Shlukování aminokyselin v proteinu

Petr Dvořáček – xdvora0n@stud.fit.vutbr.cz

5. ledna 2015

1 Úvod

Tento dokument popisuje návrh a implementaci projektu do předmětu Pokročilá bioinformatika. Cílem projektu bylo pomocí sady libovolných parametrů provést grafové shlukování aminokyselin v proteinu. Výsledky se pak mají zapsat do PDB souboru, kde každý shluk má tvořit vlastní skupinu. Pomocí programu PyMOL pak máme vytvořit vlastní vizualizaci výsledku. Jako programovací jazyk byl určen Python.

2 Návrh algoritmu

Jelikož pracujeme s proteinem v prostoru, potom by bylo vhodné použít jako vstup PDB soubor, kde jsou definované pozice jednotlivých atomů v proteinu. Nevýhodou však je, že potřebujeme určit pozici celé aminokyseliny. Situaci může zkomplikovat fakt, skládá-li se soubor PDB pouze z řádků začínající ATOM či HETATM. Potom je potřeba nalézt i celý řetězec proteinu.

Pro shlukování zvolíme několik dimenzí. První z nich představuje vzdálenost mezi dvěma AC, což jest vektor tří prvků (x, y a z). Druhá z nich je vzdálenost (rovněž vektor tří prvků) od zvoleného bodu a tou poslední je chemciká příbuznost jednotlivých aminokyselin. Tuto složku je těžké určit, proto pro vyhledání interakce mezi AC byla zvolena převedená hodnota mezi dvěma sidechains, která je dostupná v literatuře [1]. Samotný převod jsem provedl parsrováním zmíněných stránek, čímž jsem získal tabulku, která je dostupná ve zdrojovém kódu. Podle ní se pak určí váha dané vzdálenosti.

Problémem může být malá vzdálenost mezi sousedními aminokyselinami v řetězci, např. v $FITMA$ je T blízko M a I. Vzdálenosti u těchto AC můžeme naváhovat nebo lépe přidělit jim neutrální hodnoty.

Pro samotné shlukování dat jsem použil algoritmus DBSCAN [2], který zahrnuje krom vzdálenosti i hustotu daného shluku.

Algoritmus pro shlukování aminokyselin v proteinu lze shrnout v těchto krocích

1. Parsrovaní PDB souboru a parsrování jednotlivých atomů.
2. Vytvoření řetězce aminokyselin z PDB záznamu atomů.
3. Vytvoření pozic aminokyseliny z jejich atomů podle
$$\forall atom \in aminoacid : \frac{\sum_{atom}(atom_x, atom_y, atom_z)}{|aminoacid|}$$
což je vlastně průměrná pozice všech atomů jedné aminokyseliny z řetězce proteinu.
4. Z pozic jednotlivých aminokyselin se vytvoří vzdálenostní matice pomocí vzdáleností n-dimenzionálního prostoru v Euklidovské metrice podle vzorce:
$$distance = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$
.
5. Vzdálenosti mezi sousedními aminokyselinami se naváhují tak, aby byly vůči shlukování neutrální. Např. z řetězce $FITMA$ seberu aminokyselinu F. Sousední aminokyseliny jsou všechny takové, kde vzdálenosti od F roste. Nechť tedy máme vzdálenosti $||FI|| = 1$, $||FT|| = 2$, $||FM|| = 3$, $||FA|| = 2$, potom aminokyseliny ITM jsou sousedé k F. Aminokyselina A není sousedem k F.
6. Provede se shlukování podle algoritmu DBSCAN [2] a shluky se zapíší do PDB souboru obsahující jenom položky ATOM.

3 Použití skriptu

Podle navrženého algoritmu jsem implementoval skript v Pythonu. Jeho použití je následující:

```
python2 mol_cluster.py filename [-f filename] [-p double double double]
        [-k int] [-l double] [-tq] [-e double]
```

Kde `filename` značí název souboru. Parametry v hranatých závorkách jsou volitelné. K správnému fungování je vyžadováno spustit program právě s tímto parametrem značící název PDB souboru, podle kterého se provede shlukování podle vzdáleností jednotlivých AC.

Uživatel může zvolit bod přidáním parametru `-p`, který přijímá další tři parametry `x y z` značící pozici bodu (v reálné hodnotě) podle kterého se určí vzdálenost. Nebyl-li zvolen tento parametr, potom vzdálenost od tohoto bodu nehraje žádnou roli, neboť nebyl definován.

`-l` rovněž přijímá argument vzdálenosti, pro kterou je daný shluk validní. Implicitně 10.

`-k` odpovídá minimálnímu počtu aminokyselin určující hustotu v shluku, viz [2]. Implicitně 0.

`-m` je minimální počet aminokyselin, které mají být v jednom shluku. Implicitně 10.

`-t` byl-li zvolen tento parametr, použije se tabulka chemické vzdálenosti. Implicitně se nepoužije.[1]

`-q` značí běh skritpu v tichém módu, kdy se na standardní výstup nevypisují informace u shlucích.

`-f` umožní vybrat výstupní PDB soubor.Tento soubor pak můžete nahrát v PyMolu. Implicitně `output.pdb`.

4 Experimenty

Pro experimenty byl zvolen protein anti HIV proteáza, jenž má zkratku 1mf2. Viz obrázek 1 vpravo nahoře.

Pro první experiment byla zvolena vzdálenost l byla zvolená na 20.0 a počet prvků ve shluku m na 30. Celkem bylo nalezeno 7 shluků včetně šumu. Výsledek můžete vidět na obrázku 1 vlevo nahoře, kde je zahrnut i šum (bílé atomy), nebo vlevo dole, kde není zahrnut.

Druhým experimentem bylo zahrnutí chemické tabulky vzdáleností viz výsledek vpravo dole na obrázku 1.

V obou případech můžeme provést porovnáním referenčním proteinem (vpravo nahoře). Vidíme, že shluky obsahují většinou aminokyseliny, které tvoří alfa-šroubovice a beta-sheety.

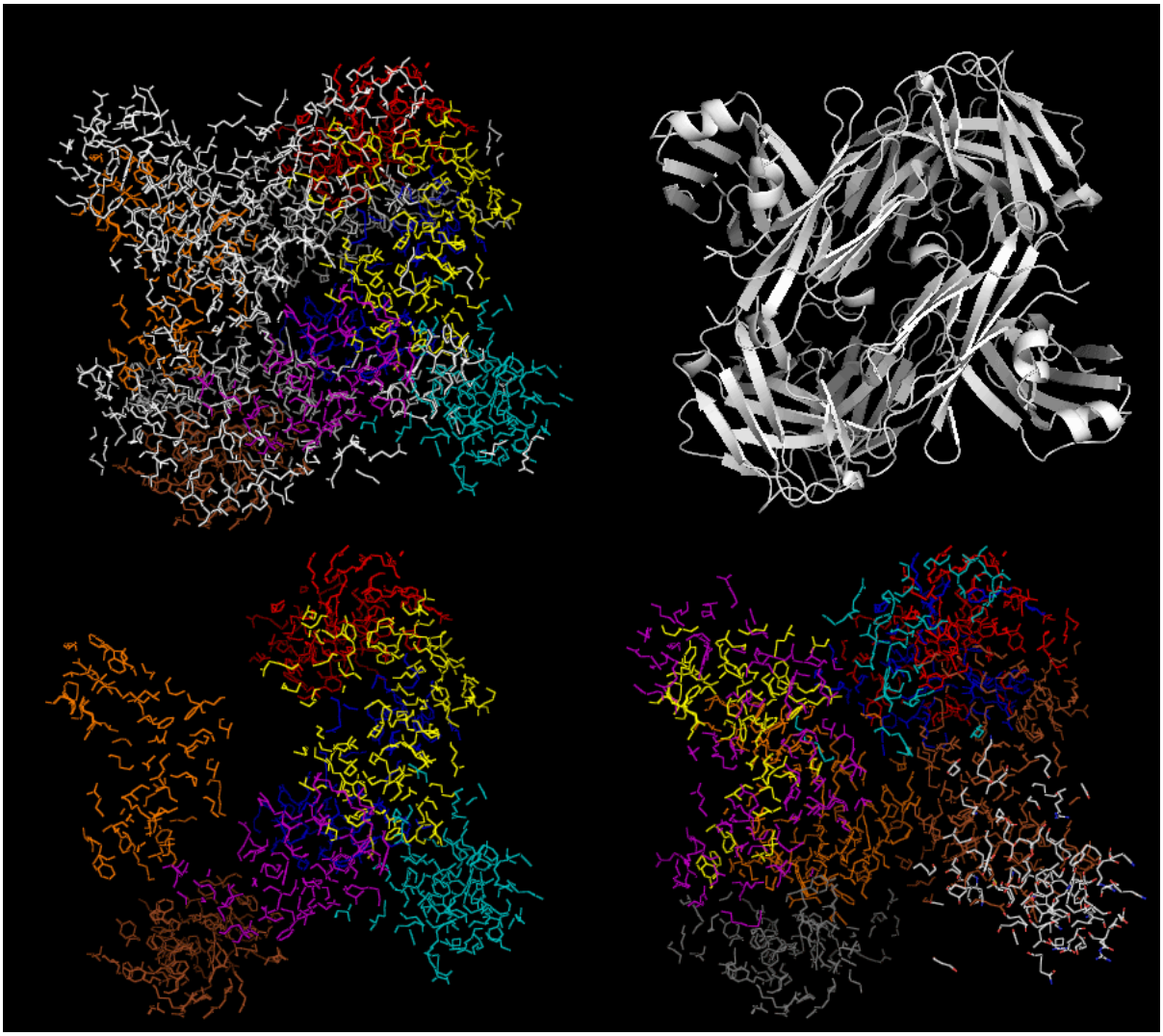
5 Závěr

Byl vytvořen skript, který umožňuje přijímat několika-dimenzionální parametry (délkavzdálenost, chemická příbuznost, vzdálenost od bodu). Mimo to přijímá parametry k shlukování (minimální počet aminokyselin v shluku, hustota shluku). V tomto dokumentu bylo představeno použití algoritmu a výsledky prezentovány na experimentech, z nichž vyplynulo, že shluky se tvoří na místech, kde je větší koncentrace alfa-šroubovic a beta-listů.

6 Literatura

1. `http://www.biochem.ucl.ac.uk/bsm/sidechains/`
2. `http://en.wikipedia.org/wiki/DBSCAN`
   Algoritmus na Wikipedii k dnešnímu datu koresponduje s algoritmem, jenž byl představen v kurzu Získávání znalostí z Databází.

[1] Změnu chemických vzdáleností jednotlivých aminokyselin provádějte v daném scriptu.



Obrázek 1: Výsledné shluky.