Report of Spider toxins (project 8)

Author: Zhang Yuanyi

1. Introduction

Venomous animals incapacitate their prey using complex venoms that can contain hundreds of unique protein toxins. The realisation that many of these toxins may have pharmaceutical and insecticidal potential due to their remarkable potency and selectivity against target receptors has led to an explosion in the number of new toxins being discovered and characterised.

Spiders are by far the largest group of venomous animals, with more than 40,000 extant species, and they maintain the largest repertoire of pharmacologically active peptides. Some spider toxins have been used for almost two decades as pharmacological tools to probe the structure and function of ion channels and cell-surface receptors. Other spider toxins are being developed as bioinsecticides or providing leads for the development of drugs to treat a diverse range of pathophysiological states such as inflammatory and neuropathic pain, cardiac arrhythmia, and erectile dysfunction.

Spider venom components are typically divided into four groups.

1) Small molecular mass compounds (SMMSs),
2) antimicrobial peptides (only a few spider families),
3) peptide neurotoxins
4) proteins and enzymes.

2. Algorithm design

Central Algorithm---How to pick the right entries/genes

First, we are only interested in spider genes which means the OC line must contain "Araneae". Then, we can have different ways to determine whether we should take it as a toxin.

The simplest way is already provided in the project details, which will contain following steps:

a) Observe the **CC lines** in the entries. It can be seen that there are several sections and they start with "-!-" and capital letters describing the section. Pay attention to the following 3 sections: FUNCTION, TISSUE SPECIFICITY and SIMILARITY. From these sections you can derive that the protein is a toxin, like this:
b) Set a counter to 0.
c) If a FUNCTION section exists and it contains the word "toxin", increase the counter by 1.
d) If a TISSUE SPECIFICITY section exists and it contains the words "venom gland", increase the counter by 1.
e) If a SIMILARITY section exists and it contains the word "toxin", increase the counter by 1.
f) If the counter is at least 2 this is a spider toxin gene.

After study the example file and compare different output result, I choose the following filters to find the toxins.

a) Set 2 **checking word sets.** check_strings contain words: ‘toxin’, ‘toxic’, ‘venom’, ‘Toxin’, ‘Venom’; exclude_strings contain words: 'Non-toxic', 'non-toxic', 'no toxic', 'not toxic', 'Not toxic', 'no toxicity', 'No toxicity'.

b) Observe the **DE lines**. If a DE section exists and it contains the word show in the check_strings, increase the counter by 1. (Actually, only use flag here, increase counter after CC lines)
c) Observe the **CC lines** in the entries. It can be seen that there are several sections and they start with "-!-" and capital letters describing the section. Pay attention to 4 sections: FUNCTION, TISSUE SPECIFICITY, SIMILARITY and TOXIN DOSE. From these sections derive that the protein is a toxin, like this:
d) Set a counter to 0.
e) If a **FUNCTION** section exists and it contains the word in check_strings, increase the counter by 1; if contains the word in exclude_strings, decrease the counter by 3.
f) If a **TISSUE SPECIFICITY** section exists and it contains the words "venom gland", increase the counter by 1.
g) If a **SIMILARITY** section exists and it contains the word in check_strings, increase the counter by 1.
h) If a **TOXIN DOSE** section exists, increase the counter by 1.
i) Observe the **KW lines**. If it contains word ‘Toxin’, increase the counter by 2.
j) If the counter is at least 3 this is a spider toxin gene (considered as a toxic gene).

3. Program design

The whole program can be divided into 4 parts:

1) Frist part is file handle. This part include file open and write and error control.
2) The second part is initialize variables, including flags and strings:
   sp_id = ''
   OCflag = False
   toxin_flag = False
   SQflag = False
   aminoseq = ''
   sum_count = 0
   check_strings = ('toxin', 'toxic', 'venom', 'Venom', 'Toxin')
   exclude_strings = ('Non-toxic', 'non-toxic', 'no toxic', 'not toxic', 'Not toxic', 'no toxicity', 'No toxicity')
   The two strings sets are what I conclude after compare many different adjectives may be used here.
3) Then, this is the core part of the program. We should realize exactly what I list in the Algorithm design.
4) The final part is write the selected swissprot ID of the entry and the protein sequence from the entry.
   So we have to get the protein sequence by doing:

```
# Getting sequence using stateful parsing
# Red line
if identifier == '//' and sum_count >= 3:
    SQflag = False
    sum_count = 0
# Collect data
if SQflag:
    aminoseq += line
# Green line
if identifier == 'SQ' and sum_count >= 3:
    SQflag = True
```

```
# deal with SQ
if aminoseq != '' and not SQflag:
    # Remove spaces, treatment of data after extraction.
    cleanseq = ''
    for i in range(len(aminoseq)):
        if aminoseq[i] != ' ':
            cleanseq = cleanseq + aminoseq[i]
    aminoseq = ''
```

Beside the main.py, I also write a program to compare my result with the standard set I found in NCBI.

4. Program manual

The input file should be the uniprot_sprot.dat file, which is the entire swissprot database, and the output is a .fasta file, where the header is the swissprot ID of the entry and the protein sequence from the entry.

Input example:

```
# Get file name and open file
# filename = input("What swisprot file should I open:")
try:
    infile = open('uniprot_sprot.dat', 'r')
except IOError as err:
    print("Can't open file, reason", str(err))
    sys.exit(1)
```

Output example:

```
# open the write file
try:
    outfile = open('toxin.fsa', 'w')
except IOError as err:
    print("Can't open file, reason", str(err))
    sys.exit(1)
```

Output file example:

```
>29C0_ANCSP
ANACTKQADCAEDECCLDNLFFKRPYCEMRYGAGKRCAAASVYKEDKDLY
>32C7_ANCSP
SDNEFPSGCIEFGKECDLDKGNCQCCRRNGYCSCAVN
>A1H1_LOXHI
LDMGHMVNAIGQIDEFVNLGANSIETDVSFDSSANPEYTYHGIPCDCGRNCKKWENFNDF
LKGLRSATTPGNSKYKEKLVLVVFDLKTGSLYDNQANDAGKKLAKNLLQHYWNNGNNGGR
AYIVLSIPDLNHYPLIKGFTDTLKQEGHPELLDKLGYDFSGNDAIGDVAKAYKKAGVSGH
```

5. Conclusion

To test the outcome, I compared the result between many different filter ways. In the end, I chose the method which have the highest correct rate (all compared to the verified toxic database download from NCBI)

By using difference.py, we can have following comparing result:

There are 1451 protein in standardfile

(uniport-taxonomy%3Aaraneae+keyword%3Atoxin+AND+reviewed%3Ayes.fasta from NCBI)

There are 1429 protein in myfile

myfile have 39 different proteins

They are

['B1H1_LOXIN', 'B1HA_LOXBO', 'F95_CTEON', 'HM2A_PHLSP', 'ICK13_TRILK', 'ICK15_TRILK', 'ICK16_TRILK', 'ICK17_TRILK', 'ICK18_TRILK', 'ICK19_TRILK', 'JZ715_CHIGU', 'OSP1A_ORPS2', 'SE1A_SELEF', 'TBO_TIBOB', 'TX13_CUPSA', 'TX17_PHORI', 'TX1_PHLXX', 'TX1_PSARE', 'TX29A_PHOKE', 'TX29A_PHONI', 'TXA1_LATGE', 'TXA1_LATHE', 'TXA1_LATTR', 'TXA2_LATGE', 'TXA2_LATHE', 'TXA2_LATTR', 'TXA2_STEGR', 'TXAC_STEGR', 'TXAD_STEGR', 'TXAG4_AGEOR', 'TXC9_CUPSA', 'TXFK1_PSACA', 'TXP1_APOSC', 'TXP4_APOSC', 'TXP6_APOSC', 'TXP9_APOSC', 'TXVL2_CORVA', 'VA5_LYCSI', 'VKT1_TRILK']

In conclusion, in these two datasets 1390 genes are shared and 39 genes are different (for myset), but 29 of them are not showed in the standard set. So, it means 1390/1400 were correctly identified and I think the rate is up to standard (>99%).

Maybe there are more efficient ways to achieve this project, as I will continue using and learning python, I hope I can find more interest in my python life.

Reference

1. Wood, D.L., Miljenović, T., Cai, S. *et al.* ArachnoServer: a database of protein toxins from spiders. *BMC Genomics* **10,** 375 (2009). https://doi.org/10.1186/1471-2164-10-375
2. Langenegger, N., Nentwig, W., & Kuhn-Nentwig, L. (2019). Spider Venom: Components, Modes of Action, and Novel Strategies in Transcriptomic and Proteomic Analyses. *Toxins*, *11*(10), 611. https://doi.org/10.3390/toxins11100611
3. Gupta, Ravi & Upadhyay, Ravi. (2016). SPIDER VENOM TOXINS ITS BIOLOGICAL EFFECTS AND ALLERGIC-IMMUNE RESPONSES: A REVIEW Volume 5, Issue 5, XXX-XXX. Review Article ISSN 2277– 7105 *Corresponding Author. World Journal of Pharmaceutical Research. 5. 10.20959/wjpr20165-6124.