Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Méréstechnika és Információs Rendszerek Tanszék

Adott célpontra történő gyógyszerhatóanyag generálás hatóanyagmolekulák látens teréből

Készítette
Pogány Domonkos

Konzulens
Sárközy Péter

2020

TARTALOMJEGYZÉK

Absztrakt

A modern gyógyszerkutatás egyik fő motivációja az új, gyógyszerként viselkedő vegyületek előállítása, egy új gyógyszer kifejlesztése azonban rendkívül költséges és időigényes folyamat. A folyamat legnagyobb részét a megfelelő molekulák keresése és azok tovább optimalizálása teszi ki, hiszen a valaha szintetizálható gyógyszerhatóanyagok csupán töredékét ismerjük, és a közöttük történő keresés is újabb nehézségeket vet fel a molekulatér diszkrét jellege miatt.

A mesterséges intelligencia területéről vett generatív neurális modellek segítségével áthidalhatjuk ezen problémákat. Segítségükkel képesek lehetünk molekulák folytonos látens terében keresve eddig nem ismert, de megfelelő kémiai tulajdonságokkal rendelkező gyógyszermolekulákat találni.

A dolgozatomban a variációs autoenkóderen alapuló modellemet mutatom be. A modell képes a molekulák szöveges reprezentációjából egy folytonos látens teret generálni visszacsatolt neurális rétegek segítségével. A látens tér eloszlását prior információ megadásával lehet alakítani, én a leggyakrabban alkalmazott Gauss eloszlásnál valamivel jobban teljesítő hiperszferikus eloszlást használtam. A látens teret úgy alakítottam, hogy az önmagában a lehető legtöbb információt hordozza a reprezentálandó molekuláról. Ezek az információk lehetnek kémiai, szerkezeti, vagy akár hatóanyag-célpont interakcióval kapcsolatos értékek is. Az így kialakított látens térben egy genetikus algoritmussal valósítottam meg a keresését és generálását.

A Guacamol benchmark által előírt mérőszámok mentén hasonlítottam össze a modellem generálóképességét a már létező state-of-the-art modellekkel. Többféle architektúra kipróbálása és hiperparaméterek vizsgálata után a modellem jobban teljesített a Benevolent AI által közölt modellek többségénél. Ezen kívül a modellem képes egy tetszőleges tulajdonságokból előre összeállított célfüggvény szempontjából megfelelő, új gyógyszerhatóanyag jelölteket generálni. Az optimalizálandó célfüggvény alakításával, és egy hatóanyag-célpont becslő tanítása után kinyert interakciós adatok segítségével sikerült megoldani az egy adott célpontra való hatóanyaggenerálás problémáját is. A generált gyógyszerjelöltek jóságát egy molekula-fehérje dokkoló program segítségével ellenőriztem.

Az elért eredményeket tekintve elmondható, hogy a mélytanulást alkalmazó módszereknek van létjogosultsága az orvosi gyógyszerkutatás terén, az egy célpontra történő generálás pedig egy jelenleg is aktívan kutatott terület. A módszerek tökéletesítésének és ötvözésének köszönhetően nagyobb áttörések várhatóak a közeljövőben, amik a gyógyszerfejlesztési folyamat lerövidítése által elősegíthetik az új kórokozók elleni küzdelmet is.

Abstract

Drug-like molecule generation for given targets from the latent space of bioactive molecules

One of the main motivations for modern drug research is the production of new compounds that act as drugs, however developing a new drug is an excessively time and resource intensive process. Finding the right molecules and further optimizing them are significant parts of the process, as only a fraction of the drug substances that can ever be synthesized are known, and searching in the space of known drugs has proven to be difficult due to the discrete nature of the compound space.

Artificial Intelligence might provide a solution, whereas we can use deep generative neural networks to mitigate some of these problems. With their help, we may be able to search in a continuous latent space to find drug molecules that are not yet known but have suitable chemical properties.

I will present my variational autoencoder based generative model. The model is able to generate a continuous latent space from the textual representation of molecules using recurrent neural layers. The distribution of the latent space can be shaped by providing prior information, I used the von Mises-Fisher distribution that performed slightly better than the most frequently used Gaussian distribution. I shaped the latent space to contain as much information as possible about the molecule to be represented. This information may contain values related to chemical or structural properties and even drug-target interaction. I also implemented a genetic algorithm-based method which can search in the latent space and generate molecules.

I compared the generating ability of the model with the already existing state-of-the-art models along the metrics provided by the Guacamol benchmark. After testing multiple architectures and optimizing hyperparameters, the model performed better than most of the models reported by Benevolent AI. Besides that, my model can generate drug candidates, which are not included in the teaching set, and suitable for a pre-specified objective function of arbitrary properties. I also solved the problem of drug generation for a given target by shaping the objective function to be optimized and by using the interaction data obtained after training a drug-target interaction predictor. To validate the goodness of the generated drug candidates, I used a protein-ligand docking software.

Based on these results, it can be said that deep learning methods are useful in the field of medical drug research, and generation to a single target is an area that is still being actively researched. Thanks to the refinement and combination of these methods, major breakthroughs are expected shortly, which could also provide help in the fight against new pathogens by shortening the drug development process.

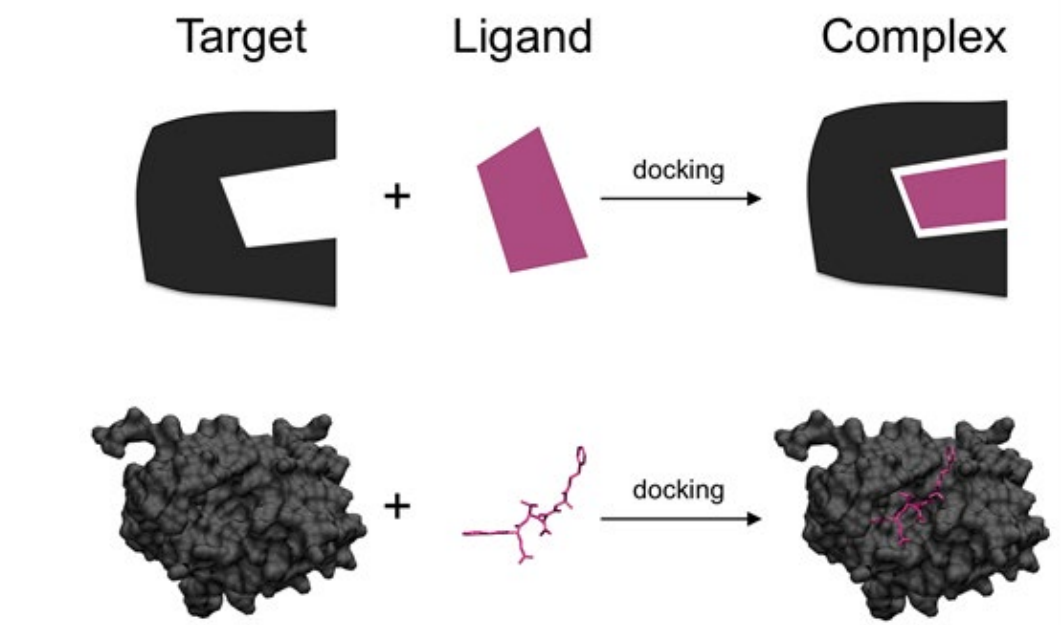
1. Motiváció

1.1. Biokémiai bevezető

A modern gyógyszerkutatás egyik fő motivációja az új, gyógyszerként viselkedő vegyületek előállítása, de egy új gyógyszermolekula kifejlesztése hosszú és költséges folyamat [1]. A teljes folyamat akár 20 évet is igénybe vehet [2], és költsége meghaladhatja a 2.5 milliárd dollárt [3]. A fejlesztés során több ezer lehetséges vegyületet vizsgálnak meg, melyek közül csupán fél tucat jelölt jut el a klinikai vizsgálatokig, ahol azok hatását embereken is tesztelik. A legtöbb esetben csupán egy gyógyszert hagynak jóvá, így az eljuthat az orvosokhoz és páciensekhez.

A fejlesztés minden esetben egy célpont keresésével kezdődik, amin a gyógyszer kifejtheti a hatását. Ezek a célpontok a sejtek jelátviteli útvonalain található fehérjék, melyek a testünkben lezajló biokémiai folyamatok jelentős részéért felelnek. Célpontok lehetnek a sejtekben vagy azok membránján elhelyezkedő receptorok, hozzájuk kötnek a hormonok és egyéb információhordozó molekulák. A másik gyakori célpontcsoport a testünkben végbemenő kémiai folyamatokat szabályozó enzimek. A fehérjemolekulák sajátos háromdimenziós felépítéssel rendelkeznek, amely csak a pontosan illeszkedő vegyületek csatlakozását engedi meg. Ezt az 1.1. ábrán látható modellt gyakran zár és kulcs modellnek nevezik. A fehérjékhez kötő molekulák felerősíthetik vagy csökkenthetik azok aktivitását. Az előbbi esetben aktivátorokról, az utóbbiban inhibitorokról beszélünk. Így működik például a Celecoxib nevű gyulladáscsökkentő gyógyszermolekula is, ami a fájdalomért és gyulladásért felelős cyclooxygenase-2 (COX-2) enzim inhibitora, vagyis a fehérjéhez kötve mérsékli annak hatását. A lehetséges célpontok meghatározásához a proteineket kódoló mRNS szál módosításával kikapcsolják egyes fehérjék hatását, ha a betegségre jellemző sejtszintű folyamat változik, akkor a kikapcsolt fehérje nagy valószínűséggel képes befolyásolni azt.

Az így kiválasztott fehérje és a későbbiekben vizsgált molekulák közötti kötési hajlam mérhető laboratóriumi körülmények között. A hatóanyagok és célpontok közötti interakciót az angol szakirodalomban drug-target interaction-nak (DTI) nevezik. Egy interakció erősségére többféle mérőszám is megadható [4], ilyen értékek az inhibíciós konstans (Ki), a disszociációs konstans (Kd), a medián effektív koncentráció (EC50), és a maximális inhibíciós koncentráció felét jelző érték (IC50). Ezen értékek az elérhető maximális hatáshoz szükséges koncentrációval arányosak, tehát minél kisebb az érték, annál erősebb az adott interakció.



1.1. ábra: A zár és kulcs modell szemléltetése, egy adott fehérje (target) egyes részeihez csak az ahhoz illeszkedő molekulák (ligand) tudnak csatlakozni. [5]

A célpont kiválasztását legtöbb esetben a high-throughput screening (HTS) követi. Ennek során a már ismert, rendelkezésre álló szintetizált vegyületek között keresnek lehetséges gyógyszerjelölteket különböző tesztek elvégzése útján. Ez egy rendkívül költséges és időigényes folyamat, ezért sokszor számítógépekkel végzik el a keresést még nem szintetizált molekulák között.

Ezután a jelölteket tovább optimalizálják, míg azok a megfelelő kémiai és fizikai tulajdonságokkal nem rendelkeznek, ez a folyamat évekig is eltarthat. Az optimalizált vegyületeket itt még kísérleti körülmények között tesztelik.

Az emberi tesztelés előtt még a vegyületet a megfelelő formába kell hozni, biztosítani kell azt, hogy az az emberi szervezetbe jutva a megfelelő helyre és megfelelő mennyiségben kerül szállításra. Ezt követheti az először kisebb, később nagyobb csoportokon történő tesztelés. Végül hatósági jóváhagyás útján kerülhet egy gyógyszer forgalomba.

Lehetséges továbbfejlesztésként a páciens genomjának ismeretében megmondható a személynek egy adott gyógyszerre elvárt reakciója. Ennek segítségével személyre lehet szabni az adott gyógyszert.

1.2. Létező megoldások

A folyamat nagy részét kitevő screening és optimalizálás lépésekre nyújthatnak alternatív megoldást a mesterséges intelligencia területéről átvett módszerek. Mindkét lépést jelentősen megnehezíti a molekulák terének jellege. Screening esetén a már ismert és szintetizált molekulák között lehet keresni, de az erőfeszítések ellenére eddig körülbelül 10^8 molekulát tudtak szintetizálni [6] a valaha szintetizálható 10^{60} gyógyszerszerű molekula közül [7]. A keresést és az optimalizálást tovább nehezíti a tér diszkrét jellege,

ugyanis az érvényes molekulák közötti érvényes átmenetet is definiálni kell. Ezeket a módszereket válthatják fel a generatív neurális modellek, melyek képesek a molekulák folytonos reprezentációjával dolgozva eddig nem ismert, de megfelelő kémiai tulajdonságokkal rendelkező molekulákat találni.

A tetszőleges kémiai tulajdonságokkal rendelkező jelöltek előállítása az utóbbi évek egyik aktív kutatási területe, számos megoldás született új molekulák generálására [8][9]. A módszerek nagy része egy, a molekulák szöveges reprezentációját használó generatív modellt mutat be. A szöveges reprezentáció maga után vonja valamilyen visszacsatolt hálózat alkalmazását [10]. A legtöbb módszer egy autoenkóderen (AE) [11] alapul, ez lehet variációs autoenkóder (VAE) [12] [13], illetve versengő autoenkóder (AAE) [14] [15]. Ezek segítségével a szöveges reprezentáció diszkrét terét egy olyan folytonos látens térbe lehet konvertálni, amiben a keresés és optimalizálás feladatát könnyebben meg lehet valósítani. Léteznek szöveges reprezentáció helyett gráfokkal dolgozó autoenkóderek is [16], az ilyen modell által generált molekulák minden esetben érvényesek lesznek, hiszen a molekulagráfon dolgozva nem véthetünk a szöveges reprezentációkra jellemző szintaktikai hibákat. A molekulagráfokat egymás utáni lépésekkel is fel lehet építeni, ezt használja ki egy megerősítéses tanuláson alapuló módszer [17], ami kiküszöböli a tanítóhalmaz specifikusságából eredő torzításokat. Egyes módszerek a teljesen új molekula generálás helyett egy már meglévő hatóanyagból indulnak ki. Egy generatív versengő hálózatot (GAN) [18] használó módszer [19] a bemeneti gyógyszerhatóanyag valamilyen szempont szerinti tovább optimalizálását valósítja meg.

A kutatási területet nagyban elősegítette a jelenleg is használt viszonyítási értékek publikálása. A Guacamol benchmark [20] által előírt értékek alapján képesek vagyunk az egyes modellek generatív képességét összehasonlítani. A szerzők az értékeken kívül bemutattak előre kiértékelt modelleket, illetve megadtak interfészeket, amiken keresztül bárki kiértékelheti a saját megoldását. A modellek között megtalálható egy szöveges reprezentációval dolgozó VAE és egy AAE, egy gráfokkal dolgozó GAN, egy véletlenszerű mintavételező, egy egyszerű visszacsatolt hálózat, és egy gráfok terében kereső modell.

Az eddig említett módszerek képesek egy adott fizikai és kémiai tulajdonságból összeállított célfüggvény szerint megfelelő molekulákat generálni, és meglévő molekulákat a célfüggvény szerint optimalizálni. Azonban önmagukban nem nyújtanak megoldást az egy adott fehérjéhez, a kulcs-zár modell alapján illeszkedő hatóanyag generálására.

Ahhoz, hogy képesek legyenek egy adott célpontra generálni gyógyszerjelölteket, mint ahogy az a hagyományos gyógyszergyártási folyamatban is történik, rendelkezniük kell az adott célpont és molekula közötti interakciós adatról. Az interakciós értékek mérésének költsége és időigénye miatt nem célszerű azt minden generált molekulára elvégezni, így szükség van valamilyen módon becsülni az értékeket. Erre is használnak gépi tanulást alkalmazó megoldásokat. A probléma egyik megközelítése a már ismert gyógyszerekhez történő új lehetséges interakciók keresése. Ebben a megközelítésben jól alkalmazhatóak a restricted Boltzmann machine (RBM) [21] alapú módszerek [22]. A másik,

generáláskor jobban alkalmazható megközelítésben ismeretlen molekulákhoz is képesnek kell lennie a modellnek interakciót becsülnie. Erre a legelterjedtebb megoldások a molekulák és fehérjék karakterekből álló szekvencia reprezentációjából kiindulva egy konvolúciós neurális hálóval [23], vagy mátrixfaktorizáció segítségével [24] adnak interakciós becsléseket. Ennél könnyebben általánosítható megoldások képesek a fehérje szerkezetének ismerete nélkül, csupán a molekula gráfjáról alkotott kép [25], vagy egyéb, a molekula struktúráját hordozó reprezentáció [26] ismeretében interakciós adatot becsülni.

Az interakciós értékek becslésén kívül léteznek molekula dokkoló szoftverek is, melyek a célpont fehérje szerkezetét figyelembe véve képesek egy megadott molekulát kiértékelni a csatlakoztathatóság szempontjából. Egy széles körben elterjedt molekuláris dokkoló program a VINA [27]. Eredményül megadja a több ezer kiértékelt pozíció közül a legjobb elhelyezkedéseket, és a hozzájuk tartozó energia értékeket is, ahol kisebb energia nagyobb kötési affinitást jelöl.

Amint azt láthattuk az utóbbi időben jelentős eredményeket értek el mind az általános molekulagenerálás, mind az interakcióbecslés területén. Ezen módszerek bizonyítják, hogy lehetséges olyan modelleket konstruálni, amik képesek megragadni a gyógyszermolekulák egyes tulajdonságait és újakat generálni, egyes modellek pedig az interakció létrejöttéért felelős információkat is képesek kinyerni. Ezek az eredmények azonban rendkívül frissek, az egy adott célpontra történő generálás területe még jelenleg is aktívan kutatott, és még áttörésre vár. Amikor elkezdtem molekulagenerálással foglalkozni, még nem volt elterjedt az egy célpontra való generálás, jelentősebb publikációk csupán az elmúlt fél évben jelentek meg. Egy ilyen, a COVID vírus által is motivált megoldás egy fehérjestruktúrával dolgozó VAE modellt mutat be [28].

1.3. Megoldásom felvezetése

Dolgozatom során a három kitűzött és teljesített feladatomat részletezem. Az általános molekulagenerálásra egy szöveges reprezentációval dolgozó variációs autoenkódert implementáltam. A modellem egy genetikus algoritmussal kiegészítve teljesíti a második feladatot, vagyis az egy adott célfüggvényre történő generálást. Képes egy tetszőleges kémiai tulajdonságokból összeállított függvény szempontjából megfelelő molekulákat generálni, illetve meglévő molekulákból kiindulva azoknál valamilyen szempont szerint jobbat találni. Ezen felül bemutatok egy megoldást a legspecifikusabb problémára, az egy adott célpontra történő gyógyszerhatóanyag generálásra is. A modellem mindezt a fehérjestruktúra ismerete nélkül teszi, így a módszer könnyen általánosítható a gyógyszerkutatástól eltérő területeken is.

Először a felhasznált módszerek elméleti hátterét ismertetem, majd vázolom a modellem szerkezetét, kifejtem az egyes egységeknek a végső megoldásban betöltött szerepét. Ezután térek csak ki az egyes részek konkrét implementációjára, a felmerülő problémákra és döntésekre, majd pedig a modellem értékelésére. Ez utóbbiakat nem időrendi sorrendben, hanem a három feladatra szétbontva teszem. Végül egy konkrét példát mutatok be a modellem felhasználhatóságára.

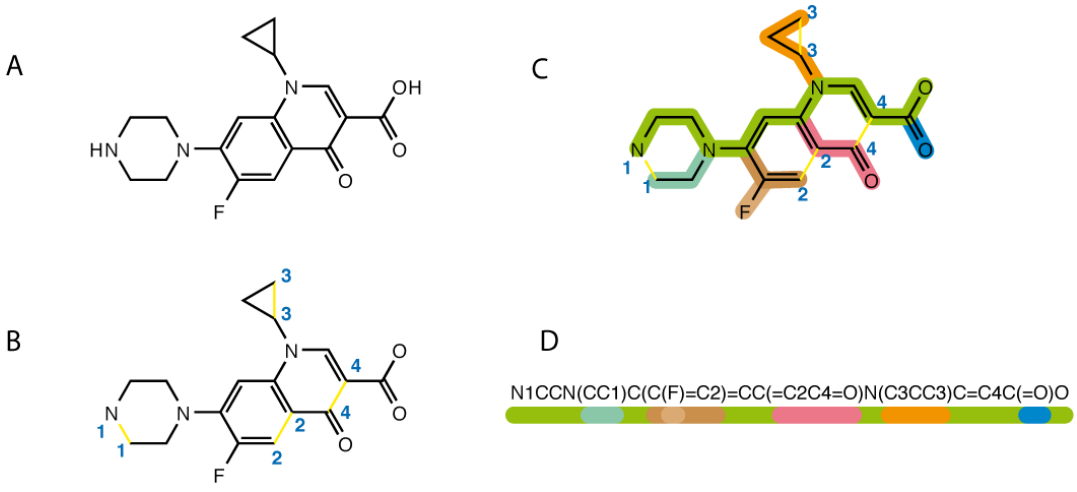
2. Elméleti háttér

Először az implementációtól függetlenül, általánosan szeretném bemutatni a legfontosabb alkalmazott módszereket.

2.1. Reprezentáció

Ahhoz, hogy a később bemutatott generatív modellek képesek legyenek molekulákat előállítani, valamilyen számítógép által is feldolgozható reprezentációt kell választani. Több ilyen molekula reprezentáció is létezik, az információtartalom és a kezelhetőség különbözteti meg őket. Léteznek gráf alapú leképezések, melyek jobban megőrzik a molekula struktúráját. Ezen belül a 3 dimenziós gráfok több információt tartalmaznak a struktúráról, de a 2 dimenziós gráfok könnyebben kezelhetőek. Léteznek ennél is egyszerűbben kezelhető, de kevesebb strukturális információt tartalmazó szöveges reprezentációk is.

A legnépszerűbb, könnyen kezelhető szöveges formátum a SMILES (simplified molecular-input line-entry system) [29]. A reprezentáció előállítását szemlélteti a 2.1. ábra. A reprezentáció készítése során a molekuláról készült 2 dimenziós gráfot képezzük le ASCII karakterek sorozatára. Először a köröket felbontjuk, majd a keletkező fát járjuk be mélységi kereséssel a gerincvonal mentén. A bejárást több ponton kezdhetjük, így egy molekulához több SMILES reprezentáció is tartozhat, de egy reprezentációhoz pontosan egy molekula tartozik, ennek következményeire a későbbiekben térek ki.



2.1. ábra: Egy molekuláról készült SMILES reprezentáció előállítása. A gráfban (A) lévő körök felbontása (B), majd a gerincvonal menti bejárás (C) után előállított szöveges reprezentációt (D) láthatjuk. [29]

A reprezentáció hátránya, hogy nem lehet garantálni az adott molekulához tartozó gráfot szintaktikailag helyesen reprezentáló szavak generálását. Erre egy megoldás a szavak helyett azok környezetfüggetlen nyelvtanát használni, a generáláskor szabályokból épített SMILES szó biztosan érvényes lesz. Én is kipróbáltam a „grammar VAE” [30] implementációt, de ennek tanítása rengeteg időt és memóriát igényelt. Végül a sima

SMILES reprezentációt választottam, mivel ezzel is kellően magas érvényességet értem el, és különösebb átalakítás nélkül felhasználhattam a természetes nyelvfeldolgozás során is használt, már kiforrott visszacsatolt neurális hálózatokat.

2.2. Visszacsatolt hálózatok

A választott szöveges reprezentáció kezelésére kézenfekvő, valamilyen visszacsatolt hálózatot alkalmazni, ugyanis a szavakban lévő karakterek sorrendisége fontos jelentéssel bír. Az ilyen jellegű adatok tanulására pedig ezek a modellek bizonyultak a legmegfelelőbbnek. Az egyes molekulákhoz tartozó SMILES szavak hossza, illetve a kötések és felbontott körök szintaktikája miatt a hosszabb szekvenciákat is kezelni képes kapuzott neurális hálózatokat [31] alkalmaztam. Ezek képesek megvalósítani hosszú-, és rövidtávú memóriát is, így képesek kezelni a felbontott körök miatt messze kerülő szám párokat és zárójeleket.

A „long short-term memory” (LSTM) cella a legelterjedtebb. Két belső állapota van, az egyik állapot a cella állapot, amit aktiváció nélkül terjeszt végig, ezzel oldja meg az időben elenyésző gradiens problémáját, a másik állapot a rejtett állapot, ebből állítjuk majd elő a kimenetet. Az ’input’, ’forget’ és ’output’ kapu segítségével képes megtanulni, hogy az egyes bemenetek, a cellaállapot és az előző kimenetek hogyan befolyásolják a következő kimenetet és a cellaállapotot.

A „gated recurrent unit” (GRU) is képes megoldani a hosszútávú emlékezet problémáját, ugyanakkor csak egy rejtett állapota, és jóval kevesebb tanítható paramétere van. Az esetek többségében sebessége miatt a GRU-t választják, de a kevesebb paraméter miatt a háló hipotézistere is kisebb, azaz elméletben nem képes az LSTM által megvalósítható összes leképezést megtanulni.

A visszacsatolást nem csak egy irányban végezhetjük el, a kétirányban visszacsatolt hálózatok [32] két cellával rendelkeznek. A két cella rejtett állapotait azonban ellentétes irányban csatolják vissza, a kimenetet pedig mindkét cella tartalma alapján számolják. A kimenet előállításakor tehát rendelkezésre áll mind a múltbéli, mind a jövőbeli cellák által hordozott információ.

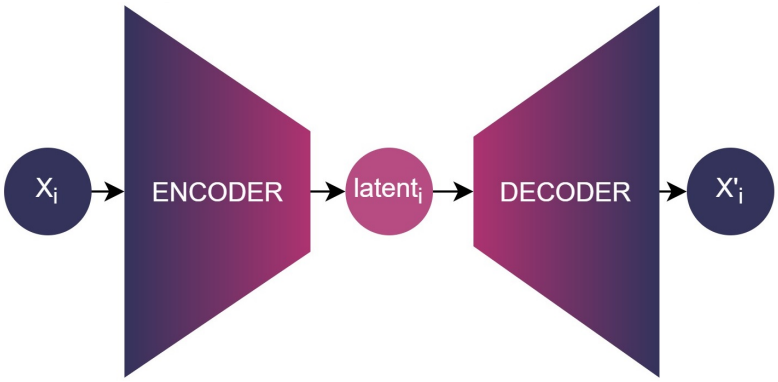
2.3. Generatív modellek

Mivel molekulák generálását tűztem ki célul, ezért valamilyen generatív modell használatát igényli a megoldásom. A generatív modellek egy adott adathalmaznak az eloszlását képesek modellezni, majd miután megtanulták azt, a tanítóadatok eloszlására illeszkedő, de új adatokat tudnak előállítani. Segítségükkel jelentős sikereket értek el képek, szövegek, és hangok generálásában. Ezek a modellek a nem felügyelten tanítható csoportba tartoznak, vagyis tanítás közben nem áll rendelkezésre felcímkézett, elvárt kimenet. A leghíresebb ilyen modell a GAN, ami két egymással versengő hálót, egy generátort és egy diszkriminátort tartalmaz. Én az autoenkóderek családjába tartozó variációs autoenkódert használtam, hiszen ennek folytonos látens tere alkalmas a benne történő keresésre és optimalizálásra.

2.3.1. Autoenkóder

Az autoenkóder [11] egy olyan neurális hálózat, aminek feladata a bemenetén kapott adat reprodukálása a kimeneten. Ezen felül azt a megkötést alkalmazzuk, hogy rendelkeznie kell egy, a bemeneti dimenziónál kisebb dimenziójú réteggel. A legkisebb ilyen réteget szokás szűk keresztmetszetnek nevezni.

A hálózat így két részre bontható, ez látható a 2.2. ábrán. Az első felét enkódernek nevezzük, ez állítja elő a bemenetek látens reprezentációját, aminek a szűk keresztmetszettel egyezik a dimenziója. A másik fele, azaz a dekóder felelős a bemeneti adat reprodukálására a látens reprezentációból.



2.2. ábra: Az autoenkóder felépítése, a bemenetet (X_i) az enkóder alakítja át egy kisebb dimenziójú reprezentációra ($latent_i$), majd ebből a dekóder állítja elő a bemenettel elvárt esetben megegyező kimenetet (X'_i).

Az AE legfőbb tulajdonsága az, hogy képes a fontosabb jellemzőket kiemelve, tömören reprezentálni a bemenetet, ezért szokták tömörítésre alkalmazni. Használható még zajtalanításra is, hiszen a tömör reprezentációban csak a lényeges jellemzőket őrzi meg, a véletlenszerű zajt nem. Használható mély neurális hálók előtanítására, pontosabban az enkóder súlyaival inicializálhatjuk a mély hálót. Mivel a bemenettől a szűk keresztmetszetig tartó rész képes a főbb tulajdonságokat kinyerni, így később az erre épülő rétegek ebből képesek egy tetszőleges klasszifikációs vagy regressziós feladatot megoldani.

A szűk keresztmetszet mérete miatt a látens tér kisebb a bemeneti adatok terénél és folytonos, vagyis jobban kezelhető. Ha a szűk keresztmetszet mérete nem lenne kisebb a bemeneti dimenziónál, akkor a modell megtanulhatná az egységmátrixszal való szorzást, mint leképezést. Viszont a kisebb látens dimenzió miatt kénytelen megtanulni a fontosabb jellemzőket, és a bemeneti adatokat egy sűrűbb, folytonos térben elhelyezni.

A szűk keresztmetszet teszi lehetővé a bemenet eloszlására illeszkedő adatok generálását is. Mivel a bemenet dimenziója nagy, és a diszkrét tér miatt messze helyezkednek el a molekulák, ezért nehéz ebben a térben egy pontot kiválasztani, ami hasonlít a többire, de mégis eltér tőlük. Például, ha új SMILES szavakat akarunk generálni, megtehetnénk, hogy összeválogatunk pár véletlenszerű karaktert, és ezeket permutálva más és más szavakat állítunk elő, de ezek nagy valószínűséggel nem is lesznek molekulává alakítható

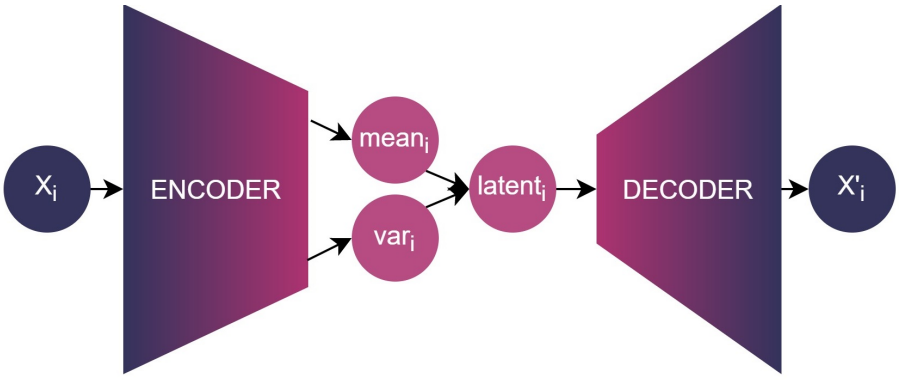
reprezentációk. Ugyanakkor, ha a kisebb dimenziójú folytonos látens térből próbálunk véletlenszerűen mintavételezni, és a kapott vektorból a már előre tanított dekóderrel egy SMILES reprezentációt előállítani, akkor nagyobb eséllyel kapunk érvényes SMILES szót.

Ilyen módon lehetséges a generálás, de az AE egyetlen célja a bemenet helyreállítása. Ha a legkisebb dimenziójú réteg elég kicsi, akkor ebben a rétegben kénytelen egy folytonos reprezentációt tanulni, viszont a lehető legjobban próbálja a látens térben szétszórni az adatokat, így később azokat könnyebben vissza tudja állítani. Generáláskor és optimalizáláskor a célunk az is, hogy a látens térből mintavételezve a lehető legnagyobb valószínűséggel kapjunk érvényes adatot. Az AE viszont ezt nem veszi figyelembe tanulás közben, mert a hibafüggvény csak a bemenet és kimenet közti különbséget nézi, nem tesz megkötést a látens térre.

2.3.2. Variációs autoenkóder

A variációs autoenkóder [12] egy speciális AE, aminek a látens terére is tehetünk megkötéseket. A hagyományos AE modellek látens terében lehetnek szakadások, így még mindig nagy az esélye annak, hogy a látens térből mintavételezett vektorhoz értelmetlen kimenetet fog generálni. Erre megoldást adhat, ha a látens tér eloszlását kikényszerítjük. Ha ez sikerül, akkor a látens térbeli interpoláció is javulni fog. Azaz, ha veszünk két adott bemeneti molekulát, és a hozzájuk tartozó látens reprezentációk között mintavételezünk, akkor az így kapott kimenet egy új molekula lesz, ami az előző kettőre hasonló.

Ahhoz, hogy ezt elérjük szükségünk lesz további két rétegre ahogyan az a 2.3. ábra is mutatja. Az egyikben az egyes látens dimenziók várható értékeket, a másikban a dimenziókhoz tartozó szórásokat számoljuk. A látens reprezentációt az előbbi két vektor segítségével mintavételezzük. A kívánt eloszlástól való eltérést így már a veszteségfüggvényben büntethetjük.



2.3. ábra: A variációs autoenkóder felépítése, az enkóder csupán a látens reprezentáció szórását (var_i) és várható értékét ($mean_i$) állítja elő, a reprezentációt ezután ezekből mintavételezzük.

Például, ha azt szeretnénk, hogy a reprezentációk egyenletesen helyezkedjenek el az origó körül, akkor a hibához hozzá kell adnunk egy, a reprezentáció és a standard normális közötti eltérést mérő tagot. Az eltérés nagyságát a Kullback-Liebler (KL) divergencia

segítségével mérhetjük. Tehát Gauss prior látens eloszlás esetén az alábbi tagot kell a hibához hozzáadni:

$$\sum_{i=1}^{n} \sigma_i^2 + \mu_i^2 - \log(\sigma_i) - 1$$

Még egy módosítás szükséges, hiszen a mintavételezés miatt a modell nem differenciálható. Erre ad megoldást az úgynevezett „reparametrization trick". Adottak a bemenetből tanult várható értékek és szórások, a mintavételezett látens vektor a következőképpen áll elő:

$$latent(x_i) = mean(x_i) + var(x_i) * N$$

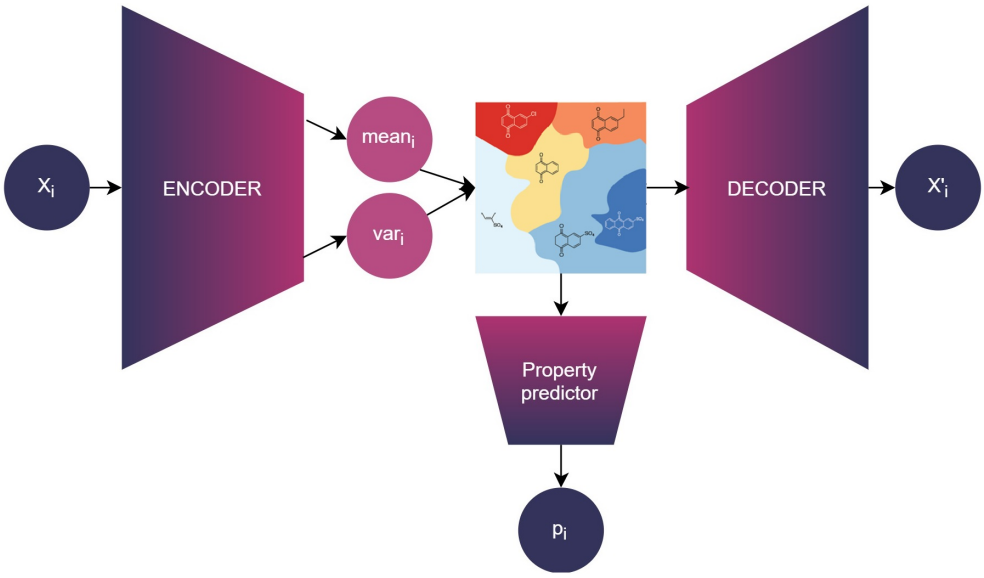
Ahol N egy normál eloszlású valószínűségi változó 0 várható értékkel és 1 szórással. Így már vissza tudjuk terjeszteni a gradienst, mert a látens réteg már közvetlenül függ a szórástól és várható értéktől, az egyetlen véletlen változó az N, de ennek nincsenek tanulható súlyai.

Az egyszerű AE modell képes rekonstruálni a bemenetéül kapott adatot, sőt új adatok generálására is alkalmas. A VAE látens terének regularizálásával csökkent a modell rekonstrukciós képessége, ugyanakkor egy általunk előírt folytonos látens térrel rendelkezik, amiből könnyeb új adatot generálni. A háló tehát képes kinyerni a bemeneti adatok főbb tulajdonságait, és egy előre meghatározott eloszlású térben ábrázolni azokat, viszont nehéz ezeket az ábrázolt tulajdonságokat visszafejteni.

2.3.3. VAE tulajdonságbecslővel

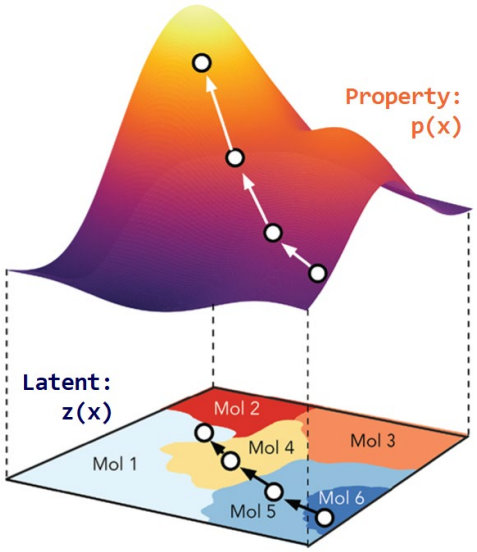
Ha adott tulajdonságokkal rendelkező kimenetet szeretnénk generálni, akkor szükségünk van egy olyan modellre, ami a látens terét ezen tulajdonságok mentén tagolja. Vagyis nem csak azt kell megadnunk a modell számára, hogy milyen adatot transzformáljon milyen térbe, hanem azt is, hogy ezt mi alapján tegye.

Ezt megtehetjük egy tulajdonságbecslő beépítésével [33]. A becslő egy egyszerű előrecsatolt háló, ami a látens térből tanulja az egyes tulajdonságokat. A becslőt egyszerre tanítjuk az autoenkóderrel, tehát a tulajdonságok becsléséből adódó hiba visszaterjed a szűk keresztmetszeten keresztül az enkóderre is, így az ennek megfelelően generálja a látens reprezentációkat. Az így kapott modell felépítését szemlélteti a 2.4. ábra.



2.4. ábra: Tulajdonságbecslővel kiegészített VAE modell, a látens reprezentációt a dekóder mellett egy becslő is megkapja, ennek feladata a bemenet tulajdonságainak (p_i) tanulása.

Ennek hatására az enkóder igyekszik a hasonló tulajdonságokkal rendelkező molekulákat közel elhelyezni a látens térben, hiszen így könnyebben meg tudja becsülni a modell a látens reprezentációból a tulajdonságokat. Ennek következtében a látens térben mozogva az egyes pontokhoz tartozó adatok tulajdonságai egy közel folytonos függvény szerint változnak. Ezt a látens-, és tulajdonságtérben történő mozgást szemlélteti a 2.5. ábra.

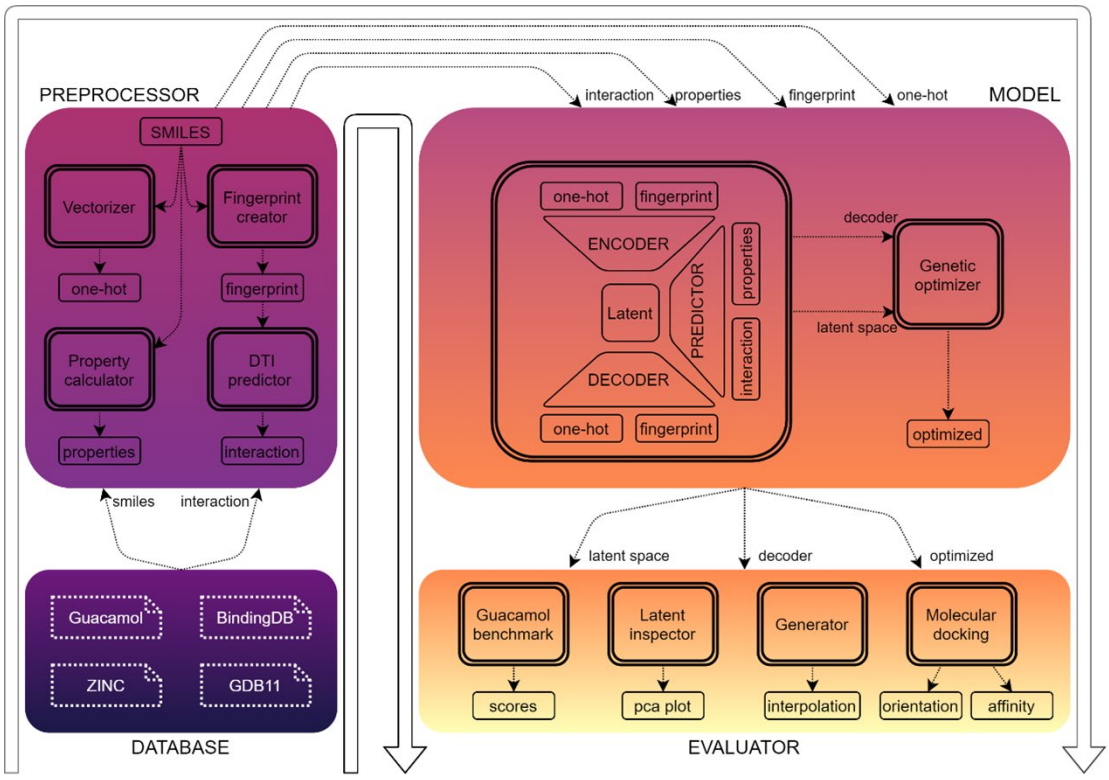


2.5. ábra: A tulajdonságbecslővel ellátott modell látens tere, a hasonló tulajdonságú molekulák közeli reprezentációt kapnak, így a látens térben mozogva az egyes tulajdonságértékek közel folytonosan változnak. [33]

Ennek az optimalizálásnál fontos szerepe lesz, hiszen, ha az optimalizálni kívánt tulajdonságokat tanuljuk a modellel, akkor könnyebben kereshetünk a látens térben az adott tulajdonságot maximalizáló pontokat.

3. Megoldásom áttekintése

Ebben a fejezetben az elkészült rendszeremben szereplő főbb egységek szerepét és működését vázolom, majd a későbbiekben részletesen kitérek az implementációjukra, és a velük elért eredményekre. A funkcionális egységek közötti kapcsolatokat és adatfolyamot mutatja be a 3.1. ábra.



3.1. ábra: Megoldásom architektúrája, jól látszik a főbb komponensek közötti adatáramlás. A különböző adathalmazokból vett adatok először egy előfeldolgozáson esnek át, majd ezután kapja őket meg a modell. A modell tanítása után többféle módon értékelem a kapott eredményeket.

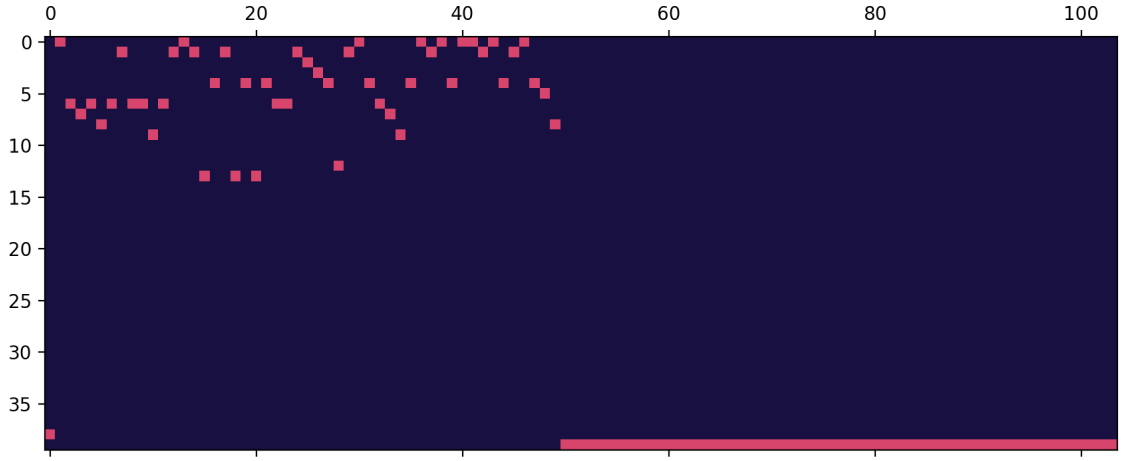
3.1. Felhasznált adathalmazok

Négy bemeneti adathalmazzal dolgoztam. A legkisebb a „gdb11_size08” [34], ami maximum 8 atomból álló kisméretű szerves molekulákat tartalmazó 66.706 elemszámú halmaz. Ezt leginkább az új funkciók gyors tesztelésére használtam, a kiértékeléseket egy nagyobb adathalmazon végeztem, hiszen a modellnek a gyógyszermolekulákat is képesnek kell lennie kezelnie, amik jóval több mint 8 atomból állnak. Az egyik ilyen nagyobb adathalmaz a „Zinc” [35] halmazból kiválogatott 249.456 darab gyógyszerszerű molekula, illetve a „Guacamol” [20] adathalmaz, ami 1.591.378 molekulát tartalmaz. A dolgozatomban közölt eredményeket a más modellekkel történő összehasonlítás érdekében az erre a célra kifejlesztett guacamol adathalmazon mutatom be. Az interakciós adatok kinyerésére pedig a BindingDB [36] adathalmazt használtam.

3.2. Előfeldolgozás

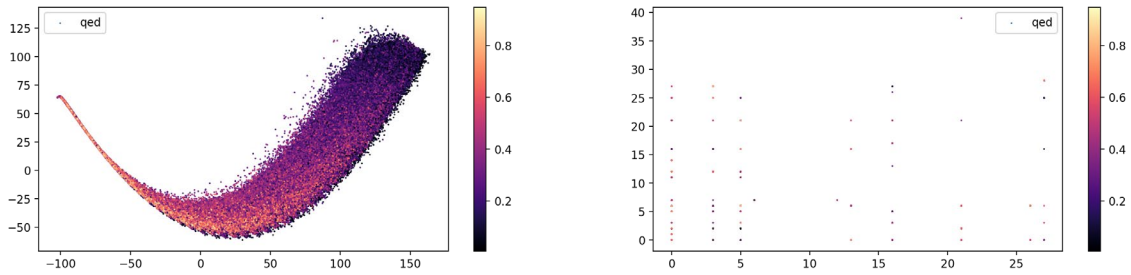
A későbbi tanításhoz elengedhetetlen a SMILES szavak feldolgozható formára alakítása [37]. A modell fix hosszúságú bemenettel fog dolgozni. Ez a hossz a leghosszabb bemeneti reprezentációnál öttel nagyobb, így képes lesz az eredetieknél akár hosszabb SMILES szavakat is generálni. A kódok elejére ’!’ karaktert fűztem, ez lesz az új molekulát jelző karakter, a végén lévő üres helyekre pedig ’E’ karaktereket raktam, ez jelzi a szó végét. Természetesen bármi mást is használhattam volna a ’!’ és ’E’ helyett, ami nem szerepel a bemeneti reprezentációban.

A kiegészített karaktersorozatot ezután one-hot kódoltam. Az egyes karaktereket jelöltem egy one-hot vektorral, és az így kapott vektorokat egy mátrixba rendezve reprezentáltam a szavakat. Egy SMILES szó tehát egy kétdimenziós mátrix, aminek minden oszlopában pontosan egy darab egyes szerepel. A guacamol halmazban lévő molekulák mátrixaniak mérete 40x104, vagyis a kiegészítő karaktereken kívül 38 féle karaktert tartalmaz a halmaz, és a beágyazások hossza 104. Egy ilyen mátrixot láthatunk a 3.2. ábrán.



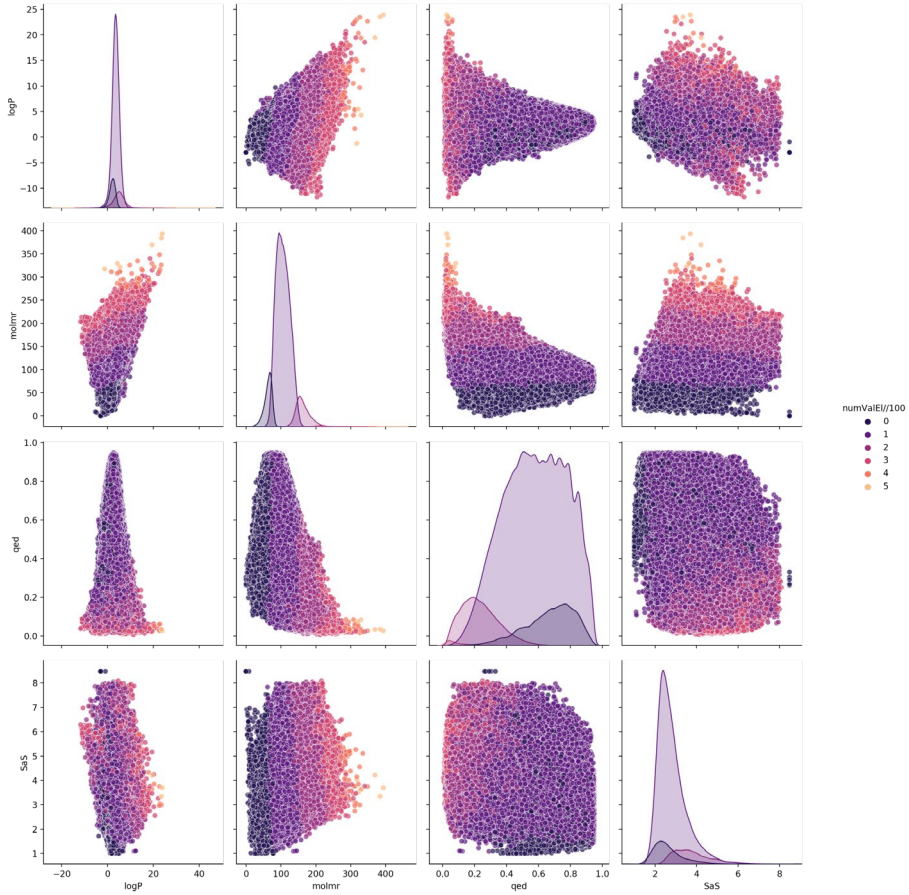
3.2. ábra: A „Cc1c2c(cc3c(C(F)(F)F)cc(=O)n(C)c13)C(C)CC(C)(C)N2” molekula one-hot mátrixa, minden oszlop egy karaktert jelöl. Látszik, hogy a kezdőkarakterből csupán egy szerepel, a végén pedig a maradék helyet a zárókarakterrel töltöttem fel.

A karakterek diszkrét jellege miatt az így kódolt molekulák tere is diszkrét lesz. A diszkrét molekulatér tulajdonságait szemlélteti a 3.3. ábra. Minden reprezentáció annyi dimenziós, amekkora a beágyazás mérete, és minden dimenzióban a lehetséges értékek száma a karakterek számával egyezik meg. A tér egy kétdimenziós metszetén látszik, hogy a molekulák egymástól távol helyezkednek el, egy érvényes molekulából egy másik érvényes molekulába történő transzformáció is egy diszkrét átalakítás. Ezen kívül elmondható, hogy nem egyenletesen tölti ki a rendelkezésre álló helyet sem. Például az első karakter mindig a kezdőkarakter, a 40-ből csupán 10 féle karakter állhat a második helyen, egyes karakterek nem állhatnak bizonyos karakterek után. Illetve a tér nem hordoz információt az egyes tulajdonságokról. Ezért is van szükség egy látens térbe történő átalakításra, ha a térben keresni, interpolálni, generálni akarunk.

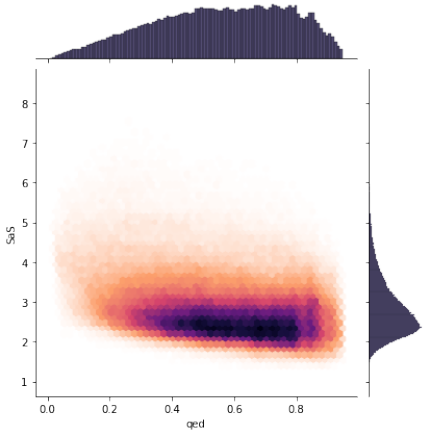


3.3. ábra: A 104 dimenziós molekulatér 2D ábrázolása, illetve metszete. A térben elhelyezkedő molekulák a „quantitative estimate of drug-likeness” (qed), vagyis a gyógyszerszerűséget mérő érték szerint vannak színezve. A képről leolvashatóak a diszkrét tér hátrányai, azaz a reprezentációk távol helyezkednek el, és nem hordoznak sok információt az egyes tulajdonságokról.

Az előfeldolgozás fontos része az egyes tulajdonságok kiszámítása és elmentése. Erre az RDKit nevű python csomagot [38] használtam. Segítségével könnyen lehet kémiai tulajdonságokat is lekérdezni egy adott SMILES-hoz tartozó molekuláról. Ilyen kémiai tulajdonság például a „logP”, ami a felszívódással kapcsolatos, és a térfogatonkénti anyagmennyiséget jelző moláris refrakció (MR). A molekula méretével kapcsolatos mérték még a vegyértékelektronok száma, illetve a „cycle score”, ami a legtöbb atomból álló kör mínusz hattal egyenlő, ha ez a leghosszabb kör nagyobb hatnál. A gyógyszerkutatás szempontjából fontosabb tulajdonságok közé tartozik a „quantitative estimate of drug-likeness” (qed), egy molekula minél inkább gyógyszer szerű, annál magasabb a qed értéke. Szintén fontos a „synthetic accessibility score” (SaS), a könnyebben szintetizálható molekulák SaS értéke kisebb. A guacamol halmazban lévő elemek néhány tulajdonságának egymáshoz való viszonyát láthatjuk a 3.4. ábrán. Feltehetőleg ezek az értékek függetlenek, így célszerű őket külön-külön bevenni a tulajdonságbecslőbe. Például, ha a qed és SaS értékeket akarjuk optimalizálni, akkor a becslőnek mindkét értéket meg kell becsülnie ahhoz, hogy azok elkülönüljenek a látens térben, hiszen az egyik ismeretében nem tudjuk a másikat becsülni, ahogy az a 3.5. ábrán is látszik.



3.4. ábra: Tulajdonságok eloszlása a guacamol halmazon a százzal osztott vegyértékelektronszám egészrészével színezve. Látszik, hogy a súly és a vegyértékelektron, illetve a méret és gyógyszerszerűség kapcsolatán (kisebb molekulák gyógyszerszerűbbek) kívül a többi tulajdonság viszonylag független egymástól.



3.5. ábra: SaS és qed együttes eloszlása a guacamol halmazon, látszik, hogy a kettő között nincs jelentős korreláció, a qed ismeretében nem tudunk a SaS értékére következtetni.

Szintén fontos előfeldolgozási lépés a későbbiekben szükséges interakciós adatok becslése. Ezt egy olyan módszerrel valósítottam meg, aminek csupán a molekula szerkezetét és pár, már ismert hatóanyagot kell ismernie. A módszert és a benne végrehajtott módosításokat később részletezem.

A becslőnek és később a modellnek is szüksége lesz a molekulák szerkezeti információját tartalmazó „molecular fingerprint" nevű 2048 hosszú bináris vektorra, ezért ezt is előre kiszámoltam az RDKit segítségével, erre a későbbiekben ujjlenyomat néven utalok.

3.3. Tanítás

Az előfeldolgozás után kapott adatokkal már tudom a modellemet tanítani. A generatív modellem egy tulajdonságbecslővel kiegészített VAE, aminek a bemenete az előbb látott one-hot mátrix. A szöveges reprezentáció miatt az enkódere és a dekódere is tartalmaz egy-egy visszacsatolt réteget. A látens tér szórását és eloszlását az enkóderben lévő visszacsatolt réteg rejtett-, és cellaállapotának konkatenáltjából tanulja, az ebből mintavételezett látens térből pedig a dekóderben lévő réteg állapotait állítja vissza. A dekóderben található rétegnek karakterenként haladva a következő karaktert kell megbecsülnie. Az elvárt kimenet megegyezik a bemeneten kapott SMILES szóval, de a kezdő karaktert nem tartalmazza, és a végén egy befejező karakterrel van kiegészítve. A VAE a bináris ujjlenyomat által hordozott információt is megkapja, amit egy dropout réteggel regularizált előrecsatolt hálóval terjeszt előre, illetve a bemeneti adat rekonstrukcióján kívül egy teljesen összekötött rétegekből álló tulajdonságbecslő is be van építve a modellbe. A súlyfüggvény ezek alapján a one-hot mátrixok rekonstrukciójáért felelős kategorikus keresztentrópián kívül tartalmazza a látens teret regularizáló Kullback-Liebler tagot (KL), illetve a tulajdonságok becsléséből eredő átlagos négyzetes eltérést (MSE).

Az adott célfüggvényre történő generálást egy genetikus algoritmussal valósítottam meg. Az algoritmus a modell látens terében keresve a dekóder segítségével generál molekulákat.

A részletes implementációt, a hiperparaméterek optimalizálását és az elért eredményeket a későbbi fejezetekben mutatom be.

3.4. Kiértékelés

A tanított modellt kvalitatív és kvantitatív módon is értékeltem.

Az enkóder teljesítményét az általa generált látens tér vizsgálatával ellenőriztem. A tesztadatok között szereplő molekulák látens reprezentációit ábrázoltam a látens térről készült PCA analízis [39] segítségével. Az így előállított két-, és háromdimenziós ábrákon a molekulákat az egyes tulajdonságuk szerint színeztem, így a tér eloszlásán kívül megfigyelhetőek benne az adott tulajdonságok elhelyezkedései is. A tulajdonságbecslőt a látens térre kifejtett hatásán kívül a tesztadaton kiértékelt tanítás pontosságával is értékeltem.

A látens térből történő molekulagenerálásra, és egyben a dekóder tesztelésére különböző interpolációk eredményét ábrázoltam. Interpolálás során 2 vagy több látens pont között

(vagy egy adott pont körül) mintavételeztem egyenletesen, és a mintavételezett látens pontokból próbáltam molekulát generálni. Az így generált molekulák és azok tulajdonságainak megfigyelésével még több információt megtudhatunk a látens térről és a dekóderről.

A generatív képesség kvantitatív értékelésére és összehasonlításra a guacamol benchmark által nyújtott interfészt implementáltam. A benchmark által figyelt öt érték segítségével megmondhatjuk, hogy egy modell mennyire képes egyedi, új, de a tanítóhalmazban szereplő molekulákhoz hasonló molekulákat generálni. Az első mérőszám a „validity”, vagyis az érvényesség, ami az érvényes és az összes generált molekulák számának hányadosával egyenlő. A „uniqueness”, másnéven egyediség, az egyedi érvényes szavak számának és az érvényes szavak számának hányadosa, míg a „novelty”, vagyis újszerűség, a tanítóhalmazban nem szereplő molekulák arányát méri. Ezeken kívül számolja még az egyes tulajdonságoknak az eredeti adathalmaztól vett KL divergenciáját, illetve a Frechet ChemNet távolságot (FCD) [40]. Az FCD a generált molekulák távolságát méri a tanítóhalmaztól. Ezeken kívül a guacamol benchmark toplistás modelljein a „rediscovery”, vagyis az újrafelfedezés értéket is közölték, azaz egy ismert, de a tanítóadatban nem szereplő gyógyszert mennyire képes a modell helyreállítani. A modell rekonstrukciós képességére továbbá egy saját tesztet is implementáltam a saját modelljeim összehasonlítására. A tesztadatok között szereplő véletlenszerűen kiválasztott 10.000 molekulát az enkóder segítségével leképeztem a látens térbe, ezután pedig a dekóder segítségével visszaalakítottam őket, és vizsgáltam, hogy hány százalékukat tudta a VAE hiba nélkül helyreállítani.

A generált molekulákat a célfüggvény szerinti eloszlásuk vizsgálatával értékeltem. Egy adott célpontra generált hatóanyagoknál pedig a molekulákra kiértékelt DTI becslő kimenetét vizsgáltam. A legígéretesebb jelölteket megvizsgáltam egy molekula dokkoló szoftverrel is, illetve összehasonlítottam a generált molekulák kötési energiáját az ismert hatóanyagok kötési energiájával.

4. Implementáció és eredmények

Implementáláshoz a python nyelvet választottam, és a Nyx nevű, 48 magos, 250 GB memóriával rendelkező tanszéki számítógépet használhattam. Az adatok beolvasására és kezelésére a pandas, numpy, illetve az sklearn csomagokat használtam, a molekulákat pedig az RDKit segítségével kezeltem. A modellemet a keras és a tensorflow-gpu segítségével készítettem el. A tanításokat egy NVIDIA TITAN Xp grafikus kártyán végeztem. A későbbiekben közölt benchmark értékek kiszámítására a guacamol csomagot használtam. A dolgozatomban bemutatott ábrákat pedig a matplotlib, plotly és seaborn csomagok segítségével készítettem.

A továbbiakban az egyre nehezedő feladatok mentén fogom bemutatni a fejlesztés főbb lépéseit, a részeredmények értelmezését, illetve a főbb hiperparaméterek és módszerek kiválasztásának folyamatát.

4.1. Általános molekulagenerálás

Először be szeretném mutatni a modellemnek azt a részét, ami az első kitűzött feladatomat, vagyis az általános molekulagenerálást valósítja meg.

4.1.1. Visszacsatolt réteg választása

Mivel az adatokból történő lényegkiemelés nagy részét a visszacsatolt rétegek valósítják meg, így fontos, hogy azok a feladat elvégzéséhez megfelelő komplexitással rendelkezzenek. Az enkóderben lévő réteg tanulja meg a molekulák rekonstruálásához, és az egyes értékek becsléséhez szükséges információt reprezentálni, a dekóderben lévő réteg pedig magáért a rekonstruálásért felel.

A két említett visszacsatolt rétegtípus közül az LSTM-nek nagyobb a hipotézistere, a GRU-val ellentétben képes megvalósítani egy számláló mechanizmust [41]. A tanítások során valóban az LSTM esetén jobb teljesítményt tapasztaltam, így a későbbiekben ezt használtam. Kiemelném ugyanakkor, hogy a GRU tanítása a kevesebb paraméterszám miatt gyorsabb, ezért sokan választják az LSTM helyett. Például a tulajdonságbecslővel kiegészített VAE modellt bemutató cikk [33] írói is GRU-t használtak a dekóderben.

A cikk írói, ezen kívül azt is közölték, hogy az enkóderben használt 1D konvolúciós réteg jobban teljesített a GRU és LSTM rétegeknél, mivel képes kezelni az ismétlődő strukturális elemeket a szövegben. Megvizsgáltam és valóban valamivel jobb eredményhez vezetett az 1D konvolúció, ugyanakkor ennek szerintem más oka is volt. Az LSTM vagy GRU réteg használata azt a feltételezést vonja maga után, hogy a karakterek csak az őket megelőző karakterektől függenek. Ez a feltételezés azonban a SMILES szavak esetén nem igaz, így a temporalitást nem feltételező konvolúciós réteg használata javíthatta a látens reprezentáció minőségét. Kipróbáltam a kétirányban csatolt LSTM réteget, ami a karakterláncban hátrébb szereplő karakterekről hordozott információ segítségével jobban tudja kezelni a SMILES szintaktikát. Ez valóban jobban teljesített, mint a sima LSTM réteg, sőt a konvolúciós módszernél is jobb látens teret generált.

A dekóder karakterről karakterre becsüli a kimenetet, így itt nem alkalmazhattam kétirányban csatolt hálókat. Ehelyett próbáltam egymásra helyezett LSTM rétegeket alkalmazni, de a sima LSTM réteg jobbnak bizonyult. A réteg a következő karakter becsléséhez nem az előző becsült karaktert, hanem az előző elvárt karaktert használja, ezt „teacher forcing" technikának nevezzük. Ezzel lényegében a kimenet visszacsatolását szüntetem meg a tanítás idejére, ezzel stabilizálva és gyorsítva azt.

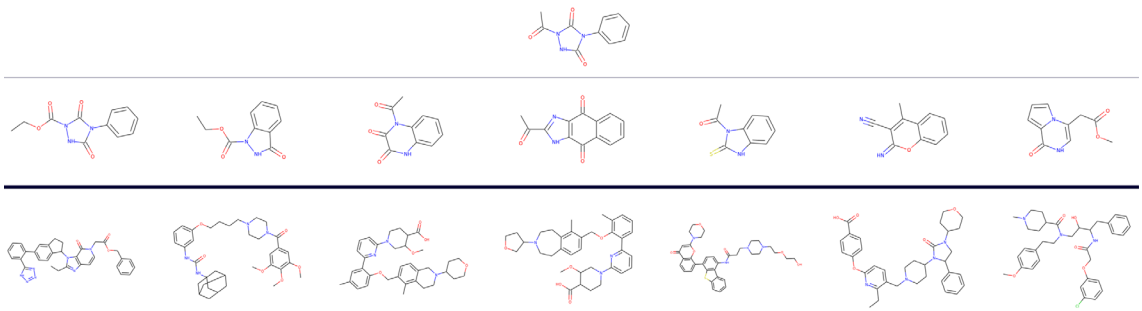
Az enkóderemben tehát egy kétirányban csatolt LSTM található, ami közvetlen a one-hot kódot kapja bemenetéül. Próbáltam a ritka mátrix fölé helyezett folytonos beágyazás használatát is, de ezzel csak romlott a modellem rekonstrukciós képessége. A dekóderembe egy sima LSTM réteget raktam. A réteg tanítása során a teacher forcing technikát alkalmaztam, mert enélkül nem képes a modell megtanulni a molekulák szintaktikáját, és közel 0%-ban érvényes kimenetet produkál.

Fontos hiperparaméter a réteg rejtett-, és cellaállapotainak dimenziószáma is, hiszen ezzel is állítható a modell komplexitása. 256 méretű dimenziókat használtam, ennél nagyobb dimenziószám túltanuláshoz vezetett. Kisebb dimenzió esetén pedig kevésbé volt képes megtanulni a bemeneti adatok jellemzőit, tőlük eltérő molekulákat generált, vagyis alacsony FCD értéket ért el, illetve a bemeneti molekulákat sem tudta ugyanakkora százalékban helyreállítani.

4.1.2. Látens tér vizsgálata

A modell az előbb bemutatott kellően nagy hipotézisterű kétirányban csatolt LSTM réteg segítségével végzi el a lényegkiemelést és a szűk keresztmetszetnek minősülő látens térbe történő leképezést. A következő két alfejezetben az így létrehozott látens tér értelmezésére és továbbfejlesztésére teszek kísérletet.

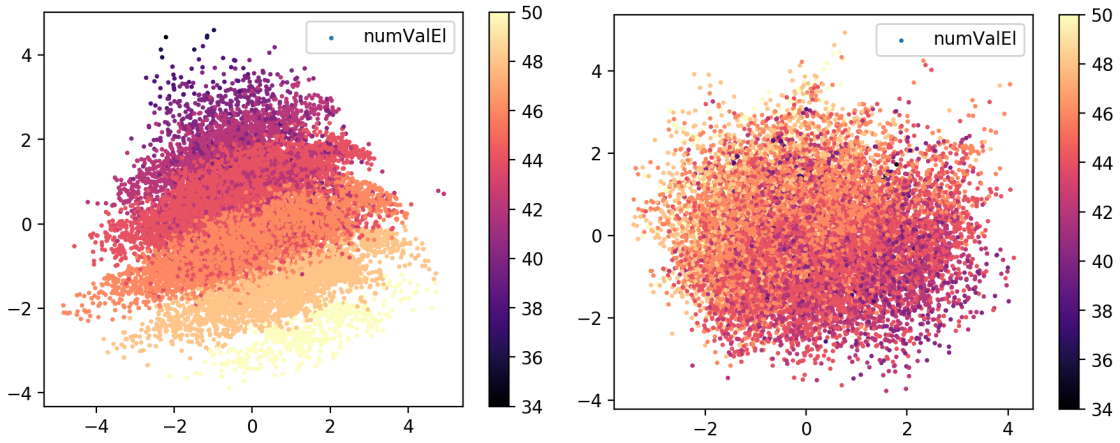
A látens tér egyik legfontosabb tulajdonsága, hogy a hasonló molekulák egymáshoz közeli reprezentációt kapnak. Ez azért van így, mert tanításkor az autoenkóder célja a molekulák rekonstrukciója a látens reprezentációjukból. A későbbi interpolálások, generálások és optimalizálás feltétele, hogy ez a tulajdonság teljesüljön. Ennek tesztelésére vettem egy molekulát és ábrázoltam a hozzá látens térben közel és távol lévő molekulákból párat a 4.1. ábrán.



4.1. ábra: Mintavételezés eredménye a látens térből, látszik, hogy az első sorban szereplő molekulához látens térben közeli molekulák valóban hasonlítanak az eredetire, míg a távolabbról mintavételezett alsó sor eltérő molekulákat tartalmaz.

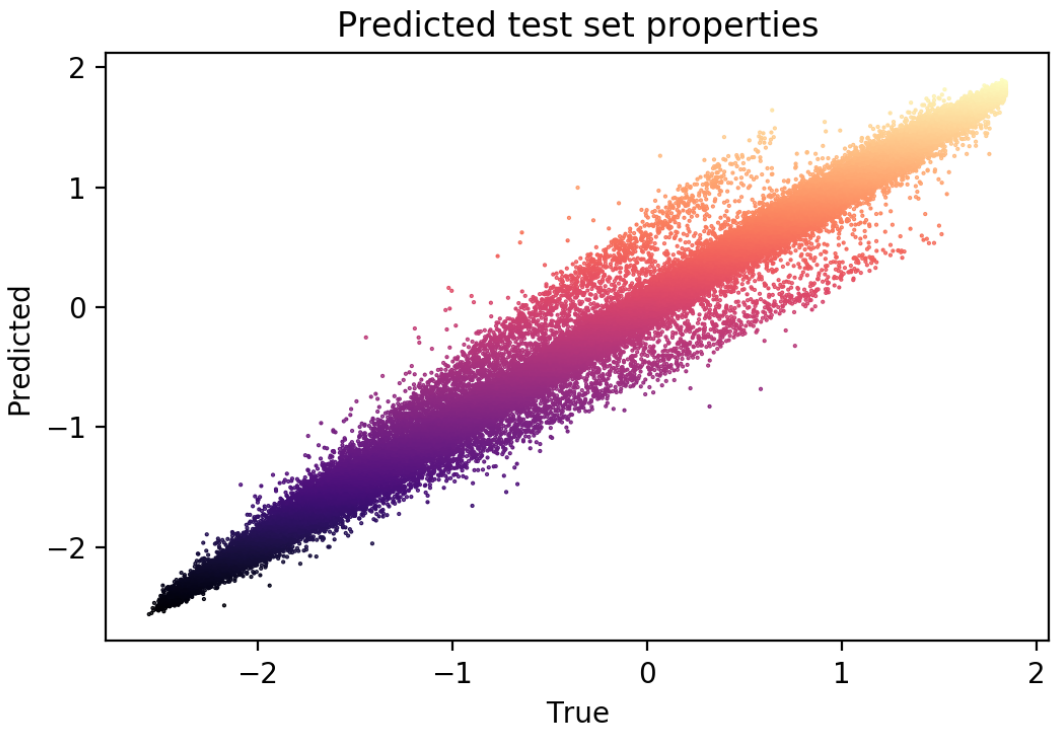
A tér másik fontos, de kevésbé mérhető tulajdonsága a kifejezőereje. A látens tér kifejezőerejét az azt előállító rétegek módosításán kívül a dimenziószámmal is befolyásolhatjuk. Általánosan elmondható, hogy magasabb látens dimenzió magasabb kifejezőerővel bír, de egy bizonyos méret fölött már nem javul tovább a modell. Ugyanakkor azt is megfigyeltem, hogy a kisebb látens dimenziójú modelleknek is vannak előnyei, stabilabban konvergál a tanításuk, és jobban követik a priort. Az előzőek miatt próbáltam azt a legkisebb dimenziószámot megtalálni, amivel még nem romlanak a guacamol benchmark eredményei, illetve ami még elég információt tartalmaz a molekulák helyreállításához. A nagyobb guacamol halmazon tanítva ez a dimenziószám 20 körüli volt, az ennél jóval kisebb látens tér esetén a rekonstrukció nem haladta meg az 5%-ot, a 20 dimenziószám körüli 50%-ot pedig az ennél jóval nagyobbak sem tudták meghaladni. Ez meglepő eredmény, de már más is jutott arra a megfontolásra, hogy nagyjából 10 dimenzió elég lehet a strukturális és kémiai információk kinyerésére [24].

A rekonstrukción kívül a modellnek képesnek kell lennie az egyes tulajdonságok látens térben lévő elkülönítésére is, ezért egy tulajdonságbecslőt is beépítettem az autoenkóderbe. Ahhoz, hogy ennek hatását megfigyeljem, PCA segítségével csökkentettem a látens tér dimenzióját, majd ábrázoltam azt. A 4.2. ábra egy gdb halmazon tulajdonságbecslővel és anélkül tanított, Gauss priort használó modell látens terét mutatja.



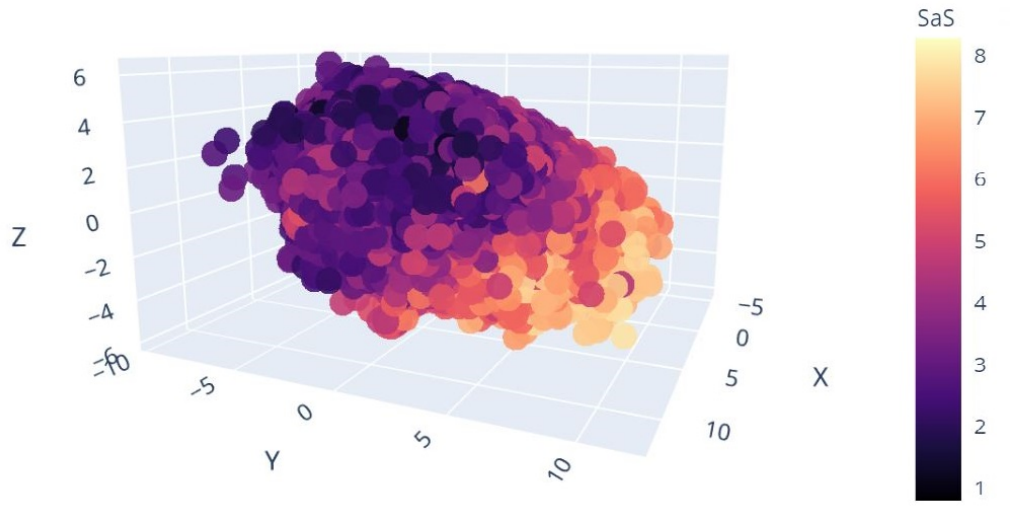
4.2. ábra: Tulajdonságbecslő használatával (bal oldalt) rendezhetjük a látens teret a benne szereplő molekulák értékei szerint, míg becslő nélkül (jobb oldalt) rendezetlenül helyezkednek el az egyes értékek a látens térben.

Látható, hogy élesen elkülönül a tulajdonságbecslős esetben az ábrázolt vegyértékelektronszám. A tulajdonságbecslőt ezután a tesztadatok látens terén kiértékelve látszik, hogy képes csupán a látens reprezentáció ismeretében kis hibával közelíteni a tulajdonságokat. A 4.3. ábra egy guacamol halmazon végzett tanítás utáni tulajdonságbecslő kiértékelését mutatja.



4.3. ábra: A SaS és qed tulajdonságokon tanult tulajdonságbecslő kiértékelése a tesztadatokon, látszik, hogy valóban képes a látens reprezentáció ismeretében becsülni az egyes értékeket, a tanítás végére a tesztadatokon számított MSE 0.0052 lett.

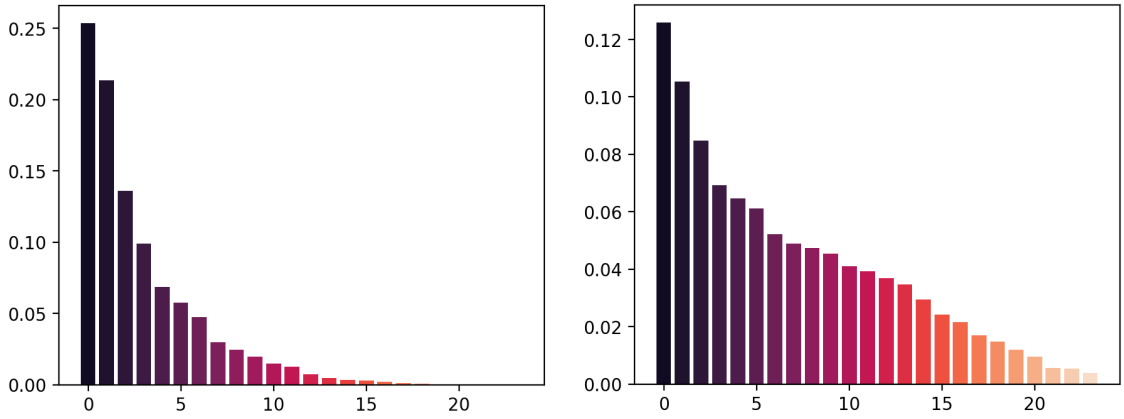
Az első PCA dimenziótól eltérő dimenziók megjelenítésére ábrázoltam a harmadik PCA dimenziót is, ezt láthatjuk a 4.4. ábrán.



4.4. ábra: Az első három PCA komponens ábrázolása, ez már a 2D ábránál valamivel többet mutat a látens térről, de még ez sem ad semmilyen információt a többi látens dimenzióról.

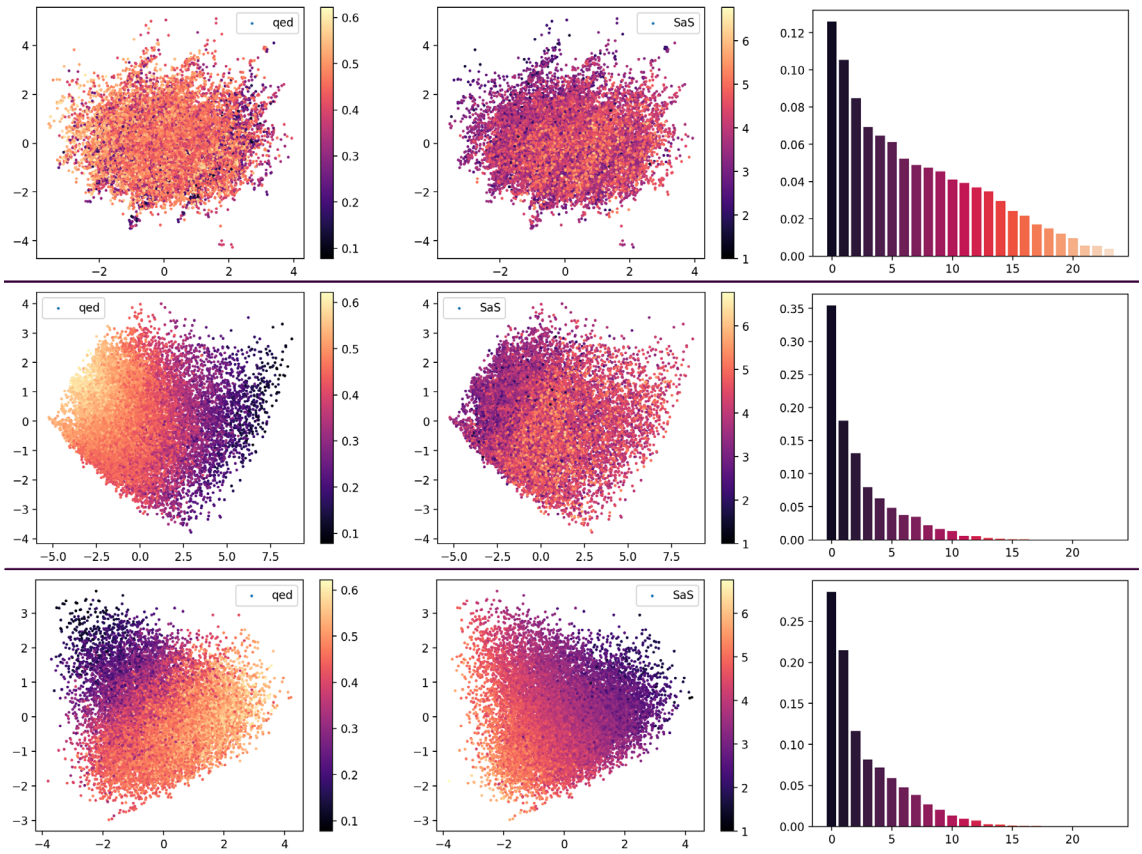
A többi dimenziót a PCA során keletkező új dimenziók varianciájával arányos, sorbarendezett sajátértékek ábrázolásával vizsgáltam. Az egyes dimenziók a látens térben

tekinthetőek azoknak a fontos látens tulajdonságoknak, amiket az enkóder kiemelt az adatból, hogy később azok segítségével helyre tudja állítani a molekulákat és tulajdonságaikat. Minél nagyobb a variancia egy adott dimenzió mentén, az annál több információt hordoz. A 4.5. ábrán láthatjuk két tanítás eredményét, mindkettő modell 24 dimenziós látens teret tanult meg ugyanarra az adathalmazra. Az elsőnek csupán öt dimenziója elég információt tartalmaz az egész variancia 90%-ának lefedéséhez, míg a másik tanulás több, de kevésbé kifejező dimenziót eredményezett. Ez alapján még nem lehet állítani, hogy az első eset a jobb, de többet tudunk meg a modellekről, mint ha csak az első két dimenziót rajzoltuk volna ki. Az első modell tulajdonságok becslésében valószínűleg jobb, míg a másik látens tere jobban hasonlít a standard normálisra, könnyebb belőle mintavételezni.

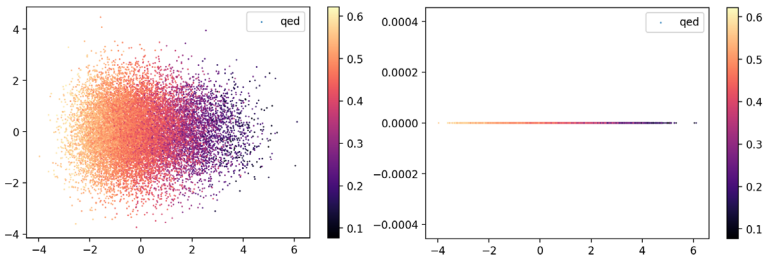


4.5. ábra: Két 24 dimenziós látens tér PCA komponenseinek sorbarendezett sajátértékei láthatóak. A bal oldali tér rendelkezik pár erős, kifejező dimenzióval, de az utolsó 10 dimenzió jelentősen kevesebb információt hordoz. A jobb oldali esetben a variancia egyenletesebben oszlik el az összes dimenzión.

Ezen kívül megfigyelhető, hogy az egyes becsült tulajdonságok közvetlenül megjelennek a PCA dimenziók formájában is. Minden hozzáadott, egymástól teljesen független tulajdonság egy-egy újabb kiugró varianciájú dimenziót eredményez. A 4.6. ábrán három tanítás eredménye látszik a gdb halmazon, az egyes tanításokba 0, 1 és 2 tulajdonság becslését vettem be. Látszik, hogy pusztán a rekonstrukció miatt is a közeli molekulák hasonlóak lesznek, így az egyes tulajdonságok sem teljesen véletlenszerűen helyezkednek el a térben. Ugyanakkor szembetűnő az újabb bevett tulajdonságok becslésének látens térre kifejtett hatása. Megfigyelhető, hogy pusztán a qed érték becslésével a SaS érték is egy bizonyos mértékben rendeződött a térben, hiszen a hasonló molekulák közelebb kerültek egymáshoz, és a két érték eloszlása nem teljesen független a gdb halmazon. A többletinformáció miatt megjelent egy kimagaslóan nagy varianciával rendelkező PCA dimenzió is. Látszik, hogy valóban az első PCA dimenzió (ábrán a vízszintes tengely) hordozza a legtöbb qed információt, ezt mutatja a 4.7. ábra. A SaS érték becslésének bevételével az jobban rendeződött, és megjelent egy másik kiugró PCA dimenzió. Ez a variancia viszont már kisebb, hiszen az előző esetben is már valamelyest rendezett volt a SaS érték.



4.6. ábra: Látens tér és hozzá tartozó varianciák 0, 1 és 2 tulajdonság becslése esetén. Látszik, hogy az első esetben nem használtam tulajdonságbecslőt, így a látens tér kevésbé rendezett, az egyes dimenziók hasonló mennyiségű információt hordoznak. A becslő bevételével viszont megjelenik egy-egy, az egyes tulajdonságokról információt hordozó, nagy varianciájú PCA komponens, ezen irányok mentén pedig rendezett lesz a látens tér.

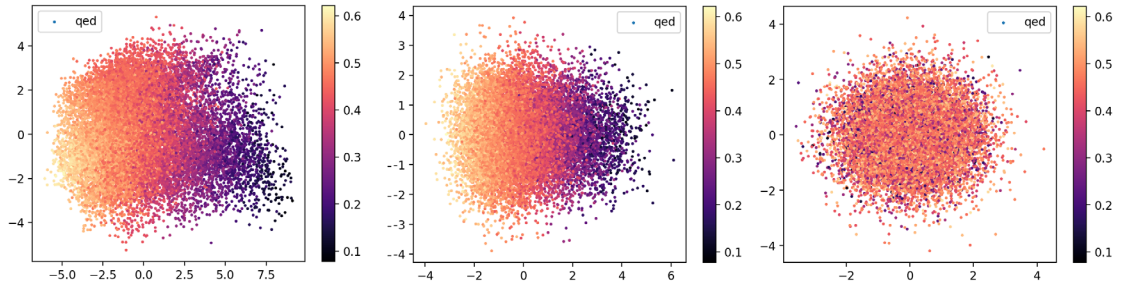


4.7. ábra: 2D PCA leképezés, illetve az első PCA dimenzióra vetített látens vektorok. Látható, hogy a qed becsülhető csupán az első PCA dimenzió ismeretében is.

Mindezt figyelembe véve megállapítható, hogy a tulajdonságok becsléséhez és a molekulák rekonstruálásához is elég információt tartalmazó látens térnek nem kell 30 dimenziósnál jelentősen nagyobbnak lennie. A rekonstrukció figyelésén kívül a PCA varianciákat követve is látszik, hogy a látens tér információtartalmának magas százalékát lefedhetjük már akár 10 dimenzióval is. Illetve a legtöbb esetben nem is használja ki a modell a rendelkezésre álló dimenziószámot, a kevésbé szükséges dimenziók varianciája ilyenkor nulla közeli.

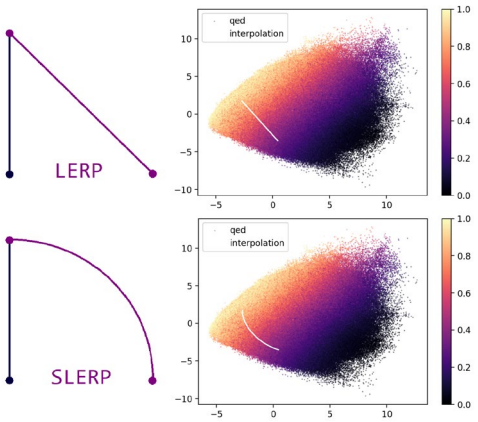
4.1.3. Látens prior eloszlás módosítása

Látszik, hogy a standard normális eloszlás látens priorként használva valóban sűrűbb, Gauss szerű látens teret eredményez. Viszont még ebben a térben is lehetnek szakadások, kiugró értékek, vagyis a látens tér eloszlása nem követi pontosan a priort. Ezen tudtam javítani a KL divergencia tag súlyának növelésével, de így sem sikerült könnyen mintavételezhető, Gauss látens teret formálnom. Sőt, a KL tag túlságos megnövelése gyakran vezetett 0%-os egyediség értékhez, erre láthatunk példát a 4.8. ábrán. Ugyanis, ha a rekonstrukciós hibatagnál erősebb a KL tag, akkor a látens tér valóban illeszkedik a prior eloszlásra, de a generátor figyelmen kívül hagyja azt, így mindig csak egyféle molekulát generál függetlenül a hozzá tartozó látens pont helyétől.



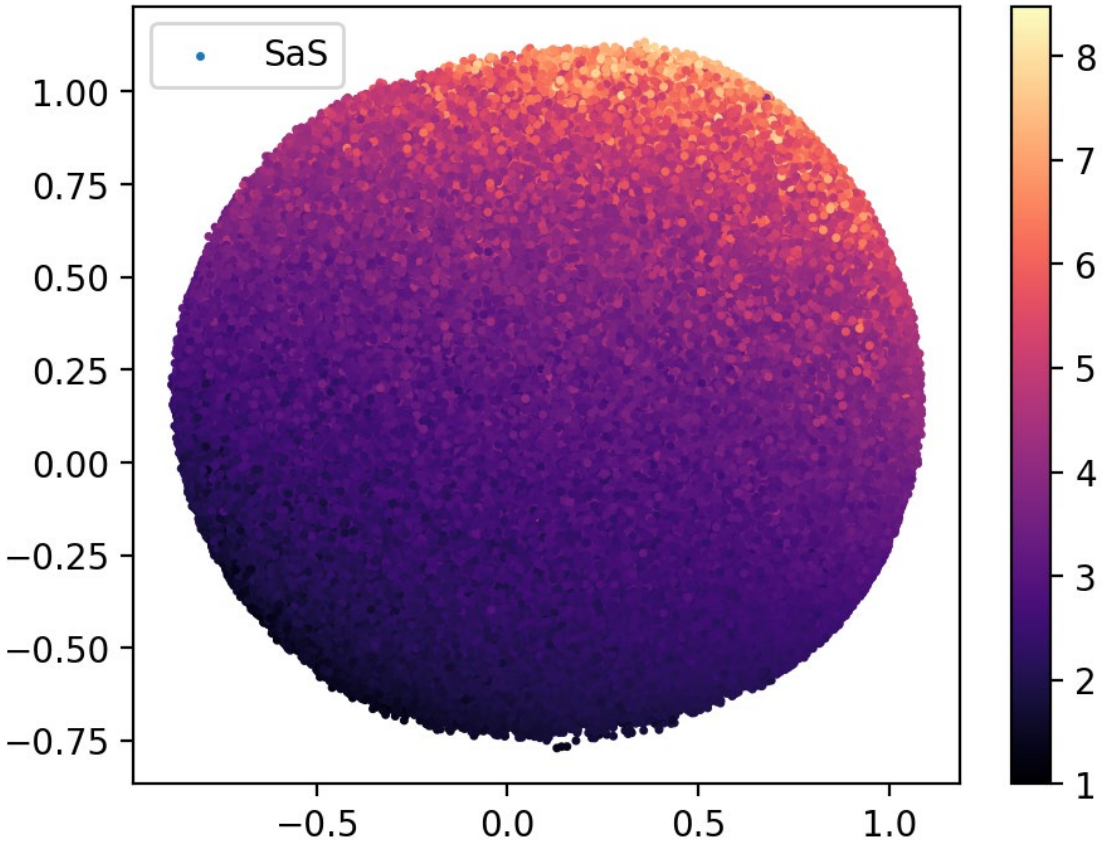
4.8. ábra: Egyre erősödő KL tag hatása. A látens tér egyre jobban követi a prior eloszlást, de egyre kevesebb információt hordoznak a legnagyobb PCA komponensek.

Két pont közötti interpoláció eredményét vizsgálva valamivel többet tudhatunk meg a látens tér belső szerkezetéről. Egy generatív modellek látens terében történő mintavételezésről szóló kutatás [42] szerint a legtöbb modell látens terében a lineáris interpoláció nemvárt eredményeket tud okozni. Ezért a korábban használt lineáris interpoláció (LERP) mellett az úgynevezett „spherical linear interpolation” (SLERP) interpolációt is kipróbáltam. Ezzel az interpolációval nem a két pont közötti egyenesen, hanem a kettő közötti átforgatási pályán tudtam mintavételezni. A 4.9. ábra mutatja a látens térben mintavételezett pontokat.



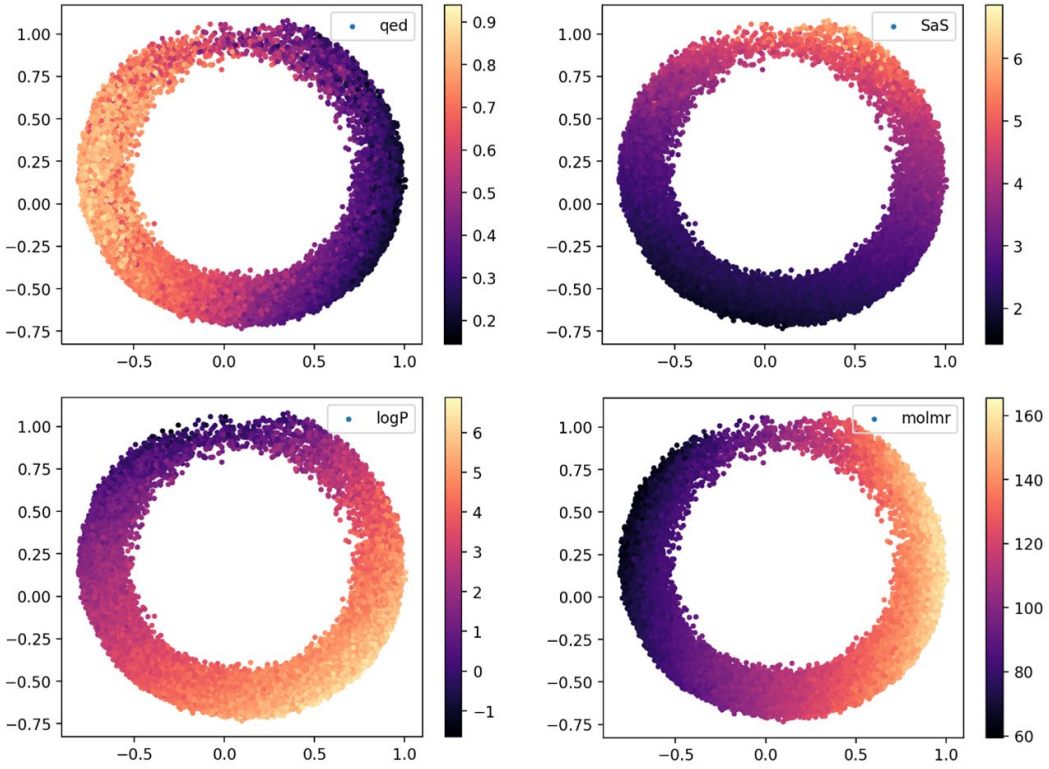
4.9. ábra: Látens mintavételezés LERP és SLERP interpolációval.

Azt tapasztaltam, hogy az íves interpolálás során generált szomszédos molekulák jobban hasonlítottak egymásra, és az egyenes helyett az ív mentén több helyen tudott érvényes molekulákat generálni a modell, vagyis kevesebb szakadás volt benne. Tehát a modell által megtanult látens reprezentációban a tulajdonságok nem lineárisan, hanem ívesen képződnek le. Egy variációs autoenkóderek látens terével foglalkozó cikk [43] szerint is egyes adatok hiperszferikus látens szerkezettel rendelkeznek, az ilyen direkcionális látens tulajdonságok pedig jobban kifejezhetőek egy gömbszerű reprezentációval. A Gauss eloszlást leváltó von Mises-Fisher (vMF) eloszlás használatával kényszeríthetjük ki, hogy a látens tér egy hipergömb szerű alakot vegyen fel. Ezt én is kipróbáltam, a 4.10. ábra mutatja a guacamol halmazból készült látens tér PCA leképezését, illetve az első 2 PCA főkomponens által kivágott altérhez közel eső pontokat is ábrázoltam, ez a 4.11. ábrán látható. Látszik, hogy valóban hiperszferikus eloszlásról van szó, illetve az adatok valóban jobban követik a vMF eloszlást, mint a Gauss priort.



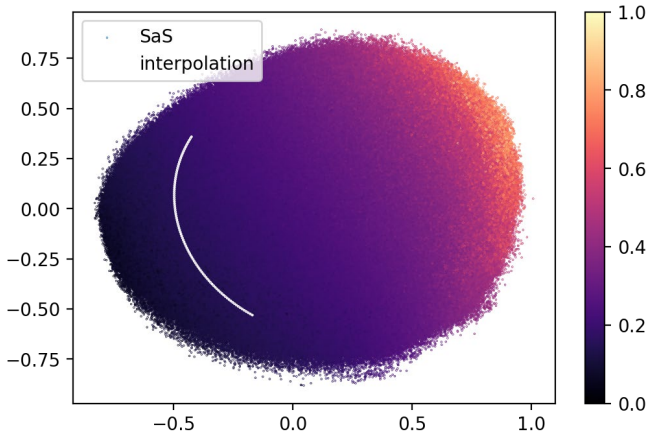
4.10. ábra: A guacamol halmaz molekuláinak vMF prior eloszlást követő reprezentációja. Az eloszlás valóban egy hipergömbre illeszkedik, és a tulajdonságbecslő képes erre a térre is kifejteni a hatását, úgy, hogy az kisebb mértékben torzul, mint a Gauss prior esetén.

Az egyes becsült tulajdonságok a gömbfelületen is élesen elkülönülnek, az alacsony értékekből kiindulva, körbe haladva magas majd ismét alacsony értékek lesznek. A 4.11. ábrán leolvashatóak a köztük lévő kapcsolatok is.



4.11. ábra: Egyes tulajdonságok eloszlása a gömbfelület egy szeletén. Leolvasható, hogy a gyógyszerszerű molekulák kisebbek, viszonylag könnyen szintetizáltaóak, logP értékük pedig se nem túl alacsony, se nem túl magas.

A vMF látens térben értelemszerűen csak a SLERP interpoláció használható a későbbiekben, de ez nem probléma, hiszen még a Gauss térben is ez bizonyult jobb módszernek. A 4.12. ábrán láthatóak a hipergömb felületén végrehajtott interpoláció által bejárt látens vektorok. Egy másik, a vMF eloszlású térre vonatkozó megkötés a dimenziószám mérete, ugyanis a gömbszerű látens tér 40-50 dimenzió felett már nem stabil. Mivel korábban arra a megfontolásra jutottam, hogy a feladat elvégzéséhez elég csupán 20-30 dimenzió is, így ez sem jelent problémát.



4.12. ábra: A SLERP interpoláció által bejárt látens vektorok ábrázolása a normalizált SaS érték szerint színezett látens gömbfelületen.

A későbbiekben fogok még példát mutatni a Gauss és vMF látens tér használatára is. A végső modellemben viszont a vMF eloszlást választottam, mert tanítása stabilabb, nem keletkeznek kiugró pontok, és a gömbszerű látens szerkezet jobban meg tudja ragadni a feladathoz szükséges információkat, a kevesebb szakadás miatt pedig több helyről tudok belőle mintavételezni.

4.1.4. Hibaarányok hatása

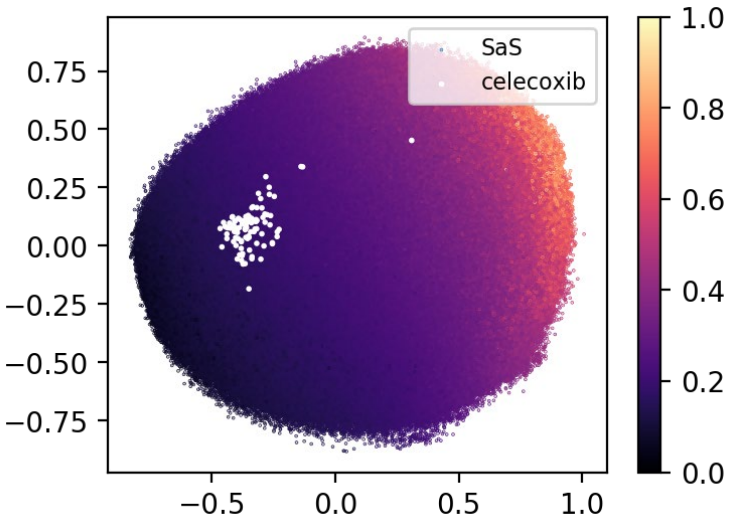
Láthattuk, hogy a veszteségfüggvény már három tagból áll, és azt is, hogy mik a hatásai a tulajdonságbecslő bevételének és a KL tag módosításának. Ezek a hatások főleg a nagyobb adathalmazon érvényesülnek, a tanításuk érzékeny a három hibatag arányára. A KL tag növelése a prior eloszlásra jobban illeszkedő látens teret eredményez, a sűrűbb, szakadás mentes látens térben történő interpolációkor és kereséskor több helyről tudunk érvénes molekulát generálni, de a szomszédos molekulák nem fognak annyira hasonlítani. Azt is láthattuk, hogy a túl nagy KL tag ronthatja a modell teljesítményét, akár 0%-os egyediség is lehet a következménye. A tulajdonságbecslő bevétele nélkül a nagyobb adathalmazokon nem különülnek el a tulajdonságok a térben, ugyanakkor túl nagy MSE választása esetén a látens tér akár klasztereződhet is, így ebből nehezebb mintavételezni, de a szomszédos molekulák hasonlóak. A nagyjából 50%-os rekonstrukció és megfelelő guacamol eredményekhez szükséges a rekonstrukciós hibatagnak is kellően nagy súlyt adni. Jellemzően, ha az egyik hiba nagyobb súllyal szerepel, akkor a többi feladatot kevésbé képes ellátni a háló. Egy összefüggő és kifejező látens teret előállítani a súlyok arányának beállításával hosszú optimalizálási feladat volt. Figyelni kellett arra is, hogy egy már beállított arány változhat, ha más adathalmazt használok, hiszen eltérő a karakterek száma és a beágyazás hossza. Ezért a súlyokat függetlenné tettem a batch és az adathalmaz méretétől is.

A hibatagok arányának változtatásán kívül azok időbeli változásával is tovább tudtam javítani a modellemet. Kezdetben a generált molekulák érvényességét mérő értékem nagyon magas, 1 körüli, az FCD értékem pedig közel 0 volt. A rekonstrukciós hibatag növelésével 0%-ról 50%-ra ugrott a helyesen helyreállított tesztmolekulák aránya, ezzel az FCD érték is emelkedett két tizedet, hiszen a modellem már képes volt a tanítóhalmazban lévő molekulákhoz hasonló kimenetet generálni. A magas érvényesség, de alacsony FCD érték problémája másnál is jelentkezett, erre adhat megoldást a változó súlyú KL divergencia bevezetése [44]. A módszer lényege, hogy a kezdeti epochokban nem figyeli a KL divergenciát, majd epochról epochra haladva egyre nagyobb súlyt ad neki, egészen addig, amíg el nem ér egy maximális küszöböt. Ezt én is alkalmaztam, sőt úgy értem el a legjobb eredményeket, ha a tulajdonságbecslőtől származó MSE tagot is hasonlóan kezeltem. Így a modellnek a kezdeti epochokban csak a rekonstrukcióra kell törekednie, és végül magasabb FCD értéket ér el anélkül, hogy az érvényesség érték jelentősen csökkenne. Végül sikerült a többi modellel összehasonlítható nagyságrendre emelnem a modellem FCD értékét.

4.1.5. Reprezentációból eredő probléma

A súlyok arányának módosításával javítható volt az FCD érték, ugyanakkor nem tudtam így sem 50% feletti rekonstrukciót elérni, és a már ismert gyógyszerek újrafelfedezése sem sikerült. A guacamol benchmark értékei mellé közölték az egyes gyógyszermolekulák helyreállítási kísérleteinek eredményét is. Például a már ismert, gyulladáscsökkentő hatása miatt alkalmazott Celecoxib molekulát is megpróbálták az egyes modellekkel helyreállítani. Pár modell esetében sikerült hiba nélkül helyreállítaniuk a molekulát, ahol nem, ott a Tanimoto hasonlóságát adták meg a leghasonlóbb generált molekulára, ami egy ujjlenyomatokkal dolgozó hasonlóságérték. Magában a guacamol adathalmazban a leghasonlóbb molekula 0.505 hasonlóságértékkel rendelkezik. Az én modellem 0.3 értékűt tudott először generálni, ha azonban nem csak arról a pontról mintavételeztem ahová az enkóder leképezte az ismeretlen gyógyszert, hanem kicsivel körülötte is, akkor sikerült 0.5-0.6 körüli értékeket elérnem.

A rekonstrukciós probléma a SMILES reprezentáció tulajdonságaiból ered. Egy szöveges reprezentáció pontosan egy molekulához tartozhat, ugyanakkor egy molekulának a gráf bejárás kezdőpontjától függően számos különböző szöveges reprezentációja lehet. Megpróbáltam több SMILES reprezentációjából is visszaállítani a Celecoxib molekulát, így már az esetek többségében sikerült hiba nélkül visszaállítani azt, az így mintavételezett látens vektorokat a 4.13. ábra szemlélteti.

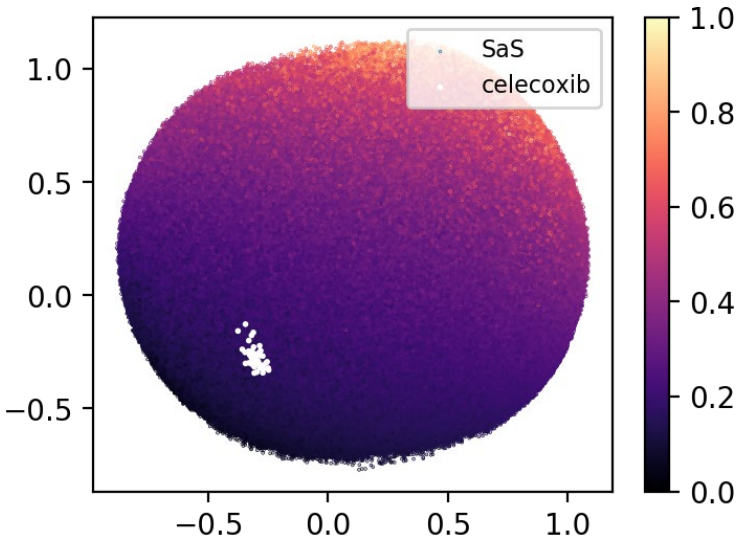


4.13. ábra: Celecoxib különböző reprezentációnak elhelyezkedése a normalizált SaS érték szerint színezett látens térben, látszik, hogy egyazon molekula különböző reprezentációi egymástól messze helyezkednek el.

Látszik, hogy a molekula különböző SMILES reprezentációi távol helyezkednek el a térben, hiszen a modell nem a molekulát, hanem annak egy SMILES reprezentációját tanulta. A problémára lehetséges megoldásként kipróbáltam a tanítóadatok dúsítását azonos molekula eltérő SMILES reprezentációival, illetve az „ALL SMILES” [45] módszert is. Ez utóbbi a VAE bemenetének több, egymástól eltérő, de azonos molekulához tartozó SMILES szót ad meg. Az elvárt kimenet ezektől is eltérő, de

ugyanazt a molekulát reprezentáló szavakból áll, ennek köszönhetően a modell nem magát a szöveges reprezentációt, hanem a hozzá tartozó molekula tulajdonságait tanulja meg. Egyik módszer esetén sem tapasztaltam javulást, sőt a tanítás konvergenciáját is jelentősen rontották.

Mivel a problémát a nem egyedi reprezentációk okozzák, ezért megpróbáltam lecserélni a SMILES szavakat a molekulákra nézve egyedi ujjlenyomatokra, az LSTM rétegeket pedig sima előrecsatolt rétegekre. Ha csak a bináris ujjlenyomat információt használtam a VAE be-, és kimenetén, akkor nem sikerült molekulákat generálnom a látens térből, de a tulajdonságbecslő jól működött így is. Ha viszont együtt kapta meg a SMILES és a bináris szerkezeti adatokat a modell, akkor ugyanúgy képes volt generálni, de az ujjlenyomat által hordozott, az egyes molekulákra egyedi információ is megjelent a látens térben. Ha ezután megnézzük a Celecoxib különböző reprezentációinak elhelyezkedését a 4.14. ábrán, akkor látjuk, hogy azok már a látens térben közelebb helyezkednek el, hiszen mindegyik tartalmaz közös, ujjlenyomatból tanult információt is.

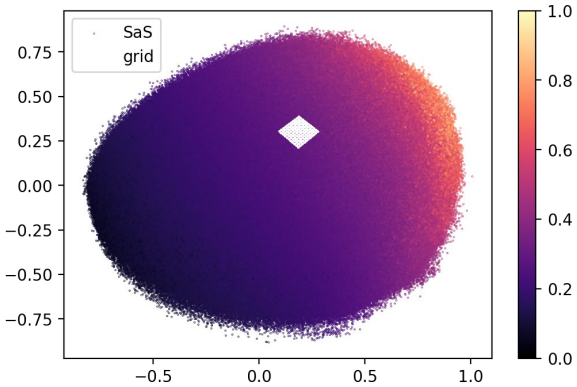


4.14. ábra: Celecoxib különböző reprezentációnak elhelyezkedése az ujjlenyomattal kiegészített modell látens terében, látszik, hogy egy molekula egy kisebb, összefüggő területet foglal el a térből.

Látszik, hogy még mindig egy kisebb területet foglalnak el az egyes molekulák a látens térből, nem egy pontot, a terület az ujjlenyomat bevételével viszont már egy pont köré csoportosul. Ezen kívül hosszabb tanítással, illetve nagy MSE taggal tovább csökkenthető a terület mérete, hiszen, ha kellően sok tulajdonságot becslünk, akkor ezek együttese is egyedi a molekulára.

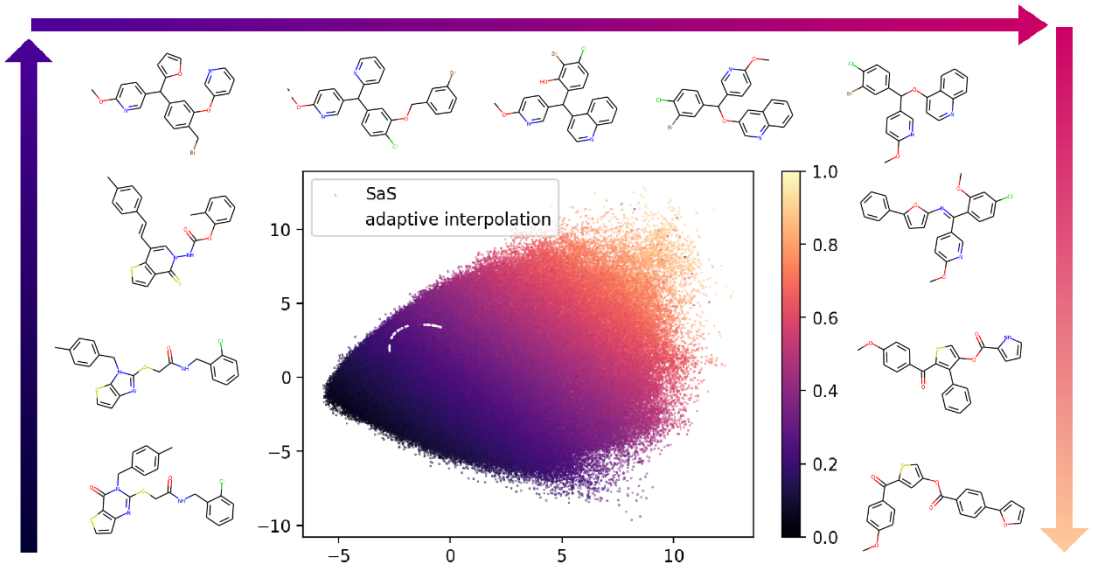
A 4.15. ábrán megfigyelhetőek ezek a területek, egy molekula körül elhelyezkedő egyenletes rácspontokon mintavételeztem a látens teret, ahonnan sikerült érvényes molekulát generálni azt kirajzoltam. A rácspontokat a látens térben szemlélteti a 4.16. ábra.

4.15. ábra: Mintavételezés eredménye egy molekula körül, megfigyelhető, hogy a szomszédos molekulák valóban hasonlítanak egymásra. Az egyes molekulához tartozó régiók ábrázolásával látszik, hogy azok egy-egy nagyobb, összefüggő területet fednek le a látens térből.



4.16. ábra: Mintavételezett rácspontok ábrázolása a látens térben, a rácspontok egyenletesen helyezkednek el egy kis térrészen belül.

Látszik, hogy az egyes molekulák különböző méretű helyet foglalnak el, van amire több rácspont is esett, azaz itt felülmintavételeztem a teret. Vannak alulmintavételezett részek is, ahol a szomszédok nem hasonlítanak egymásra, valószínűleg van közöttük más, nem ábrázolt molekula. Ez a két pont közötti LERP és SLERP interpolációnál is felmerült, a szomszédos molekulák gyakran megegyeztek, és sok ugrásszerű változás is volt az interpoláció alatt. Ennek javítására implementáltam egy adaptív lépésközöket használó SLERP interpolációt, ami, ha túlságosan eltérő molekulákat lát, akkor csökkenti, ha többször ugyanazt a molekulát látja, akkor növeli a lépésközét. Ennek eredménye a 4.17. ábrán látható.



4.17. ábra: Adaptív SLERP eredménye, látszik, hogy az ív nem minden pontjáról mintavételezett, így már nincsenek benne ismétlődő molekulák. Még így sem minden esetben hasonlítanak a szomszédos molekulák, hiszen a VAE a molekula szerkezete helyett annak SMILES reprezentációját tanulta.

Elmondható, hogy az ujjlenyomat által hordozott információ bevételével az azonos molekulák reprezentációi közelebb kerültek egymáshoz a térben, így egyszerűbben helyre tudok állítani eddig nem látott hatóanyagokat, és a rekonstrukció is megnőtt 70%-ra. Ugyanakkor az interpolálásoknál látott probléma még megmaradt, erre végső megoldást csak a SMILES reprezentáció leváltása adna. Viszont mivel ez a probléma sem a benchmark eredményeket, sem a későbbi célfüggvényre generálást nem befolyásolja, így maradtam az egyszerűen használható SMILES és bináris ujjlenyomat kombináció mellett.

4.1.6. Tanítás hiperparaméterei

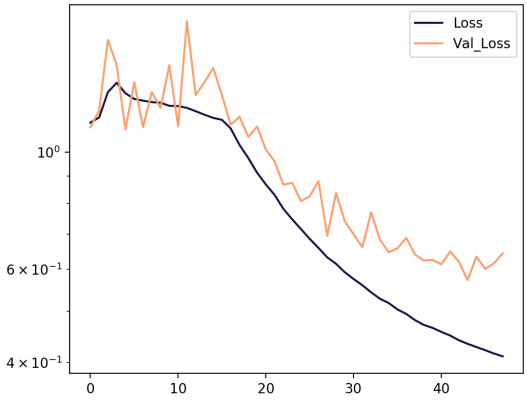
Eddig a látens térrel, súlyfüggvénnyel és visszacsatolt rétegekkel kapcsolatos paraméterek kiválasztásának folyamatát mutattam be, most gyorsan áttekinteném a tanítás főbb hiperparamétereit is.

A konvergencia stabillá tételére kipróbáltam különböző regularizáló módszereket, a batch normalizációt, a súlyokra vett L2 regularizációt, a gradiens vágást és a dropout technikát.

Mind a négy regularizáció megszűntette a tanítás közben kiugró hibaértékek problémáját, de a batch normalizáció alkalmazásával konvergált a leggyorsabban a tanítás, így ezt választottam. Ezen kívül vizsgáltam különböző optimalizálási függvényeket, a legjobb eredményeket egy 0.001 bátorsági tényezővel rendelkező RMSprop segítségével értem el. A tanítással kapcsolatos fontos érték még a batch méret, ennek méretét 128-ra állítottam, ennél nagyobb batch méret rontotta a rekonstrukciós képességet.

4.1.7. Benchmark eredmények

A bemutatott hiperparaméterekkel rendelkező modellt 72 epochig tanítottam, ez 4 napig tartott. A veszteségfüggvény alakulását láthatjuk a 4.18. ábrán.



4.18. ábra: Tanítási és validációs hiba alakulása a 72 epoch alatt. Látszik, hogy a kezdeti epochokban növekedett a hiba, ez az időszak alatt nőtt egyenletesen a KL tag és az MSE súlya. A tanítás 72 epochig tartott, utána már nem csökkent volna tovább a validációs hiba.

Az 1. táblázatban látszik a „Hyperspherical Molecular VAE” (HM-VAE) nevű modellem guacamol benchmarkon elért eredménye, és a ranglistás modellek eredményei. Fontos megjegyezni, hogy az egyediség érték részben a SMILES reprezentációk nem egyértelműsége miatt ilyen magas az összes módszernél, így összehasonlításkor az érvényességet, újszerűséget és a tanítóhalmazzal vett hasonlóságot mérő értékek a mérvadóak.

benchmark	Random Sampler	Graph MCTS	ORGAN	SMILES LSTM	VAE	AAE	HM-VAE
Validity	1.000	1.000	0.379	0.959	0.870	0.822	**0.9611**
Uniqueness	0.997	1.000	0.841	**1.000**	0.999	**1.000**	0.9997
Novelty	0.000	0.994	0.686	0.912	0.974	**0.998**	0.9718
KL	0.998	0.522	0.267	**0.991**	0.982	0.886	0.9688
FCD	0.929	0.015	0.000	**0.913**	0.863	0.529	0.8505

1. táblázat: Guacamol benchmark eredmények. Az újszerű, de az eredeti adatokhoz hasonló molekulákat generálni képes modellek közül a mérvadó értékek szerinti összehasonlításban a SMILES LSTM és a HM-VAE modell emelkedik ki.

A közzétett ranglistán a molekulák gráf reprezentációjával dolgozó MCTS módszer nem képes a tanítóhalmazban szerepelő molekulákhoz hasonló molekulákat generálni, habár a gráf reprezentáció miatt mindig érvényesek a generált kimenetei. A véletlenszerű mintavételező nem képes új molekulákat előállítani. Az ORGAN modell pedig szinte minden értékből alacsony pontszámot ért el. Tehát a többi módszerhez érdemes hasonlítani az én modellemet. Sikerült a VAE modellhez hasonló újszerűséget és FCD értéket elérnem úgy, hogy az érvényességem jóval magasabb maradt. Az AAE alapú modell valamivel újszerűbb molekulákat tud generálni, de ennek az az ára, hogy azok szerkezete nagyban eltér a tesztadatban szereplő molekuláktól, és kisebb százalékban képes érvényes kimenetet előállítani. Egyedül a SMILES LSTM modellnél nem sikerült jobb eredményeket elérnem, újszerűbb molekulákat tudok generálni, de a KL és FCD értékem kicsivel alacsonyabb. Leginkább az FCD értékben marad le a modellem, vagyis a generált molekulák a kémiai értéket nézve hasonlítanak a tanítóhalmazhoz, de szerkezetük kis mértékben eltérő, ugyanakkor elmondható, hogy ez a tulajdonság az egy célpontra történő generáláskor még akár előnyös is lehet.

Rekonstrukciót illetően a modell a tesztadatok 80%-át helyre tudja hiba nélkül állítani az első próbálkozásra. Illetve a Celecoxib gyógyszermolekula újrafelfedezését próbálva sikerült azt a látens teremben kódolni és újra előállítani a dekóderrel, az 1. táblázatban közölt modellek közül erre még a SMILES LSTM volt képes.

Összességében elmondható, hogy a modellem benchmark értékei meghaladják a legtöbb közzétett modell által elért értékeket. A HM-VAE modell által generált molekulák kiemelkedően magas százalékban érvényesek és egyediek. Nagy részük nem szerepel a modell által korábban látott bementi molekulák között, de kémiai tulajdonságokban és szerkezetileg is hasonlítanak rájuk. A modell ezen kívül képes korábban nem látott gyógyszermolekulákat rekonstruálni a látens teréből. Eddig még nem tértem ki rá, de mindezek mellett a modellem képes egy célfüggvényt figyelembe véve is generálni.

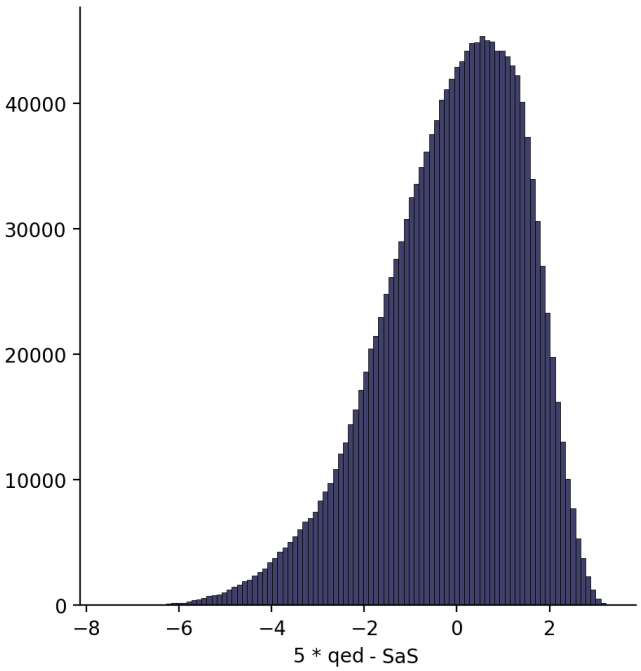
4.2. Célfüggvényre generálás

Már rendelkezem egy mérhető teljesítményű generatív modellel. Megmutattam azt is, hogyan képes a diszkrét molekulatérből áttérni egy folytonos látens térbe, amiben tetszőleges tulajdonságok mentén el tudja helyezni a molekulákat, illetve, hogyan képes abból mintavételezni. Említettem, hogy a folytonos látens reprezentáció és tulajdonságbecslő célja az, hogy könnyebben tudjunk új molekulákat generálni a tér bejárása közben, illetve a látens térbeli elhelyezkedés és egy molekula kémiai tulajdonságai között nagy legyen az összefüggés. Arról viszont még nem esett szó, hogy ezeket a célokat a célfüggvényre történő hatóanyaggenerálás megvalósítása miatt tűztem ki. Ebben a fejezetben megmutatom, hogyan lehet egy megadott célfüggvény szempontjából megfelelő molekulákat generálni a modellem segítségével.

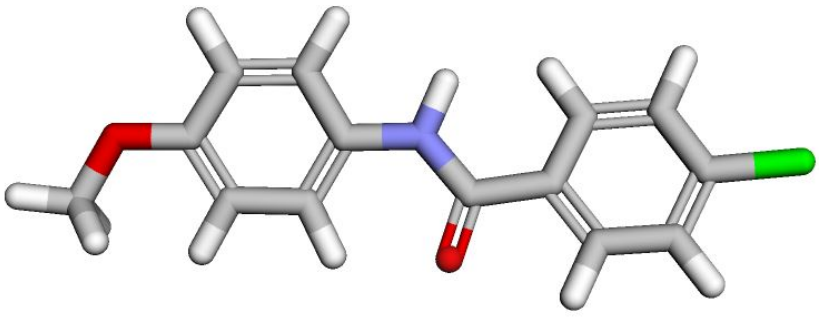
4.2.1. Célfüggvény meghatározása

Mielőtt egy célfüggvényre szeretnénk generálni, definiálni kell azt. Az előfeldolgozás részben említett és azokhoz hasonló értékekből bármilyen célfüggvényt alkothatunk, ha a modell tulajdonságbecslője tanult az egyes értékeken, akkor az itt bemutatott módszer

működni fog rá. Például, ha a cél gyógyszerszerű, de könnyen szintetizálható molekulák előállítása, akkor egy olyan célfüggvényt kell alkotnunk, ami egyszerre maximalizálja a qed értéket, és minimalizálja a SaS értéket. Az erre a célra leggyakrabban használt célfüggvény az 5 * qed – SaS, ahol az ötös szorzó az értékkészletek miatti különbségek kiegyensúlyozására kell. A 4.19. ábra mutatja a guacamol adathalmaz ezen érték szerinti eloszlását. Látszik, hogy egy 2 feletti érték, már jóval átlag feletti molekulákhoz tartozik, a maximum érték pedig 3.321604799709913, az ehhez tartozó molekula látható a 4.20. ábrán.



4.19. ábra: 5 * qed - SaS eloszlása a guacamol halmazon, egy 2 feletti értékkel rendelkező molekula a tanítóadat 95%-ánál jobb.

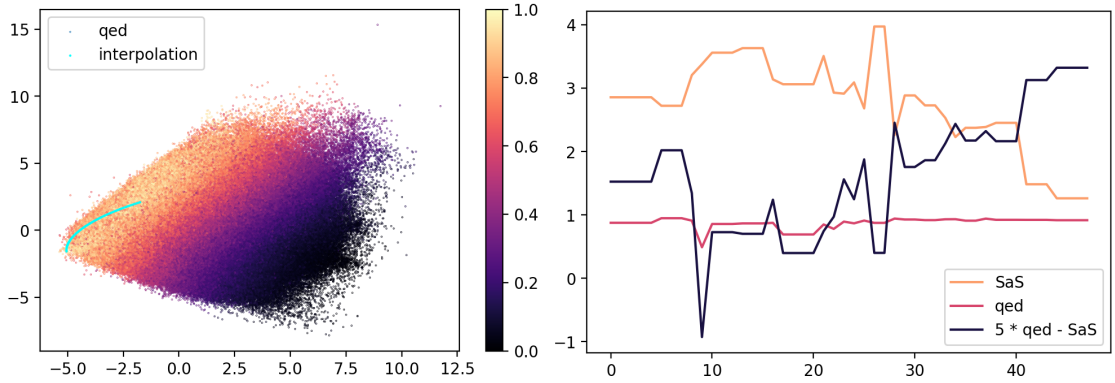


4.20. ábra: A guacamol halmazban legmagasabb 5 * qed - SaS értékkel rendelkező molekula szerkezete.

4.2.2. Keresés interpolálással

A már bemutatott interpolálás segítségével is találhatunk új molekulákat, amik a célfüggvény tekintetében jobbak az átlagnál. A példán egy tanítás után interpolálok a leginkább gyógyszerszerű és a legmagasabb 5 * qed - SaS értékű molekula között. A 4.26.

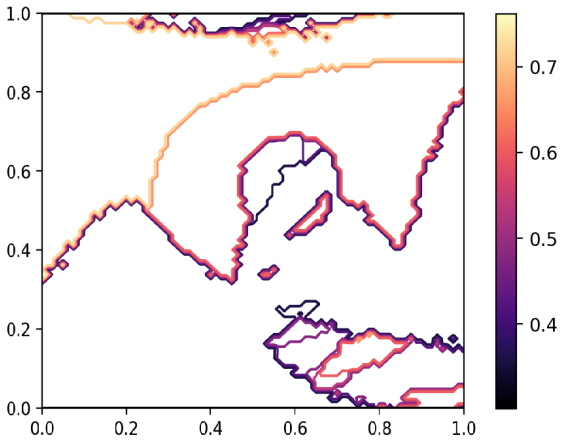
ábrán láthatjuk, hogy a látens térben milyen pontokat érintett az interpoláció, illetve az ezekhez a molekulákhoz tartozó tulajdonságokat is.



4.21. ábra: Interpolálás eredménye a tanítóhalmazban legmagasabb qed és a legmagasabb 5 * qed - SaS értékű molekula között, a bejárt útvonalon szereplő molekulák optimalizálási értéke sok lokális szélsőértékkel rendelkezik.

Ezzel a módszerrel 5, a tanítóadatban nem szereplő molekulát találtam, melyek optimalizálni kívánt értéke 2 feletti. A legmagasabb ilyen érték a 2.34 volt.

Elmondható, hogy a tulajdonságbecslő alkalmazása ellenére sem monoton a látens tér az egyes tulajdonságokra. Az optimalizálási felület rengeteg lokális szélsőértékkel rendelkezik, de látható egy növekedési tendencia az egyes irányokban. A 4.22. ábra mutatja a célfüggvény eloszlását a látens tér egy kiragadott részében.



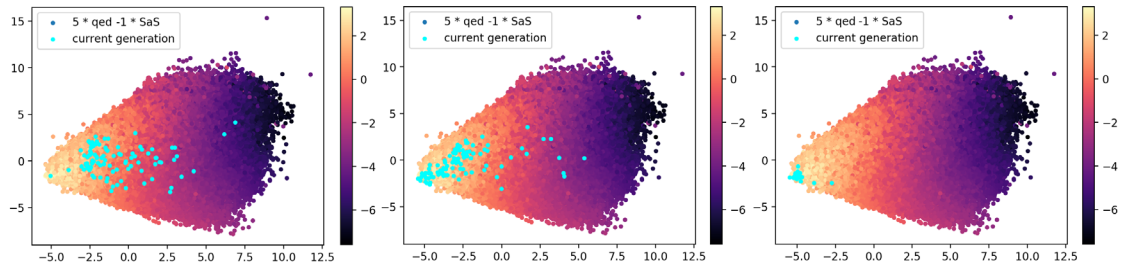
4.22. ábra: A normalizált 5 * qed – SaS értékek ábrázolása a látens tér egy szeletében. Látszik, hogy a tulajdonságokból összeállított optimalizálási függvénynek még egy kis térrészen belül is több lokális szélsőértéke lehet.

4.2.3. Keresés genetikus algoritmussal

A felület jellege és a módszer könnyű beépíthetősége miatt genetikus algoritmussal [46] végeztem a keresést. Az egyedeknek a molekulákat, genotípusuknak pedig a látens reprezentációt választottam. Az egyedek fitneszét a látens pontból történő generáció során kapott molekulára kiértékelt célfüggvény adja. A legjobb fitneszű egyedek képzik

a következő generáció szülőinek csoportját. Gauss látens tér esetén az egypontos, vMF eloszlás esetén pedig a két szülő közötti SLERP interpoláció bizonyult a legjobb keresztezési módszernek. Mutáció gyanánt pedig az egyes elemekhez egy kis véletlenszerű számot adtam hozzá. A populáció legjobb egyedét átalakítás nélkül megtartottam a következő populációban is.

Ha az egyedek fitneszét a tulajdonságbecslő által becsült értékek adják, akkor az optimalizáció végére előfordulhat, hogy a teljes populáció a látens tér egy olyan területén helyezkedik el, ahová egy tanítóadat se esett. Ez probléma, hiszen onnan valószínűleg nem tudunk érvényes molekulát generálni. Ez a jelenség akkor fordul elő gyakran, ha az optimum az érvényes látens tér szélén helyezkedik el, ami a vMF eloszlás esetén mindig igaz, ilyenkor a becslő azt hiszi, hogy az adott irányba tovább haladva nő az optimalizált érték. Illetve a becslő nem ad pontos értékeket, ezzel az egész keresést rossz irányba viheti. Ezért választottam fitnesz függvénynek a generált molekulák valódi tulajdonságaiból számított célfüggvényt. Amelyik pontból nem lehet molekulát generálni, annak pedig 0 a fitnesze, nem örökíti a génjeit tovább, így nem tudnak az egyedek nem érvényes tartományra konvergálni. Az utóbbi fitnesz függvénnyel történő keresés folyamatát mutatja a 4.23. ábra.



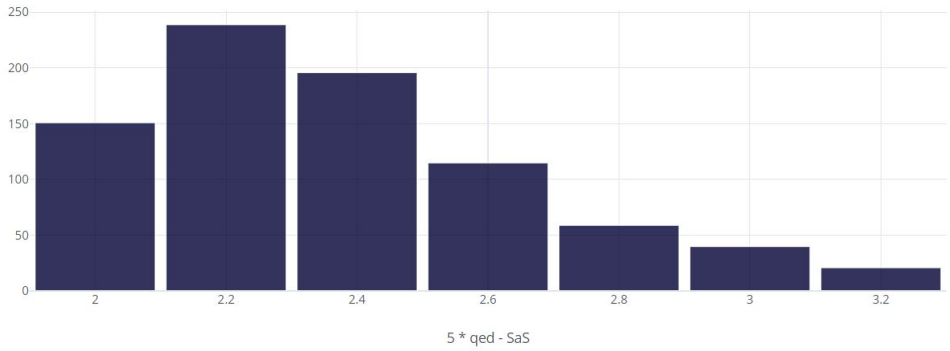
4.23. ábra: Optimalizálás során az egyes generációk egyedei ábrázolva a látens térben. Látható, hogy az egymást követő generációk valóban a látens tér optimálisnak mondható része felé konvergálnak.

Ennek a fitnesz függvénynek viszont hátránya, hogy nem mindenhol kiértékelhető. Egy populáció többségének 0 lesz a fitnesz értéke, ennek elkerülése érdekében átdefiniáltam azt. Ha egy pontból nem tud generálni a modell, akkor az egyedet kicsit mutálva újra próbálkozik mindig egy kicsivel nagyobb mutációval. Ez több véletlenszerűséget visz az optimalizációs folyamatba, viszont fontos megjegyezni, hogy a cél most nem annak legjobb pontnak a megkeresése, ahová az algoritmus bekonvergál. Mivel a célfüggvény is csak tapasztalati úton lett kiválasztva, nincs is ételme az aszerinti legjobb molekula megkeresésére, csupán a látens tér megfelelő részének bejárásával kellően jó molekulákat szeretnék keresni.

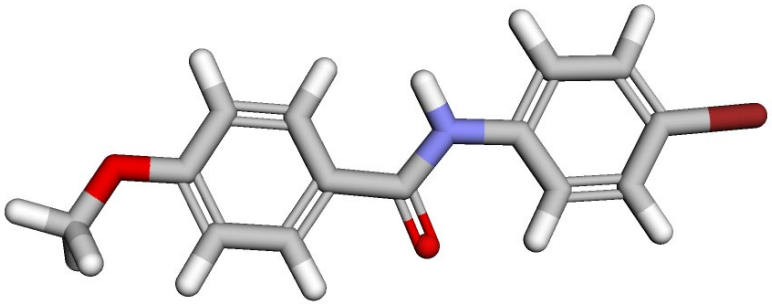
4.2.4. Generált molekulák kiértékelése

Egy 50 lépéses keresési folyamat alatt körülbelül 830 olyan új és különböző molekulát találtam, melyeknek 5 * qed - SaS értéke 2 felett volt, ezek eloszlását mutatja a 4.24. ábra. Ha 200 lépésig optimalizáltam kicsivel nagyobb mutációval, akkor 2000 új molekulát talált a modell. Közülük 80 molekula értéke 3 feletti, ezek az eredeti adathalmaz 99.96%-

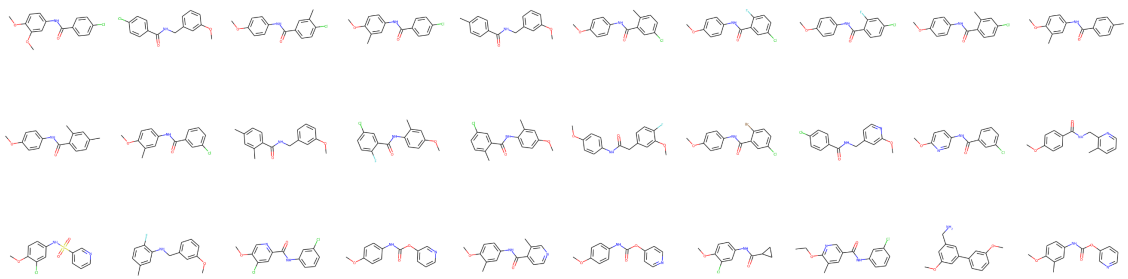
a felett helyezkednek el. Sikerült az adathalmazban lévő legjobb molekulánál magasabb értéket elérnem, ez 3.357592761673991 volt, a hozzá tartozó molekulát mutatja a 4.25. ábra. Továbbá megjelenítettem a 3 feletti értékkel rendelkező molekulák közül párat a 4.26. ábrán, láthatjuk, hogy a szerkezetük hasonló. Ennek egyik oka, hogy a guacamol adathalmazban szereplő legmagasabb értéket elérő molekula is ilyen szerkezetű, így az e körül mintavételezett molekulák is. Másik oka pedig, hogy valóban ilyen jellegű molekulák maximalizálják a célfüggvényt, hiszen gyógyszer szerűek, de a két gyűrű miatt egyszerű őket szintetizálni. Ha az optimalizálandó értékben a qed értéknek nagyobb súlyt adtam, akkor egymástól eltérő, diverz molekulákat tudtam generálni.



4.24. ábra: A generált molekulák 5 * qed – SaS szerinti eloszlása.



4.25. ábra: A legmagasabb értéket elérő molekula szerkezete, látszik, hogy szinte megegyezik az adathalmazban található legmagasabb 5 * qed – SaS pontot elérő molekulával.



4.26. ábra: További 30 darab, az 5 * qed – SaS célfüggvényre generált molekula. Látszik, hogy szerkezetük nagyon hasonló, mindegyikben megtalálható 2 gyűrű, hasonlítanak a tanítóhalmazban lévő, célfüggvény szempontjából legjobb molekulára.

A modellem tehát valóban képes egy célfüggvényt figyelembe véve új, szintetizálható és gyógyszer szerű hatóanyagokat generálni. Ez már specifikusabb megoldás, mint az általános molekulagenerálás volt, de elmondható, hogy ezekből a számított értékekből nem lehetséges használható célfüggvényt összeállítani, az így generált hatóanyagok még mindig túl általánosak lesznek.

4.3. Egy célpontra történő generálás

Láthattuk, hogy a modell generatív képessége megfelelő, illetve képes a látens tér egy célfüggvény által meghatározott részéről mintavételezni, így a célfüggvényre megfelelő molekulákat generálni. A következő lépés a megfelelő célfüggvény kiválasztása.

A legfontosabb kitűzött célom az egy adott célpontra történő hatóanyag generálása volt, ezért ennek megfelelően alakítottam tovább a célfüggvényemet. A szükséges adatok beszerzését, célfüggvény kialakítását és a generálás eredményét szeretném ebben a fejezetben bemutatni.

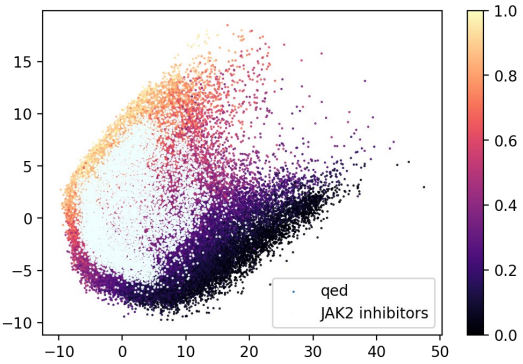
4.3.1. BindingDB adathalmaz

Keresnem kellett egy interakciós adatokat is tartalmazó adatbázist. A választásom a BindingDB halmazra esett, annak kellően nagy mérete miatt. Az RDKit-tel nem feldolgozható és 180 karakternél hosszabb SMILES szavak elhagyása után maradt 8.160 célpont, 792.638 darab molekula, és összesen 1.789.093 interakciós bejegyzés. A bejegyzésekből a Ki értéket tároltam, pontosabban, ahol ez nem volt elérhető, ott a Kd, EC50, illetve IC50 értéket használtam, hiszen ez a négy érték szinte megegyezik. Az így kapott érték a teljes adatra lognormális eloszlást mutat, és sok kiugró értékkel rendelkezik. Az előzőek miatt ennek logaritmusát használtam, a későbbiekben erre csak Ki néven utalok.

Az adathalmazban a „Tyrosine-protein kinase JAK2” célpontra található a legtöbb bejegyzés, összesen 12.391 érték különböző molekulákra. A későbbiekben ezért ezt a célpontot használtam.

Vizsgáltam az adathalmaz egyes kémiai értékeinek eloszlását, és a közöttük lévő kapcsolatokat. Arra jutottam, hogy lehet értelme a korábban vizsgált qed, SaS és logP értékek mentén tagolni a látens teret, ugyanis az eloszlásuk egy adott célpontra megfelelő molekulákon és az egész adathalmazon eltérő. Ezt hipotézis teszteléssel támasztottam alá, az egyes kémiai tulajdonságok Kolmogorov-Smirnov statisztikája kicsi lett, de az eltérés jelentős, közel 0 p-értékű.

Ezt igazolták az első tanítás eredményei is, amit a 4.27. ábra szemléltet. Egy Gauss látens térrel rendelkező modellt tanítottam a BindingDB halmazon. A látens térben az egyes kémiai értékek elkülönültek a tulajdonságbecslőnek köszönhetően, és az egy adott célpontra kis Ki értékkel rendelkező molekulák is viszonylag közel helyezkedtek el, ahhoz képest, hogy a modell a tanítás alatt nem kapott DTI információt.



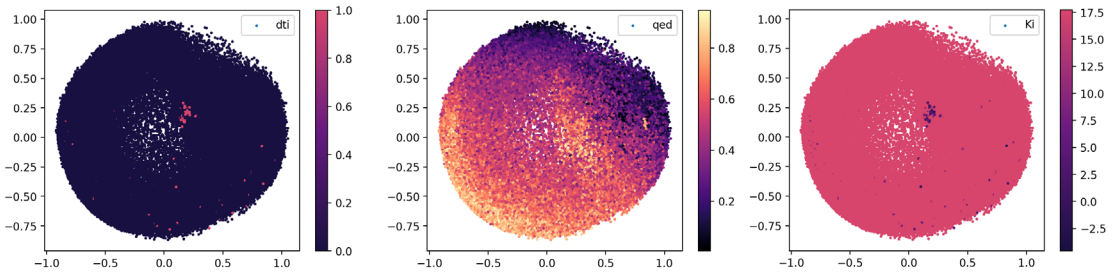
4.27. ábra: A qed szerint színezett BindingDB látens terében a JAK2-re ható molekulák ábrázolva világoskék színnel. Az aktív inhibitorok nincsenek véletlenszerűen szétszórva a látens térben, a tér gyógyszerszerűbb felének egy részén helyezkednek el.

4.3.2. Célfüggvény kiegészítése

A fenti részeredmény bizonyította, hogy a modell képes lehet a kötési információt is kezelni, és a megfelelő molekulákat egymáshoz közel elhelyezni a térben. Ennek elősegítésére implementáltam egy, a tulajdonságbecslőhöz hasonló interakcióbecslőt, ami regresszió helyett klasszifikációs problémát old meg a modell látens teréből kiindulva.

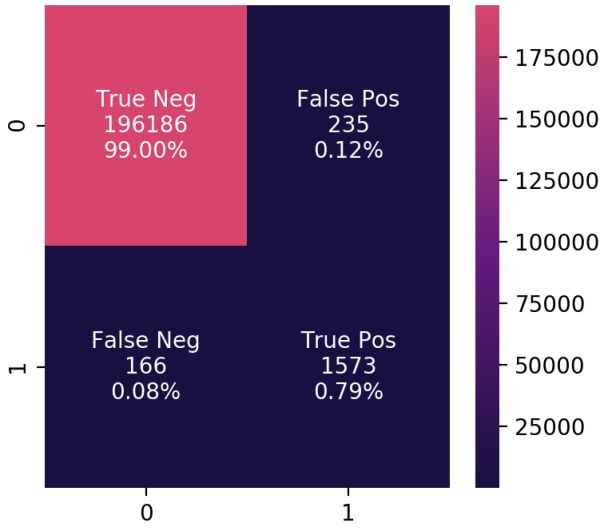
Ezen kívül új értékeket is számítottam az adathalmazokra, ezekkel egészítettem ki a meglévő kémiai értékeket, amikre eddig optimalizálni lehetett. Az egyik ilyen érték a Ki, ami a korábban említett érték a JAK2 célpontra nézve, ahol nem volt megadva adat, oda többi adat maximumát helyettesítettem be, vagyis úgy tekintettem, hogy azok a molekulák nem hatnak a célpontra. A másik új érték a DTI, ami 0 vagy 1 lehet, tehát ezt már az interakcióbecslő fogja becsülni a látens térből. Értéke akkor 1 ha a hozzá tartozó Ki érték 1000 nM alatti. Végül kiszámítottam egy „similarity score” nevű értéket is, ami a gyógyszernek nyilvánított molekulák közül a leghasonlóbbal vett Tanimoto hasonlóság értékével egyenlő.

Az interakcióbecslővel kiegészített modellt tanítottam a BindingDB adathalmazon. Az így kapott látens teret mutatja a 4.28. ábra. A becslőt az autoenkóderrel együtt tanítva valóban a látens térben jobban szétválnak az egy célpontra ható molekulák a többitől.



4.28. ábra: Interakcióbecslő hatása a látens térre. Bal oldalt látható, hogy a pozitív DTI osztály elemei egy kis helyen csoportosulnak. Emellett egy tulajdonságbecslő a már megszokott qed és SaS értékek mellett tanulta a Ki értéket is. A jobb oldalon látszik, hogy azok a molekulák kerültek a pozitív DTI osztályba, melyek Ki értéke a JAK2-re nézve alacsony volt.

Az interakcióbecslő jóságát, és egyben a látens tér kifejezőképességét szemlélteti a 4.29. ábra. A módszer arra elég, hogy elkülönítse a látens térben a molekulákat, de a DTI érték becslésére találhatók ennél jobb modellek is.



4.29. ábra: Tévesztési mátrix a tesztadatokon, a látens térből egészen pontosan becsülhető a DTI információ, még annak ellenére is, hogy a pozitív osztályban csak pár molekula szerepelt.

4.3.3. DTI adatok becslése

A hiányzó adatok pótlásával a Ki érték szinte egy bináris változóhoz közelített, az adatok nagyrészére egy fix értéket vett fel, és csak kis részre volt ennél kisebb. Valamint eddig azzal a hibás feltételezéssel éltem, hogy amelyik molekula és célpont között nincs megadva adat, ott nem is lehet interakció. Az előzőek javítására implementáltam egy DTI becslőt, és a becsült DTI értékekkel dolgoztam tovább.

Egy egyszerűen implementálható becslőt [26] választottam, aminek elég csak a molekulák szerkezetét ismernie, pontosabban a már korábban kiszámolt ujjlenyomatokra van szüksége. A módszer a „Random matrix theory” (RMT), vagyis a véletlen mátrixok elméleten alapul, miszerint véletlenszerű Gauss mátrixok kovarianciamátrixának sajátértékei „Marcenko-Pastur” (MP) eloszlást követnek. Egy kritikus érték feletti sajátérték jelenléte nem lehet véletlen, statisztikailag nem elhanyagolható. Az RMT szerint kritikus sajátérték:

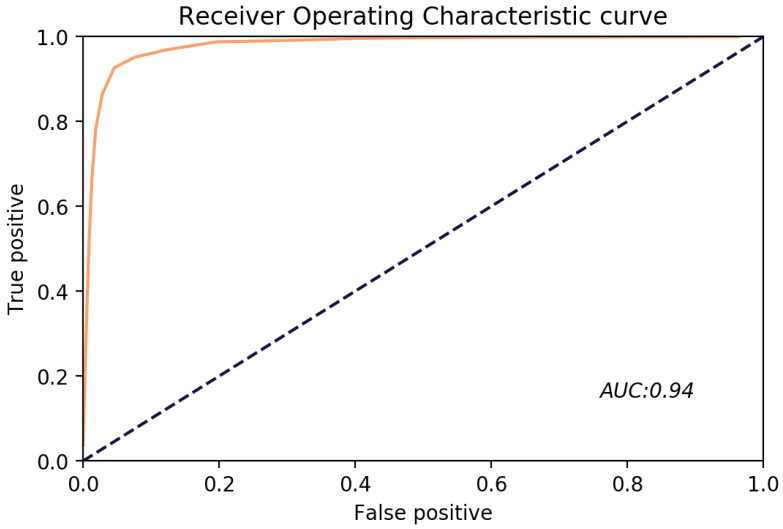
$$\alpha_{MP} = \left(1 + \sqrt[2]{\frac{p}{N}}\right)^2,$$

ahol p az oszlopok számával, N pedig a sorok számával egyezik meg.

Az algoritmusnak szüksége van egy csak pozitív osztályokból, vagyis csak a célpontra ható molekulák ujjlenyomataiból álló tanító és teszthalmazra, illetve egy véletlenszerű, nem kötő molekulákból álló negatív teszthalmazra is. Első lépésként a tanítóhalmazban szereplő, 2048 bites ujjlenyomat vektorokat egy mátrixba szervezzük, majd elhagyjuk az

ismétlődő és a nulla szórású oszlopokat és normalizáljuk a mátrixot az oszlopok mentén. Az így kapott mátrix kovarianciamátrixát képezzük és vesszük annak sajátérték-sajátvektor felbontását. Az algoritmus idáig szinte megegyezik a PCA algoritmussal. Ezután csak az RMT által szignifikánsnak vélt sajátértékekhez tartozó sajátvektorokat tartjuk meg, ezt használjuk később becslésre. Becsléskor az egyes molekulákra kiszámoljuk a sajátvektorok által kifeszített altérre vetített távolságát. Ha ez a távolság kisebb, mint egy epszilon érték, akkor az adott molekulát hatóanyagnak tekintjük. Az epszilon értékét úgy állítjuk be, hogy a tanítóadat 95%-át sorolja pozitív osztályba.

Implementálás során a tanítóhalmazba 6.000, a pozitív teszthalmazba 3.000, a negatív teszthalmazba pedig 30.000 molekulát válogattam be. Az algoritmus 175 releváns sajátvektort talált. A pozitív és negatív teszthalmazon vett becslések alapján a „true positive” (TP) ráta 0.9269 lett, míg a „false positive” (FP) 0.0537. Különböző epszilon értékek mellett kapott eredményekből rajzoltam ki a becslő „receiver operating characteristic” (ROC) görbéjét a 4.30. ábrára, az „area under roc curve” (AUC) érték 0.94, ami meglepően magas egy olyan módszernél, ami nem igényli a célpont fehérje felépítésének ismeretét.



4.30. ábra: RMT becslő ROC görbéje különböző epszilon értékek mellett.

A teszthalmaz elemeiről viszont nem lehetett tudni, hogy valóban nem lépnek interakcióba a célponttal, csupán véletlenszerűen választottam ki őket, így lehet, hogy közöttük is valóban van hatóanyag, így a FP ráta a valóságban kisebb is lehet. A módszert annyiban fejlesztettem tovább, hogy vizsgáltam a más célpontokra kötő molekulákat is. A más hatóanyagok alteréhez túl közel lévő molekulákat negatív osztályba soroltam, még akkor is, ha az algoritmus eredeti része pozitív osztályba sorolta volna azokat. Az újítással úgy sikerült tovább csökkenteni a FP rátát 0.0384-ra, hogy a TP ráta csupán 0.9233-ra csökkent le. Ezzel összességében sikerült a korábbi 0.9366 pontosság értéket 0.9425-re emelni, az AUC pedig változatlan maradt.

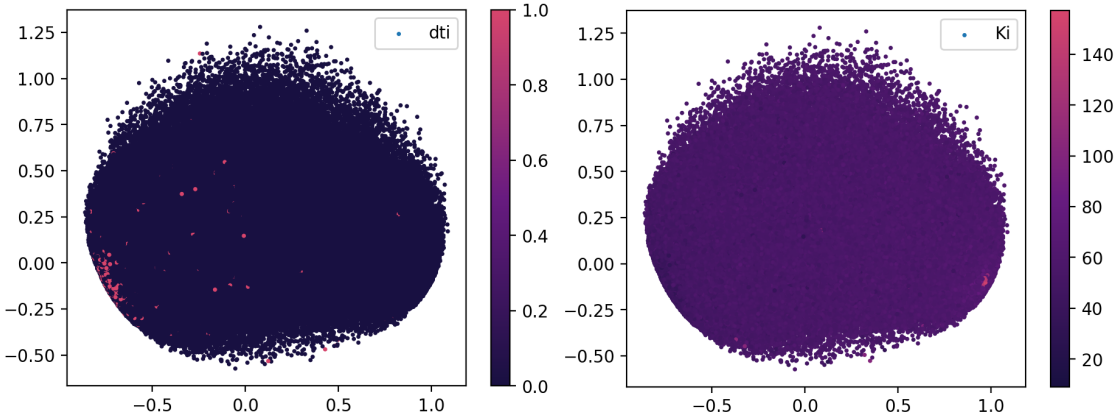
Elmondható, hogy ez a módszer a látens terembe épített interakcióbecslőnél jobb eredményeket ér el. Sőt, a 1.2. fejezetben említett state-of-the art becslők által elért

eredményeket is sikerült reprodukálnom, néhány esetben pedig felülmúlnom is. Konkrét összehasonlítást csak akkor adhatnék, ha én is implementáltam volna a többi módszert, és lemértem volna a teljesítményüket az én adataimon. Ugyanakkor a kitűzött feladatom nem a DTI becslés optimalizálása, csupán egy jól használható, gyorsan tanítható és jól teljesítő módszer keresése, az RMT algoritmus pedig optimális választás volt a feladat elvégzésére. Fontosnak tartom megjegyezni, hogy mivel a fehérjestruktúrát nem használja, így nem képes a módszer olyan célpontokra becsülni, amikhez nem ismert még egyetlen hatóanyag sem, ugyanakkor általánosítható a gyógyszerkutatás területén kívül is.

A becslő segítségével kiegészítettem a guacamol adathalmazt, a bináris DTI értéket a becslővel számítottam minden molekulára. A Ki értéknek az egyes molekulák, a becslő által számított altértől vett távolságát használtam, mivel a valódi Ki értékek nem álltak rendelkezésre, de szerettem volna egy folytonos DTI értéket is használni. Ezeken kívül a BindingDB-ben szereplő eredeti gyógyszerhatóanyagokra számolt hasonlóságot is bevettem a similarity score értékbe. Így már rendelkezésre állnak mind a BindingDB, mind a DTI becslő által szolgáltatott adatok a guacamol halmaz esetén is.

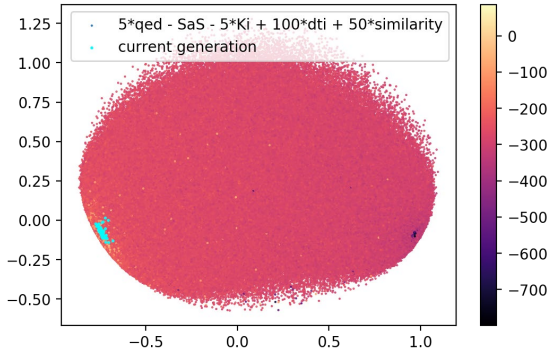
4.3.4. Generálás a JAK2 célpontra

Az így előállított adatok, és a korábban bemutatott célfüggvényre történő generálás segítségével sikerült megoldanom az egy célpontra történő generálás problémáját. A VAE modellt a módosított guacamol halmazon tanítottam, a kapott látens teret szemlélteti a 4.31. ábra.



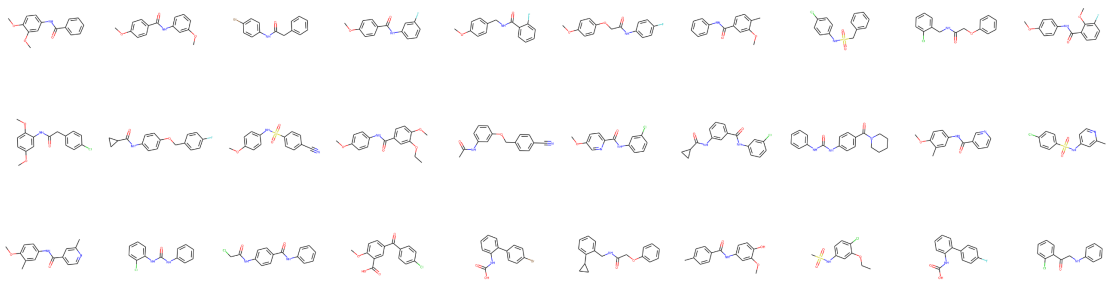
4.31. ábra: DTI adatokkal kiegészített guacamol adathalmaz látens tere, láthatóan egy helyre csoportosulnak a pozitív osztály elemei (dti), illetve a látens tér bal felén helyezkednek el a DTI becslő alteréhez közel lévő molekulák is (Ki).

Az optimalizálás során a genetikus algoritmus által használt fitnesz értéket a generált molekulákra a megszokott kémiai értékek mellett a DTI becslő által adott becslések adták. Az egyes generációk egyedei így a látens tér azon részén helyezkedtek el, ahol a pozitív osztályú molekulák is voltak, ezt mutatja a 4.32. ábra.



4.32. ábra: Az optimalizálási függvénynek köszönhetően az egyedek a látens tér megfelelő részére konvergáltak.

Az így generált több 100 molekula közül láthatjuk azt a 30 darabot a 4.33. ábrán, amikre a kiértékelt célfüggvény a legmagasabb pontot adta.



4.33. ábra: JAK2 célpontra generált molekulák. A célfüggvényben a célpontra való kötést mérő értékeken kívül továbbra is szerepeltek a gyógyszerszerűséget és a szintetizálhatóságot mérő tagok, ennek eredményeképp az így generált molekulák többségének szerkezete tartalmazza a kettős gyűrűt.

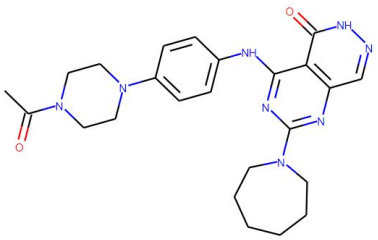
4.3.5. Generált molekulák vizsgálata

A generált molekulák többségét a becslő pozitív osztályba sorolta, vagyis a látens leképezés valóban tartalmazza a becslő által is kinyert információt. Ez önmagában még nem jelenti azt, hogy ezek a jelöltek tényleg hatnak az adott célpontra, lehet, hogy maga a becslő is rossz tulajdonságokat tanult meg.

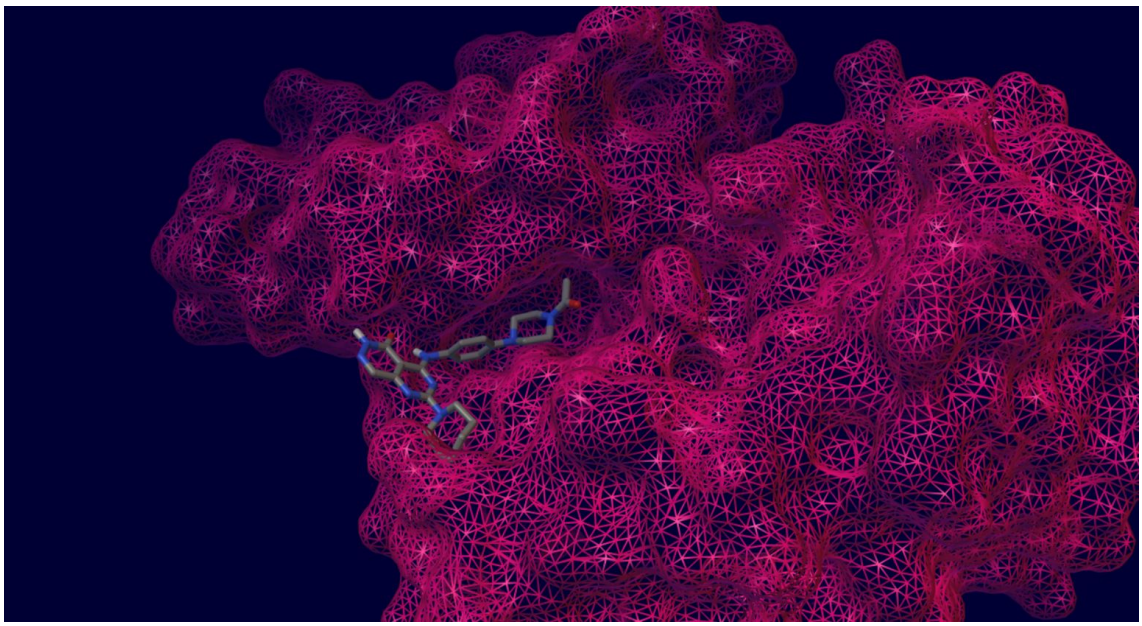
A modell generatív képességét könnyű lemérni a benchmark értékek segítségével, illetve az is tesztelhető, hogy a genetikus algoritmus valóban a tér megfelelő részét járja be, és valóban a kémiai értékeket tekintve megfelelő molekulákat generál. A célpontra generált jelöltek kiértékelése viszont nehéz feladat, a konkrét szintetizálásuk és laboratóriumi körülmények között elvégzett tesztek nélkül nehéz megállapítani a jóságukat. A legpontosabb rendelkezésemre álló metrika az „in silico” dokkolás által közölt eredmény.

Én az AutoDock VINA dokkoló programot választottam, ami megadja a legjobb kötési pozíciót, és a hozzá tartozó energia értéket, ami annál jobb minél kisebb. A BindingDB halmazban szereplő legkisebb Ki értékkel rendelkező ismert hatóanyag kötési energiája -7.4 kcal/mol volt, és a halmazban szereplő többi hatóanyag között sem találtam -7.7

kcal/mol értéknél jobbat elérő molekulát. A 4.33. ábrán bemutatott molekulák kötési energiája is -7.3 és -7.8 kcal/mol között mozog, ugyanakkor a célfüggvény többi tagja miatt azok könnyebben szintetizálhatóak. Ha a szintetizálhatóságra nem optimalizáltam, csak a kötésre, akkor ennél is alacsonyabb kötési energiát értem el, a legígéretesebb hatóanyag, amit a generált molekulák között találtam -9.1 kcal/mol energia értékkel rendelkezik a JAK2 célpontra. Ez a jelölt a többinél nehezebben szintetizálható ugyanakkor azoknál specifikusabb, vagyis azon kívül, hogy a célponthoz hatékonyabban köt, valószínűleg más fehérjékhez kevésbé társítható. A molekula szerkezetét mutatja a 4.34. ábra, a legkisebb energiához tartozó kötési pozíció pedig a 4.35. ábrán látható.



4.34. ábra: A legkisebb kötési energiával rendelkező generált molekula szerkezete, látszik, hogy generálásakor már nem optimalizáltam a szintetizálhatóságra, ezért megjelenhetett benne egy 7 hosszú, amúgy nehezen szintetizálható kör is.



4.35. ábra: A legkisebb kötési energiával rendelkező generált molekula pozíciója a JAK2 célpontra. Látható, hogy a zár és kulcs modellnek megfelelően pontosan illeszkedik a generált molekula a célfehérje egy specifikus kötési pontjához.

A fenti eredményeket értelmezve elmondható, hogy a molekulagenerálás és interakcióbecslés módszereket ötvözve sikerült megvalósítanom a harmadik kitűzött célomat is. Vagyis képes a modellem egy tetszőleges fehérjéhez kötő hatóanyagjelölteket generálni.

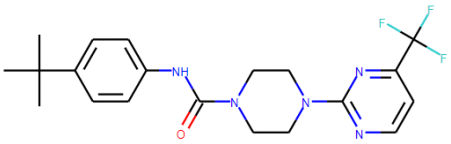
5. Példa a modellem lehetséges felhasználására

Láthattuk, hogy valóban képesek lehetnek a generatív modellek kiváltani, vagy legalábbis jelentősen felgyorsítani a gyógyszergyártás két legidőigényesebb lépését, a hatóanyagok közötti keresést, és azok tovább optimalizálást a célpontra. Azon kívül, hogy jelentős összegű befektetett pénz és idő spórolható meg használatukkal, segítségükkel hamarabb megállíthatjuk az új betegségek terjedését is. Példámban bemutatom, hogyan segíthetnek a SARS-CoV-2 vírus által terjesztett COVID-19 fertőzés elleni küzdelemben.

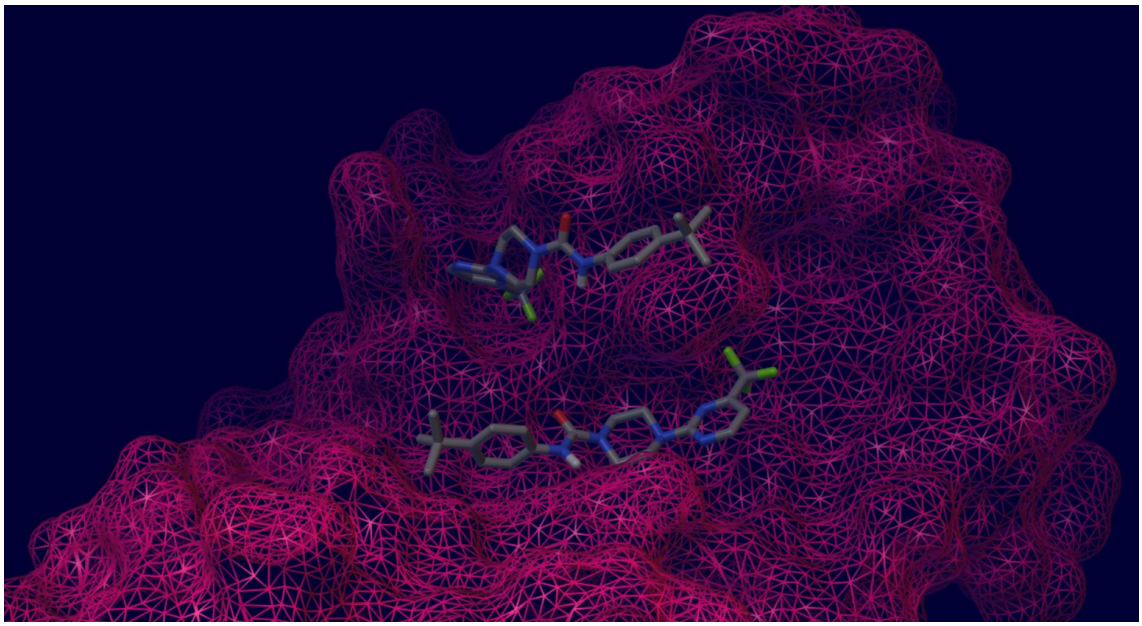
Amint említettem a gyógyszerkutatás első lépése egy lehetséges célpont keresése. A SARS-CoV-2 egy egyszálú RNS-vírus, vagyis terjedéséhez egy sejtet kell keresnie, amit RNS-ének bejuttatásával újra tud programozni. A vírus RNS-e több fehérjét is kódol, a nagyobb, összetett fehérjéket később a vírus enzim fehérjéi bontanak tovább alkotórészekre. Ezek a kisebb, funkcionális proteinek felelnek a vírus későbbi összeállításáért és terjesztéséért. A vírus által kódolt fehérjét lebontó enzimeknek tehát kulcsfontosságú szerepe van a vírus önreplikációs folyamatában, így alkalmasak lehetnek hatóanyagcélpontnak. A 16 funkcionális fehérjéből 13-mat a „3C-like protease", másnéven „main protease" (MPro) enzim állít elő. Az MPro 306 aminosavból áll, ennek megfelelően 918 nukleotid kódolja azt a vírus DNS-ében. A SARS-CoV-2 vírus ezen génje mutációt tekintve konzervatív [47], maga a vírus is lassabban mutálódik a többi vírushoz képest, és az MPro-ért felelős gén mutációs rátája pedig alacsonyabb a többinél. Én is megpróbáltam illeszteni a „Basic Local Alignment Search Tool" (BLAST) [48] által adott MPro referencia génjét az eddig szekvenált, több ezer példány teljes DNS-ére [49]. Valóban a legtöbb esetben 100%-os illeszkedést találtam, párszor fordult csak elő egypontos mutáció. Ezen felül az emberi testben nem található az MPro fehérjére hasonlító protein, így az arra fejlesztett inhibitorok kisebb eséllyel lesznek károsak a szervezetünkre nézve [47]. Én is próbáltam az MPro aminosav szekvenciáját az emberi fehérjékhez illeszteni a BLAST segítségével, de az nem adott szignifikáns találatot.

Az MPro fehérje tehát az egyik legígéretesebb célpontjelölt, aminek inhibitálásával megállíthatjuk a vírus terjedését, és már sikerült is feltérképezni a szerkezetét [50]. A célpont szerkezetének ismeretében már képesek vagyunk az egyes gyógyszerjelöltek jóságát tesztelni dokkolás segítségével, illetve már fejlesztettek ki az MPro-ra aktív hatóanyagokat [51], habár ezek mellékhatási még ismeretlenek.

Az ismert hatóanyagokból kiindulva én is generáltam új jelölteket a modellemmel. Először az interakcióbecslőt tanítottam a hatóanyagokon, majd ennek segítségével kiszámoltam a BindingDB adathalmaz Ki és DTI értékeit. Modellemet ezen a kiegészített adathalmazon tanítottam, és a genetikus algoritmus segítségével kerestem a látens térben megfelelő vektorokat, amikből aztán a dekóder segítségével molekulákat generáltam. Az így kapott jelölteket dokkolással teszteltem, sikerült a közölt molekulákhoz hasonló kötési energiaszintet elérnem, ez az energia -8.6 kcal/mol volt. A legalacsonyabb energiát elérő molekulát az 5.1. ábra szemlélteti, a hozzá tartozó kötési pozíciókból kettőt pedig az 5.2. ábra mutat.



5.1. ábra: Egy generált MPro inhibitor jelölt.



5.2. ábra: Az MPro fehérje és az arra generált inhibitor lehetséges pozíciói közül kettő látható. Az alsóhoz tartozik a legalacsonyabb kötési energiaszint, ami -8.6 kcal/mol.

Láthattuk, hogy a gyógyszerkutatás legidőigényesebb lépéseit gépi tanuláson alapuló módszerekkel leváltva valóban sikerülhet a hagyományos módszereknél akár évekkel gyorsabban hatóanyagot találni a vírusra. Viszont még így is várni kell a további lépések elvégzésére. Az egyik legfontosabb a mellékhatások keresése, ugyanis a közölt jelöltekre ez még ismeretlen. Ez a lépés is gyorsabb lehet a vártnál, hiszen az emberi testben nem található az MPro enzimre jelentősen hasonló protein, így az MPro-ra aktív hatóanyagok remélhetőleg kisebb valószínűséggel kötnek más fehérjékhez. A mellékhatások felkutatása és további optimalizálás után továbbra is szükséges a hatóanyagjelölt teljeskörű tesztelése a forgalomba hozatal előtt, ami reményeink szerint a lehető leghamarabb bekövetkezik.

6. Konklúzió, jövőbeli tervek

Az elért eredmények tekintetében kijelenthetem, hogy sikerült a kitűzött céljaimat teljesíteni. Implementáltam egy újszerű látens teret használó variációs autoenkódert, ami képes kellően nagy, akár milliós nagyságrendű adathalmazokat is kezelni, illetve általános generatív képessége a state-of-the art módszerekéhez hasonló. A modell képes a látens terét tetszőleges tulajdonságok mentén rendezni, majd a tulajdonságokból alkotott célfüggvény figyelembevételével, a látens tér megfelelő részét bejárva hatóanyag jelölteket generálni. A célfüggvény alakításával és egy hatóanyag-célpont interakciót becslő módszer segítségével sikerült kezelnem azt az esetet is, amikor kis számú molekulához kell valamilyen szempont szerint hasonlókat generálni. A modell ennek a képességének köszönhetően képes egy adott célpontra már ismert hatóanyagok ismeretében azoktól eltérő, új gyógyszerjelölteket előállítani. Mindezt sikerült a fehérjék aminosav szerkezetének ismerete nélkül megtennem, így a módszer könnyen alkalmazható más területeken is. A megoldásom így általánosítható az anyagtudomány területére, a gyógyszerhatóanyagokon kívül alkalmas lehet például újgenerációs napelemekhez szükséges, fotovolatikus tulajdonságokkal rendelkező organikus félvezetők generálására.

A jövőben tervezek több, a reprezentációt, a modellt, illetve magát az optimalizációt érintő továbbfejlesztési lehetőséget is megvizsgálni. A SMILES reprezentáció már említett hibái miatt megvizsgálnám a gráf alapú reprezentációkat. Szeretnék másfajta generatív architektúrát is kipróbálni, ilyen például a GAN, AAE, illetve az autoenkódereket GAN-okkal ötvöző megoldás [52], ami a GAN diszkriminátora által megtanult tulajdonságokat állítja elő a VAE dekóderével. Továbbá tervezem a genetikus algoritmus helyett a korlátozott bayesi optimalizációt [53] kipróbálni. Jövőbeli terveim között szerepel a DTI becslő továbbfejlesztése is, ugyanis eddig még nem figyeltem a lehetséges mellékhatásokra. Könnyen lehet, hogy a generált jelölt nem csak a célfehérjéhez, hanem más, ahhoz hasonló szerkeretű fehérjékhez is köthet. Szeretném a becslő ezen hasonló fehérjékre adott alterét is figyelembe venni a DTI információ előállításakor.

Zárásként elmondható, hogy a különböző mélytanulást alkalmazó módszereknek van létjogosultsága az orvosi gyógyszerkutatás terén is. Ezen az egy módszeren túlmenően még sok különböző mesterséges intelligenciára épülő megoldás is született eddig molekulák generálására. Az egy célpontra történő generálás kutatása pedig jelenleg is aktívan zajlik, a módszerek tökéletesítésének és ötvözésének köszönhetően az eddigieknél is nagyobb áttörések várhatóak.

Irodalomjegyzék

[1] Bayer (2020 október 27.). From molecules to medicie, https://www.bayer.com/en/media/molecules-medicine.

[2] Paul, S. M. et al. (2010). How to improve R&D productivity: the pharmaceutical industry's grand challenge. Nature reviews. Drug discovery 9, 203-214, doi:10.1038/nrd3078.

[3] Avorn, J. (2015). The $2.6 billion pill--methodologic and policy considerations. The New England journal of medicine 372, 1877-1879, doi:10.1056/NEJMp1500848.

[4] MarÉchal, E. (2011). Measuring Bioactivity: KI, IC50 and EC50. In Chemogenomics and Chemical Genetics (pp. 55-65). Springer, Berlin, Heidelberg.

[5] (2020 október 27.). Docking (molecular), https://en.wikipedia.org/wiki/Docking_(molecular).

[6] Kim, S., Thiessen, P. A., Bolton, E. E., Chen, J., Fu, G., Gindulyte, A., ... & Wang, J. (2016). PubChem substance and compound databases. Nucleic acids research, 44(D1), D1202-D1213.

[7] Polishchuk, P. G. (2013). Estimation of the size of drug-like chemical space based on GDB-17 data. Journal of computer-aided molecular design, 27(8), 675-679.

[8] Xu, Y., Lin, K., Wang, S., Wang, L., Cai, C., Song, C., ... & Pei, J. (2019). Deep learning for molecular generation. Future medicinal chemistry, 11(6), 567-597.

[9] Elton, D. C., Boukouvalas, Z., Fuge, M. D., & Chung, P. W. (2019). Deep learning for molecular design—a review of the state of the art. Molecular Systems Design & Engineering, 4(4), 828-849.

[10] Ertl, P., Lewis, R., Martin, E., & Polyakov, V. (2017). In silico generation of novel, drug-like chemical matter using the LSTM neural network. arXiv preprint arXiv:1712.07449.

[11] Ng, A. (2011). Sparse autoencoder. CS294A Lecture notes, 72(2011), 1-19.

[12] Doersch, C. (2016). Tutorial on variational autoencoders. arXiv preprint arXiv:1606.05908.

[13] Lim, J., Ryu, S., Kim, J. W., & Kim, W. Y. (2018). Molecular generative model based on conditional variational autoencoder for de novo molecular design. Journal of cheminformatics, 10(1), 1-9.

[14] Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., & Frey, B. (2015). Adversarial autoencoders. arXiv preprint arXiv:1511.05644.

[15] Kadurin, A., Nikolenko, S., Khrabrov, K., Aliper, A., & Zhavoronkov, A. (2017). druGAN: an advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico. Molecular pharmaceutics, 14(9), 3098-3104.

[16] Jin, W., Barzilay, R., & Jaakkola, T. (2018). Junction tree variational autoencoder for molecular graph generation. arXiv preprint arXiv:1802.04364.

[17] Zhou, Z., Kearnes, S., Li, L., Zare, R. N., & Riley, P. (2019). Optimization of molecules via deep reinforcement learning. Scientific reports, 9(1), 1-10.

[18] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680).

[19] Maziarka, Ł., Pocha, A., Kaczmarczyk, J., Rataj, K., Danel, T., & Warchoł, M. (2020). Mol-CycleGAN: a generative model for molecular optimization. Journal of Cheminformatics, 12(1), 1-18.

[20] Brown, N., Fiscato, M., Segler, M. H., & Vaucher, A. C. (2019). GuacaMol: benchmarking models for de novo molecular design. Journal of chemical information and modeling, 59(3), 1096-1108.

[21] Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. science, 313(5786), 504-507.

[22] Wang, Y., & Zeng, J. (2013). Predicting drug-target interactions using restricted Boltzmann machines. Bioinformatics, 29(13), i126-i134.

[23] Öztürk, H., Özgür, A., & Ozkirimli, E. (2018). DeepDTA: deep drug–target binding affinity prediction. Bioinformatics, 34(17), i821-i829.

[24] Bolgár, B., & Antal, P. (2017). VB-MK-LMF: fusion of drugs, targets and interactions using variational Bayesian multiple kernel logistic matrix factorization. BMC bioinformatics, 18(1), 440.

[25] Rifaioglu, A. S., Nalbat, E., Atalay, V., Martin, M. J., Cetin-Atalay, R., & Doğan, T. (2020). DEEPScreen: high performance drug–target interaction prediction with convolutional neural networks using 2-D structural compound representations. Chemical Science, 11(9), 2531-2557.

[26] Brenner, M. P., & Colwell, L. J. (2016). Predicting protein–ligand affinity with a random matrix framework. Proceedings of the National Academy of Sciences, 113(48), 13564-13569.

[27] Trott, O., & Olson, A. J. (2010). AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. Journal of computational chemistry, 31(2), 455-461.

[28] Chenthamarakshan, V., Das, P., Hoffman, S. C., Strobelt, H., Padhi, I., & Wai, K. (2020). CogMol: Target-specific and selective drug design for COVID-19 using deep generative models. arXiv: 2004.01215.

[29] (2020 október 27.). Simplified molecular-input line-entry system, https://en.wikipedia.org/wiki/Simplified_molecular-input_line-entry_system.

[30] Kusner, M. J.-L. (2017). Grammar variational autoencoder. In Proceedings of the 34th International Conference on Machine Learning-Volume 70 (pp. 1945-1954). JMLR.org.

[31] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.

[32] Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. IEEE transactions on Signal Processing, 45(11), 2673-2681.

[33] Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., ... & Aspuru-Guzik, A. (2018). Automatic chemical design using a data-driven continuous representation of molecules. ACS central science, 4(2), 268-276.

[34] Blum, L. C., & Reymond, J. L. (2009). 970 million druglike small molecules for virtual screening in the chemical universe database GDB-13. Journal of the American Chemical Society, 131(25), 8732-8733.

[35] Irwin, J. J., & Shoichet, B. K. (2005). ZINC− a free database of commercially available compounds for virtual screening. Journal of chemical information and modeling, 45(1), 177-182.

[36] Liu, T., Lin, Y., Wen, X., Jorissen, R. N., & Gilson, M. K. (2007). BindingDB: a web-accessible database of experimentally determined protein–ligand binding affinities. Nucleic acids research, 35(suppl_1), D198-D201.

[37] Bjerrum E. J. (2020 október 27.). Master your molecule generator, https://www.cheminformania.com/master-your-molecule-generator-seq2seq-rnn-models-with-smiles-in-keras/.

[38] Landrum, G. (2013). Rdkit documentation. Release, 1, 1-79

[39] Wold, S. E. (1987). Principal component analysis. Chemometrics and intelligent laboratory systems, 2(1-3), 37-52.

[40] Preuer, K., Renz, P., Unterthiner, T., Hochreiter, S., & Klambauer, G. (2018). Fréchet ChemNet distance: a metric for generative models for molecules in drug discovery. Journal of chemical information and modeling, 58(9), 1736-1741

[41] Weiss, G., Goldberg, Y., & Yahav, E. (2018). On the practical computational power of finite precision RNNs for language recognition. arXiv preprint arXiv:1805.04908.

[42] White, T. (2016). Sampling generative networks. arXiv preprint arXiv:1609.04468.

[43] Davidson, T. R., Falorsi, L., De Cao, N., Kipf, T., & Tomczak, J. M. (2018). Hyperspherical variational auto-encoders. arXiv preprint arXiv:1804.00891

[44] Yan, C., Wang, S., Yang, J., Xu, T., & Huang, J. (2019). Re-balancing Variational Autoencoder Loss for Molecule Sequence Generation. arXiv preprint arXiv:1910.00698

[45] Alperstein, Z., Cherkasov, A., & Rolfe, J. T. (2019). All smiles variational autoencoder. arXiv preprint arXiv:1905.13343.

[46] Whitley, D. (1994). A genetic algorithm tutorial. Statistics and computing, 4(2), 65-85.

[47] Wang, R., Hozumi, Y., Yin, C., & Wei, G. W. (2020). Decoding SARS-CoV-2 transmission, evolution and ramification on COVID-19 diagnosis, vaccine, and medicine. arXiv preprint arXiv:2004.14114.

[48] Johnson, M., Zaretskaya, I., Raytselis, Y., Merezhuk, Y., McGinnis, S., & Madden, T. L. (2008). NCBI BLAST: a better web interface. Nucleic acids research, 36(suppl_2), W5-W9.

[49] National Center for Biotechnology Information (2020 október 27.). Coronavirus genomes – NCBI Datasets, https://www.ncbi.nlm.nih.gov/datasets/coronavirus/genomes/.

[50] Douangamath, A., Fearon, D., Gehrtz, P., Krojer, T., Lukacik, P., Owen, C. D., ... & Carberry, A. (2020). Crystallographic and electrophilic fragment screening of the SARS-CoV-2 main protease. bioRxiv.

[51] PostEra (2020 október 27.). MPro Activity Data, https://covid.postera.ai/covid/activity_data.

[52] Larsen, A. B. L., Sønderby, S. K., Larochelle, H., & Winther, O. (2016, June). Autoencoding beyond pixels using a learned similarity metric. In International conference on machine learning (pp. 1558-1566). PMLR.

[53] Griffiths, R. R.-L. (2017). Constrained bayesian optimization for automatic chemical design. arXiv preprint arXiv:1709.05501.