Dizi Hizalama [Sequence Alignment] ve BLOSUM62

DNA molekülleri nükleotidlerin uzun birer dizileridir. DNA dizisi 4 harften oluşan $\mathbb{A} = \{A,G,C,T\}$ gibi bir alfabe ile harfleri bitişik olarak yazılmış bir yazıdır denebilir. DNA dizisi pürin-primidin açısından okunmak istendiğinde $A=\{P,H\}$ gibi iki harfli (P-pürin, H-primidin) bir alfabe kullanılmaktadır. Proteinler amino asitlerin uzun birer dizileridir. Amino asit dizisi $\mathcal{A} = \{A,R,N,D,B,C,Q,E,Z,G,H,I,L,K,M,F,P,S,T,W,Y,Z\}$ gibi bir alfabe ile harfleri bitişik olarak yazılmış bir yazı olarak görülebilir.

Amino Asit	Kod (1 karakterli)
Alanine	A
Arginine	R
Asparagine	N
Aspartic acid	D
Asparagine	B
Cysteine	C
Glutamine	Q
Glutamic acid	E
Glutamine	Z
Glycine	G
Histidine	H
Isoleucine	I
Leucine	L
Lysine	K
Methionine	M
Phenylalanine	F
Proline	P
Serine	S
Threonine	T
Trytophan	W
Tyrosine	Y
Valine	Z

Dizi hizalama [sequence alignment] biyoinformatiğin en temel problemlerinden biridir ve bu konu üzerine oldukça fazla çalışılmıştır. Dizilim hizalamanın temel yaklaşımı, farklı DNA, RNA veya protein dizilimlerinin [sequence] birbirine en çok benzeyen bölgelerinin saptanmasıdır. Bu şekilde biyolojik bir fonksiyonu olabilecek bölgeleri tespit edilebilir veya bir deney sonucunda elde ettiğimiz

DNA veya protein diziliminin hangi gen ve organizmaya ait olduğunu büyük ölçüde saptayabiliriz. Örnek bir hizalama aşağıdaki gibi olabilir:

GCACATATGGAAACC

||||||---|||||*

GCACAT---GAAACT

Yukarıda GCACATATGGAAACC dizilimi ile GCACATGAAACT diziliminin hizalanmış şeklini görüyorsunuz. Bu durumda ikinci dizilimin ortasında 3 bazlık bir bölümün silindiğini [deletion] ve sonundaki bazın da değişime uğradığını [mutation] söyleyebiliriz. Hizalama konusunu iki farklı çatı altında düşünebiliriz:

Bunlardan ilkini, hakkında literatürde neredeyse hiç bir bilgimizin olmadığı bir canlı türünün çalışılması olarak düşünebiliriz. Bu türe ilişkin genomun tamamını ya da bir kısmını bilinen diğer organizmalarla kıyaslayarak hangi cinse ait olduğunu tahmin edebiliriz ya da fonksiyonunu bilmediğimiz bir gen ile fonksiyonu bilinen diğer genlerdeki ortak motifleri bularak o genin işleyişi ve amacına ilişkin bir çıkarımda bulunabiliriz.

Diğerini ise genomu zaten bilinen bir organizmayla yaptığımız çalışmalar olarak düşünebiliriz. Bu tarz çalışmalarda dizilim hizalama kullanarak elimizdeki DNA veya protein dizilim parçalarının hangi gene ait olduğunu hızlı bir şekilde tespit edebiliriz veya yine referans genom bilgisini kullanarak nerelerde ne tarz farklılıklar (değişim (/mutasyon [deletion]), silinme (/delesyon [deletion]) vb.) olduğunu saptayabilir ve böylelikle örneğin bir hastalığa neden olan değişimleri ortaya koyabiliriz.

DNA ve RNA söz konusu olduğunda A bazının karşısına gelmesi gereken bazı hepimiz biliyoruz. Peki, kalan G veya C bazlarından biri gelecek olsa, A bazı

hangisini istemeyerek de olsa tercih eder? Bir pirimidin olan C bazının G'den daha fazla tercih edilebileceğini söyleyebiliriz ve bu doğrultuda hayali skorlama tablosu hazırlayabiliriz. Bu tabloda, eğer bir bazın karşısına aynı baz gelirse bu bir bazlık hizalanmayı +1 ile ödüllendirebilir, pürinin karşısına pürin veya pirimidinin karşısına pirimidin geliyorsa bu hizalanmayı -1 ile cezalandırabilir, pürinin karşısına pirimidin veya tam tersi geliyorsa da bunu -2 gibi daha etkili bir puanla cezalandırabiliriz. Konuyu daha iyi anlamak adına ATGTCC ile ATC dizilimlerini hizalayalım (her iki dizilimde de aralarda boşluklara izin verilmediğini varsayalım):

ATGTCC

||*---

ATC--

Yukarıdaki gibi hizalamanın toplam puanı 0 olacaktır: iki bazın (A ve T) tam hizalanması (1+1 puan) ve bir pirimidinin (C) pürin ile hizalanması (G) (-2 puan) sonucu 1+1-2 = 0 puan. Peki, bu iki dizilimi aşağıdaki şekilde hizalarsak?

ATGTCC

--*||-

--ATC-

Bu durumda hizalamaya ait toplam puan 1 olacaktır: iki bazın (T ve C) tam hizalanması (1+1 puan) ve bir pürinin (A) yine başka bir pürinle hizalanması (G) (-1 puan) sonucu 1+1-1 = 1 puan. Toplam hizalama puanlarını göz önüne aldığımızda, ikinci seçeneğin daha çok tercih edilir olduğunu söyleyebiliriz. Burada temel olan yaklaşım, puanlama sistemini nasıl belirlediğiniz ve sonrasında da kullandığınız algoritmadır. Algoritma kısmına şimdilik girmeyeceğiz, ancak günümüzde en sık kullanılan algoritmanın BLAST olduğunu

belirtmeliyim. DNA veya RNA için bu hesaplamayı yapmak kolay sayılır, peki ya söz konusu proteinler olduğunda nasıl bir yol izlemek gerekiyor? Konunun fazla detayına girmeden bir örnek olarak BLOSUM62 matrisini anlatarak bu soruya cevap vereceğim.

Tamamen belirleyici olmamakla birlikte, birbirine yakın iki türün protein dizilimlerinin hizalanmasında BLOSUM80 (%80 benzerlik tablosu), birbirine uzak iki türün protein dizilimlerinin hizalanmasında ise genel olarak BLOSUM45 (%45 benzerlik tablosu) tercih ediliyor. Yakınlığın veya uzaklığın tam olarak kestirilemediği durumlarda ise BLOSUM62 kullanılıyor ve bu benzerlik tablosundaki değerlerin gayet kullanışlı olduğu söyleniyor. BLOSUM62, BLAST algoritmasının kullandığı bir aminoasit benzerlik tablosu.

Belirli bir yüzdenin üzerinde benzerlik gösteren (%62) gerçek protein dizilimlerinin hizalanması sonucu oluşturulan bu tabloda puanların hesaplanmasında temel iki faktör rol oynuyor. Bunlardan ilki, bir aminoasidin karşısına diğer bir aminoasidin ne kadar sıklıkla geldiği. Böylece, bir aminoasidin farklı bir protein diziliminde diğer bir aminoaside dönüşmesi eğer fonksiyon üzerinde çok büyük bir etkiye sahip değilse ilgili değişimi sık olarak görmeyi bekleriz. Çok yüzeysel bir örnek verelim: hidrofobik bir aminoasidin başka bir hidrofobik aminoasitle değişmesi protein fonksiyonunu etkilemeyebilir ancak bu aminoasidin hidrofilik bir aminoasitle değişimi proteinin 3 boyutlu yapısını değiştirebilir ve bu proteinin fonksiyonunu tamamen değiştirebilir. Bu durumda, daha az etkisi olan değişimi küçük bir puanla, daha çok etkisi olan değişimi ise daha büyük bir puanla cezalandırmalıyız. BLOSUM tablosundaki puanların hesaplanmasındaki diğer etken ise, ilgili aminoasitlere tüm proteomda ne sıklıkla rastlandığı. BLOSUM denklemine göre bir aminoaside ne kadar seyrek rastlanırsa, o aminoasidin önemi de o kadar artar. Bu nedenle, proteomdaki en seyrek aminoasit olan Tryptophane (Trp, W) aminoasidiyle yapılan bir hizada karşısına yine aynı aminoasit gelirse, bu dizilim BLOSUM tablosuna göre en yüksek puan olan 11 ile ödüllendirilir. BLOSUM62 tablosunu aşağıda bulabilirsiniz:

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0	-2	-1
R	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3	-1	0
N	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3	3	0
D	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3	4	1
C	0	-3	-3	-3	9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2	-1	-3	-3
Q	-1	1	0	0	-3	5	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2	0	3
E	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2	1	4
G	0	-2	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-2	-3	-3	-1	-2
H	-2	0	1	-1	-3	0	0	-2	8	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3	0	0
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-3	-1	3	-3	-3
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	-2	2	0	-3	-2	-1	-2	-1	1	-4	-3
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2	0	1
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1	-3	-1
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6	-4	-2	-2	1	3	-1	-3	-3
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7	-1	-1	-4	-3	-2	-2	-1
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4	1	-3	-2	-2	0	0
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-2	-2	0	-1	-1
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11	2	-3	-4	-3
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1	-3	-2
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4	-3	-2
B	-2	-1	3	4	-3	0	1	-1	0	-3	-4	0	-3	-3	-2	0	-1	-4	-3	-3	4	1
Z	-1	0	0	1	-3	3	4	-2	0	-3	-3	1	-1	-3	-1	0	-1	-3	-2	-2	1	4

Hizalama yöntemleri

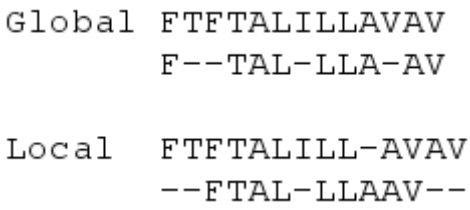
Çok kısa veya çok benzer diziler elle hizalanabilir. Ancak, çoğu ilginç problem, insan eliyle yapılamıyacak kadar uzun, karmaşık veya çok sayıda dizinin hizalanmasını gerektirir. Böyle durumlarda, kaliteli dizi hizalamaları elde edebilmek için insan bilgisine dayanan algoritmalar kullanılır. Ender olarak bu algoritmalardan elde edilen sonuçlar, algoritmik olarak ifadesi zor olan durumlar için elle düzeltilebilir.

Dizi hizalaması için kullanılan hesaplamalı yöntemler genelde iki gruba ayrılır: *global optimizasyon* ve *yerel optimizasyon*. Global hizalamanın bulunması bir global optimizasyon çeşididir ve elde edilecek hizalamanın, sorgulanan dizilerin tamamını kapsamaya "zorlar". Buna karşın, yerel hizalamalar genelde birbirinden çok farklılık gösteren uzun dizilerde benzer bölgeleri tespit eder. Çoğu zaman yerel hizalamalar tercih edilir ama bunların bulunması daha zor olabilir.

Dizi hizalama problemi için çeşitli algoritmalar uygulanmıştır, bunların bazıları, dinamik programlama gibi, yavaş ama formel olarak eniyileyici yöntemlerdir, bazıları ise hızlı ama mutlaka mükemmel sonucu vermeyebilen,

veri tabanı aramaları için tasarlanmış buluşsal algoritmalar veya olasılıksal yöntemlerdir.

Global ve lokal hizalamalar



Global ve lokal hizalama arasındaki farkın bir örneği. Eğer diziler yeterince benzer değilseler global hizalamalar 'boşluklu' olur.

Global hizalamalarda her dizideki her harfin hizalanması amaçlanır. Sorgu kümesindeki diziler birbirine benzer ve yaklaşık aynı uzunlukta olursa global hizalamaları en yararlı olur. (Ama bu, global hizalamaların boşluklarla sonlanamayacağı anlamına gelmez.) Global hizalama tekniklerinden biri, dinamik programlamaya dayalı olan Needleman-Wunsch algoritmasıdır.

Birbirine benzemeyen ama benzer bölgeler içerdiği tahmin edilen diziler için lokal hizalamalar daha yararlıdır. Keza, benzer kısa dizi motiflerinin tespitinde lokal hizalamalar kullanılır. Smith-Waterman algoritması da dinamik programlamaya dayalı bir lokal hizalama yöntemidir. Eğer diziler yeterince birbirine benziyorsa lokal ve global hizalama sonuçları arasında bir fark olmaz.

Hibrit yöntemler (yarı global veya "glokal" yöntemler olarak da adlandırılabilir) bir veya öbür dizinin başı ve sonunu da kapsayan en iyi hizalamayı bulmaya çalışır. Dizilerden birinin sonu, öbürünün başı ile örtüşüyorsa bu özellikle yararlı olabilir. Bu durumda ne global ne de lokal hizalama tamamen uygundur: global yöntem hizalamayı örtüşme bölgesinin dışına uzatmaya çalışacaktır, lokal yöntem ise örtüşme bölgesini yeterince kapsamayabilir.

Dinamik programlama

Dinamik programlama tekniği, Needleman-Wunsch algoritması ile global hizalamalar üretmek için, Smith-Waterman algoritması ile de lokal hizalamalar üretmek için uygulanabilir. Tipik kullanımda, protein hizalamalarında amino asit uyuşma veya uyuşmamalarına bir skor verebilmek için bir substitusyon matrisi; bir dizideki amino asitin öbür dizide bir boşlukla eşleştirilmesi için de bir boşluk ceza değeri kullanılır. DNA ve RNA hizalamaları bir skor matrisi kullanabilir ama pratikte basitçe pozitif bir uyuşma skoru, negatif bir uyuşmama skoru ve negatif bir boşluk cezası verilir. (Standart dinamik programlamada, her amino asitin skoru komşularının kimliğinden bağımsızdır, dolayısıyla baz istiflemesi etkileri hesaba katılmaz. Ancak, algoritmayı değiştirip bu tür etkileri de göz önüne almak mümkündür.)

İlerlemeci yöntemler

İlerlemeci, hiyerarşik veya ağaç yöntemleri, birbirine en benzer dizileri hizalamakla başlar, sonra gittikçe daha az benzeyen dizilerin hizalamaya eklenmesi ile sonunda tüm sorgu kümesi sonuca dahil edilir. Dizilerin yakınlığını betimleyen ağaç yapısı ikili kıyaslamalara dayanır, bunlar FASTA gibi heuristik ikili hizalama yöntemleri kullanır. İlerlemeci hizalama sonuçları "en benzer" dizilerin seçimine bağımlıdır, bu yüzden ilk yapılan ikili hizalamadaki hatalara duyarlıdır. Çoğu ilerlemeci, çoklu dizi hizalama yöntemi buna ek olarak, sorgu kümesindeki diziler arasındaki yakınlık derecesine göre onlara ağırlık verir, böylece ilk dizilerin kötü seçilmesi olasılığı azalır ve en son hizalamanın doğruluğu iyileşir.

Clustal ilerlemeci uygulamalarının çoğu varyasyonu çoklu dizi hizalaması, filogenetik ağaç inşası ve protein yapı hesaplamasına girdi hazırlamakta kullanılır. İlerlemeci yöntemin daha yavaş ama daha doğruluklu bir varyantı T-Coffee olarak adlandırılır. Bu algoritmaların uygulamaları ClustalW ve T-Coffee'de bulunabilir.

Tekrarlayıcı yöntemler

İlerlemeci yöntemlerin zayıf bir yönü, ilk ikili hizalamanın doğru olmasına olan büyük bağımlılıktır. Tekrarlayıcı yöntemler, bunu iyileştirmeye çalışırlar. Tekrarlayıcı yöntemler seçilmiş bir skorlama fonksiyonuna dayanan objektif fonksiyonu optimize ederler, ilk global hizalamayı oluşturup sonra dizi altkümelerini yeniden hizalayarak. Yeniden hizalanan altkümelerin kendileri de hizalanarak çoklu dizi hizalamasının bir sonraki yinelemesini oluştururlar. Dizi altkümelerini ve objektif fonksiyonu seçmek için çeşitli yollar mevcuttur.

Skor fonksiyonları

Bilinen diziler hakkında biyolojik veya istatistik gözlemleri yansıtan bir skor fonksiyonunun seçimi, iyi hizalamalar elde edilmesinde çok önemlidir. Protein dizileri genelde substitusyon matrisleri kullanılarak yapılır, bu matrisler belli karakter substitusyonlarının olma olasılıklarını yansıtır.

PAM matrisi (*Point accepted mutation*, noktasal olarak kabul edilmiş mutasyon) olarak adlandırılan bir grup matris, belli amino asit mutasyonlarının olma hızları ve olasılıklarını içerir (bu matrisler Margaret Dayhoff tarafından tanımlanmış olduğu için bazen "Dayhoff matrisleri" olarak da adlandırılır).

Sık kullanılan başka bir grup matris ise BLOSUM (*Blocks Substitution Matrix* blok substitusyon matrisi) olarak adlandırılır, bunlar empirik olarak gözlemlenmiş substitusyon olasılıklarını kodlar. Her iki tip matrislerin varyantları, farklı düzeylerde ıraksama göstermiş dizilerin tayininde kullanılır. Böylece, BLAST veya FASTA kullanıcıları aramalarını sadece yakın ilişkili uyuşmalara kendilerini sınırlayabilir, veya arzu ederlerse daha uzak ilişkili dizileri de bulabilmeleri mümkün olur. Boşluk cezaları hizalamaya bir boşluk katılmasına etki eder. Bu boşluk ceza skoru, evrimsel anlamda bir insersiyon veya delesyon mutasyonunun olma hızı ile orantılı olmalıdır. Elde edilen hizalamaların kalitesi skor fonksiyonunun kalitesine bağlıdır.

Aynı hizalamayı farklı skor matrisleri ve/veya boşluk cezaları ile tekrarlayıp sonuçları karşılaştırmak çok yararlı olabilir. Hizalama parametrelerindeki varyasyonlara dayanıklılık gösteren hizalama bölgeleri tespit edilerek, sonucun zayıf olduğu veya tek olmadığı bölgeler belirlenebilir.