Fehérjék kristályosodási hőmérsékletének megjóslása gépi tanulási eszközökkel

Fischer Kornél

Az önálló projekt első félévére választott feladatom fehérjék tulajdonságainak gépi tanulási modellekkel történő megjóslása volt. Ez a feladat alapját képezi számos bioinformatikai megoldásnak, a humán genom értelmezésétől, genetikailag módosított baktériumok gyógyászati felhasználásáig. A téma aktualitását mutatja, hogy 2020 novemberében a Google's DeepMind bemutatta AlphaFold nevű gépi tanulásra épülő algoritmusát, amivel az aminosavak sorrendjéből képes a fehérjék térstruktúráját elég nagy pontossággal meghatározni [1]. A modell sikeresen jósolta meg a SARS-CoV-2 fehérjéit, segítve az ellenanyagok kifejlesztésére tett erőfeszítéseket.

Első feladatként fehérjék kristályosodási hőmérsékletének megjóslásán dolgoztam [2]. A kristályosítás során a fehérjéket vizes környezetben oldják, amíg elérik a túltelítettség állapotát. Amint ez megtörtént, elkezdődik a fehérje kicsapódása, amely állapotban (pl. röntgenkrisztallográfiai eszközökkel) tanulmányozni tudják a proteinek térbeli struktúráját.

Az első lépés az input formájának meghatározása volt. A fehérjék elsődleges szerkezete az aminosavak kapcsolódásának sorrendje. 20 darab aminosav van, ami nagyon gyakran szerepel, és 2 további, amik nagyon ritkán fordulnak elő élő szervezetben. A fehérjék nagysága alatt most a kapcsolódó aminosavak sorozatának hosszát értjük. Ha olyan modellt választunk, ami fix méretű bemenetet vár, egységesíteni és az algoritmusok futási ideje miatt akár korlátozni kellhet a bemenetek méretét. Általában is az egyik központi feladat a fehérjéket jól reprezentáló input megtalálása.

Az inputként érkező karaktersorozatot vektorokká szeretnénk alakítani a feldolgozáshoz. Első megközelítésben egy egyszerű szótárat alkalmazhatunk, ahol minden aminosavat (amelyek egy-egy betűnek felelnek meg) megfeleltethetünk egy 1 és 20 közötti számnak. Ennek a megoldásnak a problémája, hogy numerikus értékként sorrendet definiál az aminosavak között, ami az adatfeldolgozás során torzíthatja az eredményeket. Ennél jobb megoldás az úgynevezett one-hot encoding. Legyen a betűkészletünk 21 darabos úgy, hogy a 2 ritka aminosavnak 1 elemet osztunk ki. Definiálunk egy 21 hosszú karakterisztikus vektort. Ha a beérkező aminosav a 3. helyen lévő betűnek felel meg, ott a vektorunk 1-es lesz, a többi helyen nulla. Így minden betűt egy 21 dimenziós vektorral kódolunk. Elkészítjük a mátrixot, ahol az oszlopok az előbb definiált vektorok, vagyis az i. oszlop a protein i. aminosavának karakterisztikus vektora. A továbbiakban legyen ez 21xN-es méretű ritka mátrix a fehérje reprezentánsa. Az N az aminosavak száma, pár tíztől több százig terjed. A korábban említett méret egységesítés miatt az első modellekhez minden fehérjének csak az első száz aminosavát használtuk. Ha a protein nincs száz hosszú, kiegészítettük (padding) a szekvenciáját a 21. karakterrel. Így minden fehérjét egy 21x100 mátrixszal írtunk le.

Az így kapott mátrixok egyrészt nagyok, másrészt pedig ritkák. Ezért használhatjuk a Singular Value Decomposition (SVD) módszert [3]. Ezzel a mátrix méretét csökkentjük,

valamint sűrűbben is lesznek a nem nulla értékek a mátrixban. Tömören összefoglalva, ha a mátrixunkat beszorozzunk a transzponáltjával, szimmetrikus mátrixot kapunk. Lesz rangnyi valós sajátértéke, és rangnyi darab lineárisan független páronként ortogonális sajátvektora, ezek bázist alkotnak a rangnyi dimenziós R tér fölött. Az A mátrix szinguláris értékei az $A^T A$ mátrix sajátértékei. Az SVD során az A mátrixot felbontjuk 3 mátrix szorzatává. Az egyik mátrix az $A^T A$ mátrix sajátvektoraiból áll, a második egy diagonális mátrix, aminek elemei a $A^T A$ mátrix sajátértékei. A harmadik mátrixban pedig az $A^T A$ mátrix sajátvektorainak az A-val vett szorzatuk szerepel, normalizálva. Ez a formalizmus elegendő, hogy használhassuk ezt a módszert.

Az input előkészítése után két modellt próbáltam ki, az első a Random Forest [4]. Ebben a módszerben a klasszifikátor döntési fák hibrid modellje. A módszerre jellemző, hogy a fák építését felgyorsítja, hogy a lehető legtöbbet használja. Emiatt nincs a döntési fáknál alkalmazott pruning. További cél, hogy a fák egymástól minél függetlenebbek legyenek. Ennek egyik eszköze a véletlen mintavétel a bemenetből. A másik fontos ötlet, hogy egy-egy fát nem a teljes változókészlet felhasználásával építünk, hanem ott is véletlen választunk pár változót, amivel dolgozunk. Emiatt a fák építése is felgyorsul, és eléggé el is fognak térni egymástól, ez adja az erejét ennek a modellnek.

A második kipróbált módszer a scikit-learn-ben megtalálható XGBoost volt [5]. Ez is döntési fák együttesét használja, de itt minden lépésben a következő fa függ a korábbi fázis kiértékelésétől. Ezt az eljárást nevezik boostolásnak. Az XGBoost esetében ez jól párhuzamosítható, mert a fák megépítésekor az ágakat párhuzamosan tanítják.

A következő félévben további adattípusokkal és modellekkel tervezem folytatni a vizsgálatot. Molekulák szerkezetének leírásához használható a SMILES (simplified molecular-input line-entry system), amely ASCII karakterekkel kódolja a molekulákat. Ez a fajta leírás sok hasonlóságot mutat a természetes nyelvi struktúrákkal, így kipróbálandók lesznek a természetes nyelvi modellezés neurális hálós modelljei, köztük a transformer modellek (pl BERT). Más megközelítés, ha megpróbáljuk az aminosavakat képekkel reprezentálni. Ezekből a képekből generálhatjuk egy fehérje reprezentánsát, majd különböző képfeldolgozási módszerekkel elemezhetjük a kapott képet.

[1] AlphaFold: Using AI for scientific discovery, 15 Jan 2020
https://deepmind.com/blog/article/AlphaFold-Using-AI-for-scientific-discovery

[2] Structural Protein Sequences, https://www.kaggle.com/shahir/protein-data-set

[3] Understanding Singular Value Decomposition and its Application in Data Science, 9 Jan 2020, https://towardsdatascience.com/understanding-singular-value-decomposition-and-its-application-in-data-science-388a54be95d

[4] Ho, Tin Kam. Random decision forests. In: *Proceedings of 3rd international conference on document analysis and recognition*. IEEE, 1995. p. 278-282. https://web.archive.org/web/20160417030218/http://ect.bell-labs.com/who/tkh/publications/papers/odt.pdf

[5] Chen, Tianqi; Guestrin, Carlos. Xgboost: A scalable tree boosting system. In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016. p. 785-794., https://arxiv.org/abs/1603.02754