Algoritmy pre segmentáciu biologických sekvencií

Školitel: doc. Mgr. Bronislava Brejová, PhD.
Študent: Dávid Simeunovič

Evolúcia DNA sekvencie

Evolúcia prebieha mutáciou DNA

Mutácie:

- insercia
- delécia
- transpozícia
- inverzia
- duplikácia
- Speciácia

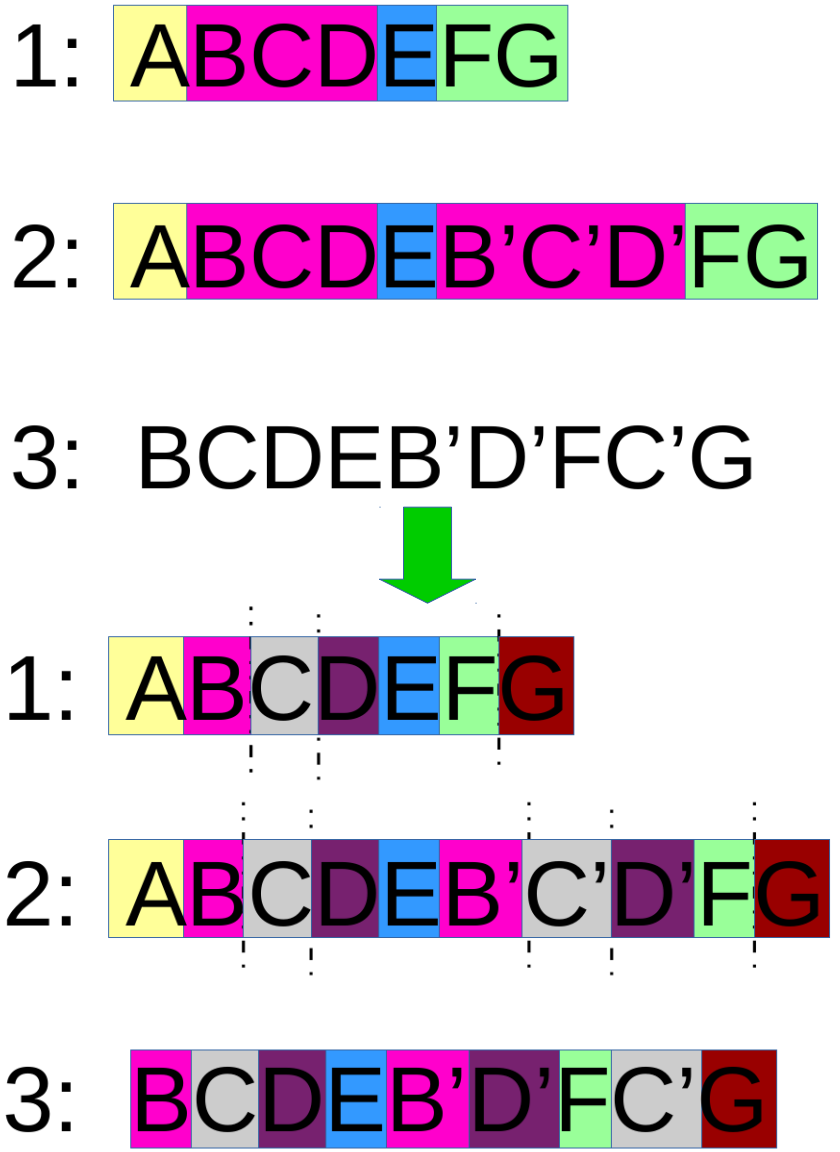
1: ABCDEFG

2: ABCDEB’C’D’FG

3: BCDEB’D’FC’G

Segmentácia

- Rozdelenie DNA na segmenty
  - Zakonzervované úseky – atómy
  - Čo najbližšie skutočnej segmentácii
- Rozdelenie segmentov do tried
  - Klastering
- Vstup pre iné problémy



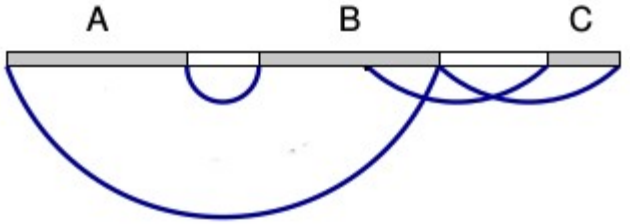
Lokálne zarovnanie

Zarovnanie dvoch sekvencií na základe skórovacej schémy.

vysoké skóre = vysoká podobnosť

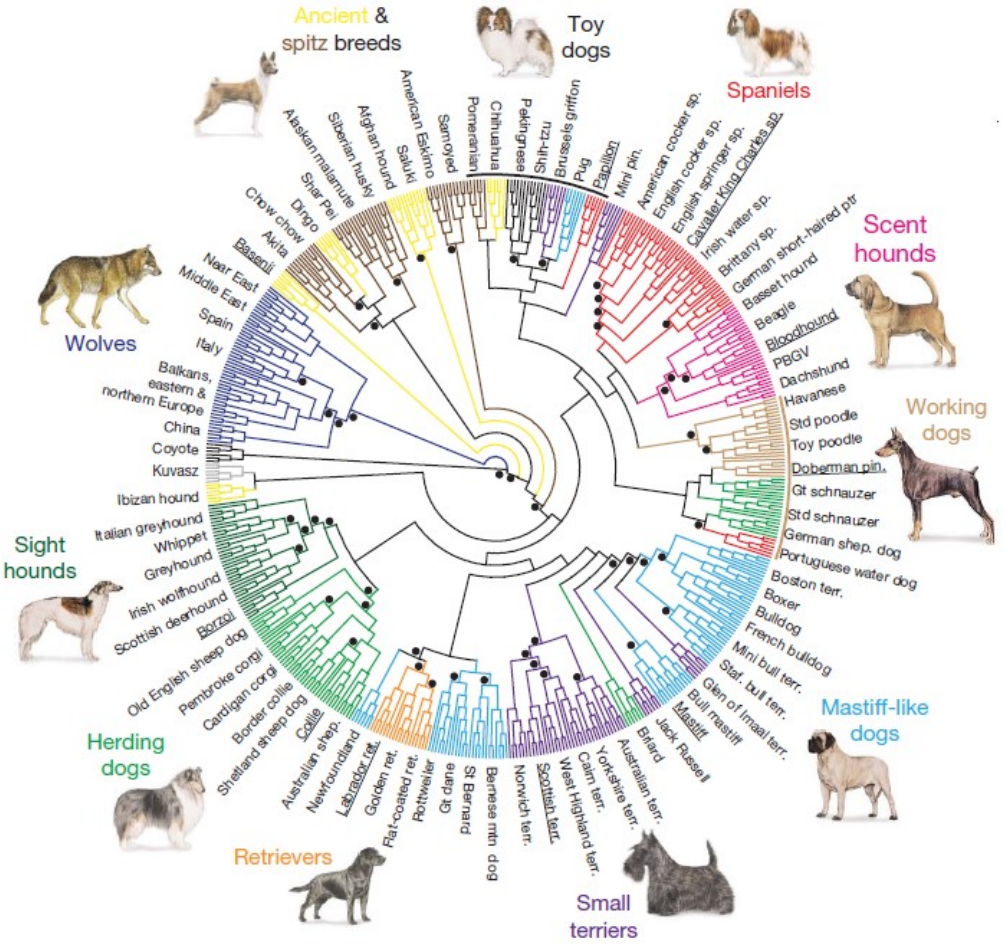
sequence T1 : *A T C - - G C G G A*

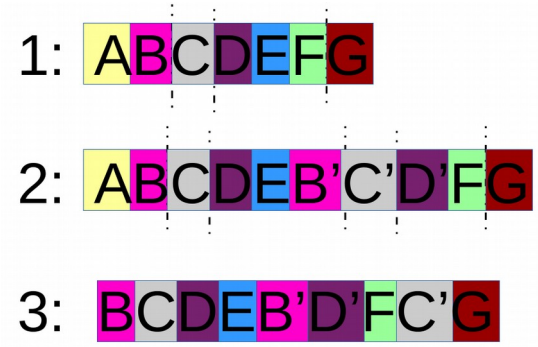
sequence T2 : *A T C A A G T G - A*



Motivácia

- Rekonštrukcia EH:
  - zdĺhavá
  - ťažko pozorovateľná
- DNA pomáha k rekonštrukcii
  - využitie segmentov
- Princíp šetrnosti
- Veľký a malý fylogenetický problém

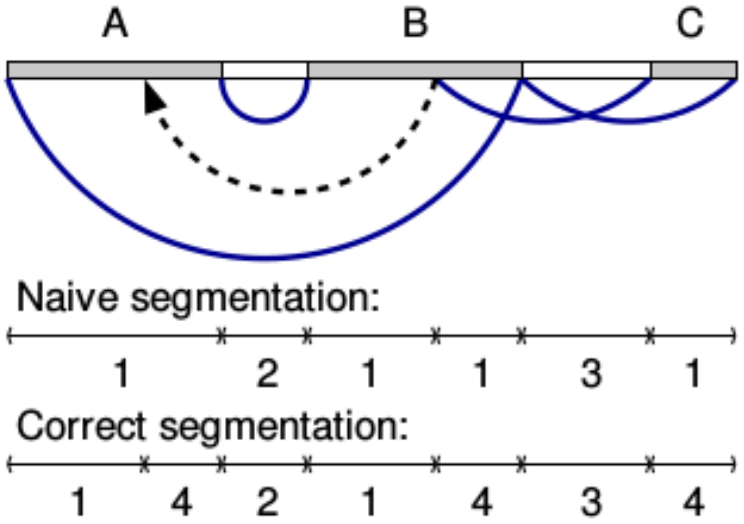




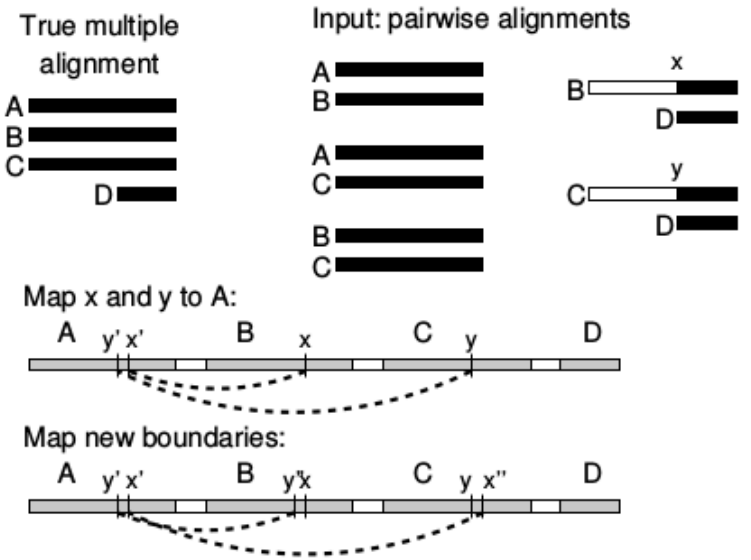
Využitie lokálnej podobnosti

Automated Segmentation of DNA Sequences with Complex Evolutionary Histories

Broňa Brejová, Michal Burger, and Tomáš Vinař [2011]



chýbajúce zarovnania



iteratívna spätná propagácia

Atomizácia s odpadom

DNA Sequence Segmentation Based on Local Similarity

Martina Višňovská, Tomáš Vinař, Broňa Brejová [2013]

- Formálna definícia problému
  - Odpad = neatomizované bázy
- heuristický algoritmus

Vstup S, atómy A a zarovnania α

1) Atómy sa neprekrývajú
2) Minimálna dĺžka atómu
3) Ak sa zdroj alebo cieľ zrovnania α prekrýva s atómom A, potom taktiež pokrýva atóm A.
4) Ak zdroj zarovnania α pokrýva atóm A, potom oblasť α(A) prekrýva 1. atóm.

Cieľ práce

- Priniesť upravenú definíciu problému
  - Lepšie zodpovedajúcu našej intuícii atomizácie
  - Zohľadňujúcu nepresnosti v zarovnaniach
- Zostrojiť algoritmické riešenie
- Porovnať výsledky pre pôvodnú a upravenú verziu

Jadro a okraje

- Jadro a okraj atómu
  - Konce zarovnaní mimo jadra

- Staré:

3) Ak sa zdroj alebo cieľ zrovnania α prekrýva s atómom A, potom taktiež pokrýva atóm A.

4) Ak zdroj zarovnania α pokrýva atóm A, potom oblasť α(A) prekrýva 1. atóm.

- Nové:

3) Ak sa zdroj alebo cieľ zrovnania α prekrýva s jadrom A, potom taktiež pokrýva jadro A.

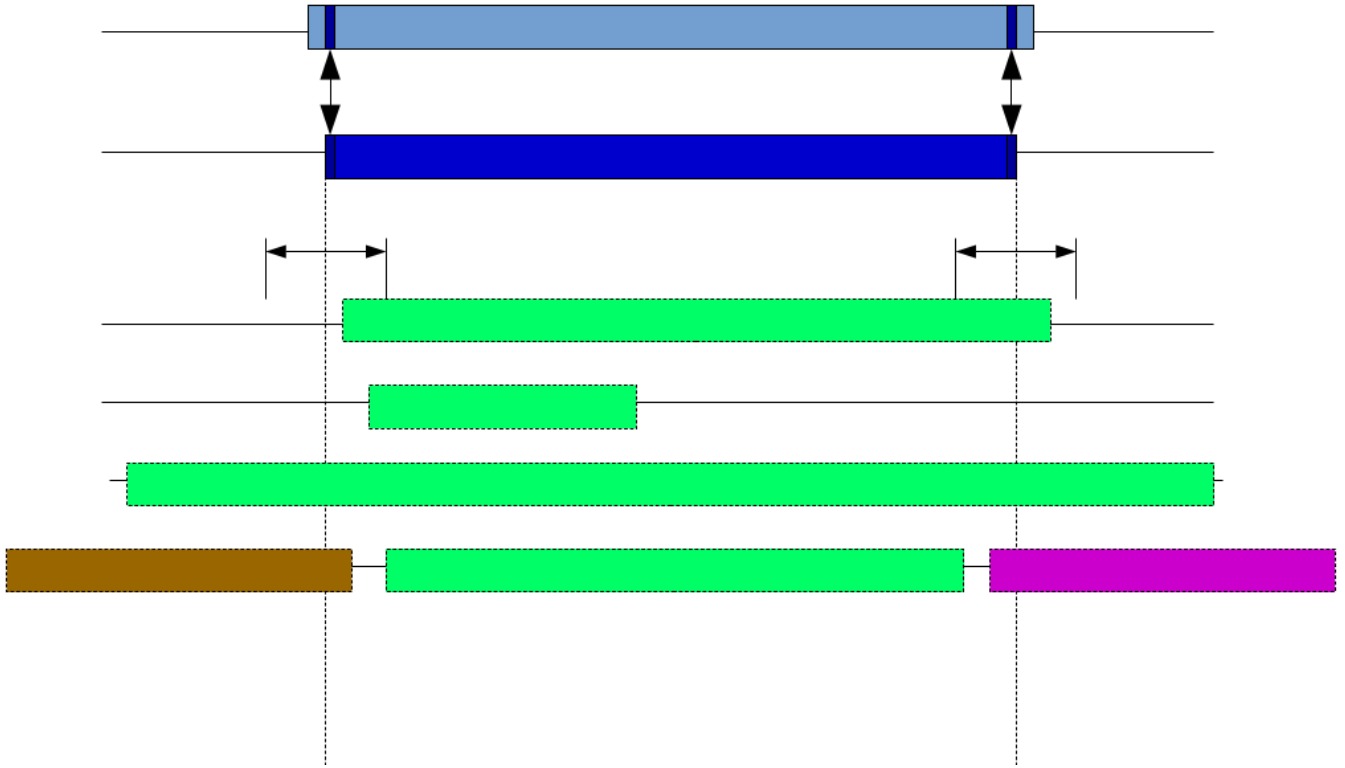
4) Ak zdroj zarovnania α pokrýva jadro A, potom sa oblasť α(A) zhoduje s toleranciou s jedným atómom.

Podmienka č. 4

4) Ak zdroj zarovnania α pokrýva atóm A, potom oblasť α(A) prekrýva 1. atóm.

vs.

4) Ak zdroj zarovnania α pokrýva jadro A, potom sa oblasť α(jadro A) zhoduje s toleranciou s jedným atómom.



Pseudoatomizácia

Skórovacia schéma:

1) Minimalizovať odpad
2) Minimalizovať počet atómov
3) Penalizácia za konce zarovnaní v okrajoch atómov

Algoritmus pre pseudoatomizáciu

- Prvé tri podmienky
- DP a jeho vylepšenia
- $O(n^2(\log(|B|) + d_1))$
- $O(|B| \cdot d_1 \cdot (\log(|B|) + d_1^2))$

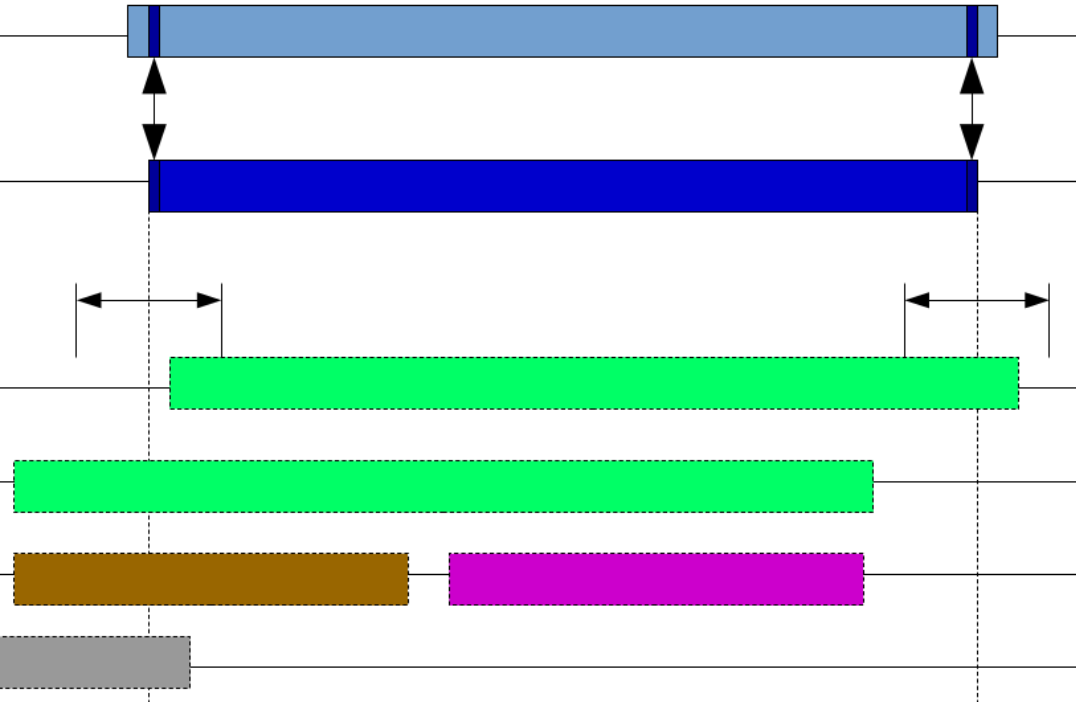
Algorithm 1 Dynamic programming

```
1:  region_type[0] = 1
2:  region_start[0] = 0
3:  subsequence[0] = p_2
4:  for i = 1, 2, ..., n do
5:      for j = 0, 1, 2, ..., i − 1 do
6:          seq_type, seq_penalty = atomize(j, i)
7:          whole_penalty = subsequence[j] + seq_penalty
8:          if whole_penalty < subsequence[i] then
9:              region_type[i] = seq_type
10:             region_start[i] = j
11:             subsequence[i] = whole_penalty
12:         end if
13:     end for
14: end for
```

Atomizácia

Algoritmus pre atomizáciu

- Z pseudoatomizáce atomizácia
- Greedy skracujeme, delíme a mažeme pseudoatómy



Algorithm 2 Atomization

1: We start with a set of p.a. P, alignments α and constant d_2
2: $Q_{process} = \{\}$
3: $Q_{split} = \{\}$
4: **for** each E from P and a from α **do**
5: **if** a source covers E core **then**
6: if $|a(E)| < L - 2 * d_2$ then delete E and continue with next E
7: $E.linked\_alignments$ add a
8: $a'.target\_atoms$ add E
9: $Q_{process}.add(E, a)$
10: **end if**
11: **end for**
12: **while** $len(Q_{process}) + len(Q_{split}) > 0$ **do**
13: **if** $len(Q_{process}) > 0$ **then**
14: E, $alignment \leftarrow Q_{process}.pop()$
15: $tasks = \{alignment\}$
16: **while** $tasks$ not empty **do**
17: $alignment \leftarrow tasks.pop()$
18: $process(E, alignment)$
19: **if** E was trimmed or soft splitted **then**
20: $tasks = E.linked\_alignments$
21: **else** E was deleted
22: $tasks = None$
23: **end if**
24: **end while**
25: If E was modified or deleted then $Notify(E.linked\_alignments)$
26: **else**
27: E, $alignment \leftarrow Q_{split}.pop()$
28: $split(E, alignment, force)$
29: $Notify(E.linked\_alignments)$
30: **end if**
31: **end while**

Výsledky

Porovnanie s IMP a SibeliaZ

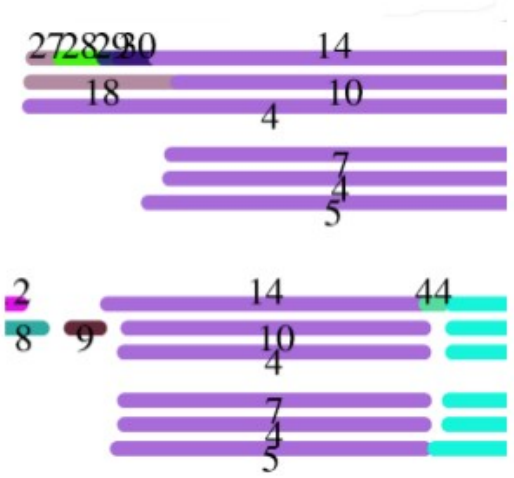
- na simulovaných dátach podľa metrík:
  - BRM (prekryv)
  - BFM (približná zhoda)

Min. atom length		L = 50			L = 500		
algorithm		TRUE	IMP	ACS	TRUE	IMP	ACS
Coverage		100.0%	**99.8%**	98.6%	98.6%	98.5%	**99.0%**
BRM	specificity	100.0%	98.0%	**98.2%**	100.0%	100.0%	100.0%
	sensitivity	97.4%	**95.9%**	94.2%	79.5%	77.2%	**78.7%**
BFM	specificity	100.0%	77.2%	**80.9%**	100.0%	97.6%	**97.9%**
	sensitivity	97.4%	75.6%	**77.6%**	79.5%	75.3%	**77.0%**
Number of atoms		375.2	**377.1**	369.4	306.3	297.2	**303**
Number of classes		111.6	**112**	109.4	97	94.3	**95.2**

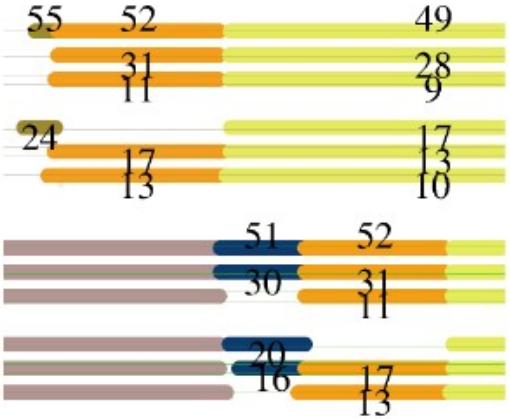
Výsledky

Porovnanie s IMP a SibeliaZ

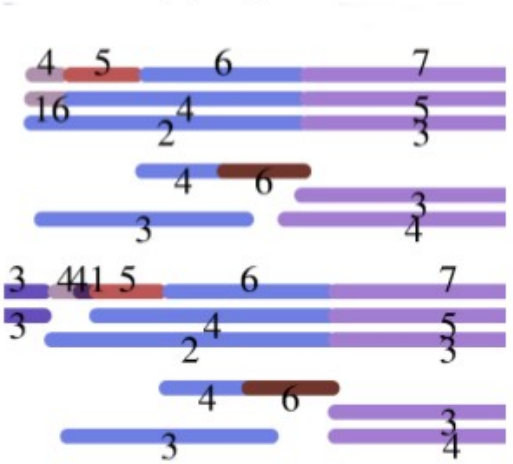
- na reálnych dátach
  - základné metriky
  - rozdiely v atomizáciách



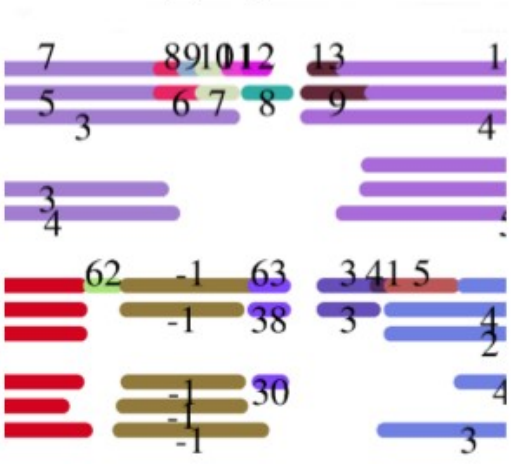
(a) Figure A



(b) Figure B



(c) Figure C



(d) Figure D

Možné vylepšenia

- Delenie pseudoatómu na viacero
  - optimálne
  - viac krát randomizovane
- Definícia
  - Konce zarovnaní v okrajoch atómov
  - Dynamicky škálovať parametre pre dĺžku okrajov a toleranciu zhody podľa dĺžky atómu

Ďakujem za pozornosť