MOLEKULÁRNÍ TAXONOMIE - 7 (2019)

Zavedení relativních rychlostí

Protože jsem si v minulých přednáškách řekli, že prozatím rezignujeme na to rozlišovat, jaký podíl na délce větve má substituční rychlost u a jaký uplynulý čas t, můžeme celkovou substituční rychlost arbitrárně nastavit rovnou 1,00. Tím pádem se nám součin ut změní na t, ale toto t od této chvíle nepředstavuje čas ale délku větve (genetickou distanci, D), tedy parametr, který obvykle chceme určit. Přešla do něj i absolutní hodnota rychlosti. Proč jsme provedli tuto operaci bude za chvíli zřejmé.

Od této chvíle budeme členy matice Q, která je rozkladem celkové rychlosti u na komponenty představující různé typy záměn, nazývat relativními rychlostmi a musíme dbát na to aby průměrná hodnota členů mimo diagonálu byla rovna 1,00 (to je ta námi nastavená celková u). Matice Q pro Jukes-Cantorův model by pak obsahovala na diagonále samé -3 a členy mimo diagonálu by měly hodnotu 1. Stále totiž platí, že součet řádku matice Q musí být 0 (viz přednáška 5). U general time reversible modelu jsem dovolili, aby různé typy záměn měly různé relativní rychlosti (α,β,γ,δ,ε,ζ). Pokud chceme ještě navíc vzít v potaz frekvence s jakými sekvence používají jednotlivé typy bazí, vynásobíme výrazy v matici frekvencí báze v řádku (π_i).

$$Q = \begin{bmatrix} -(\alpha\pi_G + \beta\pi_C + \gamma\pi_T) & \alpha\pi_A & \beta\pi_A & \gamma\pi_A \\ \alpha\pi_G & -(\alpha\pi_A + \delta\pi_C + \varepsilon\pi_T) & \delta\pi_G & \varepsilon\pi_G \\ \beta\pi_C & \delta\pi_C & -(\beta\pi_A + \delta\pi_G + \eta\pi_T) & \eta\pi_C \\ \gamma\pi_T & \varepsilon\pi_T & \eta\pi_T & -(\gamma\pi_A + \varepsilon\pi_G + \eta\pi_C) \end{bmatrix}$$

Matice rychlostí bude tedy vypadat stejně jako prve, ale hodnoty mimo diagonálu musí být po každé operaci normalizovány (vyděleny svým součtem), tak aby jejich průměr byl 1,00 a členy na diagonály musí mít zápornou hodnotu součtu zbytku řádku. Matici pravděpodobností že přes větev t dojde ke změně odvodíme opět jako $P(t)=e^{Qt}$ a tvar jejích členů bude dost komplikovaný. Připomínám, že t od teď označuje délku větve.

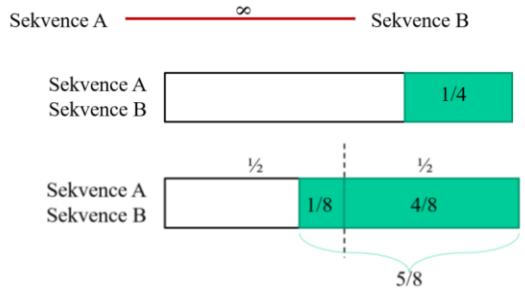
Modelování různé substituční rychlosti v různých pozicích alignmentu

Model s invarianty

Nyní se dostaneme k důvodu, proč jsme v předchozí kapitole zaváděli relativní rychlost. Pomůže nám to totiž modelovat situaci, kdy různé pozice alignmentu substituují různě rychle. To je u reálných dat docela rozumný předpoklad – aktivní místo enzymu zřejmě substituuje pomaleji než část enzymu, která neplní žádnou významnou funkci, třetí pozice kodónů zase rychleji než jiné.

Začneme s jednoduchou situací. Dříve jsme si ukázali, že pokud v alignmentu všechny pozice substituují stejně rychle, všechny typy záměn mají stejnou substituční rychlost a nukleotidy v alignmentu jsou rovnoměrně zastoupeny (tedy platí model Jukes-Cantora), tak se dvě

nepříbuzné sekvence (vzdálené ∞) budou v průměru lišit v ¾ nukleotidů (p=0,75). Nyní předpokládejme situaci, kdy polovina pozic v našem alignmentu vůbec nepodléhá substitucím. V takovém případě se dvě nepříbuzné sekvence budou lišit ve 3/8 nukleotidů. Ve zbylých 5/8 budou stejné – 4/8 představuje ta nesubstituující polovina alignmentu, 1/8 představuje ¼ ze subtituující poloviny, která vinou saturace skončí na stejném nukleoidu (viz. obrázek níže).



Nevědouc, že v našich sekvencích je taková nerovnováha v substitučních rychlostech pozic, bychom použili JC substituční model, dosadili za p pozorovaných 3/8 rozdílů a vypočítali D=0,52, což je mnohem méně než skutečná hodnota (∞). Situaci vyřešíme tím, že pozměníme JC model tak, aby uvažoval existenci dvou kategorií pozic. JC bude dále platit, ale u ½ pozic vynásobíme substituční rychlost koeficientem r_2=0, takže budou mít nulovou rychlost, u druhé poloviny pozic vynásobíme koeficientem r_1=2. Průměrná substituční rychlost zůstane 1,00. Podstatné je, že nemusíme vědět, které pozice patří do té, či oné kategorie. Stačí předpokládat, že každá pozice se nachází v obou kategoriích s pravděpodobností ½. Výraz pro pravděpodobnost, že nukleotid zůstane sám sebou p_0, bude součtem dvou výrazů odvozených z JC, z nichž každý bude přispívat ½ k výsledné pravděpodobnosti.

$$p(0) = 1/2(1/4 + 3/4e^{r_1 t}) + 1/2(1/4 + 3/4e^{r_2 t})$$

V našem případě bude druhá závorka pro jakékoliv t vždy rovna 1,00. Dosadíme-li příslušné hodnoty za r_1 a r_2 a za t=∞

$$p_0(\infty) = \left(\tfrac{1}{4} + \tfrac{3}{4}e^{-2\infty}\right)1/2 + \left(\tfrac{1}{4} + \tfrac{3}{4}e^{-0\infty}\right)1/2$$
$$= (1/4)1/2 + (1)1/2 = 5/8$$

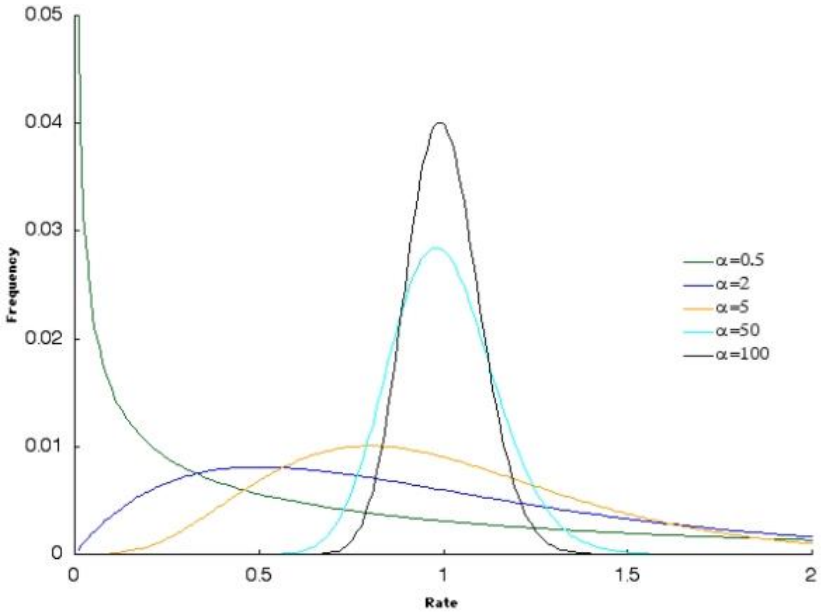
Dospějeme k hodnotě 5/8, tj. počet stejných nukleotidů u nepříbuzných sekvencí. Tento model se tedy chová podle našich předpokladů. Vytvořili jsme model předpokládající přítomnost 50% invariantních pozic. Takové modely s invarianty jsou implementovány ve většině fylogenetických softwarů. Procento invariant si lze nastavit nebo ponechat jako neznámý parametr, jehož hodnota bude optimalizována (viz. příští přednáška). Samozřejmě, že rozdělení pozic na dvě kategorie s rychlostí 0 a kladnou je příliš hrubé a je žádoucí roztřídit pozice jemněji na větší počet kategorií.

Model s více rychlostními kategoriemi

Kdybychom znali substituční rychlosti jednotlivých pozic, mohli bychom z nich opět vytvořit relativní rychlosti r_1 - r_i (vydělit je jejich součtem, aby se jejich průměr rovnal 1,00) a těmi pak násobit matici substitučních rychlostí. Pravděpodobnostní matice pro pozici i by pak měla tvar

$$P(t) = e^{r_i Q t}$$

Pozice lišící se relativní substituční rychlostí by měly různé P(t). Určit takové množství rychlostních parametrů $\underline{r}_i$, jejich počet se rovná počtu pozic alignmentu, je ovšem problematické a jejich nepřesné určení zatíží analýzu velkou chybou. V praxi se osvědčilo používat rozložení relativních rychlostí. Zjistilo se, že frekvence pozic s různými rychlostmi má rozložení podle funkce gama (Γ). Tvar křivky je histogramem relativních rychlostí pozic. Tvar této funkce je určován do značné míry parametrem α a poměrně zanedbatelně parametrem β.



Pokud je α větší než 0,5, připomíná tvar této funkce normální rozložení kolem hodnoty 1,0. Se zvyšující se hodnotou α je tento "zvon" stále užší a vyšší nad hodnotou 1,0. Takové rozložení má alignment, ve kterém jsou relativní rychlosti pozic vyrovnané a pohybují se kolem 1,0. Pokud je hodnota α menší než 0,5 připomíná funkce Γ tvarem hyperbolu a v takovém případě to znamená, že alignment obsahuje většinu pozic s malou relativní rychlostí a jen malé množství pozic má velkou relativní rychlost substituce. Připomínám, že průměr rychlostí všech pozic je 1,0. Ideálním řešením, jak zahrnout do výpočtu informaci, že rozložení rychlosti pozic odpovídá funkci Γ, je integrovat naši pravděpodobnostní matici P(t) pro všechny rychlosti spojitě podél funkce Γ.

$$P(t) = \int_0^\infty f(r)\, e^{rQt}$$

To je schůdné jen jednoduché modely. Pro Jukes-Cantorův model se to podařilo vyřešit a vznikl tak model Jin a Nei.

$$D = -3/4\ \alpha[1-(1-\ 4/3\ Ds)^{-1/\alpha}]$$

Kde α je onen parametr určující tvar křivky funkce Γ. Ds je počet rozdílných nukleotidů (p). D je genetická distance (tedy t). Všimněte si, že pro stejnou hodnotu Ds (např. Ds=0,5) v případě, že α=0,5 tak D=3, zatímco když α=10 bude D pouhých 0,87.

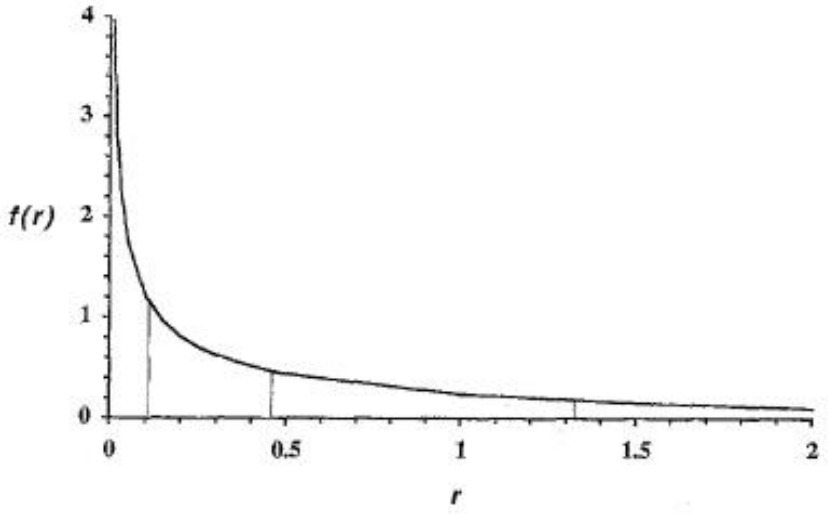
To odpovídá skutečnosti. Představte si, že na sekvenci dojde přes větev t k určitému počtu substitučních událostí. V prvním případě (α=0,5) se tento počet událostí odehraje převážně v menším množství rychle substituujících pozic, které se rychle saturují, takže počet pozorovaných rozdílů, na konci větve bude nižší než ve druhém případě (α=10), kdy se substituce rozloží na celý alignment. Podíváme-li se na to z druhé strany, znamená to, že tentýž počet pozorovaných rozdílů odpovídá u alignmentu, jehož pozice substituují rovnoměrně rychle, kratší genetické distanci než u alignmentu s nerovnoměrným rozložením substitučních rychlostí. Pokud tedy nevezmeme v potaz nerovnoměrné rozložení substitučních rychlostí (a ono bude přítomno) budeme skutečnou genetickou distanci podhodnocovat, a to tím více čím bude vyšší počet pozorovaných rozdílů mezi sekvencemi.

Jak jsem uvedl výše, pro složitější modely je těžké tuto integraci provést. Proto byl navržen diskrétní (tj. nespojitý model), který nahrubo provádí onu integraci. Pozice se rozdělí do několika rychlostních kategorií (obvykle 4 nebo 8) tak, aby každá z kategorií obsahovala stejný počet pozic. Můžeme si to představit tak, že rozdělíme plochu pod křivkou na 4 stejné díly. U každé kategorie se určí průměrná substituční rychlost. Každé pozici v alignmentu se přiřkne s pravděpodobností ¼ (protože jsme rozdělení rozložili na 4 stejné díly) relativní rychlost z každé kategorie.

$$P(t) = 1/4\, e^{r1Qt} + 1/4\, e^{r2Qt} + 1/4\, e^{r3Qt} + 1/4\, e^{r4Qt}$$

Například, při α=0,5 vypadá funkce Γ a rozdělení do 4 kategorií jako na obrázku níže. Hodnoty rychlostí pro 4 kategorie budou přibližně $r_1 = 0{,}0334$, $r_2 = 0{,}2519$, $r_3 = 0{,}8203$ a $r_4 = 2{,}8994$. Vzorec by vypadal

$$P(t) = 1/4\, e^{0{,}0334Qt} + 1/4\, e^{0{,}2519Qt} + 1/4\, e^{0{,}8203Qt} + 1/4\, e^{2{,}8994Qt}$$

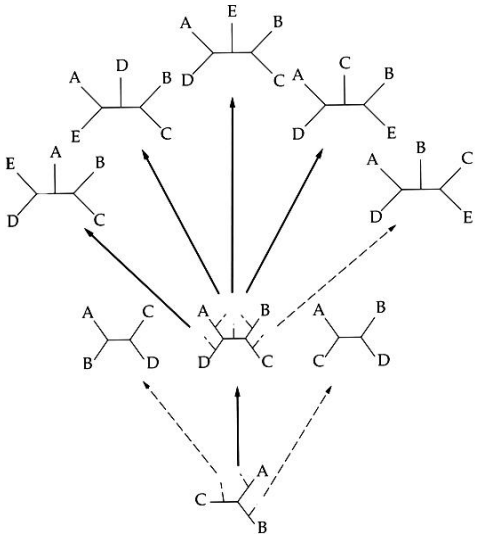


Podstatné je, že poslední vzorec poskytne při zachování stejné hodnoty t nižší celkovou pravděpodobnost záměn, než kdyby pozice substituovaly rovnoměrně (α=∞) a za tedy r1=r2=r3=r4=1.

Pokud si kladete otázku, odkud vezmeme hodnotu parametru α, která nám udává tvar funkce Γ a od které se naše výpočty odvíjí, tak vyčkejte do příští přednášky.

Kolik je možných topologií

Pro fungování metod, jako jsou nejmenší čtverce a minimální evoluce, které pracují tak, že skórují předkládané topologie a vybírají tu, která má skóre nejlepší, je podstatné se dobře vyznat v prostoru všech topologií. Je to z prostého důvodu, pokud bychom nejkvalitnější topologii během skórování zapomněli metodě předložit ke skórování, tak nemá šanci zvítězit. Samozřejmě, že ideální by bylo podrobit skórování všechny možné topologie. Tvary topologií lze systematicky uspořádat, jak naznačuje obrázek. V případě nezakořeněných stromů existuje pro tři taxony jedna možná topologie obrázek (dole). Pro čtyři taxony existují 3 topologie, protože jsou tři místa, kam můžeme čtvrtý taxon na třítaxonový strom přidat. Podobně si můžeme odvodit, že pětitaxonových stromů bude 15 atd. V případě zakořeněných stromů jsou počty topologií vždy o krok napřed, protože kořen se chová jako taxon. Takže pro 3 taxony existují 3 zakořeněné topologie.



Je patrné, že počet topologií se vzrůstajícím počtem taxonů strmě roste. Počet zakořeněných topologií se rovná

(2n-3)!!

kde n je počet taxonů a !! znamená faktoriál lichých čísel. Pro n=5 je počet zakořeněných topologií (2*5-3)**!!** = (3*5*7) = 105
Na dalším obrázku uvádím počty zakořeněných topologií pro některá n. Pro n=60 by už tento počet byl srovnatelný s počtem atomů ve vesmíru, který se odhaduje 10^{100}. Současné počítače

jsou schopny za sekundu provést asi 3 miliardy operací. I kdyby výpočet skóre pro jednu topologii představoval jednu operaci, byl by současný počítač za dobu existence vesmíru schopen projít asi 10^{27} topologií.

Species	Number of trees
1	1
2	1
3	3
4	15
5	105
6	945
7	10,395
8	135,135
9	2,027,025
10	34,459,425
11	654,729,075
12	13,749,310,575
13	316,234,143,225
14	7,905,853,580,625
15	213,458,046,676,875
16	6,190,283,353,629,375
17	191,898,783,962,510,625
18	6,332,659,870,762,850,625
19	221,643,095,476,699,771,875
20	8,200,794,532,637,891,559,375
30	4.9518×10^{38}
40	1.00985×10^{57}
50	2.75292×10^{76}

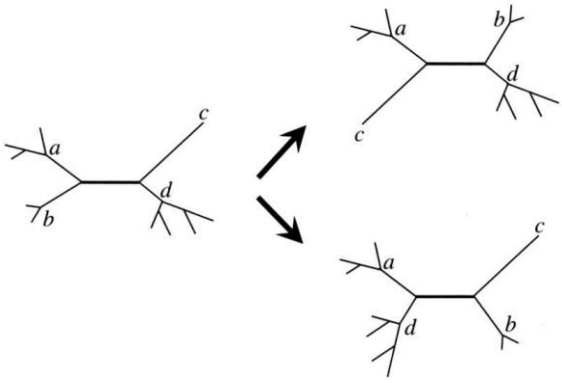
Z toho je zřejmé, že pro reálné sady dat, které mnohdy obsahují i stovky taxonů je nemožné procházet všechny topologie. Bylo tedy nutné přijít s řešením, jak chytře prohledat prostor topologií, abychom nemuseli prohledávat zdaleka všechny, a přitom pokud možno neminuli tu nejlepší. Rovnou předesílám, že ideální řešení zvládnutelné v reálném čase existuje jen pro velmi malé sady dat.

Prohledávání prostoru stromů

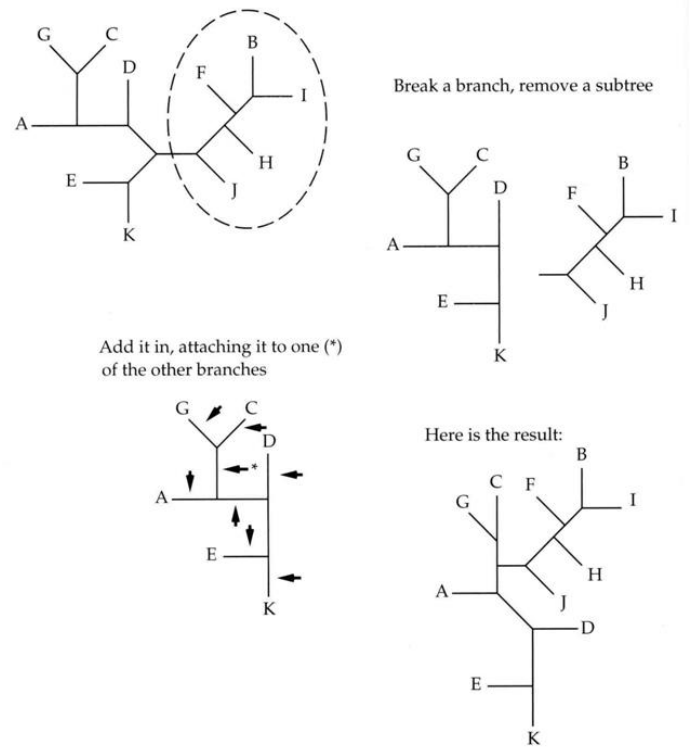
Heuristické metody

První typ algoritmů má přívlastek heuristické. Tyto algoritmy negarantují, že navštíví nejoptimálnější topologii. Dělají však vše proto, aby se pohybovaly ve vzorku těch nejkvalitnějších. Jejich práci si můžeme představit jako misi parašutisty, který se snese do “lesa” stromů. Dopadne na jeden, pro tuto chvíli řekněme náhodný, strom. U tohoto stromu změří kvalitu – vypočítá skóre (třeba metodou nejmenších čtverců). Pak se rozhlédne po sousedních stromech. U každého z nich spočítá skóre. Pokud některý z nich bude mít skóre lepší, přejde k tomuto stromu. Takto postupuje do chvíle, pokud v okolí už nenajde žádný lepší strom. V takovém případě prohlásí strom, u kterého se nachází, za ten nejlepší. Tento chamtivý algoritmus (z angl. “greedy”) má velkou nevýhodu, že vždy skončí na lokálně nejlepším stromě. V krajině hodnot pro dané skóre se vždy vyšplhá na vrchol kopce, na jehož svah dopadne. Přitom v krajině mohou existovat vyšší kopce, jenže on neumí překročit údolí. Dříve než budeme řešit tento problém, podíváme se na to, jakým způsobem lze se “rozhlížet” po okolních stromech. Představíme si tři “rozhlížecí” algoritmy. V angličtině jim říkáme algoritmy pro “tree rearangements”.

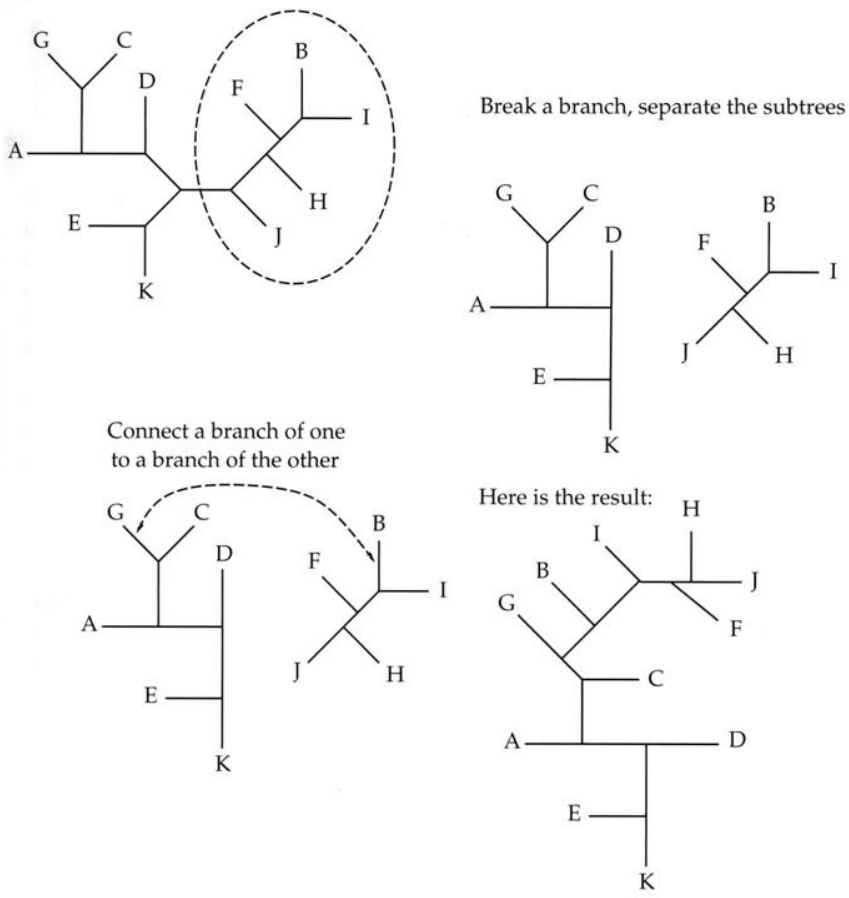
Nearest neighbour interchange (NNI), považuje za sousední topologie takové, u kterých došlo k přehození větve mezi dvěma sousedními uzly. Vnitřní větev spojující dva sousední uzly je na obrázku vyznačena tučně. Již víme, že pro čtyři taxony existují tři možné topologie a tohle je něco podobného. Na jedné topologii právě stojíme, takže máme možnost přejít na dva sousedy lišící se uspořádáním kolem zvolené vnitřní větve. Tento strom obsahuje šest vnitřních větví, a proto bude mít celkem 6*2 sousedů podle NNI.



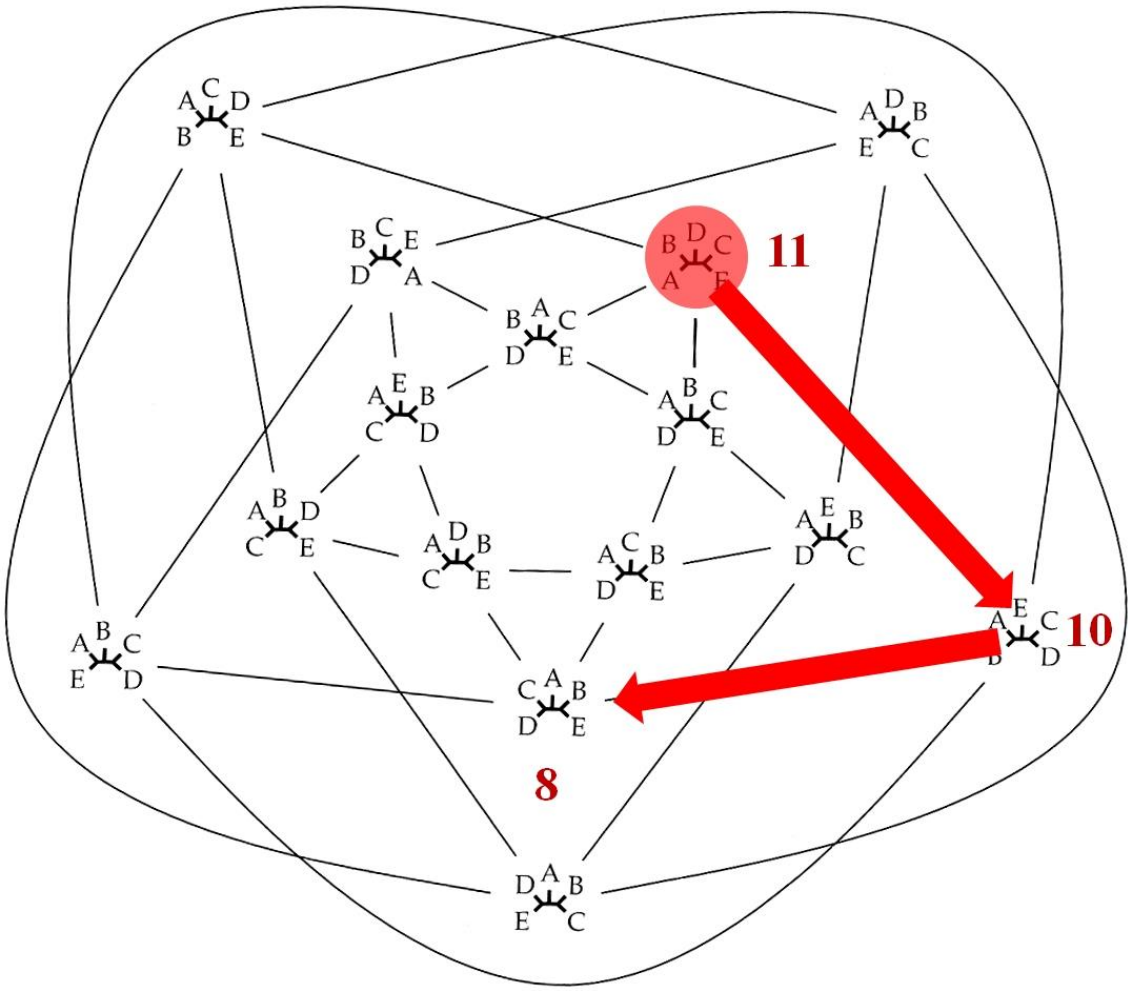
Subtree pruning and regrafting (SPR) postupuje tak, že z topologie vytrhne část (přerušovaný ovál) a tuto část pak naroubuje na všechna myslitelná místa stromu, který zbyl (šipky). Toto provede pro všechny možné části stromu (jedno taxonové i vícetaxonové). Z uvedeného je na první pohled zřejmé, že SPR vidí kolem topologie mnohem více sousedů než NNI.



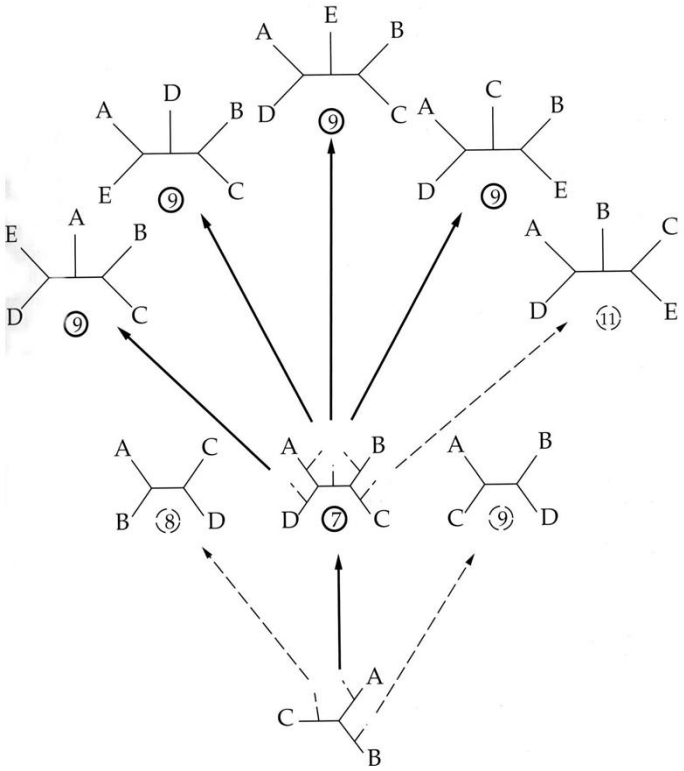
Tree bisection and reconnection (TBR) postupuje tak, že topologii roztrhne na dvě části, a potom vytváří všechny myslitelné vnitřní větve mezi těmito částmi. To provede pro všechny možné způsoby roztrhnutí původní topologie. TBR vidí ještě více sousedních stromů než SPR.



Na obrázku níže je vyobrazený kompletní prostor pro nezakořeněné topologie o 5 taxonech (celkem 15 topologií). Topologie, které spolu sousedí podle NNI jsou spojeny čarami. náš parašutista se snesl na topologii označenou kroužkem a její skóre vyčíslil na 11. V tomto případě se snaží hodnotu skóre minimalizovat (tak jako u nejmenších čtverců). Rozhlíží se po sousedních topologiích a jedna z nich má skóre nižší (10). Přejde na tuto topologii a rozhlédne se dále. Topologie s nejnižším skóre v okolí má hodnotu skóre 8. Přejde na ni, a protože v jejím okolí není topologie s nižším skóre, zůstane na této a prohlásí ji za tu nejkvalitnější.



V úvodu heuristických metod jsem uvedl, že lakomý algoritmus ženoucí se za neustálým zlepšováním skóre nejspíš uvízne na lokálním maximu či minimu (podle toho, co hledá). Co se dá udělat, aby tomu tak nebylo. Jedna z věcí, kterou lze vylepšit, je místo přistání našeho parašutisty. Vyplatí se, aby přistál v místech, kde se vyskytují rozumně dobré topologie. Metoda, která se snaží toto řešit vystaví počáteční topologii posupným přidáváním taxonů na ta místa, která nejméně zhoršují hodnotu skóre, tak jak je uvedeno na obrázku níže. Větev s taxonem D přidáme na všechna tři možná místa stromu ABC a pak vypočítáme skóre. Prostřední strom má nejlepší skóre 7, a proto pokračujeme v přidávání jen na něj. Pokud má víc stromů stejná skóre pokračujeme všemi cestami.

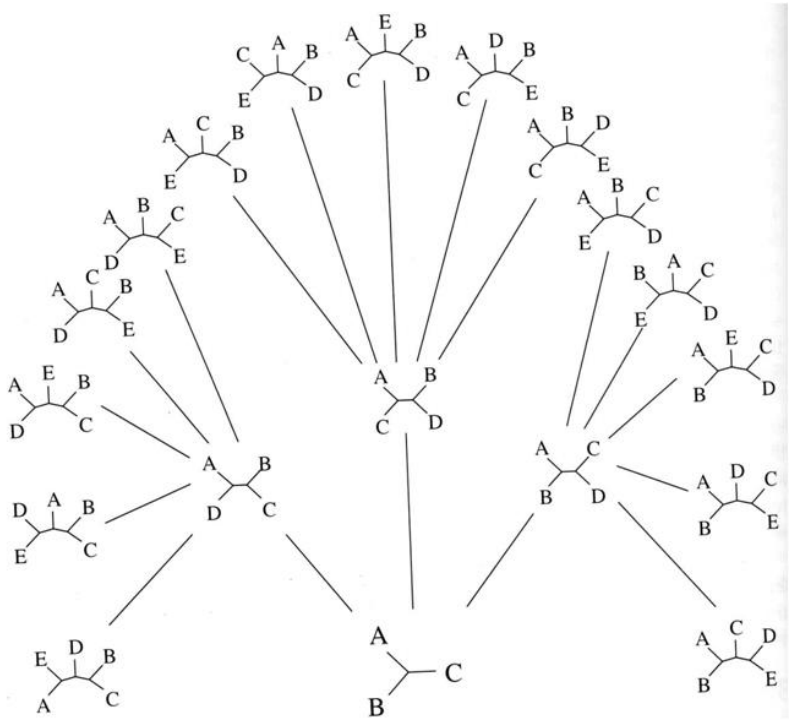


Další možností je vypočítat si počáteční strom nějakou rychlou algoritmickou metodou, např. neighbor-joining.

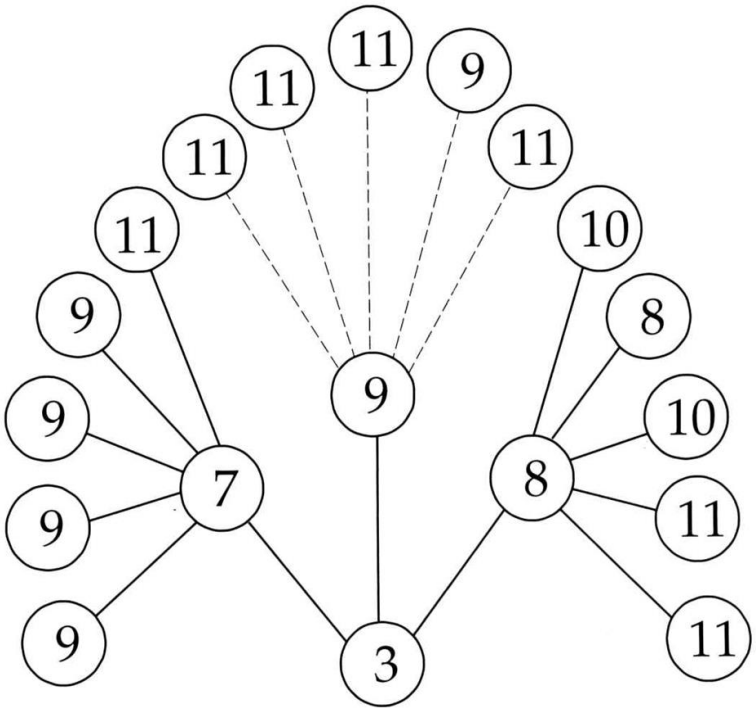
Kromě hledání vhodného místa přistání je s výhodou vyslat do krajiny celou četu výsadkářů, porovnat topologie, které najdou a vybrat z nich tu nejlepší. Čím více pošleme výsadkářů a čím pečlivěji prohledáváme okolí stromů (TBR spíše než NNI) tím zvyšujeme pravděpodobnost, že nalezneme nejlepší topologii, ale zároveň zvyšujeme výpočetní čas, nikdy však nebudeme mít jistotu, že nám nejlepší strom neuniknul. Závěrem k této části bych chtěl podotknout, že není vůbec zakázáno předhazovat algoritmu vlastní topologie, o kterých se domníváme, že jsou kvalitní, a nemusíme to nijak odůvodňovat. Čím více toho prohlédneme tím lépe. Takový postup není porušení žádných pravidel.

Branch-and-bound

Tato metoda garantuje, že nepřehlédne nejlepší topologii. Nicméně je díky tomu nucena prohledávat prostor topologií příliš důkladně, takže je obvykle nepoužitelná v reálném čase. Branch and bound si uspořádá topologie hierarchicky. Prostor všech možných 15 topologií pro 5 taxonů by vypadal jako na obrázku níže.



Tak jako metoda postupného přidávání i branch-and-bound postupuje odspodu stromu stromů a počítá skóre. Představme si, že výše uvedené stromy mají následující skóre.



Branch and bound algoritmus bude postupovat po větvích tohoto stromu stromů a bude si přitom pamatovat, jaké bylo nejlepší skóre, se kterým se setkal na vrcholových větvích. Toto

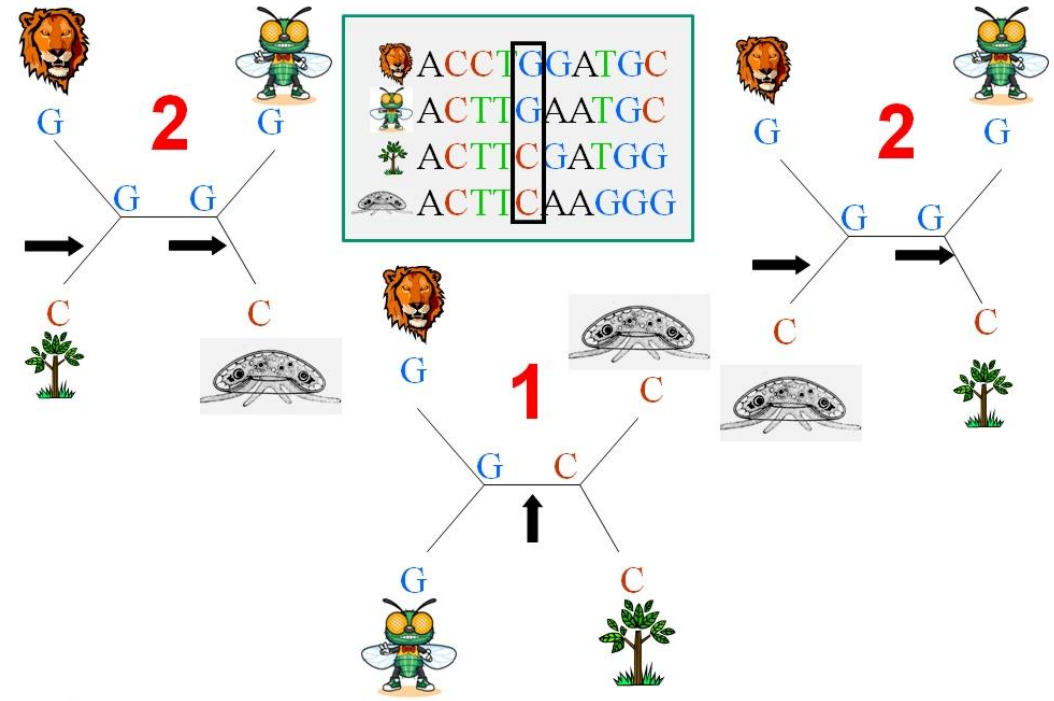
nejlepší skóre je tzv. “bound” laťka, nad kterou už ví, že se nemusí dívat. Pokud předek některé větve topologií tuto laťku překročí, je jasné, že jeho potomci budou jenom horší a tedy, že touto částí stromu stromů je netřeba se zabývat. Probereme si to na příkladu výše uvedeného obrázku.

Tři možné čtyřtaxonové topologie mají skóre 7, 8 a 9. Metoda se vydá nejprve do větve, která vychází z topologie se skóre 7, protože tam lze očekávat dobré stromy. Je chytré navštívit co nejdříve kvalitní stromy, protože si tím rychle snížíme laťku. Potomci v této větvi mají skóre 9 a 11. Zapamatujeme si ty se skóre 9 a naší laťkou je tedy skóre 9. Pak se vydáme do větve se skóre 8. Její potomci mají skóre 8, 10 a 11. 8 je zatím nejlepší skóre. Předchozí stromy zapomeneme a pamatujeme si tento. Naše laťka je nyní 8. Předek třetí větve stromu stromů má skóre 9 a překročil naši laťku. Tuto větev jeho potomků není třeba vyšetřovat, protože jeho nemohou být lepší. Jsme hotovi.

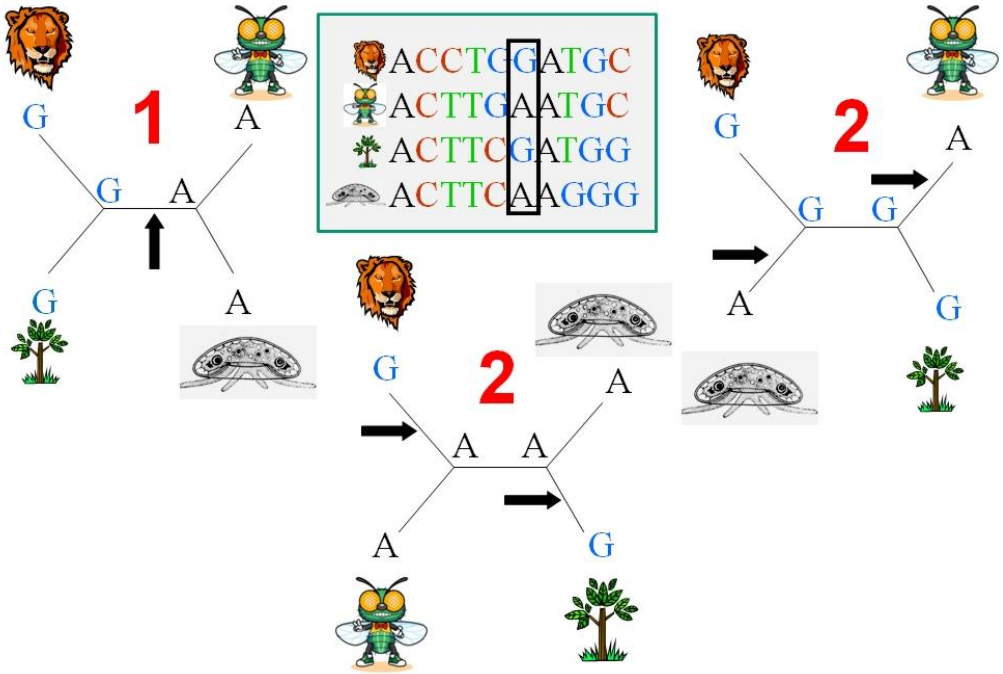
Laťku lze snižovat i rafinovaněji. U některých metod, jako je maximální parsimonie, která bude následovat, lze předem odhadnout, že přidání X sekvencí určitého typu, které nás ještě čeká, nutně povede k zvýšení skóre nejméně o hodnotu Y. Pokud předek větve stromu stromů již nyní má skóre o méně než Y menší než je laťka, nemá cenu dále pokračovat.

Maximální parsimonie

Na poslední přednášce jsme si ukázali, jak metoda nejmenších čtverců nebo minimální evoluce hodnotí předkládané topologie. Obě metody posuzovaly, jak dobře topologie “vysvětlí” pozorovaná data (distance). V jejich případě to znamenalo vyčíslit, jak dobře lze do těchto topologií “napasovat” pozorované distance. Nyní si představíme metodu maximální parsimonie, která kvantifikuje míru, jak dobře topologie “vysvětlí” pozorovaná, data naprosto jiným způsobem. Předně, nepřevádí hrubá data (alignment sekvencí, fingerprintingový vzor) na genetické distance mezi dvojicemi, ale pracuje přímo s těmito daty. Proto patří mezi tzv. **znakové metody** (na rozdíl od metod distančních). Kritériem, podle kterého maximální parsimonie hodnotí topologie, je **minimální počet změn, pomocí kterého je topologie schopná "vysvětlit" vzor forem znaků**, který se vyskytuje v alignmentu nebo fingerprintu. Při výpočtu tohoto kritéria maximální parsimonie postupuje po jednotlivých znacích, tedy pozicích alignmetu. Předvedeme si to na jednoduchém příkladu krátkého alignmentu pro 4 taxony. Budeme si všímat pozice označené obdélníkem. Tento znak nabývá formy G u lva a mouchy, formy C u stromu a améby. Jak víme, pro 4 taxony existují celkem 3 možné nezakořeněné topologie, které jsou na obrázku. Nukleotidy G a C si napíšeme na konec příslušných větví a u každé topologie se snažíme vymyslet scénář, který potřebuje minimální počet substitucí k tomu, aby se na koncích větví objevily námi pozorované nukleotidy. U stromu vpravo a vlevo neexistuje řešení s nižším počtem substitucí, než jsou 2 označené šipkami. Nalezli bychom stejně dobré (např. na vnitřních nodech nukleotidy C), ale to je v tomto případě jedno. Dolní strom má řešení s jednou substitucí na vnitřní větvi. Skóre horních topologií je pro tuto pozici 2, skóre spodního stromu 1.

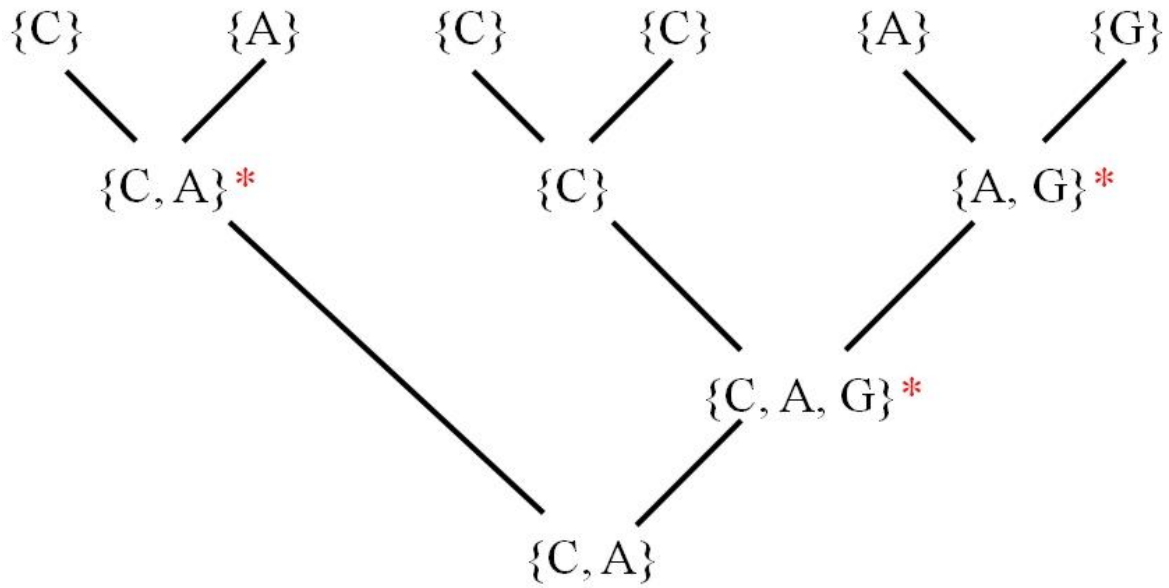


Přesuneme-li se na druhou pozici bude situace vypadat následovně



Postupně projdeme všechny pozice a pro každou topologii budeme sčítat skóre pozic pro topologie. Topologie, která bude mít celkový součet nejnižší, bude podle měřítka parsimonie tou nejlepší. Když budeme postupovat chytře, tak si uvědomíme, že některé vzory znaků nemá cenu ani zkoumat, protože v jejich případě si budou topologie rovnocenné. Jsou to samozřejmě pozice, kde mají sekvence identické nukleotidy (XXXX; za X si můžeme domyslet jakýkoli

nukleotid), ale také pozice (XXXY), u kterých budou potřebovat všechny tři topologie jednu substituci, nebo pozice (XYZW), kde budou všechny potřebovat tři substituce. Pozice s těmito vzory si nemusíme vůbec všímat, protože nejsou parsimonně informativní. U takto malých topologií je možné parsimonní řešení snadno “vykoukat”. U větších stromů už to tak snadné není, a proto, a také z důvodu, aby to mohly řešit stroje, je třeba postup algoritmizovat. Dobře pochopitelným algoritmem je tzv. Fitchův algoritmus.



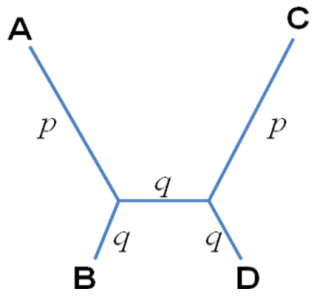
Řešíme opět každou pozici zvlášť. Opět si na konce větví napíšeme nukleotidy, které pozorujeme a postupujeme odshora dolů. U společného předka dvojice taxonů hledáme možný průnik mezi množinou nukleotidů u jeho potomků. Pokud průnik nalezneme (prostřední uzel) napíšeme průnik na tento uzel. Pokud průnik mezi potomky neexistujeme napíšeme na uzel sjednocení množin a hvězdičku, která znamená, že na jedné větvi vedoucí k potomkům (nevíme, na které, ale to je jedno) muselo dojít k substituci. Počet hvězdiček udává minimální počet substitucí. Opět můžeme postupovat chytře a uvědomit si, že když už jsme to jednou vyřešili pro nukleotidový vzor CACCAG (čteme koncové nukleotidy zleva doprava), tak stejné řešení pro tuto topologii (= 3 substituce) bude mít jakýkoli vzor XYXXYZ a nemusíme to počítat. I tady platí, že vzory XXXXXY se nemusíme vůbec zabývat, protože jejich řešení bude 1 pro jakoukoli topologii a tak dále.
Základní varianta maximální parsimonie, kterou jsme si představili je tzv. Wagnerovská parsimonie, ale existují i jiné verze. **Camin-Sokal parsimonie** předpokládá, že známe původní stav znaku a změny se dějí pouze jedním směrem, nedochází k reverzím. Nepoužívá se pro sekvence DNA a proteinů ale vhodná je pro SINE elementy. **Dollo parsimonie** zase rezolutně odmítá, že by dvě stejné formy znaků mohly vzniknout nezávisle na sobě. Daný (komplexní) znak musí podle ní vzniknou jen jednou, ale ztrácet se může opakovaně a jakkoli často. Opět není vhodná pro sekvenční data, kde k opakovaným vznikům téhož nukleotidu jasně dochází. **Vážená parsimonie** penalizuje různé typy záměn různým počtem bodů – ty pravděpodobnější

méně, ty nepravděpodobnější více. To, jak jste si všimli, Wágnerovská parsimonie vůbec neřeší.

Inkonzistence parsimonie

Skutečnost, že základní varianta maximální parsimonie neuvažuje různé substituční rychlosti pro různé typy substitucí, a ještě hůře, že žádná varianta parsimonie nepromítá do pravděpodobností záměn délky větví (delší větev přitom znamená větší pravděpodobost substituce) a považuje všechny záměny za stejně pravděpodobné, vede k její inkonzistenci. Inkonzistence je závažnou nevýhodou statistické metody, která vede k tomu, že za určitých podmínek metoda dospěje k nesprávnému řešení a je si jím jista. Není to způsobeno omezeným počtem dat a z toho plynoucí chybou výběru. Inkonzistentní metoda nás totiž za oněch specifických podmínek dovede k nesprávnému řešení i když bude mít k dispozici nekonečné množství dat. V takovém případě si dokonce bude svým řešením 100 % jistá.

V případě maximální parsimonie nastane taková situace, pokud se na stromu vyskytnou dvě nepříbuzné větve, jejichž délka výrazně převyšuje délky vnitřních větví, které je oddělují. Můžeme si to ukázat na příkladu topologie na obrázku níže. Předpokládejme opět, že známe skutečnou topologii stromu ABCD. Na této topologii se pro jednoduchost vyskytují jen dva typy délek větví *p* a *q*. Vinou zvýšené substituční rychlosti, došlo na větvích vedoucích k taxonům A a C za stejný čas k vyššímu počtu událostí, což se odráží v jejich delších větvích. Matematicky lze poměrně jednoduše ukázat, že pokud délka větví p vůči q přesáhne určitou mez, konkrétně pokud bude platit že



$p^2 > q(1-q)$

pak pravděpodobnost substituce na vnitřní větvi *q* bude nižší než pravděpodobnost konvergentních substitucí na dlouhých větvích *p*. Horní strom na obrázku dole tak bude parsimonější – bude vysvětlovat alignment sekvencí pomocí menšího počtu substitucí – i když se neshoduje se skutečnou topologií.

Řešením problému s inkonzistencí je uvažovat pravěpodobnosti záměn, které jsou ovlivněny délkami větví (tak jako u distančních metod). Takto postupuje metoda maximum likelihood, kterou se budeme zabývat příště.