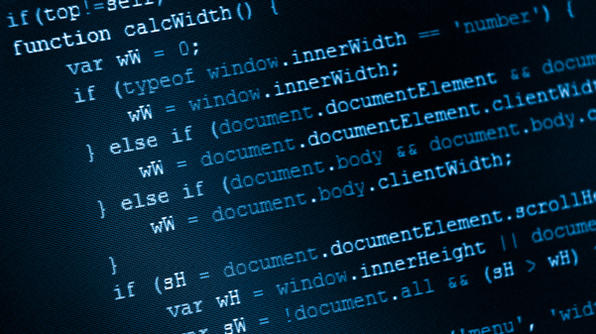
Technische documentatie



Inhoud

* Inleiding
* Werking van de applicatie
* Software architectuur
* Texstscripts en resultaten
* Referenties

Inleiding

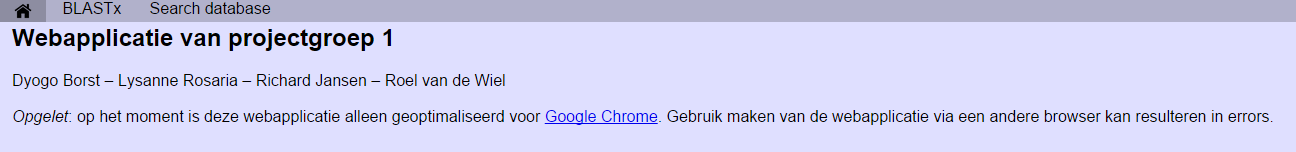
Het doel van onze web-applicatie is dat je aan de hand van een van de sequenties het desbetreffende BLAST-resultaat kan op halen uit onze database. Dit is zeer handig om snel van een sequentie de desbetreffende resultaten op te zoeken.

Het verslag is geschreven voor mensen die niet zo veel van informatica snappen en dus m.b.v. deze handleiding de database zouden kunnen gebruiken. Ook zit er een stuk bij in het verslag wat advies geeft hoe de server onderhouden moet worden en wat de systeemeisen zijn.

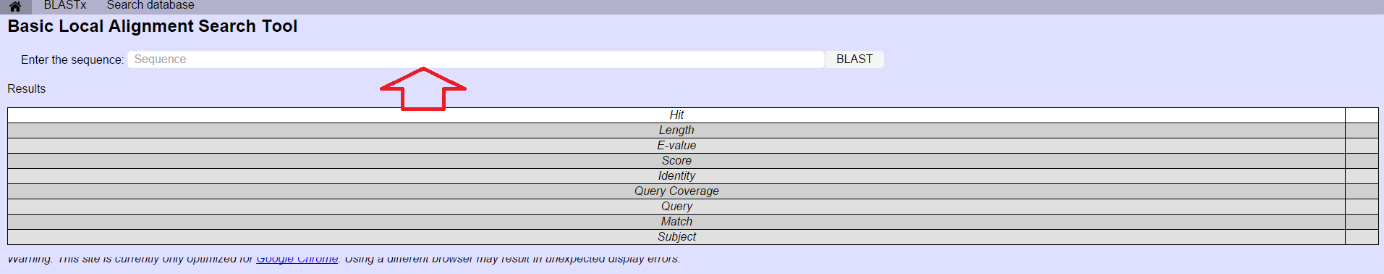
Werking van de applicatie

Voer onderstaande URL in:

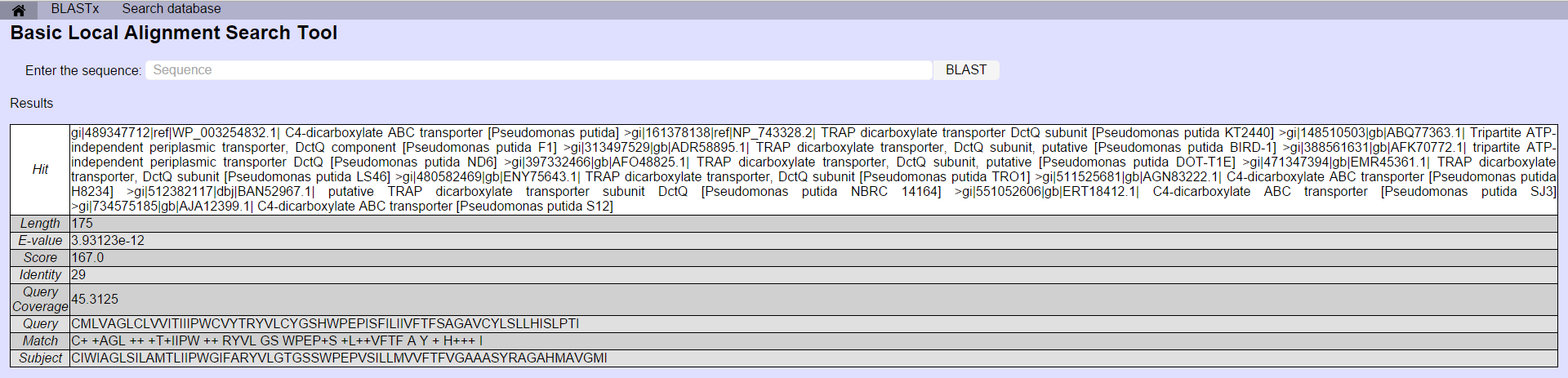
<http://ithurtswhenip.nl:666/pg1/blast.psp>



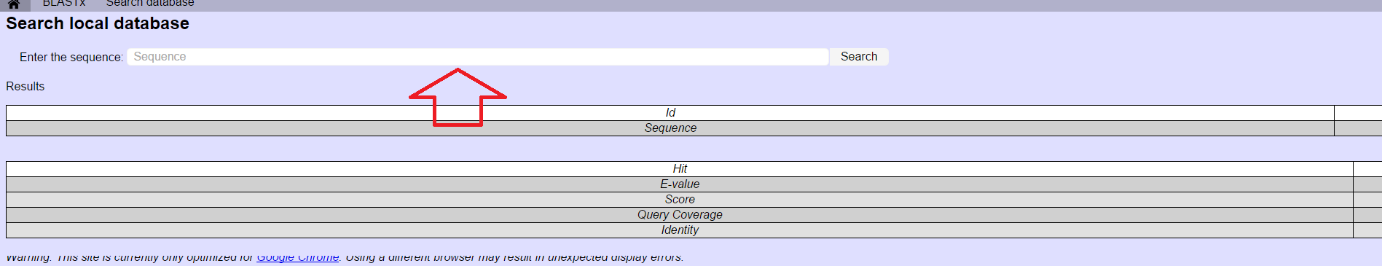
Als u op het kopje BLASTx drukt komt u op onderstaande pagina waarin je een BLAST uit kunt voeren zoals u hieronder kunt zien.



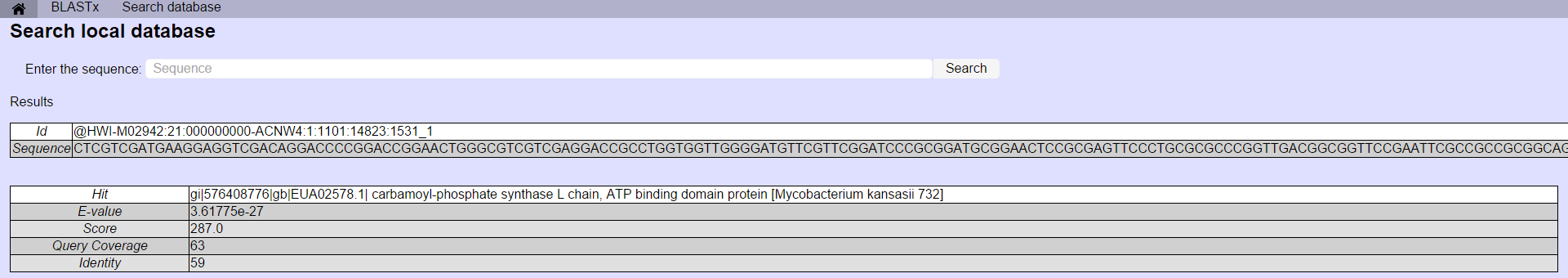
Voer in de balk de sequentie in, hierna zult u een poosje moeten wachten en krijgt u het volgende resultaat



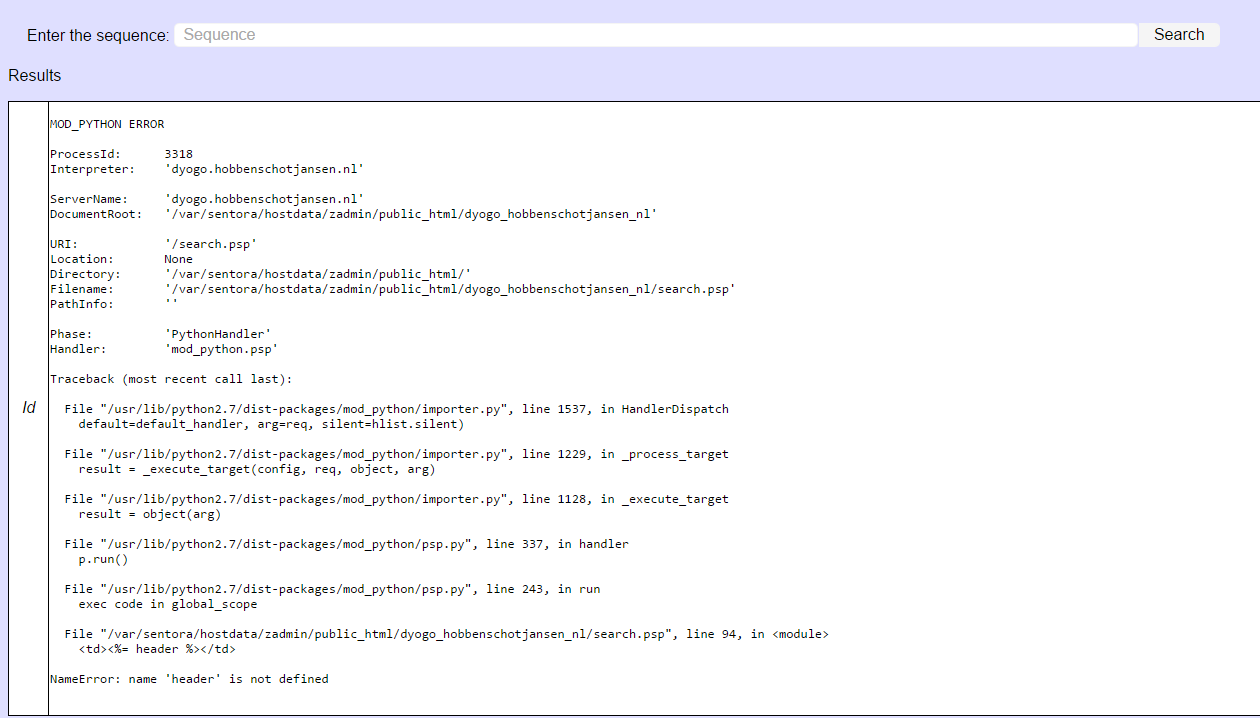
Als u een sequentie heeft en u wilt zien of hij in onze database staat en wat voor waardes deze heeft ga dan naar de “Search Database” optie. Je komt op onderstaande pagina terecht en voer in de zoekbalk de sequentie in.



Bij een Juist resultaat krijgt u dit:



En bij geen hit krijgt u dit:



Deze error zouden we eigenlijk nog met exception handling weg kunnen werken.

Sofware architectuur

**SearchRow1.py**

**Imports**

 re en mysql.connector

**Globals**

a = een default invoer om het script te testen als een client niks invult.

**Input main**

def main(invoer=a): a is de default voor invoer. Deze invoer mag variëren, een sequentie, accessioncode of een stukje woord om naar bijvoorbeeld een organisme of eiwit te zoeken in de headers van de hits uit de blast resultaten.

**Process main**

Deze main functie roopt de functie regex aan met het argument invoer. Verder niets, maar officieel moet ieder script met een main functie beginnen.

**Output main**

De functie main geeft geen output.

**Input regex**

def regex(zoek):zoek is de parameter waar de main functie invoer in stopt.

**Process regex**

Deze functie voert een conditionele controle uit over de string die in zoek zit dmv regular expresions. Deze zijn ontworpen om een DNA sequentie, accession code of een gewoon woord te herkennen voor het opzoeken in de database. Voldoet de invoer hier niet aan, wordt dit door output gemeld. Is zoek een DNA sequentie, dan wordt deze opgezocht in de "nucleotide-sequence" kolom van de tabel "SOURCE\_SEQ" van onze database. Alleen de eerste resultaten worden opgehaald en verwerkt tot leesbare hits voor de website. Later kan dit door het script Searching.py vermeerderd worden naar 10 hits. Is er niks gevonden in de database geeft dit een melding als output.  Als zoek gelijk is aan een accession code wordt deze opgezocht in de seq-id en de result(1-10)-accession-code kolommen. Is de accession code gevonden, dan zal van de eerste hit het resultaat te zien zijn. nogmaals met Searching.py wordt dit alle tien de hits. Is de accession code niet gevonden, geeft dit een melding als output. Is zoek gelijk aan een woord dat gebruikt kan worden om te zoeken op eiwit of organisme in de accession codes wordt dit opgezocht in de accessioncodes en seq-id kolommen van onze database. Wordt er iets gevonden dan wordt de eerste hit gesprint, met Searching.py alle tien. Wordt er niets gevonden, wordt er uiteraard een melding gegeven.

**Output blastOld**

Waneer zoek geen sequentie, header of woord is; "Your input probably isn't a DNA sequence, header or normal word, zo we can't perform a search in our database. Please change your input!!"

Wanneer een sequentie niet in de database gevonden is; "Your sequence can't be found in our database. Blast your sequence with the Blast option"

Wanneer een header niet in de database gevonden is; "Your header can't be found in our database."

Wanneer een woord niet in de database gevonden is; "None of the proteins or organisms in our database corresponds to your string"

**Singleblast.py**

**Imports**

NCBIWWW & NCBIXML van Bio.Blast, re en mysql.connector

**Globals**

a = een default sequentie string in het script om te gebruiken als client niks invult.

**Input main**

def main(seq=a, action = "sb"): a is de default voor seq en "sb" is de default voor action.

**Process main**

Deze main functie voert een conditionene controle uit over de ingevoerde action. Is action gelijk aan "Sequence Blast", "SB" of "sb", dan zal het programma de functie blastOld aanroepen met als argument seq. Is action gelijk aan "Complement Sequence Blast", "CSB" of "csb" dan zal het programma de functie blastNew aanroepen met als argument seq.

**Output main**

Voldoet action niet aan deze zes verschillende waarden, dan geeft het programma als output "Something went wrong!! Try 'sb' or 'csb' with quotes. " en stopt. Dit is handig omdat de server niet sb of csb als input pikt (wel "sb" en "csb" ), maar windows wel. De programmeur voor de html en psp zou dus zijn input in dit programma kunnen aanpassen op de goede manier zoals in de foutmelding is aangegeven.

**Input blastOld**

def blastOld(old):old is de parameter waar de main functie seq in stopt. De naam old omdat in deze functie niks aan de sequentie veranderd wordt, want wel in blastNew het geval is.

Deze functie voert geen proces uit, maar verwijst door naar een nieuwe functie, checkdb met argument old.

**Output blastOld**

Ook heeft deze functie geen output.Pagina-einde

**Input checkdb**

def checkdb(zoek):De parameter zoek wordt ingevuld met old of new, afhankelijk van de action input in de main.

**Process checkdb**

In deze functie wordt er een connectie gemaakt met de voor deze webapplicatie opgezette database op de server van Richard Jansen. Er wordt een query uitgevoerd die zoekt naar seq-id's die het stuk "seqwebapp" bevatten. Vervolgens worden de rows die hier uit komen geteld en in het variabel count geplaatst. Met een regular expression en een conditionele controle wordt bekeken of de input in het programma daadwerkelijk DNA is. Is dit het geval dan wordt de sequentie opgezocht in de database. Als het er al in staat wordt de blast niet uitgevoerd en de output gegeven; "Your sequence already exists in our database. Go to the search option on the website to see the blast results. " . Staat de sequentie nog niet in de database dan zal het programma aangeven dat er een blast gestart wordt en de functie blast met de argumenten zoek en count aanroepen.

**Output checkdb**

Wanneer de ingevoerde sequentie al in de database aanwezig is geeft deze functie de output;     "Your sequence already exists in our database. Go to the search option on the website to see the blast results. "

**Input blast**

def blast (seq, number="")seq vraagt om een sequentie en number vraagt om een int, tenzij die niet meegegeven wordt is dit variabel een lege string als default.

**Process blast**

De blast functie maakt gebruik van de modules NCBIWWW en NCBIXML uit Bio.Blast om connectie te maken met de Blast tool en de sequentie te blasten. De output van blast, een xml bestand wordt over geitereerd met een loop in de blast functie en alle interessante informatie wordt hierbij uit het xml bestand gehaald om weer te geven op de website.  Als er geen blast resultaten zijn wordt dit met output weergegeven en wordt de SOURCE\_SEQ tabel in de database geupdate met inclusief een boolean 0, dat staat voor False omdat er geen hits gevonden zijn. Het programma stop hierna. Zijn er wel resultaten gevonden dan zal gekeken worden op hoeveel number staat en op hoeveel een count c in deze functie om te bepalen dmv conditionele controle hoe de database geupdate moet worden. Deze controle checkt op if number == 0 and c == c, elif number == 0 and c < 11, elif number > 0 and c == 1 en elif number > 0 and c < 11.

**Output blast**

Output bij geen resultaten; "No Blast results have been found to your sequence with Blastx, BLOSUM62, e-value < 0.001, nr database. Try different arguments on the Blast tool website: htp://blast.ncbi.nlm.nih.gov/Blast.cgi. "

**Input blastNew**

def blastNew(old):new is de parameter waar de main functie seq in stopt. De naam new omdat de volgende functies aanpassingen aan de ingevoerde sequentie zullen doen.

Deze functie voert geen proces uit, maar verwijst door naar een nieuwe functie, toComplement met argument old. Hieruit komt een lijst complement dat wordt gebruikt in een volgende functie aanroeping turnAround met argument complement. Dit zorgt voor een string new die getoont wordt als output en vervolgens wordt de functie checkdb aangeroepen met new als argument. Vanaf daar gaat het programma op dezelfde manier verder als beschreven vanaf de functie checkdb en blast.

**Output blastNew**

De nieuwe complementair omgedraaide sequentie.

**Input toComplement**

def toComplement(reverse)reverse is de parameter van toComplement waarin old ingestopt wordt door blastNew. De verwarrende naar reverse omdat dit deel van het programma eerst gebruikt werd om van een reverse sequentie naar een forward sequentie te gaan. Dit kan natuurlijk ook andersom.

**Process toComplement**

Deze functie itereert over de ingevoerde sequentie en per nucleotide vult het een lege lijst met de complementaire nucleotide. Dus voor iedere A een T, T een A, G een C en C een G. Deze gevulde lijst wordt geretourneerd naar blastNew.

**Input turnAround**

def turnAround(list)list is de parameter van turnAround waarin complement wordt gestopt door blastNew.

**Process turnAround**

Dmv het tellen van het aantal nucleotiden in de list wordt een count c gecreeerd . Voor iedere nucleotide in de list word een lege string aangevuld met de nucleotide index aantal nucleotiden in de lijst. Dit wordt herhaald in een loop en steeds wordt er van de count c 1 afgetrokken. De gevulde string wordt geretourneerd naar blastNew.

Testscripts en resultaten

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STAP | Instructie | Verwacht resultaat | Feitelijk resultaat | Opmerking |
| 1 | Ga naar dyogo.hobbenschotjansen.nl en ga naar Search database, voer “ATCG” in. | Je krijgt een hit | Klopt | Hij geeft maar 1 sequence aan terwijl het er waarschijnlijk meer kunnen zijn |
| 2 | Voer “atcg” in. | Je krijgt een hit, of een error. | Een hit | Bij het zoeken maakt het niet uit of je grote of kleine letters gebruikt |
| 3 | Voer “FCTK” in. | Een error of foutmelding | Een Mod\_Python error | Dit kan nog beter afgevangen worden met exception handling. |
| 4 | Ga naar de BLASTx pagina en voer een sequentie van de database in | Je krijgt een uitslag terug | Klopt | Alle resultaten kloppen |
| 5 | Voer nu een gedeelte van een sequentie uit de database terug | Je krijgt een uitslag of een error | Je krijgt een uitslag terug | Het heeft het zelfde resultaat als dat ik de sequentie helemaal zou blasten |

Referenties

Oracle (MySQL) (z.d.). MySQL 5.5 Reference Manual. Geraadpleegd voor het laatst op 15 Juni 2015, van <http://dev.mysql.com/doc/refman/5.5/en/>

Python (Laatst geupdate, 27 Mei 2015). Python 2.7.10 documentation. Geraadpleegd voor het laatst op 22 Juni 2015, van <https://docs.python.org/2/>

W3schools (z.d.). HTML Element Reference. Geraadpleegd voor het laatst op 21 Juni 2015, van <http://www.w3schools.com/tags/>

BLAST (z.d.). BLAST documentation. Geraadpleegd voor het laatst op 22 Juni 2015, van <http://blast.ncbi.nlm.nih.gov/blast/Blast.cgi?CMD=Web&PAGE_TYPE=BlastDocs>