

UNIVERZA V LJUBLJANI
FAKULTETA ZA FARMACIJO

SAMO LEŠNIK

**NAČRTOVANJE, RAZVOJ IN TESTIRANJE
RAČUNALNIŠKEGA PROGRAMA ZA 2D IN 3D
VIRTUALNO REŠETANJE NA OSNOVI LIGANDA**

**DEVELOPMENT AND EVALUATION OF A
COMPUTER PROGRAM FOR 2D AND 3D LIGAND
BASED VIRTUAL SCREENING**

Ljubljana, 2013

Magistrsko nalogo sem opravljal na Fakulteti za farmacijo pod mentorstvom prof. dr. Stanislava Gobca in somentorstvom doc. dr. Janeza Konca. Virtualno rešetanje sem izvajal na računalnikih Kemijskega inštituta Ljubljana.

Zahvaljujem se somentorju doc. dr. Janezu Koncu za mentorstvo, pomoč in spodbudo pri programiranju, strokovnih dilemah in pisanju diplome. Iskrena hvala tudi prof. dr. Stanislavu Gobcu za mentorsko vodstvo, prof. dr. Dušanki Janežič, ki mi je omogočila delo na Kemijskem inštitutu in dr. Samu Turku, ki je za namene 3D virtualnega rešetanja molekulam iz Leads Now baze ustvaril več konformacij.

Hvala tudi bratu dr. Davorinu Lešniku za pomoč pri matematičnih vprašanjih, puncu Vesni za neomajno potrpežljivost in staršema Ireni in Albinu za moralno ter finančno podporo tekom študija.

Izjava

Izjavljam, da sem magistrsko nalogo samostojno izdelal pod mentorstvom prof. dr. Stanislava Gobca in somentorstvom doc. dr. Janeza Konca.

Samo Lešnik

Ljubljana, 2013

VSEBINA

POVZETEK	IV
ABSTRACT	V
SEZNAM OKRAJŠAV	VI
1. UVOD	1
1.1 ALGORITMI NA OSNOVI SUPERPOZICIJE MOLEKUL	3
1.2 ALGORITMI NA OSNOVI MOLEKULSKIH DESKRIPTORJEV	4
1.2.1 Dvodimenzionalni deskriptorji.....	4
1.2.1.1 Binarni deskriptorji.....	4
1.2.1.2 Drevesni deskriptorji	5
1.2.2 Tridimenzionalni deskriptorji.....	5
1.3 ALGORITMI ZA ISKANJE FARMAKOFORNIH PODOBNOSTI.....	5
1.4 ALGORITMI NA OSNOVI REKURZIVNE DELITVE	6
1.5 ALGORITMI NA OSNOVI TEORIJE GRAFOV	6
2. NAČRT DELA.....	8
3. METODE.....	9
3.1 VHODNI PODATKI	9
3.2 ALGORITEM ZA PRIMERJAVO MOLEKUL	12
3.2.1 Molekulski graf.....	12
3.2.2 Generiranje 2D produktnih grafov	14
3.2.2.1 Varianta 1	14
3.2.2.2 Varianti 2 in 3.....	15
3.2.3. Floyd-Warshall-ov algoritem.....	18
3.2.4 Generiranje 3D produktnih grafov	19
3.2.5 Algoritem za iskanje maksimalne klike.....	20
3.3. Izipis algoritma za virtualno rešetanje na osnovi liganda.....	23
4. REZULTATI	23
4.1 POSTOPEK IZBIRE NAJDENIH PODOBNIH MOLEKUL	24
4.2 2D VIRTUALNO REŠETANJE	25
4.2.1 Hitrost 2D virtualnega rešetanja.....	28
4.3 3D VIRTUALNO REŠETANJE	28
4.3.1 Hitrost 3D virtualnega rešetanja.....	29
5. RAZPRAVA.....	30

5.1 2D VIRTUALNO REŠETANJE	30
5.1.1 <i>Primerjava prve, druge in tretje variante</i>	30
5.1.1.1 Primerjava hitrosti	30
5.1.1.2 Primerjava najdenih podobnih molekul	31
5.2 3D VIRTUALNO REŠETANJE	32
5.2.1 <i>Primerjava s programom ShaEP</i>	32
5.3 PRIMERJAVA NAJDENIH MOLEKUL S PODOBNIMI MOLEKULAMI IZ ZINC BAZE	33
5.4 PREDLOGI ZA NADALJNJE DELO	33
6. SKLEP	34
LITERATURA	35

POVZETEK

Zaradi hitrega napredka računalniške tehnologije in proste dostopnosti velikih baz malih molekul, smo napisali nov program, ki omogoča 2D in 3D virtualno reševanje na osnovi liganda. Program z uporabo algoritma za iskanje maksimalne klike poišče največjo skupno podstrukturo med referenčno in tarčno molekulo nakar njuno podobnost oceni s Tanimotovim koeficientom. Za 2D reševanje smo ustvarili tri različne variante za tvorbo produktne grafa, za katere se je izkazalo, da dajejo drugačne rezultate, ter da se med seboj razlikujejo v času trajanja reševanja. Program smo preskusili na dveh acetilholinesteraznih inhibitorjih – donepezilu in GUK662. Kot tarčne molekule smo uporabili prosto dostopno ZINC bazo Leads Now, ki vsebuje približno dva milijona molekul; pri tem smo za 3D reševanje vsaki molekuli predhodno ustvarili približno tristo različnih konformacij, s čimer smo upoštevali njihovo fleksibilnost. Program je tako pri 2D kot 3D reševanju relativno visoko ocenil tudi molekule, pri katerih je razviden drugačen skelet kot pri referenčnih molekulah; torej je program izkazal tudi možnost odkrivanja skeletnih preskokov. Za zelo hitro se je izkazalo 2D reševanje v povezavi z varianto 2 in 3, saj smo dva milijona veliko bazo pregledali že v nekaj urah. Varianta 1 se je izkazala za petkrat počasnejšo, 3D reševanje pa je na 22 procesorskih jedrih trajalo sedem dni. Najdene podobne molekule so potencialni novi inhibitorji acetilholinesteraze, zato jih bomo v nadaljevanju te študije biološko ovrednotili.

Ključne besede: virtualno reševanje, maksimalna klika, molekularni graf

ABSTRACT

Due to the fast development of computer technology and free access to large small-molecule databases, we developed a computer program for 2D and 3D ligand based virtual screening. This program uses a maximum clique search algorithm, which enables us to find the maximum common substructure between the reference and target molecules. Then with the use of the Tanimoto coefficient their similarity is estimated. For the 2D virtual screening, we developed three different variants to create product graphs. It was evident that the three variants give dissimilar Tanimoto coefficients and required different lengths of time to complete virtual screening. The program was tested on two acetylcholinesterase inhibitors – donepezil and GUK662. For target molecules the freely accessible ZINC Leads Now database, which contains about two million molecules, has been used; however for the 3D virtual screening we preliminarily created approximately three hundred conformations for each molecule, thereby taking into account their flexibility. For both 2D and 3D virtual screening the program scored relatively highly molecules, in which new scaffolds were evident – the program has the feature of scaffold hopping. The program's 2D method combined with variant 2 and 3 was demonstrated to be very time efficient – it took only a few hours to screen through the Leads Now database. Variant 1 was five times slower than variant 2 and 3; moreover, 3D virtual screening of the changed Leads Now database took seven days on twenty-two processor cores to complete. Our program found a range of similar molecules to donepezil and GUK662. These molecules are potential new acetylcholinesterase inhibitors which will be evaluated in later stages with biological testing.

Key words: virtual screening, maximum clique, molecular graph

SEZNAM OKRAJŠAV

GUK662 N-((1-(2,3-dihidro-1H-inden-2-il)piperidin-3-il)metil)-N-(2-metoksieil)-2-naftamid

PDB Protein Data Bank

SAR odnos med strukturo in aktivnostjo (angl. *structure activity relationship*)

1. Uvod

Na odločitev, za katero bolezen bomo razvijali novo učinkovino, vplivajo tako medicinski kot ekonomski razlogi (1). Med medicinske razloge spada potreba po novih učinkovinah za zdravljenje bolezni, za katere še nimamo znanih učinkovin, in potreba po boljših učinkovinah za bolezni, za katere že imamo učinkovine. Zaradi velikih stroškov razvoja novih učinkovin je potrebno upoštevati tudi ekonomske razloge, zato se raziskovanje osredotoča predvsem na bolezni, ki so pogoste v razvitem svetu, na primer gripa, debelost, migrena, srčno-žilne bolezni in psihiatrične bolezni. Danes je pomembno orodje pri odkrivanju novih učinkovin računalnik. Računalniške metode so uspešne pri *problemu velikih števil* – število molekul, ki jih lahko analizira človek je zelo omejeno, računalniki pa se zelo dobro obnesejo, ko je potrebno analizirati velike baze molekul za potencialne nove učinkovine. Prav tako se računalniške metode dobro obnesejo pri *problemu multidimenzionalnega optimiziranja* učinkovin. Spojine vodnice še niso nujno uporabne kot zdravilne učinkovine. S pomočjo računalnikov optimiziramo molekulo spojine vodnice tako, da uporabimo algoritme, ki hkrati zadostijo več lastnostim in omejitvam, ki jih naj bi imela zdravilna učinkovina. Računalniške metode nam omogočajo tudi pocenitev postopka odkrivanja novih učinkovin. Omogočajo nam obogatitev kemijskih knjižnic, tako da nam ni potrebno izvesti biokemijskih testov na celotni knjižnici (2).

Ko izberemo bolezen, za katero bomo razvijali zdravilno učinkovino, moramo identificirati ustrezno tarčo. To je največkrat protein oziroma nukleinska kislina, ki je povezana z boleznijo (1). V preteklosti je bila večina tarč odkritih tako, da je bila v izhodišču že znana molekula, ki je delovala na tarčo. Primer je morfin, učinkovina, ki je bila poznana in se je uporabljala preden so poznali tarčo, na katero deluje. Šele kasneje so odkrili telesu lastne kemične prenašalce s podobnimi kemijskimi lastnostmi kot jih ima morfin, in na osnovi podobnosti med morfinom in temi prenašalci našli morfinov tarčni protein, to je opioidni receptor.

Že pred uporabo računalnikov so na osnovi podobnosti molekul razvijali nove zdravilne učinkovine (1). Naravna liganda adrenalin in noradrenalin sta bila uporabljena kot spojini vodnici pri načrtovanju β -agonistov, kot so salbutamol, dobutamin ali ksalmeterol. Razvoj nove zdravilne učinkovine na osnovi podobnosti z že znano učinkovino lahko uporabimo tudi z namenom izogibanja patentnim omejitvam (*angl.*, »me too« drugs). Pri tem kot spojino vodnico vzamemo obstoječo učinkovino, ki jo strukturno modificiramo, a hkrati

ohranimo aktivnost učinkovine. Primer je antihipertenzivna učinkovina kaptopril, ki je bila spojina vodnica za novejšo podobno antihipertenzivo. Že obstoječo učinkovino lahko tudi izboljšamo (angl. »*me better*« *drugs*). Primer so moderni penicilini, ki so bolj selektivni, učinkoviti in stabilni kot izhodni penicilini.

Učinkovina lahko služi kot spojina vodnica na osnovi njenega stranskega učinka (2). V takem primeru je naš cilj spremeniti spojino vodnico tako, da določen stranski učinek ojačimo in hkrati minimiziramo njen osnovni, terapevtski učinek. Primeri tako odkritih učinkovin so antidiabetiki skupine sulfonilsečnin. Odkrili so, da določeni sulfonamidi, ki so protibakterijske učinkovine, povzročajo konvulzije, saj kot stranski učinek inducirajo hipoglikemijo. Tako so na osnovi sulfonamidov načrtovali nove antidiabetične učinkovine. Z ustreznimi strukturnimi modifikacijami so zmanjšali protibakterijsko delovanje sulfonamidov in ojačili hipoglikemično delovanje, kar je vodilo v odkritje tolbutamida.

Mapiranje genoma, to je odkrivanje nukleotidnega zaporedja organizmov, nam omogoča, da odkrivamo mnoge nove potencialne tarčne proteine, na katere bi lahko delovale nove učinkovine (1). Za mnoge tako odkrite tarčne proteine še ni znana njihova tridimenzionalna (3D) kristalna struktura v proteinski bazi (angl. *Protein Data Bank*) (3). Imamo tudi baze oziroma knjižnice več milijonov znanih malih molekul, a za mnogo teh molekul še ne poznamo njihovega farmakološkega delovanja.

Z razvojem računalnikov se je uveljavilo virtualno reševanje na osnovi znanih ligandov, ki omogoča odkrivanje novih zdravilnih učinkovin (1). Pri tem pristopu moramo poznati le strukturo najmanj ene že znane aktivne molekule oziroma liganda, ki ima želeno delovanje oziroma farmakološki učinek na izbrano tarčo. Takšne molekule imenujemo referenčne molekule in so lahko endogeni ligandi, naravne spojine ali sintezne spojine. Pristop virtualnega reševanja temelji na preiskovanju oziroma reševanju računalniške baze molekul z namenom, da najdemo molekule, ki imajo strukturne, topološke ali farmakoforne podobnosti z referenčno molekulo. Rezultat so podobne molekule, ki jih najdemo v bazi, razvrščene glede na njihovo podobnost z referenčno molekulo, pri čemer podobnost merimo s tako imenovano stopnjo podobnosti (angl. *similarity score*). Virtualno reševanje na osnovi ligandov lahko izvajamo z različnimi računskimi postopki – algoritmi, ki temeljijo na iskanju podobnosti med molekulami v eni dimenziji, dveh dimenzijah ali treh dimenzijah. Zaželena lastnost programov za virtualno reševanje na osnovi ligandov je skeletni preskok (angl. *scaffold hopping*), to je detekcija molekul s podobnim delovanjem, a z drugačnimi osnovnimi skeleti (strukturami) kot referenčna molekula. Programi, ki

omogočajo skeletni preskok, omogočajo odkrivanje učinkovin s podobnim delovanjem kot ga ima referenčna molekula, a z drugačnimi elementi strukture, kar med drugim omogoča izogibanje patentnim omejitvam. Postopek virtualnega reševanja na osnovi liganda nam omogoča, da biokemijsko testiramo samo molekule, ki so dovolj podobne referenčni molekuli in za katere zato pričakujemo, da bodo imele podobno delovanje na izbrano tarčo. Najbolj znane računalniške baze molekul, ki se pogosto uporabljajo za virtualno reševanje na osnovi liganda, so na primer NCI (<http://cactus.nci.nih.gov/>), ZINC (<http://zinc.docking.org/>) in Cambridge (<http://www.ccdc.cam.ac.uk>) baza. Sledi opis nekaterih najbolj uporabljenih algoritmov.

1.1 Algoritmi na osnovi superpozicije molekul

Algoritmi za virtualno reševanje lahko delujejo na osnovi superpozicije malih molekul (angl. *small molecule alignment*) (4). V postopku superpozicije algoritmi prilagajajo prostorske koordinate atomov strukture referenčne molekule s strukturo tarčne molekule. Na osnovi te superpozicije nato določijo stopnjo podobnosti med primerjanima molekulama. V splošnem se tarčna molekula obravnava kot fleksibilna, referenčna pa je lahko rigidna ali fleksibilna. Obstaja več algoritmov, ki delujejo na osnovi superpozicije ligandov, na primer FlexS (5), ROCS (6), GASP (7), in tako dalje. Algoritem ROCS (angl. *Rapid Overlay of Chemical Structures*) deluje na principu prekrivanja molekularnih volumnov. Volumen molekule je predstavljen z Gaussovo funkcijo (8), superpozicija molekul pa temelji na maksimizaciji prekrivanja volumnov. ROCS poleg volumnov upošteva tudi fizikalno-kemijske lastnosti funkcionalnih skupin. Te lastnosti predstavi z barvami, na primer akceptorje vodikovih vezi z rdečo, donorje z modro, obročne sisteme z zeleno, in tako dalje. Pri superpoziciji dveh molekul algoritem prekriva le funkcionalne skupine z ujemajočimi se barvami. Algoritem referenčno molekulo obravnava kot rigidno, iz tarčnih molekul pa tvori več konformacij in vsako konformacijo obravnava kot rigidno, s čimer predstavi fleksibilnost tarčnih molekul. Na osnovi števila prileganih funkcionalnih skupin, ki so skupne obema molekulama, algoritem nato oceni stopnjo podobnosti med referenčno in tarčno molekulo, izraženo s Tanimotovim koeficientom (9). Postopek ponavlja, dokler ne prilega vseh konformacij tarčnih molekul, nato pa algoritem razporedi tarčne molekule glede na njihovo stopnjo podobnosti z referenčno molekulo. Ker referenčno molekulo obravnava kot rigidno, je pomembno, da je njena konformacija podobna biološko aktivni konformaciji v vezavnem mestu tarčnega proteina.

1.2 Algoritmi na osnovi molekulskih deskriptorjev

Algoritmi za virtualno reševanje na osnovi molekulskih deskriptorjev so predvsem primerni za pregled večjih baz spojin, na primer več milijonov spojin, saj je čas primerjave para molekul krajši kot pri algoritmih na osnovi superpozicije predstavljenih v prejšnjem poglavju (4). Molekulski deskriptorji opisujejo fizikalno-kemijske lastnosti molekul v zgoščeni obliki. Glede na število dimenzij, s katerimi opisujejo obravnavane molekule, jih delimo na enodimenzionalne (1D), dvodimenzionalne (2D) in tridimenzionalne (3D). Med 1D deskriptorje spadajo lastnosti, kot so na primer log P, molekulska masa in molekulska refraktivnost, 2D deskriptorje pa delimo na linearne deskriptorje in drevesne deskriptorje. Linearne deskriptorje delimo naprej na realne in binarne deskriptorje, binarne pa naprej na strukturne ključe in molekulske prstne odtise. V nadaljevanju bomo opisali nekaj najpogosteje uporabljenih 2D in 3D deskriptorjev.

1.2.1 Dvodimenzionalni deskriptorji

1.2.1.1 Binarni deskriptorji

Binarni deskriptorji opisujejo fizikalno-kemijske lastnosti molekule z biti (4). Bit je najmanjša enota informacije, ki ima lahko le vrednosti 0 ali 1, pri čemer 0 pomeni odsotnost, 1 pa prisotnost lastnosti. Lastnosti, ki jih opišemo z biti, so na primer prisotnost oziroma odsotnost funkcionalnih skupin ali specifičnih vezi (vodikovih, disulfidnih), z binarnimi deskriptorji pa lahko predstavimo tudi število funkcionalnih skupin ali vezi.

Algoritmi na osnovi strukturnih ključev uporabljajo tabelo binarnih vrednosti, ki vsebuje informacije o prisotnosti ali odsotnosti 2D fragmentov v molekuli, poleg tega pa tudi informacijo o številu teh fragmentov (10). Problem strukturnih ključev je, da moramo predhodno definirati fragmente, na primer benzenski obroč, alifatske verige, in tako dalje, ki jih bomo upoštevali, kar zoža nabor molekul, ki jih lahko na ta način opišemo, samo na molekule, ki imajo definirane fragmente. Algoritma razvita na tem principu, sta na primer MACCS (11) in 3D MIND (12).

Algoritmi na osnovi molekulskih prstnih odtisov (*angl.* molecular fingerprints) v primerjavi s strukturnimi ključi sami definirajo fragmente (teh ni potrebno v naprej definirati), tako da zaporedno združujejo povezane atome v večje fragmente (4). Molekulo OC=CN na primer poiščejo s fragmenti C, O, N, OC, C=C, CN, OC=C, C=CN in OC=CN (13). Na tej osnovi je bil razvit splošno uporabljeni program Daylight (14).

1.2.1.2 Drevesni deskriptorji

Drevesni deskriptorji predstavijo molekule v obliki drevesa, sestavljenega iz vozlišč in povezav (4). Vozlišča predstavljajo funkcionalne skupine molekule in vsebujejo tudi informacijo o njihovih fizikalno-kemijskih lastnostih. Povezave med vozlišči pa so vezi. Drevesni deskriptorji se od pred prej opisanih linearnih deskriptorjev razlikujejo v tem, da zajamejo relativno razporeditev funkcionalnih skupin v molekuli, medtem ko linearni deskriptorji te razporeditve ne zajamejo. Podobnost med dvema molekulama dobimo tako, da prekrijemo njuni drevesi (vozlišča enega drevesa z vozlišči drugega), ne da bi pri tem spremenili topologijo dreves, in preštejemo prekrita vozlišča. Drevesni deskriptorji natančneje kot linearni opisujejo lastnosti molekul, zaradi česar njihova uporaba poveča verjetnost odkritja strukturno različnih podobnih molekul (skeletni preskok) (12). Primeri algoritmov na osnovi drevesnih deskriptorjev so FTree (15), Mtree (16) in NipalsTree (17).

1.2.2 Tridimenzionalni deskriptorji

Iskanje podobnosti na osnovi 3D strukture molekule temelji na hipotezi, da imajo molekule s podobnim položajem ključnih funkcionalnih skupin v prostoru podobne lastnosti (4). Tridimenzionalni deskriptorji opisujejo lastnosti molekul povezane z njihovo obliko, položajem in vrsto funkcionalnih skupin. Generiramo jih lahko s pomočjo programov, kot je na primer 3D-FEATURE, ki generira 3D deskriptorje glede na položaj in prisotnost donorjev in akceptorjev vodikovih vezi ter hidrofobnih skupin. Zapise molekul lahko pretvorimo iz 2D zapisa v 3D s pomočjo programa CORINA. Ker 3D deskriptorji opisujejo samo eno konformacijo molekule, moramo, da se izognemo tej omejitvi, generirati več konformacij iste molekule, in vsako posebej pretvoriti v zapis s 3D deskriptorji. Ker 3D deskriptorji opisujejo položaje funkcionalnih skupin v prostoru, metode na osnovi 3D deskriptorjev še povečujejo možnosti skeletnega preskoka glede na 2D deskriptorje (12). Algoritem, ki deluje na osnovi 3D deskriptorjev je na primer DISCO (18).

1.3 Algoritmi za iskanje farmakofornih podobnosti

Farmakoforni model je presek steričnih in elektrostatskih lastnosti večih aktivnih molekul, in predstavlja najmanjši skupni imenovalec, ki tem molekulam omogoča, da se vežejo ter s tem aktivirajo oziroma deaktivirajo biološko tarčo (19). Farmakoforni model postavimo tako, da superponiramo več aktivnih molekul (angl. *training set*) in na osnovi te superpozicije najdemo skupne kemijske lastnosti, ki so odgovorne za aktivnost

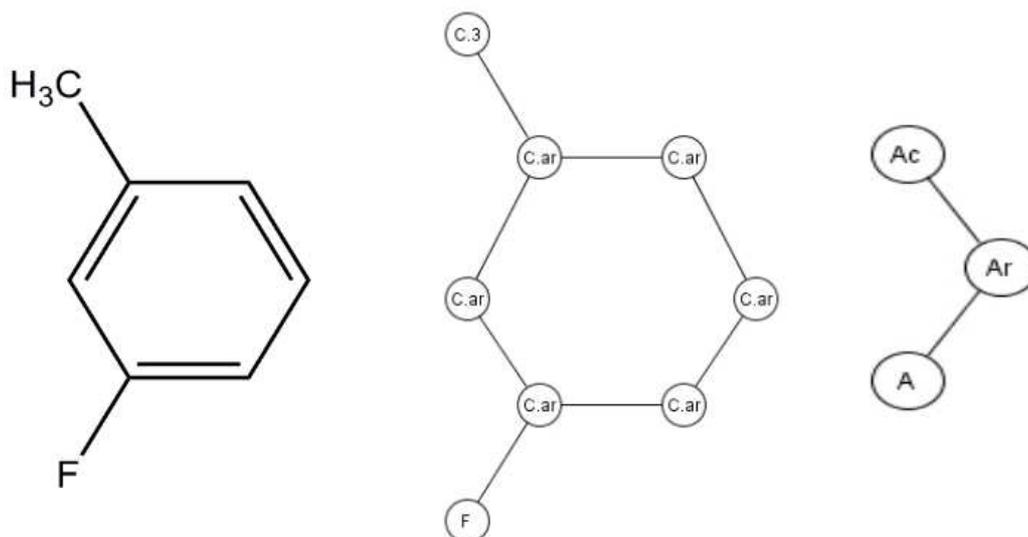
superponiranih molekul. Farmakofori so točke v prostoru, na primer ionski centri, hidrofobne skupine ali aromatski obroči. Farmakoforni model generiramo v dveh korakih: 1) konformacijska analiza aktivnih molekul, katere rezultat je več možnih konformacij aktivnih molekul in 2) superpozicija vseh konformacij aktivnih molekul ter iskanje njihovih skupnih lastnosti. Algoritem nato tako postavljeni farmakoforni model primerja s tarčnimi molekulami in določi stopnje podobnosti.

1.4 Algoritmi na osnovi rekurzivne delitve

Algoritmi na osnovi rekurzivne delitve prepoznajo statistično pomembne lastnosti množici molekul, na primer molekulske mase in prisotnost oziroma odsotnost funkcionalnih skupin, ki omogočajo razlikovanje aktivnih od neaktivnih molekul in napovedovanje aktivnosti novih molekul (20). Ti algoritmi ustvarijo odločitveno drevo, v katerem vsako vozlišče predstavlja neko lastnost molekul, povezavi iz vsakega vozlišča (dve iz vsakega vozlišča) pa predstavljata prisotnost oziroma odsotnost lastnosti (21). Primer programa za virtualno reševanje na osnovi rekurzivne delitve je SCAM (22), ki ustvari odločitveno drevo tako, da so lastnosti molekul predstavljene z binarnimi molekulskimi deskriptorji. Program za vsak deskriptor izvede statistični t-test, in sicer med skupino molekul, ki ima lastnost, ki jo opisuje deskriptor, in med skupino, ki te lastnosti nima. Kriterij z najvišjo t-vrednostjo se razvrsti najvišje v drevesu, nato pa sledijo tisti z nižjimi t-vrednostmi.

1.5 Algoritmi na osnovi teorije grafov

Pri teh algoritmih molekularno strukturo prevedemo v obliko molekulskega grafa, sestavljenega iz vozlišč in povezav (23). Atome oziroma skupine atomov predstavimo z vozlišči, kemijske vezi pa s povezavami med vozlišči (Slika 1).



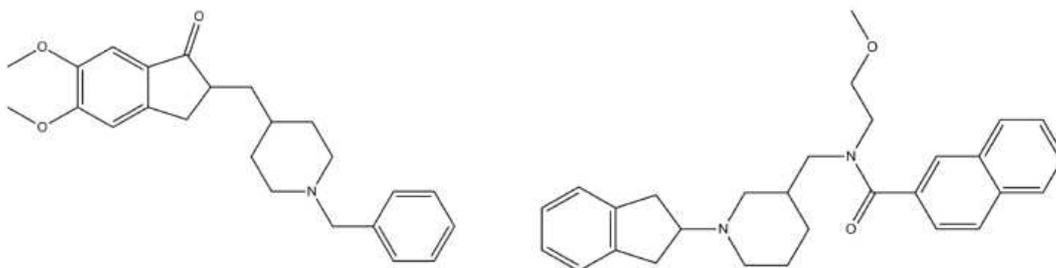
Slika 1. Primer molekule (levo), molekularni graf, kjer so vozlišča težki atomi (sredina), molekularni graf, kjer so vozlišča funkcionalne skupine (desno).

Obstaja več načinov, s katerimi molekule prevedemo v molekularne grafe. Eden izmed možnih načinov je, da so vozlišča posamezni atomi molekule. Lahko pa so vozlišča funkcionalne skupine, na primer donorji in akceptorji vodikovih vezi, hidrofobne skupine, obročni sistemi, in tako dalje. Vsako vozlišče ima tudi lastnost, ki predstavlja vrsto atoma ali funkcionalne skupine. Algoritem generira dva molekularna grafa na osnovi molekul, ki ju želimo primerjati, nato pa poišče njun maksimalni skupni podgraf, in sicer največjo podmnožico vozlišč in povezav, ki je skupna obema molekularnima grafoma. V naslednjem koraku tega procesa algoritem iz dveh molekularnih grafov tvori njun produktni oziroma korespondenčni graf, katerega vozlišča imajo dve komponenti (par vozlišč): prvo vozlišče iz para je iz prvega molekularnega grafa, drugo pa iz drugega, pri čemer obe vozlišči predstavljata enak atom oziroma funkcionalno skupino. V produktnem grafu sta dve vozlišči povezani, če je razdalja med njunima prvima komponentama enaka razdalji med njunima drugima komponentama. Razdalja med dvema vozliščema molekularnega grafa je število povezav na najbližji poti med tema vozliščema. Nadalje algoritem poišče maksimalno kliko produktnega grafa, to je največji popolnoma povezan podgraf produktnega grafa. Maksimalna klika v produktnem grafu predstavlja največjo skupno podstrukturo dveh primerjanih molekul. Končno, podobnost molekul algoritem izrazi s

stopnjo podobnosti, na primer Tanimotovim koeficientom, ki je opisan v metodah spodaj.

2. Načrt dela

Acetilholinesteraza je encim, ki hidrolizira acetilholin in je udeležen pri zelo različnih boleznih (Alzheimerjeva bolezen, miastenija gravis). Znanih je več ligandov acetilholinesteraze, med drugim so razvili tudi inhibitorje acetilholinesteraze, ki se uporabljajo za zdravljenje Alzheimerjeve bolezni. V programskem jeziku C++ bomo razvili program, ki bo, ki bo omogočal hitro virtualno reševanje na osnovi znanih ligandov. Program bomo uporabili na primeru dveh znanih ligandov, inhibitorjev acetilholin esteraze, in sicer donepezilu in GUK662 (Slika 2), s katerima bomo pregledali ZINC bazo več milijonov spojin, in v njej poiskali molekule, ki so podobne znanima inhibitorjema, in bi posledično lahko imele podobno inhibitorno delovanje na acetilholinesterazo.

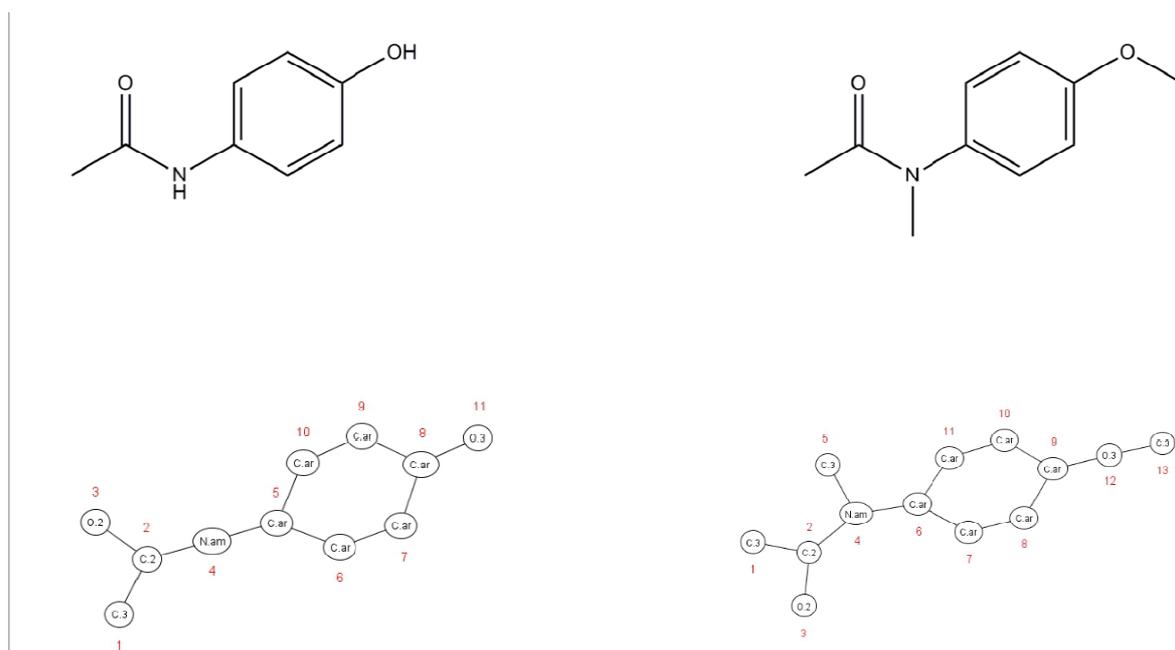


Slika 2. Inhibitorja acetilholinesteraze, ki ju uporabljamo kot referenčni molekuli. Donepezil (levo) in GUK662 (desno).

Program bo zaporedno primerjal referenčno molekulo z vsako molekulo iz baze (tarčno molekulo). Program bo bral zapise teh molekul v obliki mol2 formata, z algoritmom maksimalne klike MaxCliqueDyn (24) pa bo poiskal največjo skupno podstrukturo obema molekulama in s Tanimotovim koeficientom izrazil njuno podobnost. Program bo obravnaval molekule tako v dveh dimenzijah kot tudi v treh dimenzijah. Fleksibilnost tarčnih molekul bomo upoštevali tako, da bomo za vsako tarčno molekulo ustvarili več možnih konformacij. Referenčne molekule bomo obravnavali kot rigidne v bioaktivni konformaciji. Za molekule z visokim Tanimotovim koeficientom bomo lahko sklepali, da imajo zaradi podobnosti z že znanimi inhibitorji tudi podobno farmakološko delovanje, zato jih bomo predlagali za nadaljnje biokemijske preiskave.

3. Metode

Razvili smo nov program za 2D in 3D virtualno reševanje na osnovi liganda. Algoritem primerja 2D ali 3D strukturo malih molekul in išče njihove strukturne in topološke podobnosti. Algoritem molekule obravnava kot molekulske grafe iz katerih generira produktni graf nato pa z algoritmom za iskanje maksimalne klike poišče največjo skupno podstrukturo v primerjanih molekulskih grafih. Program smo razvili v programskem jeziku C++. Delovanje programa bomo na tem mestu prikazali na primeru dveh molekul: paracetamola in N-(4-metoksifenil)-N-metilacetamida (Slika 3).



Slika 3. Struktura paracetamola (levo zgoraj), molekularni graf paracetamola (levo spodaj), struktura N-(4-metoksifenil)-N-metilacetamida (desno zgoraj), molekularni graf N-(4-metoksifenil)-N-metilacetamida (desno spodaj).

3.1 Vhodni podatki

Kot vhodne podatke program pričakuje dve molekuli, med katerima bo iskal podobnost. Program bere molekuli v obliki mol2 formata (25), ki vsebuje informacijo o atomih, vezeh in koordinatah atomov. Pri branju izpusti vodike, saj ti niso pomembni za določanje podobnosti med molekulama. Paracetamol, zapisan v mol2 formatu je predstavljen na Sliki 4.

```

1 <TRIPOS>MOLECULE
2 ZINC18274777
3 20 20 0 0 0
4 SMALL
5 USER_CHARGES
6
7 @<TRIPOS>ATOM
8 1 C1 -2.1938 1.4701 0.6085 C.3 1 <0> -0.1738
9 2 C2 -2.1619 -0.0367 0.6077 C.2 1 <0> 0.5100
10 3 O1 -3.0611 -0.6582 1.1456 O.2 1 <0> -0.5253
11 4 N1 -1.1558 -0.6938 0.0094 N.am 1 <0> 0.6760
12 5 C3 -1.0811 -2.0905 0.0916 C.ar 1 <0> 0.1315
13 6 C4 0.1554 -2.7160 0.1778 C.ar 1 <0> -0.1278
14 7 C5 0.2283 -4.0925 0.2593 C.ar 1 <0> -0.1267
15 8 C6 -0.9336 -4.8504 0.2546 C.ar 1 <0> 0.0907
16 9 C7 -2.1693 -4.2261 0.1680 C.ar 1 <0> -0.1267
17 10 C8 -2.2439 -2.8493 0.0927 C.ar 1 <0> -0.1043
18 11 O2 -0.8611 -6.2059 0.3344 O.3 1 <0> -0.4991
19 12 H1 -3.0822 1.8186 1.1352 H 1 <0> 0.0977
20 13 H2 -1.3026 1.8478 1.1098 H 1 <0> 0.0926
21 14 H3 -2.2144 1.8337 -0.4188 H 1 <0> 0.0926
22 15 H4 -0.4783 -0.1997 -0.4782 H 1 <0> 0.4131
23 16 H5 1.0601 -2.1261 0.1808 H 1 <0> 0.1308
24 17 H6 1.1901 -4.5792 0.3264 H 1 <0> 0.1337
25 18 H7 -3.0735 -4.8168 0.1636 H 1 <0> 0.1330
26 19 H8 -3.2062 -2.3631 0.0301 H 1 <0> 0.1433
27 20 H9 -0.8763 -6.5480 1.2387 H 1 <0> 0.3908
28 @<TRIPOS>BOND
29 1 1 2 1
30 2 1 12 1
31 3 1 13 1
32 4 1 14 1
33 5 2 3 2
34 6 2 4 am
35 7 4 5 1
36 8 4 15 1
37 9 5 10 ar
38 10 5 6 ar
39 11 6 7 ar
40 12 6 16 1
41 13 7 8 ar
42 14 7 17 1
43 15 8 9 ar
44 16 8 11 1
45 17 9 10 ar
46 18 9 18 1
47 19 10 19 1
48 20 11 20 1

```

Slika 4. Paracetamol v mol2 formatu. Program v tem primeru razlikuje naslednje tipe

atomov: C.3 je sp³ hibridiziran ugljik, C.2 je sp² hibridiziran ugljik, C.ar je aromatski ugljik, O.2 je sp² hibridiziran kisik, N.am je amidni dušik.

```
1 void Graf::open_file (string infile) {
2     data.open (infile.c_str(), ifstream::in);
3 }
4
5 bool Graf::read_file() {
6     bool c = false, d = false;
7     bonds_nu = 0;
8     while (!getline(sata, line).eof()) {
9         if (line.compare (0, 4, "ZINC") == 0) {
10             set_zinc_id(line);
11         }
12         if (line.compare (0, 13, "@<TRIPOS>ATOM") == 0) {
13             c = true;
14             d = false;
15             continue;
16         }
17         if (line.compare (0, 13, "@<TRIPOS>BOND") == 0) {
18             d = true;
19             c = false;
20             continue;
21         }
22         if (line.compare(0, 1, "@") == 0 && d == true) {
23             return true;
24         }
25         if (c) {
26             char temp_str [200];
27             char temp_strr [200];
28             char H_switch [5];
29             sscanf (line.c_str(), "%*d %*s %*f %*f %*f %s %*d %*s %*f",
H_switch);
30             string struna = H_switch;
31             if (struna != "H") {
32                 sscanf (line.c_str(), "%d %s %f %f %f %s %*d %*s %*f",
&n[size].atom_nu, temp_str, &n[size].coordinates[0], &n[size].coordinates[1],
&n[size].coordinates[2], temp_str);
33                 convert [n[size].atom_nu] = size;
34                 n[size].name = temp_str;
35                 n[size++].last_name = temp_str
36             }
37         }
38         if (d) {
39             char temp_str2 [200];
40             int a,b;
41             sscanf (line.c_str(), "%*d %d %d %s", &a, &b, temp_str2);
42             if (convert[a] != -1 && convert[b] != -1) {
```

```

43         n      [convert[a]].bond[n[convert[a]].bond_size++] =
convert[b];
44         n      [convert[b]].bond[n[convert[b]].bond_size++] =
convert[a];
45         a_bond [bonds_nu] = a;
46         b_bond [bonds_nu] = b;
47         bond_type [bonds_nu] = temp_str2;
48         bonds_nu++;
49     }
50 }
51 }
52     return false;
53 }
54
55 void Graf::close_file () {
56     data.close ();
57 }

```

Slika 5. Podprogram, ki omogoča branje mol2 datoteke.

Podprogram, ki omogoča branje mol2 formata je prikazan na Sliki 5. Program odpre mol2 datoteko referenčne molekule, ki jo definiramo ob zagonu programa. Odprto mol2 datoteko bere po vrsticah. Ko naleti na vrstico, ki vsebuje ZINC (Slika 4), shrani ZINC kodo molekule, ki jo bo kasneje potreboval pri izpisu za identifikacijo molekule. Program nadaljuje, dokler ne naleti na vrstico, ki vsebuje @<TRIPOS>ATOM. Nato iz vsake vrstice prebere in shrani v spomin številko, koordinate in tip atoma. Ko program naleti na vrstico @<TRIPOS>BOND, začne branje vezi molekule. V vsaki sledeči vrstici prebere in shrani številki povezanih atomov in tip vezi. Pri branju program izpusti vodikove atome in vezi, ki se nanašajo na vodikove atome. Ko program konča z branjem referenčne molekule, na enak način prebere tudi tarčno molekulo. Tudi ime tarčne molekule definiramo ob zagonu programa.

3.2 Algoritem za primerjavo molekul

Algoritem za primerjavo molekul najprej prevede molekuli iz mol2 formata v obliko molekulskih grafov predstavljenih z vezavnima tabelama, nato iz molekulskih grafov generira 2D ali 3D produktne grafe. Nadalje z algoritmom za iskanje maksimalne klike v produktnem grafu poišče največjo skupno podstrukturo referenčne in tarčne molekule. Končno izračuna stopnjo podobnosti obeh molekul s pomočjo Tanimotovega koeficienta.

3.2.1 Molekulski graf

Molekulski graf opišemo z vezavno tabelo, ki vsebuje informacije o povezanosti atomov v

molekuli (Slika 6). V prvi vrstici in prvem stolpcu vezavne tabele so številke atomov referenčne oziroma tarčne molekule. Če obstaja povezava med atomoma s številka m in n , je vrednost pripadajoče celice (m, n) v tabeli enaka ena, v nasprotnem primeru pa je vrednost nič. Vrednosti na diagonalni so vedno nič, ker atomi med seboj niso povezani.

stolpci / vrstice	1	2	3	4	5	6	7	8	9	10	11
1	0	1	0	0	0	0	0	0	0	0	0
2	1	0	1	1	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0	0	0	0
4	0	1	0	0	1	0	0	0	0	0	0
5	0	0	0	1	0	1	0	0	0	1	0
6	0	0	0	0	1	0	1	0	0	0	0
7	0	0	0	0	0	1	0	1	0	0	0
8	0	0	0	0	0	0	1	0	1	0	1
9	0	0	0	0	0	0	0	1	0	1	0
10	0	0	0	0	1	0	0	0	1	0	0
11	0	0	0	0	0	0	0	1	0	0	0

Slika 6. Primer molekulskega grafa v obliki vezavne tabele za paracetamol. Številke atomov v molekuli (siva stolpec in vrstica). Povezava med atomoma dva in štiri (oranžno). Diagonala (modro).

```

1 void Graf::tabela () {
2     conn = new bool * [1000];
3     for (int i = 0; i < 1000; i++) {
4         conn [i] = new bool [1000];
5         memset (conn [i], 0, 1000 * sizeof (bool));
6     }
7     for (int i = 0; i < size; i++)
8         for (int j = 0; j < n[i].bond_size; j++) {
9             conn [i] [n[i].bond[j]] = true;
10        }
11    }
12 }

```

Slika 7. Podprogram za generiranje vezavne tabele. Število atomov v molekuli (*size*). Število vezi *i*-tega atoma (*n[i].bond_size*). Vezavna tabela (*conn*).

Podprogram na Sliki 7 najprej ustvari vezavno tabelo *conn* in vsem celicam dodeli vrednost nič. Nato se v for-zanki premika po nizu vseh težkih atomov in za vsak atom *i* zapiše enko v vezavno tabelo *conn*, če je atom *i* povezan z atomom *j*.

3.2.2 Generiranje 2D produktnih grafov

Algoritem iz molekulskih grafov referenčne in tarčne molekule ustvari molekulski produktni graf. Generiranje produktnega grafa poteka po naslednjem postopku. Program gre po vseh vozliščih prvega in drugega molekulskega grafa ter združuje vozlišča, ki predstavljajo atome z enakim tipom, na primer C.3 s C.3, N.am z N.am, in tako naprej, v točke produktnega grafa. Na Sliki 8 je primer produktnega grafa generiranega iz referenčne in tarčne molekule na Sliki 3. Vsaka točka produktnega grafa je tako sestavljena iz dveh komponent: vozlišča iz referenčnega grafa in vozlišča iz tarčnega grafa. Prva komponenta predstavlja atom iz referenčne, druga pa iz tarčne molekule. Iz algoritma tvorbe produktnega grafa sledi, da so atomi znotraj ene točke istega tipa. Povezave med vozlišči produktnega grafa algoritem ustvari na tri različne spodaj opisane variante. Podprogram za generiranje produktnega grafa po varianti 1 je na Sliki 9.

3.2.2.1 Varianta 1

- Če med prvo komponento referenčne molekule in prvo komponento tarčne molekule obstaja povezava in če med drugo komponento referenčne molekule in drugo komponento tarčne molekule tudi obstaja povezava, potem algoritem poveže

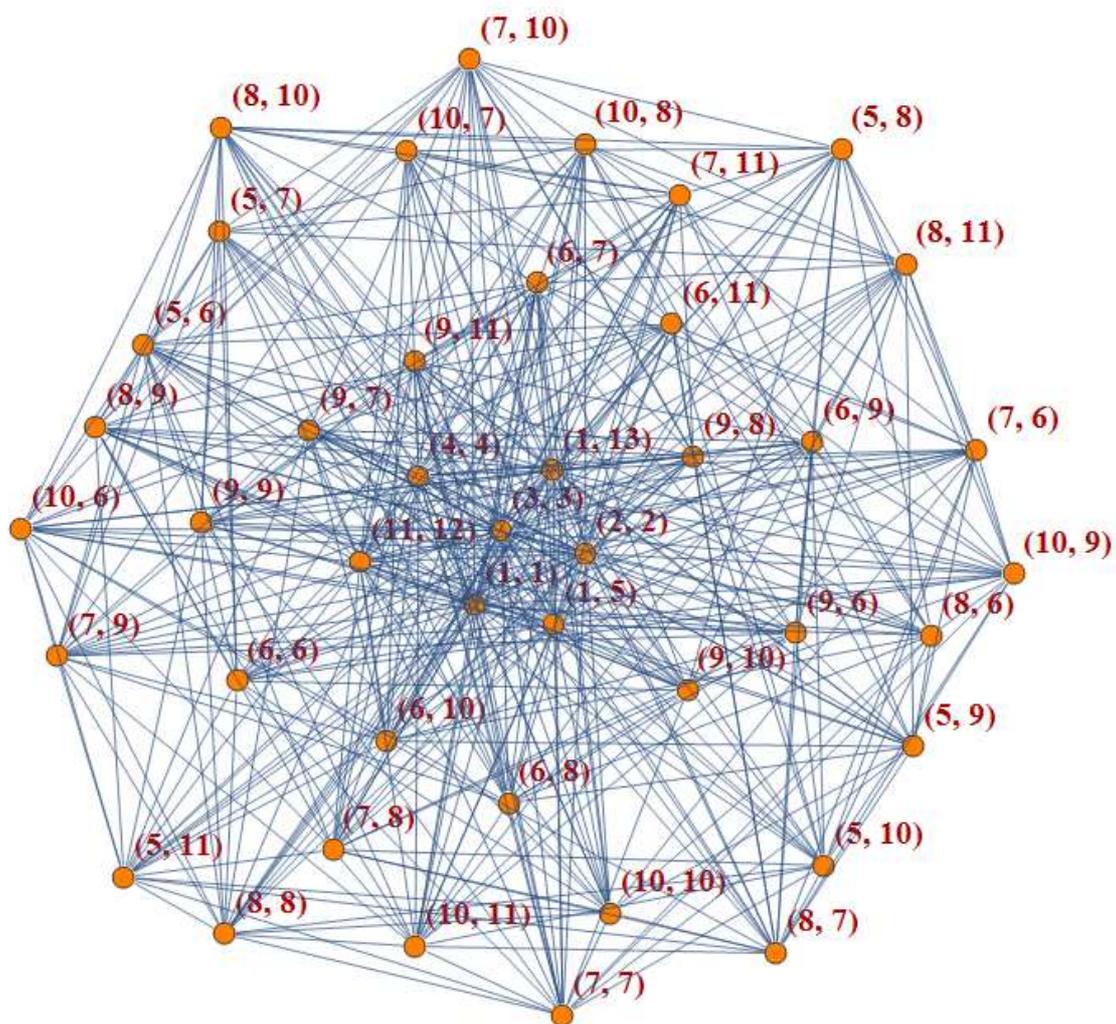
ustrezajoči točki produktnega grafa (Slika 10).

- Če med prvo komponento referenčne molekule in prvo komponento tarčne molekule ne obstaja povezava in če med drugo komponento referenčne molekule in drugo komponento tarčne molekule ne obstaja povezava, potem sta ustrezajoči točki produktnega grafa tudi povezani.
- Če med dvema komponentama obstaja povezava, med dvema pa ne, potem točki produktnega grafa nista povezani.

3.2.2.2 Varianti 2 in 3

Varianti 2 in 3 smo razvili, da bi zmanjšali produktni graf v primerjavi z varianto 1 in s tem pohitrili iskanje maksimalne klike v njem. Za izvedbo variant 2 in 3 najprej izvedemo Floyd-Warshall-ov algoritem (poglavje 3.2.3). Pravila za generiranje povezav v produktnem grafu v varianti 1 zamenjamo s sledečima praviloma:

- Če je najkrajša pot med prvima komponentama vozlišča produktnega grafa in drugima komponentama vozlišča produktnega grafa enaka, potem povezava obstaja (varianta 2).
- Če je razlika najkrajših poti med prvima komponentama vozlišča produktnega grafa in drugima komponentama vozlišča produktnega grafa manjša ali enaka dve, potem povezava obstaja (varianta 3).



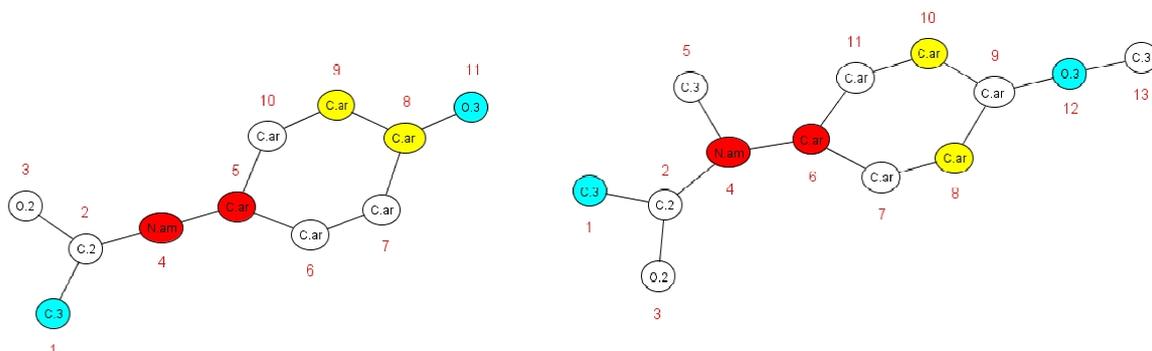
Slika 8. Primer produktne grafa iz molekularnih grafov paracetamola in N-(4-metoksifenil)-N-metilacetamida.

```

1 void GrafP::prod_graph (const Graf &g1, const Graf &g2) {
2     for (int i = 0; i < g1.get_size(); i++) {
3         for (int j = 0; j < g2.get_size(); j++) {
4             if (g1.n[i].name == g2.n[j].name) {
5                 np[size].n1 = i;
6                 np[size].n2 = j;
7                 size++;
8             }
9         }
10    }
11    for (int a = 0; a < size; a++) {
12        for (int b = a+1; b < size; b++) {
13            if (np[a].n1 != np[b].n1 && np[a].n2 != np[b].n2) {
14                if ((g1.conn [np[a].n1] [np[b].n1] && g2.conn [np[a].n2]
15 [np[b].n2]) || (!g1.conn [np[a].n1] [np[b].n1] && !g2.conn [np[a].n2] [np[b].n2]))
16                {
17                    np[a].edge[np[a].edge_size++] = b;
18                    np[b].edge[np[b].edge_size++] = a;
19                }
20            }
21        }
22    }
23 }

```

Slika 9. Podprogram za generiranje 2D produktne grafa. Molekulska grafa referenčne in tarčne molekule ($g1$, $g2$), število vozlišč oziroma atomov molekulskega grafa referenčne in tarčne molekule ($g1.get_size$, $g2.get_size$), tip i -tega atoma referenčne in tarčne molekule ($g1.n[i].name$, $g2.n[i].name$), prva komponenta vozlišča produktne grafa ($np[size].n1$), druga komponenta vozlišča produktne grafa ($np[size].n2$), vrednost celice vezavne tabele za prvo komponento vozlišča a in prvo komponento vozlišča b ($g1.conn[np[a].n1][np[b].n1]$), vrednost celice vezavne tabele za drugo komponento vozlišča a in drugo komponento vozlišča b ($g2.conn[np[a].n2][np[b].n2]$).



Slika 10. Vozlišči označeni rdeče sta v obeh grafih (levem in desnem) povezani. Zato povezava med točkama (4, 4) in (5, 6) v produktnem grafu obstaja. Vozlišči označeni s

svetlo modro nista povezani v nobenem grafu. Zato tudi med točkama (1, 1) in (11, 12) povezava v produktnem grafu obstaja. Točki (8, 8) in (9, 10) v produktnem grafu nista povezani, saj med vozliščema označenima z rumeno v levem grafu povezava obstaja, v desnem pa ne.

3.2.3. Floyd-Warshall-ov algoritem

Ta algoritem uporabljamo za iskanje najkrajše poti med vsemi pari vozlišč v referenčnem in tarčnem molekularnem grafu (26). Za uporabo tega algoritma moramo predhodno ustvariti novo tabelo *dist* (Slika 11), katere vrednosti celic bodo vsebovale najkrajše razdalje med vozlišči. Vhodna tabela za ta algoritem je enaka vezavni tabeli (poglavje 3.2.1), s to razliko, da vse ničle razen diagonalnih zamenjamo z neskončno. Podprogram, ki izvede Floyd-Warshall-ov algoritem, je na Sliki 12. Algoritem vrne izhodno tabelo (Slika 13), v kateri so vrednosti najkrajših poti (število vezi) med atomi molekule.

stolpci / vrstice	1	2	3	4	5	6	7	8	9	10	11
1	0	1	∞								
2	∞	0	1	1	∞						
3	∞	1	0	∞							
4	∞	1	∞	0	1	∞	∞	∞	∞	∞	∞
5	∞	∞	∞	1	0	1	∞	∞	∞	1	∞
6	∞	∞	∞	∞	1	0	1	∞	∞	∞	∞
7	∞	∞	∞	∞	∞	1	0	1	∞	∞	∞
8	∞	∞	∞	∞	∞	∞	1	0	1	∞	1
9	∞	1	0	1	∞						
10	∞	∞	∞	∞	1	∞	∞	∞	1	0	∞
11	∞	1	∞	∞	0						

Slika 11. Vhodna tabela za Floyd-Warshall-ov algoritem na primeru paracetamola.

```

1 void Graf::FV_algorithm () {
2     for (int k = 0; k < size; k++) {
3         for (int i = 0; i < size; i++) {
4             for (int j = 0; j < size; j++) {
5                 if (dist [i] [k] + dist [k] [j] < dist [i] [j]) {
6                     dist [i] [j] = dist [i] [k] + dist [k] [j];
7                 }
8             }
9         }
10    }
11 }

```

Slika 12. Podprogram za Floyd-Warshall-ov algoritem.

stolpci / vrstice	1	2	3	4	5	6	7	8	9	10	11
1	0	1	2	2	3	4	5	6	5	4	7
2	1	0	1	1	2	3	4	5	4	3	6
3	2	1	0	2	3	4	5	6	5	4	7
4	2	1	2	0	1	2	3	4	3	2	5
5	3	2	3	1	0	1	2	3	2	1	4
6	4	3	4	2	1	0	1	2	3	2	3
7	5	4	5	3	2	1	0	1	2	3	2
8	6	5	6	4	3	2	1	0	1	2	1
9	5	4	5	3	2	3	2	1	0	1	2
10	4	3	4	2	1	2	3	2	1	0	3
11	7	6	7	5	4	3	2	1	2	3	0

Slika 13. Izhodna tabela najkrajših poti za paracetamol po izvedbi Floyd-Warshallovega algoritma. Številke atomov molekule paracetamola (siva stolpec in vrstica), najkrajše poti (številke v celicah). Najkrajša pot med atomoma tri in pet je dolžine tri (oranžno).

3.2.4 Generiranje 3D produktnih grafov

Za razliko od generiranja 2D produktnega grafa, algoritem za generiranje 3D produktnega grafa upošteva tudi prostorske koordinate atomov referenčne in tarčne molekule. Algoritem iz molekulskih grafov referenčne in tarčne molekule najprej po enakem

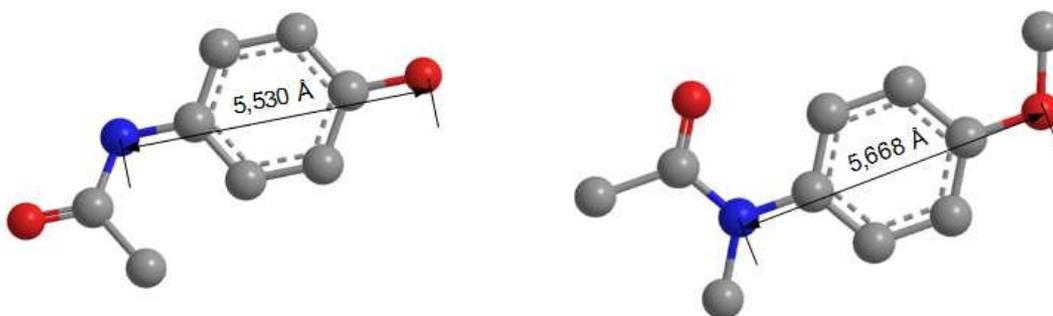
postopku kot pri generiranju 2D produktnega grafa ustvari vozlišča produktnega grafa. Povezave med vozlišči 3D produktnega grafa pa algoritem ustvari na drugačen način (Slika 14):

- Algoritem najprej izračuna prostorsko razdaljo D_R med prvima komponentama (vozliščema v referenčnem molekularnem grafu) in razdaljo D_T med drugima komponentama (vozliščema v tarčnem molekularnem grafu) po naslednjih enačbah:

$$D_R = \sqrt{(x_{R1} - x_{R2})^2 + (y_{R1} - y_{R2})^2 + (z_{R1} - z_{R2})^2}$$

$$D_T = \sqrt{(x_{T1} - x_{T2})^2 + (y_{T1} - y_{T2})^2 + (z_{T1} - z_{T2})^2}$$

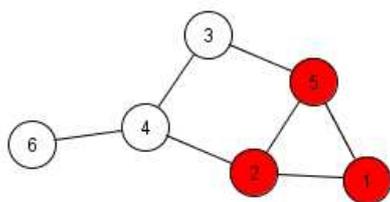
- Nato izračuna absolutno razliko R med D_R in D_T : $R = |D_R - D_T|$
- Če je R manjši od neke vnaprej določene vrednosti, na primer 1 Å, potem povezava med ustreznima vozliščema produktnega grafa obstaja v nasprotnem primeru pa ne.



Slika 14. Izračun razdalje med prvima komponentama produktnega grafa (levo) in med drugima komponentama produktnega grafa (desno). V tem primeru je razlika med razdaljama $R = |5,530 \text{ Å} - 5,668 \text{ Å}| = 0,138 \text{ Å}$. Ker je $R < 1 \text{ Å}$, povezava obstaja.

3.2.5 Algoritem za iskanje maksimalne klike

Klika v splošnem grafu je tisti podgraf (podmnožica vozlišč in povezav), v katerem so vsa vozlišča paroma povezana. Maksimalna klika je tista klika v splošnem grafu, ki jo sestavlja največ vozlišč (Slika 15).



Slika 15. Maksimalna klika v splošnem grafu. Rdeče označena vozlišča 1, 2, 5 so vsa med seboj povezana (vozlišče 1 je povezano z vozliščem 2 in 5, vozlišče 2 je povezano z vozliščem 1 in 5, vozlišče 5 je povezano z vozliščema 1 in 2). Torej vozlišča 1, 2 in 5 sestavljajo kliko velikosti tri.

Algoritem za iskanje maksimalne klike MaxCliqueDyn (24) na Sliki 16 poišče maksimalno kliko v 2D ali 3D molekulskem produktne grafu (Slika 17). Maksimalna klika v produktne grafu namreč predstavlja največjo skupno podstrukturo referenčne in tarčne molekule. Algoritem za iskanje maksimalne klike uporablja dve množici, Q in Q_{\max} , kjer množico Q sestavljajo vozlišča trenutno rastoče klike, Q_{\max} pa vozlišča največje klike, ki jo je algoritem do sedaj našel. Algoritem začne s prazno množico Q , v katero rekurzivno dodaja vozlišča iz produktne grafu. To poteka toliko časa, dokler ne more več najti klike z več vozlišči. Naslednje vozlišče, ki se bo dodalo v Q , se izbere iz množice R , ki je podmnožica množice V , ki vsebuje vsa vozlišča. Na začetku velja, da je $R = V$. V vsakem koraku algoritem izbere vozlišče $p \in R$ z maksimalno barvo $C(p)$ in to vozlišče izbriše iz R . $C(p)$ je zgornja meja velikosti maksimalne klike v dobljenem R . Če vsota $|Q| + C(p)$ kaže, da je možno v R najti večjo kliko kot tisto, ki je trenutno v Q_{\max} , potem se vozlišče p doda množici Q . Nato se izračuna nova množica kandidatov, to je ustrezno pobarvanih vozlišč: $R \cap \Gamma(p)$, pri čemer je $\Gamma(p)$ množica vseh vozlišč, ki so povezana z vozliščem p , in se kot argument poda rekurzivno klicani MaxCliqueDyn proceduri. Če je $R_p = 0$ in $|Q| > |Q_{\max}|$, je trenutna klika večja kot do sedaj največja najdena klika, zato se vozlišča iz Q prekopirajo v Q_{\max} . Algoritem nazadnje odstrani p iz Q in vzame novo vozlišče iz R . Procedura se ponavlja, dokler v množici R ne zmanjka vozlišč.

3.3. Izpis algoritma za virtualno reševanje na osnovi liganda

Algoritem za iskanje maksimalne klike predstavljen v prejšnjem poglavju vrne kot rezultat največji skupni podgraf primerjanih molekulskih grafov. Največji skupni podgraf je največja skupna podstruktura iz atomov in vezi v primerjani referenčni in tarčni molekuli. Stopnjo podobnosti primerjanih molekul izračunamo iz števila atomov v skupni podstrukturi in števila vseh atomov v obeh molekulah s tako imenovanim Tanimotovim koeficientom $T(R,T)$ po sledeči enačbi:

$$T(R,T) = \frac{N_S}{N_T + N_R - N_S}$$

pri tem je N_S število atomov, ki sestavljajo skupno podstrukturo, N_R je število atomov, ki sestavlja referenčno molekulo in N_T je število atomov, ki sestavlja tarčno molekulo.

Končni rezultat našega algoritma za 2D in 3D virtualno reševanje na osnovi liganda je mol2 datoteka, ki vsebuje referenčno in tarčno molekulo, in iz katere je razvidno, kateri atomi so si topološko in fizikalno-kemijsko podobni v obeh molekulah (Slika 18). Datoteka vsebuje tudi vrednost Tanimotovega koeficienta za primerjani molekuli.



Slika 18. Končni rezultat programa. Z rdečo je označena skupna najdena podstruktura obeh molekul. Spodaj je odebeljeno izpisan Tanimotov koeficient.

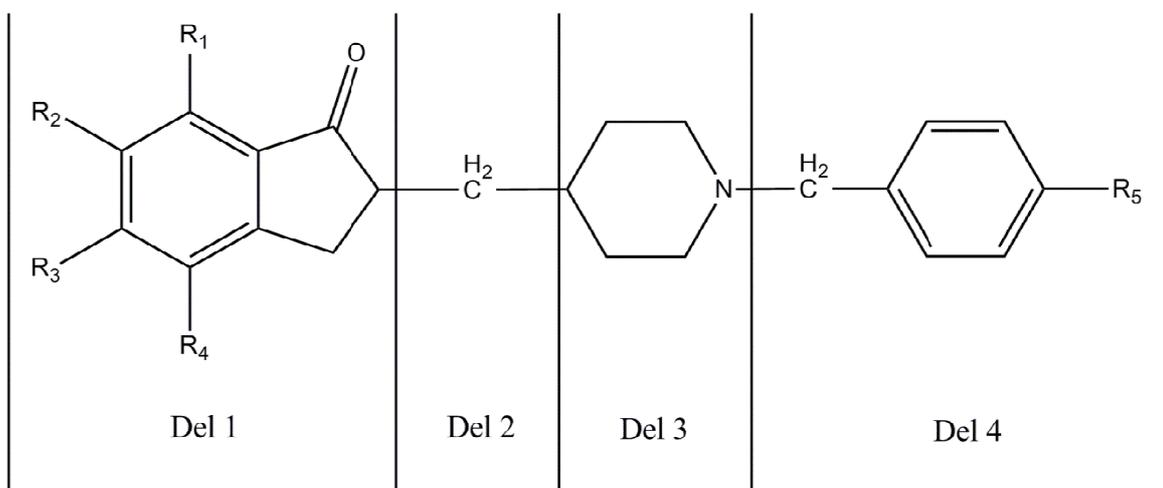
4. Rezultati

Pri 2D in 3D virtualnem reševanju na osnovi liganda s pomočjo našega programa smo uporabili približno dva milijona veliko ZINC-ovo Leads Now bazo spojin, ki je dostopna na naslovu: <http://zinc.docking.org/subsets/leads-now>. Baza vsebuje molekule, ki ustrezajo osnovnim kriterijem za spojine vodnice (27): molska masa med 250 in 350; log P je manjši ali enak 3,5; razmerje med polarno površino in vrtljivimi vezmi je manjše ali enako 7. Kot referenčno konformacijo donepezila smo vzeli njegovo biokativno konformacijo vezano v humano acetilholinesterazo najdeno v PDB (koda 4EYZ). Molekulo GUK662 smo dobili v

predhodno sidrani konformaciji v aktivnem mestu acetilholinesteraze.

4.1 Postopek izbire najdenih podobnih molekul

Z našim programom smo našli več sto podobnih molekul, zato smo za nadaljnji razvoj in testiranje izbrali le najzanimivejše. Molekule smo izbirali glede na njihov Tanimotov koeficient (T), za vsako kombinacijo referenčne molekule, variante in dimenzionalnosti (2D ali 3D) smo izbrali tak mejni T , da je imelo enak ali večji T približno sto molekul. Molekule smo izbrali tudi glede znani SAR donepezila, hkrati pa smo poskušali najti drugačen skelet, kot je v referenčni molekuli (28). V nadaljevanju na kratko opišemo SAR za molekulo donepezila, ki jo navidezno razdelimo na štiri dele (Slika 19).



Slika 19. Donepezilu podobna molekula, razdeljena na štiri dele.

V prvem delu molekule se je kot pomemben za delovanje izkazal biciklični obroč. Z povečanjem bicikla, tako da je drugi obroč šestčlenski ali sedemčlenski, aktivnost močno pade. Metoksi skupini na mestu R₂ in R₃ povečata aktivnost molekule. Za visoko aktivnost je bistvena tudi ketonska skupina.

Če v drugem delu molekule direktno povežemo indanonski in piperidinski obroč, potem aktivnost močno pade. Pri povezavah, ki niso direktne, velja naslednje zaporedje: propilen > metilen > pentilen > etilen > butilen. Aktivnost molekule je manjša, če je obroč iz dela 1 z delom 2 povezan z dvojno vezjo.

V tretjem delu je aktivnost molekule najboljša, če je dušikov atom na mestu, kot je prikazan na sliki SAR. Za delovanje je pomembna bazičnost tega dušikovega atoma in njegova oddaljenost od ketonske skupine iz prvega dela molekule.

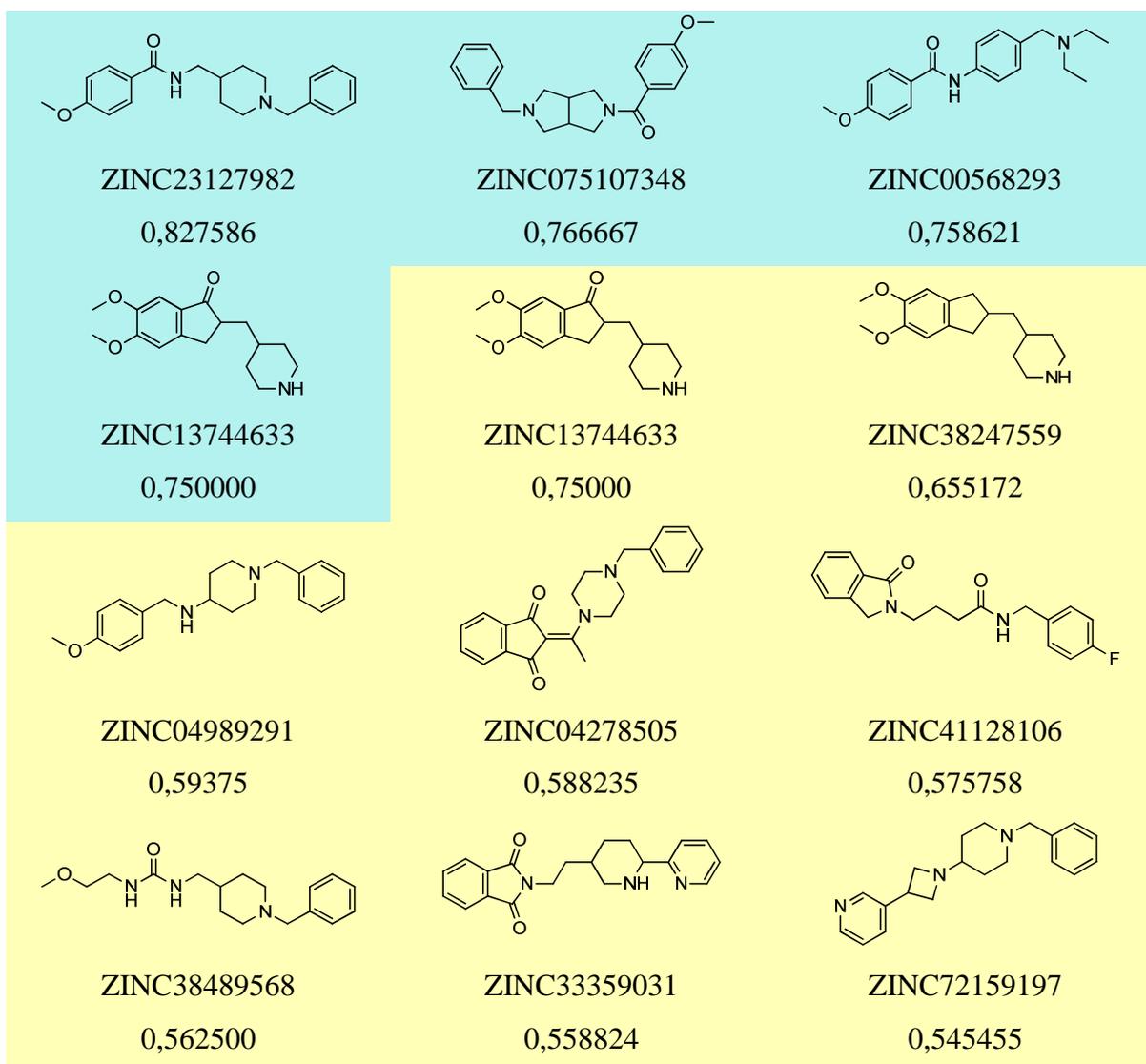
Aktivnost molekule je mogoče povečati, če v četrtem delu na tretje mesto benzilne skupine

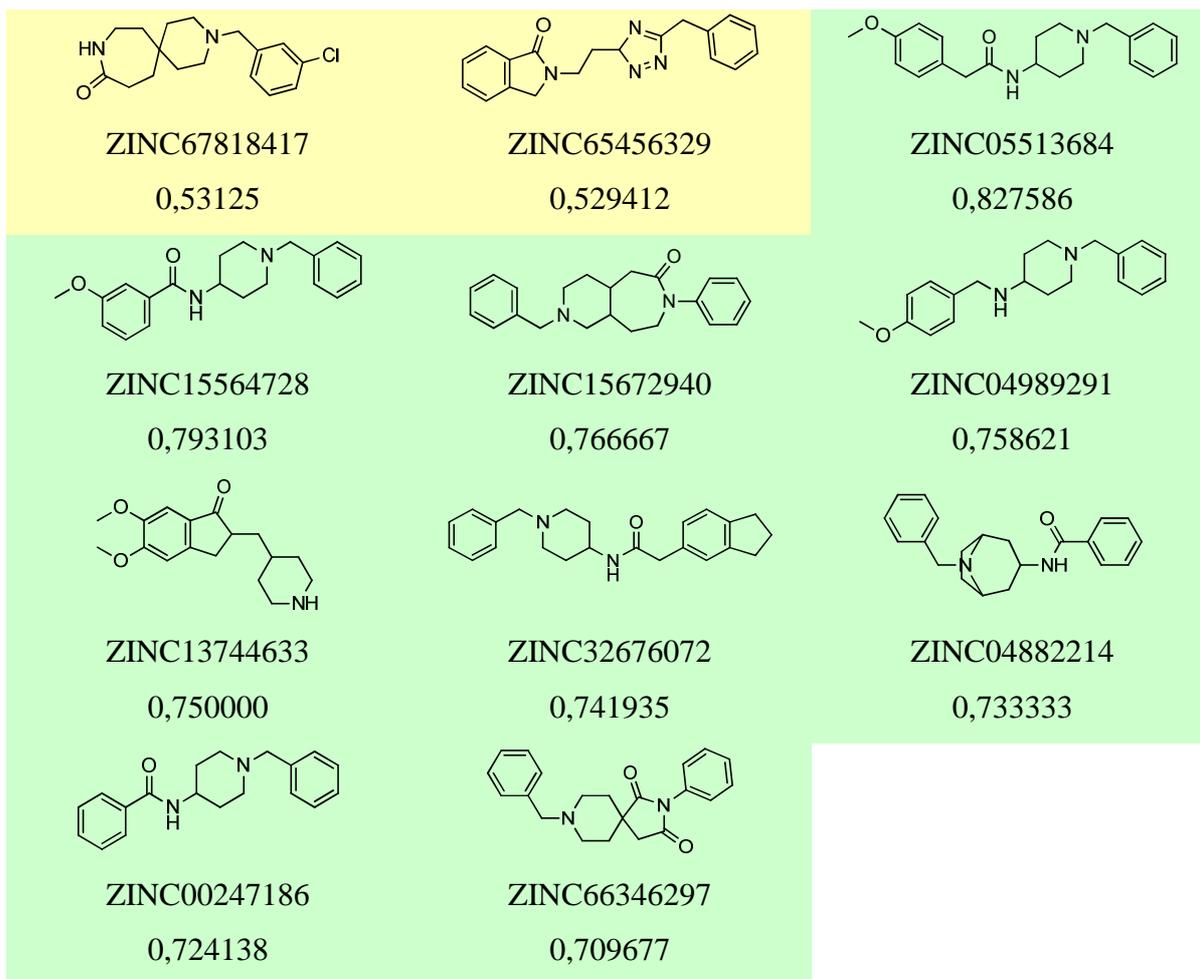
priprnemo metilno skupino. Odstranitev benzilne skupine močno zmanjša aktivnost, se pa aktivnost ohrani, če jo zamenjamo s cikloheksilmetilno. Aktivnost se tudi zmanjša, če benzilno skupino zamenjamo z fenetilno ali 2-naftilno.

4.2 2D virtualno reševanje

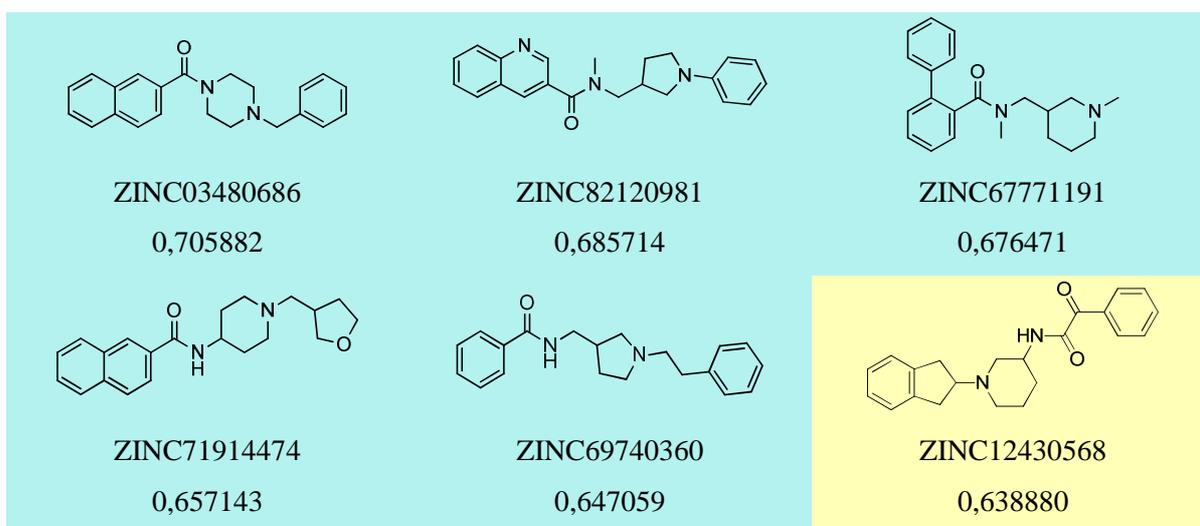
Izbrane molekule, ki so rezultat 2D virtualnega reševanja za referenčno molekulo donepezil in GUK662 so na Sliki 20 in 21. Molekule z zanimivejšimi novimi skeleti, ki jih je program našel z virtualnim reševanjem donepezila so na primer: ZINC075107348, ZINC65456329, ZINC67818417, ZINC33359031, ZINC66346297, ZINC04882214 in ZINC15672940.

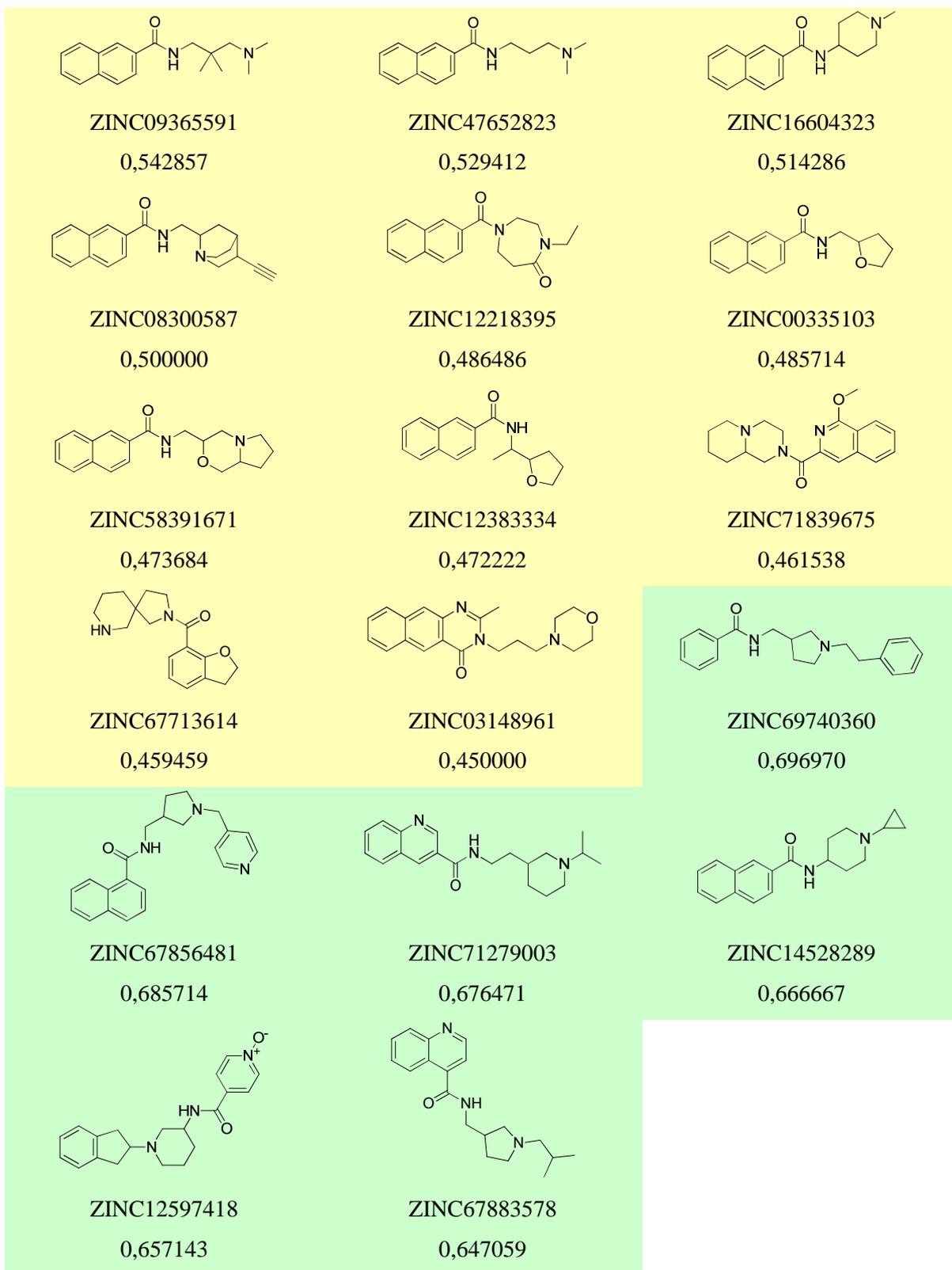
V primeru GUK662 pa molekule kot so: ZINC03148961, ZINC67713614, ZINC58391671, ZINC12218395 in ZINC08300587.





Slika 20. Izbrane molekule virtualnega rešetanja, kjer je referenčna molekula donepezil. Molekule, ki smo jih dobili z varianto 1, so označene z modro, z varianto 2 rumeno, z varianto 3 pa zeleno.





Slika 21. Izbrane molekule virtualnega rešetanja, kjer je referenčna molekula GUK662. Molekule, ki smo jih dobili z varianto 1, so označene z modro, z varianto 2 rumeno, z varianto 3 pa zeleno.

4.2.1 Hitrost 2D virtualnega rešetanja

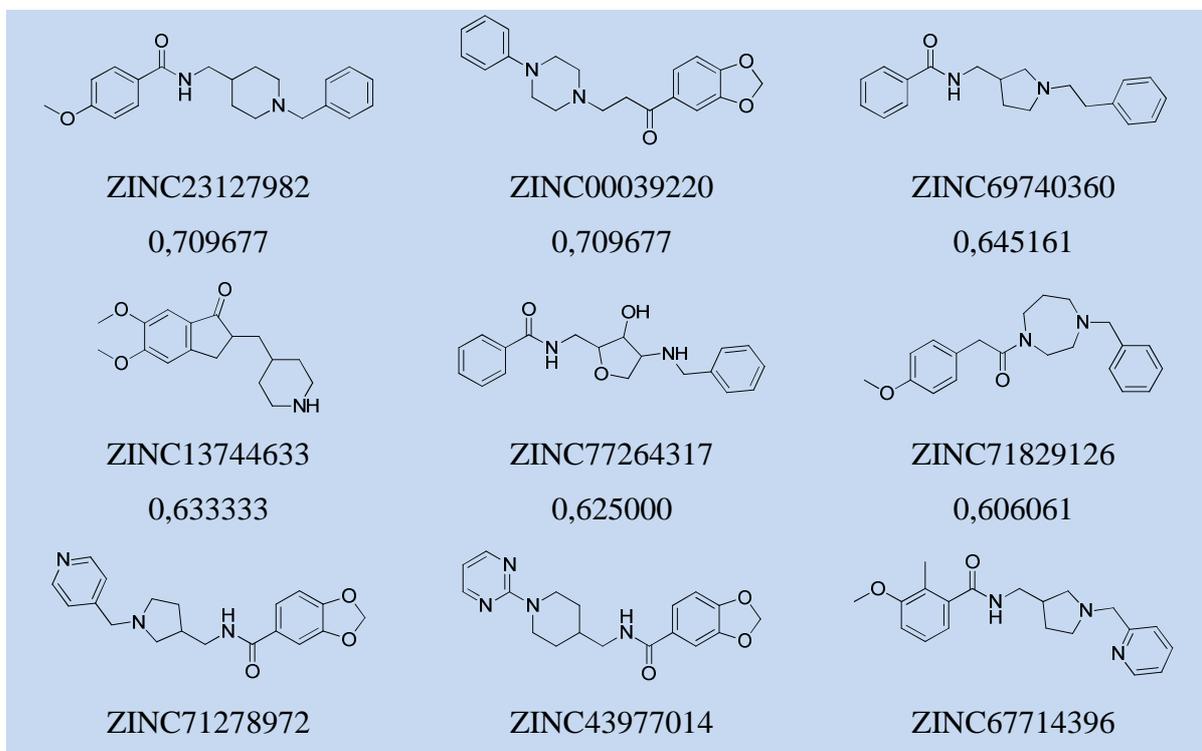
Program smo za vse tri variante zagnali na osmih jedrih (štirih pravih in štirih virtualnih) procesorja Intel Core i7-2600 CPU @ 3.40GHz. V Tabeli 1 so prikazani časi izvajanja programa za vse tri variante za virtualno rešetanje na primeru donepezila in GUK662 proti približno dva milijona spojin veliki ZINC-ovi bazi.

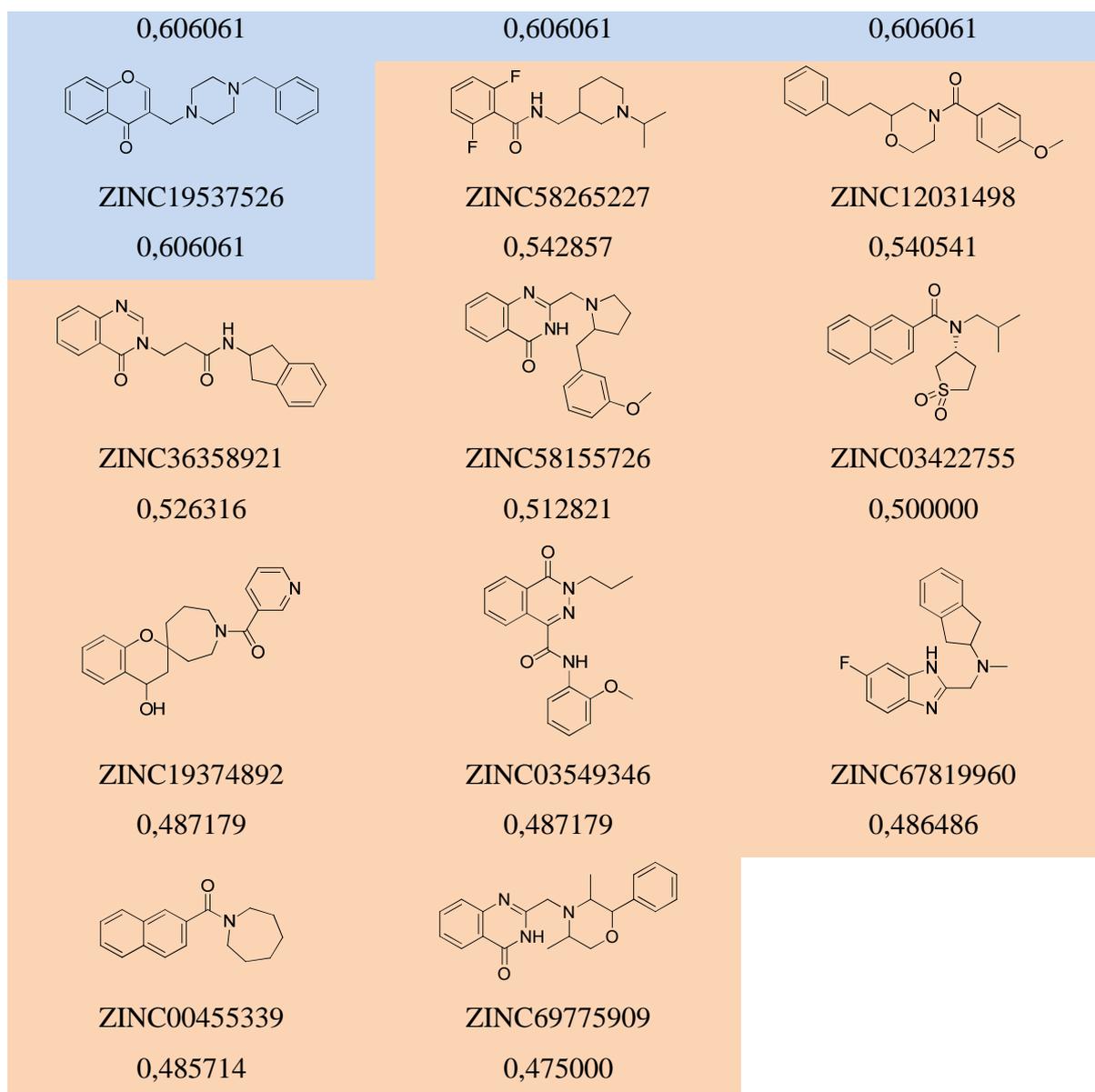
Tabela 1. Izmerjene hitrosti za donepezil in GUK662 za različne variante za 2D virtualno rešetanje.

Varianta	1	2	3
Čas [h] (donepezil)	10,9	2,3	2,3
Čas [h] (GUK662)	74,5	3,2	3,1

4.3 3D virtualno rešetanje

Za 3D virtualno rešetanje na osnovi liganda smo uporabili enako bazo spojin kot za 2D, le da smo predhodno za vsako spojino ustvarili več konformacij (do 300), s čimer smo upoštevali tudi fleksibilnost molekul. Iz Slike 22 so razvidni zanimivejši skeletni preskoki pri spojinah, ki imajo Tanimotov koeficient nižji ali enak 0,625000 za donepezil in pri vseh navedenih spojinah GUK662.





Slika 22. Izbrane molekule na podlagi 3D virtualnega rešetanja. Molekule, ki so bile dobljene z rešetanjem, kjer je bila referenčna molekula donepezil, so označene z modro, tam, kjer pa je bila referenčna molekula GUK662 pa z oranžno.

4.3.1 Hitrost 3D virtualnega rešetanja

Program z opcijo za 3D virtualno rešetanje smo poganjali na 22 jedrih na dveh računalnikih: prvi računalnik je imel dva procesorja Intel Xeon CPU E5-2630 0 @ 2.30GHz, drugi pa dva procesorjema Intel Xeon CPU E5520 @ 2.27GHz. Na vsakem računalniku smo uporabljali po 11 jeder (6 pravih in 5 virtualnih). Pregledovanje baze je trajalo 7 dni neodvisno od velikosti referenčne molekule.

5. Razprava

Zaradi naraščajočega števila molekul v virtualnih bazah spojin, kot je na primer ZINC baza, smo se odločili razviti nov, hiter in učinkovit program za virtualno reševanje velikih baz malih molekul. Program kot vhodne podatke vzame dve molekuli, ki ju želimo primerjati, nato ju prevede v molekulska grafa, tako da vsak atom postane vozlišče, vsaka vez pa povezava. Z algoritmom za iskanje maksimalne klike primerja molekulska grafa in poišče njun največji skupni podgraf, ki predstavlja največjo skupno podstrukturo obeh molekul. Program omogoča tako 2D kot tudi 3D virtualno reševanje. V 2D primeru smo razvili tri variante algoritma, ki se razlikujejo v tem, kako tvorimo produktni graf iz molekulskih grafov. Varianti 2 in 3 sta se izkazali za bistveno hitrejši od izhodne variante. Variante se med seboj tudi razlikujejo po Tanimotovih koeficientih, ki jih pripišejo paroma molekul. Program smo testirali na dveh referenčnih molekulah – donepezilu in GUK662, inhibitorjema acetilholinesteraze, s katerima smo pregledali bazo ZINC. Našli smo več sto podobnih molekul, in med njimi izbrali tiste z najbolj raznolikimi skeleti in ustrezno strukturo glede na znan SAR donepezila (28).

Prednost našega programa pred nekaterimi drugimi programi za virtualno reševanje je v tem, da naš program uporablja algoritem za iskanje maksimalne klike, medtem ko drugi programi iščejo podobnosti med molekulami z Bron-Kerbosch-evim algoritmom. Slabost slednjega je, da najde vse nevsebovane klike v grafu in ne le maksimalne, zaradi česar se ta algoritem izkaže za počasnejšega. Naš program se od podobnih razlikuje tudi v tem, da ne tvori reduciranih grafov, v katerih je eno vozlišče sestavljeno iz več atomov. V nasprotju s temi programi, v našem programu vsako vozlišče predstavlja en atom. S tem molekul ne poenostavljamo, zato dosežemo bolj občutljivo primerjavo med molekulami.

5.1 2D virtualno reševanje

5.1.1 Primerjava prve, druge in tretje variante

5.1.1.1 Primerjava hitrosti

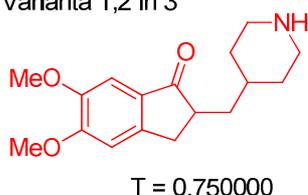
Iz Tabele 1 je razvidno, da sta varianti 2 in 3 približno petkrat hitrejši kot varianta 1. Do razlik prihaja zaradi drugačne tvorbe produktnega grafa. Varianti 2 in 3 smo razvili, da bi zmanjšali produktni graf v primerjavi z varianto 1 in s tem pohitrili iskanje maksimalne klike v njem. Ugotovili smo, da pri varianti 1 večji del produktnega grafa opisuje več razdrobljenih podobnosti, ki so na popolnoma različnih delih v primerjanih molekulah

(Slika 23, spodaj). Zato smo pri varianti 2 in 3 uvedli dodatni pogoj, pri tvorbi produktnega grafa, s katerimi upoštevamo razdaljo (najkrajša pot) med atomi (poglavje 3.2.2). Ker sta se varianti 2 in 3 izkazali za bistveno hitrejši in dajeta tudi bolj raznolike skelete, je smiselno v nadaljnje uporabljati le ti dve varianti.

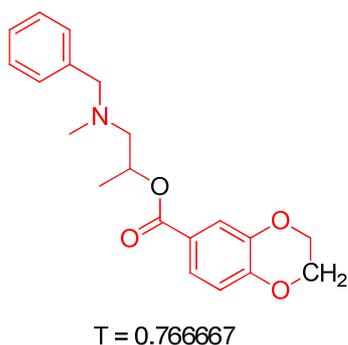
5.1.1.2 Primerjava najdenih podobnih molekul

Za varianti 2 in 3 se je izkazalo, da tudi molekulam z bolj raznolikimi skeleti pripiseta relativno visoke Tanimotove koeficiente (Slika 20 in 21), čeprav so pri teh dveh variantah Tanimotovi koeficienti v splošnem nižji kot pri varianti 1. Tretja varianta tolerira substrukture, ki se do dve vezi razlikujeta med seboj, posledično tudi za enake molekule da višji Tanimotov koeficient, druga varianta pa tega tolerančnega faktorja nima, zato so Tanimotovi koeficienti nižji. Izkazalo se je tudi, da bolj kot je tarčna molekula podobna referenčni molekuli, bolj so si Tanimotovi koeficienti med seboj podobni pri vseh treh variantah (Slika 23). Molekulam, ki niso podobne referenčni molekuli, variante pripisejo zelo različne Tanimotove koeficiente.

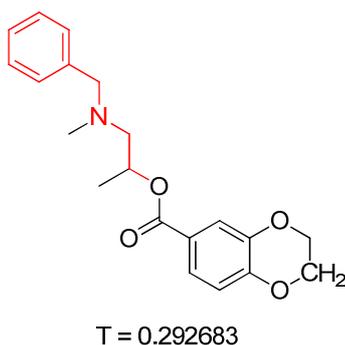
Varianta 1,2 in 3



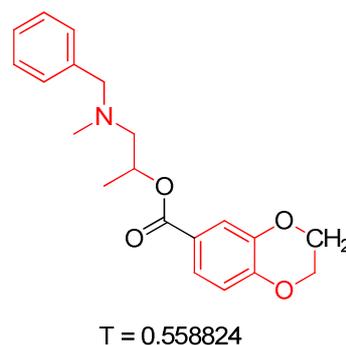
Varianta 1



Varianta 2



Varianta 3



Slika 23. Zgornji molekuli, ki je zelo podobna referenčni molekuli donepezilu, vse tri variante algoritma pripisejo enak Tanimotov koeficient. Vse tri variante prepoznajo enako skupno (rdeče) podstrukturo med referenčno in tarčno molekulo. Spodnji molekuli, ki ni preveč podobna donepezilu, vsaka varianta pripise drug Tanimotov koeficient. Z rdeče so označeni atomi, ki jih vsaka varianta prepozna kot del skupne podstrukture s tarčno

molekulo.

Tri variante algoritma lahko ocenijo podobnost dveh molekul različno. To se odraža v drugačnem povprečju Tanimotovih koeficientov med variantami (Tabela 2). Najvišje povprečje Tanimotovih koeficientov je pri obeh referenčnih molekulah dala varianta 1, nižje varianta 3, najnižje pa varianta 2. Zato moramo za vsako varianto posebej določiti mejno vrednost, od katere naprej obravnavamo molekule kot podobne. Pri donepezilu smo vzeli mejno vrednost pri varianti 1 0,750000 pri varianti 2 0,529412 in pri varianti 3 0,709677. Iz tabele 2 je razvidno tudi, da velikost referenčne molekule ne vpliva bistveno na povprečje Tanimotovih koeficientov (GUK662 je večja molekula kot donepezil). To si razlagamo tako, da v povprečju pri večjih molekulah najdemo tudi večje skupne podstrukture, zato se hkrati povečata števec in imenovalec v enačbi za Tanimotov koeficient. Tanimotovi koeficienti so seveda odvisni tudi od uporabljene baze spojin, na primer, če bi namesto Leads Now uporabili Fragment-Like bazo, bi bili v povprečju Tanimotovi koeficienti manjši.

Tabela 2. Povprečja Tanimotovih koeficientov za primerjavo referenčnih molekul z ZINC bazo.

	Varianta 1	Varianta 2	Varianta 3
Donepezil	0,378221	0,213710	0,284349
GUK662	0,346431	0,212239	0,275142

5.2 3D virtualno rešetanje

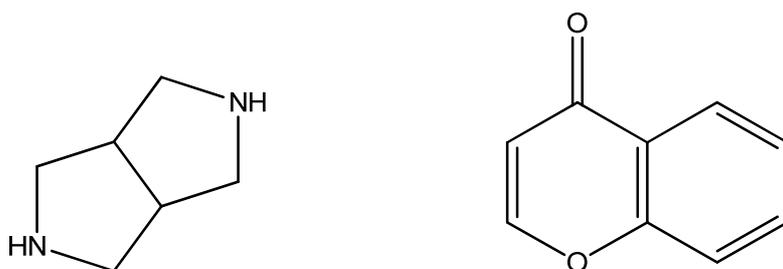
Tridimenzionalno virtualno rešetanje smo izvajali na spremenjeni ZINC bazi Leads Now (2 milijona spojin), v kateri je bila vsaka spojina v približno 300 različnih konformacijah. Konformacije so bile predhodno tvorjene za naše namene.

5.2.1 Primerjava s programom ShaEP

Naš program za 3D virtualno rešetanje smo primerjali s programom za 3D virtualno rešetanje ShaEP (29) na primeru ene konformacije donepezila in molekule ZINC00039220 (Slika 22). Programa vrneti podobne Tanimotove koeficiente, in sicer naš program 0,709677, ShaEP pa 0,75824. Povprečna časa treh meritev sta tudi podobna, in sicer za naš program 0,027 sekunde, za ShaEP pa 0,030 sekunde. Naš program daje torej primerljive rezultate in je celo nekoliko hitrejši od podobnega že obstoječega programa.

5.3 Primerjava najdenih molekul s podobnimi molekulami iz ZINC baze

Rezultate našega programa smo primerjali tudi s standardnim orodjem za iskanje podobnih molekul, ki je dostopno v ZINC bazi. S tem orodjem smo poiskali molekule, ki izkazujejo 70 % podobnost z molekulo donepezila, in ta izpis primerjali s podobnimi molekulami, ki smo jih našli z našim programom. Za donepezil nam ZINC-ova baza da 317 podobnih molekul. Od teh molekul smo z varianto 1 našega algoritma našli dve enaki molekuli (ZINC13744633 in ZINC32888401), z varianto 2 tri (ZINC44123903, ZINC38247559 in ZINC13744633) ter z varianto 3 eno (ZINC13744633) enako molekulo. Z našim programom torej najdemo zelo drugačne podobne molekule, kot so tiste, ki nam jih predlaga ZINC-ova baza. Naš program za razliko od ZINC-ove baze vrne tudi podobne molekule z različnimi skeleti, kot je skelet referenčne molekule. Primer skeletnega preskoka sta na primer molekuli ZINC075107348 (Slika 20) in ZINC19537526 (Slika 22), ki vsebujeta oktahidropirol[3,4-c]pirolsko in 4H-kromen-4-onsko strukturo (Slika 24).



Slika 24. Oktahidropirol[3,4-c]pirol (levo) in 4H-kromen-4-on (desno).

5.4 Predlogi za nadaljnje delo

V prihodnosti načrtujemo optimizirati program, in sicer:

- Vozlišče v molekulskem grafu bo lahko tudi funkcionalna skupina (sedaj atom).
- Pohitrili bomo 3D verzijo programa, in sicer tako, da bo trenutno najdena velikost maksimalne klike predhodne konformacije molekule predstavljala spodnjo mejo za nadaljnje konformacije.
- Izvedli bomo bolj obsežno virtualno reševanje na večjih bazah molekul in z več referenčnimi molekulami
- Program bomo preiskusili tudi na DUD bazi (angl. *A Directory of Useful Decoys*), ki se uporablja za testiranje programov za virtualno reševanje na osnovi liganda (30).

6. Sklep

Zaradi naraščanja števila molekul v prosto dostopnih bazah molekul, kot je na primer ZINC baza, smo se odločili, da bom razvili program, ki omogoča hitro 2D in 3D virtualno reševanje na osnovi liganda. Novo razviti program strukturi dveh primerjanih molekul, referenčne in tarčne molekule, pretvori v molekulska grafa in s pomočjo algoritma za iskanje maksimalne klike poišče njuno največjo skupno podstrukturo ter s Tanimotovim koeficientom oceni stopnjo njune podobnosti. Program smo preskusili na dveh referenčnih molekulah, acetilholinesteraznih inhibitorjih, in sicer donepezilu in GUK662. Referenčni molekuli smo primerjali z več kot 2 milijona tarčnimi molekulami iz 2D in 3D ZINC-ove Leads Now baze. Pri 3D reševanju smo tudi predhodno za vsako tarčno molekulo ustvarili okoli 300 konformacij. Pri 2D reševanju smo preizkusili tri različne variante tvorbe produktnega grafa, za katere se je izkazalo, da se močno razlikujejo v hitrosti delovanja in oceni podobnosti molekul. Program je tako pri 2D kot tudi 3D reševanju visoko ocenil tudi molekule, pri katerih je prišlo do skeletnega preskoka. V prihodnosti bomo te molekule kupili in preizkusili na bioloških sistemih kot inhibitorje acetilholinesteraze.

Literatura

1. Patrick G L: Medicinal Chemistry, 5th edition, Oxford University Press, Oxford, 2013.
2. Schneider G, K -H Baringhaus: Molecular Design: Concepts and Applications, Wiley-VCH, Weinheim, 2008: 87-88.
3. Bernstein F C, Koetzle T F, Williams G J, Meyer Jr. E F, Brice M D, Rodgers J R, Kennard O, Shimanouchi T, Tasumi M: The protein data bank: A computer-based archival file for macromolecular structures. *Journal of molecular biology* 1977; 112: 535-42.
4. Reddy A S, Pati S P, Kumar P P, Pradeep H N, Sastry G N: Virtual Screening in Drug Discovery - A Computational Perspective. *Current Protein and Peptide Science* 2007; 8(4): 329-51.
5. Lemman C, Lengauer T, Klebe G: FLEXS: a method for fast flexible ligand superposition. *Journal of Medicinal Chemistry* 1998; 41(23): 4502-20.
6. Rush T S, Grant J A, Mosyak L, Nicholls A: A shape-based 3D scaffold hopping method and its application to a bacterial protein-protein interaction. *Journal of medicinal chemistry* 2005; 48(5): 1489-95.
7. Jones G, Willett P: GASP: genetic algorithm superimposition program. *Pharmacophore Perception, Development, and Use in Drug Design* 2000. Güner, OF, ed.: 85-106.
8. Grant J A, Pickup B T: A Gaussian description of molecular shape. *The Journal of Physical Chemistry* 1995; 99(11): 3503-10.
9. Martin E J, Blaney J M, Siani M A, Spellmeyer D C, Wong A K, Moos W H: Measuring diversity: experimental design of combinatorial libraries of drug discovery. *Journal of medicinal chemistry* 1995; 38(9): 1431-36.
10. Xue L, Bajorath J: Molecular descriptors in chemoinformatics, computational combinatorial chemistry, and virtual screening. *Combinatorial Chemistry & High Throughput Screening* 2000; 3(5): 363-72.
11. Xue L, Godden J W, Stahura F L, Bajorath J: Design and evaluation of a molecular fingerprint involving the transformation of property descriptor values into a binary classification scheme. *Journal of chemical information and computer sciences*; 43(4): 1151-7.
12. Lengauer T, Lemmen C, Rarey M, Zimmermann M: Novel technologies for virtual

- screening. *Drug discovery today* 2004; 9(1): 27-34.
13. <http://www.daylight.com/dayhtml/doc/theory/theory.finger.html>
 14. Butina D: Unsupervised data base clustering based on daylight's fingerprint and Tanimoto similarity: A fast and automated way to cluster small and large data sets. *Journal of Chemical Information and Computer Sciences* 1999; 39(4): 747-50.
 15. http://www.biosolveit.com/FTrees/download/ftrees_ug.pdf
 16. Hessler G, Zimmermann M, Matter H, Evers A, Naumann T, Lengauer T, Rarey M: Multiple-ligand-based virtual screening. *Journal of medicinal chemistry* 2005; 48(21): 6575-84.
 17. Böcker A, Schneider G, Teckentrup A: NIPALSTREE: a new hierarchical clustering approach for large compound libraries and its application to virtual screening. *Journal of chemical information and modeling* 2006; 46(6): 2220-9.
 18. Martin Y C: DISCO: what we did right and what we missed. *Pharmacophore Perception, Development, and Use in Drug Design* 2000; La Jolla, CA, International University Line: 51-66.
 19. Yang S Y: Pharmacophore modeling and applications in drug discovery: changes and recent advances. *Drug discovery today* 2010; 15(11): 444-50.
 20. Stanley Y S, Drewry D H, Gerritz S W: Optimization of focused chemical libraries using recursive partitioning. *Combinatorial Chemistry & High Throughput Screening* 2002; 5(2): 125-33.
 21. http://accelrys.com/resource-center/case-studies/archive/studies/smart_virtual_screening_part2.pdf
 22. Rusinko A, Farnen M W, Lambert C G, Brown P L, Young S S: Analysis of a Large Structure/Biological Activity Data Set Using Recursive Partitioning 1. *Journal of Chemical Information and Computer Sciences* 1999; 39(6): 1017-26.
 23. Barker E J, Buttar D, Cosgrove D A, Gardiner E J, Kitts P, Willett P, Gillet V J: Scaffold hopping using clique detection applied to reduced graphs. *Journal of chemical information and modeling* 2006; 46(2): 503-11.
 24. Konc J, Janežič D: An improved branch and bound algorithm for the maximum clique problem. *MATCH Communications in Mathematical and in Computer Chemistry* 2007; 58: 569-90.
 25. <http://www.tripos.com/data/support/mol2.pdf>
 26. http://en.wikipedia.org/wiki/Floyd%E2%80%93Warshall_algorithm

27. Teague S J, Davis A M, Leeson P D, Oprea T: The design of leadlike combinatorial libraries. *Angewandte Chemie International Edition* 1999; 38(24): 3743-8.
28. Sugimoto H, Yamanishi Y, Limura Y, Kawakami Y: Donepezil hydrochloride (E2020) and other acetylcholinesterase inhibitors. *Current medicinal chemistry* 2000; 7(3): 303-39.
29. Vainio M J, Puranen J S, Johnson M S: ShaEP: molecular overlay based on shape and electrostatic potential. *Journal of chemical information and modeling* 2009; 49(2): 492-502.
30. Irwin J J: Community benchmarks for virtual screening. *Journal of computer-aided molecular design* 2008; 22(3-4): 193-99.