Bakteriális patogén és ember közötti molekuláris hálózatok vizsgálata

Horváth Balázs

2015

1. Tartalomjegyzék

2. Rövidítésjegyzék

PPI - protein protein interaction

3. Bevezetés

3.1. A bél mikrobióta fontosságának ismertetése

Miért van szükség a bél mikrobióta vizsgálatára?

A humán bél mikrobióta egy komplex ökoszisztéma. A mikrobiomot alkotó sejtek száma nagyjából a humán szomatikus és csírasejtek összegének tízszerese. A bél mikrobiom mind metabolikusan, mind immunológiailag komplex kapcsolatban áll az emberrel. [?] Eddig több mint három millió nem redundáns mikrobiális gént sikerült kimutatni az emberben [?]. Ez a nagy genetikai állomány lehetővé teszi, hogy olyan metabolikus folyamatok játszódjanak le a humán bélben, melyeket az emberi sejtek nem képesek végrehajtani. [?] A bél mikrobióta felelős bizonyos glikánok, aminosavak és xenobiotikumok metabolizmusáért valamint rövid láncú zsírsavak (*short chained fatty acids - SCFA*-k), vitaminok és kofaktorok termeléséért. A gazda által meg nem emésztett poliszacharidok bontását a bél mikrobióta végzi, mely folyamat eredményeképpen olyan rövid láncú zsírsavak keletkeznek mint az acetát, proprionát és vajsav. [?]

A bélflóra kulcsszerepet játszik az immun-homeosztázis fenntartásában. Az immunrendszerrel bakteriális mintázatokat észlelő receptorokon keresztül és GPCR-ek által van kapcsolatban. A mikroorganizmusok által termelt SCFA-k képesek GPCR-eken keresztül sejtszignalizáció indítására. A veleszületett immunrendszer nagy részét alkotó monociták és neutrofil granulociták rendelkeznek GPR43 receptorral, mely szintén SCFA érzékeny, tehát a bélflóra metabolitokon keresztül is kapcsolatban áll az immunrendszerrel. [?]

A bélflóra hatással van még a gazda metabolizmusára is. Az *Eubacterium spp.* által oligoszacharidokból képzett vajsav részt vesz az emberi szervezet energia egyensúlyának szabályzásában. [?] Az enteroendokrin sejtek és az adipociták is rendelkeznek a GPR41 receptorral mely vajsavra és proprionátra is érzékeny. Adipocitáknál ez a GPR41 szignalizáció *leptin* elválasztást eredményez. [?] A vajsav segít a karcinogenezis kivédésében mivel apoptózis indukáló és proliferáció gátló hatása van. Éppen ezen okokból a bél mikrobióta tekinthető egy új metabolikus szervnek is. [?] Kapcsolatok mutathatók ki a bél mikrobiom megváltozása és olyan betegségek között mint az IBD (*inflammatory bowl disease*), elhízás vagy ~~a~~ különböző rák típusok. [?]

A bél mikrobióta vizsgálatának módszerei

A mikrobióta vizsgálatát elsősorban a különböző meta omikák eszköztárával közelítik meg. Ezek közül is a ~~legfőbb~~ legfontosabb eszköztár a metagenomika, de alkalmaznak már metabolomikai, metatranszkriptomikai és metaproteomikai megközelítést is. A metagenomikai vizsgálatok során a környezetből származó mintát megfelelő előkészítés után közvetlenül *shotgun* szekvenálásnak vetik alá. [**?**]

Quin és ~~társai2010-re~~ társai 2010-re meghatározták a minimális bél metagenomot. A vizsgálat során Illumina GA short-read alapú technológiával 124 egy kohortba tartozó nordikus és mediterrán személy székletmintáját elemezték. Az ebből kinyert 576,7 gigabázásnyi DNS-ből 3,3 millió nem redundáns mikrobiális gént mutattak ki. Az így kimutatott gének az emberi genom százötvenszeresét teszik ki. A minták egészére jellemző, hogy a bennük található gének két fő részre osztható: A legnagyobb csoportba (86%) a sűrűn előforduló mikrobiális gének, míg a másik fő csoportba pedig a kifejezetten a humán bélflórára jellemző mikrobiális gének tartoznak. Az összes személyből származó vizsgált génhalmaz 99,1%-a *Eubacteria*, 0,8%-a *Archea* és a fennmaradó 0,1%-a pedig vegyesen *Eucaryota* és virális eredetű. A bakteriális eredetű gének összesen 1000-1150 uralkodó baktériumfajhoz tartozhatnak, ami személyenként kb. 160 domináns fajt jelent. A személyekre jellemző nagyjából 160 uralkodó baktériumfaj listái között a személyeket összevetve nagyfokú hasonlóság figyelhető meg. Egy adott személy bél metagenomjának minimálisan 40%-a megtalálható a minták legalább felében. A közelítőleg ezer fajból 75 faj található meg a minták több mint felében és 57 faj van ami a minták nagyobb mint 90%-ban kimutatható. [**?**]

3.2. A szakirodalomban publikált gazda patogén hálózatok

!TODO

3.3. A Humán-Salmonella kapcsolat ismertetése és hatása az autofágiára

Salmonella spp.

A *Salmonell*ák olyan Gram-negatív patogének melyek az állatok széles skáláját képesek fertőzni. A tudomány jelenleg több ezer szerotípust ismer, melyek két fő típusra oszt-

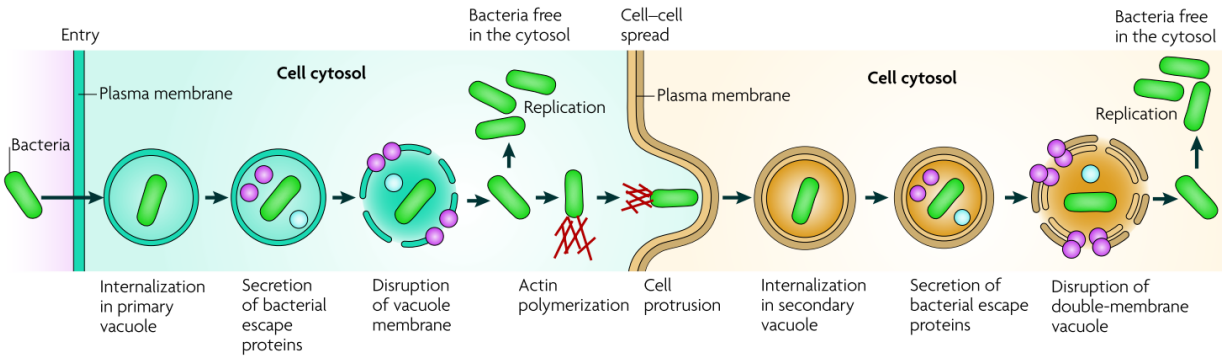
hatók. Az egyik fő típus a *Typhoid*, ebbe a csoportba tartozik a *Typhi* és *Paratyphi* melyek kifejezetten embert fertőznek. A másik fő csoport a *Non-typhoid* amelybe tartozó baktériumok már széleskörű gazdaspecificitással rendelkeznek.

A fertőzés kontaminált étel vagy folyadék fogyaztásával történik. A *Salmonell*ák az alacsony pH és oxidatív stressz ellen adaptív toleranciával rendelkeznek, így képesek eltűrni a gyomor savasságát és a veleszületett immunrendszer egyéb hatásait. A vékonybélbe jutva az epithélium sejtjeit fertőzik. Fő célpontjaik a *microfold* (*M cells*) sejtek, melyek fő feladata, hogy pinocitózissal mintákat vegyenek a középbél atnigénjeiből és ezt antigén prezentáló sejteknek adják. Azonban a *Salmonell*ák úgynevezett baktérium-közvetített endocítózissal képesek még a nem fagocita típusú enterocitákba is bejutni [**?**]

A *Salmonella* életciklusa

Az intracelluláris baktériumok életciklusa általánosan három stádiumra osztható: A bejutáshoz használt vakólum elhagyása, replikáció a citoszólban és a citoszólikus veleszületett immunitás elemeinek manipulációja. A *Salmonella* az úgynevezett *trigger* mechanizmussal jut be a sejtbe. A mechanizmus során a baktérium olyan fehérjéket juttat be az eukarióta sejtbe, melyek képesek a sejtvázzal kölcsönhatni. Ezek a bakteriális effektor-fehérjék nagyfokú sejtváz-átrendeződést váltanak ki az eukarióta gazdában. A folyamat végén a baktérium egy vakólummal határolva a sejt belsejébe kerül. [**?**] Ezt a képletet a szakirodalomban SCV-nek nevezik (*Salmonella containing vacuole*). [**?**]

A fagocitózis végeztével a *Salmonell*ák átesnek egy úgynevezett bakteriális felszín átformázáson (*bacterial surface remodeling*). A folyamat során represszálódnak az olyan bakteriális gének expressziója amit a gazda könnyen fertőzési jelnek tekinthet. Ilyen gének például a SPI1, a T3SS és a flagellin. Mindezek mellett megváltozik a baktériumok felszíni lipopoliszacharid mintázata is. [**?**]



1. ábra. **Az intracelluláris baktériumok életciklusa**

Bejutáskor a baktériumok egy elsődleges vakólumba érkeznek. A sejt a belsejében a mikróbák olyan fehérjéket szekretálnak, melyek felbontják az őket határoló elsődleges vakólum membránját. A legtöbb intracelluláris baktériumra jellemző, hogy befolyásolni tudja az aktin polimerizációt és ezáltal képes az intra- és intercelluláris mozgásra. A szomszédos sejtbe átjutott baktériumok egy másdlagos memránburokba kerülnek, melyet ugyancsak felbontanak. !TODO kép magyarítás és formázás, jobban látható feliratok

Normális körülmények között a vakólum pH-ja mindaddig fokozatosan csökken amíg érett degradatív fagolizoszómává nem válik. A baktériumok kétféleképpen képesek életben maradni ebben a környezetben: A vakólum-lizoszóma fúzió gátlásával, vagy a fagolizoszóma összetételének aktív módosításával. [**?**] A szakirodalomban még nincs kialakult álláspont arról, hogy a *Salmonell*ák melyik mechanizmust használják. Bizonyítottan képesek életben maradni, olyan SCV-ben mely már fuzionált a lizoszómával, viszont a fő útvonal valószínűleg a vakólum savanyítási folyamatának késleltetése lehet. [**?**]

Az SCV-n belül a *S. typhimurium* képes a replikációra. A hármas típusú szekréciós rendszer segítségével a baktériumsejtek olyan anyagokat tudnak kibocsájtani, melyek lehetővé teszik az SCV-ből kijutást és citoplazma inváziójét. [**?**]

A hármas típusú szekréciós rendszer (T3SS vagy TTSS)

A T3SS evolúciósan a flagelláris export rendszerrel mutat rokonságot. Jelenléte esszenciális ahhoz, hogy a *Salmonella* képes legyen a fertőzésre és gazda sejtjeinek kolonizálására. A T3SS felelős a baktérium virulencia vagy effektor fehérjéinek átviteléért. Az effektorok az eukarióta sejtbe jutva megváltoztatják annak sejtfunkcióit. Az virulenciafehérjék átalakítják a gazda citoszkeleton architektúráját, membrán anyagáramlását, szignál transz-

dukcióját és citokin expresszióját, ezzel segítve a baktériumok túlélését és további kolonizációját. [?]

A *Salmonella* és az autofágia kapcsolata

Az autofágia egy intracelluláris katabolikus folyamat melynek szerepe van a fehérjeaggregátumok és károsodott sejtorganellumok eltávolításában és a veleszületett immunrendszer működésében. A *xenofágia* az autofágiának azon formája mely során az intracelluláris baktériumok és vírusok szelektív felismerése és lebontása történik. A szelektív felismerésért az autofágia adaptor fehérjéi felelősek. Ilyen receptor fehérje például a p62 (SQSTM1), a NDP52, optineurin (OPTN) és az NBR1. Az előbb felsorolt receptorok a szubsztrátjuk megkötése után kargo adaptorként viselkednek az LC3 (ATG8) számára. *Salmonella fertőzéskor* a sérült SCV-ből kilépett baktériumok sejtfelszíni fehérjéi poliubiquitin borítást kapnak amit a kargo adaptor fehérjék érzékelnek. *S. typhimurium* fertőzéskor a poliubiquitinált baktériumokat NDP52 és a p62 is felismeri. Az így megkötött baktériumok xenofágia útján eltávolítódnak. [?]

3.4. Ökológiai hálózatok elemzésére használt topológiai mérőszámok

Miért van szükség topológiai mérőszámokra?

A konzervrációs biológia az élettudományok azon ága mely a Föld biodiverzitásának megőrzésével foglalkozik. Mivel az összes faj védelme nem megoldható, ezért szükségessé vált olyan fajok kiválogatása melyek kiemelt figyelmet igényelnek konzervációs biológiai szempontból. [?] Az 1990-es évek előtt a védelemre való kiválasztás fő szempontja a faj ritkasága volt. A fajok ilyen alapú szelekciója nem veszi figyelembe hogy például az adott taxon kulcsszerepet játszik-e az ökoszisztéma funkciók ellátásában. [?]

Kulcsfajok

1966-ban Robert Paine megalkotta a kulcsfaj koncepciót(*keystone species*). Megfigyelte hogy ha kiesik a Kaliforniai sziklás tengerparti közösségből a *Piaster ochraceus* csúcsragadozó tengeri csillag akkor az egész közösség fajösszetétele összeomlik. A mai legelfogadottabb kulcsfaj definíció szerint ezek olyan fajok, melyek ökológiai hatása aránytalanul nagy az abundanciájukhoz képest. A fogalommal kapcsolatban azonban további kérdések merülnek fel: Milyen hatás számít nagynak? Pontosan mekkora biomassza hányad után

mondható az adott faj ereje aránytalannak? [**?**] Ez utóbbi kérdések megválaszolásához szükség van olyan mérőszámokra, melyek segítségével kvantitatívvé tehető egy adott faj ökológiai fontossága. Másrészt így lehetővé válik a fajkiválasztás során a szubjektivitás csökkentése is. Az ilyen mérőszámok használatával objektív fontossági sorrendet lehet felállítani az adott élőhelyen előforduló taxonok között. [**?**]

Rangsorolásra használt topológiai mérőszámok az ökológiában

Ma már a kulcsfajok kiválasztása részben ökológiai interakciós hálózatok elemzése alapján történik. A használt hálók kizárólag biotikus-biotikus (faj-faj) kapcsolatokat tartalmaznak. Erre azért van szükség, mert például minden élőlény összekötésben áll a detritusszal és ez eltorzítaná az analízis eredményét. Sőt ilyen esetben a detritusz maga is struktúrális kulcsfajnak számítana. Egy adott fajnak az ökológiai interakciós hálóban betöltött szerepét pozicionális fontossági mérőszámokkal, vagy más néven centralitási indexekkel lehet jellemezni. A konzervációs biológiában sokfajta ilyen mérőszámot használnak, melyeknek közös tulajdonsága, hogy mindegyik valamilyen egyedi tulajdonságra fekteti a hangsúlyt és az alapján rangsorolja a hálózatban szereplő fajokat. Ilyen eltérés lehet két index között például, az hogy az egyik egy adott pont lokális kapcsolati mintázatára, míg a másik az egész hálózatra vonatkozó hatását számszerűsíti. Adott hálóra különböző mérőszámok eltérő fajsorrendeket adnak, de a hasonló tulajdonságok figyelembevételén alapuló mérőszámok között felállíthatók konszenzus fák. [**?**]

Főbb topológiai mérőszámok

Normalised degree - D

Az adott ponttal kapcsolódó pontok száma elosztva a hálózat összes pontjának számával. [**?**]

Closeness centrality - CC vagy C

A pontok száma elosztva az adott pontból eredő azt minden más ponttal összekötő legrövidebb topológiai távolságok összegével. [**?**] Ez a mérőszám megmutatja, hogy egy adott pontnak mekkora az átlagos távolsága a hálózat összes többi pontjától. Az index kicsi szám olyan pontokra melyek rövid legrövidebb útvonalakon vannak a többi ponttal összekötve. Az ilyen pontok valószínűleg könnyebben elérnek más pontokat vagy nagyobb

hatást tudnak gyakorolni más pontokra. Adott i pont átlagos legrövidebb távolságát a többi ponttól a következőképpen lehet kiszámolni: [**?**]

$$\ell_i = \frac{1}{n-1}\sum_j d_{ij} \qquad \text{vagy,} \qquad \ell_i = \frac{1}{n}\sum_{j(\neq i)} d_{ij} \tag{1}$$

Ahol:

ℓ_i : Az i pont átlagos legrövidebb távolsága a hálózat többi pontjától.

d_{ij} : Az az i pontot a j ponttal összekötő legrövidebb útvonal (geodézikus útvonal) pontjainak száma.

n: A hálózat pontjainak száma.

A két számítás között stratégiai különbség van. A baloldali egyenlet azt feltételezi, hogy adott pontnak önmagára mért hatása nem releváns a hálózat működésének szempontjából. Azonban még erre az esetre is jellemző, hogy mivel definíció szerint a d_{ii} távolság 0, ezért az összeget ez az érték nem növeli csupán az osztót. [**?**]
Az ℓ_i érték önmagában még nem centralitási index, mert kis számokat ad a magas központiságú pontokra. Ahhoz, hogy megkapjuk a *Closeness Centrality*-t az ℓ_i inverzét kell vennünk: [**?**]

$$C_i = \frac{1}{\ell_i} \tag{2}$$

Betweenness centrality - BC

A vizsgálni kívánt ponton áthaladó a hálózat többi pontpárját összekötő legrövidebb utak összege elosztva a hálózat többi pontpárját összekötő összes legrövidebb út összegével. [**?**] Ez a mérőszám azt mutatja meg, hogy egy adott pont milyen arányban szerepel a többi pont között futó útvonalakban. A *betweenness centrality* vagy röviden *betweenness* olyan hálózatok jó jellemzője, melyekben valamilyen természetű „áramlás" folyik a pontok között. Ha feltételezzük, hogy egy ilyen hálózat minden kapcsolata között az áramlás során ugyanannyi kicserélődés történik egy egységnyi idő alatt és a kicserélődés a legrövidebb útvonalakon folyik, akkor az összes geodézikus útvonalon is azonos rátával történik az áramlás. Ez azt jelenti, hogy egy adott ponton átmenő áramlás mennyisége arányos azzal, hogy a hálózat legrövidebb útvonalainak milyen arányában szerepel. [**?**]

Topological importance - TI^n

Ez egy teljesen topológiai alapú mérőszám mely összegzi az egy adott pontból kiinduló összes lehetséges n lépéshosszúságú útvonal hatását. A hálózat összes direkt kapcsolatára kiszámítható azok topológiai erőssége:

$$d_{X,Y} = \frac{1}{x} \tag{3}$$

Ahol:

$d_{X,Y}$: Az Y pont hatása X pontra.

x : Az X pont első szomszédainak száma.

Az így kiszámolt közvetlen kapcsolatok hatását egy mátrixban lehet ábrázolni, melynek indexelése a populációdinamika konvencióit követi: d_{ij} jelenti a j pontnak az i pontra gyakorolt hatását. Adott direkt kapcsolat hatásának nagysága a kapcsolat irányától is függ, tehát d_{ij} nem feltétlenül ugyanakkora mint d_{ji}. Egy n lépés hosszú útvonal erejét az ezt alkotó direkt kapcsolatok hatásának szorzataként értelmezzük:

$$d^n_{p_{XY}} = \prod_{i=1}^{n-1} d^1_{i,i+1} \tag{4}$$

Ahol:

p_{XY}: Útvonal amire igaz hogy $p \in \{X$ és Y közötti n lépés hosszúságú útvonalak$\}$

$d^n_{p_{XY}}$: Az X és Y pontok közötti n lépés hosszú p útvonal ereje.

$d^1_{n,n+1}$: Az útvonal i és $i+1$-ik pontja közötti direkt kapcsolat erőssége

Ez alapján egy Y pont hatása X-ra n lépés távolságban:

$$d^n_{XY} = \sum d^n_{p_{XY}} \tag{5}$$

Ahol:

p_{XY}: Útvonal amire igaz hogy $p \in \{X$ és Y közötti n lépés hosszúságú útvonalak$\}$

d^n_{XY}: Az összes Y pontból eredő és X-ben végződő n hosszúságú útvonalak erejének összege.

Mivel a direkt kapcsolatok ereje függ a kapcsolat irányától, így a TI tükrözi a kapcsolat asszimmetrikusságát is. Egy adott pontra TI^{n} a következő képen számítható ki:

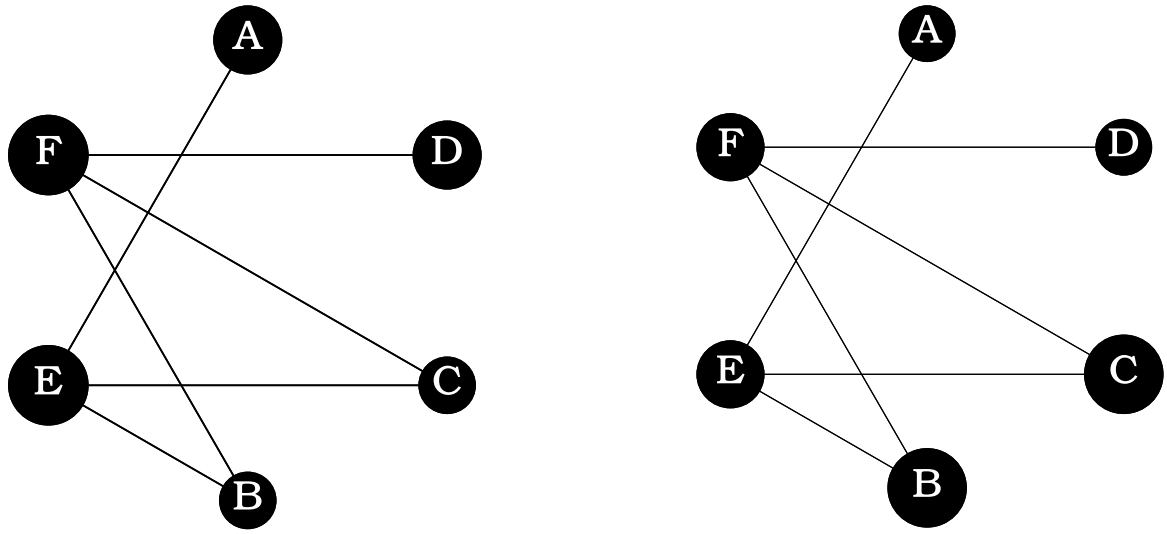
$$\mathrm{TI}_A^n = \sum d_{j,A}^n \tag{6}$$

Ahol:

TI_A^n: A pont n lépésre számított topológiai fontossága.

$d_{j,A}$: A és j pont közötti n hosszúságú útvonalak ereje.

A TI^n-t a hálózat összes pontjára ki lehet számítani és ez alapján sorrendet lehet felállítani a nódusok között.



2. ábra. d ***(bal)*** **és** $\mathbf{TI^2}$ ***(jobb)*** **szemléltetése ugyanazon a példagráfon**
A pontok átmérője arányos az adott nódusra kiszámolt d (direkt vagy közvetlen topológiai kölcsönhatás) és TI^2 (topológiai fontosság két lépésre) értékekkel. [**?**] alapján módosítva.

Az 2. ábrán látható példagráfra rendre felírhatóak a közvetlen kölcsönhatások (d) és a két lépésnyire közvetített indirekt kölcsönhatások (d^2) értékeit tartalmazó mátrixok:

$$
\begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \end{array}
\begin{array}{c}
\begin{array}{cccccc} A & B & C & D & E & F \end{array} \\
\begin{bmatrix}
0 & 0 & 0 & 0 & \frac{1}{3} & 0 \\
0 & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} \\
0 & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} \\
0 & 0 & 0 & 0 & 0 & \frac{1}{3} \\
1 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\
0 & \frac{1}{2} & \frac{1}{2} & 1 & 0 & 0
\end{bmatrix} \\
d \text{ értékek}
\end{array}
,
\begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \end{array}
\begin{array}{c}
\begin{array}{cccccc} A & B & C & D & E & F \end{array} \\
\begin{bmatrix}
\frac{1}{3} & \frac{1}{6} & \frac{1}{6} & 0 & 0 & 0 \\
\frac{1}{3} & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\
\frac{1}{3} & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\
0 & \frac{1}{6} & \frac{1}{6} & \frac{1}{3} & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{2}{3} & \frac{1}{3} \\
0 & 0 & 0 & 0 & \frac{1}{3} & \frac{2}{3}
\end{bmatrix} \\
d^2 \text{ értékek}
\end{array}
$$

Az ábrázolt mátrixok elrendezése követi a populációdinamikai konvenciókat, tehát például $d_{BF} = \frac{1}{2}$ azt jelenti, hogy F pont a B-re $\frac{1}{2}$ erővel hat. Mindkét mátrixra érvényes az, hogy az adott oszlop értékeinek összege egy. Ez a tulajdonság a d érték definíciójából fakad. d azt mutatja meg, hogy adott pont a cél pont kapcsolatainak hányad részét adja. Ezáltal minden pont egy egységnyi hatást kap ami eloszlik a vele kapcsolatban álló pontok között. [**?**] Ezt a hatást jól szemlélteti az 2. ábra jobb oldali része amin látható, hogy B, C, D és A pontok kimenő hatása kisebb, mivel célpontjaik sok hatást fogadnak.

Ugyancsak mindkét mátrixra jellemző, hogy a sorok összege azt mutatja meg, hogy egy adott pont mennyire erős kölcsönható, tehát mekkora TI^n értéke. Például B pont két lépés távolságban összesen $\frac{4}{3}$ erővel hat, ez alapján erősebb kölcsönhatónak mondható mint az A pont a maga $\frac{2}{3}$ értékű összesített kimenő két lépés hosszú hatásaival. [**?**]

Az 2. ábrán az is jól megfigyelhető, hogy C pont a gyengébb közvetlen kölcsönhatók közé tartozik. Ugyanakkor mivel a C-ből eredő két lépéses útvonalak erős elsődleges kölcsönhatókon keresztül érik el végpontjaikat, ezáltal két lépés távolság viszonylatában már C is az erős kölcsönhatók közé tartozik.

Az $n > 1$ lépésszámú d^n értékeket tartalmazó mátrixokban már egy adott pont indirekt hatása önmagára is kiterjedhet. Páros számú lépések esetén viszont mindenképpen felírhatók olyan útvonalak melyeken a pont eléri önmagát, [**?**] vagyis $d^n_{X,X} \neq 0$ ha n $\in$ $\{\ 2k : k \in \mathrm{Z}\ \}$. Az 2. ábrán látszik, hogy például az F pont két lépés távolságban a következő útvonalakon hat önmagára: $F \to B \to F$, $F \to C \to F$ és $F \to D \to F$.

Weighted Topological Importance - WI^n !TODO

4. Célkitűzések

A diplomamunka célja

A diplomamunkám célja egy több adatbázisból integrált fehérje-fehérje kapcsolatokat tartalmazó humán-*Salmonella* gazda-patogén hálózat létrehozása különböző adatbázisok alapján és az elkészült háló topológiai elemzése.

Az elkészítendő hálózatnak a következőeket kell tartalmaznia:

1. Kurált *H. sapiens* fehérje-fehérje kapcsolatok
2. Kurált *Salmonella* fehérje-fehérje kapcsolatok
3. *H. sapiens* és *Salmonella* közti prediktált fehérje-fehérje kapcsolatok

A topológiai elemzés során kapott adatok alapján véleményt szeretnék alkotni arról, hogy felhasználhatók-e az ökológiában fajok közti kapcsolatok vizsgálatára használt tisztán topológiai adatokon alapuló mérőszámok a molekuláris kapcsolati hálók elemzésére. Valamint, hogy az így előállított rangsorok mennyire korrelálnak a jelenleg használt ~~Salmonella~~ *Salmonella* és humán bélsejteket vizsgáló módszerek eredményeivel.

A célok eléréséhez tervezett feladatok

1. Program írása mely képes az ARN (Autophagy Regulatory Network) adatbázis kurált humán autofágia specifikus fehérje-fehérje kapcsolati rétegének („ARN core”) MiTab SQL formátumra átalakítására.
2. Program írása mely képes a Salmonet adatbázis kurált *Salmonella* fehérje-fehérje kapcsolati hálózatának MiTab SQL formátumra átalakítására.
3. Programok írása melyek képesek a Krishnadev és Skhirshagar féle humán-*Salmonella* fehérje-fehéjre kapcsolati predikciók MiTab SQL formátumra alakítására.
4. Program írása mely képes a létrehozott MiTab SQL fájlokban a fehérjék azonosítójának *Uniprot* azonosítóra fordítására.

5. Program írása mely képes a már csak *Uniprot* azonosítókat tartalmazó adatbázisok összeejtésére.

6. Program írása mely képes hálózatokban a megadott útvonalhosszra kiszámolni a topológiai fontosságot.

7. A kapott adatsorok értékelése, hálózatok ábrázolása, biológiai relevancia keresése.

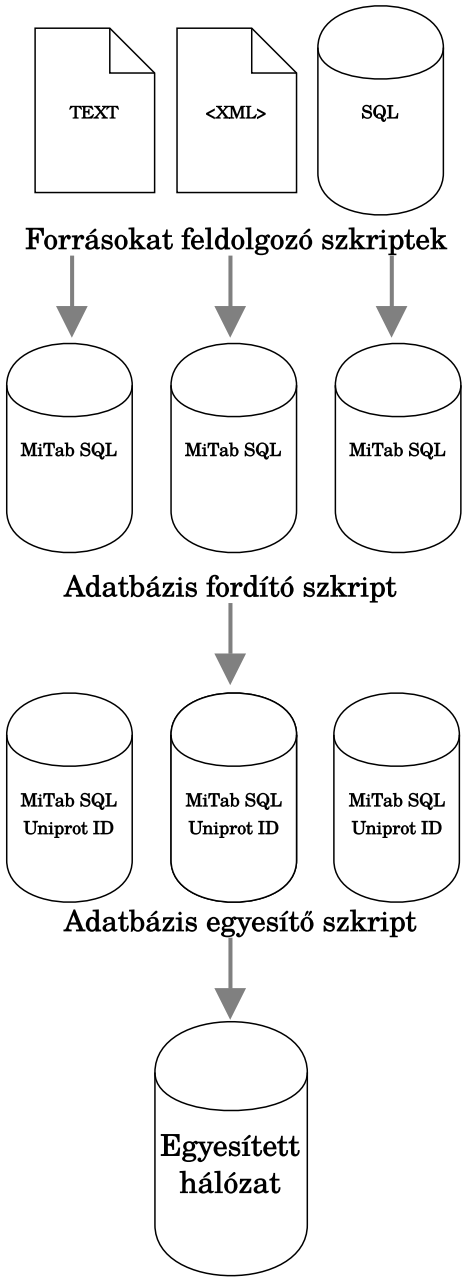
5. Források és módszertan

5.1. Informatikai módszerek

A problémák megoldására használt programnyelvek

A teljes adatbázisok feldolgozására valamint az adatbázisokból származó adatok rendszerezésére és megfelelő formátumúra alakítására *Python* programnyelven írtam szkripteket. A diplomamunkám során a *Python* 2.7-en és ~~3-on~~ 3.4-on futtatható szkripteket ~~is~~ alkalmaztam. A fehérjék azonosítójának fordítását végző szkriptek egyike témavezetőm, Kadlecsik Tamás *Javascript*ben írt fordítószkriptjének kismértékű módosítása.

A diplomamunkám során alkalmazott szkriptek egy részét a *Signalink* 3 szignalizációs adatbázis ~~nulladik~~ kézzel gyűlytött és ~~harmadik rétegének~~ külső adatokat tartalmazó rétegeinek (L0 és L3) létrehozásakor készítettem. Mivel a *Signalink* ~~nulladik rétege~~ rétegei is több adatbázisból integrál fehérje-fehérje interakciókat, így az ott alkalmazott munkafolyamat felhasználható volt a diplomamunkám gazda-patogén hálózatának létrehozásakor is. *(3.* ábra) A humán-*Salmonella* hálózat szerkezete azonban különbözik a *Signalink* ~~3 nullás rétegétől~~3-étól, mert például prediktált éleket is tartalmaz Nem ez a legnagyobb különbség, inkább
A két hálózat különbségei miatt, a diplomamunkámban az adatokat kezelő algoritmusok bár hasonlítanak a *Signalink*et létrehozókra, de azokkal nem azonosak.



3. ábra. **A hálózat létrehozásának folyamata**

A különböző forrásokból származó adatok esetén először a forrás formátumokat feldolgozni képes szkriptek átalakítják azokat a belső szabványként használt MiTab SQL formátumra. Általában a különböző adatforrások különféle azonosítókkal illetik a komponenseiket. Ahhoz, hogy több hálózatot egyesíteni tudjunk, szükség van arra, hogy egy adott biológiai entitás csak egyfajta azonosítóval szerepeljen. A fordító szkript MiTab SQL fájlból olyan MiTab SQL fájlt ~~gyárt~~készít, amiben az elsődleges azonosító már a kívánt, esetemben *Uniprot* azonosító. Legvégül az adatbázis egyesítő szkript úgy „összefűzi" a különböző hálózatok pontjait és éleit, hogy ne legyen benne redundáns információ.

Az adatok tárolása

Az adatok ideiglenes tárolására, már a *Signalink 3* készítése óta témavezetőm, Kadlecsik Tamás javaslatára *SQLite 3* adatbázis fájlokat alkalmazunk.

Az *SQLite 3* egy nyílt forráskódú, *C* nyelven írt API-val rendelkező, beágyazott relációs adatbázis motor. Az SQL sztenderd szintaxisának nagy részét tartalmazza. Sok népszerű programnyelv rendelkezik már beépített *SQLite* támogatással, ilyen például a *Python* is. [**?**]

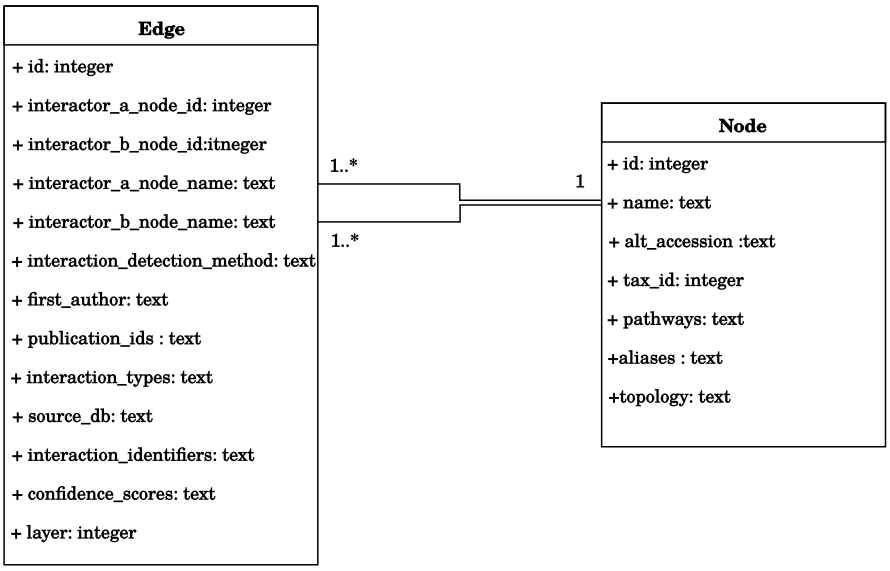
Az adatok ilyen módú tárolása lehetővé teszi azok gyors szűrését, kategorizálását és átalakítását SQL parancsok segítségével. Ilyen módon még azelőtt gyorsan információkat nyerhetünk nagy méretű hálózatokról, mielőtt azokat olyan jóval lassabb működésű hálózatkezelő programokkal elemezni kezdenénk mint például a *Cytoscape*. Az *SQLite* adatbázisfájlok másik előnye, hogy rendelkezésünkre áll az SQL nyelv. Mivel SQL parancsok segítségével gyorsan kezelhetőek a feldolgozott adatok, így csak ritkán van szükség adatmanipulálási célból egy újabb szkript írására. Amennyiben mégis szükséges újabb szkript írása, a legtöbb szkriptnyelv rendelkezik valamilyen *SQLite* adatbázis kezelési opcióval. Nagy méretű és mennyiségű biológiai adatot tároló *SQLite* fájlban a keresés is igen gyorsan megoldható az adatbázis beindexelésével, sőt még gyorsabb keresés is megvalósítható az indexelt táblák memóriába csatolásával. Az *SQLite* segítségével könnyen lehet importálni és exportálni a legtöbb népszerű adattárolási formátumba.

Ezen indokok miatt használjuk a csoportomban az *SQLite* adatbázisokat az adatok köztes tárolására, a szöveges fájlok helyett.

Hálózatok tárolására a csoport által létrehozott MiTab SQL formátumot használtam. A MiTab SQL egy *SQLite 3*-ban tárolt a PSI-MI Tab formátummal közel megegyező adatstruktúra. A PSI-MI Tab egy *HUPO Proteomics Standards Initiative* (PSI) szervezet által meghatározott proteomikai adatok tárolására használt formátum. A PSI-MI Tab formátum specifikációja a szervezet honlapján elérhető. Hivatkozás! van hozzás cikk!

A pontok és az élek külön táblában vannak letárolva az adatbázisban, így a PSI-MI Tab specifikáció pontra és az élre vonatkozó tulajdonságai a megfelelő táblába kerülnek. A MiTab SQL táblák oszlopai azonban nem teljesen egyeznek a PSI-MI Tab kategóriákkal. Ilyen különbség például, hogy a MiTab SQL nem használ néhány opcionális PSI-MI kategóriát viszont tartalmaz a PSI-MI-re nem jellemző tulajdonságokat is mint a topológia. Az éleket tartalmazó táblában a forrás (*interactor_a_node_name*) és a cél pont név oszlopa

a *node* tábla azonosító oszlopának idegen kulcsai. (4. ábra)



4. ábra. **A MiTab SQL sémája**

Verziókövetés

Diplomamunkám készítése során a *Git* verziókövető rendszert használtam, melynek tartalmát a web-alapú *GitHub* tárhely szolgáltatásra töltöttem fel. A diplomamunkám GitHub tárhelyén (**??**. hivatkozás) a következő általam írt kódok érhetők el:

- A 3. ábrán ábrázolt munkafolyamatot lebonyolító szkriptek
- A fordításhoz használt adatbázist megépítő szkript (Kadlecsik Tamás szkriptje alapján)
- A fordítást végző szkriptek
- Az adatbázisokat összeejtő szkriptek
- A MiTab SQL formátumot kezelő osztály
- A topológiai elemzést véző szkript
- Az adatszűrésre használt SQL szkriptek

Tesztelés A bonyolultabb algoritmusok esetén *egységteszteket (unit test)* alkalmaztam. A topológiai fontosságot kiszámító *Python* osztály összes metódusának működését ilyen módon ellenőriztem. A teszteléshez a *pyhthon* saját *unittest* nevű csomagját használtam.

5.2. Az források feldolgozásának eszközei

MiTab SQLite adatbázis API

A *PsimiSQL* egy *Python* 2.7-es szintaxisban írt osztály, melyet még a *Signalink 3* összeállításához ~~kezdtem el készíteni~~készítettem, de azóta más projektekben is használtam és továbbfejlesztettem. A *PsimiSQL* segítségével a molekuláris biológiai hálózatok könnyen átalakíthatók MiTab SQLite adatbázisokká. Az osztály számos függvényével megkönnyíti a MiTab SQLite adatbázisok kezelését *Python* alól. Ilyen függvény például a redundáns adatok képzését gátló *insert_unique_node()* mely ellenőrzi, hogy az adott hálózatban szerepel-e már az importálni kívánt pont. Az osztály példányosításakor a memóriában létrejön egy példányhoz kötött MiTab SQL sémával rendelkező SQLite 3 adatbázis. Az adatbázis benépesítése és az adatok keresése tehát nagy sebességgel történik. Az osztálynak vannak függvényei melyekkel könnyen importálni és exportálni lehet MiTab SQL adatbázis fájlokat.

A szótárak építése és a fordítás

Ahhoz hogy a feldolgozott forrásokat össze lehessen fűzni egy nagy ~~gráfba~~ adat szettbe, szükség van arra, hogy a hálózatokban ne szerepeljen ugyanaz a biológiai entitás más azonosítóval. Ennek érdekében a mindegyik hálózat fehérjéit a legfrisebb *Uniprot* adatbázis azonosítókra fordítottam. Ehhez két szkriptet kellett írnom.

A Salmonet és az ARN már eleve *Uniprot* azonosítókat használ. Azonban az *Uniprot* adatbázis állandó frissítései miatt, fenn áll a lehetőség, hogy nem egy időben készült fájlok ugyanarra a fehérjére más *Uniprot* azonosítót használnak. Egy másik hibaforrás az lehet, hogy a *Uniprot* adatbázis egy fehérjét több azonosítóval is tárol. Előfordulhat, hogy egy fehérje többször is szerepel csak más *Uniprot* azonosítókkal. Amikor egy fehérjét beletesznek a *Uniprot* adatbázisba, akkor kap egy elsődleges azonosítót. Primer azonosítót kapnak még olyan fehérjék is, melyek már benne voltak az adatbázisban de később külön izoformákra lettek szétválasztva. Új elsődleges azonosítót kapnak olyan fehérjék is melyeket több vélt fehérjéből egyesítettek. Minden ilyen művelet után, a legfrissebb elsődleges azonosító marad az új elsődleges, az összes többi pedig másodlagos azonosítók lesznek. A *Uniprot* azonosítókat még csoportosítani lehet az alapján is, hogy a fehérje manuálisan vagy automatikusan lett annotálva. Az első típusba az úgynevezett *Swissprot* az utóbbiba pedig a *trEMBL* azonosítók tartoznak. Minden swissprot azonosító egyben elsődleges azonosító is ez tuti? van rá forrás?. A szkriptem az összes pont azonosítójára, ha az

nem *Swissprot*, kikeresi a swissprot azonosítót ha létezik, vagy az elsődleges trembl azonosítót. Az ARN és a Salmonet fordításához szükség volt egy *Salmonella-Salmonella* és egy humán-humán szótárra, amiket a Kadlecsik Tamás féle szótárépítő scripttel állítottam elő. Az azonosítókat a szótárak alapján saját készítésű szkripttel fordítottam.

A predikciókhoz egy olyan szótárat kellet létrehozni, mely *Salmonella* génazonosítókhoz rendel *Salmonella Uniprot* azonosítókat na itt ki kellene fejteni hogy ez miért kell. Ezt szintén Kadlecsik Tamás szkriptjével állítottam elő. Egy általam írt másik fordítószkript segítségével pedig az előzőhöz hasonló módon fordítottam a predikciókat.

5.3. A források

5.3.1. Autophagy Regulatory Network (ARN)

Az ARN egy széles terjedelmű autofágia adatbázis. Az adatbázis az irodalomból kézi gyűjtéssel kapott élek mellett tartalmaz még 19 más adatbázisból importált valamint 4 féle predikcióval készült feltételezett kapcsolatokat is. Az ARN-ben található 1485 darab fehérje között 4013 kapcsolat van. Az adatbázis komponensei között vannak az autofágia mechanizmusában szerepet játszó fehérjék és ezek regulátorai valamint transzkripciós faktorai. Az adatbázisban 413 transzkripciós faktor valamint 386 olyan miRNS melyek képesek lehetnek autofágia komponensek szabályzására. [**?**]
Az ARN hat rétegből épül fel:

1. Autofágia fehérjék.
2. Az első réteg fehérjéinek autofágia specifikus forrásokból származó regulátorai.
3. Olyan poszt-transzlációs regulátorok melyek közvetlenül hatnak az első két réteg fehérjéire.
4. Az első három réteg transzkripciós szabályzói.
5. Az első négy réteg poszt-transzlációs regulátorai.
6. Olyan jelátviteli útvonalak és fehérje-fehérje interakciók melyek különböző útvonalakat az autofágia szabályzókhoz kötnek.

[**?**]

Az ARN feldolgozása

A gazda patogén hálózat összeállításához az ARN adatbázisnak csak az első, autofágia fehérjéket tartalmazó rétegét használtam fel. A hálózat letöltését követően azt egy általam írt *Python* scripttel MiTab SQL formátumba alakítottam. A fordító szkripttel az akkor legfrissebb *Uniprot* adatbázis azonosítókra fordítottam. ide kellene egy minimális megjegyzés hog yakk

5.3.2. Salmonet

A *Salmonet* a csoportunk által készített de még nem publikált molekuláris hálózat. A háló kézi gyűjtésű *Salmonella-Salmonella* fehérje-fehérje kapcsolatokat tartalmaz.

A Salmonet feldolgozása

ezek szándékosan számozatlan alfejezetek? Az *Salmonet* átalakítása az ARN-hez hasonló módon történt, azzal a különbséggel, hogy fordításkor *Salmonella-Salmonella* szótárat használtam. ide majd kellene egy in prep. hivatkozás

5.3.3. A predikciók

Az eddig ismertetett források csak fajon belüli kapcsolatokból felépülő hálózatokat tartalmaztak. Ahhoz, hogy szakdolgozatomban tudjam tanulmányozni a humán-*Salmonella*~~*kapcsolatot*~~ kapcsolatot szükségem van még interspecifikus élekre is. A predikciós forrásokból származó interspecifikus kapcsolatok fogják összekapcsolni a gazda hálózatát a patogénével. Szakdolgozatomban [**?**] és [**?**] humán-*Salmonella* predikcióit használtam.

A predikciók feldolgozása

A két predikció feldolgozására külön *Python* szkripteket írtam. Csakúgy mint az előző forrásokat, az így elkészült adatbázisokat a legújabb *Uniprot* azonosítóra fordítottam.

5.3.4. Az adatbázisok egyesítése

Az adatbázisok egyesítésekor a fő szempont az, hogy az végleges hálózatban ne legyenek redundáns pontok vagy élek. Az adatbázis egyesítő szkript beolvassa az összes adatbázisfájlt és eg *has-map*ben eltárolja a pontokat és éleket és ezek tulajdonságait. A *hasm-map*-ből gyorsan ki kereshető, hogy egy adott él vagy pont benne van-e már

ide jó lenne egy megjegyzés hogy sokkal gyorsabb mint az sqlite. Ami a szkript végigmegy az összes adatbázisfájlon a *has-map*-ek tartalmából létrehoz egy MiTab SQL fájlt mely nem redundánsan tartalmazza az összes forrás adatbázis tartalmát.

5.4. A hálózatok topológiai elemzése

5.4.1. A TopologyAnalyser osztály

A *TopologyAnalyser* osztály *Python 3* szintaxist használ. Az osztály segítségével kiszámítható a Jordán Ferenc féle topológiai fontosság (TI^n). Az TopologyAnalyser egyetlen külső függősége a NetworkX csomag. A NetworkX egy *Python*ban írt, komplex hálózatok létrehozására, manipulálására és elemzésére használható ingyenes csomag. ide gyorsan leírhatnád hogy a

Az osztály konstruktorának egyetlen paramétere egy éllista. Példányosítás után a *TopologyAnalyser* típusú objektum, egy éllistát, egy *Graph* típusú objektumot és egy egység-élerősségi mátrixot tartalmaz.

A *Graph* osztály a NetworkX csomag része és irányítatlan gráfok tárolására alkalmas. A hálózat példányhoz kötött *Graph* típusú objektum (*self.graph*) tárolása előnyös, mert így elérhetők a NetworkX csomag különböző gráf elemzésre használható metódusai. Ilyen függvény például a *Graph.neighbors_iter(node)* mely egy adott pont szomszédjainak iterálható listáját adja vissza.

Az élerősség mátrix (*self.edgeStrength*) egy példányhoz kötött *Python Dictionary*. A *Dictionary* osztály *Pythonban hashmap* adatstruktúrával van implementálva, ezáltal a benne tárolt adatok gyorsan elérhetők. A szótár kulcsai a mátrix indexei, az értékei pedig az élerősségek *Fraction* típusú objektumokként letárolva. Az egységerő mátrix celláinak indexei maguk az élt alkotó pontok *Uniprot* azonosítói. Az élerősség mátrix 0 értékkel rendelkező cellái nincsenek letárolva a *hasmap*-ben.

Az osztály fontosabb metódusai és működésük

- *getNthNeighbors()*

  A függvény rekurzívan kikeresi az adott pont n. szomszédját.

- *pathFinder()*

Ez a függvény megkeresi az összes n hosszúságú útvonalat egy forrás és egy cél nódus között.

```
def pathfinder(mélység, forrás nódus, cél nódus, útvonal, összegyűjtött útonalak):
    if útvonalhossz == 0 then
        forrás nódus hozzáadása az útvonalhoz ;
        mélység =- 1
    else if mélység == 0 és forrás nódus == cél nódus then
        return útvonal
    else if mélység == 0 és forrás nódus ≠ cél nódus then
        return
    for forrás nódus első szomszédai do
        szomszéd nódus hozzáadása az útvonalhoz;
        pathfinder( mélység - 1, szomszéd nódus, cél nódus, útvonal, összegyűjtött útvonalak) ;
        az utolsó pont eltávolítása az útvonalból;
```

- *buildEdgeUnitStrengthDict()*

  Ez a függvény egyszer hívódik meg a konstruktor lefutása során. A metódus imperatív módon egy **for** ciklussal ~~végigmegy~~ végig iterálja a kapott éllistán. A függvény visszatérési értéke egy egységerő mátrix.

- *getNthPathwaysForNodes()*

  A függvény először minden pontnak megkeresi az n. szomszédját. Mint az irodalmi bevezetőben ismertettem, a topológiai fontosság számolásánál páros lépésszámnál hurkon keresztül egy adott pont mindig önmagának a szomszédja is. A metódus két egymásba ágyazott **for** ciklust használ. A külső ciklus minden ponra előállítja a pont n. szomszédjainak halmazát. A belső ciklus pedig minden cél és forrás pont párra meghívja a *pathFinder()* függvényt, majd a kapott útvonalakat imperatívan hozzáadja egy listához, amit még a ciklusokon kívül definiáltam. A metódus visszatérési értéke egy lista amelyben minden lehetséges forrás és célpont közötti n hosszúságú útvonal benne van.

- *getEdgesFromPathway()*

  A metódus egy útvonalat fogad és az azt alkotó élek listájával tér vissza.

- *countPathwayStrength()*

A metódus egy útvonalat kap, amire meghívja a *getEdgesFromPathway* függvényt. A kapott éllistán végigmegy és minden élre kikeresi az egységerő mátrixból a megfelelő értéket. Az összes él erejének összeszorzása után az út erősségével tér vissza.

- *getPathWaysToNthNeighbours()*

  Ez a függvény kikeresi egy adott pont összes n. szomszédját és egy listában letárolja. Amikor elkészült a szomszédok listájával, akkor ezen egy **for** ciklus segítségével végigmegy. Ekkor minden periódusban meghívja a forrás és az aktuális cél nódusra a *pathfinder* függvényt, majd a kapott útvonalakat imperatíven hozzáfűzi egy a cikluson kívül definiált listához. A ciklus után a függvény visszatér az elkészül listával.

- *countTI()*

  A függvény egy adott pontra meghatározza annak topológiai fontosságát. Első lépésként létrehoz egy útvonal listát a *getPathWaysToNthNeighbours* metódus segítségével, majd végigmegy az útvonalak listáján egy **for** ciklusban. Minden periódusban meghívja a *countPathwayStrength(pathway)* függvényt az aktuális útvonalra, és annak eredményét egy a cikluson kívül definiált változóba akkumulálja.

6. Eredmények (A hálózat elemzése)

6.1. Főbb statisztikák

6.2. Kapott topológiai adatok és jelentésük

7. Diszkusszió

8. Összefoglalás

9. Summary

10. Hivatkozások jegyzéke

1. link TODO github link

11. Köszönetnyilvánítás

12. Nyilatkozat