

# Bioinformatyka strukturalna

**Autor: Anna Glinka**

**Temat: Porównanie zbioru struktur drugorzędowych RNA (z pominięciem pseudowęzłów)**

## Wstęp

Drugorzędowe struktury RNA ściśle wiążą się z funkcją którą pełnią w żywych organizmach. Dlatego też ewolucyjnie konserwowane wydają się być nie tylko same sekwencje RNA, ale także struktury wyższych rzędów [1]. W ciągu ostatnich lat stworzono wiele algorytmów służących porównywaniu struktur RNA, wydaje się jednak, że do dziś nie stworzono jednego do końca zadowalającego algorytmu [4]. W ramach projektu zaimplementowano więc 3 różne podejścia do prezentowanego problemu: porównanie za pomocą metryki Hausdorffa, metryki „góry” oraz metryki „drzewa”. Każda z metod daje inne wyniki i zwraca uwagę na inne elementy struktury, dlatego też nie jest możliwe bezpośrednie porównywanie wyników uzyskanych różnymi metodami.

## Cel

Celem prezentowanego projektu jest implementacja algorytmów umożliwiających porównanie zbioru struktur drugorzędowych RNA nie zawierających pseudowęzłów, przedstawionych w notacji dot-brackets.

## Metodyka

### Metryka „drzewa”

Metoda należy do tzw. metod gruboziarnistych, w których nie analizujemy poszczególnych par, a jedynie całe struktury drugorzędowe. Do stworzenia drzewa wymagane jest najpierw przyporządkowanie do sekwencji nawiasów i kropek odpowiednich struktur drugorzędowych. Z ciągu rozpoznanych struktur budowane jest następnie drzewo, w którym wierzchołki odpowiadają rozpoznany częściom, jak na przykład pętla, spinka do włosów czy wybrzuszenie, a krawędzie to fragmenty sparowanych nukleotydów pnia. Wierzchołkom przyporządkowane zostają numery zgodnie z liczbą odchodzących od nich krawędzi (pętla, wybrzuszenie, niesparowane fragmenty na końcach sekwencji = 1, spinka do włosów = 0, skrzyżowanie – zależy od liczby odchodzących gałęzi). Dalsza część algorytmu obejmuje przeszukiwanie drzew porównywanych sekwencji w głąb. Stworzone w ten sposób ciągi liczb są następnie dopasowywane (w prezentowanym projekcie wykorzystano algorytmu Needleman -Wunsh'a), co daje wzajemne podobieństwo sekwencji [3].

Zastosowane skróty i oznaczenia :

d – pień

s – spinka do włosów

l – niesparowane nukleotydy na końcach łańcucha

p – pętla wewnętrzna

w – wybrzuszenie

x – skrzyżowanie

### **Szczegóły implementacyjne – metryka „drzewa”**

#### Funkcja elementyStruktury

Do rozpoznawania struktur wykorzystano strukturę stosu (odnalezienie łączących się nukleotydów) oraz wyrażenia regularne. Stos dodatkowo pełni funkcję kontrolną, w przypadku podania nieprawidłowej liczby nawiasów zgłoszony zostaje błąd. Program umożliwia analizę struktur posiadających pseudowęzły, w takim wypadku nawiasy inne niż „(”, „)” zostają zamienione na kropki oznaczające brak parowania. Parametrem przekazywanym do funkcji jest sekwencja w notacji kropkowo-nawiasowej, wartością zwracaną jest natomiast stri2 (ciąg symboli będący

reprezentacją obecnych w RNA struktur drugorzędowych oraz ich topologii, poszczególne gałęzie drzewa oddzielane są nawiasami „,]”, „[”).

Wyłącznie dla potrzeb implantacyjnych zastosowano dodatkowe oznaczenie:

r – oznaczenie struktury prezentowanej w notacji kropkowo-nawiasowej podobnie do spinki do włosów, jednak bez niesparowanych nukleotydów w środku. np. (((()))

z – oznaczenie fragmentu z lewej strony będącego odpowiednikiem wybrzuszenia występującego po prawej stronie struktury, np. ((((((...(((.....))))))...)))...)). Dodatkowo wyświetlane jest rozpoznanie struktury przyporządkowane do konkretnego nukleotydu – stri.

### **Funkcja drzewoDFS**

Argumentem przekazywanym funkcji jest stri2 (kod drzewa zwracany przez poprzednią funkcję), z którego odczytuje ona ciąg cyfr, odpowiadający przeszukiwaniu grafu w głąb. Zwracana sekwencja wykorzystywana jest następnie przez funkcję NeedlemanWunsch.

### **Funkcja NeedlemanWunsch**

Funkcja przyjmuje 2 parametry, będące sekwencją powstałą w czasie czytania porównywanych drzew algorytmem DFS. Punktacja podstawowa ustawiona została na dopasowywanie = 1, niedopasowanie i przerwa = -1. Wyświetlane są sekwencje po dopasowaniu, procent ich identyczności oraz punktacja.

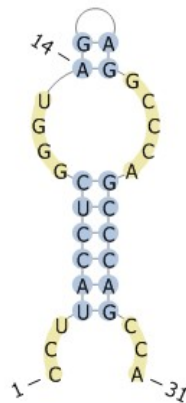
### **Testowanie rozpoznawania struktury**

#### **Struktury stworzone na potrzeby testowania:**

##### **Sek 1**

CCUUACCUCGGGUAGAGGCCCCAGCCCAGCCA

...(((((((.....(O).....))))))....



```
RNA:  ...(((((((.....(O).....))))))....
stri:  111dddddppppdddddppppddddd111
stri2: 1prp1
dfs: 110
```

##### **Sek 2**

CCUUACCUCGGGUAGAGGCCCCACGGGUAGAGGCCCAAG

(((((((((.....(((O))).....)))))).....))))



```

RNA:  (((((((((((.....((((())))).....)))))).....))))))
stri:  ddddddddddpppppdddddddddpppppddddddwwwwwwddddd
stri2:  zprpw
dfs:  110

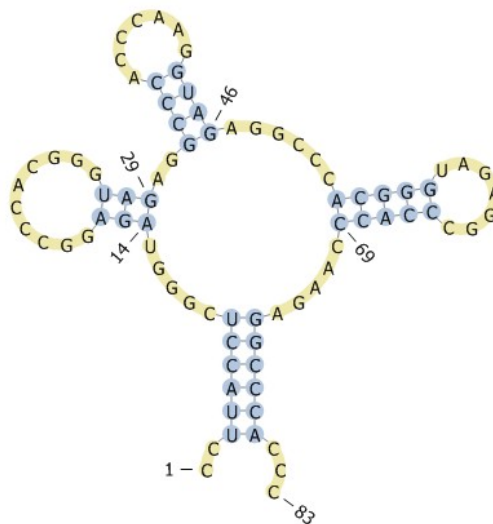
```

### Sek 3

```

CCUUACCUCGGGUAGAGGCCCCACGGGUAGAGGCCCCACCCAAGGUAGAGGCCCCACGGG
UAGAGGCCCCACCCAAGAGGCCCCACCC
..(((((((.....(((.....))))..((((.....)))).....((((.....))))..(((.....))))))....

```



```

RNA:  ..(((((((.....(((.....))))..((((.....)))).....((((.....))))..(((.....))))))....
stri:  11ddddddxxxxdddssssssssssdddxxxxdddssssssssdddxxxxxxdddssssssssdddxxxxxxddd111
stri2:  1[s][[s]][s]1
dfs:  13000

```

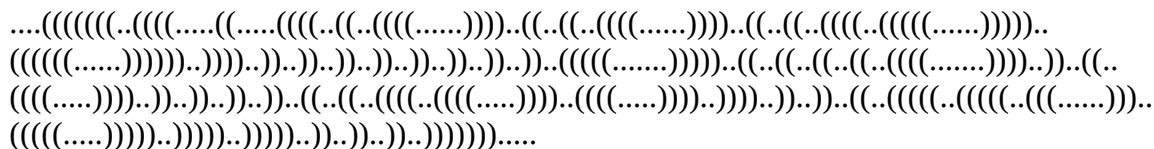
### Sek 1\_1

```

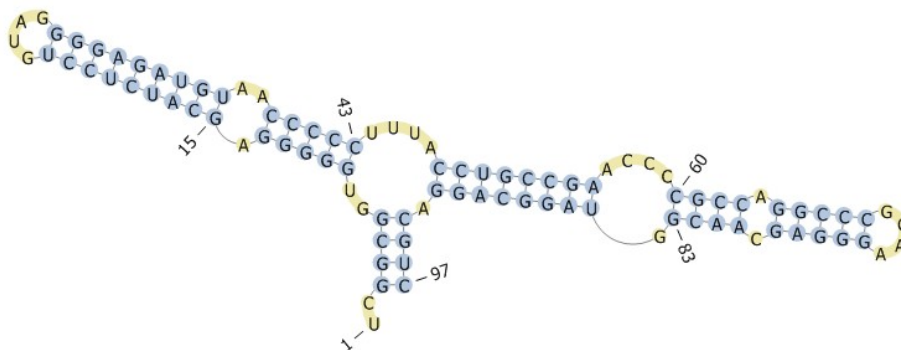
CCUUACCUCGGGUAGAGGCCCCACCUUACCUCGGGUAGAGGCCCCACCUUACCUCGGGU
AGAGGCCCCACCUUACCUCGGGUAGAGGCCCCACCUUACCUCGGGUAGAGGCCCCACCUU
ACCUCGGGUAGAGGCCCCACCUUACCUCGGGUAGAGGCCCCACCUUACCUCGGGUAGAG
GCCACCUUACCUCGGGUAGAGGCCCCACCUUACCUCGGGUAGAGGCCCCACCUUACCU
CGGGUAGAGGCCCCACCUUACCUCGGGUAGAGGCCCCACCUUACCUCGGGUAGAG

```

CCUUACCUCGGGUAGAGGCCCCACCUUACCUCGGGUAGAGGCCCCACCUUACCUCGGGU  
AGAGGCCCCACCUUACCUCGGGUAGAGGCCCCACCUUACCUCGGGUAGAGGCCCCACCUU  
ACCUCGGGUAGAGGCCCCACCUUACCUCGGGUAGAGGCCCCACCUUACCUCGGGUAGAG  
GCCCCACCUUACCUCGGGUAGAGGCCCCACCUUACCUCGGGUAGAGGCCCCACCUUACCU  
CGGGUAGAGGCCCCACCUUACCUCGGGUAGAGGCCCCACCUUACCUCGGGUAGAGGCCUU  
ACCUCGGGUAGAGGCCCCACCUUACCUCGGGUAGAGGCCCCACCUUACCUCGGGUAGAG  
GCCCCACCUUACCUCGGGUAGAGGCCCCACCUUACCUCGGGUAGAGGCCCCCUCGGGUAG  
AGGCCC

[illegible]

UCGGCGGUGGGGGAGCAUCUCCUGUAGGGGAGAUGUAACCCCUUUACCU  
GCCGAACCCCGCCAGGCCCGGAAGGGAGCAACGGUAGGCAGGACGUC

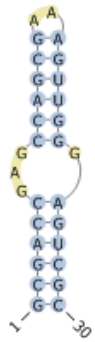
[illegible]

((((....((((((((....))))).))))).))))))

RNA: ..((((..((((((((((((....)))))))).))))).(((.....((((..  
((((....))))).))))).))))))  
stri:   
11dddxdddpdddddssssdddddppdddxdddddppppdddpdddddssssdddddppdddpdddddxd  
stri2: 1[psp][ppspp]  
dfs: 1211010

### Sek PDB\_00142

GCGACCGAGCCAGCGAAAGUUGGGAGUCGC

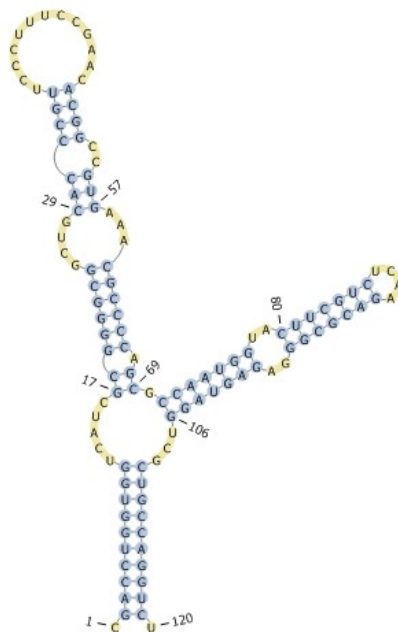


((((((...(((((..)))))).))))))

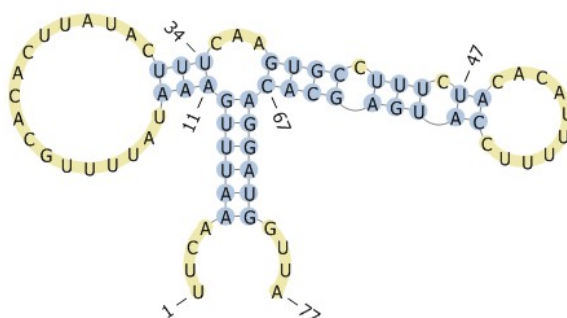
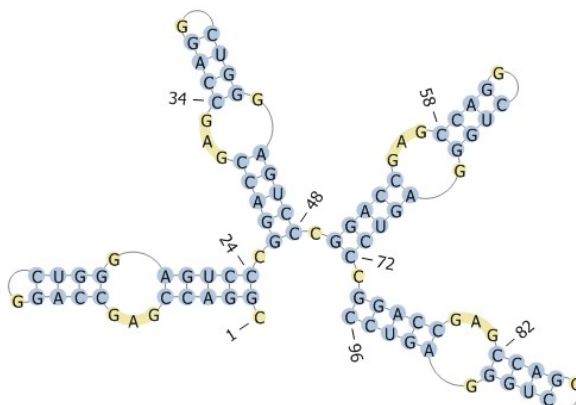
RNA: ((((((...((((((..)))))).))))))  
stri: ddddddpppdddddssdddddppddddd  
stri2: psp  
dfs: 10

### Sek CRW\_00552

CGACCUGGUGGUCAUCGCGGGGCGGCUGCACCCGUUCCCUUUCCGAACAC  
GGCCGUGAAACGCCCCAGCGCCAAUGGUACUUCGUCUCAAGACGCGGGAG  
AGUAGGUCGCUGCCAGGUCU



```
RNA: .((((((((((....((((((((((((((((((.....)))))).)))))).)).(((((((...
((((((((((....)))))).)))))).)))))).)))))).)))))).
str1: lddddddddddxxxxddddddppppdddddssssssssssssdddwddpppdddddwdxdddddpppdddddssssd
dddddpppdddddxxddddddddd1
str2: l[zpswpw][psp]l
dfs: 12101110
```

$$\dots((((((((((((\dots)))))))))\dots(((((((((\dots)))))))))\dots$$

$$\begin{aligned} & .((((...(((.)))) .)))) .((((...(((.)))) .)))) .( \\ & (((...(((.)))) .)))) .((((...(((.)))) .)))) \end{aligned}$$


```
RNA: .((((...(((.)))..)))..((((...(((.)))..)))..((((...(((.)))..)))..((((...
((((..)))..))) )
stri: ldddddppppdddsddddpdddddxddddpppdddsdddpdddddxddddpppdddsdddpdddddxddddpppdddsdddpd
ddddd
stri2: 1[psp][[psp]][[psp]][psp]
dfs: 1410101010
```

## Metryka Hausdorffa

Metryka Hausdorffa umożliwia określenie odległości pomiędzy podzbiorami. W przypadku RNA została wykorzystana do porównywania zestawów par w poszczególnych sekwencjach. Odległość  $d_h$  jest definiowana w 3 krokach: maksimum z odległości pomiędzy parami w porównywanych sekwencjach (Tabela 1), minimum z obliczonych poprzednio odległości po rzędach i kolumnach (Tabela 2), maksimum z tych minimalnych odległości i na końcu maksymalna wartość z tych dwóch [4].

### Przykład 1.

$S1 = \dots((\dots))$ .

$S2 = ..((\dots))..$

$PS1 = \{\{4, 10\}, \{5, 9\}\}$

$PS2 = \{\{3, 9\}, \{4, 8\}\}$

Tabela 1.

Pary PS2/PS1	PS2_1	PS2_2
PS1_1	$\max(1, 1)$	$\max(0, 2)$
PS1_2	$\max(2, 0)$	$\max(1, 1)$

Tabela 2.

Pary PS2/PS1	PS2_1	PS2_2	Rz min
PS1_1	1	2	1
PS1_2	2	1	1
kol min	1	1	

$da(PS1, PS2) - \max \text{ z min po kolumnach} = 1$

$da(PS2, PS1) - \max \text{ z min po rzędach} = 1$

$d_h - \max \text{ z max} = 1$

### Przykład 2.

$S3 = (((((\dots))))))\dots\dots\dots$

$S4 = (((((\dots))))))\dots\dots\dots(\dots)\dots$

$PS3 = \{\{1, 11\}, \{2, 10\}, \{3, 9\}, \{4, 8\}\}$

$PS4 = \{\{1, 11\}, \{2, 10\}, \{3, 9\}, \{4, 8\}, \{23, 27\}\}$

Tabela 3.

Pary PS4/PS3	PS4_1	PS4_2	PS4_3	PS4_4	PS4_5	Rz min
PS3_1	0	1	2	3	22	0
PS3_2	1	0	1	2	21	0
PS3_3	2	2	0	1	20	0
PS3_4	3	2	1	0	19	0
kol min	0	0	0	0	19	



$dh(S3, S4) = 19$

### Szczegóły implementacyjne - metryka Hausdorffa

#### Funkcja metrykaHausdorffa

Argumentami funkcji są sekwencje w notacji kropkowo-nawiasowej. Wykorzystuje ona strukturę stosu do identyfikacji par nukleotydów oraz dwuwymiarową macierz do dalszych obliczeń. Zwracaną wartością jest odległość porównywanych sekwencji.

#### Metryka „góry” (mountain metric)

Dla każdej sekwencji tworzony jest wektor z elementów będących różnicą pomiędzy liczbą nawiasów otwierających i zamykających. Poszczególne wektory są sumowane, a wartość bezwzględna pomiędzy różnicą stanowi odległość między sekwencjami [2, 4].

$S1 = ..(((.....)))..$        $vS1 = (0; 0; 1; 2; 3; 3; 3; 3; 3; 3; 2; 1; 0; 0; 0) = 24$

$S2 = .(((.....)))..$        $VS1 = (0; 1; 2; 3; 4; 4; 4; 4; 4; 4; 3; 2; 1; 0; 0) = 36$

$S3 = ..((((.....)))..$        $vS3 = (0; 0; 1; 2; 3; 4; 4; 4; 4; 4; 3; 2; 1; 0; 0; 0) = 28$

$dm(S1; S2) = |24 - 36| = 12$        $dm(S1; S3) = |24 - 28| = 4$

### Szczegóły implementacyjne – metryka „góry”

#### Funkcja metrykaGóry

Funkcja przyjmuje 2 argumenty będące sekwencjami w notacji kropkowo-nawiasowej. W implementacji również obecna jest struktura stosu. Funkcja zwraca wartość odległości pomiędzy sekwencjami.

Plik [Bioinformatyka\\_projekt\\_wszystkie\\_funkcje.py](#) zawiera pełną funkcjonalność programu, do poszczególnych funkcji została dodana obsługa plików, przyjmują one dodatkowe parametry, jak wejście lub indeks, umożliwiające zapisywanie wyników do wybranych plików oraz identyfikację numeru sekwencji.

W folderze [Testy](#) znajdują się przykładowe testy projekty, natomiast w folderze [Przykładowe\\_wyniki](#) zaprezentowane są 2 różne pliki wynikowe.

#### Bibliografia:

1. GRUBER, ANDREAS R, BERNHART, STEPHAN H, HOFACKER, IVO L, & WASHIETL, STEFAN. (2008). *Strategies for measuring evolutionary conservation of RNA secondary structures*. BioMed Central Ltd. BioMed Central Ltd.  
<http://www.biomedcentral.com/1471-2105/9/122>.
2. HOGEWEG, P, & HESPER, B. (n.d.). *Energy directed folding of RNA sequences*.  
<http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=320984>.
3. KITAGAWA, JUNJI, FUTAMURA, YASUHIRO, & YAMAMOTO, KENJI. (n.d.). *Analysis of the conformational energy landscape of human snRNA with a metric based on tree representation of RNA structures*. Oxford University Press.  
<http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=152804>.
4. SCHIRMER, S. (2012). *Comparing forests*. Bielefeld, Universitätsbibliothek Bielefeld, Hochschulschriften. <http://nbn-resolving.de/urn:nbn:de:hbz:361-24742380>.

