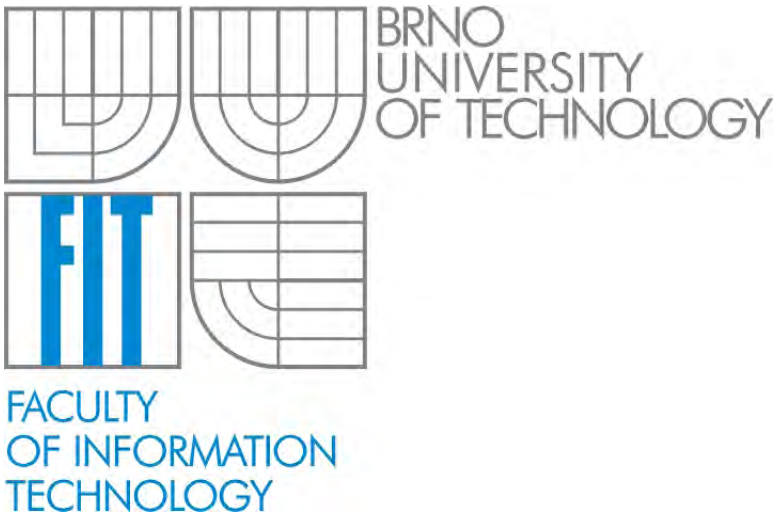


Detekce strukturních celků v proteinových strukturách

Projekt do předmětu Pokročilá bioinformatika, varianta B

Jaroslav Bendl

xbendl00@stud.fit.vutbr.cz

Obsah:

1. Úvod a formulace zadání

Proteiny jsou podstatou všech živých organizmů. Jedná se o anorganické sloučeniny tvořené řetězcem aminokyselin vzájemně propojených peptidovými vazbami. Konkrétní funkce proteinu lze zjistit analýzou jejich struktury. Rozpoznáváme přitom strukturu primární, určenou sekvencí jednotlivých aminokyselin, sekundární, určenou propojením aminokyselin pomocí vodíkových vazeb a terciální, určenou trojrozměrným uspořádáním elementů sekundární struktury.

V rámci této práce je studována analýza terciální struktury proteinu. Cílem je vyvinout program detekující společné strukturní elementy pro zadanou množinu proteinů a následně je spojit do větších strukturních celků. Za strukturní element je přitom považována čtveřice aminokyselin tvořená dvěma páry aminokyselin sousedících na kostře proteinu, přičemž vzdálenost mezi žádnou dvojící C_α uhlíků nesmí být větší než určený počet angstromů. Za strukturní celek je potom považována množina strukturních elementů, ve které se každý překrývá nejméně s jedním dalším právě třemi aminokyselinami a topologie nově vzniklého celku je stejná ve všech analyzovaných proteinech.

2. Popis implementace

V této kapitole jsou postupně představeny metody implementace jednotlivých částí projektu včetně odkazů do zdrojového kódu.

2.1 Zpracování konfiguračního souboru se zadáním úlohy

Program zjišťuje konkrétní zadání ze souboru `configuration.txt`, ve kterém je možné nastavit parametry běhu a určit množinu analyzovaných proteinů. Formát konfiguračního souboru je následující:

`OUTPUT_FILE_1`	Cesta k souboru pro uložení nalezených strukturních elementů.
`OUTPUT_FILE_2`	Cesta k souboru pro uložení nalezených strukturních celků.
`DIST_ANGSTROM`	Maximální vzdálenost (v angstromech) mezi dvěma C_α uhlíky, při které lze příslušné aminokyseliny považovat za sousedící.
`UNIT_COUNT`	Počet největších vzájemně nekolidujících strukturních celků tvořících výsledek.
`SHOW_PROTEIN_ID`	Identifikátor proteinové struktury, na které budou vyznačeny nalezené strukturní celky.
`PROTEIN_LIST`	Označení části konfiguračního souboru uvozující jména vyšetřovaných proteinových struktur. Jméno se zapisuje ve tvaru `XXXX_Y`, kde `XXXX` je identifikátor analyzovaného proteinu v databázi PDB a `Y` je označení konkrétního řetězce.

Program tyto parametry přečte a ověří. Záznamy proteinových struktur z databáze PDB jsou očekávány ve složce `PROTEINS`, pakliže se tam však některý ze záznamů nenachází, program se jej pokusí stáhnout z internetu. Implementace zpracování konfiguračního souboru se nachází v souboru `parameters.py` a `record.py`.

2.2 Identifikace strukturních elementů

Jak již bylo popsánu v první kapitole, jedním ze dvou hlavních úkolů programu je identifikace strukturních elementů. V této etapě program obdrží seznam proteinových struktur a na základě jejich analýzy strukurní elementy nalezne a zapíše je do prvního výstupního souboru. Formát zápisu je pro každý strukturní element následující:

A_1A_2 A_3A_4 XXX1_Y:pos_1:pos_2:pos_3:pos_4 XXX2_Y:pos_1:pos_2:pos_3:pos_4 …

kde A_1 , A_2 , A_3 , A_4 představují dva páry aminokyselin sousedících na kostře, XXX1_Y a XXX2_Y označují analyzované proteiny a za nimi následují pos_1, pos_2, pos_3, pos_4, na kterých se aminokyseliny A_1 , A_2 , A_3 , A_4 v daných proteinových strukturách nacházejí. Proteinové struktury jsou zkoumány na základě znalosti jejich struktury, která je popsána na oficiálních stránkách databáze PDB[1].

Identifikace strukturních celků je implementována v souboru `strElemFinder.py` s využitím `aminoAcidRecord.py` a `strElement.py`. Při hledání se nejprve náhodně vybere jedna z proteinových struktur, pro kterou jsou postupně získány všechny možné dvojice párů sousedících aminokyselin splňující podmínku minimální vzájemné vzdálenosti pro libovolnou z dvojic. Existence těchto elementů je poté ověřována ve všech analyzovaných proteinech, přičemž do výsledku se zahrnou pouze elementy společné pro všechny proteiny.

2.3 Identifikace strukturních celků

Druhým úkolem programu je identifikace strukturních celků. V této etapě program obdrží seznam strukturních elementů a na základě jejich analýzy nalezne největší vzájemně nekolidující strukturní celky a zapíše je do druhého výstupního souboru. Formát zápisu výsledků přitom následující:

	XXX1_Y	XXX2_Y	…
X_1	p_{11}	p_{21}	
X_2	p_{21}	p_{22}	
X_3	p_{31}	p_{32}	
X_4	p_{41}	p_{42}	
X_5	p_{51}	p_{52}	
X_6	p_{61}	p_{62}	
…			

kde X_1-X_6 představují aminokyseliny a p_{11}-p_{61} určují pozice těchto aminokyselin v proteinové struktuře XXX1_Y, obdobně pak pro další proteinové struktury.

Identifikace strukturních celků je implementována v souboru `strElemJoiner.py` s využitím `strUnit.py`. Při hledání se postupuje v iteracích, v nichž je každý strukturní celek podroben pokusu o přidání jednotlivých strukturních elementů. Pakliže je splněna podmínka překrytí právě ve třech aminokyselinách alespoň s jedním strukturním elementem daného celku, je tento celek rozšířen. Zároveň však musí být splněna podmínka stejné topologie ve všech proteinech, přičemž topologií (konektivitou) se rozumí pořadí spojení jednotlivých strukturních elementů peptidovými vazbami. Na závěr se z množiny nalezených strukturních celků vybere daný počet těch největších a vzájemně nekolidujících.

[1] http://www.wwpdb.org/documentation/format32/v3.2.html

3. Experimenty

V této kapitole je představen jeden z pokusů s vytvořeným programem. Tento pokus využívá proteinů `1_LE5_A`, `1MY7_A`, `1OY3_B`, podmínka minimální vzdálenosti je nastavena na `8`, počet největších zobrazovaných strukturních celků je nastaven na `5`.

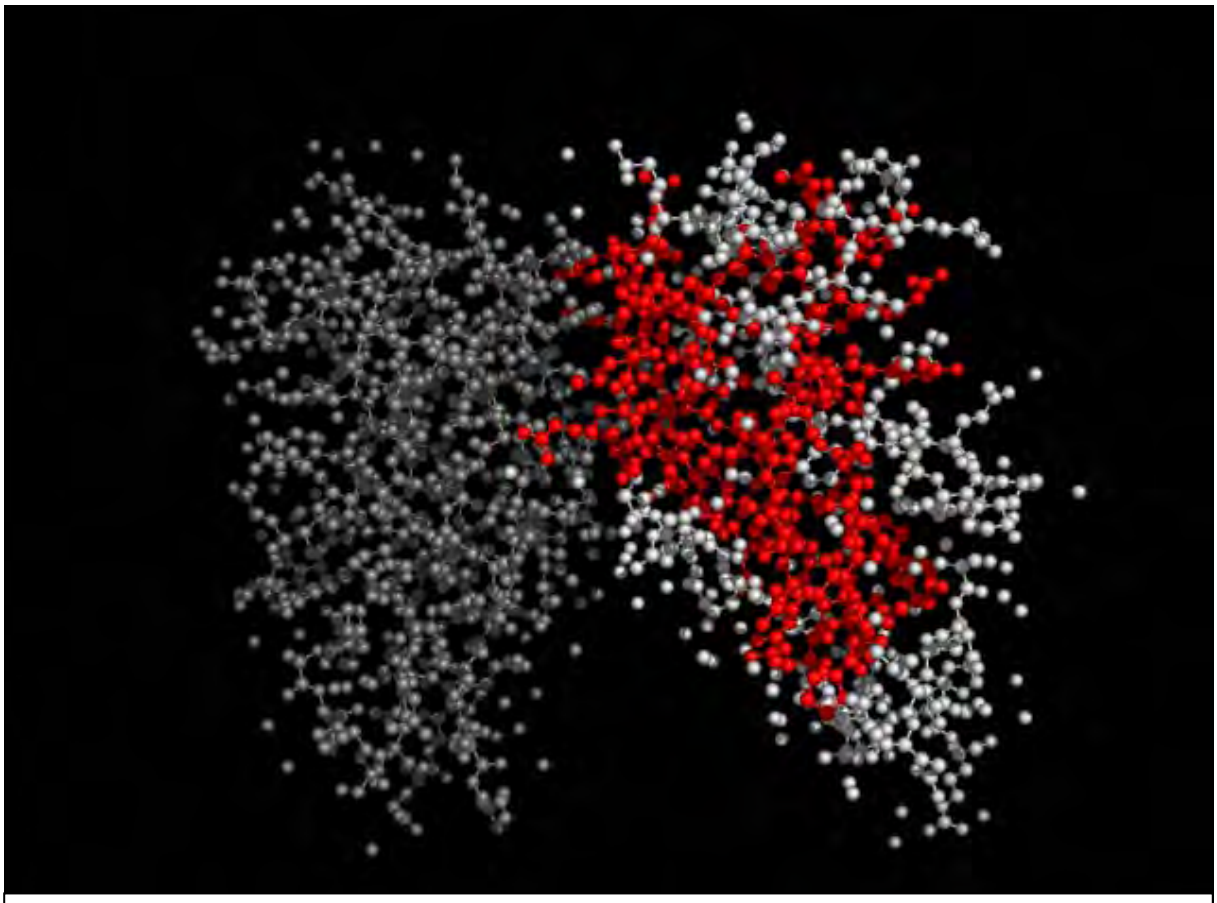
Pro danou úlohu bylo nalezeno 194 strukturních elementů, přičemž 5 největších vzájemně nekolidujících strukturních celků obsahovalo dohromady 87 aminokyselin. Část z prvního i druhého výstupního souboru je na níže uvedených ukázkách č.1 a č.2, kompletní podoba výstupních souborů se pak nachází ve složkách `dokumentace/experiment_vystup2.txt` a `dokumentace/experiment_vystup2.txt`. Na obrázcích č.1, č.2, č.3 jsou ukázány vyznačené strukturní celky pro všechny analyzované proteiny – bílou barvou je zobrazený zkoumaný řetězec proteinu, obdobně šedou barvou nezkoumané řetězce proteinu. Výsledné strukturní celky jsou pak vyznačeny červeně. Na obrázcích č.4, č.5 a č.6 jsou ukázány tytéž proteiny, ale zde jsou jednotlivé strukturní celky odlišeny barvami.

Ukázka č. 1: Výstupní soubor se strukturními elementy

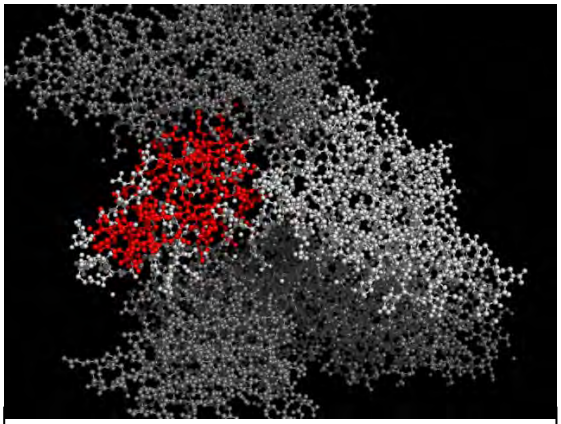
```
GS LG 1LE5_A:44:45:116:117 1MY7_A:204:205:207:208 1OY3_B:204:205:207:208
VR FQ 1LE5_A:72:73:161:162 1MY7_A:266:267:286:287 1OY3_B:266:267:286:287
EL DK 1LE5_A:193:194:217:218 1MY7_A:193:194:217:218 1OY3_B:193:194:217:218
EL LS 1LE5_A:193:194:280:281 1MY7_A:193:194:280:281 1OY3_B:193:194:280:281
LK CD 1LE5_A:194:195:216:217 1MY7_A:194:195:216:217 1OY3_B:194:195:216:217
LK DK 1LE5_A:194:195:217:218 1MY7_A:194:195:217:218 1OY3_B:194:195:217:218
LK QL 1LE5_A:194:195:271:272 1MY7_A:194:195:271:272 1OY3_B:194:195:271:272
LK SE 1LE5_A:194:195:281:282 1MY7_A:194:195:281:282 1OY3_B:194:195:281:282
KI LC 1LE5_A:195:196:215:216 1MY7_A:195:196:215:216 1OY3_B:195:196:215:216
KI CD 1LE5_A:195:196:216:217 1MY7_A:195:196:216:217 1OY3_B:195:196:216:217
KI QL 1LE5_A:195:196:271:272 1MY7_A:195:196:271:272 1OY3_B:195:196:271:272
...
```

Ukázka •. 2: Výstupní soubor se strukturními celky

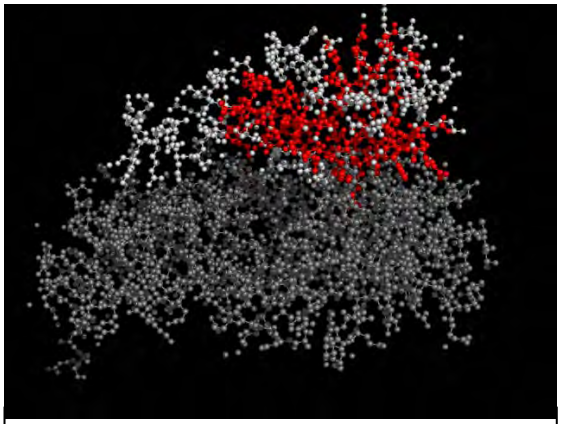
```
1LE5_A 1MY7_A 1OY3_B
P 260 260 260
S 261 261 261
L 262 262 262
Q 263 263 263
A 264 264 264
P 265 265 265
V 266 266 266
R 267 267 267
V 268 268 268
S 269 269 269
M 270 270 270
...
```



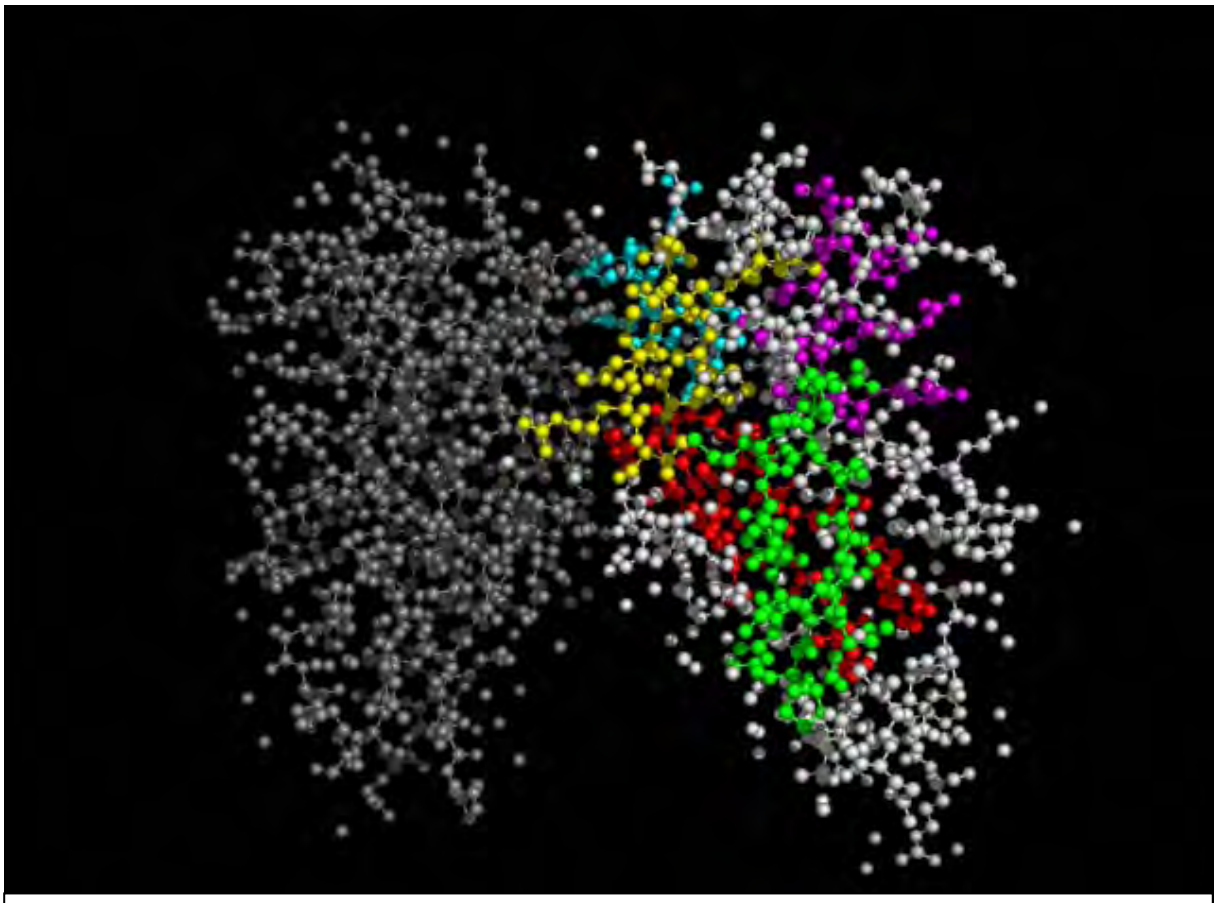
Obrázek č.1: Vyznačené strukturní celky na proteinu 1MY7



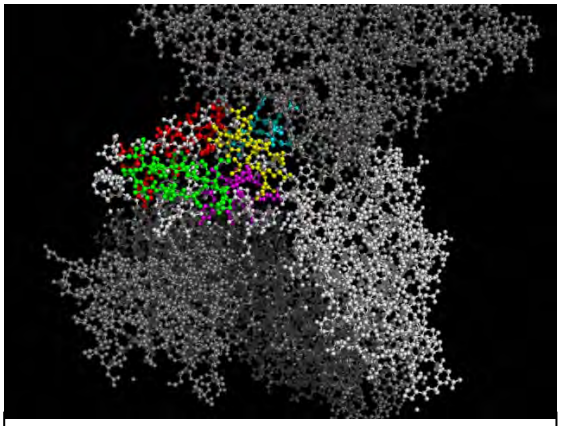
Obrázek č.2: Vyznačené strukturní celky na proteinu 1LE5



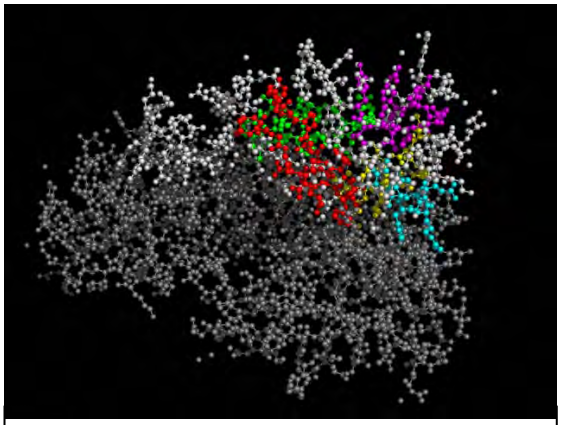
Obrázek č.3: Vyznačené strukturní celky na proteinu 1OY3



Obrázek č.4: Vyznačené strukturní celky na proteinu 1MY7



Obrázek č.5: Vyznačené strukturní celky na proteinu 1LE5



Obrázek č.6: Vyznačené strukturní celky na proteinu 1OY3

4. Závěr

Zvolené řešení klade důraz na dodržení všech požadavků zadání. Program má pečlivě navrženou objektově-orientovanou strukturu a množství přehledných komentářů, tudíž je možné jej snadno modifikovat a rozšiřovat. Testování a ladění proběhlo v prostředí operačního systému Linux s nainstalovaným programem Pymol a podporou jazyka Python.