VILNIAUS UNIVERSITETAS

GYVYBĖS MOKSLŲ CENTRAS

Biochemijos magistro studijų programos II k. studentas

Irmantas ROKAITIS

Magistro baigiamasis darbas

GENERATYVINIAIS PRIEŠIŠKAIS TINKLAIS SUGENERUOTŲ BALTYMŲ SEKŲ ANALIZĖ

Darbo vadovas:

Prof. Rolandas MEŠKYS

VILNIUS, 2021

GENERATYVINIAIS PRIEŠIŠKAIS TINKLAIS SUGENERUOTŲ BALTYMŲ SEKŲ ANALIZĖ

Darbas atliktas:

Vilniaus universiteto Biochemijos instituto

Molekulinės mikrobiologijos ir biotechnologijos skyriuje

Studentas:

Irmantas ROKAITIS _____

Darbo vadovas:

Prof. Rolandas MEŠKYS _____

VILNIUS, 2021

Turinys

Santrumpos

BI – bendra informacija (angl. *mutual information*)

DNR – deoksiribonukleorūgštis

DSP – daugybinis sekų palyginys

EC – fermentų komisijos numeris (angl. *the enzyme commission number*)

GDT – globalaus atstumo testas (angl. *global distance test*)

GPT – generatyviniai priešiški tinklai

JSA – jaučio serumo albuminas

MAE – vidutinė standartinė paklaida (angl. *mean average error*)

MDH – malato dehidrogenazė

MM – mašininis mokymasis

Neff – efektyvių sekų skaičius (angl. *number of effective sequences*)

NW – Needleman-Wunsch porinis palyginys

PDB – baltymų duomenų bankas (angl. *protein data bank*)

PMM – paslėptieji Markovo modeliai

PSM – pozicinė svorių matrica

PSSM – pozicijai specifinė įverčių matrica (angl. *position specific scoring matrix*)

PSR – protėvių sekų rekonstrukcija (angl. *ancestral sequence recontruction*)

ĮVADAS

Baltymų inžinerija apibūdina procesus, kurių pagalba yra kuriami baltymai su pagerintu aktyvumu, stabilumu ar pakeistu substratiniu specifiškumu. Tokių baltymų kūrimas išlieka sudėtinga užduotimi dėl itin didelio galimų baltymų variantų kiekio ir vis dar prastai suprantamo ryšio tarp baltymo sekos ir jo savybių. Šiuo metu kylančias problemas yra įprasta spręsti dviem pagrindiniais būdais – kryptinga baltymų evoliucija ir racionaliu baltymų dizainu.

Kryptinga evoliucija, kartu su specifinėmis atrankos sistemomis, simuliuoja natūralią baltymų evoliuciją. Pagrindinis šio metodo privalumas yra galimybė įvertinti didelius atsitiktinių mutantų kiekius. Kryptingos evoliucijos pagalba sėkmingai atrenkami baltymai su didesniu aktyvumu, stabilumu ir kitomis savybėmis, tačiau visas procesas yra apsunkintas lokalių baltymų erdvės minimumų, dėl kurių optimalus baltymo variantas gali būti nepasiekiamas. Racionalus baltymų dizainas remiasi molekuliniu baltymų modeliavimu: pasitelkiant fizikinius ir empirinius dėsnius yra siekiama nustatyti sąryšius tarp baltymo sekos ir jo funkcijos. Abu šie metodai reikalauja didelių kompiuterinių ir laboratorinių resursų bei ilgų optimizavimo ir testavimo ciklų.

Mašininio mokymosi (MM) metodai remiasi automatiniu algoritmo mokymusi, šiam išmokstant sąryšius, esančius duotajame duomenų rinkinyje. Sėkmingam prižiūrimų MM metodų pritaikymui itin svarbūs dideli, anotuoti ir subalansuoti duomenų rinkiniai, tačiau didžioji prieinamų biologinių duomenų dalis išlieka neanotuota ir tiksliai funkciškai neįvertinta. Neanotuotų baltymų sekų įvairovę gali panaudoti generatyviniai MM modeliai. Šie modeliai gali išmokti funkcinius baltymų sekų ir struktūrų sąryšius, kylančius iš natūralių baltymų sekų įvairovės. Naudojant baltymų sekomis apmokytus generatyvinius modelius galima generuoti naujus baltymų variantus, pasižyminčius panašiomis savybėmis kaip ir duotasis mokymo rinkinys. Be to, panaudojant funkcinius baltymų įverčius generavimo procesas gali būti kontroliuojamas – tai leidžia tikslingai keisti baltymų savybes.

Baltymų sekų generavimas išlieka multidisciplinine problema, reikalaujančia inovatyvių sprendimų tiek biologijos, tiek ir informatikos srityse. Modelių kūrimui ir tobulinimui būtina gebėti įvertinti generuojamų baltymų sekų kokybę ir jų panašumą į mokymui skirtas sekas.

Šio darbo tikslas: įvertinti generatyviniais priešiškais tinklais sugeneruotų baltymų sekų kokybę bei įvairovę.

Darbo tikslui pasiekti iškelti šie uždaviniai:

1. Palyginti pirmos eilės statistikas tarp natūralių ir sugeneruotų baltymų sekų rinkinių.
2. Palyginti antros eilės statistikas tarp natūralių ir sugeneruotų baltymų sekų rinkinių.
3. Įvertinti sugeneruotų baltymų sekų įvairovę.
4. Nustatyti įvesties vektoriumi kontroliuojamas baltymų sekų savybes.
5. Įvertinti biologinį sugeneruotų baltymų sekų funkcionalumą.

1. LITERATŪROS APŽVALGA

1.1. Baltymų inžinerija

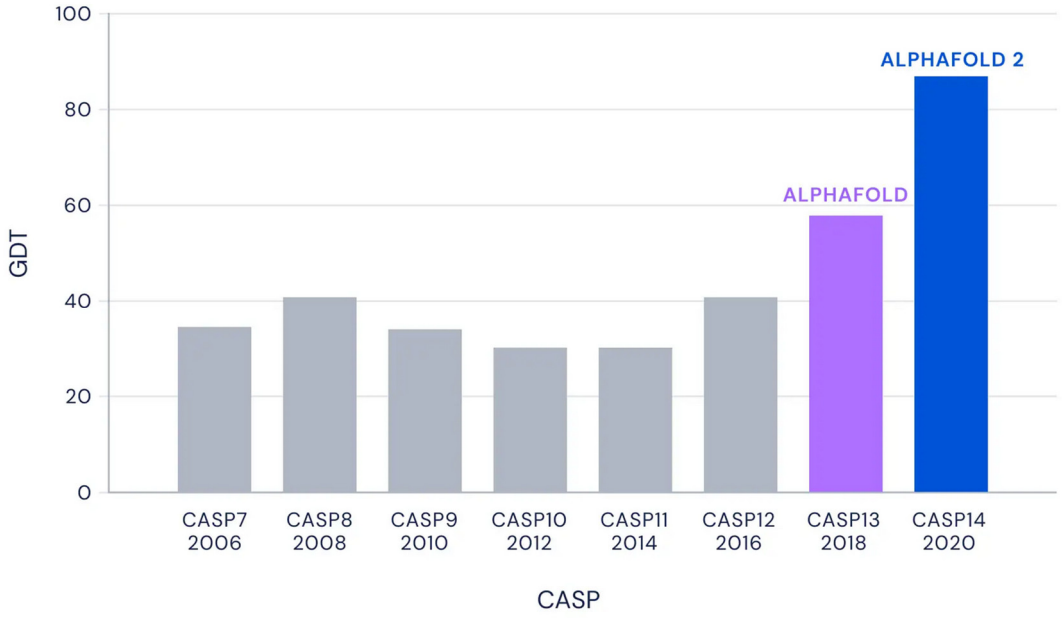
Baltymų inžinerija apibūdina procesus, kurių pagalba yra kuriami naujomis arba patobulintomis savybėmis pasižymintys baltymai (Brannigan ir Wilkinson, 2002). Nepaisant itin didelės galimų variantų įvairovės, ši mokslo sritis leido sukurti naujus ir optimizuotus baltymų variantus cheminėje ir farmacinėje biosintezėje, medicinoje, maisto gamyboje, atliekų perdirbime ir biojutiklių kūrime (Arnold, 2019; Bornscheuer ir kt., 2019; Qu ir kt., 2020; Sheldon ir Pereira, 2017). Baltymų inžinerijos pagalba kuriami fermentai įprastai yra tobulinami, siekiant pakeisti jų substratinį specifiškumą, patobulinti katalizinį aktyvumą, enantioselektyvumą, termodinaminį stabilumą, raišką, tirpumą ar stabilumą tirpikliuose (Poluri ir Gulati, 2017).

Šiuo metu plačiausiai naudojamos ir geriausiai apibūdintos baltymų inžinerijos strategijos yra racionalus dizainas ir kryptinga evoliucija (angl. *directed evolution*) (Arnold, 2015; Romero-Rivera ir kt., 2016). Racionalus dizainas apibūdina eilę metodų: nuo patirtimi ir žiniomis paremto hipotezių iškėlimo iki struktūrinės baltymų analizės ir kompiuterinių baltymų simuliacijų (Wilson, 2015). Baltymų ir jų naujų variantų modeliavimas padeda įvertinti fiziko-chemines aminorūgščių savybes ir simuliuoti jų sąveikas tiek su aplinka, tiek ir viena su kita. Naudojant kryptingąją evoliuciją siekiama simuliuoti ir pagreitinti natūralią baltymų evoliuciją. Tai yra iteratyvus procesas, kurio metu yra kuriamos mutantinių baltymų bibliotekos ir joms pritaikoma geresnėmis savybėmis pasižyminčio varianto atranka (Packer ir Liu, 2015). Abi šios strategijos yra sėkmingai pritaikomos praktikoje, tačiau jos reikalauja itin daug kompiuterinių ir/ar eksperimentinių lėšų kiekvienam specifiniam taikiniui optimizuoti (Steiner ir Schwab, 2012).

1.2. Mašininis mokymasis baltymų inžinerijoje

Mašininis mokymasis (MM) yra trečiasis būdas leidžiantis kurti naujus baltymų variantus. Skirtingai nuo racionalaus dizaino, ši strategija remiasi bendra biologinių duomenų gausa (Xu ir kt., 2020). MM padeda surasti duomenyse egzistuojančius sąryšius ir panaudoti juos nuspėjant naujų, modeliui dar nematytų, variantų savybes ar tiesiogiai kuriant naujus baltymų variantus. Augantis susidomėjimas MM yra stipriai susijęs su sėkmingu jo pritaikymu sprendžiant problemas, kurios seniau buvo neišsprendžiamos arba yra itin sudėtingos, pvz., natūralios kalbos apdorojimas, žmogaus rašto ir veidų atpažinimas, objektų klasifikavimas ir kitos (Guo ir kt., 2016; Young ir kt., 2018; Silver ir kt., 2017). Vienas iš garsiausių MM pasiekimų biologijoje – tai laimėjimai baltymų struktūrų modeliavimo konkurse CASP (angl. *Critical Assessment of protein Structure Prediction*). Pastaruosiuose dviejuose konkursuose (CASP13, 2018 ir CASP14, 2020)

MM paremti metodai pasiekė prieš tai dar nematyto prognozės tikslumo (**1.1 pav.**) (Callaway, 2020).



1.1 pav. MM paremtų modelių AlphaFold ir AlphaFold 2 rezultatai CASP konkursuose. Atvaizduoti vidutiniai globalaus atstumo testo (angl. *global distance test*, GDT) įverčiai. Pagal deepmind.com „AlphaFold: a solution to a 50-year-old grand challenge in biology“.

Įvairūs MM algoritmai yra pritaikomi baltymų inžinerijoje, pvz., atsitiktinių medžių rinkinys (angl. *random forest*) baltymų tirpumo nuspėjimui (Y. Yang ir kt., 2016), atraminių vektorių mašina (angl. *support vector machine*) baltymų stabilumo pokyčiui įvertinti (Teng ir kt., 2010), K-artimiausių kaimynų klasifikatoriai (angl. *K-nearest neighbor classifier*) – nuspėti baltymų funkcijas ir veikimo mechanizmus (De Ferrari ir Mitchell, 2014), klasterizavimo algoritmai – greitam sekų funkciniam anotavimui (Falda ir kt., 2012). Pagrindinis MM privalumas baltymų inžinerijoje yra jo gebėjimas apibendrinti duotąjį duomenų rinkinį: išmokius modelį žinomais duomenimis, MM gali atlikti spėjimus apie dar nematytus variantus, ir tai dažniausiai atlikti itin greitai. Tuo tarpu, racionaliam baltymų dizainui reikia sukonstruoti naują modelį, tai dažnai užtrunka savaites ar mėnesius ir reikalauja daug kompiuterinių resursų, o kryptinga baltymų evoliucija reikalauja ilgo laboratorinio eksperimentavimo ir atrankos sistemų kūrimo (Steiner ir Schwab, 2012).

MM modelio tikslumas ir jo panaudojimo galimybės priklauso tiek nuo mokymui naudotų duomenų kokybės, tiek nuo naudojamo algoritmo efektyvumo (Gao ir kt., 2020; Xu ir kt., 2020). Dideliu MM pritaikymo baltymų inžinerijoje iššūkiu šiuo metu yra plati baltymų struktūrų, fermentinių mechanizmų, reakcijų ir eksperimentinių sąlygų įvairovė. Be to, viešai prieinamiems duomenims dažnai trūksta sistemingos kokybės kontrolės, duomenų formato standartizavimo ar metaduomenų (Musil ir kt., 2019; Stourac ir kt., 2021). Nepaisant sąlyginai didelių viešai

prieinamų duomenų bazių, homogeniški duomenų rinkiniai vis dar išlieka sąlyginai maži, o naujų, kokybiškų duomenų surinkimas yra daug laiko ir lėšų reikalaujantis procesas.

Daugelio MM algoritmų esmė yra atrasti sąryšius pateiktuose duomenyse. Šie duomenys įprastai yra sudaryti iš duomenų taškų su jiems priskirtais požymiais ar aprašymais, pvz., fermentų sekos, jų antrinės, tretinės struktūros, aminorūgščių fiziko-cheminės savybės ir kita. Pagrindiniai MM tipai yra prižiūrimas mokymas (angl. *supervised learning*), neprižiūrimas mokymas (angl. *unsupervised learning*) ir dalinai prižiūrimas mokymas (angl. *semi-supervised learning*). Neprižiūrimo mokymo tikslas yra suspausti daug dimensijų turinčius duomenis į mažesnį dimensijų skaičių arba surasti duomenų klasterius. Šiam mokymo tipui naudojami duomenys neturintys juos atitinkančių žymenų. Prižiūrimame mokyme tinklas apmokomas nuspėti duotojo duomenų taško žymenį. Pavyzdžiui, tai gali būti fermento aktyvumo ar stabilumo nuspėjimas pagal pateiktą baltymo seką. Mokymas apjungiant tiek prižiūrimą, tiek ir neprižiūrimą mokymą yra vadinamas dalinai prižiūrimu mokymu (Alloghani ir kt., 2020).

1.2.1. Neprižiūrimas mokymas

Dėl jau minėto gerai anotuotų biologinių duomenų trūkumo, pastaraisiais metais populiarėja neprižiūrimo mokymo modeliai (Alley ir kt., 2019; Rao ir kt., 2020; Riesselman ir kt., 2018). Šį populiarumą lemia itin didelės sekų duomenų bazės, kurių dydis siekia iki 2,5 milijardo baltymų sekų (Steinegger, Mirdita, ir kt., 2019; Steinegger ir Söding, 2018). MM modeliai apmokyti naudojant neanotuotas baltymų sekas rodo itin gerus rezultatus baltymų kontaktų nuspėjime, homologinių sekų grupavime, mutacijų poveikio įvertinime ir kitais atvejais (Alley ir kt., 2019; Rao ir kt., 2020; Riesselman ir kt., 2018).

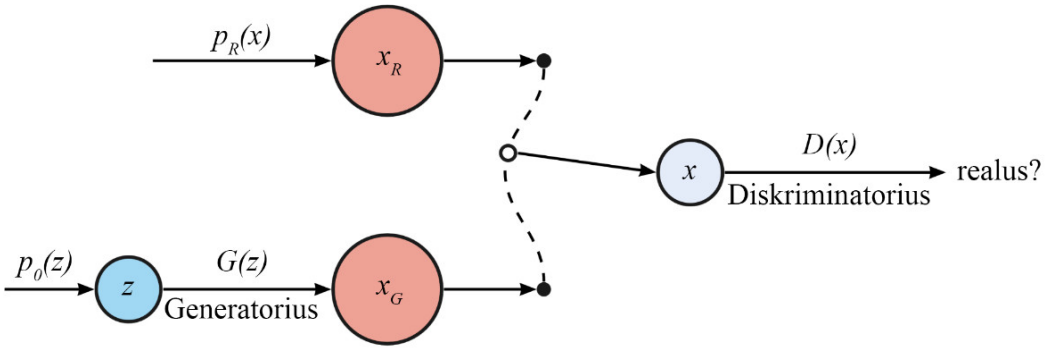
Panašūs rezultatai dažnai gali būti pasiekiami ir MM neparemtais įrankiais. Kontaktų nuspėjimas gali būti atliekamas įvertinus baltymo pozicijų tarpusavio kovariaciją (Dunn ir kt., 2008), mutacijų poveikio įvertinimas gali būti atliekamas pagal sekų profilį/pozicijos konservatyvumą (Hopf ir kt., 2017; Porebski ir Buckle, 2016), homologinių sekų paieška gali būti atliekama pagal paslėptojo Markovo modelio profilį ar sekų panašumą (Altschul ir kt., 1990; Remmert ir kt., 2012). Nepaisant to, daugelyje minėtų sričių geresni rezultatai yra pasiekiami su MM modeliais. Dažnai tai lemia kelios ypatybės:

- Statistika paremtiems įrankiams dažnai reikia sąlyginai didelio kiekio homologinių sekų, iš kurių turi būti sudaromas daugybinis sekų palyginys. Palyginio sudarymo metu gali būti padaromos klaidos, kurios tiesiogiai turi neigiamą poveikį galutiniams rezultatams. MM modeliai gali naudoti ir nesulygiuotus sekų rinkinius (Shin ir kt., 2021).

- MM neparemti įrankiai dažniausiai gali išnaudoti tik homologinių sekų informaciją. Tuo tarpu MM modeliai gali išnaudoti visų jam pateikiamų baltymų variantų informaciją (Alley ir kt., 2019).
- Paprastomis statistikomis paremti įrankiai dažniausiai atsižvelgia tik į specifines, jau iš anksto numatytas baltymų rinkinio savybes (pvz., pozicinis konservatyvumas, kovariacija). Priklausomai nuo modelio dydžio ir mokymo trukmės MM įrankiai gali išmokti įvairius duotajame duomenų rinkinyje esančius dėsningumus, neapribotus iš anksto numatytais kriterijais (Rives ir kt., 2021).

Neprižiūrimu mokymu apmokyti modeliai gali panaudoti itin didelius, neanotuotus duomenų rinkinius. Modelio mokymo metu yra siekiama, kad modelis išmoktų sąryšius, esančius duotajame mokymo rinkinyje, ir sugebėtų duomenis generalizuoti. Apmokyti modeliai gali būti pritaikomi spręsti įvairioms problemoms. Įprastai naudojamos modelio sudarytos baltymų reprezentacijos, kurios gali būti naudojamos tiesiogiai arba pernaudojamos prižiūrimame mokyme, taip specifiškai pritaikant modelį nustatytai užduočiai vykdyti (Biswas ir kt., 2021).

1.3. Generatyviniai priešiški tinklai



1.2 pav. Schema, apibūdinanti generatyvinius priešiškus tinklus. Pritaikyta pagal Pérez-Enciso ir Zingaretti 2019.

Pirmą kartą generatyviniai priešiški tinklai (GPT) (angl. *generative adversarial networks*) aprašyti 2014 metais (Goodfellow ir kt., 2014). Kaip nurodo pavadinimas, tai yra generatyviniai modeliai mokomi priešiško proceso būdų. Šiuo atveju „priešiškumas“ kyla dėl tinklo mokymo, kuriame dalyvauja dvi besirungiančios pusės – generatorius ir diskriminatorius. Generatyviniai modeliai yra statistinių modelių grupė, kuri siekia išmokti duotojo duomenų rinkinio X su atitinkamais žymenimis Y apjungtą tikimybę $p(X, Y)$ (arba $p(X)$, jeigu nenaudojami žymenys). Tuo tarpu diskriminaciniai modeliai siekia išmokti priklausomą tikimybę $p(Y \mid X)$ (Bernardo ir kt., 2007).

Kaip minėta, GPT yra sudaryti iš dviejų modulių: generatoriaus ir diskriminatoriaus (**1.2 pav.**). GPT apmokymui naudojamų duomenų rinkinį sudaro realūs pavyzdžiai *x*. Realių duomenų erdvę atitinka fiksuotas skirstinys p_R. Jeigu skirstinys $p_R(x)$ yra žinomas, visi realūs duomenų rinkinio pavyzdžiai x_R gali būti atkurti atsitiktinai imant bandinius iš skirstinio. GPT generatoriaus tikslas yra išmokti šį skirstinį, o atitinkamai generatoriaus išmoktas skirstinys yra vadinamas $p_G(x)$.

GPT mokymo metu diskriminatoriui yra pateikiami x_G ir x_R pavyzdžiai, kurie kyla atitinkamai iš sugeneruoto $p_G(x)$ ir realaus $p_R(x)$ skirstinių. Diskriminatoriaus tikslas šiuo atveju yra atskirti iš kurio skirstinio kilo duotasis pavyzdys *x*. Tuo tarpu generatorius siekia apgauti diskriminatorių: jo tikslas yra sugeneruoti skirstinį $p_G(x)$, kuris būtų kiek įmanoma panašesnis į realų skirstinį $p_R(x)$. Pavyzdžių generavimui generatorius naudoja neuroninius tinklus ir jo parametrus, kurių pagalba atsitiktinį įvesties skirstinį $p_0(z)$ paverčia sugeneruotu pavyzdžiu x_G.

Generavimo uždavinys yra sprendžiamas kaip neprižiūrimo mokymo problema, tačiau GPT atveju jo mokymas vykdomas remiantis prižiūrimo mokymo principais. GPT mokymo metu, generatoriaus ir diskriminatoriaus parametrai (θ^g ir θ^d atitinkamai) yra nuolat atnaujinami. Po generatoriaus apmokymo ciklo, generatoriaus sugeneruoti pavyzdžiai yra pateikiami diskriminatoriui, tada, parametras θ^g yra atnaujinamas taip, kad būtų maksimizuota diskriminatoriaus išvesties paklaida. Tai reiškia, kad mokymo ciklo metu diskriminatoriaus parametrai yra atnaujinami taip, kad jis gebėtų geriau atskirti realius ir sugeneruotus variantus. Kartu atnaujinami ir generatoriaus parametrai, parametrų pokytis priklauso nuo to, kaip sėkmingai diskriminatoriui pavyko atskirti realius ir sugeneruotus variantus.

Pagal žaidimų teoriją, generatoriaus ir diskriminatoriaus mokymas gali būti apibūdintas kaip nulinės sumos žaidimas. Tokio žaidimo principas teigia, kad privalo būti laiminti ir pralaiminti pusės (Binmore, 2007). Šiuo atveju tai reiškia, kad kai diskriminatorius sėkmingai atskiria realius ir sugeneruotus pavyzdžius jo parametrai nėra atnaujinami, tuo tarpu generatorius yra nubaudžiamas keičiant modelio parametrus. Priešingu atveju – generatoriui apgavus diskriminatorių, generatoriaus parametrai nėra keičiami, o diskriminatoriaus parametrai – modifikuojami.

Dėl aptartos GPT tinklų sandaros tokių tinklų kūrimas ir modifikavimas reikalauja, kad abi tinklo pusės – generatorius ir diskriminatorius būtų pakankamai efektyvios. Jeigu itin gerai veikia tik viena tinklo pusė, tinklo mokymasis nebevyksta, todėl tinklo kūrimo ir mokymo metu būtina užtikrinti ir stebėti tolygų abiejų modelių tobulėjimą (Salimans ir kt., 2016).

1.3.1. Generatyvinių priešiškų tinklų latentinė erdvė

GPT generatorius kaip įvestį priima latentinės erdvės tašką ir šio taško „koordinates“ konvertuoja į išvestį. Pati latentinė erdvė prasmės neturi. Įprastai tai yra 128-dimensijų hipersfera kurioje kiekvienos dimensijos vertė yra atsitiktinai parenkama pagal normalųjį skirstinį. Vykstant GPT mokymui, generatorius išmoksta priskirti specifines išvestis ar jų grupes, specifinėms latentinės erdvės sritims (Ayoob, 2020).

1.3 pav. Tiesinė interpoliacija tarp dviejų erdvės taškų (kairėje ir dešinėje paveikslėlių pusėse). Tarp galutinių taškų sugeneruojami variantai pasižymi tarpinėmis abiejų variantų savybėmis. Pagal Abdal, Qin, ir Wonka 2019.

Apmokyto generatoriaus atžvilgiu latentinė erdvė turi specifinę struktūrą, ši struktūra gali būti naršoma ar joje ieškoma specifinių verčių. Įprastai atliekant atsitiktinį objektų generavimą ar tiesiog mokymą latentinėje erdvėje taškai yra pasirenkami atsitiktinai. Pasirenkant specifinius latentinės erdvės taškus galima generuoti jiems specifinius objektus, t.y. kiekvienas latentinės erdvės taškas turi jį atitinkantį objektą. Tarp dviejų latentinės erdvės taškų galima sukurti kelią jungiantį šiuos du taškus, tokio kelio suradimas vadinamas interpoliacija (Bojanowski ir kt., 2019). Pavyzdžiui, kai yra generuojami paveikslėliai ėjimas šia tiese leidžia generuoti paveikslėlius kurie rodo kitimą tarp dviejų galutinių taškų (**1.3 pav.)**.

1.3.2. GPT panaudojimas biologijoje

Goldsborough ir kt., 2017 pritaikė GPT ląstelių nuotraukoms analizuoti. GPT mokymo tikslas šiuo atveju buvo išmokti ląstelių nuotraukų reprezentacijas naujų nuotraukų generavimui, taip siekiant pritaikyti tinklą morfologiniam nuotraukų generavimui. Ląstelės turi paprastesnę ir lengviau geometriškai aprašomą struktūrą, tai palengvina GPT pritaikymą tokio tipo duomenims lyginant su įprastomis nuotraukomis.

Hong ir kt., 2020 pritaikė GPT aukšto našumo chromosomų konformacijos pagavimo (angl. *high-throughput chromosome conformation capture (Hi-C)*) rezultatų apdorojimui. GPT paremtas tinklas pavadintas DeepHiC pritaikytas Hi-C kontaktų žemėlapiams atkurti naudojant žemo padengimo sekoskaitos duomenis. Pritaikius DeepHiC tyrėjai gali pasiekti aukštą kontaktų atkūrimo raišką su 100 kartų mažesniu sekoskaitos padengimu. Toks uždavinys atitinka aukštos raiškos paveikslėlių generavimą iš žemos raiškos paveikslėlių.

Subramaniya ir kt., 2020 panaudojo GPT baltymų kontaktų žemėlapių taisymui. Baltymų kontaktų žemėlapiai dažnai naudojami baltymų modeliavime. Jie yra nuspėjami pagal taikinio sekos daugybinį sekų palyginį. Nuspėtų žemėlapių taisymui šiuo atveju buvo naudojamas GPT. Kontaktų žemėlapiai patikslinti GPT buvo tikslesni nuo 1 % iki 50 % priklausomai nuo kontaktų žemėlapio kilmės.

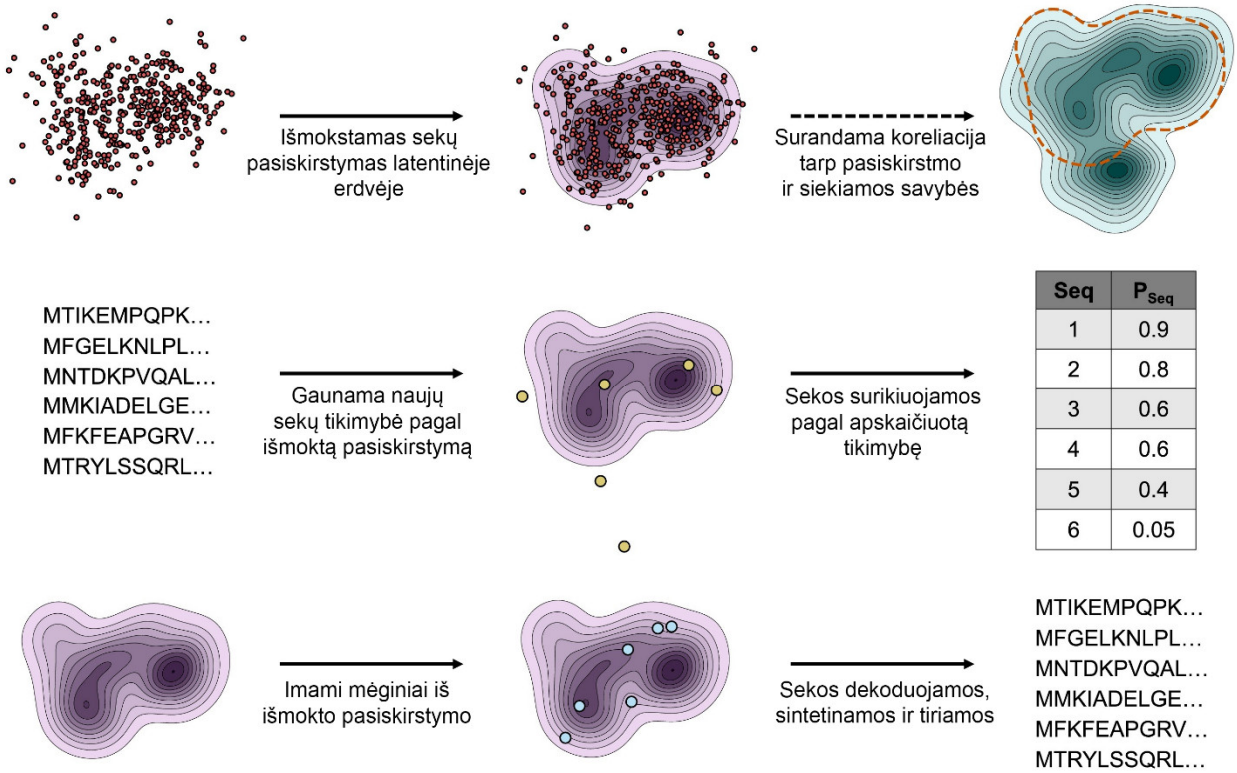
1.4. Sugeneruotų baltymų variantų panaudojimas

Dėl riboto kiekio variantų, kuriuos galima patikrinti kryptinga baltymų evoliucija, šis metodas dažniausiai apsiriboja pavienių pozicijų optimizacija. Tokia optimizacija įprastai vadinama godžia vieno žingsnio optimizacija (angl. *single-step greedy optimization*) ir yra atliekama etapais. Kiekvieno etapo metu nustatoma pagal atrankos sistemą labiausiai tinkantis aminorūgšties variantas pasirinktoje pozicijoje ir visos kitos pozicijos yra laikomos fiksuotos (Fasan ir kt., 2019). Kadangi kiekvienos mutacijos poveikis priklauso nuo jau esamų aminorūgščių, tokios optimizacijos efektyvumas priklauso nuo pradinio optimizuojamo baltymo varianto ir nuo eilės tvarkos, kuria pasirenkamos pozicijos mutacijų įvedimui. Šis procesas yra jautrus lokaliems minimumams, kurie kryptingai evoliucijai gali būti neįveikiami priklausomai nuo pradinio sekos varianto (Kaznatcheev, 2019). Pilnai generatyvinis baltymų sekų sukūrimas gali leisti pereiti per lokalius minimumus. Sugeneravus pakankamai platų spektrą pirminių sekų variantų galima greičiau rasti globalų minimumą arba didesnį lokalų minimumą.

Klasifikaciniams mašininio mokymosi tinklams yra būdingi tie patys apribojimai kaip ir kryptingai evoliucijai: pasirinkus pradinį variantą reikia pasirinkti kryptį kuria išbandomos įvairios mutacijos. Dažniausiai tai atlieka tikimybiniai statistiniai modeliai (Biswas ir kt., 2021; Wittmann ir kt., 2020). Dėl itin plačios galimų baltymų variantų įvairovės, pilnas, visų galimų

baltymo mutantų įvertinimas tampa neįmanomas. Pavyzdžiui, pilnam, penkiuose pozicijose galimų mutantų patikrinimui reikia įvertinti virš milijono variantų. Pilnas tokios įvairovės įvertinimas yra nebeįmanomas augant analizuojamų pozicijų skaičiui, netgi turint itin greitus modelius variantų vertinimui. Sekų generavimas leidžia praleisti pavienių variantų tikrinimą, o sugeneruoti variantai gali pasižymėti didesne įvairove ir didesniu aktyvių/neaktyvių fermentų variantų santykiu (**1.4 pav.**) (Russ ir kt., 2020).

Generatyvinių tinklų latentinė erdvė suteikia įrankį generuoti variantus su iš anksto pasirinktomis savybėmis. Tam yra reikalingas atitinkamų krypčių suradimas latentinėje erdvėje ir tinkamas modelio apmokymas. Tai leistų visiškai atsisakyti klasifikacinių modelių, tiesiogiai generuojant pasirinktų savybių variantus.



1.4 pav. Generatyvinių tinklų panaudojimas sekų generavimui ir savybių nuspėjimui. A, Generayviniai modeliai išmoksta mokymui duotųjų baltymų sekų pasiskirstymo reprezentaciją. Šis pasiskirstymas gali atitikti įvairias baltymų savybes (aktyvumas, stabilumas, tirpumas ir kt.). B, Generatyvinius modelius naudojant savybių nuspėjimui yra laikoma, kad išmoktas pasiskirstymas atitinka siekiamą pakeisti savybę. Naudojant modelį tokiu principu, sekoms galima priskirti tikimybinius atitikimo įverčius pagal kuriuos galima atrinkti sekų variantus su galimai geresnėmis savybėmis. C, generatyvinius modelius naudojant sekų generavimui naujos sekos yra imamos iš išmokto sekų pasiskirstymo. Pritaikyta pagal Wittmann ir kt. 2021.

1.5. Sekų analizė

1.5.1. Porinis sekų sulygiavimas

Porinis baltymų sekų palyginimas apibrėžia išskirtinai dviejų baltymų sekų sulygiavimą. Dažniausiai išskiriami dviejų tipų poriniai baltymų sekų palyginiai: globalūs ir lokalūs. Globalūs palyginiai yra paremti Needleman-Wunsch (NW) algoritmu ir, kaip nurodo pavadinimas, siekia sulygiuoti sekas per pilną sekos ilgį (Needleman ir Wunsch, 1970). Lokalūs palyginiai remiasi Smith-Waterman algoritmu ir siekia dvi sekas sulygiuoti tik ties panašiausiais šių sekų regionais (Smith ir Waterman, 1981).

Algoritmo pasirinkimas įprastai priklauso nuo sekų kilmės bei porinio palyginio sudarymo tikslo. Norint palyginti panašaus ilgio homologinės kilmės sekas, dažniau pasirenkamas globalus palyginys. Palyginį sudarant tarp evoliuciškai nutolusių ar nepilnų sekų dažniausiai pasirenkmas lokalus palyginys. Sudarius palyginį galima tiesiogiai įvertinti sekų panašumą. Panašumas įprastai gali būti apibrėžiamas dviem būdais: kaip identiškas panašumas (angl. *identity*) arba panašumas (angl. *similarity*). Identiškas panašumas įprastai nurodo visiškai sutampančių aminorūgščių skaičiaus santykį su palyginio ilgiu. Panašumas kartu įvertina ir dviejų neidentiškų aminorūgščių fiziko-cheminį ar statistinį panašumą. Toliau šiame darbe sekų panašumu bus vadinamas identiškas panašumas.

1.5.2. Daugybinis sekų palyginys

Daugybiniai sekų palyginiai (angl. *multiple sequence alignment*) apibrėžia pozicinį biologinių, evoliuciškai susijusių, sekų palyginį. Sudarant DSP atsižvelgiama į evoliucinį sekų kintamumą: mutacijas, insercijas ir delecijas. Palyginys atitinka stačiakampę baltymų sekų matricą. Siekiama, kad stulpelis šioje matricoje atitiktų:

- homologinę poziciją (kilusi iš vienos pozicijos pirminio protėvio sekoje);
- sulygiuojamą struktūrinę poziciją (apibrėžta pozicija lokaliame struktūrų palyginyje);
- vienodą funkcionalumą.

Artimiems sekų homologams visos šios sąlygos dažniausiai gali būti išpildomos kartu, tačiau labiau evoliuciškai nutolusiems baltymams struktūrinės, funkcinės ir sekos pozicijos gali išsiskirti. Dėl to skirtingi sulygiavimo kriterijai gali nulemti skirtingus sekų palyginius (Edgar ir Batzoglou, 2006).

DSP sudarymas gali būti pritaikytas sulygiuojant DNR, RNR ir baltymų sekas. Didelė dalis *in silico* atliekamų biologinių sekų analizių priklauso nuo DSP sudarymo, pvz., domenų

analizė, filogenetinė rekonstrukcija, motyvų paieška, kovariacinė analizė ir kita (Kemena ir Notredame, 2009; Thompson ir kt., 2011).

1.5.3. Pozicijai specifinės matricos

Pozicinė svorių matrica (PSM) yra supaprastinta daugybinio sekų palyginio reprezentacija, nurodanti aminorūgščių dažnį duotojoje pozicijoje (Ben-Gal ir kt., 2005). Baltymams šios matricos dydis atitinka $L \times aa$, kur L yra palyginio sekų ilgis, aa – aminorūgščių skaičius (atitinkantis 20 skaičiuojant tik standartines aminorūgštis).

Pozicijai specifinė įverčių matrica (angl. *position specific scoring matrix, PSSM*) yra sudaroma iš pozicinės dažnių matricos šiai pritaikius pseudo-įverčius (angl. *pseudocount*) ir logaritminį normalizavimą. Pseudo-įverčių pritaikymas užtikrina, kad nei viena PSSM matricos pozicija neturėtų tikimybės lygios nuliui, kuri taptų neigiama begalybe pritaikius logaritminį normalizavimą. Kartu tai padeda išspręsti ir DSP sudarymo šališkumą, kai ji sudaroma iš mažo skaičiaus sekų (Gribskov ir kt., 1987).

Pozicijai specifinės matricos dažniausiai yra naudojamos sekų paieškai atlikti ar sekų konservatyvumo įvertinimui. Be to, yra žinoma, kad pozicijai specifinės matricos gali būti naudojamos mutacijų poveikiui nuspėti (Hopf ir kt., 2017).

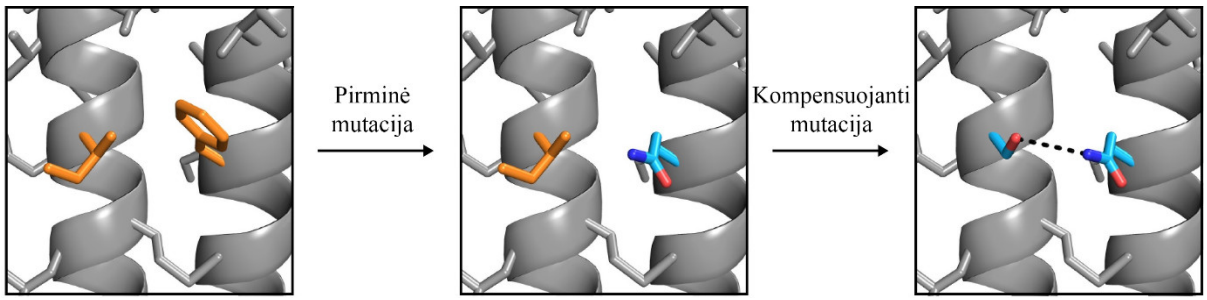
1.5.4. Baltymų konservatyvumas

Informacija apie baltymo funkcines pozicijas tiesiogiai padeda nustatant baltymo funkciją, prijungiamus ligandus, baltymas-baltymas sąveikas, planuojant eksperimentinę analizę ar tiriant molekulinius mechanizmus (Guharoy ir Chakrabarti, 2005; Kalinina ir kt., 2004; Liang ir kt., 2006). Vienas iš dažniausiai naudojamų baltymų funkcinių pozicijų įvertinimo būdų yra sekų analizė (Capra ir Singh, 2007). Kaip alternatyva jai gali būti naudojama struktūrinė analizė (Jones ir Thornton, 2004). Sekų analizės metu yra laikomasi prielaidos, kad homologinių baltymų daugybinio sekų palyginio stulpeliai nurodo funkciškai ar struktūriškai svarbias pozicijas tada, kai šiuose pozicijose yra stebimas mažas variabilumas. Maža aminorūgščių variacija yra siejama su neigiamu evoliuciniu spaudimu kylančiu dėl mutacijų atsiradusių šiuose pozicijose. Ši savybė įprastai vadinama konservatyvumu (Cooper ir Brown, 2008).

Vienas iš paprasčiausių ir lengviausiai interpretuojamų būdų statistiškai įvertinti pozicinį konservatyvumą yra Šanono entropija (Strait ir Dewey, 1996). Informacijos teorijoje Šanono entropija apibūdina kintamojo informacijos kiekį arba jo neapibrėžtumą, kuris priklauso nuo kintamojo reikšmių (Shannon, 1948). Šanono entropija yra lygi nuliui, jeigu DSP pozicijoje matoma tik viena aminorūgštis. Entropija įgauna didžiausią reikšmę, kai visos aminorūgštys pozicijoje pasirodo vienodu dažniu. Reikia paminėti, kad šiuo metu yra priimta naudoti kitus,

tikslesnius metodus funkcinėms pozicijoms įvertinti. Jie dažnai kartu naudoja ir pozicijos kaimynų informaciją ir/ar pakeitimų matricas (angl. *substitution matrix*) (Johansson ir Toh, 2010).

1.5.5. Baltymų kovariacija ir koevoliucija



1.5 pav. Baltymo pozicijų koevoliucija. Vienoje pozicijoje įvykusi mutacija yra kompensuojama greta esančioje pozicijoje įvykusios mutacijos. Pritaikyta pagal Nicoludis ir Gaudet 2018.

Baltymų funkcijai itin svarbi yra ne tik duotoji pozicija, bet ir jos aplinka. Šiuo atveju aplinka yra laikoma tiek pirminėje struktūroje (t.y. sekoje) arti esančios aminorūgštys, tiek ir antrinėse, tretinėse bei ketvirtinėse baltymo struktūrose esantys aminorūgščių kaimynai.

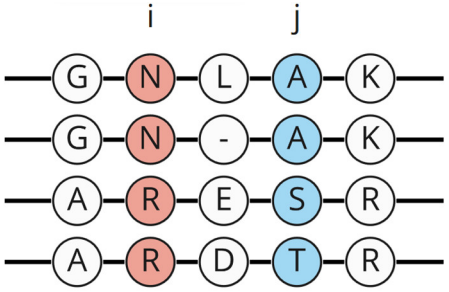
Siekiant iš baltymų sekų išgauti papildomos informacijos pastaraisiais dešimtmečiais itin intensyviai pradėta tyrinėti aminorūgščių koevoliuciją (de Juan ir kt., 2013). Tai kartu lėmė ir nuolat augantys prieinami kompiuteriniai resursai bei sparčiai didėjančios biologinių duomenų bazės (Suzek ir kt., 2007). Koevoliucijos tyrimai remiasi prielaida, kad dvi aminorūgštys, kurių mutacijos tarpusavyje koreliuoja, mutuoja taip, kad vienos aminorūgšties mutacija yra kompensuojama kitos aminorūgšties. Koevoliucija yra interpretuojama kaip funkcinė tarpusavio priklausomybė, t.y. jeigu dvi aminorūgštys koevoliucionuoja, atsiradus tik vienos aminorūgšties mutacijai, kartu atsiranda ir evoliucinis spaudimas įvykti kitos aminorūgšties mutacijai (**1.5 pav.**) (Pazos ir Valencia, 2008).

Koevoliucijai nustatyti dažniausiai tiriama baltymo DSP pozicijų kovariacija (Ashenberg ir Laub, 2013). Dabar jau standartu tapęs kovariacijos nustatymo tikslumo įvertinimas yra aminorūgščių kontaktų nuspėjimas remiantis apskaičiuota kovariacija. Įprastai laikoma, kad kontaktas tarp aminorūgščių yra tada, kai atstumas tarp jų C-α arba C-β atomų yra mažesnis nei 8 Å (Adhikari ir Cheng, 2016). Šis tikslumo įvertinimo būdas yra sąlyginai paprastas, tačiau jis turi trūkumų: koevoliucija (kartu ir kovariacija) gali atsirasti ne tik dėl kontaktų nustatomų galutinėje struktūroje, bet ir dėl aminorūgščių funkcinio bendrumo, tarpinių baltymo lankstymosi būsenų, baltymo dinamikos ar tarp oligomerizacijoje dalyvaujančių aminorūgščių, todėl metodai tiksliau iš kovariacijos nustatantys baltymo kontaktus, nebūtinai tiksliau nustato baltymo pozicijų kovariaciją (Anishchenko ir kt., 2017).

Pirminiai būdai įvertinti kovariaciją buvo paremti bendros informacijos (angl. *mutual information*) įverčiu (Korber ir kt., 1993). Galiausiai, apjungus eksponentiškai augantį baltymų sekų skaičių ir technikas, sprendžiančias atvirkštinę statistinę problemą, sukurta tiesioginio sujungimo analizė (angl. *direct coupling analysis*). Šis metodas sulaukė plataus pritaikymo ir buvo panaudotas sprendžiant baltymų struktūras, nuspėjant mutacijų poveikį bei nustatant baltymas-baltymas sąveikas (Hopf ir kt., 2017; Kamisetty ir kt., 2013; Morcos ir kt., 2011).

Šiuo metu yra priimta, kad aminorūgščių kovariacija nustatyta iš DSP gali gana tiksliai apibendrinti baltymų lankstymąsi bei funkcijas (Russ ir kt., 2005, 2020; Socolich ir kt., 2005). Nepaisant to, didžiausias dėmesys interpretuojant DCA rezultatus skiriamas stipriausiai pagal modelį kovarijuojančioms pozicijoms, nes šios gali būti tiesiogiai identifikuojamos kaip kontaktai struktūroje. Tačiau vien kontaktai negali atkurti nei DSP statistikų (Russ ir kt., 2020), nei įvertinti funkcinių pozicijų (Salinas ir Ranganathan, 2018). Baltymų funkcijos priklauso nuo daug, silpniau kovarijuojančių pozicijų, kurios vis dar neturi tikslios fizikinės interpretacijos. Dėl šių priežasčių, lyginant sekų rinkinius svarbu atsižvelgti ne tik į stipriausiai statistiškai kovarijuojančias pozicijas, bet ir į bendrą kovariacinį foną (Russ ir kt., 2020).

1.5.6. Pirmos eilės sekų statistikos



Poziciniai aminorūgščių dažniai	i = {N: 0,5 R: 0,5} j = {A: 0,5 S: 0,25 T: 0,25}	Pirmos eilės
Šanono entropija	i = 0.23 j = 0.35	
Poriniai aminorūgščių dažniai	ij = {NA: 0,5 RS: 0,25 RT: 0,25}	Antros eilės
Porinė kovariacija	ij = {NA: 0 RS: 0,125 RT: 0,125}	
Bendra informacija	ij = 1	

1.6 pav. Daugybinio sekų palyginio dalis ir jo pozicijų *i* ir *j* statistikos. Poziciniai aminorūgščių dažniai ir Šanono entropija priklauso tik nuo duotosios pozicijos ir jos aminorūgščių pasiskirstymo. Poriniai aminorūgščių dažniai, porinė kovariacija ir bendra informacija priklauso nuo abiejų pozicijų ir jų porinių aminorūgščių dažnių.

Pirmos eilės statistikos apibrėžia pozicinį aminorūgščių pasiskirstymą sekose ar jų DSP, nepriklausomai nuo kitų pozicijų (**1.6 pav.**).

Paprasčiausia pirmos eilės statistika apibūdina pozicinius aminorūgščių dažnius duotojoje pozicijoje. Iš esmės tai atitinka sekų rinkinių PSM palyginimą.

Kita pirmos eilės statistika yra Šanono entropija. Informacijos teorijoje Šanono entropija atitinka neapibrėžtumo ar atsitiktinumo įvertį (Shannon, 1948). Kaip minėta anksčiau, biologine prasme Šanono entropija atitinka pozicijos konservatyvumą. Duotosios palyginio pozicijos Šanono entropija įvertinama kaip:

$$H_i = -\sum_{\alpha=1}^{20} p(x_\propto) \log_{20} p(x_\alpha)$$

Čia, i – duotasis palyginio stulpelis, $p(x_\alpha)$ – aminorūgšties α dažnis pozicijoje. Įverčius skaičiuojant $\log_{20}$ skalėje visos reikšmės normalizuojamos rėžiuose nuo 0 iki 1. Šanono entropija neapibrėžia ir nevertina specifinių aminorūgščių, o priklauso tik nuo jų bendro dažnio sekoje (Gloor ir kt., 2005).

1.5.7. Antros eilės sekų statistikos

Antros eilės statistikos apibrėžia statistikas, priklausančias nuo dviejų pozicijų baltymo sekoje ar jų rinkinyje (**1.6 pav.**). Paprasčiausias antros eilės statistikos įvertis yra porinis aminorūgščių dažnis, kuris atspindi specifinių aminorūgščių dažnį tarp dviejų specifinių pozicijų sekoje (Johnson ir kt., 2021).

Poriniai aminorūgščių dažniai kartu nurodo aminorūgščių pozicinį dažnį ir jų porų tarpusavio priklausomybę. Šiuo atveju šių dviejų ypatybių atskirti neįmanoma. Siekiant iš porinių aminorūgščių dažnių įverčių nustatyti tik dviejų aminorūgščių tarpusavio kovariaciją, šie yra normalizuojami pagal porų pozicinius aminorūgščių dažnius. Šis įvertis yra vadinamas porine aminorūgščių kovariacija (McGee ir kt., 2021). Jis apibrėžiamas:

$$C_{\alpha\beta}^{ij} = f_{\alpha\beta}^{ij} - f_\alpha^i f_\beta^j$$

Čia, $f_{\alpha\beta}^{ij}$ – porinis aminorūgščių dažnis pozicijose *i* ir *j* tarp aminorūgščių α ir β; $f_\alpha^i f_\beta^j$ – pozicinių aminorūgščių dažnių sandauga atitinkamose pozicijose.

Siekiant įvertinti bendrą pozicijų poros, o ne specifinių aminorūgščių kovariaciją, galima naudoti bendros informacijos įvertį (Gloor ir kt., 2005). BI yra apskaičiuojama kaip:

$$BI_{ij} = H_i + H_j - H_{ij}$$

BI dydis priklauso nuo dviejų pozicijų kovariacijos ir nuo bendros pozicijų entropijos. H_{ij} – nurodo jungtinę pozicijų *i* ir *j* entropiją, kuri yra apskaičiuojama pagal visų šiuose pozicijose sudaromų aminorūgščių porų dažnį. Jungtinės entropijos vertės kinta nuo maksimalios H_i ar H_j vertės iki $H_i + H_j$. BI_{ij} vertės kinta nuo 0 iki minimalios H_i ar H_j vertės. Bendrąja prasme BI nurodo, kiek informacijos yra žinoma apie *j*, žinant pilną *i* skirstinį. Baltymų atveju BI nurodo mutacijų pozicijose *i* ir *j* susietumą, o tai tiesiogiai atitinka jau aptartą pozicijų kovariaciją (Gloor ir kt., 2005).

1.5.8. Sekų rinkinio normalizavimas

Baltymų sekos prieinamos viešose duomenų bazėse neatspindi realaus baltymų pasiskirstymo ir jų dažnio. Tai lemia išskirtinai intensyvus modelinių organizmų tyrinėjimas ir didesnis susidomėjimas specifinėmis organizmų populiacijomis ar jų augimo aplinkomis (Li ir kt., 2012). Neatsižvelgus į šį duomenų netolygumą, sunku priimti objektyvias išvadas galiojančias ir pritaikomas visai baltymų populiacijai.

Priklausomai nuo tyrimo ir uždavinio sekų normalizavimą galima atlikti pagal baltymų funkcijas ar jų struktūrines sanklodas. Tačiau universaliausias ir dažniausiai pritaikomas normalizavimo būdas yra normalizavimas sekas klasterizuojant pagal jų panašumą. Klasterizavimo tikslas yra surasti tokį reprezentatyvų sekų rinkinį, kad kiekviena duomenų rinkinio seka būtų pakankamai gerai reprezentuota vienos iš *K* reprezentatyvių sekų. „Pakankamai gerai“ yra nustatomas pagal pasirinktą sekų tarpusavio panašumo įvertinimo būdą (Steinegger ir Söding, 2018).

Sudarius klasterius normalizavimas įprastai atliekamas keliais būdais:

1. Iš kiekvieno klasterio išrenkama ir naudojama tik reprezentatyvi seka.
2. Klasterio viduje kiekvienam klasterio nariui priskiriamas svoris. Paprasčiausiu atveju jis atitinka atvirkštinį klasterio dydį.
3. Iš kiekvieno klasterio vienodu dažniu imami ėminiai.

Pirmasis normalizavimo būdas yra paprastas ir greitas, be to, leidžia reikšmingai sumažinti duomenų rinkinio dydį. Antrasis būdas išlaiko visas duomenų rinkinio sekas, kurios gali būti panaudojamos kaip papildoma informacija, tačiau tokio duomenų rinkinio panaudojimas ir paruošimas yra sudėtingesnis. Trečiasis būdas atitinka tarpinį pirmų dviejų būdų variantą.

1.5.9. Sekų rinkinio įvairovės įvertinimas

Sekų klasterizavimas taip pat naudojamas sekų rinkinio įvairovei įvertinti. Vien tik sekų skaičius gali nebūtinai tiksliai atspindėti duomenų rinkinio įvairovę, kadangi netgi didelio sekų rinkinio atstovai gali būti itin artimi homologai. Įvertinti sekų rinkinio dydį naudojamas efektyvių sekų skaičius (angl. *number of effective sequences, Neff/Nf*) (Zhang ir kt., 2020). Jis įvertinamas kaip:

$$Neff = \frac{1}{\sqrt{L}} \sum_{n=1}^{N} \frac{1}{1 + \sum_{m=1, m \neq n}^{N} I[S_{m,n} \geq 0{,}8]}$$

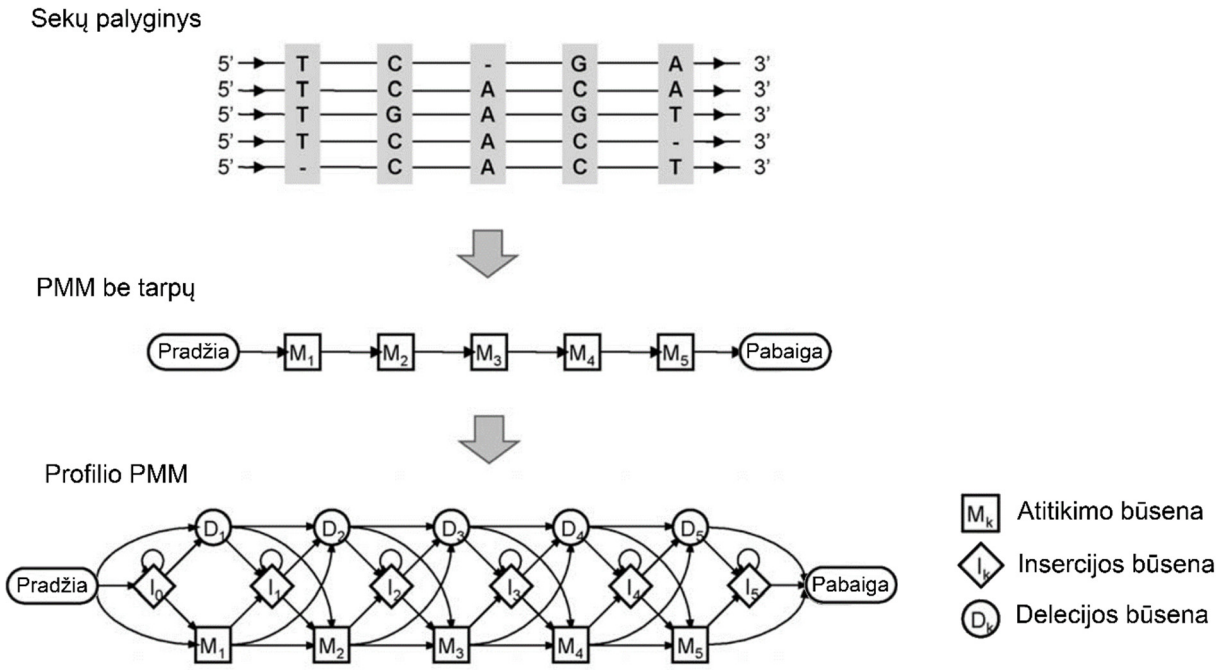
Čia, L – tikslinio baltymo ilgis, N – sekų skaičius, $S_{m,n}$ – *m* ir *n* sekų tarpusavio panašumas. Bendrai Neff atitinka reprezentatyvių 80 % panašumo klasterių sekų skaičių

normalizuotą pagal tikslinės sekos ilgį. Toks sekų rinkinio įvairovės matavimo standartas yra populiarus sekų rinkiniui taikant kovariacijos analizę.

Pagal sekos ilgį nenormalizuotas Neff įvertis vadinamas Meff.

1.6. Paslėptieji Markovo modeliai

Paslėptieji Markovo modeliai (angl. *hidden Markov models*) (PMM) yra statistiniai modeliai, kurie gali būti naudojami aprašyti stebimų įvykių, priklausančių nuo nematomų vidinių veiksnių, kitimą laike. Stebima būsena yra vadinama simboliu, o nematomas veiksnys – būsena. PMM yra sudarytas iš dviejų stochastinių procesų: nematomo paslėptų būsenų proceso ir matomo stebimų simbolių proceso. Paslėptos būsenos sudaro Markovo grandinę (angl. *Markov chain*), stebimo simbolio tikimybių pasiskirstymas priklauso nuo šios būsenos (Rabiner, 1989). Markovo grandinėje tikimybių pasiskirstymas duotojoje būsenoje priklauso tik nuo praeitos būsenos ir nepriklauso nuo visų kitų būsenų. Kitaip tariant, sekančios būsenos tikimybė tiesiogiai priklauso tik nuo dabartinės būsenos, o praeities būsenos tampa nebesvarbios, esant dabartinėje būsenoje. Markovo grandinėje būsenų skaičius yra baigtinis, o perėjimas tarp būsenų priklauso nuo būsenoms specifinio tikimybių rinkinio, vadinamų perėjimo tikimybėmis. Kalbant apie biologinių molekulių sekas, esama būsena atitinka specifinį simbolį sekoje, o sekanti ir praeita būsena atitinka sekantį ir praeitą simbolį (Eddy, 1996).

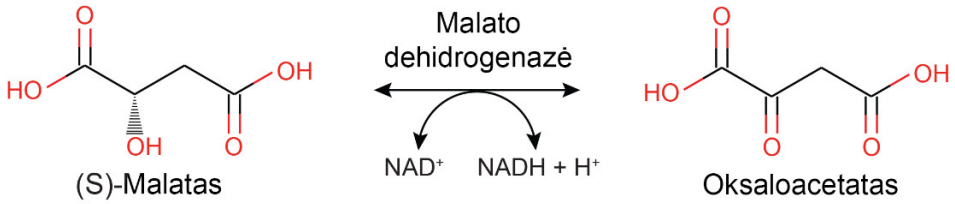


1.7 pav. Profilio PMM modelio sudarymas ir jo reprezentacija biologinei sekai. Pritaikytas pagal (Yoon, 2009).

PMM yra Markovo grandinės generalizacija, kurioje vidinė būsena nėra tiesiogiai stebima, tačiau ji sukuria stebimą būseną. Kiekviena paslėpta būsena emituoja simbolį priklausomai nuo simbolių emisijos tikimybės, tarpusavyje būsenos yra apjungtos būsenos perėjimo tikimybėmis (Mor ir kt., 2021). Pradedant nuo pradinės būsenos, būsenų seka yra sukuriama einant iš būsenos į būseną priklausomai nuo perėjimo tikimybių iki kol pasiekiama pabaigos būsena. Baltymų sekų palyginime ir paieškoje dažniausiai naudojami profilio PMM (Eddy, 1996). Profilio PMM naudoja pozicijai specifinius aminorūgščių tikimybių skirstinius vadinamus atitikimo būsenomis (angl. *match states*) ir pozicijai specifinius tikimybių skirstinius insercijoms ir delecijoms modeliuoti. Ši profilių savybė leidžia išsaugoti informacija tiek apie daugybinio sekų palyginio konservatyvumą skirtinguose pozicijose, tiek ir skirtingą insercijų ir delecijų pasitaikymo dažnį (**1.7 pav.**) (Eddy, 2004).

Vienas iš pagrindinių profilio PMM apribojimų yra prarasta informacija apie aukštesniojo laipsnio koreliacijas. Taikant PMM yra daroma prielaida, kad aminorūgštis, esanti specifinėje pozicijoje, nepriklauso nuo visų kitų pozicijų. Dėl šios priežasties sekos generuojamos naudojant profilio PMM yra panašios į natūralias pagal pirmos eilės statistikas (pozicinį konservatyvumą), tačiau tokios sekos nepasižymi reikšmingomis antros eilės statistikomis. Egzistuoja tam tikros profilio PMM variacijos, kurios gali modeliuoti porines simbolių tarpusavio priklausomybes, vienas iš tokių PMM variantų yra kontekstui jautrūs profilio PMM (Yoon ir Vaidyanathan, 2006).

1.7. Malato dehidrogenazės



1.8 pav. Nuo NAD^+ priklausomos MDH (EC 1.1.1.37) katalizuojama reakcija.

Trikarboksirūgščių ciklas – tai oksidacinis kelias, randamas aerobiniuose prokariotuose ir eukariotuose. Viena iš ciklo reakcijų yra malato oksidacija iki oksalacetato katalizuojama nuo NAD^+ arba $NADP^+$ priklausomų malato dehidrogenazių (MDH) (**1.8 pav.**). MDH taip pat dalyvauja aspartato biosintezėje, malato-aspartato šaudyklėje, gliukoneogenezėje ir lipogenezėje (Takahashi-Íñiguez ir kt., 2016).

MDH yra itin plačiai gyvuose organizmuose paplitęs fermentas, tai lemia didelę fermento sekų įvairovę. Bakterinės kilmės MDH pasižymi skirtingomis molinėmis masėmis, subvienetų struktūra, domenų organizacija ir katalizinėmis savybėmis. Prokariotuose randamos MDH įprastai

būna vienos iš dviejų oligomerinių būsenų: homodimerai arba tetramerai. Gram-neigiamuose organizmuose įprastai randama dimerinės būsenos MDH, kurios vieno subvieneto masė varijuoja nuo 30 kDa iki 38 kDa. Daugelis gram-teigiamų bakterijų ir archėjų turi tetramerinės būsenos MDH (subvieneto masė kinta nuo 32 kDa iki 43 kDa).

MDH gali būti klasifikuojamos pagal jų specifiškumą kofaktoriams: nuo NAD^+ priklausomos MDH (EC 1.1.1.37) ir nuo NADP priklausomos MDH (EC 1.1.1.82). Dauguma bakterinių MDH yra nuo NADH priklausomos, tačiau yra ir išimčių. Be to, dalis archėjų turi MDH, kurios yra vienodai priklausomos nuo NAD^+ ir $NADP^+$.

1.7.1. Bakterinių MDH struktūra

MDH pasižymi dideliu struktūrų panašumu netgi tada, kai stebimas menkas tarpusavio sekų panašumas. Homodimerinės formos MDH subvienetai yra sudaryti iš 11 beta klosčių ir devynių alfa spiralių. Subvienete galima išskirti du atskirus domenus: N-galinis domenas turi kofaktoriaus prisijungimo vietą, C-galiniame domene yra lokalizuotas aktyvusis centras ir substrato prijungimo vieta. N-galinis domenas yra sudarytas iš paralelių beta klosčių (Rozmano sanklodos motyvas). Iki tam tikro lygio šis domenas yra konservatyvus ir kituose dehidrogenazėse. Pilnas aktyvusis centras susidaro įduboje tarp dviejų monomerų (Minárik ir kt., 2002).

Dimerų sąveikos paviršius homodimerinėse MDH yra sudarytas iš penkių spiralių kompaktiškai išsidėliojusių tarp dimerų (Breiter ir kt., 1994). *E. coli* MDH dimerizacijos paviršius sudaro apie 1600 $Å^2$, tai atitinka apie 13,5 % bendro monomero paviršiaus (Breiter ir kt., 1994). Dimero struktūrinį stabilumą palaiko tiesioginiai vandeniliniai ryšiai, per vandens molekules sudaromi vandeniliniai ryšiai ir hidrofobiniai kontaktai (Minárik ir kt., 2002).

1.8. Apibendrinimas

In vitro baltymų funkcionalumo įvertinimas išlieka brangus ir daug laiko reikalaujantis procesas, kuriuo įprastai patikrinti galima tik dešimtis ar šimtus baltymų variantų. Dėl šios priežasties vis svarbesnis tampa baltymų ar jų rinkinių įvertinimas *in silico*, nes nuo to didžiąją dalimi priklauso ir rezultatai, kurių galima pasiekti tobulinant MM paremtus metodus.

Šiame darbe analizuojamos baltymų sekos sugeneruotos generatyviniais priešiškais tinklais. Pagal apmokymui naudotą baltymų sekų rinkinį GPT gali sugeneruoti naujus sekų variantus. Šios sekos kyla iš neuroninio tinklo išmokto sekų skirstinio. Sugeneruoti sekų variantai turėtų reprezentuoti GPT išmoktą baltymų sekų pasiskirstymą, kartu atkuriant ir mokymo sekoms būdingą biologinį funkcionalumą.

Darbe siekiama įvertinti sugeneruotų ir realių/natūralių baltymų sekų rinkinių panašumus ir skirtumus, jų įvairovę bei galimybę tikslingai keisti generuojamų baltymų savybes. Tokia

analizė padeda aiškinantis MM modelių išmoktas savybes biologinių sekų kontekste ir jų veikimo principus. Skirtumų radimas tarp sugeneruotų ir realių variantų padeda toliau tobulinti MM modelius, išvengiant brangių ir ilgų laboratorinio testavimo ciklų.

Modeliniu baltymu GPT apmokymui pasirinktos bakterinės kilmės MDH su priskirtu EC numeriu 1.1.1.37. MDH pasirinktos dėl šių priežasčių:

- Šis fermentas turi daug viešose duomenų bazėse prieinamų sekų, kurių tarpusavio panašumas varijuoja plačiame diapazone.
- Tai yra pakankamai sudėtingas taikinys, kurio pilnam aktyvumui reikia, kad jis sudarytų atitinkamą oligomerinę būseną bei kartu prisijungtų substratą ir kofaktorių.
- Fermento aktyvumas gali būti lengvai stebimas *in vitro* dėl reakcijos metu vykstančios NADH konversijos į NAD^+.

2. METODAI

2.1. Programiniai įrankiai

Lokaliai naudoti įrankiai:

- AliView 1.26 (Larsson, 2014) – sekų palyginių vizualizavimui ir redagavimui
- BLAST+ 2.8.1 (Camacho ir kt., 2009) – sekų panašumo įvertinimui
- ClustalO v1.2.3 (Sievers ir kt., 2011) – daugybiniams sekų palyginiams sudaryti
- HHsuite 3.3.0 (Steinegger, Meier, ir kt., 2019, p.) – sekų paieškai pagal sekų panašumą atlikta
- HMMER v3.3.2 (Eddy, 2009) – PMM profiliui sudaryti ir sekų generavimui pagal PMM profilį
- iPython 7.22.0 (Perez ir Granger, 2007) – darbo aplinka
- MAFFT v7.471 (Katoh ir Standley, 2013) – daugybiniams sekų palyginiams sudaryti
- MMseqs2 (Steinegger ir Söding, 2017, p. 2) – sekų klasterizavimui
- MODELLER 10.1 (Eswar ir kt., 2006) – homologinių baltymų modelių sudarymui
- Pymol 2.4.0 – baltymų struktūrų vizualizacijai ir modifikavimui
- Python 3.7.6
- trRosetta (J. Yang ir kt., 2020) – sekos struktūriniam atitikimui įvertinti
- USearch v11.0.667 (Edgar, 2010) – sekų porų panašumui įvertinti

Įrankiai serveriuose:

- BLAST (Madden ir kt., 1996) – sekų variantų paieškai
- ConSurf (Ashkenazy ir kt., 2016) – nepriklausomam baltymo konservatyvumo įvertinimui
- WebLogo (Crooks ir kt., 2004) – sekų logo sudarymui

Python bibliotekos:

- Biopython 1.78 (Cock ir kt., 2009) – porinių palyginių sudarymui, sekų modifikavimui, baltymų sekų analizei
- Matplotlib 3.3.4 (Hunter, 2007) – grafiniam rezultatų atvaizdavimui
- NumPy 1.20.2 (Harris ir kt., 2020) – sekų rinkinių modifikavimui ir statistinei analizei
- Pandas 1.2.3 (McKinney, 2010) – sekų rinkinių modifikavimui ir statistikų apskaičiavimui
- Scikit-learn 0.24.2 (Pedregosa ir kt., 2011) – duomenų dimensijų sumažinimui su t-SNE
- SciPy 1.6.2 (Virtanen ir kt., 2020) – statistinei analizei
- Seaborn 0.11.1 (Waskom, 2021) – grafiniam rezultatų atvaizdavimui
- Statsmodels v0.12.2 (Seabold ir Perktold, 2010) – statistinei analizei

2.2. Kompiuteriniai metodai

2.2.1. Mokymo rinkinio sekų surinkimas

Mokymo rinkinio duomenys gauti iš UniprotKB duomenų bazės (2019-01-10). Mokymui pasirinkti visi unikalūs bakteriniai fermentai su priskirtu EC numeriu 1.1.1.37 (nuo NAD^+ priklausomos malato dehidrogenazės). Iš pradinio sekų rinkinio pašalintos sekos ilgesnės nei 516 aminorūgščių ir trumpesnės nei 64 aminorūgštys. Be to, pašalintos sekos turėjusios nestandartines arba tiksliai nenustatytas aminorūgštis.

2.2.2. Mokymo rinkinio normalizavimas

Prieš GPT mokymą mokymo duomenų rinkinys normalizuotas pagal sekų panašumo pasiskirstymą. Normalizavimui atlikti mokymo rinkinio sekos klasterizuotos ties 70 % panašumo riba naudojant MMseqs2 įrankį. Klasterizavimas atliktas „easy-cluster“ moduliu naudojant standartinius parametrus su „--min-seq-id 0.7“ nustatymu. Iš klasterių grupės, turėjusios mažiau nei 3 sekas, 20 % klasterių atskirta į validacijos sekų rinkinį, likusios 13272 sekos naudotos tinklo mokymui.

Mokymo metu tolygiai iš kiekvieno klasterio tinklui suteikiamas atsitiktinis klasterio atstovas, taip reprezentuojant kiekvieną klasterį tolygiu dažniu.

2.2.3. Sugeneruotų sekų panašumo nustatymas tinklo mokymo metu

Sugeneruotų sekų panašumo įvertinimui tinklo mokymo metu kiekvienam mokymo žingsniui sugeneruotos 64 sekos. Šių sekų panašumas į natūralias įvertintas BLAST įrankiu. Panašumui įvertinti naudoti standartiniai parametrai.

2.2.4. Mokymo rinkinio normalizavimas sekų analizei

Atliekant sekų analizę mokymo rinkinys normalizuotas pagal GPT mokymui naudotą normalizavimo principą. Iš kiekvieno mokymui naudoto klasterio traukiama po 21 seką, iš pradžių traukiamos unikalios atsitiktinės sekos. Jeigu ištraukus unikalias sekas klasteris reprezentuojamas mažiau nei 21 seka, sekos toliau traukiamos atsitiktinai. Procesas kartojamas kol kiekvieną klasterį reprezentuoja 21 seka.

Tokiu būdu iš 922 70 % panašumo klasterių išrenkamos 19362 sekos. Šiuo atveju toks galutinis kiekis sekų pasirinktas tam, kad natūralių ir sugeneruotų sekų rinkiniai turėtų apytiksliai vienodą sekų skaičių ($19362 \approx 20000$).

2.2.5. Bendro daugybinio sekų palyginio sudarymas

Siekiant kartu analizuoti ir palyginti sugeneruotų ir mokymui naudotų sekų rinkinius, iš jų yra sudaromas bendras sekų palyginys.

Sudarant bendrą mokymo ir sugeneruotų sekų rinkinių palyginį sekos pirma apjungiamos kartu. Iš šio bendro sekų rinkinio yra sudaromas DSP. Palyginys sudarytas naudojant ClustalO įrankį su standartiniais parametrais. Sudarius bendrą palyginį mokymo ir sugeneruotos sekos yra atskiriamos į atitinkamai mokymo ir sugeneruotų sekų rinkinius tolimesnei analizei.

2.2.6. Sekų generavimas pagal PMM profilį

Iš normalizuoto mokymo sekų rinkinio sudaromas sekų palyginys. Palyginiui sudaryti naudojamas ClustalO įrankis su standartiniais parametrais. Šis palyginys naudojamas PMM profilio sudarymui. PMM profilis sudarytas HMMER paketo įrankiu hmmbuild naudojant standartinius parametrus su nustatymu „-wnone“. Šis parametras naudotas išjungti sekų normalizavimui, kurį taiko hmmbuild, kadangi profilio sudarymui yra naudojamas jau normalizuotas sekų rinkinys. Sudarytas PMM profilis panaudotas sekų generavimui, tam naudotas hmmemit įrankis iš HMMER paketo. Viso sugeneruota 20000 PMM sekų.

2.2.7. PMM sugeneruotų sekų sulygiavimas

PMM sugeneruotos sekos sulygiuotos kartu su natūraliomis naudojant MAFFT įrankį. Šiuo atveju ClustalO nebuvo naudojamas, nes palyginio sudarymas trunka ilgiau nei 3 dienas. Sulygiavimui su MAFFT naudoti standartiniai įrankio parametrai.

2.2.8. Sekų palyginio pozicijų filtravimas

Dėl itin didelės sekų įvairovės būtina pašalinti mažo padengimo palyginio pozicijas, kadangi šios gali reikšmingai pakeisti toliau lyginamų statistikų reikšmes. Iš bendro palyginio pašalinti stulpeliai su mažesniu nei 75 % padengimu abiejuose sekų rinkiniuose (mokymo ir GPT/PMM sugeneruotame). Aminorūgščių dažniui įvertinti ir stulpeliams pašalinti naudotos Python Pandas bei NumPy bibliotekos.

2.2.9. Šanono entropijos apskaičiavimas

Šanono entropija apskaičiuojama sugeneruotų ir natūralių sekų rinkiniams. Entropijai apskaičiuoti naudojamas SciPy paketas pasirinkus logaritmą pagrindu 21 (atitinka 20 standartinių aminorūgščių ir tarpo simbolį). Grafiniam entropijos atvaizdavimui apskaičiuojamas slenkantis vidurkis su lango dydžiu 15.

2.2.10. Pozicinio aminorūgščių dažnio palyginimas

Pozicinis aminorūgščių dažnis įvertintas bendrai sulygiuotiems sekų rinkiniams su atliktu palyginio pozicijų filtravimu. Kiekvienam sekų rinkiniui apskaičiuotas pozicinis aminorūgščių dažnis kiekvienoje palyginio pozicijoje (f_i^{α}). Sekų rinkiniams palyginti apskaičiuojamas Pirsono koreliacijos koeficientas tarp sekų rinkinių aminorūgščių dažnių.

2.2.11. Porinių aminorūgščių dažnių palyginimas

Porinių aminorūgščių dažnių apskaičiavimui tarp visų galimų baltymo aminorūgščių pozicijų porų (i, j) nustatytas aminorūgščių porų (α, β) pasiskirstymo dažnis ($f_{ij}^{\alpha\beta}$). Statistiniam sugeneruotų ir natūralių rinkinių porinių aminorūgščių dažnių palyginimui apskaičiuojamas Pirsono koreliacijos koeficientas. Grafiniam atvaizdavimui pašalinamos poros, kurių dažnis abiejuose sekų rinkiniuose yra mažesnis nei 1 %.

2.2.12. Aminorūgščių kovariacijos palyginimas

Porų kovariacija apskaičiuota pagal formulę:

$$C_{\alpha\beta}^{ij} = f_{\alpha\beta}^{ij} - f_{\alpha}^{i} f_{\beta}^{j}$$

Kur α – aminorūgštis pozicijoje i, β – aminorūgštis pozicijoje j. $f_{\alpha\beta}^{ij}$ – jungtinis aminorūgščių α ir β dažnis. $f_{\alpha}^{i} f_{\beta}^{j}$ pozicinių α ir β dažnių sandauga.

Aminorūgščių kovariacija apskaičiuojama sugeneruotų ir natūralių sekų rinkiniams. Rezultatams palyginti naudojamas Pirsono koreliacijos koeficientas.

2.2.13. Bendros informacijos palyginimas

Bendra informacija apskaičiuota pagal formulę:

$$BI_{ij} = H_i + H_j - H_{ij}$$

Kur H_i ir H_j atitinka entropiją pozicijose i ir j atitinkamai, H_{ij} atitinka jungtinę pozicijų i ir j entropiją.

Triukšmo ir filogenetinių komponentų pašalinimui iš BI pritaikomas vidutinio produkto pataisymas (angl. *average product correction*) (Dunn ir kt., 2008). BI skirstiniai tarp sekų rinkinių palyginami Pirsono koreliacijos koeficientu.

2.2.14. Dimensijų sumažinimas klasterių pasiskirstymo erdvėje įvertinimui

Dimensijų sumažinimui pasirinktas t-SNE metodas, esantis Sklearn Python bibliotekoje. Atvaizdavimui pasirinkti 75 % panašumo reprezentatyvūs klasterių nariai. Natūralių ir sugeneruotų reprezentatyvių sekų rinkiniai buvo apjungti. Tarp visų reprezentatyvių sekų sudaryta seka panašumo matrica. Ji sudaryta ClustalO įrankiu nustačius „--distmat-out“ parametrą. Sekų panašumo matricos dimensijų sumažinimui naudotas t-SNE su parametrais: „early exaggeration = 12, learning rate = 200, maximum number of iterations = 1000, perplexity = 7“. Dvidimensinių taškų koordinatės atvaizduotos erdvėje, taško dydis nustatytas pagal klasterio dydį. Taško atvaizdavimo dydis apskaičiuotas pagal formulę:

$$r = 6{s_{cl}}^{0,8}$$

Kur s_{cl} – klasterį sudarančių sekų skaičius.

2.2.15. Sekų įvairovės palyginimas

Natūralių ir sugeneruotų sekų rinkinių įvairovė palyginama kaip klasterių skaičiaus santykis. Sekos klasterizuojamos sekų panašumo rėžiuose nuo 0 iki 100 %, klasterizavimui naudojamas MMseqs įrankis su standartiniais parametrais. Ties kiekvienu panašumo lygiu apskaičiuojamas sugeneruotų ir natūralių sekų klasterių santykis. Jis yra vertinamas kaip įvairovės padidinimas/sumažinimas ties atitinkamu panašumo lygiu.

2.2.16. Sekų logo sudarymas funkcinių pozicijų palyginimui

Funkcinės pozicijos pasirinktos pagal *E. coli* MDH. Specifinės pozicijos pasirinktos pagal UniProt anotacijas (UniProt ID: A1AGC9). Pozicijų aminorūgščių skirstiniai gauti iš bendrai sulygiuotų normalizuoto mokymo ir GPT sugeneruoto sekų rinkinių. Pozicijoms sekų logo sudarytas WebLogo įrankiu, naudojami standartiniai WebLogo serverio parametrai.

2.2.17. Baltymų sekų įvertinimas pagal trRosetta nuostolius

2.2.17.1. Sekų rinkinių paruošimas

trRosetta įvertinimui sudaryti keturi sekų rinkiniai: natūralių, sugeneruotų, pagal PSM sukurtų ir atsitiktinių sekų. Visų sekų rinkinių galutiniai variantai atitiko daugybinio sekų palyginio struktūrą: visos sekos vienodo ilgio, stulpeliai atitinka vienodą poziciją struktūroje.

Natūralių ir sugeneruotų subrinkinių paruošimui atlikta sekų paieška atitinkamuose pradiniuose (nenormalizuotose) sekų rinkiniuose. Sugeneruotų ir natūralių sekų atveju visos rinkinių sekos sulygiuotos poromis su taikinio seka pagal NW algoritmą. Pašalintos tos sekų poros, kurių palyginys buvo ilgesnis nei 315 ir pašalintos sekos trumpesnės nei 300 aminorūgščių.

Atrinkti pradiniai subrinkiniai sulygiuoti naudojant MAFFT (dėl panašaus sekų ilgio pasirinkti LINS-I parametrai). Palyginiai vizualiai įvertinti (Aliview) ir pašalintos sekos su didesnėmis nei penkių aminorūgščių insercijomis. Iš likusių palyginių sekų pašalintos insercijos, o delecijos užpildytos taikinio seką atitinkančiomis aminorūgštimis. Palyginio galai sutrumpinti taip, kad atitiktų baltymo struktūrą (PDB ID: 6KA1).

Atsitiktinio sekų rinkinio sukūrimui naudotas sugeneruotų sekų subrinkinys (paruoštas taip kaip aprašyta aukščiau). Sugeneruotų sekų subrinkinio aminorūgštys, pozicijose, kuriose jos skiriasi nuo taikinio sekos, atsitiktinai pakeistos viena iš 20 standartinių aminorūgščių. Taip sukurtas atsitiktinis rinkinys. Dėl atsitiktinio aminorūgščių keitimo šio rinkinio baltymų sekos pasižymėjo nežymiai didesniu panašumu į natūralias sekas lyginant su sugeneruotų sekų rinkiniu.

PSM rinkinys sukurtas pagal analizei normalizuoto mokymo rinkinio sekų palyginio PSM. Naudotas prieš tai sulygiuotas mokymo sekų rinkinys. Iš sulygiuotų sekų rinkinio pašalinti stulpeliai, kurie atitinka palyginio tarpus taikinio sekoje. Kiekvienam šio palyginio stulpeliui apskaičiuotas procentinis aminorūgščių dažnis. Sekų pagal PSM kūrimui kiekvienai PSM pozicijai priskirta aminorūgštis. Aminorūgšties priskyrimo tikimybė tiesiogiai priklausė nuo toje pozicijoje apskaičiuoto aminorūgščių dažnio.

Rinkinių normalizavimui pagal panašumą į taikinio seką, iš galutinio palyginimo pašalintos visos sekos turėjusios didesnį nei 50% panašumą į taikinio seką.

2.2.17.2. Struktūros paruošimas

Struktūra *E. coli* malato dehidrogenazei parsiųsta iš www.rcsb.org serverio (PDB ID: 6KA1). Struktūros modifikavimui naudotas Pymol įrankis. Iš struktūros C galo pašalinta pirmoji aminorūgštis (G), visos grandinės išskyrus A ir nebaltyminės kilmės atomai. Modifikuota struktūra pernumeruota taip, kad pirmoji aminorūgštis turėtų eilės numerį 1.

2.2.17.3. trRosetta nuostolių apskaičiavimas

trRosetta nuostoliai įvertinti pagal baltymo θ, φ, ω kampų ir atstumų matricos spėjimus, bei tų pačių metrikų realias reikšmes gautas pagal PDB struktūrą. Šių metrikų spėjimui panaudota atskira seka be DSP, taip gaunant įvertį specifiškai pasirinktai sekai nepriklausomai nuo kitų sekų.

trRosetta nuostolių įvertinimui naudoti įrankiai patalpinti github.com/gjoni/trDesign (Norn ir kt., 2020).

2.2.18. Sekų generavimas tikslingai keičiant įvesties vektorių

Sekų generavimui yra naudojamas 128 dimensijų vektorius, kurį tinklas transformuoja į baltymo seką. Kiekviena vektoriaus dimensija gali būti keičiama pasirinktinai. Todėl siekiant

nustatyti generuojamų sekų savybių priklausomybę nuo įvesties vektoriaus verčių, baltymų sekos generuotos tiesiškai keičiant po vieną įvesties vektoriaus dimensiją. Sekos generuotos kiekvieną dimensiją atskirai keičiant nuo -1 iki 1, viso sugeneruojama 1024 sekų per dimensiją. Tokiu būdu per visas įvesties vektoriaus dimensijas sugeneruotos 131072 MDH sekos.

Imtis nuo -1 iki 1 pasirinkta pagal mokymo metu naudotą imtį: mokymo įvesties vertės parenkamos iš normalaus skirstinio su vidurkiu 0 ir standartiniu nuokrypiu 0,5, vertei viršijus dvejus standartinius nuokrypius reikšmė renkama iš naujo.

2.2.19. Įvesties vektoriumi keičiamų sekų savybių įvertinimas

Visoms sekoms sugeneruotoms specifiškai keičiant įvesties vektoriaus dimensijas apskaičiuotos įvairios savybės. Baltymų molinė masė, izoelektrinis taškas, nestabilumo indeksas ir lankstumas įvertinti Biopython bibliotekos įrankiais. Panašumas į mokymo sekas įvertintas Usearch įrankiu. Be minėtų savybių įvertintas ir atskirų aminorūgščių bei jų grupių dažnis sekose (**1 priedas, 2 lentelė**). Prieš įvertinant kiekvienos dimensijos sekų savybes pašalinamos pasikartojančios sekos. Savybių priklausomybei nuo įvesties vektoriaus įvertinti tarp apskaičiuotų savybių verčių ir įvesties vektoriaus vertės apskaičiuojamas Pirsono koreliacijos koeficientas.

2.2.20. Nepriklausomas konservatyvumo įvertinimas

2.2.20.1. Konservatyvumo įvertinimas pagal sugeneruotas sekas

Konservatyvumui įvertinti panaudotas sugeneruotų sekų subrinkinys naudotas trRosetta nuostolių įvertinime. Pagal pirmąją subrinkinio seką (atitinka sugeneruotos sekos ID 29) iš palyginio pašalintos insercijos. Kiekvienam šio sekų rinkinio stulpeliui apskaičiuota Šanono entropija. Apskaičiuota entropija šiuo atveju laikyti sekų rinkinio konservatyvumu.

2.2.20.2. Šablono paieška homologiniam modeliui

Šablono paieška taikiniui (ID 29) vykdyta dviem etapais. Pirmame etape atliekama sekų paieška ir sudaromas taikinio sekos palyginys su homologinių sekų rinkiniu. Homologinių sekų paieškai ir palyginio sudarymui naudotas hhblits įrankis, paieška atlikta Uniref30 sekų duomenų bazėje (2020_06 leidimas) (Mirdita ir kt., 2017), naudota viena paieškos iteracija (-n 1). Gautas homologinių sekų palyginys naudotas sekų su nustatytomis struktūromis paieškai. Ji atlikta PDB70 duomenų bazėje (2021-04-14 leidimas) naudojant HHsearch įrankį. Modeliavimui pasirinktas šablonas su didžiausiu atitikimo įverčiu pagal HHsearch, tai atitiko struktūrą su PDB ID 3NEP. Struktūros paruošimui ir poriniam sekų palyginiui gauti naudotas hhmakemodel.py įrankis esanti HHsuite įrankių pakete.

2.2.20.3. Homologinio modelio sudarymas

Homologinis modelis sudarytas pagal praeitame etape gautą pir formato porinį palyginį ir paruoštą taikinio struktūrą. Modeliui sudaryti naudojamas MODELLER įrankis (parametrai „refine.slow, repeat_optimization = 3“). Viso sukurta 60 modelių, iš jų pasirinktas modelis įvertintas žemiausiu zDOPE įverčiu (-1.386).

2.2.20.4. Konservatyvumo įvertinimas pagal ConSurf serverį

Konservatyvumui įvertinti pagal ConSurf serverį naudota 3NEP MDH struktūra. Homologų paieškai ir konservatyvumo įvertinimui naudoti standartiniai ConSurf serverio parametrai.

2.2.21. GPT tinklo architektūrą

Pasirinkta GPT architektūra yra sudaryta iš dvejų neuroninių tinklų: generatoriaus ir diskriminatoriaus, abu neuroniniai tinklai naudoja likutinius neuroninio tinklo (angl. *residual neural network (*ResNet)) blokus. Kiekvienas diskriminatoriaus ResNet blokas yra sudarytas iš trijų konvoliucinių sluoksnių ir nesandarių ReLU aktyvacijų. Generatoriaus ResNet blokai sudaryti iš vieno paprasto konvoliucinio ir dviejų transponuotų konvoliucinių sluoksnių bei nesandarių ReLU aktyvacijų. Kiekvienas neuroninis tinklas turėjo po vieną savęs dėmesio sluoksnį (angl. *self attention*).

Diskriminatoriaus įvestis buvo užkoduota naudojant vienetinį kodavimą (angl. one-hot encoding), kodavimo žodynas buvo sudarytas iš 21 reikšmės (20 kanoninių aminorūgščių ir ženklas, nurodantis baltymo sekos pradžią ir/ar pabaigą).

Generatoriaus įvestimi naudotas 128 reikšmių vektorius. Kiekviena vektoriaus reikšmė atsitiktinai parinkta pagal normalųjį skirstinį su vidurkiu 0 ir standartiniu nuokrypiu 0,5, maksimali-minimali reikšmės buvo apribotos iki dviejų standartinių nuokrypių (atitinka kitimą nuo -1 iki 1).

2.2.22. Duomenų prieinamumas

Analizei atlikti ir pateiktiems rezultatams atkartoti skirtas programinis kodas pateiktas: https://github.com/irmantasr/GPT-sugeneruotu-seku-analize.

GPT naudotas MDH sekų generavimui prieinamas: https://github.com/Biomatter-Designs/ProteinGAN.

2.3. Laboratoriniai metodai

Siekiant nustatyti kiek įmanoma daugiau tirpių MDH variantų gryninimas atliekamas dviem metodais, su skirtingais ekspresijos vektorių kamienais ir skirtingomis gryninimo sąlygomis. Šie metodai vadinami: metodas 1 ir metodas 2.

2.3.1. Baltymo sekų užsakymas

Prie užsakomų GPT sekų pridėtas C-galinis jungtukas su keturiais histidinais (AAALEHHHH), galutiniai sekų variantai turėjo šešis papildomus histidinus, esančius ekspresijos vektoriuje. GPT sugeneruotų sekų sintezė, klonavimas į pET21a raiškos vektorių ir galutinio varianto sekos patvirtinimas atliktas Twist Bioscience. Užsakomos DNR sekos optimizuotos pagal *E. coli* kodonų dažnius su vidiniais Twist Bioscience įrankiais.

2.3.2. Kompetentinių ląstelių paruošimas

Į mėgintuvėlį su 5 mL LB terpės pernešamas mažas kiekis *E. coli* ląstelių. Ląstelės auginamos 37 °C temperatūroje purtant tol, kol sugertis esant 600 nm bangos ilgiui pasiekia 0,6 – 0,8 optinius vienetus. Ląstelėms surinkti, ląstelės centrifuguojamos 4 °C temperatūroje 10 min. 1000×*g* pagreičiu. Terpė nupilama, ląstelės resuspenduojamos 5 mL NaCl tirpalo (5 mM Tris-HCl pH 8, 100 mM NaCl, 5 mM $MgCl_2$) ir vėl centrifuguojamos. Pašalinus supernatantą, ląstelės resuspenduojamos 2,5 mL šalto $CaCl_2$ tirpale (5 mM Tris-HCl pH 8, 100 mM $CaCl_2$, 5 mM $MgCl_2$) ir inkubuojamos 30 min. ledo vonioje. Po to ląstelės vėl surenkamos centrifuguojant ir resuspenduojamos 200 µL šalto $CaCl_2$ tirpale.

2.3.3. Ląstelių transformacija

Transformacijai į 50 µL $CaCl_2$ metodu paruoštų kompetentinių ląstelių įpilama 10 ng plazmidinės DNR ir 15 min. inkubuojama ledo vonelėje. Vykdomas temperatūrinis šokas 42 °C temperatūroje 2 min. ir vėsinama ledo vonioje. Mišinys skiedžiamas 450 µL LB terpės ir bakterijos gaivinamos 37 °C temperatūroje 30-60 min. Praėjus šiam laikui, ant lėkštelių su agarizuota LB mitybine terpe bei atrankai naudojamu ampicilinu išsėjama 100 µL bakterijų.

2.3.4. Metodas 1

2.3.4.1. MDH raiška

Užsakyti baltymų konstruktai transformuoti į BL21(DE3) *E. coli* kamieną. 15 µL transformacijos mišinio inokuliuota 500 µL LB terpės papildytos 100 µg/mL karbenicilino. Ląstelės augintos per naktį 96 šulinėlių lėkštelėje 32 °C temperatūroje purtant. Baltymų

ekspresijai į 1 mL autoindukcinės TB terpės papildytos su 100 µg/mL karbenicilino pernešta 30 µL naktinės kultūros. Ląstelės autoindukcinėje terpėje purtant augintos 4 h 37 °C po to, per naktį 18 °C. Ląstelės surinktos centrifuguojant ir užšaldytos -80 °C.

2.3.4.2. Baltymo gryninimas

Baltymų gryninimui ląstelės atšildomos, resuspenduojamos 200 µL lizės buferio (50 mM HEPES pH 7,4, 5 % glicerolio, 300 mM NaCl, 0,5 mM TCEP, 0,5 mg/mL lizocimo, 10 U/mL DNazės I, 2 mM $MgCl_2$) ir 30 min. inkubuojamos kambario temperatūroje. Po inkubacijos į ląstelių mišinį pridedama triton-X-100 iki galutinės koncentracijos 0,125 % (v/v). Ląstelės pakartotinai užšaldomos -80 °C 30 min. Po užšaldymo ląstelės atšildomos vandens vonelėje. Atšilę lizatai centrifuguojami 10 min. 3000×g taip pašalinant ląstelių liekanas. Nuo ląstelių pašalintas supernatantas perkeliamas į 96 šulinėlių lėkštelę. Į šulinėlius pridedama po 50 µL Talon reagento. Siekiant sumažinti nespecifinį baltymų prisijungimą, į kiekvieną šulinėlį pridėta po 10 mM imidazolo. Lėkštelė purtant inkubuojama kambario temperatūroje 30 min. ir po to perkeliama į 96 šulinėlių filtravimo lėkštelę. Filtravimo lėkštelė patalpinama virš 96 šulinėlių surinkimo lėkštelės ir centrifuguojama 1 min. 500×*g*. Likusios Talon mikrodalelės tris kartus nuplaunamos su 200 µL plovimo buferio (50 mM HEPES pH 7.4, 5 % glicerolio, 300 mM NaCl, 0,5 mM TCEP, 40 mM imidazolo). Baltymai pašalinami nuo mikrodalelių dvejomis 50 µL frakcijomis naudojant eliucijos buferį (50 mM HEPES pH 7,4, 5 % glicerolio, 300 mM NaCl, 0,5 mM TCEP, 250 mM imidazolo). 96 šulinėlių nudruskinimo lėkštelė ekvilibruojama su mėginio buferiu (50 mM HEPES pH 7,4, 5 % glicerolio, 300 mM NaCl, 0,5 mM TCEP), po ekvilibracijos į lėkštelę pernešamos abi eliucijos frakcijos. Nudruskinimo lėkštelė centrifuguojama 1 min. 1000×*g*, taip surenkant išgrynintą baltymą.

2.3.5. Metodas 2

2.3.5.1. MDH raiška

ArcticExpress *E. coli* kompetentinės ląstelės transformuojamos atitinkamais genų konstruktais. Transformuotos ląstelės inokuliuojamos 500 µL LB terpės papildytos su 15 µg/mL gentamicino, 50 µg/mL ampicilino ir auginamos per naktį 30 °C temperatūroje. 250 µL naktinės kultūros pernešama į 10 mL pusiau sintetinės terpės (1 % triptono, 0,5 % mielių ekstrakto, 0,268 % $(NH_4)_2SO_4$, 0,15 % NH_4Cl, 0,6 % KH_2PO_4, 0,4 % K_2HPO_4, 1 % glicerolio, pH 7,0) papildytos su 15 µg/mL gentamicino ir 50 µg/mL ampicilino. Ląstelės auginamos 2 h 37 °C, kol optinis tankis (OD600) pasiekia 0,6–0,8, tada terpė praturtinama 0,5 M sacharozės. Indukcija vykdoma 12 °C su 0,5 mM IPTG per naktį. Ląstelės surenkamos centrifuguojant (4000×*g* 10 min.

4 °C), resuspenduojamos 0,1 M kalio fosfato buferyje (pH 7,0) ir sonifikuojamos ledo vonelėje. Ląstelių liekanoms pašalinti lizatai centrifuguojami 16000×g, 4 °C.

2.3.5.2. Baltymo gryninimas

Baltymai gryninti naudojant HisPur™ Ni-NTA kolonėles. Kolonėlės su supernatantu praplaunamos plovimo buferiu (0,1 M kalio fosfatinis buferis, pH 7,4, NaCl 250 mM ir 40 mM imidazolo). Eliucija vykdoma su eliucijos buferiu (0,1 M kalio fosfatinis buferis, pH 7,4, NaCl 250 mM ir 300 mM imidazolo). Eliuatas dializuojamas prieš 0,1 M kalio fosfatinį buferį, pH 7,4.

2.3.6. Fermentinio aktyvumo matavimas

Aktyvumo matavimui išgrynintas MDH baltymas dedamas į reakcijos mišinį (0,15 mM NADH, 0,2 mM oksaloacetato ir 20 mM HEPES (pH 7,4)). Galutinis reakcijos tūris 100 µL, reakcija vykdoma kambario temperatūroje UV pralaidžioje 96 šulinėlių lėkštelėje, kiekvienam variantui atliekami trys pakartojimai. Aktyvumas įvertinamas matuojant NADH oksidacija iki NAD^+, sugertis matuojama ties 340 nm kas 30 s 15 min. laikotarpyje. Nespecifinei NADH oksidacijai įvertinti sugertis matuojama mėginiuose, kuriuose MDH variantai yra pakeisti jaučio serumo albuminu.

3. REZULTATAI

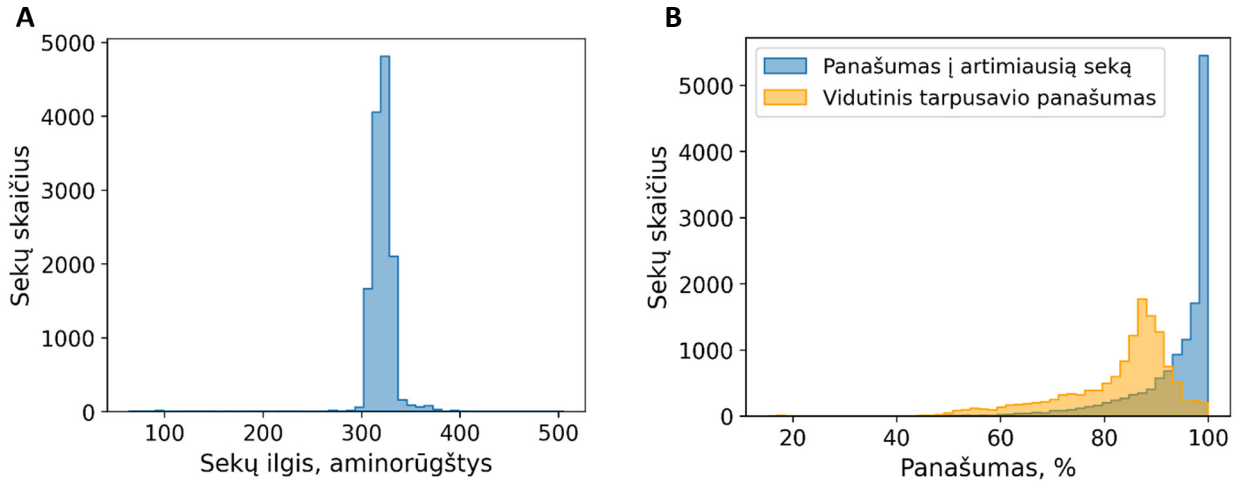
3.1. Modelio mokymas

3.1.1. GPT architektūros sukūrimas

GPT architektūros baltymų generavimui sukūrimą, parametrų optimizavimą ir tinklo apmokymą atliko Donatas Repečka (UAB „Biomatter Designs“). Tinklo architektūriniai pasirinkimai atlikti taip, kad tinklo dėmesys būtų skiriamas ir arti esantiems aminorūgščių kaimynams, ir tarpusavyje nutolusioms baltymo sekos aminorūgščių poroms. Dalis tinklo parametrų optimizuota pagal eksperimentinius rezultatus, pvz.: didinamojo sluoksnio (angl. *upsampling*) tipo pasirinkimas (Shi ir kt., 2016), nuostolių funkcija (Mescheder ir kt., 2018), generatoriaus-diskriminatoriaus žingsnių santykis (Arjovsky ir kt., 2017) ir kita. Tiksli GPT sandara aprašyta Repecka ir kt. 2021.

3.1.2. Mokymo sekų rinkinio paruošimas

Viso mokymui naudotos 13272 MDH sekos. Iš pradinio sekų rinkinio pašalintos sekos trumpesnės nei 64 ir ilgesnės nei 516 aminorūgščių. Be to, pašalinti sekų dublikatai ir sekos su nestandartinėmis aminorūgštimis. Galutiniame sekų rinkinyje MDH ilgiai varijuoja nuo 64 iki 505 aminorūgščių, vidutinis sekų ilgis lygus 319 ± 20 aminorūgščių (**3.1A pav.**). Didžioji dalis sekų mokymo rinkinyje turi itin panašius kaimynus (panašumas[1] >95 %) (**3.1B pav.**). Tai savaime nurodo tam tikrą sekų disbalansą mokymo rinkinyje. Vidutinio sekų tarpusavio panašumo vidurkis siekia 83 %.



3.1 pav. A – MDH sekų ilgių pasiskirstymas mokymo rinkinyje. B – Sekų tarpusavio panašumas mokymo rinkinyje. Panašumas į artimiausią seką nurodytas pagal globalų porinį palyginį. Vidutinis tarpusavio panašumas išreikštas kaip kiekvienos mokymo rinkinio sekos vidutinis panašumas į visas kitas mokymo rinkinio sekas.

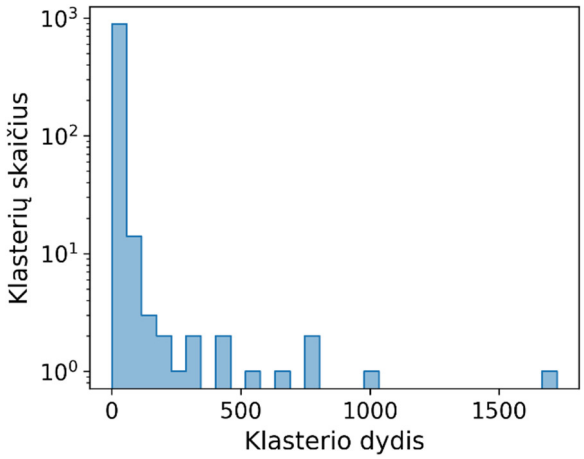
[1] Panašumas apskaičiuojamas pagal identiškas aminorūgštis lyginant su pilnu palyginio ilgiu.

Nehomogeniškas sekų pasiskirstymas sekų erdvėje (tarpusavio panašumo atžvilgiu) privalo būti taisomas prieš tinklo mokymą. Kitu atveju tikėtina, kad generuojami sekų variantai netolygiai reprezentuos baltymų erdvę. Be to, tai gali sukelti tinklo režimo žlugimą (angl. *mode collapse*).

Sekų rinkinio įvairovę galima įvertinti standartizuotais Meff ir Neff įverčiais. Nenormalizuotas efektyvių sekų skaičius (Meff) MDH duomenų rinkinyje siekia 2100, šį skaičių normalizavus pagal vidutinį sekų ilgį gaunamas Neff įvertis lygus 118. Tai sąlyginai didelis įvertis. Įprastai atliekant kontaktų nuspėjimą yra laikoma, kad pakankamas Neff įvertis yra 128 (Zheng ir kt., 2019). Reikia pabrėžti, kad kontaktų nuspėjimo atveju sekos renkamos atsižvelgiant į bendrą sekų panašumą. Tai nebūtinai užtikrina vienodą baltymų funkcionalumą. Šiuo atveju sekų variantų parinkimas vykdytas renkantis sekas su priskirta funkcija.

3.1.3. Mokymo sekų rinkinio normalizavimas

Mokymo sekų rinkinio normalizavimui taikytas sekų rinkinio klasterizavimas. Klasterizavimui pasirinktas minimalus 70 % sekų tarpusavio panašumas. Klasterizavimo metu sudaryta 918 klasterių (**3.2 pav.**). Nors didžiąją dalį visų klasterių sudaro maži (mažiau nei penkios sekos klasteryje) klasteriai, daugiau nei pusę sekų priklauso klasteriams, kurie yra didesni nei 200 sekų. Šiuo atveju normalizuojant siekiama, kad didesni klasteriai nebūtų per daug reprezentuojami, t.y. neužgožtų mažesnių klasterių.



3.2 pav. Klasterių dydžių ir jų skaičiaus pasiskirstymas pritaikius 70 % sekų panašumo klasterizavimą mokymo sekų rinkiniui.

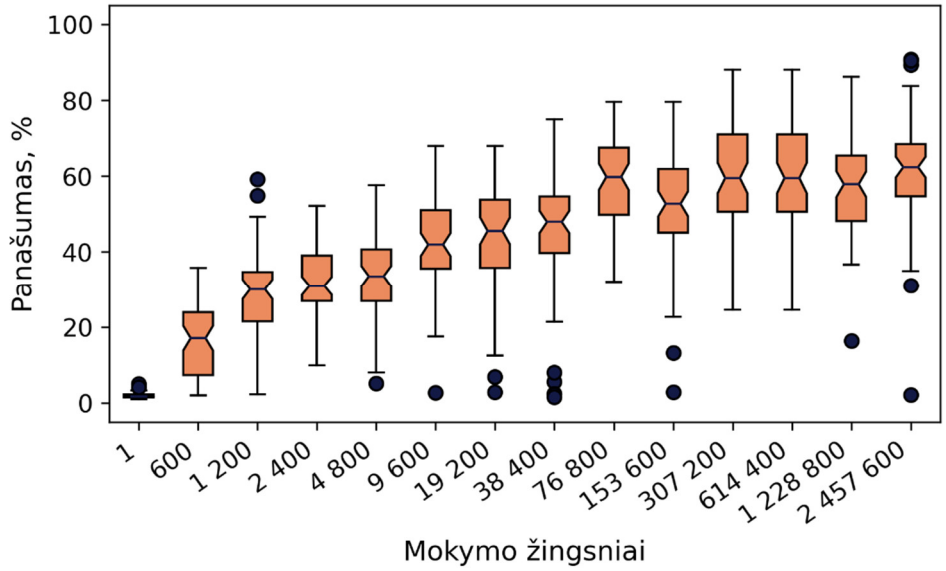
Naudojant tik reprezentatyvius klasterių atstovus būtų prarandama itin daug informacijos, kadangi mokymo rinkinys, lyginant su pradiniu, sumažėtų 14 kartų. Todėl GPT mokymas atliekamas iš kiekvieno klasterio atsitiktinai renkantis po vieną atstovą. Šis atsitiktinis atstovų

rinkimas yra vis kartojamas, todėl kiekvienas klasteris yra reprezentuojamas vienodai, kartu mokymui išnaudojant visas turimas sekas.

3.1.4. Modelio mokymas

Modelio mokymo metu sugeneruojamos sekos buvo įvertinamos pagal jų panašumą į natūralias mokymo rinkinio sekas. Toks palyginimas buvo atliekamas kas 1200 modelio mokymo žingsnių, atitinkamai kiekviename šio palyginimo taške naudojant 64 sugeneruotas sekas.

Nors modelio mokymo metu sekų panašumas gan anksti pasiekė maksimalią reikšmę, vien tai neleidžia daryti išvados, kad modelio mokymas galėjo būti stabdomas anksčiau (**3.3 pav.**). Didelis sekų panašumas gali būti atkurtas modeliui dar nespėjus išmokti sudėtingesnių sekų statistikų. Įprastai manoma, kad naudojant stabilią modelio architektūrą ir pakankamai didelius bei gerai normalizuotus duomenų rinkinius, ilgesnis tinklo mokymo laikas padeda gauti geresnius rezultatus. Tokiu atveju yra svarbu atskirti, kada ilgesnis mokymas atneša nykstamai mažą grąžą. Tai yra viena iš priežasčių, dėl kurios svarbu atlikti tinklo generuojamų duomenų analizę. Analizė padeda įvertinti modelio išmoktas duomenų rinkinio savybes bei surasti metrikas, pagal kurias gali būti vertinamas tinklo mokymas.



3.3 pav. Sugeneruotų sekų panašumo į natūralias sekas kitimas GPT mokymo metu.

Po 2,5 mln. žingsnių tinklo mokymas buvo sustabdytas. Mokymo pabaigoje GPT sugeneruotos sekos pasiekė vidutinį 60 % sugeneruotų sekų panašumą į mokymo rinkinio sekas. Vien pagal sekų panašumą tolimesnio modelio tobulėjimo įvertinti nebeįmanoma, o didesnis sekų tarpusavio panašumas nebūtinai atspindi sugeneruotų sekų kokybę. Dėl šių priežasčių yra reikalingi geresni įverčiai, kurie galėtų leisti kartu įvertinti sekų kokybę ir modelio tobulėjimą jo mokymosi metu.

3.2. GPT sugeneruotų sekų analizė

Malato dehidrogenazės sekomis apmokytas GPT panaudotas naujų sekų generavimui. Sekų generavime naudojamas vien tik generatorius, diskriminatorius šiuo atveju nenaudojamas. Pagal atsitiktinius įvesties vektorius sugeneruotos 20 tūkst. MDH sekų. Šios sekos toliau naudotos duomenų rinkinio analizėje ir palyginime.

3.2.1. Daugybinio sekų palyginio sudarymas

Didžiajai daliai sekų statistikų įvertinti yra reikalingi sekų palyginiai. Dėl generuojamų sekų ilgių įvairovės yra būtina sulygiuoti struktūriškai/funkciškai tarpusavyje atitinkančias sekų pozicijas. Kartu siekiant palyginti natūralias ir sugeneruotas sekas šios sulygiuojamos į bendrą sekų palyginį.

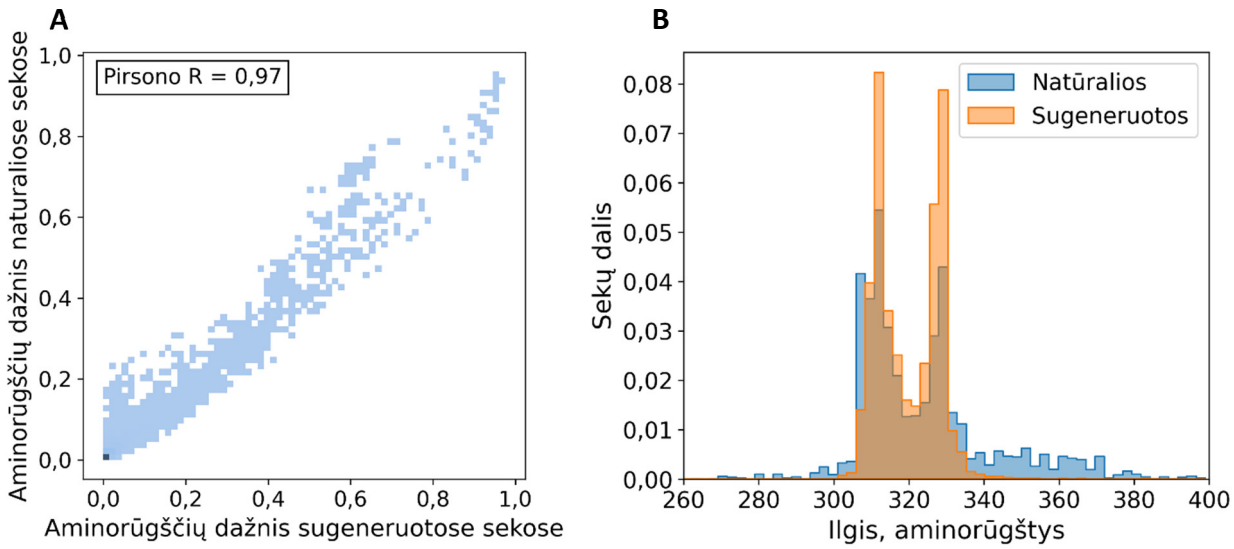
Daugybinio palyginio naudojimas analizėje įveda tam tikrą paklaidą. Nesutapimai gali atsirasti dėl neteisingai sudaryto palyginio (klaidos dėl palyginio sudarymui naudoto algoritmo), sekų generavimo atveju tai gali kilti ir dėl prastos generuojamų sekų kokybės ar dėl didelės jų įvairovės. Iš esmės šios problemos yra sunkiai išvengiamos, kadangi be DSP yra praktiškai neįmanoma palyginti specifinių pozicijų tarp sekų, o metodų leidžiančių vienareikšmiškai įvertinti DSP kokybę neegzistuoja. Siekiant neužgožti įvertinamų statistikų mažo padengimo pozicijomis, iš palyginio pašalintos pozicijos turinčios mažesnį nei 25 % padengimą abiejuose duomenų rinkiniuose.

Toliau aprašytoje analizėje naudotas normalizuotas mokymo sekų rinkinys. Normalizavimu siekta atkartoti natūralių sekų rinkinį tokia forma, kokia buvo naudota GPT mokymui. Sekos normalizuotos pagal mokymui naudotus 70 % sekų klasterius iš kiekvieno klasterio atsitiktinai imant po 21 seką. Tokiu principu gautas sekų rinkinys, kurio statistikos nėra užgožtos didžiųjų klasterių sekų.

3.2.2. Pirmos eilės statistikos

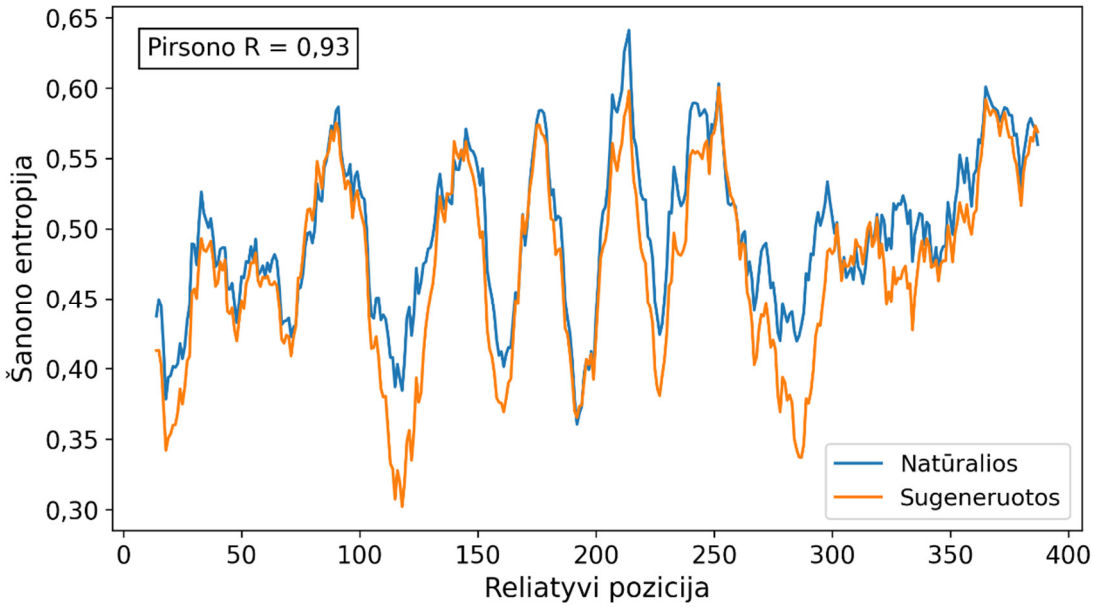
Pirmos eilės statistikos apibrėžia individualių pozicijų įverčius. Šias statistikas apibūdina ir gali atkartoti pagal profilį sukurti sekų rinkiniai, su sąlyga, kad yra įmanoma gauti pakankamai gerą sekų palyginį. GPT sugeneruotoms sekoms pozicinis aminorūgščių pasiskirstymas yra itin artimas natūraliose sekose matomam pasiskirstymui (**3.4A pav.**), tai atspindi ir itin aukštas[2] Pirsono koreliacijos koeficientas (Pirsono R = 0,97). Tarp sekų rinkinių taip pat išlaikytas panašus sekų ilgių pasiskirstymas (**3.4B pav.**).

[2] Koreliacijos lygis darbe įvardinamas remiantis (Hinkle ir kt., 2003)



3.4 pav. Pirmos eilės statistikų įverčiai. A, pozicinių aminorūgščių dažnių pasiskirstymo priklausomybė sekų rinkiniuose. B, natūralių ir sugeneruotų sekų ilgio pasiskirstymas.

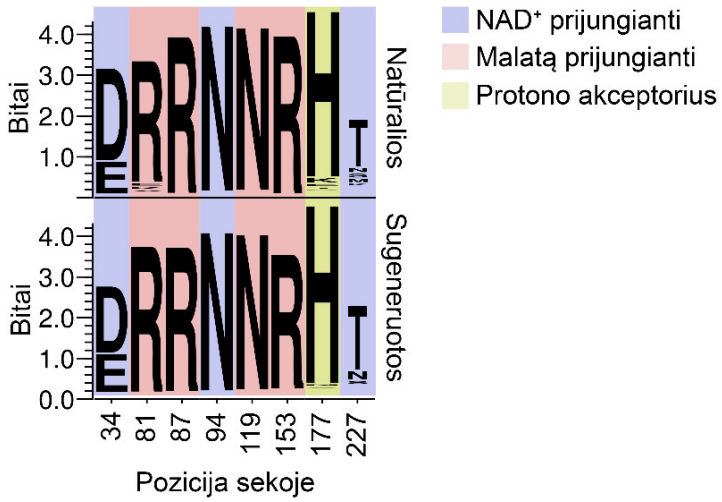
Pozicinė Šanono entropija šiuo atveju atvaizduoja aminorūgščių įvairovę duotojoje palyginio pozicijoje (**3.5 pav.**). Jei entropija būtų lygi 1, tai nurodytų vienodą visų galimų aminorūgščių pasiskirstymą pozicijoje. Jeigu pozicijoje dominuotų tik viena aminorūgštis, entropija būtų lygi 0. Entropija yra atvirkščiai proporcinga pozicijos konservatyvumui. Tarp sugeneruotų ir natūralių sekų Šanono entropijos pasiskirstymas išlieka itin panašus, tai matoma pagal aukštą koreliacijos koeficientą (Pirsono R = 0,93) ir žemą vidutinę absoliučią paklaidą (MAE = 0,053).



3.5 pav. Sugeneruotų ir natūralių sekų Šanono entropijos palyginimas. Atvaizdavimui naudotas slenkantis vidurkis (lango dydis 15).

Pozicijos su žema Šanono entropija nurodo ir aukštesnį konservatyvumą. 3.5 pav. grafike minimumai atitinka funkcines malato dehidrogenazės pozicijas. Šių pozicijų sekų logo matomas aukštas funkcinių aminorūgščių skirstinių panašumas (**3.6 pav.**). Toks rezultatas tikėtinas atsižvelgiant į aukštą pozicinių aminorūgščių skirstinių ir Šanono entropijos koreliaciją tarp sekų rinkinių.

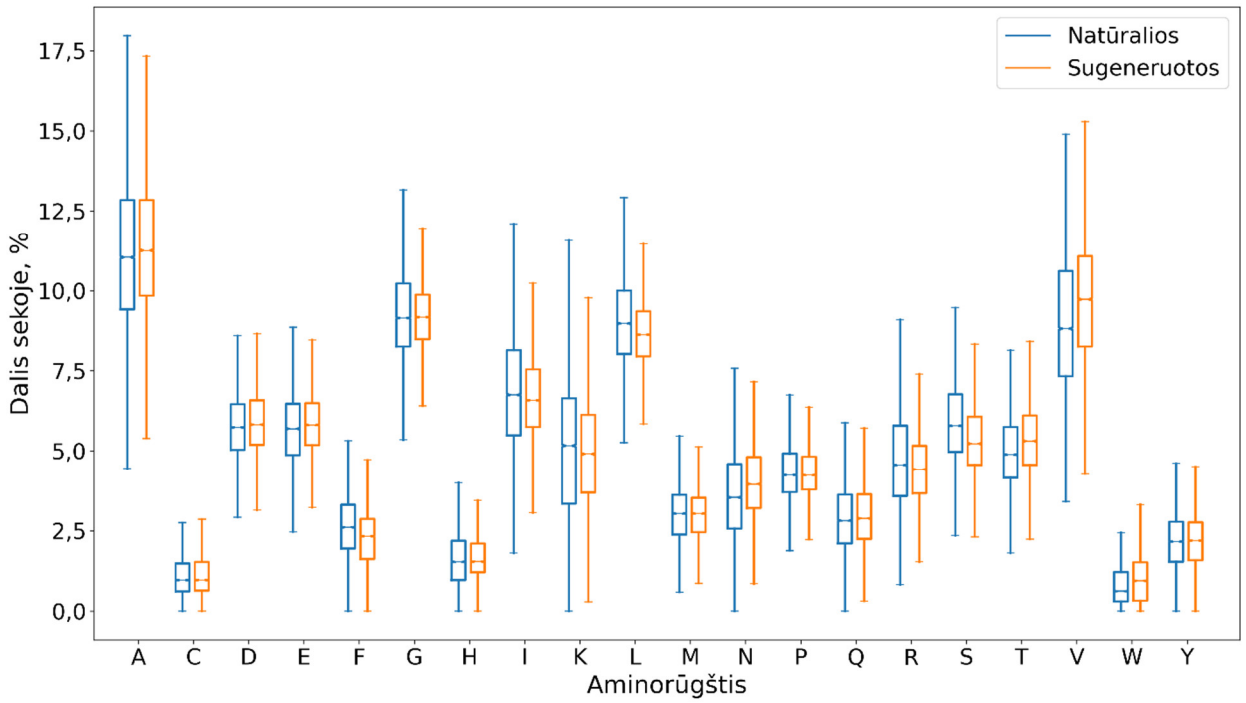
Sugeneruotos sekos pasižymi bendrai mažesne entropija. Iš 388 palyginio pozicijų 67 % pozicijų entropija sugeneruotų sekų rinkinyje yra žemesnė nei natūralių sekų rinkinyje. Tai nurodo, kad sugeneruotose sekose per poziciją matoma vidutiniškai mažesnė įvairovė. Šis dėsningumas matomas ir sekų logo pavyzdyje. 81, 177 ir 227 pozicijose mažai reprezentuotų aminorūgščių dalis dar labiau sumažėja sugeneruotose sekose (**3.6 pav.**).



3.6 pav. Funkcinių aminorūgščių dažnio pasiskirstymas natūraliuose ir sugeneruotuose sekų rinkiniuose atvaizduotas kaip sekų logo. Atvaizduotos ir atrinktos pozicijos atitinka *E. coli* MDH (UniProt ID: A1AGC9).

Sekų rinkinių panašumo be DSP įtakos įvertinimui nustatytas natūralių ir sugeneruotų sekų rinkinių aminorūgščių dažnių pasiskirstymas (**3.7 pav.**). Šiam palyginimui naudotos pilnos baltymų sekos. Kaip ir kitas pirmos eilės statistikas, GPT sugebėjo atkartoti bendrus aminorūgščių dažnius.

Sugeneruotų sekų rinkinyje beveik idealiai atkartojamos pirmos eilės baltymų sekų statistikos. Tai yra minimalus rezultatas, kurio galima tikėtis iš modelio, siekiančio atkurti mokymo rinkinio sekų statistikas. Kaip minėta, tokiems sekų variantams sukurti užtenka ir daug paprastesnių įrankių nei GPT.

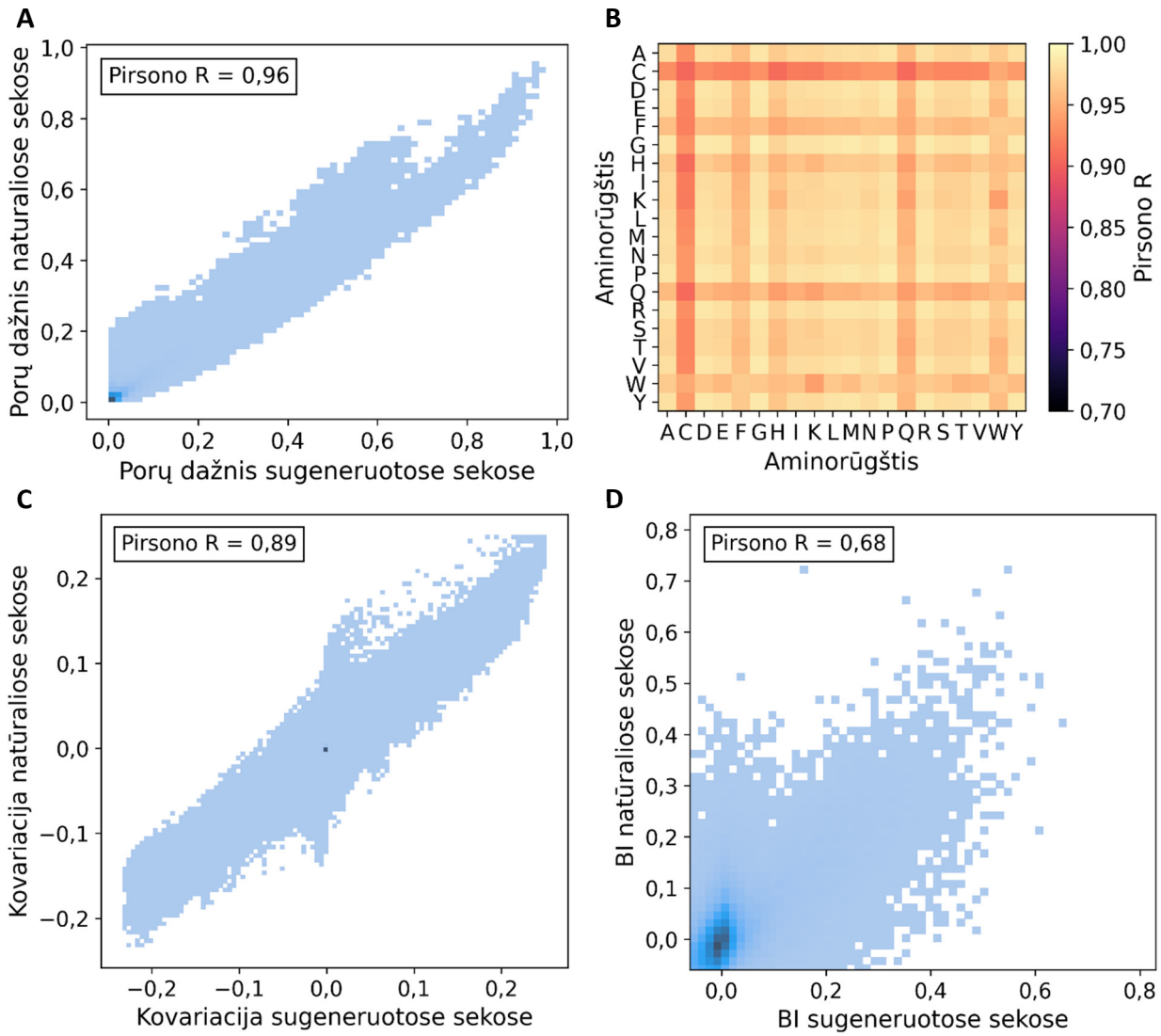


3.7 pav. Aminorūgščių dažnių pasiskirstymas sekų rinkiniuose. Aminorūgščių dažniai apskaičiuoti pagal visas sekų pozicijas.

3.2.3. Antros eilės statistikos

Antros eilės statistikomis yra apibrėžiamos statistikos, kurios priklauso nuo dviejų skirstinių (skirstinių pora gali būti tarpusavyje priklausoma arba nepriklausoma). Įvertinus aminorūgščių porų dažnius matomas itin aukštas skirstinių panašumas (**3.8A pav.**). Koreliacijos koeficiento vertė (Pirsono R = 0,96) išlieka beveik vienoda lyginant su poziciniu aminorūgščių dažniu. Šis įvertis kartu apskaičiuotas ne tik pozicijų poroms, bet ir specifinėms aminorūgščių poroms (**3.8B pav.**). Mažiausia koreliacija matoma tarp aminorūgščių porų sudaromų su triptofanu. Triptofanas kartu yra ir rečiausiai tarp natūralių MDH sekų pasitaikanti aminorūgštis (**3.7 pav.**).

Pirmos eilės dažnių panašumas gali nulemti ir antros eilės dažnių panašumą. Todėl būtina įvertinti ir normalizuotą porų dažnių pasiskirstymą, kuris nebūtų užgožtas pirmos eilės statistikų. Šiam tikslui pasiekti įvertinta aminorūgščių porų kovariacijos koreliacija (**3.8C pav.**). Pašalinus pozicinių skirstinių panašumo įtaką, porinių dažnių koreliacija sumažėja (Pirsono R = 0,89). Tai leidžia daryti prielaidą, kad modelis aminorūgščių kovariacijas išmoko blogiau nei pozicinius aminorūgščių dažnius ar bendrai pirmos eilės statistikas. Nepaisant to, sugeneruotų sekų rinkinyje porų koreliacijos atkurtos sąlyginai aukštu panašumo lygmeniu.

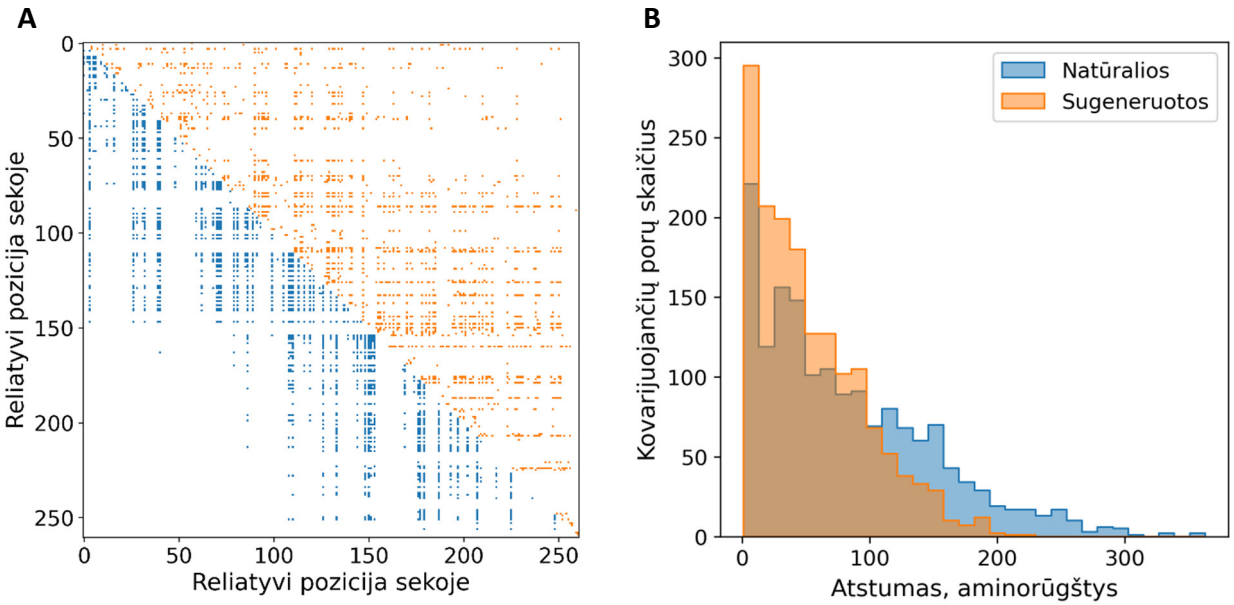


3.8 pav. Antros eilės statistikų įverčiai. A, porinių aminorūgščių dažinių pasiskirstymo priklausomybė tarp sekų rinkinių. B, Porinių aminorūgščių dažnių koreliacija atvaizduota specifinėms aminorūgščių poroms. C, Aminorūgščių porų kovariacijos priklausomybė tarp sekų rinkinių. D, bendros informacijos pasiskirstymo priklausomybė tarp sekų rinkinių.

Siekiant tarp sekų rinkinių nustatyti pozicijų porų įvairovę ir kovariacijos lygį naudotas bendros informacijos įvertis. Bendros informacijos panašumas tarp sekų rinkinių yra pastebimai žemesnis už prieš tai įvertintas statistikų koreliacijas (Pirsono R = 0,68) (**3.8D pav.**).

Įvertinant BI dažniausiai svarbiausiomis yra laikomos pozicijų poros, turinčios didžiausią įvertį. Lyginant *5L* (*L* – vidutinis MDH sekos ilgis) pozicijų poras, tarp natūralių ir sugeneruotų sekų sutampa 52,7 % pozicijų poros (**3.9A pav.**). Iš viso viename sekų rinkinyje esančių pozicijų porų skaičius yra 150544, todėl atsitiktinis sutapimas būtų lygus 1,1 %. BI žemėlapyje matoma, kad dalis tolimiausių porų yra neatkartota sugeneruotų sekų rinkinyje. Įvertinant BI porą sudarančių pozicijų atstumą sekoje matoma, kad natūralių sekų rinkinyje atstumas yra didesnis

nei sugeneruotų sekų rinkinio atveju (**3.9B pav.**). Tai nurodo ir atstumų vidurkis: 54 ir 83 aminorūgščių atstumas atitinkamai sugeneruotų ir natūralių sekų rinkiniams.



3.9 pav. A, 5L pozicijų porų su didžiausia BI žemėlapis. Vizualizavimui pasirinktos tik tos eilučių ir stulpelių poros, kurios atvaizduoja bent vieną tašką. B, 5L pozicijų poros su didžiausiu bendros informacijos įverčiu ir jų atstumas aminorūgščių sekoje.

Antros eilės statistikų įvertinimas leidžia atskirti sugeneruotą ir mokymo sekų rinkinius. Šiame kontekste matoma modelio silpnoji pusė. Dalis kovarijuojančių pozicijų ir specifinių aminorūgščių mokymo rinkinyje liko neišmoktos/nepastebėtos GPT. Šis skirtumas suteikia metriką, kuri leidžia toliau tobulinti GPT fokusuojantis ne į bendrą sekų panašumą, o į antros eilės statistikų pagerinimą.

3.2.4. Sekų struktūrinio atitikimo įvertinimas

Sugeneruotų sekų struktūriniam įverčiui gauti naudotas trRosetta neuroninio tinklo modelis (J. Yang ir kt., 2020). Šio tinklo spėjimais paremti struktūriniai modeliai gerai įvertinti CASP konkurse, be to, jis sėkmingai panaudotas *de novo* baltymų dizaine kuriant naujas sekas pagal duotąją atstumų matricą (Anishchenko ir kt., 2020; Norn ir kt., 2020). Tokie rezultatai leidžia tikėtis, kad šio tinklo įverčiai gebės atspindėti sugeneruotų sekų struktūrinį atitikimą pagal pateiktą baltymo karkasą (angl. *backbone*). Didelis šio tinklo privalumas yra ir tai, kad jam nereikia pateikti pilno struktūrinio modelio, užtenka tik baltymo kontaktų žemėlapio. Šiuo atveju tai reiškia, kad išvengiama paklaidų, kurios galėtų atsirasti modeliuojant pilną struktūrą pagal pateiktus sekų variantus.

Sekų variantų įvertinimui naudoti trRosetta tinklo įverčiai atitinkantys tinklo nuostolių funkcijos rezultatą. Kuo tinklo nuostoliai mažesni, tuo sekos ir struktūros pora tinklui atrodo

tinkamesnė ir atvirkščiai. Kadangi toks sekų įvertinimas literatūroje nėra aprašytas į palyginimą įtraukiami ir kontroliniai variantai.

Įvertinimui naudoti keturi sekų rinkiniai: GPT sugeneruotos sekos, mokymo rinkinio sekos, sekos sugeneruotos pagal mokymo sekų PSM ir atsitiktinai mutuoti sekų variantai. Šiam palyginimui pasirinktas *E. coli* MDH struktūrinis modelis (PDB ID: 6KA1) ir jį atitinkanti baltymo seka. Tarp sugeneruotų sekų surasti sekų variantai pasižymintys kiek įmanoma mažesniu insercijų ir delecijų skaičiumi poriniame sekų palyginyje. Iš surastų sekų variantų pašalintos insercijos, o delecijos užpildytos atitinkamuose stulpeliuose esančiomis taikinio aminorūgštimis. Taip gautas sekų palyginys, kurio stulpelių skaičius atitinka taikinio aminorūgščių skaičių, ir visi sekų variantai neturi nei delecijų nei insercijų. Tokia sekų paieška/modifikavimas taikytas sugeneruotoms ir mokymo sekoms. Tai taikoma tam, kad nebūtų reikalingas struktūros modifikavimas įterpiant atitinkamas insercijas/delecijas, dėl ko gali atsirasti modeliavimo paklaidos.

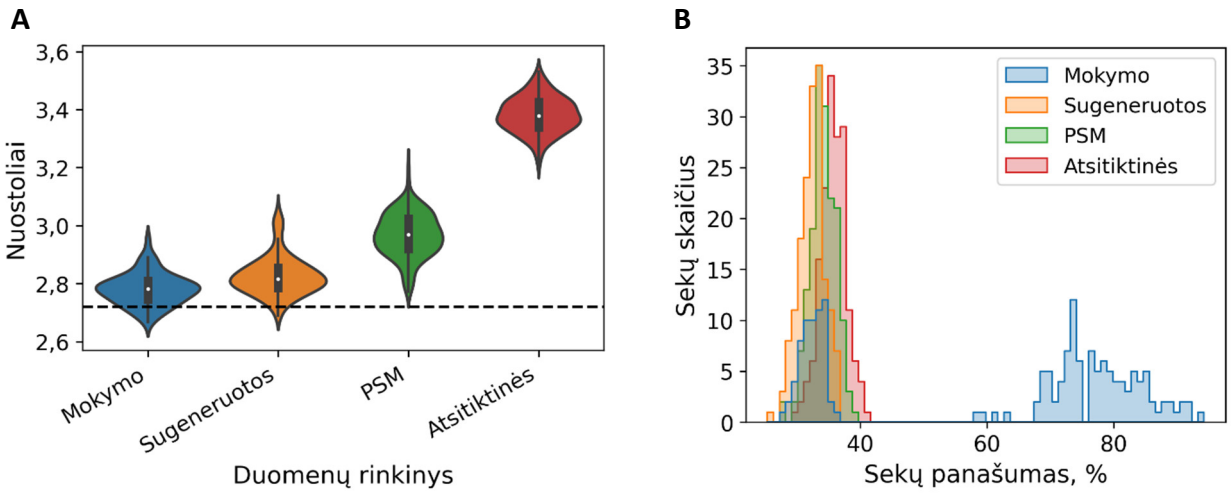
Siekiant turėti atskaitos tašką, palyginant sugeneruotų sekų struktūrinį atitikimą, simuliuojami du scenarijai:

- Teigiamas. Atitinkantis natūralių sekų gaunamus įverčius.
- Neigiamas. Atitinkantis atsitiktinai arba pagal PSM sukurtų sekų įverčius.

Kadangi natūralių sekų rinkinyje nemaža dalis sekų yra itin panaši į taikinio seką (**3.10B pav.**), sekų panašumui normalizuoti iš visų sekų rinkinių pašalintos sekos panašios į taikinio seką daugiau nei 50 %.

Sekų įvertinimai pagal trRosetta tinklo nuostolius pateikti 3.10A pav. Tarp kontrolinių variantų matomas tolygus rezultatų pasiskirstymas atitinkantis: natūralios > PSM > atsitiktinės. Tokio rezultato galima tikėtis atsižvelgiant į šių sekų kilmę. Visų tirtų sekų rinkinių vidurkiai tarpusavyje reikšmingai skiriasi (Tjukio testas $p<0,05$).

Sugeneruotas sekų rinkinys įvertintas geriau nei pagal PSM sukurti variantai, tačiau blogiau nei natūralios sekos (vidutinis įvertis atitinkamai natūralioms, sugeneruotoms ir PSM sekoms: 2,78, 2,83, 2,97). Kadangi pagal PSM sukurtos sekos turėtų visiškai atkurti pirmos eilės statistikas, aukštesnį įvertį galėjo lemti PSM sekų rinkinyje trūkstamos antros eilės statistikos. Tai patvirtinti arba paneigti sunku, nes mokslinių darbų apie trRosetta įverčio priklausomybę nuo šių savybių šiuo metu nėra.



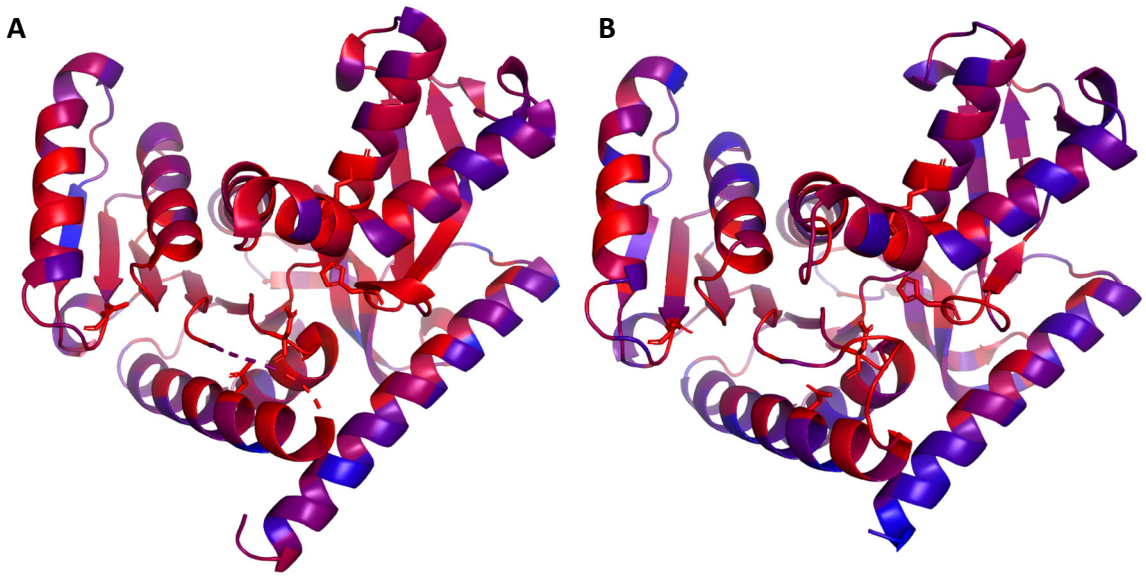
3.10 pav. A, trRosetta įverčių pasiskirstymas pavaizduotas kaip branduolio tankio įvertis (angl. *Kernel Density Estimation*). Juoda punktyrinė linija atitinka taikinio sekos įvertį. Mažesni nuostoliai atitinka geresnį įvertį. B, Sekų panašumo į taikinio seką (PDB ID: 6KA1) pasiskirstymas prieš normalizavimą. Po normalizavimo pašalinti sekų variantai panašesni į taikinį daugiau nei 50 %.

Eksperimentiškai parodyta, kad pagal trRosetta modelį ir pasirinktą baltymo struktūrą optimizuotos baltymo sekos susilanksto ir įgauną numatytą struktūrą. Praktiškai ši optimizacija yra atliekama pagal trRosetta nuostolius – vykdoma jų minimizacija. Tai leidžia teigti, kad sekų variantai pasižymintys nuostoliais panašiais arba mažesniais už taikinio seką atitinka ir jo struktūrą. Iš šioje dalyje tirtų 165 GPT sugeneruotų sekų variantų 4 buvo įvertinti žemesniu nei taikinys trRosetta įverčiu.

3.2.5. Nuo duomenų rinkinio nepriklausomas konservatyvumo atkūrimas

Sugeneruotų sekų subrinkinys, pagal kurį nustatyti trRosetta įverčiai, panaudotas ir konservatyvumo įvertinimui. Šiam įvertinimui naudotas neapdorotas 165 GPT sekų subrinkinys (nepašalintos insercijos ir neužpildytos delecijos). Siekiant nepriklausomai nustatyti natūralių sekų pozicijų konservatyvumą, konservatyvumas įvertintas ne pagal mokymo rinkinio sekas, bet pagal nepriklausomai surinktą taikinio homologų rinkinį. Šiam tikslui panaudotas ConSurf serveris (Ashkenazy ir kt., 2016).

Kaip sugeneruotų sekų rinkinio taikinys (seka pagal kurią pasirenkamos pozicijos konservatyvumui nustatyti), pasirinkta pirmoji pagal eilės numerį sugeneruota seka iš sugeneruotų sekų subrinkinio (sekos ID 29). Šiai sekai nustatytas artimiausias struktūrinis homologas PDB70 duomenų bazėje, tai šiuo atveju atitiko *Salinibacter ruber* malato dehidrogenazės seką (PDB ID 3NEP). *S. ruber* MDH seka pasirinkta ir kaip natūralių sekų taikinio seka pagal kurią ConSurf serveryje nustatytas konservatyvumas.



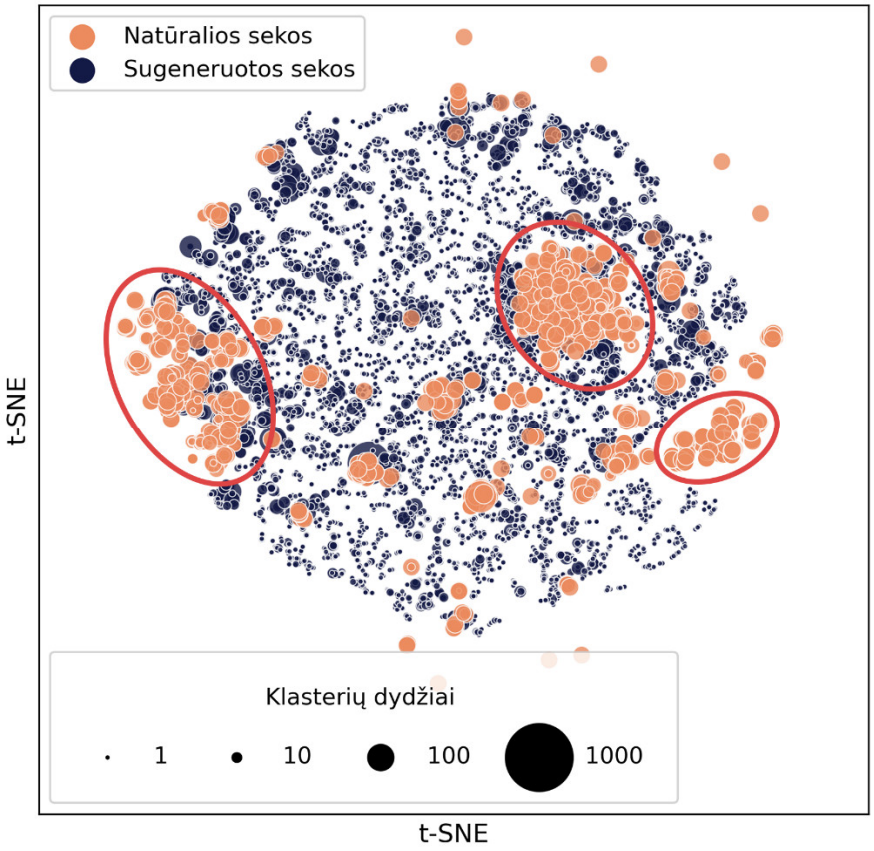
3.11 pav. A, *Salinibacter ruber* MDH struktūra (PDB ID 3NEP), spalvų gradientas atitinka konservatyvumą apskaičiuotą pagal ConSurf serverį. B, sugeneruotos sekos 29 homologijos modelis, spalvų gradientas atitinka konservatyvumą įvertintą Šanono entropija pagal sugeneruotų sekų subrinkinio palyginį. Raudona spalva atitinka didesnį konservatyvumą, mėlyną – mažesnį. Pavaizduotos šoninės grandinės vaizduoja funkcines pozicijas (pagal UniProt ID Q2S289).

3.11 pav. matomas vizualinis konservatyvumo palyginimas. Tiek vizualiai, tiek ir pagal koreliacijos koeficientą (Spirmano ρ = 0,88) galima matyti aukštą konservatyvumų panašumą tarp abiejų sekų rinkinių. Pagal abu metodus funkcinės aminorūgštys buvo įvertintos kaip itin konservatyvios. Sekų konservatyvumas GPT sugeneruotose sekose yra atkuriamas tiek ir bendrai pilnam sekų rinkiniui, tiek ir mažesniems sekų rinkiniams/klasteriams.

3.2.6. Sekų pasiskirstymas erdvėje ir sekų įvairovė

Sekų generavimo uždavinyje yra svarbu ne tik gebėti atkartoti įvairias mokymo rinkinio savybes, bet ir sugebėti generalizuoti pateiktus mokymo duomenis. Generalizavimas leidžia generuoti naujus sekų variantus su natūralioms sekoms būdingais bruožais ar įvertinti jau egzistuojančius variantus. Sekų įvairovės ir jų pasiskirstymo vizualizavimui panaudotas t-SNE dimensijų sumažinimo metodas. t-SNE metodu siekiama atvaizduoti taškus dvimatėje erdvėje, kuo tiksliau išsaugant kiekvieno taško kaimynus.

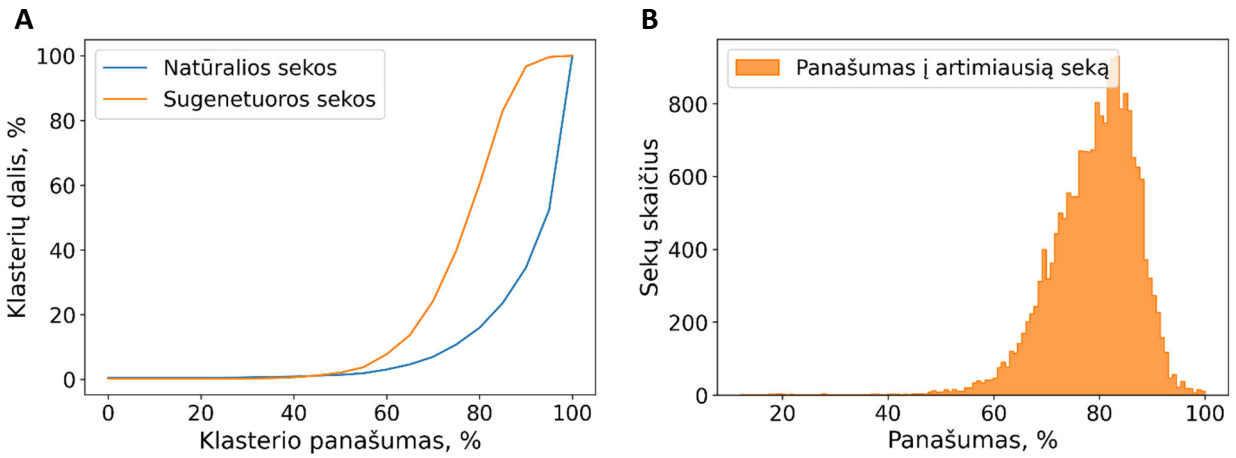
Atvaizdavimui naudojamas normalizuotas mokymo sekų rinkinys, tai atvaizduoja GPT matomą sekų pasiskirstymą. Natūralios sekos erdvėje sudaro sekų superklasterius (klasterius sudarytus iš klasterių) (**3.12 pav.**). Tokius galima išskirti tris. Šie klasteriai skiriasi padėtimi erdvėje ir sugeneruotų sekų apsuptimi. Tai galimai susiję su šių superklasterių dydžiu ir jų reprezentatyvumu mokymo rinkinyje: mažesni superklasteriai mokymo metu atitinkamai rečiau pateikiami tinklui. To galimai būtų galima išvengti normalizavimą atliekant keliais sekų panašumo lygiais arba sekas normalizuojant joms priskiriant svorius pagal jų kaimynų skaičių sekų erdvėje.



3.12 pav. Natūralių ir sugeneruotų sekų pasiskirstymas t-SNE sekų erdvėje. Raudonai apibrėžti didžiausi superklasteriai.

Sugeneruotos sekos t-SNE atvaizduotoje erdvėje pasižymi tolygesniu pasiskirstymu ir nesudaro itin didelių klasterių. Visgi, didesnieji sugeneruotų sekų klasteriai yra pasiskirstę šalia natūralių sekų, o mažesnieji užpildo erdvę tarp natūralių sekų superklasterių.

Darant prielaidą, kad 20000 sugeneruotų sekų sudarė visą galimą sugeneruotų sekų erdvės įvairovę, sugeneruotose sekos pasižymi iki 3,8 karto didesne įvairove nei natūralios sekos, priklausomai nuo pasirinkto panašumo įvertinimo procento (įvairovę vertinant kaip klasterių skaičių santykį) (**3.13A pav.**). Tikėtina, kad absoliutus skaičius galimų sugeneruoti unikalių sekų yra žymiai didesnis nei 20000. Nors šios sekos ir būtų unikalios, jos pasižymėtų ir vis didesniu tarpusavio panašumu. T.y. generuojant didesnį skaičių sekų didėtų reprezentatyvių klasterių dydis, o ne pačių klasterių skaičius. Sugeneruotos sekos pasižymi aukštu unikalumo lygiu, tik maža dalis sekų turi didesnį nei 95 % tarpusavio panašumą (**3.13B pav.**). Vidutinis sugeneruotų sekų tarpusavio panašumas siekia 79 %.



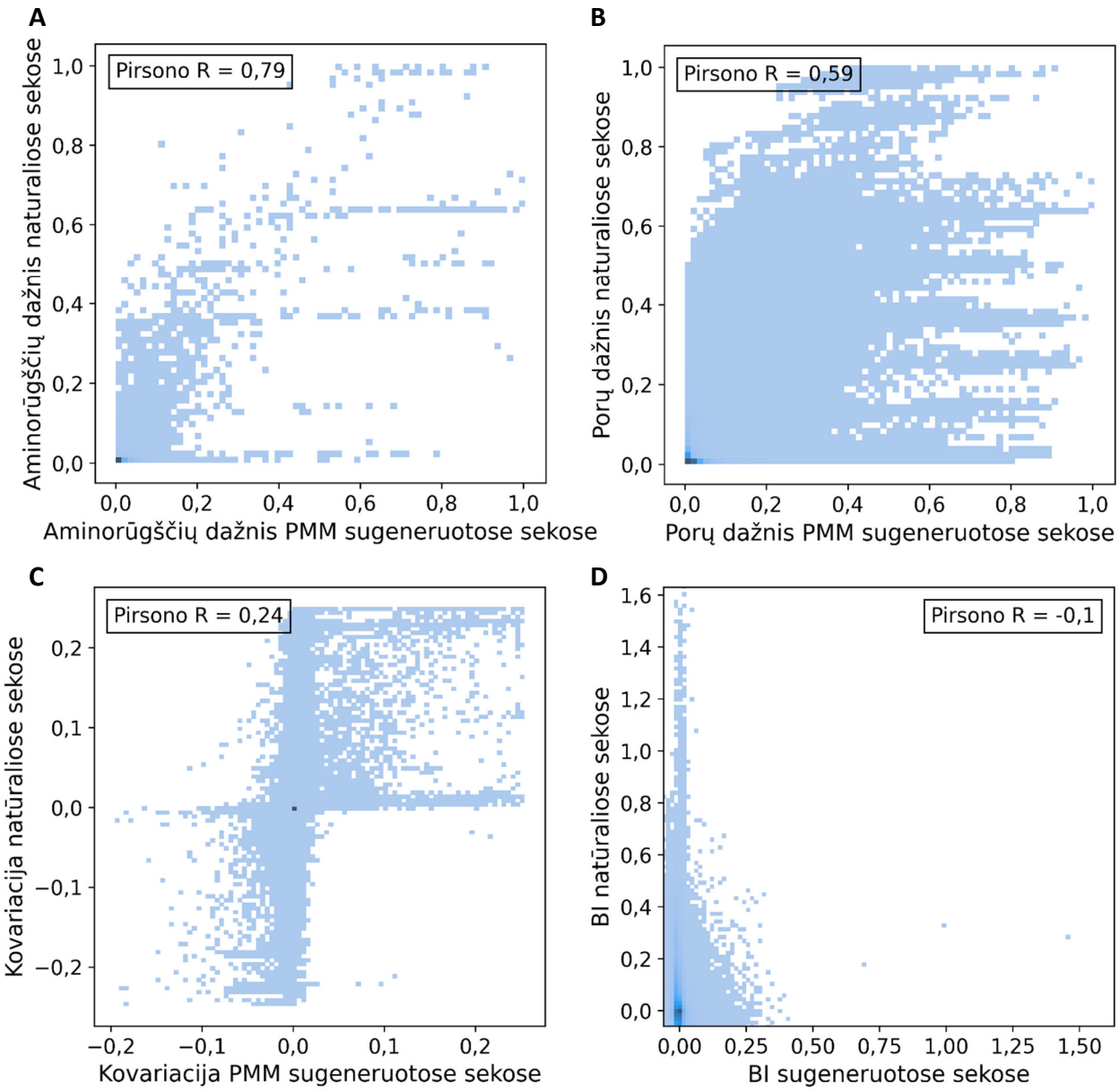
3.13 pav. A, sekų įvairovė įvertinta kaip klasterių skaičius ties atitinkama klasterizavimo riba. B, sugeneruotų sekų rinkinio artimiausių sekų porų tarpusavio panašumo pasiskirstymas.

3.2.7. Sekų generavimas pagal PMM profilį

Siekiant palyginti GPT sugeneruotas sekas su paprastesniu generatyviniu modeliu, pasirinktas generavimas pagal PMM profilį. Pagal normalizuotą mokymo sekų rinkinį sudarytas PMM profilis, pagal kurį toliau generuotos baltymų sekos. PMM profilis išsaugo pozicinį palyginio pasiskirstymą: kiekviena pozicija PMM profilyje turi tikimybę, kad sekanti būsena bus insercija, delecija arba aminorūgštis, o kiekviena aminorūgšties būsena savyje turi tai pozicijai specifinį aminorūgščių dažnį. Dėl šių PMM savybių iš anksto yra keliama hipotezė, kad PMM gebės atkurti pozicinį aminorūgščių pasiskirstymą, tačiau dėl profilio techninių apribojimų, aukštesnės eilės statistikos atkūrimas bus ribotas.

PMM sugeneruotos sekos pasižymi mažesniais koreliacijos koeficiento įverčiais nei GPT sugeneruotos sekos, tai būdinga pirmos ir antros eilės statistikoms. Kaip ir buvo tikėtasi PMM sugebėjo dalinai atkurti pirmos eilės statistikas, tačiau aukštesnės eilės statistikos atkuriamos tik iki mažo panašumo arba išvis neatkuriamos (**3.14 pav.**).

Neatitikimai pirmos eilės statistikose galėjo būti sukelti bendro PMM sugeneruotų sekų ir mokymo rinkinio sekų palyginio sudarymo klaidų. Dėl mažo PMM sugeneruotų sekų panašumo į natūralių sekų rinkinį gautas bendrai blogesnės kokybės palyginys, tai tiesiogiai nulemia blogesnius statistinius įverčius.



3.14 pav. Profilio PMM sugeneruotų sekų statistikos. A, pozicinių aminorūgščių dažnių pasiskirstymo priklausomybė sekų rinkiniuose. B, porinių aminorūgščių dažinių pasiskirstymo priklausomybė natūraliose ir PMM sugeneruotose sekose. C, Aminorūgščių porų kovariacijos priklausomybė tarp PMM sugeneruotų ir natūralių sekų rinkinių. D, bendros informacijos pasiskirstymo priklausomybė tarp pozicijų porų natūraliose ir PMM sugeneruotose sekose.

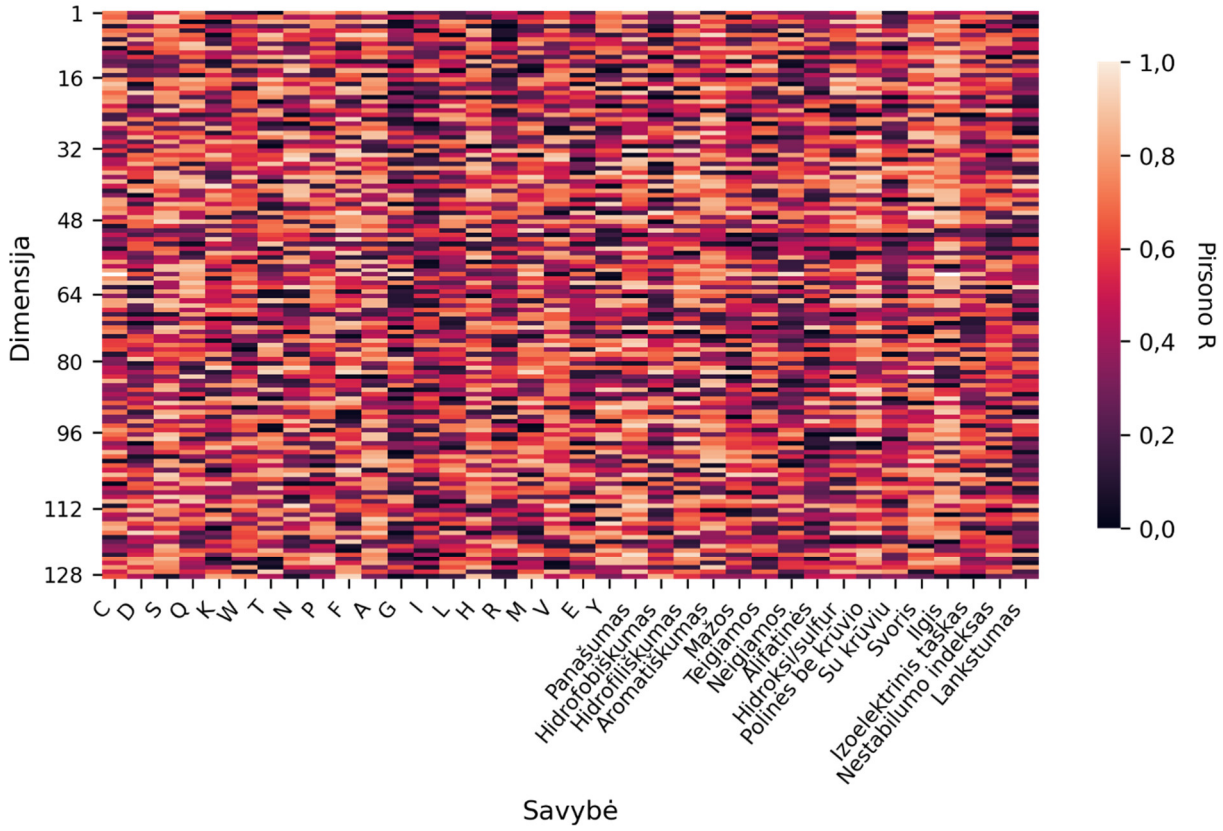
3.2.8. Latentinė erdvė

Sekų savybių priklausomybei nuo įvesties vektoriaus nustatyti kiekviena įvesties vektoriaus dimensija tolygiai keičiama nuo -1 iki 1, taip kiekvienai dimensijai sugeneruojant po 1024 sekas. Ieškant priklausomybės tarp įvairių baltymų sekų savybių ir įvesties vektoriaus reikšmių kiekvienai sekai apskaičiuotos šios savybės:

- Atskirų aminorūgščių dažnis sekoje;

- Atskirų aminorūgščių grupių dažnis sekoje (hidrofobinės, hidrofilinės, aromatinės, mažos, teigiamos, neigiamos, alifatinės, su hidroksi/tioline grupe, polinės neįkrautos, įkrautos);
- Bendros baltymų savybės (ilgis, svoris, izoelektrinis taškas, nestabilumo indeksas, lankstumo indeksas) (Guruprasad ir kt., 1990; Vihinen ir kt., 1994);
- Panašumas į mokymo sekas.

Įvertinus koreliaciją tarp įvesties vektoriaus dimensijų reikšmės ir savybių, išvardintų aukščiau, nustatyta, kad 123 dimensijos stipriai koreliuoja (Pirsono R > 0,8) bent su viena savybe (**3.15 pav.**). Didžiausias koreliacijų skaičius nustatytas su baltymo sekos ilgiu. Mažiau nei penkios koreliacijos nustatytos tarp dimensijų vertės ir izoleucino dažnio, įkrautų aminorūgščių grupės dažnio ir nestabilumo indekso.

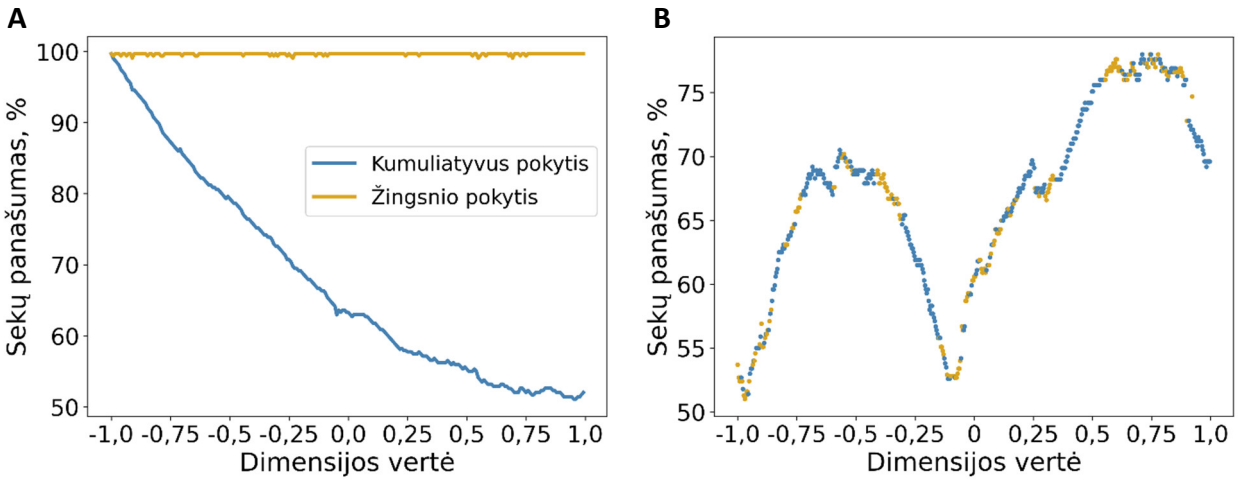


3.15 pav. Sekų savybių priklausomybė nuo įvesties vektoriaus dimensijų verčių įvertinta Pirsono koreliacijos koeficientu. Pavienės aminorūgštys nurodo aminorūgščių dažnį.

Aukšta koreliacija tarp dimensijų reikšmės ir įvairių baltymo savybių pokyčio leidžia tikėtis, kad pasirinktos baltymo savybės gali būti tikslingai keičiamos pagal įvesties vektoriaus dimensijų vertes. Keičiant individualias dimensijas apibrėžtose ribose maksimaliai pasiektas nuo 3 iki 10 % pavienių aminorūgščių dažnio pasikeitimas. Atskirų aminorūgščių grupių dažnis pakeičiamas iki 15 %, baltymo ilgis maksimaliai pakeičiamas 23 aminorūgštimis (**1 priedas**). Ši variacija pasiekta keičiant tik po vieną įvesties vektoriaus dimensiją. Siekiant įvesti į sekas

didesnius pokyčius reikia atrasti specifines kryptis, kai įvesties vektorius modifikuojamas keičiant ne vieną, o kelias dimensijas vienu metu (Tran et al., 2019).

Sekų panašumo pokyčiui keičiantis dimensijos vertei nustatyti apskaičiuotas sekų tarpusavio panašumas **(3.16A pav.)**. Pagal eilę $i-1$ ir i einančios sekos tarpusavyje skiriasi tik keliomis aminorūgštimis (žingsnio pokytis), tačiau tolygiai keičiantis dimensijos vertei matomas ir tolydus sekos kitimas lyginant su pirmąją seka eilėje. Pokyčiams susidedant sekų panašumas mažėja iki apytiksliai 50 % tarpusavio panašumo. Tolimesnis dimensijos pokytis tiesiogiai nelemia kumuliatyvaus sekų panašumo pokyčio, yra stebimi tik lokalūs panašumo sumažėjimai ir padidėjimai.



3.16 pav. A, sekų panašumo kitimas tolygiai keičiantis dimensijos vertei. Žingsnio pokytis nurodo dviejų i ir $i-1$ pagal eilę einančių sekų panašumą. Kumuliatyvus pokytis nurodo pirmos pagal eilę sekos panašumą į seką i. B, sekų panašumo į mokymo rinkinio sekas kitimas keičiantis dimensijos reikšmei. Spalvos pasikeitimas nurodo, kad sugeneruota seka yra panaši į kitą mokymo seką nei prieš tai buvusi sugeneruota seka.

Toliau įvertintas sugeneruotų sekų panašumas į mokymo sekas ir šio panašumo kitimas. Kiekvienai sugeneruotai sekai apskaičiuotas panašumas į jai artimiausią seką iš tinklo mokymui naudoto sekų rinkinio. Analizuojant šį panašumo pokytį keičiantis dimensijos reikšmei pastebimas pastovus panašumo kitimas ir artimiausios natūralios sekos pasikeitimas (**3.16B pav.**). Tai leidžia daryti išvadą, kad tinklas geba interpoliuoti tarp mokymo sekų, o šiam tikslui naudojami mokymo duomenyse aptinkami sekų elementai. Ši savybė turėtų leisti kombinuoti įvairias mokymo rinkinyje matomas sekas, idealiu atveju gauti sekų variantus, kurie pasižymėtų šioms sekoms bendromis savybėmis.

3.2.9. MDH sekų variantų aktyvumo įvertinimas

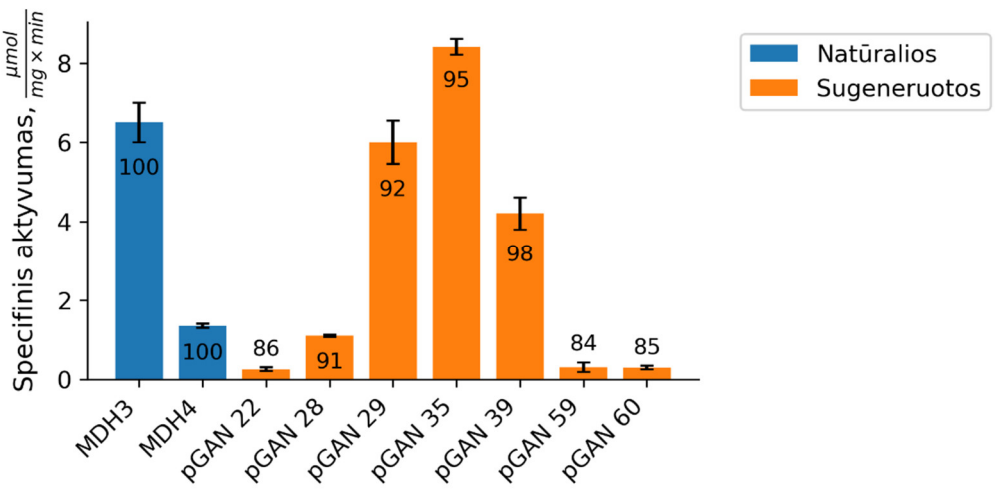
GPT sugeneruotų sekų aktyvumas *in vitro* įvertintas 55 skirtingiems sugeneruotų sekų variantams. Tirtos sekos į artimiausias mokymo rinkinio sekas panašios nuo 45 % iki 98 %. Toks

panašumo lygis atitinka nuo 7 iki 157 aminorūgščių pakeitimų lyginant su artimiausiais natūralių sekų variantais.

Baltymų gryninimą ir aktyvumo matavimus atliko:

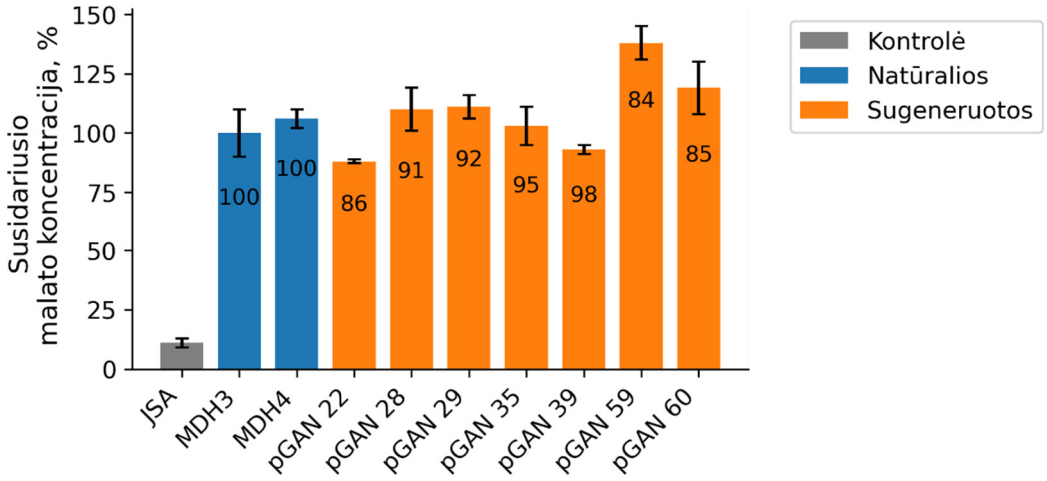
- Simona Povilonienė. Biochemijos institutas, Gyvybės mokslų centras, Vilniaus universitetas;
- Elzbieta Rembeza. Biologijos ir biologinės inžinerijos skyrius, Chalmers technologijų universitetas.

Iš 60 pradinių variantų, 55 buvo sėkmingai susintetinti ir įklonuoti į raiškos vektorių. Siekiant surasti kiek įmanoma daugiau tirpių variantų, naudoti du skirtingi *E. coli* kamienai ir du skirtingi baltymų raiškos ir gryninimo būdai (metodas 1 ir metodas 2). Pirmo metodo atveju nustatyta 14 tirpių baltymo variantų, iš kurių 11 pasižymėjo kataliziniu aktyvumu. Antro metodo atveju nustatyta 19 tirpių MDH variantų, iš kurių 15 pasižymėjo kataliziniu aktyvumu. Bendrai, įtraukiant variantus, kurie buvo tirpūs/aktyvūs naudojant bent vieną iš metodų, nustatyta 19 tirpių ir 13 aktyvių variantų (atitinkamai 35 % ir 24 % iš visų testuotų variantų). Mažiausias aktyvios MDH sekos panašumas į natūralią seką siekė 66 % (**2 priedas**). Tarp sugeneruotų sekų panašumo į natūralias sekas ir tirpumo nustatyta vidutinė koreliacija (taškinis dviserijinis koreliacijos koeficientas (angl. *point biserial correlation*) R_{pb} = 0,50).



3.17 pav. Specifinis aktyvumas natūraliems ir sugeneruotiems MDH variantams. Skaičiai grafike nurodo panašumą į artimiausią natūralią seką. Atvaizduoti tik tie variantai, kurių koncentracija buvo tiksliai nustatyta. Paklaidos apskaičiuotos kaip trijų pakartojimų standartinė paklaida.

Aktyvūs sugeneruoti MDH variantai pasižymėjo panašiu arba mažesniu specifiniu aktyvumu, lyginant juos su natūraliais sekų variantais (**3.17 pav.**). Nepaisant to, galutinė susidariusio malato koncentracija išliko panaši visais atvejais (**3.18 pav.**).



3.18 pav. Fermentinės oksaloacetato redukcijos metu susidariusio malato koncentracija reliatyviai MDH3 fermento atžvilgiu. Skaičiai grafike nurodo panašumą į artimiausią natūralią seką. Atvaizduoti tik tie variantai, kurių koncentracija buvo tiksliai nustatyta. Paklaidos apskaičiuotos kaip trijų pakartojimų standartinė paklaida.

4. DISKUSIJA

4.1. Sekų statistikos

Darbe tirti GPT geba atkartoti pozicinius ir tarppozicinius aminorūgščių dažnius bei pozicijų kovariacijas. GPT išmoktos kovariacijos išryškėja lyginant jas su kovariacijomis nustatytomis PMM sugeneruotų sekų rinkinyje (**3.8 pav.** ir **3.14 pav.**). Remiantis sugeneruotų sekų entropijos įverčiais nustatyta dalinai sumažėjusi pozicinė aminorūgščių įvairovė. Mažesnė pozicinė įvairovė yra nulemta retai natūralių sekų rinkinyje pasitaikančių aminorūgščių dalies sumažėjimu. Taigi, GPT nėra linkę didinti pozicinės aminorūgščių įvairovės.

GPT generuojamos sekos itin aukštu lygiu atkartoja baltymų konservatyvumą. Konservatyvumo atkūrimas yra matomas ir tarp pilnų sekų rinkinių, ir tarp mažesnių sekų subrinkinių. Aukštas konservatyvumo ir pozicinio aminorūgščių dažnio atkūrimas padeda užtikrinti itin svarbių ir baltymų šeimai bendrų aminorūgščių atkūrimą. Tai gali būti katalizinės, substratą ar kofaktorių prijungiančios, oligomerizacijoje dalyvaujančios aminorūgštys. Šios aminorūgštys įprastai yra būtinos baltymo funkcionalumui palaikyti, todėl aukštas jų atkūrimo lygis yra itin svarbus.

Sugeneruotame sekų rinkinyje matomos pozicijos su žemesniu BI įverčiu. Panašu, kad GPT gebėjimas išmokti pozicijų kovariaciją yra limituotas dviejų aminorūgščių atstumo sekoje. MDH atveju efektyvus atstumas sugeneruotų sekų rinkinyje ties kuriuo dar aptinkamas reikšmingas skaičius kovarijuojančių pozicijų porų yra apie 200 aminorūgščių. Pagal įvertintas sekų rinkinio statistikas šiuo metu tai yra didžiausia tinklo problema. Ji gali tapti dar labiau pastebima pasirinkus didesnius taikinius nei MDH ir jos gali praktiškai nelikti naudojant mažesnius nei 150–200 aminorūgščių ilgio baltymus. Visgi, šiuo metu tolimų kovariacijų atkūrimas yra pagrindinis GPT tobulinimo kriterijus.

Įprasta *de novo* baltymų dizaino dalis yra sukurti sekas, kurios biologinėmis sąlygomis įgautų nustatytą struktūrą (Korendovych ir DeGrado, 2020). Šios savybės įvertinimas *in silico* metodais vis dar išlieka itin sudėtingas. Baltymo karkaso užpildymas specifinėmis aminorūgštimis dažnai validuojamas molekulinės dinamikos simuliacijomis arba *ab initio* struktūros nuspėjimu (Carvalho ir kt., 2017; Leaver-Fay ir kt., 2011; Marcos ir kt., 2017). Abu šie metodai yra sąlyginai lėti, be to, sėkmingas jų panaudojimas dažniausiai galimas tik mažiems baltymams. Ši paradigma iš esmės pradeda keistis MM metodų pagalba. Šioje tyrimų sferoje labiausiai pasižymi trRosetta MM modelis. Pasirodo vis daugiau sėkmingų šio modelio pritaikymo būdų. Su šiuo darbu labiausiai susiję pavydžiai yra baltymų struktūrų nuspėjimas, baltymų sekų dizainas ir baltymo energetinio minimumo įvertinimas (Anishchenko ir kt., 2020; J. Yang ir kt., 2020; Norn ir kt., 2021). Kiek kitaip nei minėtuose darbuose, šiuo atveju yra įvertinama eilė bendros kilmės sekų

variantų nevykdant papildomos jų optimizacijos. Dėl šio pritaikymo unikalumo kartu įtraukti sąlyginai paprasti, tačiau lengvai interpretuojami kontroliniai variantai: atsitiktinėmis aminorūgštimis ir pagal PSM užpildytos sekos. Atsitiktinių sekų trRosetta įverčiai[3] aiškiai nurodo, kad neatkartojant nei pozicinių, nei tarppozicinių aminorūgščių statistikų, pasiekiamas itin mažas struktūrinis sekų atitikimas. Atkartojant pozicinius sekų dažnius (PSM variantas) gaunami daug geresni trRosetta įverčiai, tačiau jie išlieka žemesni nei GPT sugeneruotų ar natūralių sekų atveju. Sugeneruotos sekos, puikiai atkuriančios pirmos eilės statistikas ir iki tam tikro lygio antros eilės statistikas, pasižymi natūralioms sekoms artimais įverčiais. Tikėtina, kad aukštesnio lygio statistikų atkartojimas yra viena iš sąlygų generuojant baltymų sekas, tinkančias specifiniam baltymo karkasui. Dalis GPT sugeneruotų sekų pasižymi netgi geresniais trRosetta įverčiais nei taikinio seka.

GPT sugeneruotos MDH sekos pasižymi itin didele sekų įvairove. Sugeneruoti sekų variantai užpildo tarp natūralių baltymų esančią erdvę, tai turėtų suteikti galimybę interpoliuoti tarp natūralių sekų variantų. Ši interpoliacija stebima keičiant įvesties vektorių – palaipsniui keičiantis įvesties vektoriaus vertei yra stebimas sugeneruotų sekų judėjimas link skirtingų natūralių sekų variantų. Visgi panašu, kad itin nuo natūralių sekų nutolę sugeneruotų sekų variantai pasižymi blogesnėmis savybėmis (mažesnė raiška/tirpumas, aktyvumas). Sugeneruotų sekų įvairovė atsiranda iš natūraliose sekose jau egzistuojančių pozicinių aminorūgščių skirstinių ar jų porų. Tai reiškia, kad sugeneruotos sekos atitinka pagal PSM rekombinuotas pozicijas, kurioms kartu yra išlaikomi natūralioms sekoms būdingi poriniai aminorūgščių dažniai. Iš to kyla itin didelis skaičius galimų sekų variantų, kurie ir yra generuojami GPT.

4.2. Latentinė erdvė ir GPT pritaikymas

Daug žadanti GPT dalis yra latentinė GPT erdvė. Kaip parodyta šiame darbe, atskiros įvesties vektorių dimensijos itin koreliuoja su eile paprastų baltymų savybių. Tai skatina atlikti platesnius šio lauko tyrinėjimus. Šiuo metu vis dar lieka neaišku, ar latentinėje erdvėje įmanoma atrasti krypčių, kurios galėtų keisti sudėtingesnes baltymų savybes: stabilumą, aktyvumą ar specifiškumą. Toks panaudojimas gali būti tiesiogiai taikomas praktikoje ir yra itin perspektyvus akademinėse, industrinėse ir farmacinėse sferose.

Judėjimas latentinėje GPT erdvėje tarp dviejų ar daugiau taškų teoriškai gali veikti panašiai kaip protėvių sekų rekonstrukcija (PSR) (angl. *ancestral sequence reconstruction*). Sekos iš PSR gali pasižymėti geresnėmis savybėmis lyginant su pradiniu sekos variantu: didesnis termostabilumas, geresnė raiška, platesnis substratinis specifiškumas (Babkova ir kt., 2017, 2020; Watanabe ir kt., 2006). PSR atveju mutacijų kombinacijos yra parenkamos priklausomai nuo

[3] trRosetta įverčiai nustatomi pagal tinklo nuostolius.

filogenetinės mutacijų ir baltymų kilmės. Kombinuojant mutacijas pagal GPT latentinę erdvę filogenetinis sekų atkūrimas yra mažai tikėtinas, tačiau galutinis efektas gali būti panašus. Erdvėje judant tarp dviejų baltymus atitinkančių taškų, pvz.: termostabilaus ir itin aktyvaus baltymų variantų, galima bandyti sukurti tarpinį šių baltymų atitikmenį, pasižyminti padidintu termostabilumu ir aktyvumu.

Įrankis, generuojantis funkcionalius sekų variantus, leidžia itin sumažinti *in silico* ir *in vitro* atliekamų paieškų mastus. *In silico* kuriami baltymų dizaino metodai įprastai negali įvertinti visų galimų baltymo sekos variantų. Dėl šios priežasties įprastai naudojami įvairūs stochastiniai metodai: Markovo grandinės Monte-Karlo metodas, Gibso imties išrinkimo algoritmai, stochastinė aproksimacija ir kt. Šie metodai yra jautrūs įvairiems erdvėje esantiems minimumams, todėl galutiniai sukuriami sekų variantai labai stipriai priklauso nuo pasirinkto pradinio taško. Panašūs apribojimai yra būdingi ir kryptingos evoliucijos metodams. GPT tinklai gali sukurti plačiai erdvėje pasiskirsčiusius baltymų variantus, turinčius daug didesnę tikimybę reprezentuoti aktyvias baltymų sekas. Tokie variantai gali būti naudojami kaip pradžios taškai *in silico* ir *in vitro* tyrimuose. Tai suteikia galimybę atlikti paiešką platesnėje baltymų erdvėje, neapribotoje vien tik žinomų sekų variantų.

4.3. Sugeneruotų baltymų aktyvumo tyrimas

Sekų biologinio aktyvumo įvertinimas parodė, kad didesnė problema sugeneruotų sekų atveju yra pačių baltymų tirpumas, o ne jų aktyvumas. Vykdant du skirtingus baltymų gryninimo protokolus pavyko gauti bendrai didesnį kiekį tirpių baltymų variantų, iš kurių didelė dalis pasižymėjo biologiniu aktyvumu. Gali būti, kad žemo baltymų tirpumo problema prasideda nuo mokymui parenkamų sekų. Pradiniai sekų variantai GPT mokymui nebuvo nei parenkami, nei optimizuojami taip, kad būtų užtikrinamas jų tirpumas ir raišką darbe aprašytose baltymo raiškos sistemose. Tokie natūralūs sekų variantai gali būti netirpūs, kai yra naudojami rekombinantinių baltymų raiškoje. Nepriklausomi sisteminiai tyrimai parodė, kad tikėtinas netirpių natūralių rekombinantinių baltymų variantų skaičius varijuoja nuo 80 % iki 60 % (Huang ir kt., 2015; Mashiyama ir kt., 2014; Pertusi ir kt., 2015). Tai savaime reiškia, kad ir sugeneruotos sekos nebūtinai gali išlikti tirpios, netgi jei jos yra itin panašios į natūralias. Deja, bet šiuo metu tai yra neišvengiamybė, kadangi aprašytų tirpių baltymų sekų variantų dalis (specifiškai *E. coli* raiškos sistemoje) yra itin maža palyginus su duomenų bazių dydžiais, tuo tarpu baltymų tirpumo klasifikatoriai pasižymi per mažu tikslumu, kad jų spėjimai būtų panaudojami praktikoje.

Tirtuose sekų variantuose mažėjant sekų panašumui į natūralias sekas stebimas sistemingai mažėjantis tirpių/aktyvių variantų skaičius (**2 priedas**). Didžioji aktyvių variantų dalis matoma iki 80 % sekų tarpusavio panašumo, ties žemesniu panašumo lygiu aktyvių MDH variantų skaičius

mažėja. Tokį rezultatą sunku paaiškinti vien tik bendra rekombinantinių baltymų tirpumo problema. Pagal panašumo į natūralias sekas ir tirpumo koreliaciją galima padaryti išvadą, kad tikimybė, jog sugeneruoti baltymų variantai bus tirpūs priklauso nuo sekų panašumo į natūralias sekas – mažėjant panašumui kartu mažėja ir tikimybė, kad sekos variantas bus tirpus.

Pateikti rezultatai iškelia klausimą, kodėl pagal pateiktas metrikas gerai įvertintas sekų rinkinys nebūtinai pasižymi tokiais pat gerais eksperimentiniais rezultatais. Tai išryškina šiame darbe ir bendrai bioinformatikos moksle naudojamų metrikų trūkumus: baltymai yra kompleksinės, dinamiškos makromolekulės, todėl jų savybių apibūdinimas ir įvertinimas yra itin sudėtingas. Dėl šių priežasčių vis dar nėra standartų, kurie leistų pagal baltymo struktūrą ar seką vienareikšmiškai įvertinti jo savybes. Šiame darbe daugiausia dėmesio skirta sekų rinkinių palyginimui, tačiau laboratorijoje yra testuojami pavieniai sekų variantai, o ne jų rinkiniai.

In vitro tyrimams baltymų variantai parinkti pagal jų panašumą į natūralias sekas. Praktiniame pritaikyme galima vykdyti GPT sugeneruotų sekų variantų atranką pagal kitus įverčius. Toks kompozicinis baltymų generavimas ir atrinkimas gali padidinti funkcionalių baltymų dalį. Variantų atrankai gali būti naudojamas jau aptartas trRosetta įvertis ar kiti MM metodai.

4.4. Tolimesnės perspektyvos

Rezultatų analizė leido suprasti galimus tinklo ir jo mokymo trūkumus bei planuoti GPT patobulinimus. Pagal tirtas sekų statistikas mažiausias panašumas tarp sugeneruotų ir natūralių sekų rinkinių matomas BI įvertyje. Tai galėjo lemti kelios tinklo savybės:

- Nepakankamas tinklo parametrų skaičius. Esant nepakankamai tinklo parametrų mažiau sekose išreikštos statistikos nėra išmokstamos dėl per mažos tinklo „atminties“.
- Nepakankamas tinklo mokymo laikas. Tinklui mokantis pirmiausia yra išmokstamos bendriausios ir lengviausiai įvertinamos sekų ypatybės. Tobulėjant generatoriui ir diskriminatoriui generatorius yra verčiamas išmokti vis sudėtingesnes sekų ypatybes.
- Netinkama tinklo architektūra. Tinklo išmokstamus sekų bruožus gali limituoti netinkamai uždaviniui optimizuota architektūra.

Turint specifinius sekų įverčius išvardintos problemos gali būti tikrinamos ir joms pasitvirtinus taisomos.

Greitesniam tinklo mokymui, didesnei sekų įvairovei, bei geresnei jų kokybei pasiekti galima taikyti pradinį apmokymą (angl. *pre-training*). Pradinis tinklo apmokymas įprastai būna atliekamas su įvairiomis reprezentacinėmis baltymų grupėmis. Taikant pradinį tinklo apmokymą galima itin paspartinti vėlyvesnius tinklo apmokymus su tiksliniais baltymai. Pradinio apmokymo metu tinklas gali išmokti bendrai visoms baltymų sekoms būdingus bruožus. Tai leidžia sparčiau

atlikti ir tirti tinklo pakeitimus, ilgainiui padeda sutaupyti resursų bei gaunami geresni galutiniai rezultatai.

Nors šiame darbe neanalizuota, galutinė sekų kokybė priklauso ir nuo mokymo rinkinio dydžio. Šiuo atveju naudotas fiksuotas mokymo rinkinio dydis, todėl vis dar išlieka neaišku, koks minimalus skaičius sekų yra reikalingas patenkinamiems rezultatams gauti. Mažiau duomenų bazėse reprezentuotos baltymų grupės gali turėti tik kelis šimtus sekų variantų. Geresniems rezultatams su tokiomis sekų grupėmis pasiekti galėtų padėti jau minėtas pradinis tinklo apmokymas. Kitas sprendimas gali būti sąlyginis GPT (angl. *conditional GAN*). Toks tinklas galėtų būti apmokomas pagal baltymų grupes su žymenimis. Žymuo šiuo atveju gali turėti fizikinę reikšmę: reakcijos mechanizmo aprašymas (fermentų grupių atveju) ar reprezentacinis sekų grupės PSSM profilis. Toks apmokymas leistų GPT naudoti įvairių baltymų grupių generavimui po vieno bendro mokymo. Didžiausias iš tokio apmokymo kylantis privalumas yra naujų baltymų grupių generavimas pagal duotąjį fizikinį žymenį. Yra tikimasi, kad po sąlyginio GPT apmokymo tinklas išmoksta bendrus baltymams būdingus bruožus (pvz., specifinis aminorūgščių dažnis ir pasiskirstymas atitinkamose antrinėse struktūrose, specifinė domenų sandara ir kt.) ir pagal juos geba generuoti ne tik naujus tinklui jau matytos baltymų grupės sekų variantus, bet ir tinklui nematytos ar mažai reprezentuotos grupės sekų variantus.

IŠVADOS

1. Generatyviniais priešiškais tinklais (GPT) sugeneruotos baltymų sekos atkuria natūralioms sekoms būdingas pirmos eilės baltymų rinkinio statistikas.
2. GPT gebėjimas atkurti antros eilės sekų statistikas priklauso nuo pozicijų atstumo sekoje, todėl matomas tik dalinis šių statistikų atkūrimas.
3. GPT sugeneruoti baltymų sekų variantai pasižymi bent 3,5 karto didesne sekų įvairove lyginant su natūralių sekų rinkiniu.
4. Kryptingas GPT latentinės erdvės dimensijų keitimas leidžia selektyviai keisti generuojamų baltymų sekų savybes.
5. 13 iš 55 GPT sugeneruotų malato dehidrogenazės variantų katalizuoja nuo NAD^+ priklausomą oksaloacetato redukciją į malatą.

Publikacijos darbo tema

Repecka, D., Jauniskis, V., Karpus, L., Rembeza, E., Rokaitis, I., Zrimec, J. Poviloniene, S., Laurynenas, A., Viknander, S., Abuajwa, W., Savolainen, O., Meskys, R., Engqvist, M. K. M., Zelezniak, A. Expanding functional protein sequence spaces using generative adversarial networks. *Nat Mach Intell* 2021, 3, 324–333. https://doi.org/10.1038/s42256-021-00310-5;.

Generatyviniais priešiškais tinklais sugeneruotų baltymų sekų analizė

SANTRAUKA

Baltymų inžinerijos metodais siekiama sukurti naujomis ar patobulintomis savybėmis pasižyminčius baltymus. Populiariausi šios mokslo srities įrankiai išlieka kryptinga evoliucija ir racionalus baltymų dizainas, tačiau platus jų panaudojimas vis dar yra ribojamas specifinio pritaikymo kiekvienam taikiniui, ilgų testavimo ciklų ir laboratorinių išteklių reikmės. Nuolat augančios biologinių duomenų bazės leidžia pritaikyti mašininio mokymosi modelius, kurie gali pasižymėti greitu veikimu ir dideliu tikslumu, taip papildant klasikinius baltymų inžinerijos metodus.

Šio darbo tikslas yra ištirti generatyviniais priešiškais tinklais sugeneruotas fermento malato dehidrogenazės sekas. Nustatyta, kad sugeneruotos baltymų sekos atkuria natūralioms sekoms būdingas pirmos ir antros eilės statistikas bei praplečia natūralių baltymų sekų erdvę, užpildant ją naujais variantais. *In vitro* atlikti tyrimai atskleidė, kad 13 iš 55 sugeneruotų baltymų variantų pasižymi natūralioms sekoms būdingu kataliziniu aktyvumu. Parodyta, kad kryptingas latentinės generatyvinių priešiškų tinklų erdvės dimensijų keitimas leidžia kontroliuoti generuojamų baltymų sekų savybes, tai suteikia galimybę juos pritaikyti tiesiogiai sprendžiant baltymų inžinerijos problemas.

Analysis of Protein Sequences Generated by Generative Adversarial Networks

SUMMARY

A long-standing goal of protein engineering is the design of proteins with novel or improved properties. Despite years of research aimed towards improving traditional approaches of rational design and directed evolution, their application remains hindered due to high time, labor and resource requirements. The ever-growing availability of biological data empowers the use of machine learning methods at solving protein engineering tasks that were either hard or impossible to solve using the conventional tools.

This study aimed to analyze malate dehydrogenase sequences generated by generative adversarial networks. The generated sequences recapitulate first and second order sequence statistics, while substantially expanding the natural sequence space. 13 out of 55 sequence variants sampled from generative adversarial network's latent space retained biological activity. Guided changes of the latent space variables correlate with various proteins sequence features – a capability directly applicable in protein engineering.

1 priedas

1 lentelė. Apskaičiuotų baltymų savybių minimalios bei maksimalios reikšmės ir maksimalus koreliacijos koeficientas keičiant pavienių įvesties vektoriaus dimensijų vertes. Visos trys reikšmės gali atitikti skirtingas dimensijas.

Savybė	Minimali vertė sugeneruotose	Maksimali vertė sugeneruotose	Maksimalus koreliacijos koeficientas
C	0	0,64	0,9
D	5,4	7,99	0,95
S	2,56	5,1	0,95
Q	1,28	3,51	0,93
K	4,79	7,99	0,97
W	0	0,96	0,9
T	3,51	6,71	0,95
N	1,92	4,76	0,97
P	4,13	5,75	0,93
F	0,95	2,86	0,96
A	7,96	12,78	0,93
G	8,63	11,18	0,95
I	6,35	8,95	0,82
L	7,64	10,83	0,87
H	0,63	2,24	0,92
R	3,5	5,43	0,89
M	1,92	3,51	0,96
V	8,28	11,82	0,94
E	4,76	7,3	0,93
Y	1,92	4,46	0,96
Panašumas	55,1	80	0,98
Hidrofobiškumas	33,23	37,38	0,97
Hidrofiliškumas	37,06	43,17	0,98
Aromatiškumas	5,11	7,96	0,95
Mažos	21,73	27,16	0,95
Teigiamos	10,19	14,01	0,93
Neigiamos	10,22	14,06	0,97
Alifatinės	28,25	32,27	0,93
Hidroksi/sulfur	9,58	14,33	0,96
Polinės neįkrautos	14,7	21,27	0,95
Įkrautos	21,9	26,2	0,92
Svoris	32901,39	34233,06	0,97
Ilgis	313	315	0,95
Izoelektrinis taškas	4,71	8,5	0,88
Nestabilumo indeksas	16,31	39,78	0,9
Lankstumas	1	1	0,94

2 lentelė. Aminorūgščių grupės ir joms priklausančios aminorūgštys.

Grupė	**Priklausančios aminorūgštys**
Alifatinės	V, I, L, M
Aromatinės	F, W, Y, H
Hidrofobinės	V, I, L, F, W, Y, M
Hidrofilinės	S, T, H, N, Q, E, D, K, R
Mažos	G, A, S
Teigiamos	K, R, H
Neigiamos	D, E
Hidroksi/sulfur grupę turinčios	S, C, T, M
Polinės be krūvio	S, T, C, M, N, Q
Su krūviu	H, K, R, E, D

2 priedas

1 lentelė. *In vitro* įvertintų MDH variantų tirpumo ir aktyvumo rezultatai. „+“ nurodo, kad variantas yra tirpus/aktyvus. „-“ nurodo, kad variantas yra netirpus/neaktyvus. „NA“ nurodo, kad matavimas neatliktas. Nurodytas globalus panašumas į artimiausią natūralios sekos variantą.

		Metodas 1		Metodas 2	
Baltymas	**Panašumas, %**	**Tirpumas**	**Aktyvumas**	**Tirpumas**	**Aktyvumas**
MDH 2	1	+	+	+	-
MDH 3	1	+	+	+	NA
MDH 4	1	+	+	+	+
pGAN 5	0,52	-	NA	-	NA
pGAN 6	0,48	-	NA	-	NA
pGAN 7	0,58	-	NA	-	NA
pGAN 8	0,6	-	NA	-	NA
pGAN 9	0,66	-	NA	+	+
pGAN 10	0,75	-	NA	-	NA
pGAN 11	0,67	-	NA	-	NA
pGAN 12	0,75	-	NA	-	NA
pGAN 13	0,82	+	-	-	-
pGAN 14	0,55	-	NA	-	NA
pGAN 15	0,61	-	NA	+	-
pGAN 17	0,77	-	NA	-	NA
pGAN 18	0,77	-	NA	-	NA
pGAN 19	0,75	-	NA	-	NA
pGAN 20	0,79	-	NA	-	NA
pGAN 21a	0,88	-	NA	-	NA
pGAN 22a	0,83	-	NA	-	-
pGAN 21b	0,84	-	-	+	-
pGAN 22b	0,86	+	+	+	+
pGAN 23	0,87	-	-	+	-
pGAN 24	0,88	+	+	+	-
pGAN 25	0,88	+	+	+	-
pGAN 26	0,9	-	-	+	-
pGAN 27	0,89	-	-	-	-
pGAN 28	0,91	+	+	+	+
pGAN 29	0,92	+	+	-	NA
pGAN 30	0,92	-	-	-	NA
pGAN 31	0,92	+	+	-	-
pGAN 32	0,93	-	-	-	NA
pGAN 33	0,94	-	-	-	-
pGAN 34	0,95	+	+	-	-
pGAN 35	0,95	+	+	+	+
pGAN 36	0,96	-	-	-	NA
pGAN 37	0,96	-	-	+	+
pGAN 39	0,98	+	+	+	+
pGAN 40	0,65	-	-	-	-
pGAN 41	0,6	-	-	-	-
pGAN 42	0,62	-	-	-	NA
pGAN 43	0,64	-	-	-	NA
pGAN 44	0,64	-	-	-	NA
pGAN 45	0,66	-	-	-	NA
pGAN 46	0,64	-	-	-	NA
pGAN 47	0,69	-	-	-	NA
pGAN 48	0,7	-	-	-	NA
pGAN 49	0,69	-	-	-	NA
pGAN 50	0,7	-	-	-	NA
pGAN 51	0,72	-	-	-	NA
pGAN 52	0,77	-	-	-	NA
pGAN 53	0,76	-	-	-	NA
pGAN 54	0,77	-	-	-	NA
pGAN 55	0,77	-	-	-	NA
pGAN 56	0,79	+	-	+	-
pGAN 57	0,82	-	-	-	NA
pGAN 59	0,84	+	+	-	NA
pGAN 60	0,85	+	+	+	+

LITERATŪROS SĄRAŠAS

1. Abdal, R., Qin, Y., ir Wonka, P. (2019). Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space? *arXiv:1904.03189 [cs]*. http://arxiv.org/abs/1904.03189
2. Adhikari, B., ir Cheng, J. (2016). Protein Residue Contacts and Prediction Methods. *Methods in molecular biology (Clifton, N.J.)*, *1415*, 463–476. https://doi.org/10.1007/978-1-4939-3572-7_24
3. Ayoob, M. (2020). *Manifold Learning of Latent Space Vectors in Generative Adversarial Networks for Image Synthesis.*
4. Alley, E. C., Khimulya, G., Biswas, S., AlQuraishi, M., ir Church, G. M. (2019). Unified rational protein engineering with sequence-based deep representation learning. *Nature Methods*, *16*(12), 1315–1322. https://doi.org/10.1038/s41592-019-0598-1
5. Alloghani, M., Al-Jumeily, D., Mustafina, J., Hussain, A., ir Aljaaf, A. J. (2020). A Systematic Review on Supervised and Unsupervised Machine Learning Algorithms for Data Science. M. W. Berry, A. Mohamed, ir B. W. Yap (Sud.), *Supervised and Unsupervised Learning for Data Science* (p. 3–21). Springer International Publishing. https://doi.org/10.1007/978-3-030-22475-2_1
6. *AlphaFold: A solution to a 50-year-old grand challenge in biology*. Deepmind. Gauta 2021 m. gegužės 23 d., deepmind.com/blog/article/alphafold-a-solution-to-a-50-year-old-grand-challenge-in-biology
7. Altschul, S. F., Gish, W., Miller, W., Myers, E. W., ir Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, *215*(3), 403–410. https://doi.org/10.1016/S0022-2836(05)80360-2
8. Anishchenko, I., Chidyausiku, T. M., Ovchinnikov, S., Pellock, S. J., ir Baker, D. (2020). De novo protein design by deep network hallucination. *BioRxiv*, 2020.07.22.211482. https://doi.org/10.1101/2020.07.22.211482
9. Anishchenko, I., Ovchinnikov, S., Kamisetty, H., ir Baker, D. (2017). Origins of coevolution between residues distant in protein 3D structures. *Proceedings of the National Academy of Sciences*, *114*(34), 9122–9127. https://doi.org/10.1073/pnas.1702664114
10. Arjovsky, M., Chintala, S., ir Bottou, L. (2017). Wasserstein GAN. *arXiv:1701.07875 [cs, stat]*. http://arxiv.org/abs/1701.07875
11. Arnold, F. H. (2015). The nature of chemical innovation: New enzymes by evolution. *Quarterly Reviews of Biophysics*, *48*(4), 404–410. https://doi.org/10.1017/S003358351500013X
12. Arnold, F. H. (2019). Innovation by Evolution: Bringing New Chemistry to Life (Nobel Lecture). *Angewandte Chemie International Edition*, *58*(41), 14420–14426. https://doi.org/10.1002/anie.201907729
13. Ashenberg, O., ir Laub, M. T. (2013). Using analyses of amino Acid coevolution to understand protein structure and function. *Methods in Enzymology*, *523*, 191–212. https://doi.org/10.1016/B978-0-12-394292-0.00009-6
14. Ashkenazy, H., Abadi, S., Martz, E., Chay, O., Mayrose, I., Pupko, T., ir Ben-Tal, N. (2016). ConSurf 2016: An improved methodology to estimate and visualize evolutionary conservation in macromolecules. *Nucleic Acids Research*, *44*(Web Server issue), W344–W350. https://doi.org/10.1093/nar/gkw408
15. Babkova, P., Dunajova, Z., Chaloupkova, R., Damborsky, J., Bednar, D., ir Marek, M. (2020). Structures of hyperstable ancestral haloalkane dehalogenases show restricted conformational dynamics. *Computational and Structural Biotechnology Journal*, *18*, 1497–1508. https://doi.org/10.1016/j.csbj.2020.06.021
16. Babkova, P., Sebestova, E., Brezovsky, J., Chaloupkova, R., ir Damborsky, J. (2017). Ancestral Haloalkane Dehalogenases Show Robustness and Unique Substrate Specificity.

Chembiochem: A European Journal of Chemical Biology, *18*(14), 1448–1456. https://doi.org/10.1002/cbic.201700197
17. Ben-Gal, I., Shani, A., Gohr, A., Grau, J., Arviv, S., Shmilovici, A., Posch, S., ir Grosse, I. (2005). Identification of transcription factor binding sites with variable-order Bayesian networks. *Bioinformatics*, *21*(11), 2657–2666. https://doi.org/10.1093/bioinformatics/bti410
18. Bernardo, J., Bayarri, M., Berger, J., Dawid, A., Heckerman, D., Smith, A., West, M., Bishop, C., ir Lasserre, J. (2007). Generative or Discriminative? Getting the Best of Both Worlds. *BAYESIAN STATISTICS*, *8*, 3–24.
19. Binmore, K. (2007). *Playing for Real: A Text on Game Theory*. Oxford University Press.
20. Biswas, S., Khimulya, G., Alley, E. C., Esvelt, K. M., ir Church, G. M. (2021). Low- N protein engineering with data-efficient deep learning. *Nature Methods*, *18*(4), 389–396. https://doi.org/10.1038/s41592-021-01100-y
21. Bojanowski, P., Joulin, A., Lopez-Paz, D., ir Szlam, A. (2019). Optimizing the Latent Space of Generative Networks. *ArXiv:1707.05776 [Cs, Stat]*. http://arxiv.org/abs/1707.05776
22. Bornscheuer, U. T., Hauer, B., Jaeger, K. E., ir Schwaneberg, U. (2019). Directed Evolution Empowered Redesign of Natural Proteins for the Sustainable Production of Chemicals and Pharmaceuticals. *Angewandte Chemie International Edition*, *58*(1), 36–40. https://doi.org/10.1002/anie.201812717
23. Brannigan, J. A., ir Wilkinson, A. J. (2002). Protein engineering 20 years on. *Nature Reviews Molecular Cell Biology*, *3*(12), 964–970. https://doi.org/10.1038/nrm975
24. Breiter, D. R., Resnik, E., ir Banaszak, L. J. (1994). Engineering the quaternary structure of an enzyme: Construction and analysis of a monomeric form of malate dehydrogenase from Escherichia coli. *Protein Science : A Publication of the Protein Society*, *3*(11), 2023–2032.
25. Callaway, E. (2020). 'It will change everything': DeepMind's AI makes gigantic leap in solving protein structures. *Nature*, *588*(7837), 203–204. https://doi.org/10.1038/d41586-020-03348-4
26. Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., ir Madden, T. L. (2009). BLAST+: Architecture and applications. *BMC Bioinformatics*, *10*(1), 421. https://doi.org/10.1186/1471-2105-10-421
27. Capra, J. A., ir Singh, M. (2007). Predicting functionally important residues from sequence conservation. *Bioinformatics*, *23*(15), 1875–1882. https://doi.org/10.1093/bioinformatics/btm270
28. Carvalho, H. F., Barbosa, A. J. M., Roque, A. C. A., Iranzo, O., ir Branco, R. J. F. (2017). Integration of Molecular Dynamics Based Predictions into the Optimization of De Novo Protein Designs: Limitations and Benefits. *Computational Protein Design*, 181–201. https://doi.org/10.1007/978-1-4939-6637-0_8
29. Cock, P. J. A., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B., ir de Hoon, M. J. L. (2009). Biopython: Freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, *25*(11), 1422–1423. https://doi.org/10.1093/bioinformatics/btp163
30. Cooper, G. M., ir Brown, C. D. (2008). Qualifying the relationship between sequence conservation and molecular function. *Genome Research*, *18*(2), 201–205. https://doi.org/10.1101/gr.7205808
31. Crooks, G. E., Hon, G., Chandonia, J.-M., ir Brenner, S. E. (2004). WebLogo: A sequence logo generator. *Genome Research*, *14*(6), 1188–1190. https://doi.org/10.1101/gr.849004
32. De Ferrari, L., ir Mitchell, J. B. (2014). From sequence to enzyme mechanism using multi-label machine learning. *BMC Bioinformatics*, *15*(1), 150. https://doi.org/10.1186/1471-2105-15-150
33. de Juan, D., Pazos, F., ir Valencia, A. (2013). Emerging methods in protein co-evolution. *Nature Reviews Genetics*, *14*(4), 249–261. https://doi.org/10.1038/nrg3414

34. Dunn, S. D., Wahl, L. M., ir Gloor, G. B. (2008). Mutual information without the influence of phylogeny or entropy dramatically improves residue contact prediction. *Bioinformatics*, *24*(3), 333–340. https://doi.org/10.1093/bioinformatics/btm604
35. Eddy, S. R. (1996). Hidden Markov models. *Current Opinion in Structural Biology*, *6*(3), 361–365. https://doi.org/10.1016/S0959-440X(96)80056-X
36. Eddy, S. R. (2004). What is a hidden Markov model? *Nature Biotechnology*, *22*(10), 1315–1316. https://doi.org/10.1038/nbt1004-1315
37. Eddy, S. R. (2009). A new generation of homology search tools based on probabilistic inference. *Genome Informatics. International Conference on Genome Informatics*, *23*(1), 205–211.
38. Edgar, R. C. (2010). Search and clustering orders of magnitude faster than BLAST. *Bioinformatics*, *26*(19), 2460–2461. https://doi.org/10.1093/bioinformatics/btq461
39. Edgar, R. C., ir Batzoglou, S. (2006). Multiple sequence alignment. *Current Opinion in Structural Biology*, *16*(3), 368–373. https://doi.org/10.1016/j.sbi.2006.04.004
40. Eswar, N., Webb, B., Marti-Renom, M. A., Madhusudhan, M. S., Eramian, D., Shen, M., Pieper, U., ir Sali, A. (2006). Comparative Protein Structure Modeling Using Modeller. *Current protocols in bioinformatics / editoral board, Andreas D. Baxevanis ... [et al.]*, *0 5*, Unit-5.6. https://doi.org/10.1002/0471250953.bi0506s15
41. Falda, M., Toppo, S., Pescarolo, A., Lavezzo, E., Di Camillo, B., Facchinetti, A., Cilia, E., Velasco, R., ir Fontana, P. (2012). Argot2: A large scale function prediction tool relying on semantic similarity of weighted Gene Ontology terms. *BMC Bioinformatics*, *13*(4), S14. https://doi.org/10.1186/1471-2105-13-S4-S14
42. Fasan, R., Jennifer Kan, S. B., ir Zhao, H. (2019). A Continuing Career in Biocatalysis: Frances H. Arnold. *ACS Catalysis*, *9*(11), 9775–9788. https://doi.org/10.1021/acscatal.9b02737
43. Gao, W., Mahajan, S. P., Sulam, J., ir Gray, J. J. (2020). Deep Learning in Protein Structural Modeling and Design. *Patterns*, *1*(9), 100142. https://doi.org/10.1016/j.patter.2020.100142
44. Gloor, G. B., Martin, L. C., Wahl, L. M., ir Dunn, S. D. (2005). Mutual Information in Protein Multiple Sequence Alignments Reveals Two Classes of Coevolving Positions. *Biochemistry*, *44*(19), 7156–7165. https://doi.org/10.1021/bi050293e
45. Goldsborough, P., Pawlowski, N., Caicedo, J. C., Singh, S., ir Carpenter, A. E. (2017). CytoGAN: Generative Modeling of Cell Images. *BioRxiv*, 227645. https://doi.org/10.1101/227645
46. Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., ir Bengio, Y. (2014). Generative adversarial nets. *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, 2672–2680.
47. Gribskov, M., McLachlan, A. D., ir Eisenberg, D. (1987). Profile analysis: Detection of distantly related proteins. *Proceedings of the National Academy of Sciences of the United States of America*, *84*(13), 4355–4358.
48. Guharoy, M., ir Chakrabarti, P. (2005). Conservation and relative importance of residues across protein-protein interfaces. *Proceedings of the National Academy of Sciences*, *102*(43), 15447–15452. https://doi.org/10.1073/pnas.0505425102
49. Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., ir Lew, M. S. (2016). Deep learning for visual understanding: A review. *Neurocomputing*, *187*, 27–48. https://doi.org/10.1016/j.neucom.2015.09.116
50. Guruprasad, K., Reddy, B. V., ir Pandit, M. W. (1990). Correlation between stability of a protein and its dipeptide composition: A novel approach for predicting in vivo stability of a protein from its primary sequence. *Protein Engineering*, *4*(2), 155–161. https://doi.org/10.1093/protein/4.2.155
51. Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van

Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2
52. Hinkle, D. E., Wiersma, W., ir Jurs, S. G. (2003). *Applied Statistics for the Behavioral Sciences*. Houghton Mifflin.
53. Hong, H., Jiang, S., Li, H., Du, G., Sun, Y., Tao, H., Quan, C., Zhao, C., Li, R., Li, W., Yin, X., Huang, Y., Li, C., Chen, H., ir Bo, X. (2020). DeepHiC: A generative adversarial network for enhancing Hi-C data resolution. *PLOS Computational Biology*, *16*(2), e1007287. https://doi.org/10.1371/journal.pcbi.1007287
54. Hopf, T. A., Ingraham, J. B., Poelwijk, F. J., Schärfe, C. P. I., Springer, M., Sander, C., ir Marks, D. S. (2017). Mutation effects predicted from sequence co-variation. *Nature biotechnology*, *35*(2), 128–135. https://doi.org/10.1038/nbt.3769
55. Huang, H., Pandya, C., Liu, C., Al-Obaidi, N. F., Wang, M., Zheng, L., Keating, S. T., Aono, M., Love, J. D., Evans, B., Seidel, R. D., Hillerich, B. S., Garforth, S. J., Almo, S. C., Mariano, P. S., Dunaway-Mariano, D., Allen, K. N., ir Farelli, J. D. (2015). Panoramic view of a superfamily of phosphatases through substrate profiling. *Proceedings of the National Academy of Sciences*, *112*(16), E1974–E1983. https://doi.org/10.1073/pnas.1423570112
56. Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55
57. Yang, Y., Niroula, A., Shen, B., ir Vihinen, M. (2016). PON-Sol: Prediction of effects of amino acid substitutions on protein solubility. *Bioinformatics (Oxford, England)*, *32*(13), 2032–2034. https://doi.org/10.1093/bioinformatics/btw066
58. Yang, J., Anishchenko, I., Park, H., Peng, Z., Ovchinnikov, S., ir Baker, D. (2020). Improved protein structure prediction using predicted interresidue orientations. *Proceedings of the National Academy of Sciences*, *117*(3), 1496–1503. https://doi.org/10.1073/pnas.1914677117
59. Yoon, B.-J. (2009). Hidden Markov Models and their Applications in Biological Sequence Analysis. *Current Genomics*, *10*(6), 402–415. https://doi.org/10.2174/138920209789177575
60. Yoon, B.-J., ir Vaidyanathan, P. P. (2006). Profile Context-Sensitive HMMs for Probabilistic Modeling of Sequences With Complex Correlations. *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, *3*, III–III. https://doi.org/10.1109/ICASSP.2006.1660654
61. Young, T., Hazarika, D., Poria, S., ir Cambria, E. (2018). Recent Trends in Deep Learning Based Natural Language Processing. *arXiv:1708.02709 [cs]*. http://arxiv.org/abs/1708.02709
62. Johansson, F., ir Toh, H. (2010). A comparative study of conservation and variation scores. *BMC Bioinformatics*, *11*, 388. https://doi.org/10.1186/1471-2105-11-388
63. Johnson, S. R., Monaco, S., Massie, K., ir Syed, Z. (2021). Generating novel protein sequences using Gibbs sampling of masked language models. *BioRxiv*, 2021.01.26.428322. https://doi.org/10.1101/2021.01.26.428322
64. Jones, S., ir Thornton, J. M. (2004). Searching for functional sites in protein structures. *Current Opinion in Chemical Biology*, *8*(1), 3–7. https://doi.org/10.1016/j.cbpa.2003.11.001
65. Kalinina, O. V., Mironov, A. A., Gelfand, M. S., ir Rakhmaninova, A. B. (2004). Automated selection of positions determining functional specificity of proteins by comparative analysis of orthologous groups in protein families. *Protein Science*, *13*(2), 443–456. https://doi.org/10.1110/ps.03191704
66. Kamisetty, H., Ovchinnikov, S., ir Baker, D. (2013). Assessing the utility of coevolution-based residue–residue contact predictions in a sequence- and structure-rich era. *Proceedings of the National Academy of Sciences*, *110*(39), 15674–15679. https://doi.org/10.1073/pnas.1314045110

67. Katoh, K., ir Standley, D. M. (2013). MAFFT Multiple Sequence Alignment Software Version 7: Improvements in Performance and Usability. *Molecular Biology and Evolution*, *30*(4), 772–780. https://doi.org/10.1093/molbev/mst010
68. Kaznatcheev, A. (2019). Computational Complexity as an Ultimate Constraint on Evolution. *Genetics*, *212*(1), 245–265. https://doi.org/10.1534/genetics.119.302000
69. Kemena, C., ir Notredame, C. (2009). Upcoming challenges for multiple sequence alignment methods in the high-throughput era. *Bioinformatics (Oxford, England)*, *25*(19), 2455–2465. https://doi.org/10.1093/bioinformatics/btp452
70. Korber, B. T., Farber, R. M., Wolpert, D. H., ir Lapedes, A. S. (1993). Covariation of mutations in the V3 loop of human immunodeficiency virus type 1 envelope protein: An information theoretic analysis. *Proceedings of the National Academy of Sciences of the United States of America*, *90*(15), 7176–7180.
71. Korendovych, I. V., ir DeGrado, W. F. (2020). De novo protein design, a retrospective. *Quarterly Reviews of Biophysics*, *53*. https://doi.org/10.1017/S0033583519000131
72. Larsson, A. (2014). AliView: A fast and lightweight alignment viewer and editor for large datasets. *Bioinformatics*, *30*(22), 3276–3278. https://doi.org/10.1093/bioinformatics/btu531
73. Leaver-Fay, A., Tyka, M., Lewis, S. M., Lange, O. F., Thompson, J., Jacak, R., Kaufman, K., Renfrew, P. D., Smith, C. A., Sheffler, W., Davis, I. W., Cooper, S., Treuille, A., Mandell, D. J., Richter, F., Ban, Y.-E. A., Fleishman, S. J., Corn, J. E., Kim, D. E., … Bradley, P. (2011). ROSETTA3: An object-oriented software suite for the simulation and design of macromolecules. *Methods in Enzymology*, *487*, 545–574. https://doi.org/10.1016/B978-0-12-381270-4.00019-6
74. Li, W., Fu, L., Niu, B., Wu, S., ir Wooley, J. (2012). Ultrafast clustering algorithms for metagenomic sequence analysis. *Briefings in Bioinformatics*, *13*(6), 656–668. https://doi.org/10.1093/bib/bbs035
75. Liang, S., Zhang, C., Liu, S., ir Zhou, Y. (2006). Protein binding site prediction using an empirical scoring function. *Nucleic Acids Research*, *34*(13), 3698–3707. https://doi.org/10.1093/nar/gkl454
76. Madden, T. L., Tatusov, R. L., ir Zhang, J. (1996). Applications of network BLAST server. *Methods in Enzymology*, *266*, 131–141. https://doi.org/10.1016/s0076-6879(96)66011-x
77. Marcos, E., Basanta, B., Chidyausiku, T. M., Tang, Y., Oberdorfer, G., Liu, G., Swapna, G. V. T., Guan, R., Silva, D.-A., Dou, J., Pereira, J. H., Xiao, R., Sankaran, B., Zwart, P. H., Montelione, G. T., ir Baker, D. (2017). Principles for designing proteins with cavities formed by curved β sheets. *Science*, *355*(6321), 201–206. https://doi.org/10.1126/science.aah7389
78. Mashiyama, S. T., Malabanan, M. M., Akiva, E., Bhosle, R., Branch, M. C., Hillerich, B., Jagessar, K., Kim, J., Patskovsky, Y., Seidel, R. D., Stead, M., Toro, R., Vetting, M. W., Almo, S. C., Armstrong, R. N., ir Babbitt, P. C. (2014). Large-scale determination of sequence, structure, and function relationships in cytosolic glutathione transferases across the biosphere. *PLoS Biology*, *12*(4), e1001843. https://doi.org/10.1371/journal.pbio.1001843
79. McGee, F., Novinger, Q., Levy, R. M., Carnevale, V., ir Haldane, A. (2021). Generative Capacity of Probabilistic Protein Sequence Models. *arXiv:2012.02296 [physics, q-bio]*. http://arxiv.org/abs/2012.02296
80. McKinney, W. (2010). Data Structures for Statistical Computing in Python. *Proceedings of the 9th Python in Science Conference*, 56–61. https://doi.org/10.25080/Majora-92bf1922-00a
81. Mescheder, L., Geiger, A., ir Nowozin, S. (2018). Which Training Methods for GANs do actually Converge? *arXiv:1801.04406 [cs]*. http://arxiv.org/abs/1801.04406
82. Minárik, P., Tomásková, N., Kollárová, M., ir Antalík, M. (2002). Malate dehydrogenases—Structure and function. *General Physiology and Biophysics*, *21*(3), 257–265.
83. Mirdita, M., von den Driesch, L., Galiez, C., Martin, M. J., Söding, J., ir Steinegger, M. (2017). Uniclust databases of clustered and deeply annotated protein sequences and

alignments. *Nucleic Acids Research*, *45*(D1), D170–D176. https://doi.org/10.1093/nar/gkw1081
84. Mor, B., Garhwal, S., ir Kumar, A. (2021). A Systematic Review of Hidden Markov Models and Their Applications. *Archives of Computational Methods in Engineering*, *28*(3), 1429–1448. https://doi.org/10.1007/s11831-020-09422-4
85. Morcos, F., Pagnani, A., Lunt, B., Bertolino, A., Marks, D. S., Sander, C., Zecchina, R., Onuchic, J. N., Hwa, T., ir Weigt, M. (2011). Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences*, *108*(49), E1293–E1301. https://doi.org/10.1073/pnas.1111471108
86. Musil, M., Konegger, H., Hon, J., Bednar, D., ir Damborsky, J. (2019). Computational Design of Stable and Soluble Biocatalysts. *ACS Catalysis*, *9*(2), 1033–1054. https://doi.org/10.1021/acscatal.8b03613
87. Needleman, S. B., ir Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, *48*(3), 443–453. https://doi.org/10.1016/0022-2836(70)90057-4
88. Nicoludis, J. M., ir Gaudet, R. (2018). Applications of sequence coevolution in membrane protein biochemistry. *Biochimica et Biophysica Acta (BBA) - Biomembranes*, *1860*(4), 895–908. https://doi.org/10.1016/j.bbamem.2017.10.004
89. Norn, C., Wicky, B. I. M., Juergens, D., Liu, S., Kim, D., Koepnick, B., Anishchenko, I., Players, F., Baker, D., ir Ovchinnikov, S. (2020). Protein sequence design by explicit energy landscape optimization. *BioRxiv*, 2020.07.23.218917. https://doi.org/10.1101/2020.07.23.218917
90. Norn, C., Wicky, B. I. M., Juergens, D., Liu, S., Kim, D., Tischer, D., Koepnick, B., Anishchenko, I., Players, F., Baker, D., ir Ovchinnikov, S. (2021). Protein sequence design by conformational landscape optimization. *Proceedings of the National Academy of Sciences*, *118*(11). https://doi.org/10.1073/pnas.2017228118
91. Packer, M. S., ir Liu, D. R. (2015). Methods for the directed evolution of proteins. *Nature Reviews Genetics*, *16*(7), 379–394. https://doi.org/10.1038/nrg3927
92. Pazos, F., ir Valencia, A. (2008). Protein co-evolution, co-adaptation and interactions. *The EMBO Journal*, *27*(20), 2648–2655. https://doi.org/10.1038/emboj.2008.189
93. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., ir Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*(85), 2825–2830.
94. Perez, F., ir Granger, B. E. (2007). IPython: A System for Interactive Scientific Computing. *Computing in Science Engineering*, *9*(3), 21–29. https://doi.org/10.1109/MCSE.2007.53
95. Pérez-Enciso, M., ir Zingaretti, L. M. (2019). A Guide on Deep Learning for Complex Trait Genomic Prediction. *Genes*, *10*(7), 553. https://doi.org/10.3390/genes10070553
96. Pertusi, D. A., Stine, A. E., Broadbelt, L. J., ir Tyo, K. E. J. (2015). Efficient searching and annotation of metabolic networks using chemical similarity. *Bioinformatics (Oxford, England)*, *31*(7), 1016–1024. https://doi.org/10.1093/bioinformatics/btu760
97. Poluri, K. M., ir Gulati, K. (2017). Biotechnological and Biomedical Applications of Protein Engineering Methods. K. M. Poluri ir K. Gulati (Sud.), *Protein Engineering Techniques: Gateways to Synthetic Protein Universe* (p. 103–134). Springer. https://doi.org/10.1007/978-981-10-2732-1_5
98. Porebski, B. T., ir Buckle, A. M. (2016). Consensus protein design. *Protein Engineering, Design and Selection*, *29*(7), 245–251. https://doi.org/10.1093/protein/gzw015
99. Qu, G., Li, A., Acevedo-Rocha, C. G., Sun, Z., ir Reetz, M. T. (2020). The Crucial Role of Methodology Development in Directed Evolution of Selective Enzymes. *Angewandte Chemie International Edition*, *59*(32), 13204–13231. https://doi.org/10.1002/anie.201901491

100. Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*(2), 257–286. https://doi.org/10.1109/5.18626
101. Rao, R., Meier, J., Sercu, T., Ovchinnikov, S., ir Rives, A. (2020). Transformer protein language models are unsupervised structure learners. *BioRxiv*, 2020.12.15.422761. https://doi.org/10.1101/2020.12.15.422761
102. Remmert, M., Biegert, A., Hauser, A., ir Söding, J. (2012). HHblits: Lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nature Methods*, *9*(2), 173–175. https://doi.org/10.1038/nmeth.1818
103. Repecka, D., Jauniskis, V., Karpus, L., Rembeza, E., Rokaitis, I., Zrimec, J., Poviloniene, S., Laurynenas, A., Viknander, S., Abuajwa, W., Savolainen, O., Meskys, R., Engqvist, M. K. M., ir Zelezniak, A. (2021). Expanding functional protein sequence spaces using generative adversarial networks. *Nature Machine Intelligence*, *3*(4), 324–333. https://doi.org/10.1038/s42256-021-00310-5
104. Riesselman, A. J., Ingraham, J. B., ir Marks, D. S. (2018). Deep generative models of genetic variation capture the effects of mutations. *Nature Methods*, *15*(10), 816–822. https://doi.org/10.1038/s41592-018-0138-4
105. Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., ir Fergus, R. (2021). Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, *118*(15). https://doi.org/10.1073/pnas.2016239118
106. Romero-Rivera, A., Garcia-Borràs, M., ir Osuna, S. (2016). Computational tools for the evaluation of laboratory-engineered biocatalysts. *Chemical Communications*, *53*(2), 284–297. https://doi.org/10.1039/C6CC06055B
107. Russ, W. P., Figliuzzi, M., Stocker, C., Barrat-Charlaix, P., Socolich, M., Kast, P., Hilvert, D., Monasson, R., Cocco, S., Weigt, M., ir Ranganathan, R. (2020). An evolution-based model for designing chorismate mutase enzymes. *Science*, *369*(6502), 440–445. https://doi.org/10.1126/science.aba3304
108. Russ, W. P., Lowery, D. M., Mishra, P., Yaffe, M. B., ir Ranganathan, R. (2005). Natural-like function in artificial WW domains. *Nature*, *437*(7058), 579–583. https://doi.org/10.1038/nature03990
109. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., ir Chen, X. (2016). Improved Techniques for Training GANs. *arXiv:1606.03498 [cs]*. http://arxiv.org/abs/1606.03498
110. Salinas, V. H., ir Ranganathan, R. (2018). Coevolution-based inference of amino acid interactions underlying protein function. *ELife*, *7*. https://doi.org/10.7554/eLife.34300
111. Seabold, S., ir Perktold, J. (2010). *Statsmodels: Econometric and Statistical Modeling with Python*. 92–96. https://doi.org/10.25080/Majora-92bf1922-011
112. Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, *27*(3), 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x
113. Sheldon, R. A., ir Pereira, P. C. (2017). Biocatalysis engineering: The big picture. *Chemical Society Reviews*, *46*(10), 2678–2691. https://doi.org/10.1039/C6CS00854B
114. Shi, W., Caballero, J., Theis, L., Huszar, F., Aitken, A., Ledig, C., ir Wang, Z. (2016). Is the deconvolution layer the same as a convolutional layer? *arXiv:1609.07009 [cs]*. http://arxiv.org/abs/1609.07009
115. Shin, J.-E., Riesselman, A. J., Kollasch, A. W., McMahon, C., Simon, E., Sander, C., Manglik, A., Kruse, A. C., ir Marks, D. S. (2021). Protein design and variant prediction using autoregressive generative models. *Nature Communications*, *12*(1), 2403. https://doi.org/10.1038/s41467-021-22732-w
116. Sievers, F., Wilm, A., Dineen, D., Gibson, T. J., Karplus, K., Li, W., Lopez, R., McWilliam, H., Remmert, M., Söding, J., Thompson, J. D., ir Higgins, D. G. (2011). Fast, scalable

generation of high-quality protein multiple sequence alignments using Clustal Omega. *Molecular Systems Biology*, *7*, 539. https://doi.org/10.1038/msb.2011.75
117. Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., ir Hassabis, D. (2017). Mastering the game of Go without human knowledge. *Nature*, *550*(7676), 354–359. https://doi.org/10.1038/nature24270
118. Smith, T. F., ir Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, *147*(1), 195–197. https://doi.org/10.1016/0022-2836(81)90087-5
119. Socolich, M., Lockless, S. W., Russ, W. P., Lee, H., Gardner, K. H., ir Ranganathan, R. (2005). Evolutionary information for specifying a protein fold. *Nature*, *437*(7058), 512–518. https://doi.org/10.1038/nature03991
120. Steinegger, M., Meier, M., Mirdita, M., Vöhringer, H., Haunsberger, S. J., ir Söding, J. (2019). HH-suite3 for fast remote homology detection and deep protein annotation. *BMC Bioinformatics*, *20*(1), 473. https://doi.org/10.1186/s12859-019-3019-7
121. Steinegger, M., Mirdita, M., ir Söding, J. (2019). Protein-level assembly increases protein sequence recovery from metagenomic samples manyfold. *Nature Methods*, *16*(7), 603–606. https://doi.org/10.1038/s41592-019-0437-4
122. Steinegger, M., ir Söding, J. (2017). MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, *35*(11), 1026–1028. https://doi.org/10.1038/nbt.3988
123. Steinegger, M., ir Söding, J. (2018). Clustering huge protein sequence sets in linear time. *Nature Communications*, *9*(1), 2542. https://doi.org/10.1038/s41467-018-04964-5
124. Steiner, K., ir Schwab, H. (2012). Recent advances in rational approaches for enzyme engineering. *Computational and Structural Biotechnology Journal*, *2*. https://doi.org/10.5936/csbj.201209010
125. Stourac, J., Dubrava, J., Musil, M., Horackova, J., Damborsky, J., Mazurenko, S., ir Bednar, D. (2021). FireProtDB: Database of manually curated protein stability data. *Nucleic Acids Research*, *49*(D1), D319–D324. https://doi.org/10.1093/nar/gkaa981
126. Strait, B. J., ir Dewey, T. G. (1996). The Shannon information entropy of protein sequences. *Biophysical Journal*, *71*(1), 148–155.
127. Subramaniya, S. R. M. V., Terashi, G., Jain, A., Kagaya, Y., ir Kihara, D. (2020). Protein Contact Map Denoising Using Generative Adversarial Networks. *BioRxiv*, 2020.06.26.174300. https://doi.org/10.1101/2020.06.26.174300
128. Suzek, B. E., Huang, H., McGarvey, P., Mazumder, R., ir Wu, C. H. (2007). UniRef: Comprehensive and non-redundant UniProt reference clusters. *Bioinformatics*, *23*(10), 1282–1288. https://doi.org/10.1093/bioinformatics/btm098
129. Takahashi-Íñiguez, T., Aburto-Rodríguez, N., Vilchis-González, A. L., ir Flores, M. E. (2016). Function, kinetic properties, crystallization, and regulation of microbial malate dehydrogenase. *Journal of Zhejiang University. Science. B*, *17*(4), 247–261. https://doi.org/10.1631/jzus.B1500219
130. Teng, S., Srivastava, A. K., ir Wang, L. (2010). Sequence feature-based prediction of protein stability changes upon amino acid substitutions. *BMC Genomics*, *11*(Suppl 2), S5. https://doi.org/10.1186/1471-2164-11-S2-S5
131. Thompson, J. D., Linard, B., Lecompte, O., ir Poch, O. (2011). A Comprehensive Benchmark Study of Multiple Sequence Alignment Methods: Current Challenges and Future Perspectives. *PLOS ONE*, *6*(3), e18093. https://doi.org/10.1371/journal.pone.0018093
132. Vihinen, M., Torkkila, E., ir Riikonen, P. (1994). Accuracy of protein flexibility predictions. *Proteins: Structure, Function, and Bioinformatics*, *19*(2), 141–149. https://doi.org/10.1002/prot.340190207
133. Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson,

J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … van Mulbregt, P. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, *17*(3), 261–272. https://doi.org/10.1038/s41592-019-0686-2

134. Waskom, M. L. (2021). seaborn: Statistical data visualization. *Journal of Open Source Software*, *6*(60), 3021. https://doi.org/10.21105/joss.03021
135. Watanabe, K., Ohkuri, T., Yokobori, S., ir Yamagishi, A. (2006). Designing thermostable proteins: Ancestral mutants of 3-isopropylmalate dehydrogenase designed by using a phylogenetic tree. *Journal of Molecular Biology*, *355*(4), 664–674. https://doi.org/10.1016/j.jmb.2005.10.011
136. Wilson, C. J. (2015). Rational protein design: Developing next-generation biological therapeutics and nanobiotechnological tools. *Wiley Interdisciplinary Reviews. Nanomedicine and Nanobiotechnology*, *7*(3), 330–341. https://doi.org/10.1002/wnan.1310
137. Wittmann, B. J., Yue, Y., ir Arnold, F. H. (2020). Machine Learning-Assisted Directed Evolution Navigates a Combinatorial Epistatic Fitness Landscape with Minimal Screening Burden. *BioRxiv*, 2020.12.04.408955. https://doi.org/10.1101/2020.12.04.408955
138. Wittmann, B. J., Johnston, K. E., Wu, Z., ir Arnold, F. H. (2021). Advances in machine learning for directed evolution. *Current Opinion in Structural Biology*, *69*, 11–18. https://doi.org/10.1016/j.sbi.2021.01.008
139. Xu, Y., Verma, D., Sheridan, R. P., Liaw, A., Ma, J., Marshall, N. M., McIntosh, J., Sherer, E. C., Svetnik, V., ir Johnston, J. M. (2020). Deep Dive into Machine Learning Models for Protein Engineering. *Journal of Chemical Information and Modeling*, *60*(6), 2773–2790. https://doi.org/10.1021/acs.jcim.0c00073
140. Zhang, C., Zheng, W., Mortuza, S. M., Li, Y., ir Zhang, Y. (2020). DeepMSA: Constructing deep multiple sequence alignment to improve contact prediction and fold-recognition for distant-homology proteins. *Bioinformatics*, *36*(7), 2105–2112. https://doi.org/10.1093/bioinformatics/btz863
141. Zheng, W., Li, Y., Zhang, C., Pearce, R., Mortuza, S. M., ir Zhang, Y. (2019). Deep-learning contact-map guided protein structure prediction in CASP13. *Proteins: Structure, Function, and Bioinformatics*, *87*(12), 1149–1164. https://doi.org/10.1002/prot.25792