Autor:
Bogumił Konopka
Instytut Inżynierii Biomedycznej i Pomiarowej
WPPT, Politechnika Wrocławska, 2011

Przewidywanie struktury drugorzędowej białek za pomocą sieci neuronowych

1 Wprowadzenie

Białka to skomplikowane cząsteczki chemiczne biorące udział we wszystkich procesach zachodzących w organizmach żywych. Są polimerami, składają się z mniejszych jednostek budulcowych – aminokwasów. Istnieje dwadzieścia aminokwasów różniących się atomami tworzącymi łańcuch boczny. Ze względu na zróżnicowaną budowę łańcuchów bocznych aminokwasy charakteryzują się różnymi właściwościami fizykochemicznymi (małe, duże, naładowane dodatnio lub ujemnie, hydrofilowe lub hydrofobowe… więcej o aminokwasach http://en.wikipedia.org/wiki/Amino_acids).

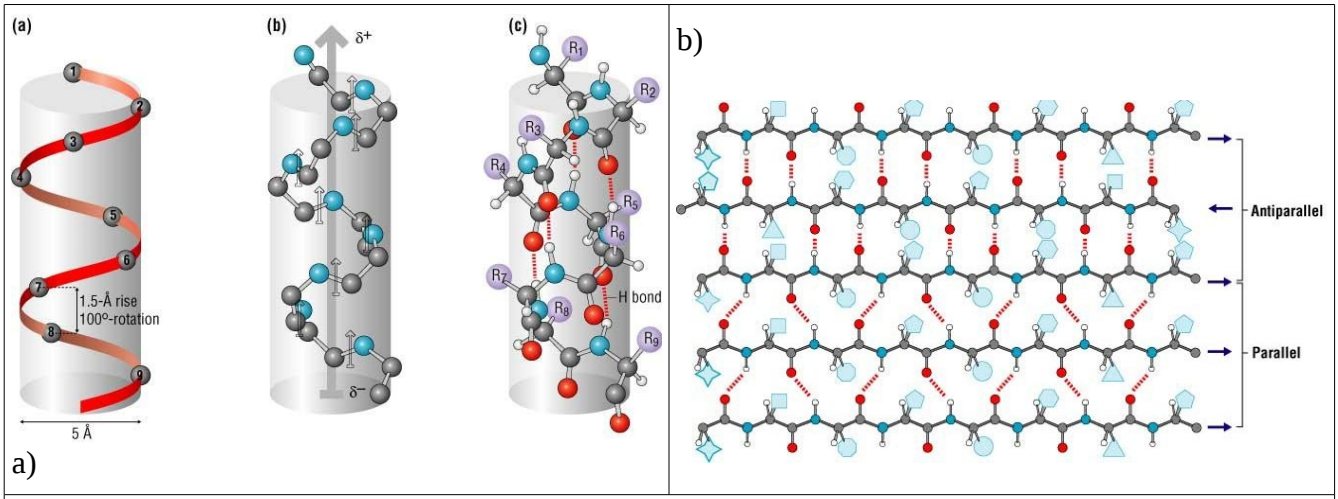
Aminokwasy mają swoje nazwy, które mogą być skrócone do formy trzy lub jednoliterowej np. :

Fenyloalanina (ang. Phelnylalanine) => Phe => F

Kody stosuje się aby u łatwić analizę białek.

Z powodu złożonej struktury, budowę białek analizuje się na różnych poziomach.

1. Struktura pierwszorzędowa to kolejność aminokwasów w sekwencji.
2. Struktura drugorzędowa to charakterystyczne formowanie łańcucha białkowego wynikające z wiązań wodorowych pomiędzy atomami, które go tworzą. Podstawowe struktury drugorzędowe to:
   - alfa helisy
   - struktury beta



Rys. 1 Najczęściej spotykane elementy struktury drugorzędowej: a) alfa helisy; b) struktury beta

3. Struktura trzeciorzędowa białek to ogólne trójwymiarowe ułożenie łańcucha oraz jego topologia. Struktura trzeciorzędowa wynika z oddziaływań pomiędzy atomami łańcuchów bocznych. Bardzo istotne są oddziaływania hydrofobowe oraz siły Van der Waalsa.
4. Struktura czwartorzędowa opisuje interakcje pomiędzy różnymi łańcuchami polipeptydowymi tworzącymi jedną funkcjonalną jednostkę białkową. Struktura czwartorzędowa wynika z podobnych oddziaływań do tych warunkujących strukturę trzeciorzędową.

Znajomość struktur molekularnych, znacznie ułatwia poznanie mechanizmów funkcjonowania białek, umożliwia opracowanie metod ich kontroli – tworzenie leków. Eksperymentalne metody pozwalające na określenie struktury 3D białek są drogie, czasochłonne i nie zawsze skuteczne (szczególnie w przypadku dużych białek). Stopniowo alternatywą stają się metody obliczeniowe. Bardzo często pierwszym krokiem w przewidywaniu pełnej trójwymiarowej struktury białek jest określenie jego struktury drugorzędowej. Najskuteczniejsze dostępne obecnie metody posiadają dokładność przewidywania sięgającą 90-95% (np. PSI-PRED).

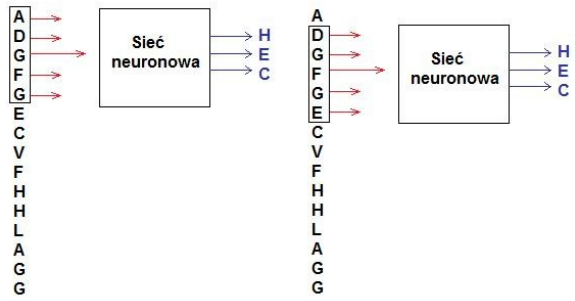
Twoim zadaniem jest zaprojektowanie własnej metody przewidującej strukturę drugorzędową białek w oparciu o znajomość sekwencji aminokwasów. Do tego celu wykorzystasz poznaną na wcześniejszych zajęciach technologię sieci neuronowych. Ćwiczenie ma charakter konkursu. Ten kto stworzy najskuteczniejszy predyktor struktury drugorzędowej otrzyma dodatkowe 3 pkt do zaliczenia kursu.

2 Przewidywanie struktury drugorzędowej

Sieci neuronowe były już wykorzystywane do rozwiązania problemu przewidywania struktury drugorzędowej białek (dla ciekawych:

- D. G. Kneller, F. E. Cohen and R. Langridge (1990) "Improvements in Protein Secondary Structure Prediction by an Enhanced Neural Network" J. Mol. Biol. (214) 171-182.
- H. Zhu, I. Yoshihara, K. Yamamori, (2002) Prediction of Protein Secondary Structure by Multi-Modal Neural Networks, Procceding of 2002 International Joint Conference on Neural Networks IJCNN'02

Program pisany na zajęciach będzie skanował sekwencję aminokwasów oknem przesuwnym o określonej szerokości i prezentował aminokwasy wyuczonej sieci neuronowej. Sieć będzie klasyfikować aminokwas znajdujący się w środku okna jako fragment helisy (H), struktury beta (E) lub część niestrukturyzowaną (C).



Rys. 1. Schemat działania programu.

3 Przygotowanie zbioru uczącego

Zbiorem uczącym będą sekwencje aminokwasowe białek, których struktury zostały określone eksperymentalnie i o których wiadomo, gdzie znajdują się alfa helisy i struktury beta. Do budowy zbioru uczącego wykorzystaj 10 (lub więcej) białek losowo wybranych z listy podanej w serwisie EVA (PDB statistics => Index for PDB files)
http://www.pdg.cnb.uam.es/eva/doc/intro_sec.html

Przygotuj listę i zapisz ją w pliku Learning_PDB.txt

Skorzystaj z bazy Stride opisanej w artykule "STRIDE: a web server for secondary structure assignment from known atomic coordinates of proteins" Matthias Heinig and Dmitrij Frishman

(http://www.ncbi.nlm.nih.gov/pmc/articles/PMC441567/)

Wejdź na stronę Stride http://webclu.bio.wzw.tum.de/stride/ i przejdź do bazy wyliczonych danych (Stride Database)

Żeby uzyskać listę przypisań struktur drugorzędowych dla każdego wybranego białka w oknie „All indices" wklej listę swoich wybranych białek z pliku Learning_PDB.txt (przykład: 16vp| 1a26 |1a5t)

Wybierz konkretne białko z tabelki i przygotuj dla niego plik wejściowy dla sieci neuronowej. Zanotuj w pliku Learning_PDB.txt, co to za białko i jaką ma długość.

W zakładce „Plain" pokazują się surowe dane tekstowe opisujące białka. Przyjrzyj się otrzymywanym wynikom. Czy wiesz co znajduje się w każdej części pliku?

Skopiuj sekcję „REM ---------- Detailed secondary structure assignment---------- 16VP"

do arkusza kalkulacyjnego OpenOffice. Wklej specjalnie tekst niesformatowany (Ctrl+Shift+V) wydzielając kolumny z trzyliterowym kodem aminokwasów oraz jednoliterowym kodem rozpoznanej struktury, resztę usuń. Dla ułatwienia kopiuj informację dotyczącą tylko jednego łańcucha aminokwasowego !!

Za pomocą programu http://molbiol.ru/eng/scripts/01_17.html zamień trzyliterowy kod aminokwasów na kod jednoliterowy.

Stride oferuje pełniejszy opis struktur drugorzędowych. Rozróżnia 8 typów struktur. My uprościmy opis do trzech typów (H-alpha helisy, E – beta struktury, C – pętle). Aby uprościć opis usuń z arkusza kolumnę z kodem 3 literowym. Następnie zamień odpowiednio litery w pliku (Ctrl+F):

- T na C;
- G oraz I na H,
- B oraz b na E.

Do pierwszej kolumny wklej otrzymany wcześniej kod jednoliterowy aminokwasów. W arkuszu kalkulacyjnym powinny się znajdować dwie kolumny: pierwsza z kodem jednoliterowym aminokwasów druga z kodem struktury drugorzędowej (H,E,C). Wklej obie kolumny do pliku tekstowego i zapisz pod nazwą <pdb_id>_ss.txt (użyj odpowiedniego identyfikatora pdb). Zanim rozpoczniesz uczenie przygotuj 20 plików wejściowych.

4 Uczenie sieci

Aby wprowadzić do sieci informację o kolejności aminokwasów w sekwencji konieczne jest ich zakodowanie. Do kodowania zastosujesz macierz zamian aminokwasów (substitution matrix) BLOSUM (http://en.wikipedia.org/wiki/BLOSUM). Macierze zamian są wynikiem analiz statystycznych przeprowadzonych na zbiorach zachowywanych ewolucyjnie sekwencji aminokwasowych. W uproszczeniu przedstawiają one prawdopodobieństwo zamiany w wyniki mutacji jednego aminokwasu w sekwencji na inny. Prawa ewolucji eliminują z przyrody drastyczne, niefunkcjonalne mutacje. Podobne aminokwasy pozwalają zachować funkcję, dlatego też macierze BLOSUM są w pewnym sensie sposobem na wyrażenie podobieństwa pomiędzy aminokwasami.

Kodem określającym dany aminokwas jest jego wiersz z macierzy BLOSUM. Aby określić strukturę drugorzędową, której fragmentem jest dany aminokwas, będziesz dostarczać sieci informację o aminokwasach znajdujących się w jego sąsiedztwie. Będziesz skanować sekwencję aminokwasów oknem przesuwnym, pozwalając sieci na klasyfikację aminokwasu znajdującego się w jego środku. Zbuduj zbiór uczący następnie wytrenuj sieć i ją przetestuj.

Teraz konieczne jest przeformatowanie danych, aby mogły być użyte w procesie uczenia sieci. Do tego celu wykorzystaj podane poniżej funkcje. Co robi każda z podanych funkcji?

Procedura postępowania:

1.Wczytaj plik => 2. Zakoduj aminokwasy => 3. Zakoduj prawidłowe struktury => 4. Przygotuj informację wejściową do sieci (okienko przesuwne).

```
%MATLAB
clear all
%preparing blosum
blosum_aa_order='ARNDCQEGHILKMFPSTWYVBZX*'
blosum_mat=blosum62; %chose blosum type here

window_size=9 %should be odd
additional_shift=(window_size-1)/2;

%reading in a file with amino acids and their assigned secondary structures
[aa,targets_char]=read_ss_file('16vp_ss.txt');



%coding aminoacids with blosum
aa_coded=code_aa_seq(aa,blosum_mat);

%adding 0 records at the ends of sequence
added=zeros(additional_shift,length(blosum_aa_order));
aa_coded=[added;aa_coded;added];
```

```
%conding targets
targets=code_targets_seq(targets_char);

%sliding window
inputs=zeros(length(aa),window_size*length(blosum_aa_order));
for i=1:length(aa)
    inputs(i,:)=inputs2nn(aa_coded(i:(i+window_size-1),:));
end
    inputs=inputs' %matlab prefers samples in columns
    targets=targets' %matlab prefers samples in columns
************************************************************
```

Funkcje składowe

```
function [aa,target]=read_ss_file(file_name)

f_id=fopen(file_name);
code=fscanf(f_id,'%s\t',[2 inf]);
code=code';
fclose(f_id);
aa=code(:,1);
target=code(:,2);

function aa_coded=code_aa_seq(aa,blosum_mat)
%preparing blosum62
blosum_aa_order='ARNDCQEGHILKMFPSTWYVBZX*'
%blosum62_mat=blosum62;

aa_coded=zeros(length(aa),length(blosum_aa_order));
for i=1:size(aa,1)
    aa_coded(i,:)=blosum_mat(find(blosum_aa_order==aa(i)),:);
end

function targets=code_targets_seq(targets_char)

targets=zeros(length(targets_char),3);

for i=1:length(targets_char)
     switch(targets_char(i))
        case 'H'
```

```
        targets(i,1)=1;
        case 'E'
        targets(i,2)=1;
        case 'C'
        targets(i,3)=1;
    end
end
function nn_inputs=inputs2nn(input)
nn_inputs=input(:);

**************************************
```

Uruchom skrypt i zobacz jakie zmienne pojawiły się w przestrzeni roboczej matlaba.

Za pomocą narzędzia nprtool zbuduj i naucz sieć do rozpoznawania struktury drugorzędowej. Sprawdź krzywe ROC i macierze pomyłek. Czy w tym przypadku mają one sens?

Jedno białko to bardzo mały zbiór uczący. Teraz stwórz duży zbiór uczący dla wszystkich białek w tym celu trzeba zautomatyzować proces kodowania białek. Wykorzystaj funkcję „nn_ss_format_input” (poprzedni skyrpt zamknięty w pojedynczej funkcji):

```
function
[aa,targets_char,targets,inputs]=nn_ss_format_input(file_name,window_size,blosum_ma
t)
blosum_aa_order='ARNDCQEGHILKMFPSTWYVBZX*';
    additional_shift=(window_size-1)/2;
%reading in a file with amino acids and their assigned secondary structures
    [aa,targets_char]=read_ss_file(file_name);



    %coding aminoacids with blosum
    aa_coded=code_aa_seq(aa,blosum_mat);

    %adding 0 records at the ends of sequence
    added=zeros(additional_shift,length(blosum_aa_order));
    aa_coded=[added;aa_coded;added];

    %conding targets
    targets=code_targets_seq(targets_char);
    %sliding window
```

```
    inputs=zeros(length(aa),window_size*length(blosum_aa_order));
    for i=1:length(aa)
        inputs(i,:)=inputs2nn(aa_coded(i:(i+window_size-1),:));
    end
    inputs=inputs' %matlab prefers samples in columns
    targets=targets' %matlab prefers samples in columns
****************************************
```

Główny skrypt zmodyfikuj w następujący sposób:

```
clear all
%preparing blosum
blosum_aa_order='ARNDCQEGHILKMFPSTWYVBZX*'
blosum_mat=blosum62; %chose blosum type here

window_size=9 %should be odd
additional_shift=(window_size-1)/2;
file_names(1)={'1ihm_ss.txt'};
file_names(2)={'1gny_ss.txt'};
file_names(3)={'1g39_ss.txt'};
file_names(4)={'1fiw_ss.txt'};
file_names(5)={'1f6a_ss.txt'};
file_names(6)={'1ddm_ss.txt'};
file_names(7)={'1c01_ss.txt'};
file_names(8)={'1b8e_ss.txt'};
file_names(9)={'1b0c_ss.txt'};
file_names(10)={'16vp_ss.txt'};
inputs_final=[];
targets_final=[];
for i=1:10
    [aa,targets_char,targets,inputs]=nn_ss_format_input(file_names{i},window_size,b
losum_mat);
inputs_final=[inputs_final inputs];
    targets_final=[targets_final targets];
end
```

5 Badanie i testowanie

Tak jak na poprzednich zajęciach wygeneruj skrypt do tworzenia sieci neuronowej. Do uczenia wykorzystaj wybrany przez siebie zbiór białek. Testuj różne konfiguracje sieci, zmieniaj architektury, algorytmu uczenia, funkcje transferu. Wszystkie chwyty dozwolone. Twoim celem jest wytrenowanie sieci, która będzie najlepiej przewidywała strukturę drugorzędową. Do oceny skuteczność sieci możesz wykorzystać funkcję „ss_pred_eval" lub inne poznane wcześniej metody.

Zanim skorzystasz z „ss_pred_eval" musisz użyć funkcji decode_output. Po co?

```
function [outputs_char,outputs_bin]=decode_output(outputs)
max_values=max(outputs);
outputs_bin=zeros(3,size(outputs,2));
for i=1:size(outputs,2)
    structure=find(outputs(:,i)==max_values(i));
    switch structure
        case 1
            outputs_char(i)='H';
            outputs_bin(1,i)=1;
        case 2
            outputs_char(i)='E';
            outputs_bin(2,i)=1;
        case 3
            outputs_char(i)='C';
            outputs_bin(3,i)=1;
    end
end
```

Publikacje mówią o różnych miarach o oceny skuteczności przewidywań struktury drugorzędowej. Bardzo popularne są : QH, QE, QC, Q3. Zdefiniowane jak poniżej. Zastosuj podaną funkcję do wyliczenia odpowiednich parametrów dla swoich sieci.

$$Q_3 = 100 * \frac{1}{N_{Res}} \cdot \sum_{i=1}^{3} pred_i ,$$

pred – poprawnie przewidziany element struktury.

$$Q_i = 100 \cdot \frac{pred_i}{obs_i}$$

pred – poprawnie przewidziane elementy strutktury,

obs – wszystlkie obserwowane elementy struktury.

```
function [QH,QE,QC,Q3]=ss_pred_eval(targets,outputs_bin)
```

```
QH=sum(outputs_bin(1,targets(1,:)==1))/sum(targets(1,:)==1)*100;
QE=sum(outputs_bin(2,targets(2,:)==1))/sum(targets(2,:)==1)*100;
QC= sum(outputs_bin(3,targets(3,:)==1))/sum(targets(3,:)==1)*100;
Q3=(sum(outputs_bin(1,targets(1,:)==1))+sum(outputs_bin(2,targets(2,:)==1))...
    +sum(outputs_bin(3,targets(3,:)==1)))/size(targets,2)*100;
```

Możesz zwizualizować wyniki wykorzystując funkcję ss_fprintf_res.m .

```
function ss_fprintf_res(file_name,aa_char,targets_char,outputs_char)
f_id=fopen(file_name,'w');
fprintf(f_id,' aa: %s\nobs: %s\n pr: %s\n',aa_char,targets_char,outputs_char);
fclose(f_id);
```

Do wybranego pliku zostają zapisane informacje o predykcji. Jak je interpretować?

6 Dodatek - meta-metoda

Skonstruuj meta-metodę do przewidywania struktur. Połącz wyjścia z różnych sieci. Często takie połączenie może poprawiać efekt samotnego działania pojedynczej sieci.

7 Raport

W raporcie opisz krótko swój zbiór uczący (jakie białka i jakich długości, ile sumarycznie alfa-helis, beta-struktur – podpowiedź: “plotconfusion”).

Opisz jakie architektury sieci i układy parametrów przetestowałeś. Zaprezentuj wyniki. Pokaż różne zależności (np. liczba neuronów w warstwie ukrytej, rozmiar i skład zbioru uczącego...) Pamiętaj, że proces uczenia sieci jest losowy i sieć o tej samej architekturze dobrze jest testować kilka razy, żeby sprawdzić jej skuteczność. Zmierz wpływ zbioru uczącego na skuteczność najlepszej sieci (dodaj/odejmij białka).

8 Uwagi

Uwagi proszę kierować na:

bogumil.konopka@pwr.wroc.pl