	Instructie	Verwacht resultaat	Feitelijk resultaat	Opmerking
1	Gebruik navigatie en klik op 'database'	database scherm toont zich	Werkt	
2	Geef zoekwoorden in het tekstvak op volgens de manier van het voorbeeld	Result scherm toont zich met een grafiek onderaan	Werkt	Geen witregels toegestaan behalve binnen organismen namen

Uitloggen (vanuit het homescherm)

Stap #	Instructie	Verwacht resultaat	Feitelijk resultaat	Opmerking
1	Gebruik navigatie en klik op 'Log out'	Login scherm toont zich	Werkt	

Team scherm navigatie (vanuit het homescherm)

Stap #	Instructie	Verwacht resultaat	Feitelijk resultaat	Opmerking
1	Gebruik navigatie en klik op 'Team'	Team scherm toont zich	Werkt	De foto's zijn klikbaar en sturen de gebruiker naar de website LinkedIn.

Verwijzingen

Grinberg, M. (2014). *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media, Inc.

John Hunter, D. D. (2018, may). *Matplotlib*. Opgehaald van Installation and documentation: <https://matplotlib.org/>

Mehdi Pirooznia, 1. E. (2007, october). *Batch Blast Extractor: an automated blastx parser application*. Opgehaald van PMC-NCBI: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2559874/>

Oracle Corporation and/or its affiliates . (2018). *MySQL Connector/Python Developer Guide*. Opgehaald van MySQL: <https://dev.mysql.com/doc/connector-python/en/>

Peter J. A. Cock, T. A. (2018, april 4). *Module Bio.Blast.NCBIXML*. Opgehaald van Package Bio: Package Blast: Module NCBIXML: <http://biopython.org/DIST/docs/api/Bio.Blast.NCBIXML-pysrc.html>

Peter J. A. Cock, T. A. (2018, april 4). *Module NCBIWWW*. Opgehaald van Package Bio; Package Blast: Module NCBIWWW: <http://biopython.org/DIST/docs/api/Bio.Blast.NCBIWWW-module.html>

Peter J. A. Cock, T. A. (2018, april 4). *Package Entrez*. Opgehaald van Package Bio: Package Entrez: <https://biopython.org/DIST/docs/api/Bio.Entrez-module.html>

Peter J. A. Cock, T. A. (march 2009). *Biopython: freely available Python tools for computational molecular biology and bioinformatics, 1st edition*. ISCB (International Society for Computational Biology).