Uniwersytet Warszawski
Bioinformatyka i Biologia Systemów
Projektowanie leków
Semestr zimowy 2019/2020

Podobieństwa kompleksów białko-ligand w bazie PDBBind (CoreSet 2016)

Wstęp

CoreSet bazy PDBBind używany jest do kalibracji funkcji oceny, które to pozwalają oceniać jak silnie dana cząsteczka potrafi wiązać się z białkiem. Owe funkcje oceny pełnią więc bardzo ważną rolę przy odkrywaniu i projektowaniu nowych leków i każdy niewielki mankament w takiej funkcji, może powodować ogromne straty. W przypadku jeśli opierając się na ocenie z takiej funkcji, koncern farmaceutyczny zdecyduje się na zaawansowane badania danej cząsteczki, poświęcając czas i środki, gdzie w przypadku błędnej oceny, okazuje się potem, że dana cząsteczka słabo nadaje się jako inhibitor danego receptora.

Bardzo ważne jest więc aby owa baza kompleksów białko-ligand składała się możliwie z jak najbardziej unikalnych danych. Ponieważ jeśli składałaby się z danych bardzo do siebie podobnych, to błąd w ocenie takiego klastra kompleksów wpływałby znacząco na kalibrację funkcji oceny. W naszym więc interesie leży, aby w takiej bazie znajdować podobne cząsteczki i je usuwać.

Metodologia

Kroki wykonane w tym projekcie polegały na:

1. Pobranie bazy PDBBind CoreSet 2016
2. Wyekstrahowanie ligandów w formacie SMILES
3. Znalezienie najbardziej podobnych cząsteczek i klastrów
4. Zadokowanie podobnych ligandów w ‘nie swoich’ receptorach
5. Analiza wyników

Do problemu wyszukiwania podobieństw w takich zbiorach danych podszedłem na 3 różne sposoby, dając użytkownikowi narzędzia do tego, aby w wygodny dla siebie sposób mógł przeprowadzać takie analizy.

Pierwszą funkcjonalnością jest tworzenie wykresu na podstawie macierzy podobieństwa, który dzięki kolorom obrazuje podobieństwa cząsteczek w sposób “każdy z każdym”. Możliwe jest tu ustalanie thresholdów, wyświetlając tylko najbardziej podobne cząsteczki.

Kolejną funkcjonalnością jest tworzenie grafu, a w zasadzie rysowanie jedynie jego wierzchołków, gdzie odległości między nimi odzwierciedlają ich podobieństwo - cząsteczki bardziej podobne będą miały swoje wierzchołki bliżej siebie. Pozwala to na szybką ocena ilości i wielkości klastrów podobnych ligandów, ale także sprawdzenie ich odległości w bazie danych dzięki kolorom, które tutaj odzwierciedlają ich kolejność w zbiorze - różnokolorowe zbiory będą wskazywały na klastrowanie cząsteczek pozornie do siebie nie podobnych, bo z różnych obszarów danych.

Kolejną funkcjonalnością jest sprawdzanie scaffoldów cząsteczek, na podstawie których możemy szacować ich podobieństwo. Owa metoda może sprawdzać się przy szacowaniu większych zbiorów danych, gdzie umożliwi od razu wychwycenie grup, w których mogą znajdować się klastry podobnych cząsteczek.

Ostatnią funkcjonalnością jest dokowanie cząsteczek podobnych do siebie w innych receptorach, czyli mając podobne cząsteczki 1, 2, 3, funkcjonalność umożliwia zbadanie jakie będą wyniki dokowania dla kompleksu cząsteczka-białko w ramach “każdy z każdym”. Pozwoli to na sprawdzenie czy podobne cząsteczki mogą być takimi samymi, czy może nawet lepszymi inhibitorami niż oryginalne.

Program

Przede wszystkim zależało mi na możliwości ponownego wykorzystania tego narzędzia więc cały program tworzony był bez sztywnych założeń i ograniczeń co do danych. Do podstawowych funkcjonalności wymagana jest jedynie jakakolwiek baza danych smiles. Dodatkowo można uruchamiać każdą z funkcjonalności osobno.

Program jest bogato skomentowany, zarówno w postaci komentarzy w kodzie jak i plików README. Składa się on z 2 skryptów python’owych oraz 1 skryptu bash’owego.

Skrypt **redun.py** umożliwia skorzystanie z pierwszych trzech funkcjonalności (wykres podobieństwa, graf odległości skorelowanych z podobieństwem, scaffolding).

Skrypt **dock.sh** umożliwia dokowanie cząsteczek, których wzory pobierane są na podstawie indeksu z bazy cząsteczek w postaci smiles, a potem dokowanie ich w receptorach cząsteczek do nich podobnych.

Skrypt **scorun.py** umożliwia użycie funkcjonalności dokowania w szerszym kontekście, badając zależności “każdy z każdym” z listy przekazanych mu ligandów.

Kod źródłowy programu dostępny jest na platformie github, pod linkiem: https://github.com/moozeq/DD_Redun

Problemy

Pierwszy problem napotkany przy pracowaniu na danej bazie znajdował się w kompleksie **4mme**, którego ligand po przerobieniu do formatu SMILES powodował błąd przy tworzeniu fingerprintu:

```
[23:42:54] Explicit valence for atom # 16 N, 4, is greater than permitted
[23:42:54] ERROR: Could not sanitize molecule on line 0
[23:42:54] ERROR: Explicit valence for atom # 16 N, 4, is greater than permitted
```

Po usunięciu danego ligandu, tworzenie fingerprintów wykonało się prawidłowo.

Kolejnym problemem napotkanym w danej bazie, były cząsteczki, z których nie można było wydobyć scaffoldów programem strip-it. Problem prawdopodobnie leży w samym programie, jednak z racji ograniczeń czasowych nie wgłębiałem się w jego kod źródłowy. Zamiast tego zdecydowałem, żeby mój program do tworzenia scaffoldów był przygotowany na takie sytuacje, wykrywał je i próbował naprawiać zadany zestaw danych. Udało się to i program potrafi wykrywać kiedy nie powiodło się scaffoldowanie zestawu cząsteczek i usuwać z niego cząsteczki, które powodują błędy tak długo, aż program strip-it zwróci kod poprawnego wykonania (ligandy powodujące błędy zapisuje do pliku z końcówką *_wrong_ligands* do późniejszego wglądu w razie potrzeby). Owe poprawki spowodowały wyrzucenie z zestawu do scaffoldowania (ale nie z oryginalnego zestawu branego pod uwagę przy tworzeniu wykresów) parunastu ligandów.

Analiza

OGÓLNA ANALIZA

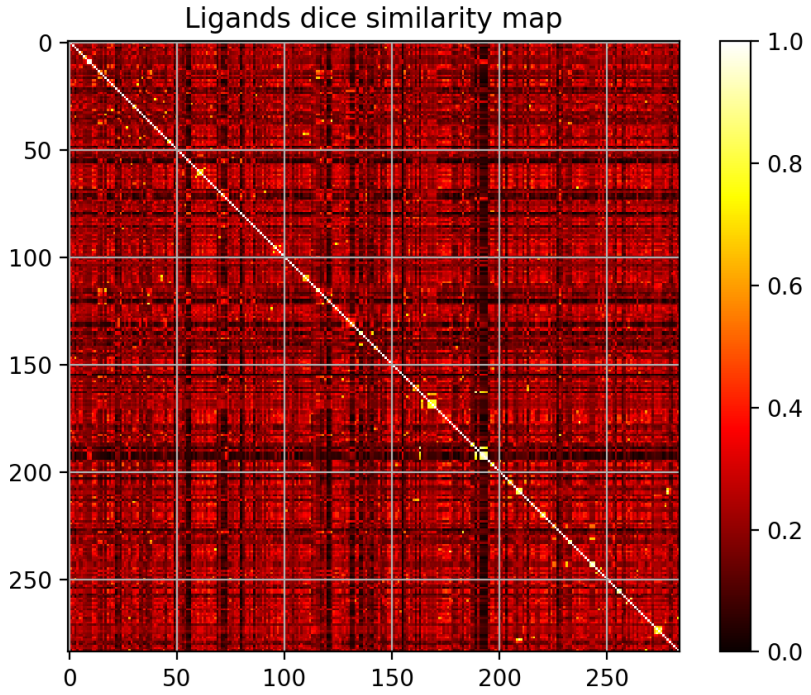
parametry:

dice similarity, threshold = 0.0

komenda:

./redun.py db.smi

Z wykresów bez thresholdu, zobaczyć można ogólne prawidłowości zbioru danych:



WYKRES LIGANDÓW Z PODOBIEŃSTWEM OKREŚLONYM KOLOREM, BEZ OBCIĘCIA

Na powyższym wykresie, gdzie podobieństwo wizualizowane jest kolorami, można zauważyć niewielkie klastry podobnych do siebie cząsteczek, zazwyczaj bardzo blisko siebie. Ogólnie im cząsteczki są od siebie dalej, tym ich podobieństwo jest mniejsze, choć zdarzają się wyjątki, np.:

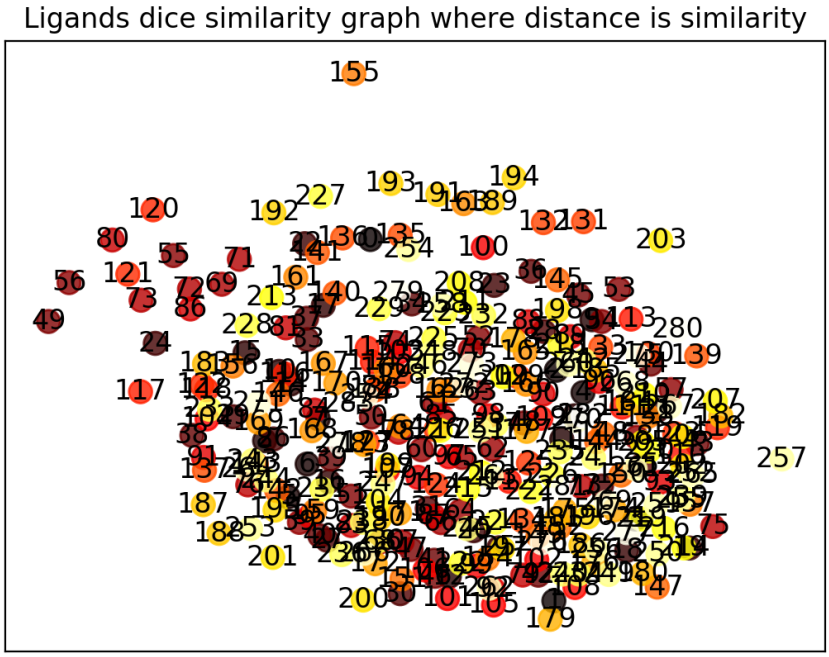
```
1.0000    [68]:    2vkm c1ccccc1[C@@H](C)NC(=O)c1cc(cc(c1)C(=O)N[C@@H](Cc1cc…
0.7586    [236]:   4gid CC(C)CNC(=O)[C@H]([C@@H](C)O)[NH2+]C[C@H](Cc1cc…
```

Gdzie widzimy już po wzorach SMILES, że są to cząsteczki mające podobne do siebie sekwencje, choć w zbiorze znajdujące się dość daleko od siebie.

Z ogólnych danych zauważyć można przede wszystkim, że istnieją niewielkie klastry cząsteczek praktycznie identycznych, zgrupowane bardzo blisko siebie.

Można także zauważyć, że jeden z największych klastrów (189, 191-194) choć są bardzo podobne do siebie, to w porównaniu do całej reszty bazy, są praktycznie oryginalne (czarne pasy świadczące o niskim podobieństwie ciągnące się wzdłuż i wszerz całej mapy), z wyjątkiem jednego ligandu - 163, który także należy do owego klastra, co wykażę w dalszej części raportu.

Stworzony został także graf, który próbuje dopasować odległości między wierzchołkami, tak aby odpowiadały ich podobieństwu. Z ogólnego grafu niestety niewiele można wywnioskować:



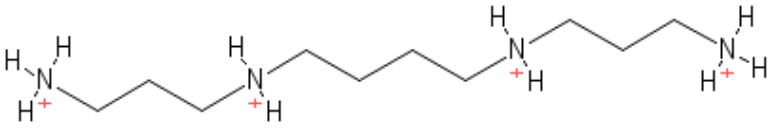
WYKRES LIGANDÓW Z PODOBIEŃSTWEM OKREŚLONYM ODLEGŁOŚCIĄ POMIĘDZY WIERZCHOŁKAMI, BEZ OBCIĘCIA

Większość cząsteczek zagęszczona jest w środku, co świadczy o ich wspólnym podobieństwie do siebie. Możemy za to zauważyć pewne cząsteczki na obrzeżach wykresu, które świadczą o ich oryginalności względem całej reszty.

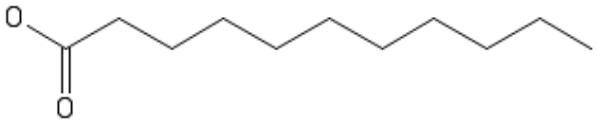
Jedną z grup takich cząsteczek jest właśnie poprzednio wspomniany klaster (163, 189, 191-194), który jest bardzo dobrze widoczny na obrzeżach wykresu, w jego górnej części. Można tutaj także zauważyć inne klastry, które cechuje niskie podobieństwo do całej reszty zbioru: (69, 71-73), (55-56), (164, 167-170) oraz jedną z bardzo ciekawych cząsteczek (155), na samej górze wykresu, którą cechuje praktycznie zerowe podobieństwo do większości cząsteczek i niewielkie do klastra cząsteczek wspomnianego wyżej (163, 189, 191-194).

```
1.0000    [155]:    3kwa [NH3+]CCC[NH2+]CCCC[NH2+]CCC[NH3+]
0.3544    [191]:    3ueu O=C(O)CCCCCCCCCCC
0.3294    [192]:    3uev C(=O)(O)CCCCCCCCCCCCC
0.3288    [189]:    3u9q C(=O)(CCCCCCCCC)O
0.3077    [193]:    3uew C(=O)(O)CCCCCCCCCCCCCCC
0.2985    [163]:    3nq9 C(=O)(CCCCCCC)O
0.2887    [194]:    3uex C(=O)(O)CCCCCCCCCCCCCCCCC
```

Po spojrzeniu na wzoru smiles, można zauważyć powód ich podobieństwa. Są to cząsteczki posiadające długie łańcuchy węglowe, bez żadnych pierścieni, zaś różnice między (155) a klastrem polegają przede wszystkim na przetykaniu łańcucha azotem, do którego dołączone są wodory z jednej strony naładowane dodatnio.



REPREZENTACJA WZORU CZĄSTECZKI NR 155 = 3KWA

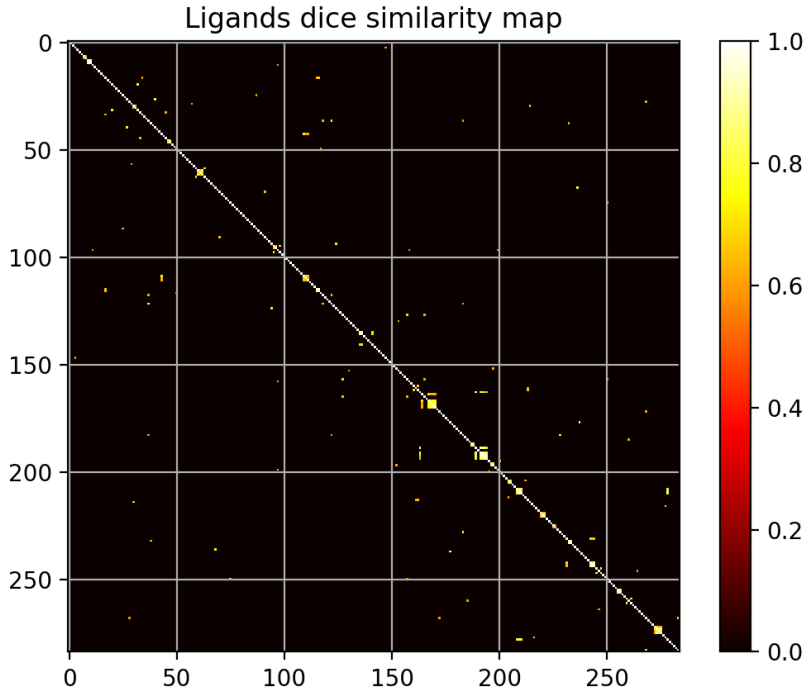


REPREZENTACJA WZORU CZĄSTECZKI NR 191 = 3UEU

ANALIZA Z OBCIĘCIEM

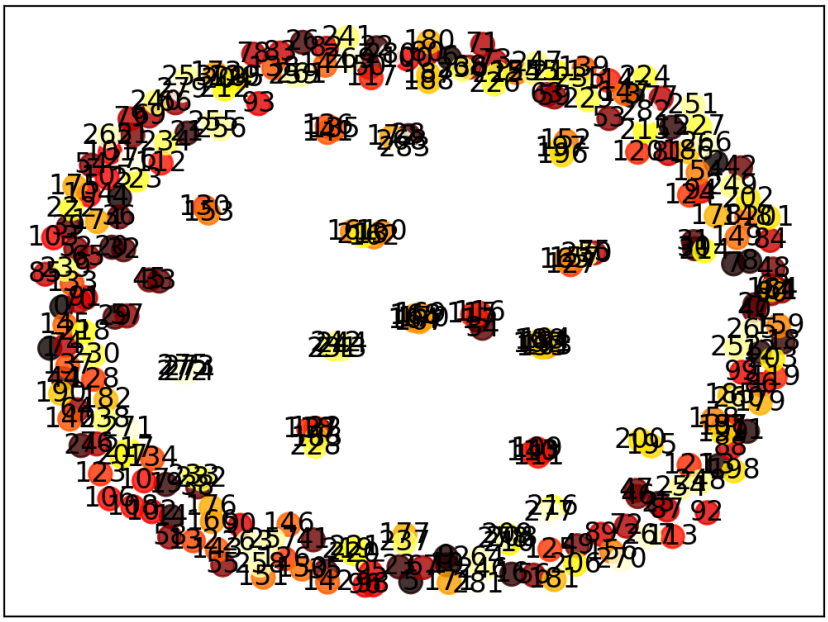
parametry:
dice similarity, threshold = 0.6
komenda:
./redun.py db.smi -t 0.6

Przy obcięciu danych thresholdem na poziomie 0.6, widzimy wyraźnie, zauważone w ogólnej analizie klastry podobnych cząsteczek, ale widzimy również pojedyncze cząsteczki podobne do innych, leżące czasami wręcz po drugiej stronie wykresu (wcześniej wspomniane (68) i (236)):



WYKRES LIGANDÓW Z PODOBIEŃSTWEM OKREŚLONYM KOLOREM, Z OBCIĘCIEM T = 0.6

Więcej możemy teraz ujrzeć na drugim rodzaju wykresu, czyli grafie (bez krawędzi), gdzie odległości między wierzchołkami mają odzwierciedlenie w ich podobieństwie.



WYKRES LIGANDÓW Z PODOBIEŃSTWEM OKREŚLONYM ODLEGŁOŚCIĄ POMIĘDZY WIERZCHOŁKAMI, Z OBCIĘCIEM T = 0.6

Można tutaj zauważyć pogrupowane wspólnie cząsteczki (oczywiście po przybliżeniu w interaktywnej wersji), które uwidaczniają nam istniejące klastry, które już wcześniej mogliśmy zauważyć:

(164, 167-170), (163, 189, 191-194)

Ale także mniej oczywiste, które zawierają w sobie cząsteczki znacznie od siebie oddalone w bazie (można wnioskować o klastrze po kolorze wierzchołków, ciemne kolory oznaczają cząsteczki z niskim indeksem, im jaśniejszy tym cząsteczka ma wyższy index):

(37, 118, 122, 183, 228), (75, 127, 157, 165, 250), (17, 34, 115-116), (75, 127, 157, 165, 250)

Już na tym etapie więc możemy wnioskować o istnieniu pewnych podobieństw pomiędzy ligandami w tej bazie. Po pierwsze istnieją klastry, leżących blisko siebie w bazie cząsteczek, które są do siebie bardzo podobne. Po drugie istnieją klastry łączące nawet daleko leżące w bazie cząsteczki, które zawierają dość znacząco podobne do siebie ligandy.

TANIMOTO

Przed przejściem do kolejnego etapu, warto spojrzeć na podobieństwa uzyskiwane metodą Tanimoto. Spójrzmy na wspomniany wcześniej największy klaster podobnych do siebie ligandów:

```
[163]:    3nq9 C(=O)(CCCCCCC)O
[189]:    3u9q C(=O)(CCCCCCCCC)O
[191]:    3ueu O=C(O)CCCCCCCCCC
[192]:    3uev C(=O)(O)CCCCCCCCCCCC
[193]:    3uew C(=O)(O)CCCCCCCCCCCCCC
[194]:    3uex C(=O)(O)CCCCCCCCCCCCCCCC
```

Tak jak można się spodziewać, bardzo duże podobieństwa są pomiędzy kolejnymi cząsteczkami:

(163) - (189): dice = 0.900, tanimoto = 1.0

(189) - (190): dice = 0.917, tanimoto = 1.0
(191) - (192): dice = 0.929, tanimoto = 1.0
(192) - (193): dice = 0.938, tanimoto = 1.0
(193) - (194): dice = 0.944, tanimoto = 1.0

Można zauważyć, że wraz z wydłużaniem łańcucha, kolejne cząsteczki są coraz bardziej do siebie podobne (ale tylko przy wskaźniku dice, Tanimoto pozostaje niezmienny), można więc podejrzewać, że ich właściwości fizykochemiczne będą bardziej podobne przy dłuższych łańcuchach, ponieważ ich względna różnica będzie mniejsza.

Zaś jeśli spojrzymy na podobieństwa obliczone metodą dice między (163) a resztą klastra, jego podobieństwa są wysokie do cząsteczek (189) i (191), ale im dłuższy łańcuch ma cząsteczka w klastrze, tym mniej podobna jest do tej o najkrótszym łańcuchu (163):

```
1.0000   [163]:   3nq9 C(=O)(CCCCCCC)O
0.9000   [189]:   3u9q C(=O)(CCCCCCCCC)O
0.8182   [191]:   3ueu O=C(O)CCCCCCCCCC
0.7500   [192]:   3uev C(=O)(O)CCCCCCCCCCCC
0.6923   [193]:   3uew C(=O)(O)CCCCCCCCCCCCCC
0.6429   [194]:   3uex C(=O)(O)CCCCCCCCCCCCCCCC
```

W porównaniu do metody oceny podobieństwa Tanimoto, metoda dice jest o wiele dokładniejsza. Przy metodzie Tanimoto, powyższe podobieństwa są po prostu jednością dla całego klastra:

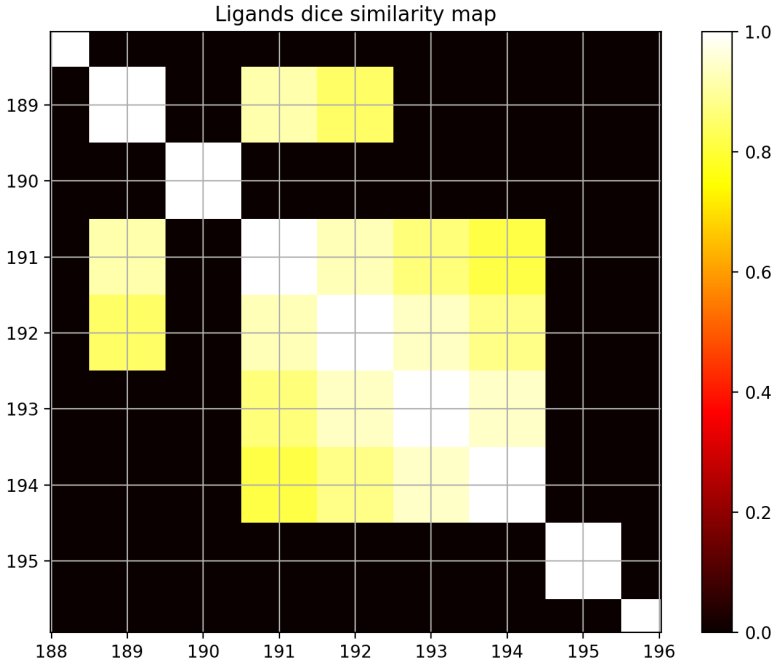
```
1.0000   [163]:   3nq9 C(=O)(CCCCCCC)O
1.0000   [189]:   3u9q C(=O)(CCCCCCCCC)O
1.0000   [191]:   3ueu O=C(O)CCCCCCCCCC
1.0000   [192]:   3uev C(=O)(O)CCCCCCCCCCCC
1.0000   [193]:   3uew C(=O)(O)CCCCCCCCCCCCCC
1.0000   [194]:   3uex C(=O)(O)CCCCCCCCCCCCCCCC
```

Ogólnie obie metody pokazują podobne klastry wspólnych cząsteczek, różnice są w wynikach, gdzie metoda Tanimoto często wspólne klastry klasyfikuje jako praktycznie identyczne cząsteczki. W poważnych badaniach proponowałbym użyć metody Tanimoto do identyfikacji grup cząsteczek mogących należeć do jednego klastra, zaś metody dice do wyszczególniania podobieństwa między cząsteczkami już w obrębie danego klastra.

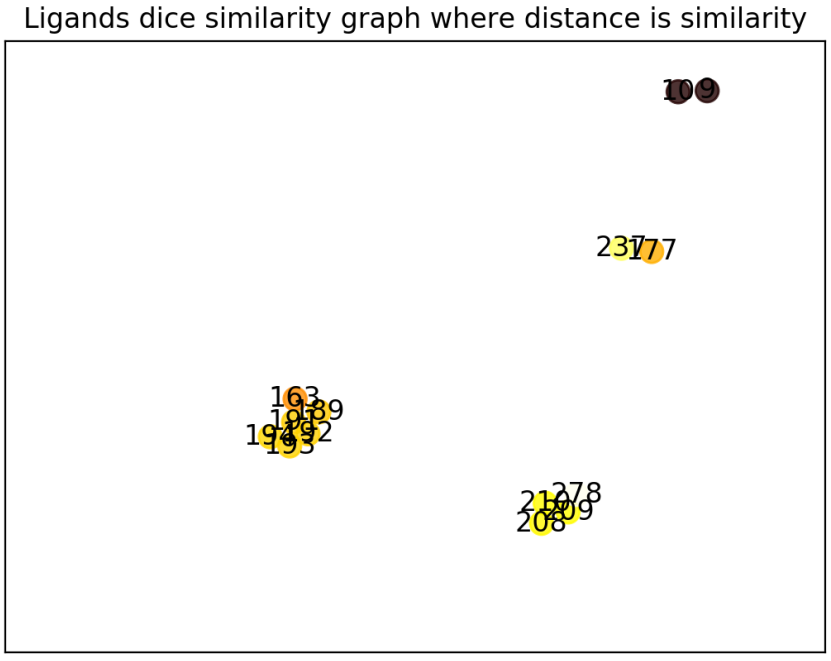
ANALIZA Z OBCIĘCIEM

parametry:
dice similarity, threshold = 0.8
komenda:
./redun.py db.smi -t 0.8

Przy takim obcięciu widać wyraźnie na obu wykresach klaster najbardziej podobnych do siebie cząsteczek czyli (163, 189, 191-194):



WYKRES LIGANDÓW Z PODOBIEŃSTWEM OKREŚLONYM KOLOREM, Z OBCIĘCIEM T = 0.8, KLASTER (163, 189, 191-194), BRAK NA WYKRESIE (163)

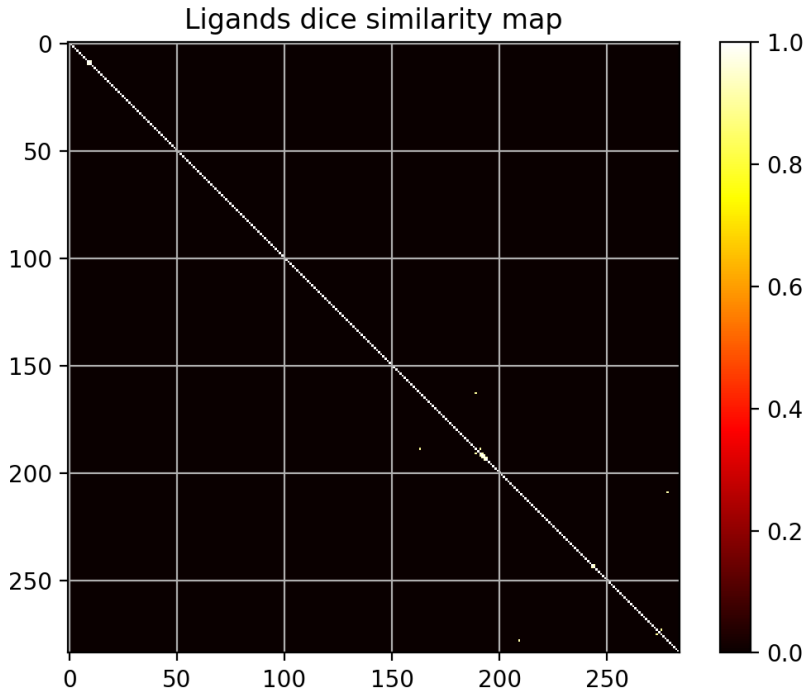


WYKRES LIGANDÓW Z PODOBIEŃSTWEM OKREŚLONYM ODLEGŁOŚCIĄ POMIĘDZY WIERZCHOŁKAMI, Z OBCIĘCIEM T = 0.8, NIEKTÓRE KLASTRY

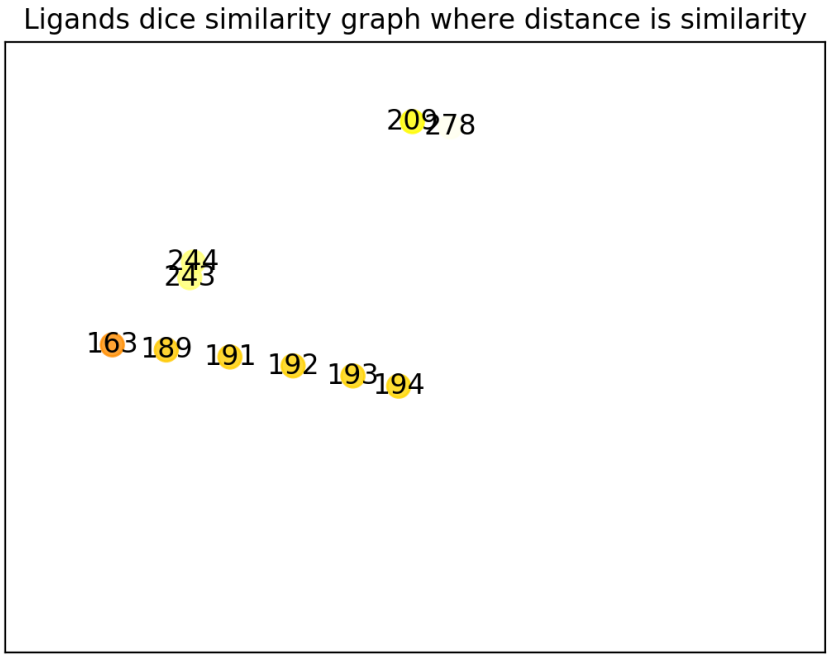
ANALIZA Z OBCIĘCIEM

parametry:
dice similarity, threshold = 0.9
komenda:
./redun.py db.smi -t 0.9

Po zidentyfikowaniu niektórych klastrów, można zająć się najbardziej podobnymi do siebie cząsteczkami. Ustawiając threshold na 0.9 zobaczyć możemy tylko najbardziej do siebie podobne ligandy:



WYKRES LIGANDÓW Z PODOBIEŃSTWEM OKREŚLONYM KOLOREM, Z OBCIĘCIEM T = 0.9



WYKRES LIGANDÓW Z PODOBIEŃSTWEM OKREŚLONYM ODLEGŁOŚCIĄ POMIĘDZY WIERZCHOŁKAMI, Z OBCIĘCIEM T = 0.9, NIEKTÓRE KLASTRY

Niewiele jest podobnych do siebie ligandów w takim stopniu. Widzimy przede wszystkim jeden znaczący klaster, który zauważyliśmy już w poprzednich analizach, w którym wyraźnie teraz widzimy kolejność podobieństwa w cząsteczkach, czyli:

(163, 189, 191-194)

oraz pojedyncze podobne do siebie cząsteczki:

(9, 10), (209, 278), (243, 244), (273, 275)

Na wykresie z odległościami możemy potwierdzić nasze spostrzeżenia, z wyjęciem par (9, 10) i (273, 275), które wręcz nachodzą na siebie, ale ze względu na bardzo niewielkie podobieństwo z cząsteczkami wewnątrz okręgu, wylądowały na jego obrzeżach.

Wybierzmy więc do dalszej analizy największy kompleks podobnych cząsteczek (163, 189, 191-194) i owe pary najbardziej podobnych cząsteczek.

DOKOWANIE

komenda:

./scorun.py db.smi [lista indeksów cząsteczek]

Zmapujmy indeksy cząsteczek na odpowiadające im nazwy i spróbujmy zadokować je w receptorach, cząsteczek, które są do nich podobne:

(163, 189, 191-194):

```
[163]:    3nq9 C(=O)(CCCCCCC)O
[189]:    3u9q C(=O)(CCCCCCCCC)O
[191]:    3ueu O=C(O)CCCCCCCCCCC
[192]:    3uev C(=O)(O)CCCCCCCCCCCCC
[193]:    3uew C(=O)(O)CCCCCCCCCCCCCCC
[194]:    3uex C(=O)(O)CCCCCCCCCCCCCCCCC
```

(9, 10):

```
[9]:      1h22 c1(=O)[nH]c2c(cc1)[C@H](CCC2)[NH2+]CCCCCCCCCC[NH2+][C@@H]1c2c…
[10]:     1h23 c1(=O)[nH]c2c(cc1)[C@@H]([NH2+]CCCCCCCCCCCC[NH2+][C@@H]1c3c…
```

(209, 278):

```
[209]:    4agp c1(C#CCOc2ccccc2)cc(c(c(C[N@@H+]2CC[C@@H]([NH+](CC)CC)CC2)c1)O)I
[278]:    5a7b C#Cc1cc(cc(C[N@@H+]2CC[C@@H]([NH+](CC)CC)CC2)c1O)C#CCOc1ccccc1
```

(243, 244):

```
[243]:    4ivc C[C@@H](O)c1n(c2c(n1)cnc1c2cc[nH]1)[C@H]1CC[C@@H](CC1)CC#N
[244]:    4ivd C[C@@H](O)c1n(c2c(n1)cnc1c2cc[nH]1)[C@H]1CC[C@@H](CC1)CCC#N
```

(273, 275):

```
[273]:    4w9h CC(=O)N[C@@H](C(C)(C)C)C(=O)N1C[C@H](O)C[C@H]1C(=O)NCc1ccc(cc1)…
[275]:    4w9l CC(=O)N[C@@H](C(C)(C)C)C(=O)N[C@@H](C(C)(C)C)C(=O)N1C[C@H]…
```

Dokowanie przeprowadzone zostało przy użyciu autorskiego skryptu wykorzystującego openbabel (konwersja formatów, generowanie struktury 3D z formatu SMILES), autodock vina (dokowanie ligandu do białka), oraz oddt (rescorowanie wyników funkcjami rfscore/nnscore).

W grupie par podobnych ligandów udało się osiągnąć następujące wyniki (format <ligand> -> <receptor>):

(9, 10):

```
9->9      1h22->1h22    0.99304998
9->10     1h22->1h23    1.00332526
10->9     1h23->1h22    0.99657079
10->10    1h23->1h23    1.00670944
```

(209, 278):

```
209->209  4agp->4agp    0.96576595
209->278  4agp->5a7b    0.97644919
278->209  5a7b->4agp    1.01830012
278->278  5a7b->5a7b    1.03719363
```

(243, 244):

```
243->243  4ivc->4ivc    1.00100415
```

243->244	4ivc->4ivd	0.98060521
244->243	4ivd->4ivc	1.01761134
244->244	4ivd->4ivd	1.00028245

(273, 275):

273->273	4w9h->4w9h	1.03317145
273->275	4w9h->4w9l	0.97287870
275->273	4w9l->4w9h	0.99693451
275->275	4w9l->4w9l	0.99465560

Jak widać po powyższych wynikach, w prawie każdej z powyższych sytuacji, jeden z ligandów miał trochę lepsze dopasowanie do centrum aktywnego białka, niż jego oryginalny inhibitor.

We wszystkich jednak przypadkach dokowanie cząsteczki podobnej przynosiło bardzo podobne rezultaty, można wnioskować zatem, że pomimo różnic w budowie kompleksy takie są niemal identyczne pod względem produkowania danych, co może powodować problemy przy kalibracji funkcji scorujących.

Ostatnim krokiem będzie sprawdzenie wyników z wielokrotnie wspominanego klastra:

(163, 189, 191-194):

163->163	3nq9->3nq9	0.85676112
163->189	3nq9->3u9q	0.85821108
163->191	3nq9->3ueu	0.88631163
163->192	3nq9->3uev	0.87455105
163->193	3nq9->3uew	0.87622973
163->194	3nq9->3uex	0.88970642
189->163	3u9q->3nq9	0.83642772
189->189	3u9q->3u9q	0.94515395
189->191	3u9q->3ueu	0.92985737
189->192	3u9q->3uev	0.92883530
189->193	3u9q->3uew	0.92889773
189->194	3u9q->3uex	0.94375980
191->163	3ueu->3nq9	0.93408289
191->189	3ueu->3u9q	0.98057993
191->191	3ueu->3ueu	1.00754948
191->192	3ueu->3uev	0.97525603
191->193	3ueu->3uew	0.97845763
191->194	3ueu->3uex	1.02162653
192->163	3uev->3nq9	0.98829298
192->189	3uev->3u9q	1.01787893
192->191	3uev->3ueu	1.06959521
192->192	3uev->3uev	1.03436388
192->193	3uev->3uew	0.99917071
192->194	3uev->3uex	1.06782302
193->163	3uew->3nq9	1.01257863
193->189	3uew->3u9q	1.04866773
193->191	3uew->3ueu	1.10183142
193->192	3uew->3uev	1.08168844
193->193	3uew->3uew	1.07005046
193->194	3uew->3uex	1.10930727
194->163	3uex->3nq9	0.99637145
194->189	3uex->3u9q	1.10731162
194->191	3uex->3ueu	1.14262727
194->192	3uex->3uev	1.11699508
194->193	3uex->3uew	1.07656556

```
194->194 3uex->3uex    1.14073375
```

W przypadku klastra podobnych cząsteczek widzimy wręcz identyczne wyniki dokowania dla każdej cząsteczki z osobna. Cząsteczki o krótszych łańcuchach miały praktycznie ten sam wynik dla każdego receptora z danego klastra.

Widzimy jednak trend, który pokazuje, że cząsteczki o dłuższych łańcuchach mają lepsze dopasowanie od tych o krótszych łańcuchach. Nie powinno to dziwić, ponieważ jak już wcześniej wyjaśniliśmy, cząsteczki te w zasadzie niczym się nie różnią oprócz długości łańcucha i krótszy ligand zmieści się cały do wnęki centra aktywnego, tworząc odpowiednie wiązania, ale nie wypełni go tak jak dłuższy ligand. Co zaś tyczy się płytszych centrów aktywnych (tam gdzie ligandy mają krótsze łańcuchy) dłuższa cząsteczka może utworzyć także wiązania z wgłębieniami na powierzchni białka, co bardziej ją ustabilizuje i da lepszy wynik.

Podsumowanie

Pomimo bycia CoreSet'em, baza PDBind kompleksów ligand-białko zawiera niewielkie klastry z podobnymi ligandami, które w obrębie klastrów otrzymują wręcz identyczne wyniki.

Ligandy z niektórych kompleksów potrafią być lepszymi inhibitorami, niż oryginalne cząsteczki z zestawu. Zaś wszystkie cząsteczki podobne na poziomie 0.9 dice similarity, były praktycznie identyczne pod względem łączenia się z centrum aktywnym receptora.

Takie dane mogą wprowadzać pewne błędy w kalibracji funkcji scorujących, jednakże klastrów podobnych do siebie w stopniu bardzo znacznym jest niewiele, a zazwyczaj są to też cząsteczki niepodobne do żadnych innych. Istnieje także niewiele par cząsteczek do siebie bardzo podobnych.

W zbiorze istnieją także odwrotne skrajności, czyli cząsteczki z bardzo niskim podobieństwem do wszystkich innych.

W poszukiwaniu podobieństw warto byłoby również przyjrzeć się samym receptorom, ich sekwencjom aminokwasowym i konformacjom w przestrzeni, na co niestety zabrakło mi już czasu.

Podobieństwa kompleksów białko-ligand w bazie PDBBind (CoreSet 2016)

Część 2

Wstęp

Jako iż zbadaliśmy już podobieństwa ligandów, warto teraz porównać receptory w kompleksach białko-ligand. Porównywanie będzie bardzo podobne do tego użytego przy ligandach, zmieni się jedynie metoda oceny podobieństwa.

Metodologia

Różnica pomiędzy badaniem podobieństw między ligandami, a receptorami, polegać będzie na zmianie programów oceniających owe podobieństwo. Przy receptorach korzystać będziemy z oprogramowania G-LoSA, oraz z plików .pdb w coreset'cie zawierających jedynie struktury centrów aktywnych receptorów (pliki o nazwie x_pocket.pdb).

Do badania owych podobieństw użyto nowego modułu, specjalnie do tego stworzonego, który analizuje podobieństwa w jednej bazie plików .pdb, która powstaje poprzez złączenie wszystkich plików .pdb, zawierających centra aktywne receptorów. Kod programu znajduje się na githubie: https://github.com/moozeq/DD_Redun/blob/master/sredun.py.

Problemy

Podstawowym problemem jest przystosowanie zestawu danych tak aby była możliwa ich analiza programem G-LoSA.

Wymagania co do plików przyjmowanych do G-LoSA są następujące:

1. Białka w plikach o formacie .pdb, zakończone sekwencją TER
2. Pliki z chemicznymi właściwościami wygenerowane przez program w pakiecie G-LoSA

Oba podpunkty osiągnięte zostały skryptem, który wykorzystując program AssignChemicalFeatures z pakietu G-LoSA, generuje pliki z właściwościami chemicznymi, oraz zamienia sekwencję końcową END na TER, przed ich porównaniem.

Największym problemem była jednak ilość obliczeń. Przy 286 receptorach należało wykonać porównania ok. 45 tys. razy. Niestety nawet wykorzystując multiprocessing, nie zdołałem obliczyć wszystkich podobieństw. Zająłem się więc jedynie tymi, gdzie wcześniej widoczne były podobieństwa między ligandami.

Kolejny błąd, na który się natknąłem, to G-LoSA z niejasnej dla mnie przyczyny, często rzucał segmentation fault, zwłaszcza przy wieloprocesowym wykonywaniu. Próba rozwiązania tego problemu to trzykrotne uruchamianie G-LoSA dla poszczególnych receptorów, za każdym razem informując użytkownika o ponownej próbie.

Analiza

Poniżej wyniki analizy podobieństwa receptorów dla poprzednio znalezionych podobnych ligandów (niezgodność numerów indeksów ligandów i receptorów wystąpiła przez usunięcie z zestawu ligandów jednej cząsteczki, która powodował błąd fingerprintu w pakiecie Chem i niektóre indeksy są przesunięte o 1 (mapowanie indeks -> nazwa można wywołać przy pomocy opcji *-m map)*:

(9, 10):
komenda: *./sredun.py prots.pdb -m pro -p 9 10*

```
[  1 / 285] 9<->10      1h22<->1h23   score:    0.999179
```

(209, 278):
komenda: *./sredun.py prots.pdb -m pro -p 209 279*

```
[  1 / 285] 209<->279   4agp<->5a7b   score:    0.967307
```

(243, 244):
komenda: *./sredun.py prots.pdb -m pro -p 243 244*

```
[  1 / 285] 243<->244   4ivc<->4ivd   score:    0.963505
```

(273, 275):
komenda: *./sredun.py prots.pdb -m pro -p 274 276 (<—indeksy +1)*

```
[  1 / 285] 274<->276   4w9h<->4w9l   score:    0.990518
```

(163, 189, 191-194):
komenda: *./sredun.py prots.pdb -m pro -p 163 189*
komenda: *./sredun.py prots.pdb -m pro -p 163 191*
komenda: *./sredun.py prots.pdb -m pro -p 163 192*
itd.

```
[  1 / 285] 163<->189   3nq9<->3u9q   score:    0.428358
[  1 / 285] 163<->191   3nq9<->3ueu   score:    0.836737
[  1 / 285] 163<->192   3nq9<->3uev   score:    0.828733
[  1 / 285] 163<->193   3nq9<->3uew   score:    0.831355
[  1 / 285] 163<->194   3nq9<->3uex   score:    0.890530
[  1 / 285] 189<->191   3u9q<->3ueu   score:    0.471181
[  1 / 285] 189<->192   3u9q<->3uev   score:    0.470454
[  1 / 285] 189<->193   3u9q<->3uew   score:    0.466859
[  1 / 285] 189<->194   3u9q<->3uex   score:    0.492897
[  1 / 285] 191<->192   3ueu<->3uev   score:    0.940702
[  1 / 285] 191<->193   3ueu<->3uew   score:    0.881044
[  1 / 285] 191<->194   3ueu<->3uex   score:    0.937876
[  1 / 285] 192<->193   3uev<->3uew   score:    0.916900
[  1 / 285] 192<->194   3uev<->3uex   score:    0.936019
[  1 / 285] 193<->194   3uew<->3uex   score:    0.944608
```

Jeśli chodzi o pary podobnych do siebie ligandów, to ich całe kompleksy są wręcz identyczne, podobieństwo receptorów na poziomie 0.99 (4w9h i 4w9l), a ligandów na poziomie 0.9 dice, wskazuje na praktycznie jednakowość owych kompleksów.

Jeśli zaś chodzi o klaster (163, 189, 191-194), to tak jak można było się spodziewać, w znaczącej większości przypadków tam gdzie podobne były ligandy, tam też podobne były receptory. W zasadzie wyłamuje się tutaj tylko jeden kompleks - 189 (3u9q), który mimo przynależącego do klastra ligandu, jako receptor nie jest w żaden sposób podobny do nich strukturalnie. Cała reszta klastra, zachowuje się dokładnie tak samo jak przy ligandach.