5. A TKF MODELLEK KOMBINATORIKUS TOVÁBBFEJLESZTÉSEI

5.1 Összegzés az ősi szekvenciákon

A TKF modellek kombinatorikus továbbfejlesztésein olyan modelleket értek, amelyekben a szekvenciák evolúciója továbbra is a TKF modellek alapján történik, de a kapcsoltsági valószínűséget az ősi szekvenciákon való összegzéssel határozom meg. Az ősi szekvenciák tulajdonságára (pl.: a hosszak eloszlására) különböző feltételezéseket lehet tenni, ezáltal a modellek jobban megközelítik a valóságot. Az ősi szekvenciákra tett feltételezések megsértik a TKF modellek reverzibilitását. Az így kapott modellek ezáltal irreverzibilissé válnak (Miklós, 2001a).

Mint azt már a 4.7 fejezetben említettem, az ősi szekvenciákon való összegzés lehetővé teszi a szekvenciák többszörös statisztikus illesztését is (Steel & Hein, 2001; Hein, 2001). Az 5.3 fejezetben bemutatok egy olyan algoritmust, amely az eddig ismertnél nagyságrendekkel gyorsabban végzi el kettőnél több olyan szekvencia statisztikus illesztését, amelyek egy csillag alakú fa mentén evolválódtak.

5.2 Szekvenciák evolválódása Poisson szekvenciahossz eloszlásból

A TKF modelleknek egy eddig vélt (Hein et al., 2000) gyenge pontjuk az, hogy ezen modellekben a szekvenciák hosszának az egyensúlyi eloszlása a geometriai eloszlás. Ez azt jelentené, hogy a legrövidebb szekvenciák lennének a leggyakoribbak, ami ellentmond a biológiai megfigyeléseknek (Zhang, 2000) Ebben az alfejezetben egy olyan modellt mutatok be, amelyben a szekvenciák Poisson szekvenciahossz eloszlásból evolválódtak irreverzibilis úton (Miklós, 2001c).

A modell tehát felteszi, hogy a szekvenciák hosszának eloszlása Poisson volt t idővel ezelőtt, és a szekvenciák azóta a TKF91 modell alapján evolválódtak. A nyilvánvaló különbség ezen modell és a TKF91 modell között a reverzibilitásban van: habár a szubsztitúció modellezése továbbra is reverzibilis marad, a beszúrás és törlés modellezése irreverzibilissé válik. Mivel a szekvenciák ebben a modellben nem-reverzibilis úton

evolválódnak, a (4.4.2) képlet nem áll fenn, így a kapcsoltsági valószínűséget szükségképpen a definíció alapján kell kiszámolni:

$$P_t(A,B) = \sum_C P(C)P_t(A|C)P_t(B|C) \qquad (5.2.1)$$

ahol $P(C)$ a C szekvencia valószínűsége t idővel ezelőtt. Ha $C= c_1c_2\ldots c_n$, akkor

$$P(C) = e^{-\kappa}\frac{k^n}{n!}\prod_{i=1}^{n}\pi(c_i) \qquad (5.2.2)$$

ahol κ-a Poisson eloszlás paramétere.

Egy ősi szekvencia sorsát illesztés segítségével lehet bemutatni. Például a következő illesztés azt mutatja, hogy a halhatatlan linknek nincs halandó link leszármazottja az A szekvenciában és egy halandó leszármazottja van a B szekvenciában. Az ősi szekvencia első halandó linkje túlélt mindkét szekvenciában, és van egy valódi leszármazottja a B szekvenciában. A második ősi halandó link kihalt, a harmadik pedig meghalt, de hagyott egy-egy valódi leszármazottat mindkét szekvenciában

```
az ősi C szekvencia      o   *   *  *
az A szekvencia            -  A  -  -  -  A
a B szekvencia             U  C  G  -  -  G
```

Ennek a speciális átmenetnek a valószínűsége

$$P(C)P_t(A\,|\,C)P_t(B\,|\,C) = [p''_1(t)p''_2(t)\pi(\mathrm{U})][p_1(t)p_2(t)f_{AC}(2t)\pi(\mathrm{G})][p'_0(t)p'_0(t)]\mathrm{x}$$

$$[p'_1(t)p'_1(t)\pi(\mathrm{A})\pi(\mathrm{G})][e^{-\kappa}\frac{\kappa^3}{3!}] \qquad (5.2.3)$$

Vegyük észre, hogy az illesztés és így a kiszámolt valószínűség magában foglalja az összes 3 hosszúságú ősi szekvenciát, és a szubsztitúciós folyamat reverzibilitását felhasználtam.

Nagyon fontos megjegyezni a különbséget az itt bemutatott illesztés és a hagyományos illesztés között. Az itt ismertetett illesztés az ősi szekvenciához illeszt két modern szekvenciát. Ebből adódóan nem homológ karakterek egymás alá kerülhetnek az illesztésben, míg a hagyományos illesztésben ez nem fordulhat elő. Azonban ez nem okoz hibát a kapcsoltsági valószínűség kiszámításában, mert a homológ és nem homológ párok megkülönböztethetőek az illesztésben. Nevezetesen, csak azokat az illesztett párokat kell homológnak tekinteni, amelyek asszociálva vannak egy ősi halandó linkkel (mint a példában az első halandó linkhez tartozó pár), az összes többi illesztett párt nem homológként kell kezelni. Valóban, a (5.2.3) képlet ez alapján számolta ki ennek az illesztésnek a

valószínűségét. Erre az illesztés típusra azért van szükség, hogy dinamikus programozási algoritmussal meg lehessen kapni két szekvencia likelihood-ját.

Egy speciális tranzíció valószínűségét k+2 tényezőre lehet bontani, ahol k a halandó linkek száma az ősi szekvenciában. Az első tényező az immortális link sorsát írja le, az utolsó tényező megadja az ősi szekvencia hosszának a valószínűségét. A többi tényezők a halandó linkek sorsát írják le. Nevezzük el maradékszorzatnak az immortális link sorsát leíró tényező és az összes olyan tényező szorzatát, amelyek olyan halandó linkek sorsát írják le, amelyeknek legalább egy utódjuk van valamelyik szekvenciában! Tehát a maradékszorzatot úgy lehet megkapni, hogy el kell hagyni az utolsó tényezőt (az ősi szekvenciahossz valószínűségét), valamint a $[p'_0(t)p'_0(t)]$ tényezőket. Jelölje ${}^{m}RP_t(A,B)$ az összes olyan maradékszorzat összegét, amely m+1 tényezőből áll, azaz m halandó link sorsát írja le. A dinamikus programozási algoritmus ötlete az, hogy legfeljebb $|A|+|B|$ halandó linknek lehet leszármazottja A vagy B szekvenciákban, így $0 \le m \le |A|+|B|$. Mivel $\binom{m+l}{l}$ darab $m+l$ halandó linket tartalmazó illesztésből származik ugyanaz a maradékszorzat (ennyiféleképpen helyezhetjük el az illesztésben az l darab kihalt linket), A és B szekvenciák kapcsoltsági valószínűsége

$$P_t(A,B) = \sum_{l\ge 0} \sum_{m=0}^{|A|+|B|} {}^{m}RP_t(A,B) \binom{m+l}{l} p_0'(t)^{2l} e^{-\kappa} \frac{\kappa^{m+l}}{(m+l)!} \tag{5.2.4}$$

azaz a korábban kitörölt tényezők visszakerültek, és az azonos értéket adó valószínűségek a megfelelő multiplicitással szerepelnek a teljes összegzésben. Ha az m-től nem függő tagokat kiemeljük és az l-től függő végtelen sor összegét meghatározzuk

$$P_t(A,B) = \sum_{m=0}^{|A|+|B|} {}^{m}RP_t(A,B) e^{-\kappa(1-\mu^2\beta^2(t))} \frac{\kappa^m}{m!} \tag{5.2.5}$$

A cél tehát olyan dinamikus programozási algoritmus keresése, amely kiszámolja minden A_i és B_j és m-re a maradékszorzatokat. Ha m>0, akkor a maradékszorzatokat hat komponensre kell szétszedni, összhangban az alábbi lehetséges illesztés típusokkal.

- Az illesztés utolsó illesztett karaktere az A szekvenciából származik, és ennek a karakternek a linkje az ősi szekvenciában is élt már. Az összes olyan maradékszorzat összegzését, amely ilyen típusú illesztésből származik, és m halandó link sorsát írja le, ${}^{m}_{ha}RP_t(A_i,B_j)$ jelöli

- Az illesztés utolsó illesztett karaktere a *B* szekvenciából származik, és ennek a karakternek a linkje az ősi szekvenciában is élt már. Az összes olyan maradékszorzat összegzését, amely ilyen típusú illesztésből származik, és *m* halandó link sorsát írja le, ${}^{m}_{hb}RP_t(A_i,B_j)$ jelöli
- Az illesztés utolsó illesztett párja két karakter az *A* és *B* szekvenciákból, és ezeknek a karaktereknek a linkje az ősi szekvenciában is élt már. Az összes olyan maradékszorzat összegzését, amely ilyen típusú illesztésből származik, és *m* halandó link sorsát írja le, ${}^{m}_{hab}RP_t(A_i,B_j)$ jelöli
- Az illesztés utolsó illesztett karaktere az *A* szekvenciából származik, és ennek a karakternek a linkje valódi leszármazott, azaz az ősi szekvenciában még nem élt. Az összes olyan maradékszorzat összegzését, amely ilyen típusú illesztésből származik, és *m* halandó link sorsát írja le, ${}^{m}_{ra}RP_t(A_i,B_j)$ jelöli
- Az illesztés utolsó illesztett karaktere a *B* szekvenciából származik, és ennek a karakternek a linkje valódi leszármazott, azaz az ősi szekvenciában még nem élt. Az összes olyan maradékszorzat összegzését, amely ilyen típusú illesztésből származik, és *m* halandó link sorsát írja le, ${}^{m}_{rb}RP_t(A_i,B_j)$ jelöli
- Az illesztés utolsó illesztett párja két karakter az *A* és *B* szekvenciákból, és ezeknek a karaktereknek a linkjei valódi leszármazottak, azaz az ősi szekvenciában még nem éltek. Az összes olyan maradékszorzat összegzését, amely ilyen típusú illesztésből származik, és *m* halandó link sorsát írja le, ${}^{m}_{rab}RP_t(A_i,B_j)$ jelöli

Ezen maradékszorzatok segítségével a dinamikus programozási algoritmus már könnyen megadható.

Először a ${}^{0}RP_t(A_i,B_j)$-t határozom meg. Ez csak egyetlen maradékszorzatot tartalmaz, mivel az üres szekvenciához egyféleképpen lehet szekvenciákat illeszteni: minden link a halhatatlan link leszármazottja. Így

$$ {}^{0}RP_t(A_i,B_j) = p''_{i+1}(t)p''_{j+1}(t)\prod_{k=1}^{i}\pi(a_k)\prod_{k=1}^{j}\pi(b_k) \tag{5.2.6}$$

Világos, hogy amikor $m>i+j$, akkor ${}^{m}_{x}RP_t(A_i,B_j)=0$, minden $x\in\{ha, hb, hab, ra, rb, rab\}$/re. A rekurzió $m>0$-ra

$$ {}^{m}_{ha}RP_t(A_i,B_j) = {}^{m-1}RP_t(A_{i-1},B_j)p'_0(t)p_1(t)\pi(a_i) \tag{5.2.7}$$

$$ {}^{m}_{hb}RP_t(A_i,B_j) = {}^{m-1}RP_t(A_i,B_{j-1})p_0'(t)p_1(t)\pi(b_j) \tag{5.2.8} $$

$$ {}^{m}_{hab}RP_t(A_i,B_j) = {}^{m-1}RP_t(A_{i-1},B_{j-1})p_1(t)p_1(t)\pi(a_i)\, f_{a_ib_j}(2t) \tag{5.2.9} $$

$$ \begin{aligned} {}^{m}_{ra}RP_t(A_i,B_j) &= {}^{m-1}RP_t(A_{i-1},B_j)p_0'(t)p_1'(t)\pi(a_i) + \\ \{ {}^{m}_{ha}RP_t(A_{i-1},B_j) &+ {}^{m}_{ra}RP_t(A_{i-1},B_j) + {}^{m}_{hab}RP_t(A_{i-1},B_j) + {}^{m}_{rab}RP_t(A_{i-1},B_j)\}\lambda\beta(t)\pi(a_i) + \\ {}^{m}_{hb}RP_t(A_{i-1},B_j)&\frac{p_1'(t)}{p_0'(t)}\pi(a_i) \end{aligned} \tag{5.2.10} $$

$$ \begin{aligned} {}^{m}_{rb}RP_t(A_i,B_j) &= {}^{m-1}RP_t(A_i,B_{j-1})p_0'(t)p_1'(t)\pi(b_j) + \\ \{ {}^{m}_{hb}RP_t(A_i,B_{j-1}) &+ {}^{m}_{rb}RP_t(A_i,B_{j-1}) + {}^{m}_{hab}RP_t(A_i,B_{j-1}) + {}^{m}_{rab}RP_t(A_i,B_{j-1})\}\lambda\beta(t)\pi(b_j) + \\ {}^{m}_{ha}RP_t(A_i,B_{j-1})&\frac{p_1'(t)}{p_0'(t)}\pi(b_j) \end{aligned} \tag{5.2.11} $$

$$ \begin{aligned} {}^{m}_{rab}RP_t(A_i,B_j) &= {}^{m-1}RP_t(A_{i-1},B_{j-1})p_1'(t)p_1'(t)\pi(a_i)\pi(b_j) + \{ {}^{m}_{hab}RP_t(A_{i-1},B_{j-1}) + \\ {}^{m}_{rab}RP_t(A_{i-1},B_{j-1})\}&\lambda^2\beta^2(t)\pi(a_i)\pi(b_j) + \{ {}^{m}_{ha}RP_t(A_{i-1},B_{j-1}) + \\ {}^{m}_{hb}RP_t(A_{i-1},B_{j-1})\}&\frac{p_1'(t)}{p_0'(t)}\lambda\beta(t)\pi(a_i)\pi(b_j) \end{aligned} \tag{5.2.12} $$

Az algoritmus helyességét könnyen lehet ellenőrizni, végiggondolva a lehetséges illesztéseket, amik az utolsó illesztett karakter vagy karakterek elhagyásával keletkeznek.

- (5.2.7-9) Ha az utolsó illesztett karakter vagy karakterek linkje már élt az ősi szekvenciában is, akkor ezt vagy ezeket elhagyva olyan illesztést kapunk, amelyben eggyel kevesebb ősi halandó link található. A maradékszorzat értéke ezen illesztés maradékszorzatának az értéke szorozva az utolsó halandó link sorsát leíró tényezővel
- (5.2.10-11) Ha az utolsó illesztett karakter linkje még nem élt az ősi szekvenciában, akkor három eset lehetséges. Az első esetben az ősi link szülőnek ez az egyetlen leszármazottja. Ekkor elhagyva az utolsó karaktert, olyan illesztéshez jutunk, amelyben eggyel kevesebb halandó linknek van legalább egy utódja. A második esetben az ősi link szülőnek van legalább két utódja abban a

szekvenciában, amelyből az utolsó karakter származik. Ekkor az utolsó karaktert elhagyva olyan illesztést kapunk, amelyben ugyanannyi halandó linknek van leszármazottja, csak a maradékszorzatban az utolsó tényező egy $\lambda\beta(t)\pi(a_i)$ vagy egy $\lambda\beta(t)\pi(b_j)$ faktorral kevesebb, függve attól, hogy melyik szekvenciából származik az utolsó karakter. A harmadik esetben az ősi link szülőnek csak egyetlen egy leszármazottja van abban a szekvenciában, ahonnan a karakter származik, de a másik szekvenciában túlélt, ott viszont nincs valódi leszármazottja.

- (5.2.12) Ha az utolsó illesztett pár két olyan karakter, amelyek linkjei még nem éltek az ősi szekvenciában, akkor az előbbihez hasonló három eset lehetséges.

Így $O(|A|\,|B|\,(|A| + |B|))$ idő alatt kiszámíthatóak a maradékszorzatok. A (5.2.5) képlet segítségével a maradékszorzatokból $O(|A| + |B|)$ idő alatt kiszámítható a két szekvencia kapcsoltsági valószínűsége.

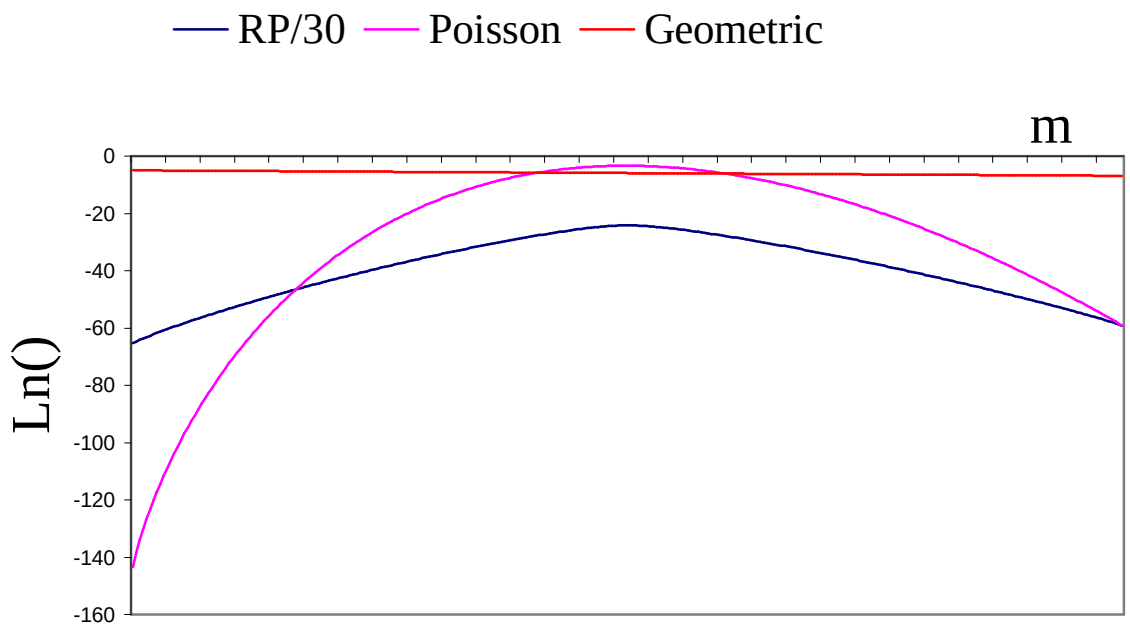
Ezen modell segítségével vizsgálható az a kérdés, hogy milyen hibát okoz a TKF91 modellben az a feltételezés, hogy a szekvenciák hosszának az eloszlása geometriai. Természetesen feltehetjük, hogy a szekvenciák hosszának az eloszlása geometriai volt *t* idővel ezelőtt, ekkor a (5.2.5) képlet a következőképpen alakul (Miklós, 2001a):

$$P(A,B) = \sum_{m=0}^{l(A)+l(B)} {}^{m}RP(A,B)\frac{\left(1-\frac{\lambda}{\mu}\right)\left(\frac{\lambda}{\mu}\right)^{m}}{\left(1-p_0^{'}(t)^2\frac{\lambda}{\mu}\right)^{m+1}} \qquad (5.2.13)$$

Az (5.2.13) képlet megadja két szekvencia kapcsoltsági valószínűségét a TKF91 modell alapján. Mint látható, az (5.2.13) és az (5.2.5) képletek csak azokban a faktorokban különböznek, amelyekkel a maradékszorzatok vannak megszorozva, azaz a maradékszorzatok függetlenek az ősi szekvenciák hosszának az eloszlásától.

Empirikus eredmények azt mutatják, hogy a Poisson modell és a TKF91 modell alapján meghatározott maximum likelihood paraméterek szinte alig különböznek egymástól. A maradékszorzatok és a maradékszorzatokat szorzó faktorok vizsgálata megmagyarázza ezt a jelenséget. Az **5.2.1** ábra a maximum likelihood paraméterek mellett mutatja a maradékszorzatok és a faktorok értékeit. A két vizsgált szekvencia az emberi alfa- és béta-globin volt. Az ábrán jól megfigyelhető, hogy a maradékszorzatnak *m* függvényében egy maximuma van. A Poisson faktor éppen ennél az értéknél maximális, míg a geometriai faktor szigorúan monoton csökken *m* függvényében. A teljes likelihood értékét zömében a maximális érték körüli maradékszorzatok adják Ezért az ősi szekvenciák hosszának az

eloszlását meghatározó paramétereknek azok lesznek a maximum likelihood értékei, amelyekre a faktorok értékei maximálisak lesznek annál az *m*-nél, ahol a maradékszorzat maximális. Megmutatható, hogy egy adott *m*-re a Poisson és a geometriai faktornak ugyanannál a szekvenciahossz várható értéknél van a maximuma. Így mindkét modell ugyanazt a szekvenciahossz várható értéket adja meg, mint maximum likelihood paramétert.



5.2.1 ábra Maradékszorzat és faktor értékek *m* függvényében, logaritmikus skálán. A maradékszorzat esetében az érték logaritmusa még 30-cal el lett osztva, hogy a három görbe egy ábrán is jól ábrázolható legyen. A vizsgált két szekvencia az emberi alfa- és béta-globin volt. Az (5.2.5) képlet Poisson faktora lilával, az (5.2.13) képlet geometriai faktora pirossal, a maradékszorzat szürkével van jelölve.

Az **5.2.1** ábráról az is leolvasható, hogy a maradékszorzat maximumánál a Poisson faktor értéke lényegesen nagyobb, mint a geometriai faktoré. Így a Poisson modell maximum likelihood értéke nagyobb lesz, mint a TKF91 modellé. Ez azonban csak azt mutatja, hogy ha adott két rokon szekvencia, akkor a közös ősük hosszának az eloszlása közelebb áll a Poisson eloszláshoz, mint a geometriaihoz. Nem rokon szekvenciák esetében azonban változik a helyzet. Ahogy $p_0'(t)$ tart az egyhez, a szekvenciahossz várható értékénél (azaz ahol a maradékszorzat maximális) a geometriai faktor nagyobb lesz a Poisson faktornál. Ez közvetlenül adódik abból, hogy ha a geometriai eloszlás várható értéke megegyezik a Poisson eloszlás várható értékével, akkor

$$\kappa = \frac{\frac{\lambda}{\mu}}{1-\frac{\lambda}{\mu}} \qquad (5.2.14)$$

A fentiekből adódik, hogy nem homológ szekvenciák esetén a Poisson modell kisebb maximum likelihood értéket ad, mint a TKF91 modell. Ezek szerint a Hein és munkatársai által bemutatott (Hein et al., 2000) rokonsági tesztnél jobb tesztet kapunk, ha abban a tesztben a TKF91 modellt lecseréljük a Poisson modellre.

Összefoglalásként: habár a TKF91 modellben a geometriai szekvenciahossz eloszlás biológiailag irreleváns, a maximum likelihood paraméterek becslését ez a feltételezés nem befolyásolja. A Poisson modell egyik jelentősége ennek a kimutatása. A Poisson modell másik jelentősége az, hogy jobb statisztikai tesztet nyújt a szekvenciák rokonsági kapcsolatának a kimutatására. További jelentősége ennek a modellnek, hogy elősegítette egy gyorsabb algoritmus kidolgozását a többszörös statisztikus szekvencia illesztésben, mint az a következő fejezetből kiderül.

5.3 Egy megjavított algoritmus a többszörös statisztikus szekvencia illesztésre

Az (5.2.13) képlet két szekvencia kapcsoltsági valószínűségét számolja ki a TKF91 modellt feltételezve, összegezve az összes lehetséges ősi szekvencián. A számolás természetesen kiterjeszthető több szekvenciára is, amelyek egy csillag alakú fa mentén evolválódtak. Az (5.2.13) képlet közvetlen kiterjesztése egy $O(l^{2n+1})$ idejű algoritmushoz vezetne, ahol l a szekvenciák átlagos hossza, n pedig a szekvenciák száma. Az algoritmus futási ideje tovább csökkenthető $O(l^{2n})$-re, ha kihasználjuk a geometriai eloszlás tulajdonságait. Az algoritmust három szekvencia statisztikus illesztésén mutatom meg, több szekvenciára a kiterjesztés nyilvánvaló (Miklós, 2001b).

Az *A*, *B* és *C* szekvenciák kapcsoltsági valószínűsége definíció szerint az összes lehetséges leszármazási történet valószínűségének az összege. Egy adott leszármazást a három szekvencia illesztésével lehet bemutatni, amit $\alpha(A,B,C)$-vel jelölök. Hasonlóan az előző alfejezetben bemutatott illesztéshez, ez az illesztés is különbözik a hagyományos illesztéstől, mivel nem homológ karakterek kerülhetnek egymás alá. Homológnak csak azokat a karaktereket kell tekinteni, amelyek össze vannak kapcsolva egy ősi halandó linkkel. A

kapcsoltsági valószínűség számításakor figyelembe kell venni az összes lehetséges ősi karakterállapotot, így egyetlen egy illesztés több ősi szekvencia lehetséges leszármazását mutatja be egyszerre. Például, a következő illesztés azt mutatja, hogy a halhatatlan linknek nincs halandó link leszármazottja az *A* szekvenciában, viszont van egy-egy leszármazottja a *B* és a *C* szekvenciákban. Az első halandó link túlélt mindegyik szekvenciában, és van egy valódi leszármazottja a *B* szekvenciában. A második linktől az utolsó előttiig mindegyik halandó link kihalt. Az utolsó halandó link csak a *B* és *C* szekvenciákban élt túl, és van egy-egy leszármazottja az *A* és *B* szekvenciákban.

```
o   |  *   |  *  *     *  |  *
o -  | A* - |  -  -  ...  -  |  - A*
o U* | C* G*|  -  -  ...  -  | U* G*
o G* | C*  -|  -  -  ...  -  | A*  -
```

Ennek a speciális átmenetnek a valószínűsége

$$P(\alpha(A,B,C))=$$

$$p''_1(t_1)p''_2(t_2)p''_2(t_3)\pi(U)\pi(G)(1-\frac{\lambda}{\mu})\times p_1(t_1)p_2(t_2)p_1(t_3)\pi(G)\,F_{t_1,t_2,t_3}(A,C,C)\,\frac{\lambda}{\mu}\times \qquad (5.3.1)$$

$$\times\left(p'_0(t_1)\,p'_0(t_2)\,p'_0(t_3)\frac{\lambda}{\mu}\right)^n\times p'_1(t_1)p_2(t_2)p_1(t_3)\,\pi(U)\,f_{t_1+t_2}(A|U)\,\pi(A)\pi(G)\frac{\lambda}{\mu}$$

ahol *n* a kihalt linkek száma, $f_{t_1+t_2}(A|U)$ annak a valószínűsége, hogy egy nukleotid, amelyik U volt a nulladik időpontban, A lett t_1+t_2 idő után, $F_{t_1,t_2,t_3}(A,C,C)$ pedig A, C és C észlelése a fa terminális pontjain, amikor az élekhez rendelt időpontok rendre t_1, t_2 és t_3. Ez utóbbi könnyen meghatározható, Felsenstein algoritmusát használva (Felsenstein, 1981).

Ahogy a példából is látszik, minden illesztés, és ekképpen az illesztés valószínűsége is, blokkokra osztható. Két típusú blokk létezik. Az első típusú blokk egy olyan link sorsát írja le, amelyiknek legalább egy leszármazottja van. A második típusú blokk olyan linkek sorsát írja le, amelyek mind kihaltak. A linkek száma egy ilyen blokkban 1-től ∞-ig terjedhet. *S*(*A*,*B*,*C*) azon illesztések halmazát jelöli, amely illesztésben az utolsó blokk egyes típusú. *S*(*A*,*B*,*C*)-t 14 részhalmazra kell felosztani, az utolsó illesztett karakter vagy karakterek szerint.

Mindegyik link két osztály közül pontosan az egyiknek a tagja. Azokat a linkeket, amelyek az ősi szekvenciában is éltek már, *h*-val jelölöm. Az összes többi link jele *r*. Bevezetem a következő rövidítések halmazát: *ABB*={*ha*, *hb*, *hc*, *hab*, *hac*, *hbc*, *habc*, *ra*, *rb*, *rc*, ,*rab*, *rac*, *rbc*, *rabc*}. $S^{ha}(A,B,C)$ azon illesztések halmaza, amelyek utolsó blokkja egyes

típusú, az utolsó illesztett karaktere az *A* szekvenciában található meg, és ezen utolsó karakter linkje *h* típusú. $S^{rab}(A,B,C)$ azon illesztések halmaza, amelyek utolsó blokkja egyes típusú, az utolsó illesztett karakterei az *A* és a *B* szekvenciákhoz tartoznak, és ezen karaktere linkjei *r* típusúak, stb.

Legyen θ paraméterek halmaza, {*s*, λ, μ, t_1, t_2, t_3}.Az *A*, *B* és *C* szekvenciák likelihoodja

$$L_\theta(A,B,C) = P(A,B,C|\theta) = \sum_\alpha P(\alpha(A,B,C)|\theta) \tag{5.3.2}$$

Az *S*(*A*,*B*,*C*) halmaz mindegyik részhalmazának definiálom a likelihoodját

$$L_\theta^x(A,B,C) = \sum_{\alpha \in S^x(A,B,C)} P(\alpha(A,B,C)|\theta) \tag{5.3.3}$$

minden $x \in ABB$-re.

Jelölje az *A* szekvencia *i* hosszú prefix-ét A_i, a *B* szekvencia *j* hosszú prefix-ét B_j, a *C* szekvencia *k* hosszú prefix-ét C_k. Minden illesztésben, a legutolsó illesztett karakter vagy karakterek után áll néhány — 0-tól ∞-ig — ősi kihalt link. Ez egy $\left(p_0'(t_1) p_0'(t_2) p_0'(t_3) \frac{\lambda}{\mu} \right)^n$ faktor az illesztés valószínűségében, ahol *n* a kihalt linkek száma. Mivel

$$\sum_{n=0}^{\infty} \left(\frac{\lambda}{\mu} \prod_{i=1}^{3} p_0'(t_i) \right)^n = \frac{1}{1 - \frac{\lambda}{\mu} \prod_{i=1}^{3} p_0'(t_i)}, \tag{5.3.4}$$

az A_i, B_j és C_k prefixek likelihoodja

$$L_\theta(A_i, B_j, C_k) = \sum_{x \in ABB} L_\theta^x(A_i, B_j, C_k) \frac{1}{1 - \frac{\lambda}{\mu} \prod_{i=1}^{3} p_0'(t_i)}, \tag{5.3.5}$$

A dinamikus programozási algoritmus a következő rekurziós szabályokat követi

$$L_\theta^{ha}(A_i, B_j, C_k) = L_\theta(A_{i-1}, B_j, C_k) \frac{\lambda}{\mu} p_1(t_1) \pi(a_i) p_0'(t_2) p_0'(t_3), \tag{5.3.6}$$

hasonlóan $L_\theta^{hb}(A_i, B_j, C_k)$-ra és $L_\theta^{hc}(A_i, B_j, C_k)$-ra.

$$L_\theta^{hab}(A_i, B_j, C_k) = L_\theta(A_{i-1}, B_{j-1}, C_k) \frac{\lambda}{\mu} p_1(t_1) p_2(t_2) \pi(a_i) f_{t_1+t_2}(b_j | a_i) p_0'(t_3), \tag{5.3.7}$$

hasonlóan $L_\theta^{hac}(A_i, B_j, C_k)$-ra és $L_\theta^{hbc}(A_i, B_j, C_k)$-ra.

$$L_\theta^{habc}(A_i,B_j,C_k)=L_\theta(A_{i-1},B_{j-1},C_{k-1})\frac{\lambda}{\mu}p_1(t_1)p_1(t_2)p_1(t_3)F_{t_1,t_2,t_3}(A_i,B_j,C_k), \tag{5.3.8}$$

$$\begin{aligned}L_\theta^{ra}(A_i,B_j,C_k)&=L_\theta(A_{i-1},B_j,C_k)\frac{\lambda}{\mu}p_1'(t_1)\pi(a_i)p_0'(t_2)p_0'(t_3)+\\&+\sum_{\substack{x\in\{ha,hab,hac,habc,\\ra,rab,rac,rabc\}}}L_\theta^x(A_{i-1},B_j,C_k)\lambda\beta(t_1)\pi(a_i)+\\&+\sum_{y\in\{hb,hc,hbc\}}L_\theta^y(A_{i-1},B_j,C_k)\frac{p_1'(t_1)}{p_0'(t_1)}\pi(a_i)\end{aligned}, \tag{5.3.9}$$

hasonlóan $L_\theta^{rb}(A_i,B_j,C_k)$-ra és $L_\theta^{rc}(A_i,B_j,C_k)$-ra.

$$\begin{aligned}L_\theta^{rab}(A_i,B_j,C_k)&=L_\theta(A_{i-1},B_{j-1},C_k)\frac{\lambda}{\mu}p_1'(t_1)\pi(a_i)p_1'(t_2)\pi(b_j)p_0'(t_3)+\\&+\sum_{x\in\{hab,habc,rab,rabc\}}L_\theta^x(A_{i-1},B_{j-1},C_k)\lambda\beta(t_1)\pi(a_i)\lambda\beta(t_2)\pi(b_j)+\\&+(L_\theta^{ha}(A_{i-1},B_{j-1},C_k)+L_\theta^{hac}(A_{i-1},B_{j-1},C_k))\frac{p_1'(t_2)}{p_0'(t_2)}\pi(a_i)\pi(b_j)\lambda\beta(t_1)+,\\&+(L_\theta^{hb}(A_{i-1},B_{j-1},C_k)+L_\theta^{hbc}(A_{i-1},B_{j-1},C_k))\frac{p_1'(t_1)}{p_0'(t_1)}\pi(a_i)\pi(b_j)\lambda\beta(t_2)+\\&+L_\theta^{hc}(A_{i-1},B_{j-1},C_k)\frac{p_1'(t_1)}{p_0'(t_1)}\frac{p_1'(t_2)}{p_0'(t_2)}\pi(a_i)\pi(b_j)\end{aligned} \tag{5.3.10}$$

hasonlóan $L_\theta^{rac}(A_i,B_j,C_k)$-ra és $L_\theta^{rbc}(A_i,B_j,C_k)$-ra.

$$
\begin{aligned}
&L_\theta^{rabc}(A_i,B_j,C_k) = L_\theta(A_{i-1},B_{j-1},C_{k-1})\frac{\lambda}{\mu}p_1'(t_1)\pi(a_i)p_1'(t_2)\pi(b_j)p_1'(t_3)\pi(c_k) + \\
&+(L_\theta^{habc}(A_{i-1},B_{j-1},C_{k-1}) + L_\theta^{rabc}(A_{i-1},B_{j-1},C_{k-1}))\lambda\beta(t_1)\pi(a_i)\lambda\beta(t_2)\pi(b_j)\lambda\beta(t_3)\pi(c_k) + \\
&+(L_\theta^{ha}(A_{i-1},B_{j-1},C_{k-1})\frac{p_1'(t_2)p_1'(t_3)}{p_0'(t_2)p_0'(t_3)}\pi(a_i)\pi(b_j)\pi(c_k)\lambda\beta(t_1) + \\
&+L_\theta^{hb}(A_{i-1},B_{j-1},C_{k-1})\frac{p_1'(t_1)p_1'(t_3)}{p_0'(t_1)p_0'(t_3)}\pi(a_i)\pi(b_j)\pi(c_k)\lambda\beta(t_2) + \\
&+L_\theta^{hc}(A_{i-1},B_{j-1},C_{k-1})\frac{p_1'(t_1)p_1'(t_2)}{p_0'(t_1)p_0'(t_2)}\pi(a_i)\pi(b_j)\pi(c_k)\lambda\beta(t_3) + \\
&+L_\theta^{hab}(A_{i-1},B_{j-1},C_{k-1})\frac{p_1'(t_3)}{p_0'(t_3)}\pi(a_i)\pi(b_j)\pi(c_k)\lambda\beta(t_1)\lambda\beta(t_2) + \\
&+L_\theta^{hac}(A_{i-1},B_{j-1},C_{k-1})\frac{p_1'(t_2)}{p_0'(t_2)}\pi(a_i)\pi(b_j)\pi(c_k)\lambda\beta(t_1)\lambda\beta(t_3) + \\
&+L_\theta^{hbc}(A_{i-1},B_{j-1},C_{k-1})\frac{p_1'(t_1)}{p_0'(t_1)}\pi(a_i)\pi(b_j)\pi(c_k)\lambda\beta(t_2)\lambda\beta(t_3)
\end{aligned}
\tag{5.3.11}
$$

A kezdeti feltételek

$$
L_\theta^{habc}(A_0,B_0,C_0) = \left(1-\frac{\lambda}{\mu}\right)\prod_{i=1}^{3}p_1''(t_i), \tag{5.3.12}
$$

$$
L_\theta^{x}(A_0,B_0,C_0) = 0, \qquad x \in ABB \setminus \{habc\}, \tag{5.3.13}
$$

$$
L_\theta(A_0,B_0,C_0) = \left(1-\frac{\lambda}{\mu}\right)\prod_{i=1}^{3}p_1''(t_i)\frac{1}{1-\frac{\lambda}{\mu}\prod_{i=1}^{3}p_0'(t_i)}, \tag{5.3.14}
$$

A teljes szekvenciákra is érvényes az (5.3.5) képlet, amelyből közvetlenül adódik a három szekvencia likelihoodja, miután az *S*(*A*,*B*,*C*) halmaz mind a 14 részhalmazának a likelihoodja ismert.

Az algoritmus helyessége könnyen bizonyítható, végiggondolva, hogy milyen illesztésekből és hogyan származhattak az adott típusú illesztések

- (5.3.6-8) Ha az utolsó illesztett karakter vagy karakterek linkje (ill.) linkjei *h* típusúak, akkor ezeket elhagyva mindenképpen eggyel kevesebb blokkból áll az illesztés. Ezen típusú illesztések likelihoodja az egy blokkal rövidebb illesztés likelihoodja szorozva az utolsó blokk likelihood-jával.
- (5.3.9-11) Ha az utolsó illesztett karakter vagy karakterek linkje (ill.) linkjei *r* típusúak, akkor több eset lehetséges. Az első esetben az ősi link szülő egyetlen

szekvenciában sem élt túl. Ekkor elhagyva az utolsó illesztett hármast, olyan illesztéshez jutunk, amelyben eggyel kevesebb blokk van. A második esetben az ősi link szülőnek van legalább két utódja minden olyan szekvenciában, amelyből utolsó illesztett karakter származik. Ekkor az utolsó illesztett hármast elhagyva olyan illesztést kapunk, amelyben ugyanannyi halandó linknek van leszármazottja, csak a blokk likelihoodja valahányszor $\lambda\beta(t_i)\pi(x)$ faktorral kevesebb $i \in \{1,2,3\}$és $x \in \{a_i, b_j, c_k\}$, függve attól, hogy mely szekvenciákból származnak az utolsó karakterek. A többi esetekben az ősi link szülő egyes szekvenciákban kihalt, de legalább egyben túlélt. Ezeket azért kell külön kezelni, mert $p'_1(t_i) \neq \lambda\beta(t_i)$.

Több szekvenciára az algoritmus hasonlóan működik, *n* szekvencia esetén az *ABB* halmaz 2^{n+1}-2 elemből áll. Az algoritmus jelenlegi formájában csak három-négy szekvencia likelihoodját tudja elfogadható idő alatt kiszámolni. Ebben a formában alkalmazni lehet olyan lokális optimalizálási algoritmusokban, mint a Sankoff és munkatársai algoritmusa (Sankoff et al., 1976). Nevezetesen, legyen adva egy gyökerezetlen bináris fa, a leveleken adott szekvenciákkal. A fa belső pontjaira helyezzük el az adott ponthoz legközelebb eső levél szekvenciáját. Ha több ilyen van, véletlenszerűen válasszuk az egyiket. Ezután kívülről befelé, majd belülről kifelé haladva minden hármasra végezzük el az illesztést, és minden egyes illesztésnél a háromágú csillag közepét cseréljük le a konszenzusszekvenciára. Statisztikus illesztés esetén ez az a szekvencia, amelyből származó illesztések összes likelihoodja maximális a maximum likelihood paramétereket véve. Az iteráció addig tart, ameddig valamelyik belső ponton van változás.

A bemutatott algoritmust lehetséges kombinálni a sarokvágási technikával, melyről a hetedik fejezetben lesz szó. Az ismertetett illesztés valamint algoritmus és a TKF91 modell HMM-ként (Hidden Markov Model) való leírása közötti hasonlóságot a 8.3 fejezetben mutatom be.

Ezen alfejezet végén pedig megmutatom az ebben a fejezetben tárgyalt algoritmus és az előző fejezetben ismertetett maradékszorzat technika közötti kapcsolatot.

Egy olyan illesztésben, amely m+1 egyes típusú blokkot tartalmaz, m db halandó link található, mivel a halhatatlan link sorsát leíró blokk mindig egyes típusú. Így egy ilyen illesztés valószínűségéből elhagyva a kettes típusú blokkokat és az egyes típusú blokkokból az $\left(1-\frac{\lambda}{\mu}\right)$ illetve $\frac{\lambda}{\mu}$ faktorokat *m*-maradékszorzatot kapunk. Az *m*-maradékszorzat faktora pedig éppen

$$\frac{\left(1-\frac{\lambda}{\mu}\right)\left(\frac{\lambda}{\mu}\right)^{m}}{\left(1-p_0'(t_1)p_0'(t_2)p_0'(t_3)\frac{\lambda}{\mu}\right)^{m+1}} \qquad (5.2.13)$$

ami az egyes típusú blokkokból elhagyott faktorok és a lehetséges kettes típusú blokkok valószínűségeinek a szorzata, összegezve az összes lehetséges kettes típusú blokkra. A gyorsítás $O(l^{n+1})$-ről $O(l^n)$-re azért lehetséges, mert a maradékszorzat úgy szorzódik a faktorával, hogy a halhatatlan link blokkja egy

$$\frac{1-\frac{\lambda}{\mu}}{1-p_0'(t_1)p_0'(t_2)p_0'(t_3)\frac{\lambda}{\mu}} \qquad (5.2.13)$$

faktort kap, a halandó linkek blokkjai pedig

$$\frac{\frac{\lambda}{\mu}}{1-p_0'(t_1)p_0'(t_2)p_0'(t_3)\frac{\lambda}{\mu}} \qquad (5.2.13)$$

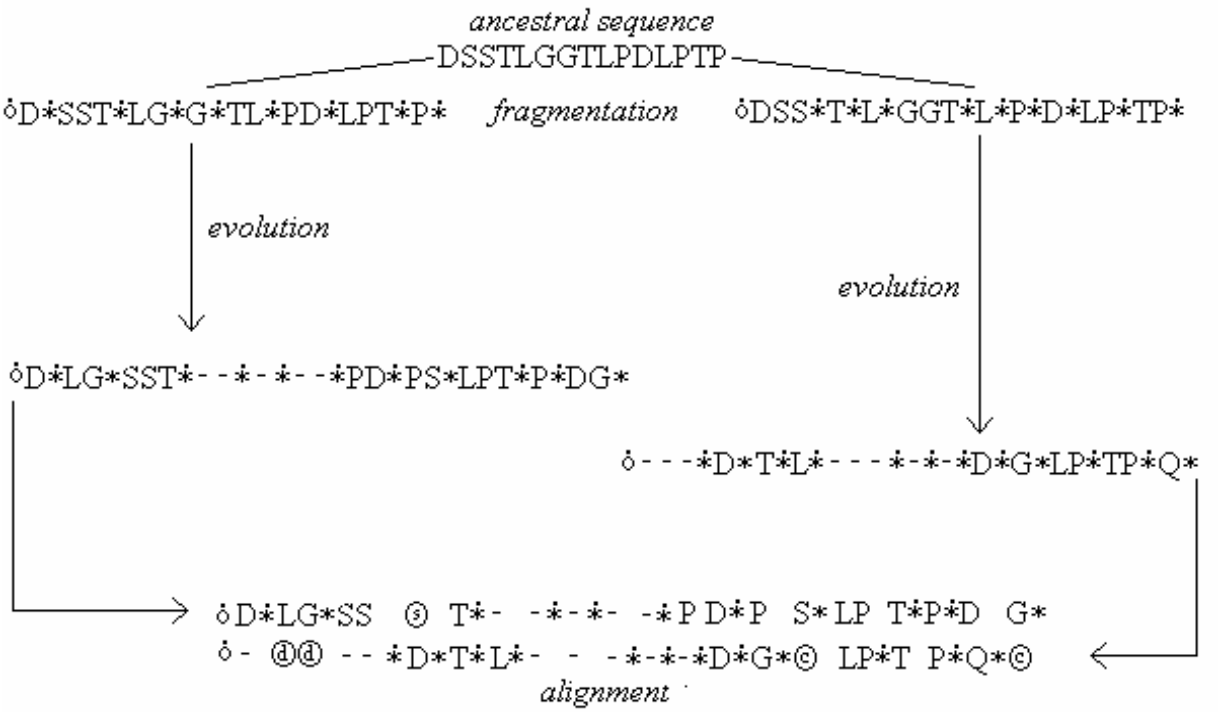
faktort kapnak *függetlenül* attól, hogy ez az illesztés hányadik egyes típusú blokkja. Ez a függetlenség teszi lehetővé az előbb említett algoritmikai gyorsítást.

5.4 Egy ötlet a TKF92 (fragmentum) modell javítására

A TKF92 modell egyik hibája, hogy nem tud átfedő törléseket két eseménnyel modellezni. Most, hogy sikerült az összes lehetséges ősi szekvencia összegezésére $O(l^2)$ idejű algoritmust készíteni, lehetővé válik egy olyan javított fragmentum modell kidolgozása, amelyre szintén van $O(l^2)$ idejű algoritmus (Miklós, 2001a). A modell a következő (ld. **5.4.1** ábra):

A két szekvencia szétválásával egy időben az ősi szekvencia fragmentálódott mindkét ágon, a másik ágtól függetlenül, a következő módon. A következő karakter az előzőhöz kapcsolódik — és így osztozik az előző karakter linkjével — r valószínűséggel, és egy új fragmentumot kezd el 1-r valószínűséggel. A szekvenciák egymástól függetlenül evolválódnak mindkét ágon a szétválás után, a TKF92 modellnek megfelelően. Definíció szerint két szekvencia, *A* és *B*, likelihoodja

$$L_\theta(A,B) = \sum_C P(C|\theta)\left\{\sum_{frag_1} P(frag_1|\theta)\left(\sum_{frag_2} P(frag_2|\theta)P_t(A|C, frag_1, \theta)P_t(B|C, frag_2, \theta)\right)\right\} \qquad (5.4.1)$$



5.4.1 ábra Két szekvencia evolúciója a javított fragmentum modell alapján. A magyarázatot ld. a szövegben.

Ezt a modellt úgy is lehet értelmezni, mintha a szekvenciák evolúciója predesztinálva volna, de a likelihood számításakor minden predesztináción összegzek, és minden predesztináció kap egy olyan faktort, ami arányos a predesztinált esemény valószínűségével.

Az evolúciós folyamatot a modern szekvenciák illesztésével lehet reprezentálni. Algoritmikai okokból — ahogy az előző két alfejezetben is — nem homológ karakterek kerülhetnek egymás alá. Ezért az ősi linkek egy ponttal vannak megjelölve, hogy a homológ és nem homológ karaktereket meg lehessen egymástól különböztetni. Az olyan illesztett karaktereket, amelyek linkje nincs egy ponttal megjelölve, nem homológokként kell kezelni. Szintén algoritmikai okokból, azok a leszármazott linkek karakterei, amelyek nincsenek a másik szekvencia egy karakterével illesztve, nem a szokásos gap jellel (-) vannak társítva, hanem a következő szimbólumok egyikével (ld. az **5.4.1** ábrát):

- s egy körben: ez a szimbólum azt jelöli, hogy az ezt megelőző link túlélt. A túlélt fragmentum folytatódhat a másik szekvencia valódi leszármazottjai után, de nem feltétlenül.

- d egy körben: ez a szimbólum azt jelenti, hogy az ezt megelőző ősi link kihalt. A fragmentumának a homológ párjai a másik szekvenciában folytatódhatnak, de nem feltétlenül.
- c egy körben: a 'lezárás' szimbóluma. A legutolsó illesztett link egy valódi leszármazott volt, ezért sem túlélt, sem kihalt fragmentum nem folytatódhat az adott illesztett pár után.

Egy adott illesztést $\alpha(A,B)$-vel jelölök. Az $S(A,B)$ halmaz azon illesztések halmaza, amelyeknek az utolsó illesztett párja legalább egy túlélt karaktert tartalmaz. $S(A,B)$ 10 részhalmazra bontható, az utolsó illesztett pár alapján, a következő módon.

Bevezetem a rövidítések egy halmazát, $ABB=\{ha,hb,hab,ras,rad,rac,rbs,rbd,rbc,rab\}$. $S^{ha}(A,B)$ az a részhalmaz, amely olyan illesztéseket tartalmaz, amelyeknek a legutolsó illesztett karaktere az A szekvenciához tartozik, és az ehhez a karakterhez kapcsolt link már élt az ősi szekvenciában is (azaz h típusú). $S^{rab}(A,B)$ az a részhalmaz, amelyeknek a legutolsó illesztett párja két karakter, és a karakterekhez asszociált linkek valódi leszármazottak. $S^{rad}(A,B)$ az a részhalmaz, amelynek az utolsó illesztett karaktere egy valódi leszármazott, és ez a karakter egy 'd betű egy körben'-nel van társítva, stb.

Legyen θ a paraméterek halmaza, $\{s,\lambda,\mu,r,q\}$, ahol q az ősi szekvenciahossz geometriai eloszlásának a paramétere, r a fragmentumok hosszát leíró geometriai eloszlás paramétere. Az A és B szekvenciák likelihoodja

$$L_\theta(A,B) = P(A,B|\theta) = \sum_\alpha P(\alpha(A,B)|\theta) \tag{5.4.2}$$

Az $S(A,B)$ halmaz mindegyik részhalmazának definiálom a likelihoodját

$$L_\theta^x(A,B) = \sum_{\alpha \in S^x(A,B)} P(\alpha(A,B)|\theta) \tag{5.4.3}$$

minden $x \in ABB$-re

Habár a modern szekvenciákkal csak a túlélt site-ok ismertek, illeszteni kell a kihalt site-okat is. Ezért feltételezni kell, hogy minden két, nem kihalt site között van néhány — 1-től ∞-ig — kihalt site, ahol csak ez lehetséges. Egy kihalt pár pontosan az egyike a következő négy típusnak

- Mindkét site egy új fragmentumot kezd el kialakítani
- Mindkét site folytatása egy-egy megkezdett fragmentumnak
- Az első site egy új fragmentumot alakít ki, a második folytatása egy fragmentumnak.

- Az első site folytatása egy fragmentumnak, a második egy új fragmentumot kezd el kialakítani.

Ekképpen a kihalt site-okból származó faktorok egy geometriai sorozatot alkotnak, $q[r + p_0'(t_1)(1-r)][r + p_0'(t_2)(1-r)]$ kvócienssel. A geometriai sorozat első tagja azonban függ attól, hogy mi volt az utolsó olyan illesztett pár, amelynek legalább az egyik tagja túlélt. Ha ez a pár *hab*, *rab*, *ras*, *rbs*, *rac* vagy *rbc*, akkor az első kihalt pár mindkét site-jának szükségképpen egy új fragmentumot kell kialakítania. Ha az utolsó illesztett pár *ha* vagy *rad*, akkor a *B* szekvenciához tartozó site nem szükségképpen kezd egy új fragmentumot. Ha az utolsó illesztett pár *hb* vagy *rbd*, akkor pedig az *A* szekvenciához tartozó site-nak nem kell szükségszerűen egy új fragmentumot kialakítani. Ennek megfelelően, *A* és *B* szekvencia likelihoodja

$$L_\theta(A,B) = \sum_{\substack{x\in\{hab,rab,ras,\\ rac,rbs,rbc\}}} L_\theta^x(A,B)\left[1+\frac{qp_0'(t_1)(1-r)p_0'(t_2)(1-r)}{1-q[r+p_0'(t_1)(1-r)][r+p_0'(t_2)(1-r)]}\right] +$$
$$+ \sum_{y\in\{ha,rad\}} L_\theta^x(A,B)\left[1+\frac{qp_0'(t_1)(1-r)[1+p_0'(t_2)(1-r)]}{1-q[r+p_0'(t_1)(1-r)][r+p_0'(t_2)(1-r)]}\right] + \qquad (5.4.4)$$
$$+ \sum_{z\in\{hb,rbd\}} L_\theta^z(A,B)\left[1+\frac{[r+qp_0'(t_1)(1-r)]p_0'(t_2)(1-r)}{1-q[r+p_0'(t_1)(1-r)][r+p_0'(t_2)(1-r)]}\right]$$

A cél tehát dinamikus programozási algoritmust felírni minden $L_\theta^x(A_i,B_j)$, $x\in ABB$-re. A dinamikus programozási algoritmus a következő rekurziós formulákat követi:

$$L_\theta^{ras}(A_i,B_j) = L_\theta^{ras}(A_{i-1},B_j)[r+(1-r)\lambda\beta(t_1)]\pi(a_i) + \qquad (5.4.5)$$
$$+ L_\theta^{hab}(A_{i-1},B_j)(1-r)\lambda\beta(t_1)\pi(a_i) + L_\theta^{hb}(A_{i-1},B_j)\frac{p_1'(t_1)}{p_0'(t_1)}(1-r)\pi(a_i)$$

$$L_\theta^{rbs}(A_i,B_j) = L_\theta^{rbs}(A_i,B_{j-1})[r+(1-r)\lambda\beta(t_2)]\pi(b_j) + \qquad (5.4.6)$$
$$+ L_\theta^{hab}(A_i,B_{j-1})(1-r)\lambda\beta(t_2)\pi(b_j) + L_\theta^{ha}(A_i,B_{j-1})\frac{p_1'(t_2)}{p_0'(t_2)}(1-r)\pi(b_j)$$

$$L_\theta^{rac}(A_i,B_j) = [L_\theta^{rac}(A_{i-1},B_j) + L_\theta^{rab}(A_{i-1},B_j)][r+(1-r)\lambda\beta(t_1)]\pi(a_i) \qquad (5.4.7)$$

$$L_\theta^{rbc}(A_i,B_j)=[L_\theta^{rbc}(A_i,B_{j-1})+L_\theta^{rab}(A_i,B_{j-1})][r+(1-r)\lambda\beta(t_2)]\pi(b_j) \tag{5.4.8}$$

$$\begin{aligned}L_\theta^{rad}(A_i,B_j)&=L_\theta^{rad}(A_{i-1},B_j)[r+(1-r)\lambda\beta(t_1)]\pi(a_i)+L_\theta^{ha}(A_{i-1},B_j)(1-r)\lambda\beta(t_1)\pi(a_i)+\\&+\sum_{\substack{x\in\{hab,rab,ras,\\rbs,rac,rbc\}}}L_\theta^{x}(A_{i-1},B_j)\frac{qp_1'(t_1)p_0'(t_2)(1-r)^2}{1-q[r+p_0'(t_1)(1-r)][r+p_0'(t_2)(1-r)]}(1-r)\pi(a_i)+\\&+\sum_{y\in\{hb,rbd\}}L_\theta^{y}(A_{i-1},B_j)\frac{qp_0'(t_2)(1-r)[r+p_0'(t_1)(1-r)]}{1-q[r+p_0'(t_1)(1-r)][r+p_0'(t_2)(1-r)]}\frac{p_1'(t_1)}{p_0'(t_1)}(1-r)\pi(a_i)+\\&+\sum_{z\in\{ha,rad\}}L_\theta^{z}(A_{i-1},B_j)\frac{qp_1'(t_1)(1-r)[r+p_0'(t_2)(1-r)]}{1-q[r+p_0'(t_1)(1-r)][r+p_0'(t_2)(1-r)]}(1-r)\pi(a_i)\end{aligned} \tag{5.4.9}$$

$$\begin{aligned}L_\theta^{rbd}(A_i,B_j)&=L_\theta^{rbd}(A_i,B_{j-1})[r+(1-r)\lambda\beta(t_2)]\pi(b_j)+L_\theta^{hb}(A_i,B_{j-1})(1-r)\lambda\beta(t_2)\pi(b_j)+\\&+\sum_{\substack{x\in\{hab,rab,ras,\\rbs,rac,rbc\}}}L_\theta^{x}(A_i,B_{j-1})\frac{qp_1'(t_2)p_0'(t_1)(1-r)^2}{1-q[r+p_0'(t_1)(1-r)][r+p_0'(t_2)(1-r)]}(1-r)\pi(b_j)+\\&+\sum_{z\in\{ha,rad\}}L_\theta^{z}(A_i,B_{j-1})\frac{qp_0'(t_1)(1-r)[r+p_0'(t_2)(1-r)]}{1-q[r+p_0'(t_1)(1-r)][r+p_0'(t_2)(1-r)]}\frac{p_1'(t_2)}{p_0'(t_2)}(1-r)\pi(b_j)+\\&+\sum_{y\in\{hb,rbd\}}L_\theta^{y}(A_i,B_{j-1})\frac{qp_1'(t_2)(1-r)[r+p_0'(t_1)(1-r)]}{1-q[r+p_0'(t_1)(1-r)][r+p_0'(t_2)(1-r)]}(1-r)\pi(b_j)\end{aligned} \tag{5.4.10}$$

$$\begin{aligned}L_\theta^{rab}(A_i,B_j)&=L_\theta^{rab}(A_{i-1},B_{j-1})[r+(1-r)\lambda\beta(t_1)][r+(1-r)\lambda\beta(t_2)]\pi(a_i)\pi(b_j)+\\&+L_\theta^{hab}(A_{i-1},B_{j-1})(1-r)^2\lambda\beta(t_1)\lambda\beta(t_2)\pi(a_i)\pi(b_j)+\\&+L_\theta^{ha}(A_{i-1},B_{j-1})\frac{p_1'(t_2)}{p_0'(t_2)}(1-r)^2\lambda\beta(t_1)\pi(a_i)\pi(b_j)+\\&+L_\theta^{hb}(A_{i-1},B_{j-1})\frac{p_1'(t_1)}{p_0'(t_1)}(1-r)^2\lambda\beta(t_2)\pi(a_i)\pi(b_j)+\\&+\sum_{\substack{x\in\{hab,rab,ras,\\rbs,rac,rbc\}}}L_\theta^{x}(A_{i-1},B_{j-1})\frac{qp_1'(t_1)p_1'(t_2)(1-r)^2}{1-q[r+p_0'(t_1)(1-r)][r+p_0'(t_2)(1-r)]}(1-r)^2\pi(a_i)\pi(b_j)+\\&+\sum_{y\in\{ha,rad\}}L_\theta^{y}(A_{i-1},B_{j-1})\frac{qp_1'(t_1)(1-r)[r+p_0'(t_2)(1-r)]}{1-q[r+p_0'(t_1)(1-r)][r+p_0'(t_2)(1-r)]}\frac{p_1'(t_2)}{p_0'(t_2)}(1-r)^2\pi(a_i)\pi(b_j)+\\&+\sum_{z\in\{hb,rbd\}}L_\theta^{z}(A_{i-1},B_{j-1})\frac{qp_1'(t_2)(1-r)[r+p_0'(t_1)(1-r)]}{1-q[r+p_0'(t_1)(1-r)][r+p_0'(t_2)(1-r)]}\frac{p_1'(t_1)}{p_0'(t_1)}(1-r)^2\pi(a_i)\pi(b_j)\end{aligned} \tag{5.4.11}$$

$$
\begin{aligned}
&L_\theta^{hab}(A_i,B_j)=L_\theta^{hab}(A_{i-1},B_{j-1})q\left[r(1-r)[p_1(t_1)+p_1(t_2)]+r^2\right]f_{t_1+t_2}(b_j|a_i)\pi(a_i)\\
&+[L_\theta^{ha}(A_{i-1},B_{j-1})p_1(t_2)+L_\theta^{hb}(A_{i-1},B_{j-1})p_1(t_1)]qr(1-r)f_{t_1+t_2}(b_j|a_i)\pi(a_i)+\\
&+[L_\theta^{ras}(A_{i-1},B_{j-1})p_1(t_1)+L_\theta^{rbs}(A_{i-1},B_{j-1})p_1(t_2)]qr(1-r)f_{t_1+t_2}(b_j|a_i)\pi(a_i)+\\
&+\left\{\sum_{\substack{x\in\{hab,rab,ras,\\ rbs,rac,rbc\}}}L_\theta^{x}(A_{i-1},B_{j-1})\left[1+\frac{qp_0^{'}(t_1)p_0^{'}(t_2)(1-r)^2}{1-q[r+p_0^{'}(t_1)(1-r)][r+p_0^{'}(t_2)(1-r)]}\right]+\right.\\
&+\sum_{y\in\{ha,rad\}}L_\theta^{y}(A_{i-1},B_{j-1})\left[1+\frac{qp_0^{'}(t_1)(1-r)[r+p_0^{'}(t_2)(1-r)]}{1-q[r+p_0^{'}(t_1)(1-r)][r+p_0^{'}(t_2)(1-r)]}\right]+\\
&\left.+\sum_{z\in\{hb,rbd\}}L_\theta^{z}(A_{i-1},B_{j-1})\left[1+\frac{q[r+p_0^{'}(t_1)(1-r)]p_0^{'}(t_2)(1-r)}{1-q[r+p_0^{'}(t_1)(1-r)][r+p_0^{'}(t_2)(1-r)]}\right]\right\}\times\\
&\times q(1-r)^2p_1(t_1)p_1(t_2)f_{t_1+t_2}(b_j|a_i)\pi(a_i)
\end{aligned}
\tag{5.4.12}
$$

$$
\begin{aligned}
&L_\theta^{ha}(A_i,B_j)=L_\theta^{ha}(A_{i-1},B_j)q\left[r^2+[p_0^{'}(t_2)+p_1(t_1)]r(1-r)\right]\pi(a_i)+\\
&+\left[L_\theta^{hab}(A_{i-1},B_j)+L_\theta^{rbs}(A_{i-1},B_j)\right]qrp_0^{'}(t_2)(1-r)\pi(a_i)+\\
&+\sum_{\substack{x\in\{hab,rab,ras,\\ rbs,rac,rbc\}}}L_\theta^{x}(A_{i-1},B_j)\left[1+\frac{qp_0^{'}(t_1)(1-r)[r+p_0^{'}(t_2)(1-r)]}{1-q[r+p_0^{'}(t_1)(1-r)][r+p_0^{'}(t_2)(1-r)]}\right]\times\\
&\times q(1-r)^2p_0^{'}(t_2)p_1(t_1)\pi(a_i)+\\
&+\sum_{y\in\{ha,rad\}}L_\theta^{y}(A_{i-1},B_j)\left[1+\frac{qp_0^{'}(t_1)(1-r)[r+p_0^{'}(t_2)(1-r)]}{1-q[r+p_0^{'}(t_1)(1-r)][r+p_0^{'}(t_2)(1-r)]}\right]\times\\
&\times q(1-r)p_1(t_1)[r+p_0^{'}(t_2)(1-r)]\pi(a_i)+\\
&+\sum_{z\in\{hb,rbd\}}L_\theta^{z}(A_{i-1},B_j)\left[1+\frac{q[r+p_0^{'}(t_1)(1-r)][r+p_0^{'}(t_2)(1-r)]}{1-q[r+p_0^{'}(t_1)(1-r)][r+p_0^{'}(t_2)(1-r)]}\right]\times\\
&\times q(1-r)^2p_1(t_1)p_0^{'}(t_2)\pi(a_i)
\end{aligned}
\tag{5.4.13}
$$

$$
\begin{aligned}
&L_\theta^{hb}(A_i,B_j) = L_\theta^{hb}(A_i,B_{j-1})q\left[r^2+[p_1(t_2)+p_0^{'}(t_1)]r(1-r)\right]\pi(b_j)+\\
&+\left[L_\theta^{hab}(A_i,B_{j-1})+L_\theta^{ras}(A_i,B_{j-1})\right]qrp_0^{'}(t_1)(1-r)\pi(b_j)+\\
&+\sum_{\substack{x\in\{hab,rab,ras,\\ rbs,rac,rbc\}}} L_\theta^{x}(A_i,B_{j-1})\left[1+\frac{q[r+p_0^{'}(t_1)(1-r)]p_0^{'}(t_2)(1-r)}{1-q[r+p_0^{'}(t_1)(1-r)][r+p_0^{'}(t_2)(1-r)]}\right]\times\\
&\times q(1-r)^2 p_0^{'}(t_1)p_1(t_2)\pi(b_j)+\\
&+\sum_{y\in\{hb,rbd\}} L_\theta^{y}(A_i,B_{j-1})\left[1+\frac{qp_0^{'}(t_2)(1-r)[r+p_0^{'}(t_1)(1-r)]}{1-q[r+p_0^{'}(t_1)(1-r)][r+p_0^{'}(t_2)(1-r)]}\right]\times\\
&\times q(1-r)p_1(t_2)[r+p_0^{'}(t_1)(1-r)]\pi(b_j)+\\
&+\sum_{z\in\{ha,rad\}} L_\theta^{z}(A_1,B_{j-1})\left[1+\frac{q[r+p_0^{'}(t_2)(1-r)][r+p_0^{'}(t_1)(1-r)]}{1-q[r+p_0^{'}(t_1)(1-r)][r+p_0^{'}(t_2)(1-r)]}\right]\times\\
&\times q(1-r)^2 p_1(t_2)p_0^{'}(t_1)\pi(b_j)
\end{aligned}
\tag{5.4.14}
$$

A fenti képletekkel lehet a dinamikus programozási táblázat belsejét kitölteni. Az első sor és oszlop kitöltése azonban más formulákat követ, mert az első ősi site szükségképpen egy új fragmentumot kezd. Így a kezdeti feltételek

$$
L_\theta^{rac}(A_1,B_0) = \frac{(1-q)p_2^{''}(t_1)p_1^{''}(t_2)\pi(a_1)}{(1-r)^2} \tag{5.4.15}
$$

$$
\begin{aligned}
&L_\theta^{ha}(A_1,B_0) = (1-q)p_1^{''}(t_1)p_1^{''}(t_2)qp_1(t_1)p_0^{'}(t_2)\pi(a_1)+\\
&+(1-q)p_1^{''}(t_1)p_1^{''}(t_2)qp_1(t_1)(1-r)[r+p_0^{'}(t_2)(1-r)]\pi(a_1)\frac{qp_0^{'}(t_1)p_0^{'}(t_2)}{1-q[r+p_0^{'}(t_1)(1-r)][r+p_0^{'}(t_2)(1-r)]}
\end{aligned}
\tag{5.4.16}
$$

$$
L_\theta^{rad}(A_1,B_0) = (1-q)p_1^{''}(t_1)p_1^{''}(t_2)(1-r)\pi(a_1)\frac{qp_1^{'}(t_1)p_0^{'}(t_2)}{1-q[r+p_0^{'}(t_1)(1-r)][r+p_0^{'}(t_2)(1-r)]} \tag{5.4.17}
$$

$$
L_\theta^{x}(A_1,B_0) = 0, \qquad x\in ABB\setminus\{rac,rad,ha\} \tag{5.4.18}
$$

$$
L_\theta^{rbc}(A_0,B_1) = \frac{(1-q)p_1^{''}(t_1)p_2^{''}(t_2)\pi(b_1)}{(1-r)^2} \tag{5.4.19}
$$

$$L_\theta^{hb}(A_0,B_1)=(1-q)\,p_1^{"}(t_1)\,p_1^{"}(t_2)\,qp_0^{'}(t_1)\,p_1(t_2)\pi(b_1)+$$
$$+(1-q)\,p_1^{"}(t_1)\,p_1^{"}(t_2)\,q[r+p_0^{'}(t_1)(1-r)]\,p_1(t_2)(1-r)\pi(b_1)\frac{qp_0^{'}(t_1)\,p_0^{'}(t_2)}{1-q[r+p_0^{'}(t_1)(1-r)][r+p_0^{'}(t_2)(1-r)]} \tag{5.4.20}$$

$$L_\theta^{rbd}(A_0,B_1)=(1-q)\,p_1^{"}(t_1)\,p_1^{"}(t_2)(1-r)\pi(b_1)\frac{qp_0^{'}(t_1)\,p_1^{'}(t_2)}{1-q[r+p_0^{'}(t_1)(1-r)][r+p_0^{'}(t_2)(1-r)]} \tag{5.4.21}$$

$$L_\theta^{x}(A_0,B_1)=0,\qquad x\in ABB\setminus\{rbc,rbd,hb\} \tag{5.4.22}$$

$$L_\theta^{hab}(A_1,B_1)=(1-q)\,p_1^{"}(t_1)\,p_1^{"}(t_2)\,qp_1(t_1)\,p_1(t_2)\,f_{t_1+t_2}(b_1|a_1)\pi(a_1)\times$$
$$\left[1+\frac{qp_0^{'}(t_1)\,p_0^{'}(t_2)(1-r)^2}{1-q[r+p_0^{'}(t_1)(1-r)][r+p_0^{'}(t_2)(1-r)]}\right] \tag{5.4.23}$$

$$L_\theta^{rab}(A_1,B_1)=\frac{(1-q)\,p_2^{"}(t_1)\,p_2^{"}(t_2)\pi(a_1)\pi(b_1)}{(1-r)^2}+$$
$$(1-q)\,p_1^{"}(t_1)\,p_1^{"}(t_2)\pi(a_1)\pi(b_1)\frac{qp_0^{'}(t_1)\,p_0^{'}(t_2)}{1-q[r+p_0^{'}(t_1)(1-r)][r+p_0^{'}(t_2)(1-r)]} \tag{5.4.24}$$

Ez a rekurzió minden egyes illesztés valószínűségét helyesen adja meg, kivétel azt, amelyikben nincs ősi halandó link. Így ha a rekurzió a Λ értékkel tér vissza, akkor a két szekvencia likelihoodja

$$L_\theta(A,B)=\Lambda+(1-q)\,p_{l(A)}^{"}(t_1)\,p_{l(B)}^{"}(t_2)\prod_{i=1}^{l(A)}\pi(a_i)\prod_{j=1}^{l(B)}\pi(b_j)\left[1-\frac{1}{(1-r)^2}\right] \tag{5.4.25}$$

A bemutatott modell ugyan jobb, mint a TKF92 modell, de a többszörös beszúrás-törlés problémáját mégsem oldja meg teljes egészében, mert problémák vannak a modellnek több szekvenciára való kiterjesztésével. Például, tegyük fel, hogy emberi, csimpánzból és gorillából származó szekvenciákat kell összehasonlítani. Ekkor két fragmentáció párt kell feltételezni, egyet a gorilla és csimpánz-ember ős szétválásnál, és egy újrafregmentálódást a csimpánz-ember szétválásnál. Azonban, ha csak az emberi és a gorillából származó szekvenciákat elemezzük, csak egyetlen fragmentáció párt feltételez a modell, vagyis ekkor a csimpánz-ember szétválást nem vettük figyelembe. Pedig tudjuk, hogy volt egy ember-csimpánz szétválás, és ezt figyelembe kellene venni akkor is, ha csak az emberi és a gorillából származó szekvenciák adottak, és a csimpánzból származó szekvencia nem ismert. Egy

lehetséges megoldás lehetne az, hogy a csimpánz-ember szétválásnál nem egy pár, hanem csak egyetlen újrafragmentálódást tételezünk fel. Ezzel a kiterjesztéssel az a baj, hogy más-más eredményt kapunk attól függően, hogy az újrafragmentálódás az emberi vagy a csimpánz vonalon történt.

Egy másik továbbfejlesztési lehetőség az lenne, ha két szekvencia esetén is több újrafragmentálódást tételeznénk fel, úgy, hogy miközben az újrafragmentálódások számával tartanánk a végtelenhez, a két szekvencia közötti időintervallum egyre finomabb felosztását kapnánk, azaz bármely két fragmentálódás közötti időintervallum tartana a 0-hoz. Határátmenetben egy olyan modellt kapnánk, amely bármely időpillanatban bármilyen hosszú fragmentum beszúrását és törlését engedélyezi. Azonban jelen algoritmus szerint minden egyes fragmentálódás az $S(A,B)$ halmaz újabb és újabb részhalmazokra történő felbontását kívánja meg. Így határátmenetben végtelen sok részhalmazra kellene felbontani a lehetséges illesztéseket, tehát ez az út jelenleg járhatatlan. Egy ilyen továbbfejlesztéshez először szükségképpen egy olyan javítását kellene megadni az algoritmusnak, amely nem követeli meg az illesztések csoportosítását. Jelenleg ilyen algoritmus nem létezik, de az eredeti TKF91 modell esetében már adtak meg ilyen algoritmust (Hein et al., 2000) ld. 4.7 fejezet. Egy ilyen javított algoritmus esetén elképzelhető lenne, hogy akárhány újrafragmentálódás esetén a dinamikus programozási algoritmus olyan egyszerű maradjon, mint a távolságalapú dinamikus programozási algoritmusok, persze közben a rekurziót leíró függvények egyre bonyolultabbá válnának. A feladat ezek után megtalálni a rekurziót leíró függvények sorozatainak a határfüggvényeit.