Iğdır Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 14(2), 544-562, 2024
Journal of the Institute of Science and Technology, 14(2), 544-562, 2024

ISSN: 2146-0574, eISSN: 2536-4618

Bilgisayar Mühendisliği / *Computer Engineering* **DOI: 10.21597/jist.1404898**

Araştırma Makalesi / *Research Article*

Geliş tarihi / *Received*: 14.12.2023 **Kabul tarihi / *Accepted*: 04.04.2024**

Atıf İçin: Erdirik, H., Karcıoğlu, A. A., Tanyolaç, M. B. ve Bulut, H. (2024). Meta-Sezgisel Tabanlı Clustal-SA Algoritmasını Kullanarak DNA Sekanslarında Çoklu Dizi Hizalama. *Iğdır Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 14(2), 544-562.

To Cite: Erdirik, H., Karcıoğlu, A. A., Tanyolaç, M. B. & Bulut, H. (2024). Multiple Sequence Alignment for DNA Sequences by Meta-Heuristic Based Clustal-SA Algorithm. *Journal of the Institute of Science and Technology*, 14(2), 544-562.

Meta-Sezgisel Tabanlı Clustal-SA Algoritmasını Kullanarak DNA Sekanslarında Çoklu Dizi Hizalama

Hatice ERDİRİK[1*], Abdullah Ammar KARCIOĞLU[2], Muhammed Bahattin TANYOLAÇ[3], Hasan BULUT[4]

Öne Çıkanlar:
- Çalışmada çoklu dizi hizalaması için hem sezgisel yaklaşım hem de aşamalı hizalama kullanılmıştır
- Başarı karşılaştırması için hizalama skoru ve eşleşen sütun sayısı baz alınmıştır
- Aynı uzunluğa sahip dizi hizalamaları üzerinde önerilen Clustal-SA algoritması ClustalW ye göre daha iyi hizalama skoru vermiştir

Anahtar Kelimeler:
- Çoklu dizi hizalama
- Sezgisel yaklaşım
- Clustal algoritması
- Benzetimli tavlama
- Biyoinformatik

ÖZET:

Biyoinformatik, biyolojik verilerin analizi ve kalıtsal ilişkilerin ortaya çıkarılması için matematik, biyoloji ve bilgisayar bilimlerini birleştiren bir disiplindir. Bu alandaki en kritik görevlerden biri, biyolojik dizilerin hizalanmasıyla ilgili olan dizi hizalama problemini çözmektir. Ancak, biyolojik verilerin hızla artması, bu problemi manuel olarak çözülemez hale getirmiş ve bilgisayar sistemlerinin biyoinformatikte daha yaygın bir şekilde kullanılmasına yol açmıştır. Bu çalışmada, mevcut Clustal algoritması ve benzetimli tavlama algoritması kullanılarak yeni bir dizi hizalama algoritması önerilmiştir. Clustal algoritmasının hız avantajını kullanarak ve benzetimli tavlama algoritmasını entegre ederek, Clustal'ın aç gözlü yaklaşımından uzaklaşılarak optimal hizalama skoru elde etmek amaçlanmıştır. Geliştirilen algoritmanın başarısını değerlendirmek için SP (Çiftlerin Toplamı) puanlama sistemi kullanılmış ve hizalama sonucunda sütun eşleşme sayısı dikkate alınmıştır. Elde edilen sonuçlar, geliştirilen algoritmanın aynı uzunluktaki dizi veri kümeleri üzerinde ClustalW programından daha iyi performans gösterdiğini, MUSCLE programına göre ise bazı veri setlerinde daha başarılı olduğu veya yakın sonuçlar verdiğini ortaya koymuştur. Bu gelişme, biyoinformatik alanında dizi hizalama problemini çözmek için yeni ve daha etkili bir yaklaşımın potansiyelini vurgulamaktadır. Gelecekte, bu tür geliştirmelerin biyolojik veri analizi alanında daha geniş bir uygulama alanı bulabileceği düşünülmektedir.

Multiple Sequence Alignment for DNA Sequences by Using Meta-Heuristic Based Clustal-SA Algorithm

Highlights:
- Both heuristics and progressive alignment are used for multiple sequence alignment
- Success comparison is based on the alignment score and the number of matched columns
- On sequence alignments of the same length, the proposed Clustal-SA algorithm gave better alignment scores than ClustalW

Keywords:
- Multiple Sequence Alignment
- Heuristic Approach
- Clustal Algorithm
- Simulated Annealing
- Bioinformatics

ABSTRACT:

Bioinformatics is a discipline that combines mathematics, biology and computer science to analyze biological data and reveal genetic relationships. One of the most critical tasks in this field is to solve the sequence alignment problem, which is related to the alignment of biological sequences. However, the rapid increase in biological data has made this problem unsolvable manually and led to the more widespread use of computer systems in bioinformatics. In this study, a new sequence alignment algorithm is proposed using the existing Clustal algorithm and simulated annealing algorithm. By using the speed advantage of the Clustal algorithm and integrating the simulated annealing algorithm, it is aimed to obtain an optimal alignment score by moving away from the greedy approach of Clustal. To evaluate the success of the developed algorithm, the SP (Sum of Pairs) scoring system was used and the number of column matches as a result of the alignment was taken into account. The results obtained revealed that the developed algorithm performed better than the ClustalW program on sequence data sets of the same length, and was more successful or gave similar results on some data sets compared to the MUSCLE program. This development highlights the potential of a new and more effective approach to solving the sequence alignment problem in bioinformatics. It is thought that in the future, such developments may find wider application in the field of biological data analysis.

[1]Hatice ERDİRİK (**Orcid ID**: 0000-0002-5816-4804), Yıldız Teknik Üniversitesi, Elektrik-Elektronik Fakültesi, Bilgisayar Mühendisliği Bölümü, İstanbul, Türkiye

[2]Abdullah Ammar KARCIOĞLU (**Orcid ID:** 0000-0002-0907-751X), Atatürk Üniversitesi, Yazılım Mühendisliği Bölümü, Erzurum, Türkiye

[3]Muhammed Bahattin TANYOLAÇ (**Orcid ID:** 0000-0002-4368-0988), Ege Üniversitesi, Biyomühendislik Bölümü, İzmir, Türkiye

[4]Hasan BULUT (**Orcid ID:** 0000-0002-4872-5698), Ege Üniversitesi, Bilgisayar Mühendisliği Bölümü, İzmir, Türkiye

***Sorumlu Yazar/*Corresponding Author**: Hatice Erdirik, e-mail: herdirik@yildiz.edu.tr

GİRİŞ

Biyoloji bilimler arasında kilit bir rol oynamaktadır. Biyologlar tarafından toplanan büyük miktarda veri ve bu verilerin incelenip, yorumlanma ihtiyacı biyoinformatik biliminin ortaya çıkmasına yol açmıştır (Cohen,2004). Biyoinformatik, biyolojideki verileri kavramsallaştırmak ve veriler arasındaki ilişkileri anlamak için matematik, bilgisayar ve istatistik gibi bilim dallarından yararlanan bir bilim dalıdır (Luscombe ve ark., 2001). Biyoinformatikte birçok uygulama ve görev bulunmaktadır.

Biyolojik dizilerin incelenmesi ve yorumlanmasındaki en yaygın işlemlerden biri dizi hizalama işlemidir. Diziler arası benzer bölgelerin çıkarılması, evrimsel süreç boyunca korunan bölgelerin tespiti, genetik hastalıklarda değişen bölgelerin tespiti gibi birçok araştırmada, genetik dizilerin hizalama işlemi kullanılmaktadır. Bu nedenle dizi hizalama biyoinformatik alanında önemli bir araştırma alanıdır (Botta ve Negro, 2010).

Dizi hizalama işlemi ikili ve çoklu dizi hizalama (Multiple Sequence Alignment-MSA) olarak ikiye ayrılmaktadır. İkili dizi hizalama iki adet dizinin hizalaması işlemidir. MSA, üç veya daha fazla dizinin hizalanması işlemidir (Haque ve ark., 2009). MSA işlemi protein yapısı işlev tahmini, filogen çıkarımı ve dizi analizi için kullanılmaktadır. MSA çözümü NP-complate bir problemdir. Bu nedenle çözümü üstel bir karmaşıklıkla sonuçlanmaktadır (Lee ve ark., 2008).

MSA problemin çözümü için dinamik programlama, meta-sezgisel metotlar, sekanslarla hizalamalar, istatiksel ve olasılıksal yöntemler kullanılmaktadır. Dinamik programlama, dizilerin matematiksel olarak en uygun hizalanmasını garanti etmektedir. Fakat fazla bellek kullanımı gerektirdiğinden dolayı çoklu dizi hizalama probleminin çözümü için tercih edilmezler. Bu nedenle çözüm için sezgisel tabanlı algoritmalar tercih edilir. Meta-sezgisel algoritmaların çoğu global ve yerel hizalama yöntemlerini birleştirerek aşamalı bir hizalama yaklaşımı sunar. Aşamalı yöntemlerin dezavantajı, erken bir adımda meydana gelen bir hatanın sonraki aşamalarda düzeltilememesidir. Aşamalı ve global hizalamayı birleştiren ilk MSA uygulaması ClustalW'dir. ClustalW'den sonra birçok aşamalı yaklaşımı baz alan program ve algoritma geliştirilmiştir (Pais ve ark. 2014).

Meta-sezgisel algoritmalar tek çözüm tabanlı meta-sezgisel algoritmalar ve popülasyon tabanlı meta-sezgisel algoritmalar diye sınıflandırılır. Tek çözümlüler; Tepe Tırmanışı, Tavlama Benzetimi, Tabu Araması gibi tek bir aday çözümü geliştirmeye odaklanırlar. Meta-sezgisel algoritmalar genellikle deterministtik (rastgelelik içermeyen) algoritmalara göre daha yavaştır. Bu yüzden bazı araştırma alanlarında meta-sezgisel algoritmanın çalışma süresini hızlandırmak için girdi verisini ön işlemeden geçirmek popüler hale gelmiştir. Meta-sezgisel algoritmalardaki son çalışmalar genellikle 3 temel amaca odaklanmıştır ve bunlar; uygunluk fonksiyonunu değiştirmek, operatörleri değiştirmek, melez meta-sezgisel kullanmaktır (Aktan ve Bulut, 2022).

Bu çalışmanın ana hedefi, Clustal algoritması ve Simulated Annealing (SA) algoritmasını birleştirerek yeni bir Clustal-SA algoritması önermek ve bu algoritmanın performansını değerlendirmektir. Önerilen Clustal-SA algoritması, Clustal algoritmasının ürettiği hizalamayı girdi olarak alır ve SA algoritmasının her iterasyonunda hizalama üzerinde mutasyon işlemi uygular. Bu mutasyon işlemi, algoritmanın nasıl gerçekleştirildiği detaylı bir şekilde açıklanmıştır.

Ayrıca, önerilen Clustal-SA algoritması, ClustalW programıyla karşılaştırılarak değerlendirilmiştir. Karşılaştırma, hizalama skoru, eşleşen sütun sayısı ve çalışma süresi gibi metrikler üzerinden yapılmıştır. Bu değerlendirme, Clustal-SA algoritmasının ClustalW'ye kıyasla performansını ve etkinliğini belirlemeye yöneliktir.

Sonuç olarak, bu çalışma Clustal ve Simulated Annealing algoritmalarını birleştirerek yeni bir hizalama yöntemi önermektedir. Önerilen yöntem, ClustalW programıyla karşılaştırılarak değerlendirilmiş ve performansı belirlenmiştir. Bu çalışma, biyoinformatik alanında hizalama problemini çözmek için yeni bir yaklaşım sunmaktadır.

İlerleyen bölümlerde, DNA, RNA ve proteinin yapıları, dizi hizalama, ikili dizi hizalama, MSA probleminin çözümü için geliştirilen çalışmalardan bahsedilmiştir. Clustal, SA algoritması ve çalışma kapsamında önerilen Clustal-SA algoritması açıklanmıştır. MSA probleminin çözümü için önerilen algoritmanın doğruluğunun ve çalışma süresinin değerlendirilmesi için hangi metrik ve fonksiyonların kullanıldığı, hizalama işlemi için kullanılan veri setlerinin özellikleri (dizi uzunlukları, dizilerin adedi) açıklanmıştır. Veri setleri üzerinde uygulanması sonucu elde edilen hizalamaların skorları ve çalışma sureleri karşılaştırmalı olarak gösterilmiştir. Önerilen algoritmanın ve yöntemlerin başarısı değerlendirilmiş ve literatüre olan katkıları belirtilmiştir.

MATERYAL VE METOT

Bu bölümde MSA probleminin tanımı ve temel özelliklerine, gerçekleştirilen hizalama işleminin performansının nasıl değerlendirildiğine ve MSA probleminin çözümü için geliştirilmiş yaklaşımlara değinilmiştir.

Dizi Hizalama Problemi

Dizi hizalama, DNA, RNA ve protein dizilerinin düzenlenmesine ve incelenen diziler arasındaki benzer bölgelerin bulunmasına yardımcı olmaktadır. Dizi hizalama probleminde her bir dizi $S = s_1, \ldots . s_n$ olarak temsil edilir. Boşluk(indel) (‘-’ sembolüyle ifade edilen), bir dizinin bazı bölümlerine ekleme veya silme işlemi yapıldığını gösterir. Nükleotid bazları Adenin (A), Sitozin (C), Guanin (G), Timin (T) ve Urasildir (U). Alfabe sırasıyla DNA ve RNA için A, C, G, T ve A, C, G, U şeklindedir. Bir proteinin birincil yapısı doğrusal bir amino asit zinciridir. A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y ve V ile gösterilen yirmi amino asit vardır (Omar ve ark., 2004). Şekil 1’ de “ATTGT” ve “AGGACAT” dizilerinin ikili hizalaması gösterilmektedir.

```
D1 = A T T G - - - T
D2 = A - G G A C A T
```

Şekil 1. İkili hizalama örneği

Dizi hizalama problemi yerel hizalama ve global hizalama diye ikiye ayrılır. Global hizalamada diziler bir bütün olarak hizalanırken, yerel dizi hizalamasında benzerlikler tespit edilir (Bucak ve Uslan, 2011). Smith-Waterman algoritması yerel hizalama algoritmasının örneğidir. Global dizi hizalama ise, her iki dizinin kendi bütünlüğü içerisinde en iyi hizalamayı bulmak için kullanılmaktadır. Needleman-Wunch algoritması global hizalama algoritmasına bir örnektir. Yerel ve global hizalama örneği şekil 2’de hedef ve sorgu dizi üzerinde gösterilmektedir. İkili hizalama probleminin çözümünde genellikle dinamik programlama tercih edilmektedir (Major Differences,2022).

Şekil 2. Yerel ve global hizalama örneği (Major Differences,2022)

MSA problemi

Biyolojik makro molekül diziliminin belirlenmesindeki büyük artışı nedeniyle, üç veya daha fazla nükleik asit ve amino asit diziliminin aynı anda hizalanması yaygın bir gereklilik haline gelmiştir. Basitçe hizalama, benzerliği üst düzeye çıkarmak için boşlukların eklenmesi ile oluşur. Çoklu dizi hizalamaları, dizi ailesinin üyeleri arasında benzerliği göstermeye yaramaktadır. MSA korunmuş bölgelerin vurgulanması amacına da sahiptir. Ayrıca çoklu dizi hizalama proteinlerin ikincil yapılarının tahmininde, aile üyeleri arasındaki evrimsel ilişkileri çıkarmak için kullanılmaktadır.

MSA probleminde amaç, diziler arasındaki eşlesen sembollerin sayısını en üst düzeye çıkarmak ve ayrıca boşluklara izin veriliyorsa yalnızca minimum boşluk eklemeyi kullanmaktır (Likic,2008). MSA, kombinatoryal problemler olarak adlandırılan, üstel zaman karmaşıklığına sahip bir optimizasyon problemi sınıfına aittir. Zaman karmaşıklığı $O(L^N)$'dir. L Hizalanacak dizilerin ortalama uzunluğudur ve N ise hizalanacak dizilerin sayısıdır. Biyolojide, diziler yüzlerce (protein), binlerce (RNA) veya milyonlarca (DNA) uzunluklara sahip olduğundan hizalama işleminde çalışma zamanı uzundur. Bu yüzden, MSA problemi için genellikle dinamik programlama ve sezgisel metotlar kullanılmaktadır (Likic,2008).

Hizalanmış dizilerin skorlarının hesaplanması

Uygulanan dizi hizalama algoritmaları sonucu elde edilen hizalamalar arasında optimal hizalamayı diğer hizalamalardan ayırt etmek için her bir hizalamanın maliyeti tanımlanmaktadır. Bir çoklu hizalamanın maliyetini hesaplamak için amaç fonksiyonu kullanılmaktadır. Amaç fonksiyonları sayesinde çoklu dizi hizalamasının maliyeti hesaplanmaktadır. MSA'ların maliyetini hesaplamak için farklı amaç fonksiyonları kullanılmaktadır. Yaygın olarak kullanılan amaç fonksiyonu aşağıda açıklanmaktadır.

Çiftlerin Toplamı (Sum of Pairs -SP): SP en popüler amaç fonksiyonlarındandır. MSA' da SP puanı, N adet dizi bulunan algoritmada $\binom{n}{2}$ ikili hizalama bulunmaktadır. SP skoru sütun tabanlıdır. N hizalanmış dizi sayısı olarak kabul edildiğinde j. sütun için SP puanının hesaplanması Formül 1' de gösterildiği gibidir. Hizalamanın uzunluğu L olarak kabul edildiğinde hizalama (Alignment-A) için SP skoru Formül 2'de gösterildiği gibi hesaplanmaktadır.

$$SP(j) = \sum_{i=0}^{N-1} \sum_{K=i+1}^{N} PS(C_{ij}, C_{ik}) \quad (1)$$

$$SP(A) = \sum_{i=1}^{L} \sum_{j=0}^{N-1} \sum_{K=j+1}^{N} PS(C_{ij}, C_{ik}) \quad (2)$$

PS fonksiyonu mevcut hizalamada her bir sütun için ikili hizalama yapmaktadır. PS fonksiyonundaki C_{ij} ve C_{ik} parametreleri $j.$ ve $k.$ dizilerin, $i.$ sütunda bulunan değerlerini temsil etmektedir. PS fonksiyonun C_{ij} ve C_{ik} parametreleri ile ilgili üreteceği değerlerin belirli bir kuralı yoktur. Bu kurallar her bir programın amacına uygun olarak farklı şekillerde belirlenebilir. Bu parametrelerin alabileceği durumlar ve bu durumların değerleri Formül 3'te gösterildiği gibi hesaplanmaktadır.

$$SP(C_{ij}, C_{ik}) = \begin{cases} x = 1, & C_{ik} = C_{ij}, C_{ik} \neq " - ", C_{ij} \neq " - " \\ x = 0, & C_{ik} = C_{ij}, C_{ik} = " - ", C_{ij} = " - " \\ x = -1, & C_{ik} \neq C_{ij}, C_{ik} \neq " - ", C_{ij} \neq " - " \end{cases} \quad (3)$$

SP fonksiyonunun C_{ij} ve C_{ik} parametreleri için üreteceği değerler, akrabalıkları ve fonksiyonel özellikleri bilinen diziler kullanılarak yapılan istatiksel çalışmalar sonucunda üretilen Yerine Koyma (Substitution) Matrisleri kullanılarak belirlenmektedir. Bu matrisler PAM ve BLOSUM matrisleridir. Aşağıda DNA çoklu dizi hizalamasının skorlama işleminin nasıl gerçekleştirildiğinin örneği verilmiştir. Örneğin;

D1 = A C C C G A — D1 = ACCCGA
D2 = A C T A — Diziler hizalandıktan sonra → — D2 = AC--TA
D3 = T C C T A — D3 = TC-CTA

Örnek hizalamanın skoru aşağıdaki gibi hesaplanır;

$$SP(A) = [SP(A,A) + SP(A,T) + SP(A,T)] + [SP(C,C) + SP(C,C) + SP(C,C)] + [SP(C,-) + SP(C,-) + SP(C,-) + SP(-,-)] + [SP(C,-) + SP(C,C) + SP(-,C)] + [SP(G,T) + SP(G,T) + SP(T,T) + [SP(A,A) + SP(A,A) + SP(A,A)]$$

Örnek uygulamada boşluk, yanlış hizalama ve uyum puanlarına göre bir hizalama skoru belirlenmektedir. Bu skorlama değerleri için BLOSUM ve PAM matrisleri kullanılmaktadır. Gerçekleştirilen çalışmada skorlama işlemi için BLOSUM matrisi kullanılmıştır. BLOSUM matrisinde yer alan harfler aminoasitleri temsil etmektedir. Satır ve sütunda yer alan harflerin kesiminde yer alan sayısal değerler, hizalama işleminde iki amino asidin hizalanmasının kaç puan olduğunu(değerini) göstermektedir. SP skoru bu değerlerin kullanılması sonucu elde edilmektedir. BLOSUM matrisi şekil 3'te gösterilmektedir.

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	4																			
R	−1	5																		
N	−2	0	6																	
D	−2	−2	1	6																
C	0	−3	−3	−3	9															
Q	−1	1	0	0	−3	5														
E	−1	0	0	2	−4	2	5													
G	0	−2	0	−1	−3	−2	−2	6												
H	−2	0	1	−1	−3	0	0	−2	8											
I	−1	−3	−3	−3	−1	−3	−3	−4	−3	4										
L	−1	−2	−3	−4	−1	−2	−3	−4	−3	2	4									
K	−1	2	0	−1	−3	1	1	−2	−1	−3	−2	5								
M	−1	−1	−2	−3	−1	0	−2	−3	−2	1	2	−1	5							
F	−2	−3	−3	−3	−2	−3	−3	−3	−1	0	0	−3	0	6						
P	−1	−2	−2	−1	−3	−1	−1	−2	−2	−3	−3	−1	−2	−4	7					
S	1	−1	1	0	−1	0	0	0	−1	−2	−2	0	−1	−2	−1	4				
T	0	−1	0	−1	−1	−1	−1	−2	−2	−1	−1	−1	−1	−2	−1	1	5			
W	−3	−3	−4	−4	−2	−2	−3	−2	−2	−3	−2	−3	−1	1	−4	−3	−2	11		
Y	y2	−2	−2	−3	−2	−1	−2	−3	2	−1	−1	−2	−1	3	−3	−2	−2	2	7	
V	0	−3	−3	−3	−1	−2	−2	−3	−3	3	1	−2	1	−1	−2	−2	0	−3	−1	4

Şekil 3. BLOSUM skorlama matrisi (Doğan ve Otu, 2014)

Eşleşen karakterlerin bulunması

Dizi hizalama algoritmalarının uygulanmasıyla elde edilen hizalamada, eşleşen karakter sayısının fazla olması, hizalamanın başarısını gösteren önemli bir metriktir. Bu başarıyı değerlendirmek için elde edilen hizalamadaki eşleşen karakter sayısı, sütunlar gezilerek hesaplanır. Bu, homolog bölgelerin doğru bir şekilde hizalandığını ve biyolojik benzerliklerin korunduğunu gösteren bir ölçüdür.

Çalışma kapsamında öne çıkan Clustal-SA algoritması da başarısını test etmek amacıyla, kullanılan test veri setleri üzerinde elde ettiği hizalamalardaki eşleşen sütun sayısına göre değerlendirilmiştir. Eşleşen sütun sayısının yüksek olması, algoritmanın homolog dizileri doğru bir şekilde hizalama yeteneğini yansıtır.

Bu değerlendirme, özellikle biyolojik verilerin analizi ve evrimsel ilişkilerin belirlenmesi bağlamında önemlidir. Eşleşen sütun sayısındaki artış, algoritmanın biyolojik bilgiyi daha doğru bir şekilde yorumladığını ve homolog bölgeleri başarılı bir şekilde hizaladığını gösterir.

Sonuç olarak, Clustal-SA algoritmasının performansını ölçmede kullanılan eşleşen sütun sayısı, algoritmanın biyolojik diziler üzerindeki hizalama başarısını değerlendirmek için önemli bir kriterdir. Bu değerlendirme, algoritmanın biyoinformatik uygulamalarda ne kadar etkili olduğunu anlamak için kullanışlı bir metriktir.

İlgili çalışmalar

MSA probleminin çözümü için kullanılan yöntemler genellikle iki kategoriye ayrılabilir: Dinamik Programlama ve Sezgisel Metotlar. Sezgisel metotlar içerisinde öne çıkan yaklaşımlar, Aşamalı Global Hizalama ve Yinelemeli Global Hizalamadır.

Dinamik programlama kullanılarak geliştirilen çözümler

Dinamik programlama, orijinal problemde en iyi çözümü bulmak için alt problemleri kullanır. Daha küçük problemleri en iyi şekilde çözmeye çalışır. Dinamik programlama böl ve yönet (divide-and-conquer) stratejisini kullanır (Diamantis ve Anna, 2005). MSA probleminde en uygun çözümü üretmek için en doğru yöntem dinamik programlamadır. Fakat N ayrı dizi için, standart ikili dizi hizalamasında oluşturulan matrisin N boyutlu eşdeğerini oluşturmak gerekir. Arama uzayı N ile üstel olarak artar. Üstel olarak artan karmaşıklıktan dolayı dinamik programlama MSA probleminin çözümünde genellikle tercih edilmemektedir. Dinamik programlama ikili hizalama problemin çözümünde tercih edilmektedir. Needleman–Wunch ve Smith-Waterman algoritmalarıdır dinamik programlamaya örnek verilebilir. Dinamik programlama algoritmaları çoklu dizi hizalama gerçekleştiren sezgisel yöntemlerin işlem adımlarında kullanılmaktadır (Diamantis ve Anna, 2005).

Her bir dizi indislenerek A skor matrisi $D(i,j)$ özyinelemeli şekilde Formül 4'te gösterildiği gibi oluşturulur.

$$D(i,j) = max \begin{cases} D(i-1,j-1) + S(x_i, x_j) \\ D(i-1,j) + G \\ D(i,j-1) + G \end{cases} \quad (4)$$

S(i, j) i ve j indekslerinde yer alan her bir amino asit karakterine karşılık gelen skor matrisindeki değeridir. G ise boşluk cezasıdır. Dinamik programlamadaki skorlama matrisi bu şekilde oluşturulmaktadır.

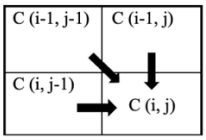
Şekil 4'te "GCAT" ve "GAT" dizilerinin dinamik programlama kullanılarak hizalanması gösterilmiştir. Skorlama ve geri izleme matrislerinin ilk sıra ve sütunları başlatma puanları ile doldurulmuştur. Daha sonra $D(2,2)$ hücresinden başlayarak skorlama matrisi doldurulmaya başlanmıştır. $D(2,2)$ hücresi sol üst çaprazındaki hücreye, üstündeki hücreye ve solundaki hücreye

bakılarak doldurulmaktadır. $D(1,1)$, $D(1,2)$ ve $D(2,1)$ skor matrisinden okunur. $S(S,A)$ Değeri ise BLOSUM62 skor matrisindeki $S \Leftrightarrow A$ eşleşmesinden alınır. Bu işlem, Formül 5'te gösterildiği gibidir. Formülde görüldüğü gibi en yüksek değer q_{diag}'de elde edilmiştir. $D(2,2)$ Hücresinde olduğu gibi tüm hücrelerin doldurulması sonucu skor ve geri izleme matrisi elde edilmiştir. Geri izleme matrisinde sol en alt hücreden başlanarak hücre içerisindeki yönler takip edilerek "Bitiş" noktasına gelinmiştir.

$$q_{diag} = D(1,1) + S(S,A) = 0 + 1 = 1,$$
$$q_{up} = D(1,2) + g = -10 + (-10) = -20,$$
$$q_{left} = D(2,1) + g = -10 + (-10) = -20 \quad (5)$$

		G	C	A	T
	0	-10	-20	-30	-40
G	-10				
A	-20				
T	-30				

		G	C	A	T
	Bitti	Sol	Sol	Sol	Sol
G	Yukarı				
A	Yukarı				
T	Yukarı				



		G	C	A	T
	0	-10	-20	-30	-40
G	-10	6	-4	-14	-24
A	-20	-4	6	0	-10
T	-30	-14	-4	6	5

		G	C	A	T
	Bitti	Sol	Sol	Sol	Sol
G	Yukarı	Köşegen	Sol	Sol	Sol
A	Yukarı	Yukarı	Köşegen	Köşegen	Sol
T	Yukarı	Yukarı	Yukarı	Köşegen	Köşegen

		G	C	A	T
	Bitti	Sol	Sol	Sol	Sol
G	Yukarı	Köşegen	Sol	Sol	Sol
A	Yukarı	Yukarı	Köşegen	Köşegen	Sol
T	Yukarı	Yukarı	Yukarı	Köşegen	Köşegen

Hizalama:
GCAT
G_AT

Şekil 4. Dinamik programlama kullanarak ikili dizi hizalama örneği [18]

Global hizalama dizilerin tüm uzunluklarını hizalanmaya çalışır. Çoğunlukla yüksek benzerliğe sahip diziler arasında kullanılır. Needleman-Wunch algoritması (Smith ve Waterman, 1981), boşlukların eklenmesine izin veren iki diziyi global olarak hizalamak için kullanılan dinamik programlama yöntemidir (Thompson ve ark. 1994) Algoritma, aslında büyük bir problemi (örneğin tam diziyi) bir dizi daha küçük probleme böler ve daha büyük probleme en uygun çözümü bulmak için daha küçük problemlerin çözümlerini kullanır. Dizileri uzunlukları boyunca hizalamaya çalışmak yerel alanlarda uyumsuzluğa neden olabilir. Smith-Waterman algoritması (Smith ve Waterman,1981), global hizalama sonucu oluşan sorunu gidermek için yerel hizalama yapan bir çözüm önermiştir.

Sezgisel metotlar kullanılarak geliştirilen çözümler

Sezgisel metotlar MSA probleminin çözümünde aşamalı, yinelemeli ve meta yöntemler olarak üçe ayrılır.

Aşamalı Global Hizalama: En benzer çiftle başlayan ve en uzak akraba olana doğru ilerleyen ikili hizalamaları birleştirerek bir MSA oluşturur. Burada iki temel sorun bulunmaktadır: yerel minimum sorunu ve hizalama parametrelerinin seçimi. Yerel minimum sorunu, hizalama stratejisinin "açgözlü" yaklaşımından kaynaklanmaktadır. Algoritma ilk olarak kılavuz ağacı takip ederek dizileri bir araya getirir. Bu nedenle hizalama işleminin başında yapılan hata ilerleyen adımlarda düzeltilemez. Hizalama parametrelerinin seçimi problemi ise yerel minimum problemi kadar önemlidir. Ağırlık matrisi ve belirlenen boşluk cezalarını algoritma için uygun olduğu varsayılmaktadır. Parametreler uygun olmadığında optimale yakın bir çözüm bulunmayacaktır (Diamantis ve Anna,2005). Aşamalı yaklaşımı baz alan ve MSA probleminin çözümünde yaygın olarak kullanılan uygulamalar ise CLUSTAL W, T-COFFEE ve MUSCLE programlarıdır. Aşamalı hizalamanın hızını ve hizalama doğruluğunu arttırmak için devam eden çalışmalar, sıralı ve hiyerarşik kümeleme yöntemlerini birleştirerek hızlı bir şekilde kılavuz ağacı oluşturmayı hedeflemektedir. Bu amaçla, Chen ve diğer araştırmacılar tarafından 2023 yılında StarTree adı verilen yeni bir yöntem tanıtılmıştır. StarTree yöntemi, hız ve doğruluk açısından aşamalı hizalama işlemlerini iyileştirmeyi amaçlamaktadır (Chen ve ark., 2023).

Clustal algoritmasından meydana gelen Clustal seri programları, küresel çoklu dizi hizalaması için yaygın olarak kullanılmaktadır. İlk Clustal programı 1988 yılında Des Higgins tarafından yazılmıştır. Geliştirildiği dönem şartlarında zayıf hesaplama gücüne sahip olan kişisel bilgisayarlarda verimli olarak çalışacak şekilde tasarlanmıştır. Dinamik programlamayı kullanan aşamalı hizalama stratejisini benimsenmiştir.

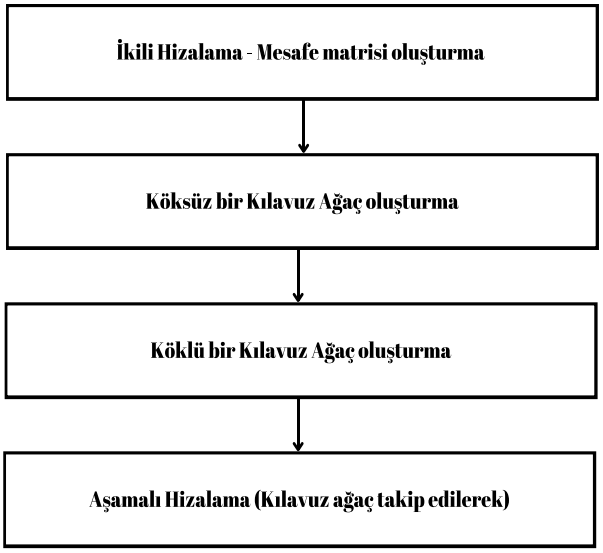
Çoklu hizalama, bir kılavuz ağacındaki dallanma sırasını takip eden bir dizi ikili hizalama ile aşamalı olarak oluşturulur. İlk ön karşılaştırmada hızlı bir kelime bazlı hizalama algoritması kullanılmış ve kılavuz ağacı UPGMA yöntemi kullanılarak oluşturulmuştur. 1992'de, profil hizalamalarını (mevcut hizalamaların hizalamaları) ve Komşu Birleştirme (NJ) yöntemini kullanarak çoklu hizalamadan ağaçlar oluşturma olanağını içeren ClustalV adı verilen yeni bir sürüm piyasaya sürülmüştür. Serinin üçüncü nesli, 1994 yılında piyasaya sürülen ClustalW, dizi ağırlıklandırma, konuma özgü boşluk cezaları ve çoklu hizalamanın her aşamasında uygun bir kalıntı karşılaştırma matrisinin otomatik seçimi dahil olmak üzere hizalama algoritmasında bir dizi iyileştirme içeriyor (Aarts ve Van Laarhoven, 1987).

Temel Clustal algoritması üç adımdan oluşur.

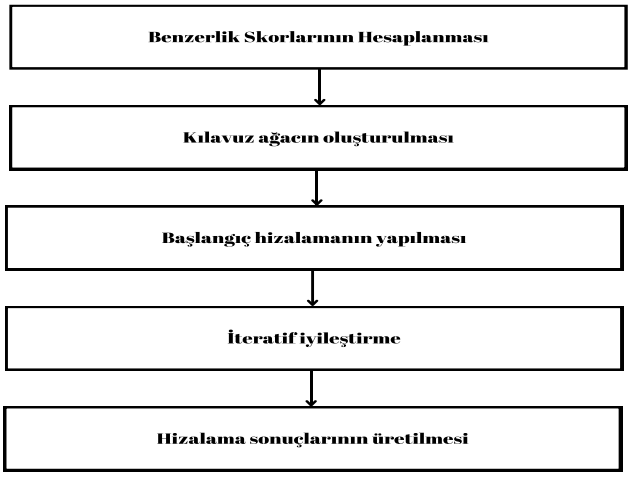
- Her dizi çiftinin uzaklığını veren bir mesafe matrisini hesaplamak için tüm dizi çiftleri hesaplanır.
- Mesafe matrisinden yararlanılarak bir kılavuz ağaç (guide tree) oluşturulur.
- Diziler kılavuz ağaçtaki dallanma sırasına göre aşamalı olarak hizalanır.

Clustal algoritmasının işlem adımları Şekil 5'te gösterilmiştir.

MUSCLE programı, birden fazla dizi hizalamak için kullanılan etkili bir araçtır. Bu algoritma, iteratif bir ilerleme yöntemi kullanarak çoklu diziler arasındaki benzerlikleri belirler ve hizalamayı iyileştirir. Öncelikle, diziler arasındaki benzerlikleri tahmin etmek için bir rehber ağaç oluşturur ve ardından bu rehber ağacı kullanarak başlangıç bir hizalama yapar. Daha sonra, hizalama iteratif olarak iyileştirilir, böylece diziler arasındaki benzerlikler ve tutarlılık artar. MUSCLE, genellikle hızlı, verimli ve doğru hizalamalar üretme yeteneğiyle öne çıkar ve moleküler biyoloji ve biyoinformatik çalışmalarında yaygın olarak kullanılır (Edgar, 2004).



Şekil 5. Clustal algoritmasın temel işlem adımları



Şekil 6: MUSCLE programı temel işlem adımları

MUSCLE programının işleyiş adımları aşağıda açıklanmış ve işlem adımları kısaca şekil 6'da gösterilmiştir.

• Giriş Dizilerinin Alınması: Kullanıcı tarafından sağlanan veya bir dosyadan okunan giriş dizileri alınır.

• Benzerlik Skorlarının Hesaplanması: Giriş dizileri arasındaki benzerlik skorları hesaplanır. Bu skorlar genellikle iki dizi arasındaki hizalamada kullanılan bir skor matrisini oluşturur.

• Rehber Ağacının Oluşturulması: Hesaplanan benzerlik skorlarına dayanarak, rehber ağacı oluşturulur. Bu ağaç, diziler arasındaki evrimsel ilişkileri gösterir.

• Başlangıç Hizalamasının Yapılması: Rehber ağacı kullanılarak, başlangıç bir hizalama oluşturulur. Bu hizalama genellikle önyüklemeli hizalama yöntemiyle yapılır.

• İteratif İyileştirme Adımları: Başlangıç hizalaması üzerinde bir dizi iteratif iyileştirme adımı gerçekleştirilir. Her adımda, hizalama daha da iyileştirilir ve diziler arasındaki benzerlikler daha doğru bir şekilde yorumlanır.

• Sonuçların Üretilmesi: İteratif iyileştirme adımları tamamlandıktan sonra, son hizalama sonuçları üretilir. Bu sonuçlar genellikle bir dosyaya veya ekrana yazdırılır.

Yinelemeli Global Hizalama: Yinelemeli hizalama, bir hizalama üretebilen ve daha fazla iyileştirme yapılamayana kadar bir dizi döngü (yineleme) aracılığıyla onu iyileştirebilen algoritmalara dayanır. Yinelemeli yöntemler, hizalamayı iyileştirmek için kullanılan stratejiye bağlı olarak deterministtik veya skolastik olabilir. En basit yinelemeli stratejiler deterministtir. Çoklu hizalamadan dizileri tek tek çıkarmayı ve bunları kalan dizilerle yeniden hizalamayı içerirler. Bazı yöntemler hem aşamalı hem de yinelemeli yaklaşımı kullanıyor olabilir. Yinelemeli yaklaşımda elde edilen hizalamada iyileştirme olmuyorsa bir müddet sonra algoritma sonlandırılır. Stokastik yinelemeli yöntemler, Hidden Markov Model (Gizli Markov Modeli) eğitimi, Simulated Annealing (Benzetimli Tavlama), genetik algoritmalar, yapay arı kolonisi ve parçacık sürü optimizasyonu gibi çeşitli algoritmaları içerir. Bu tür yöntemler, çoklu dizi hizalaması gibi birçok uygulamada kullanılmaktadır. Örneğin, literatürde yaygın olarak kullanılan SAGA stokastik yinelemeli yöntemlere bir örnektir (Diamantis ve Anna, 2005). Ayrıca, Paruchuri ve arkadaşlarının (Paruchuri ve ark., 2023) parçacık sürü optimizasyonunu kullandığı çoklu dizi hizalama uygulaması da bu tür bir yaklaşımın bir örneğidir. Bu tür yöntemler, karmaşık diziler arasında hizalama yaparak biyolojik verilerin analizinde kullanılır ve genellikle büyük veri setlerini ele almak için etkili bir şekilde çalışırlar.

Çalışmada kullanılan Benzetimli Tavlama algoritması, temel olarak Monte Carlo yönteminin bir genellemesidir. Konsept, tavlama sürecinde sıvıların donması veya metallerin yeniden kristalleştirilmesine dayanmaktadır. Bu yaklaşımın kombinatoryal (bütünleştirici) problemlere genelleştirilmesi kolaydır. Termodinamik sistemin mevcut durumu, kombinatoryal(bütünleştirici) problemin mevcut çözümüne benzer, termodinamik sistemin enerji denklemi amaç fonksiyonuna benzer ve temel durum global minimuma benzer (Russel ve Norvig, 2010).

Benzetimli tavlama (SA) algoritmasının MSA problemi çözümünde kullanımı

SA algoritması, MSA problemini çözmek için kullanılan etkili bir optimizasyon algoritmasıdır. SA, kombinatoryal optimizasyon problemlerini çözmek ve genel çözüm alanlarını araştırmak için kullanılan olası bir algoritmadır. SA algoritmasının sözde kodu Şekil 7'te gösterilmiştir. Benzetimli Tavlama Algoritmasının MSA problemlerindeki kullanımıyla ilgili bazı temel bilgiler şöyledir:

Optimizasyon süreci

SA, bir enerji fonksiyonu üzerinden çalışır. MSA probleminde enerji, hizalama skoru veya benzer bir ölçü olabilir. Çalışmada enerji hizalama skoru olarak ele alınmıştır. Algoritma, enerji fonksiyonunu maksimize etmeye çalışarak en iyi hizalamayı bulmaya çalışır.

Başlangıç durumu

SA, başlangıçta rastgele bir çözümle başlar. Bu, MSA probleminde başlangıç hizalamasının belirlenmesinde kullanılır. Gerçekleştirilen çalışmada başlangıç hizalama ClustalW programının çalıştırılmasıyla elde edilen hizalamadır.

Sıcaklık azaltma

SA'nın belirgin bir özelliği, zamanla sıcaklığın azaltılmasıdır. Sıcaklık, çözüm alanında rastgele gezinme olasılığını etkiler. Yüksek sıcaklıkta daha fazla kabul edilebilir olasılıklı çözümlere izin verirken, düşük sıcaklıkta daha dar bir arama alanında odaklanmayı sağlar. SA algoritmasının uygulanmasındaki en büyük sorun, kombinatoryal problemdeki serbest bir parametreye göre T sıcaklığı için açık bir analojinin bulunmamasıdır. Ayrıca, yerel minimumlarda sürüklenmenin önlenmesi başlangıç sıcaklığının seçimine, her sıcaklıkta kaç yinelemenin gerçekleştiğine ve her adımda sıcaklığın ne kadar azaldığına bağlıdır.

Gezgin hareket

SA, mevcut çözümü bir miktar değiştirerek, yeni bir çözümü değerlendirir. Bu, MSA probleminde dizilerin sırasını veya hizalamayı değiştirmek anlamına gelir. Gerçekleştirilen çalışmada bu değişiklik mutasyon işlemi ile sağlanmıştır. Mutasyon işlemi çözümlerin benzer olmasını engeller ve SA algoritmasının yerel çözümlerden kaçma olasılığını artırır. SA algoritmalarında mutasyon oranı düşük olarak tutulmaktadır. Bunun nedeni, yüksek mutasyon oranlarının SA algoritmasını ilkel rastgele algoritmaya dönüştürmesidir (Mirjalili, 2019). SA algoritmasında bir önceki adımda oluşan C1 kromozomu üzerinde uygulanmaktadır. Kromozom nükleotid ve boşluk karakterlerinden oluşmaktadır. Mutasyon işlemi C1 kromozomunda rastgele belirlenen bir konuma boşluk karakterinin eklenmesi veya boşluk karakterinin silinmesi ile gerçekleşir. Boşluk karakterinin eklenmesi ve silinmesi değeri 0,5 olan bir mutasyon olasılığına bağlıdır.

Kabul kriteri

Yeni çözümün enerji fonksiyonu (hizalama skoru) eski çözümden daha iyi ise, yeni çözüm kabul edilir. Ancak, daha kötü bir çözüm durumunda bile belirli bir olasılıkla kabul edilebilir. Bu, yerel minimumlardan kaçınmaya ve genel çözüm alanını daha iyi keşfetmeye yardımcı olur.SA algoritması MSA problemi çözümü için uygulanırken döngü, kromozom (aday çözüm) üzerindeki rastgele hareketi seçmektedir ve mutasyon olarak temsil edilir. Güncel (current) değişkeni ise hizalanan dizilerimizin olduğu bir kromozomdur. Bu kromozom üzerinde 0,5 olasılığına dayanarak mutasyon işlemi uygulanmaktadır. Bu işlemde uygunluk değerini arttırmak hedeflenir. Eğer mutasyon işlemi uygunluk değerini yükseltiyorsa hemen kabul edilir ve güncel değerine mutasyon sonucu elde edilen kromozom geçer. Aksi durumda $e^{-DE/T}$ ifadesinin 0-1 aralığında rastgele seçilen bir sayıdan büyük olması durumunda mutasyon sonucu elde edilen kromozom kabul edilir.

Yinelemeli iyileştirme

SA, belirli bir sıcaklık azaltma oranında ve belirli bir iterasyon sayısında çalışarak çözümü iyileştirmeye devam eder. Bu süreç, global optimuma yaklaşmayı amaçlar. Eğer sıcaklık yavaş bir şekilde düşürülürse, olasılık 1'e yaklaşırken algoritma global optimumu bulur. Algoritma başarısı için ilk sıcaklık oranı ve soğuma oranı oldukça önemlidir (Russel ve Norvig, 2010).

```
Algoritma SimulatedAnnealing(problem):
  Çözüm = BaşlangıçÇözümüOluştur(problem)
  EnİyiÇözüm = Çözüm
  T = BaşlangıçSıcaklık
  T_min = SonSıcaklık
  α = SoğutmaOranı

  while T > T_min:
    for i in range(iterasyon_sayısı):
      Yeni_Çözüm = Rastgele_Tetikle(Çözüm) # Mutasyon ile rastgele çözüm üretme
      Eski_Maliyet = Hesapla_Maliyet(Çözüm)
      Yeni_Maliyet = Hesapla_Maliyet(Yeni_Çözüm)
      ΔE = (Yeni_Maliyet) - (Eski_Maliyet)

      if ΔE < 0:
        Çözüm = Yeni_Çözüm
        if Yeni_Maliyet < Hesapla_Maliyet(EnİyiÇözüm):
          EnİyiÇözüm = Yeni_Çözüm
      else:
        P = exp(-ΔE / T)
        if Rastgele(0, 1) < P:
          Çözüm = Yeni_Çözüm

    T = T * α

  Return EnİyiÇözüm
```

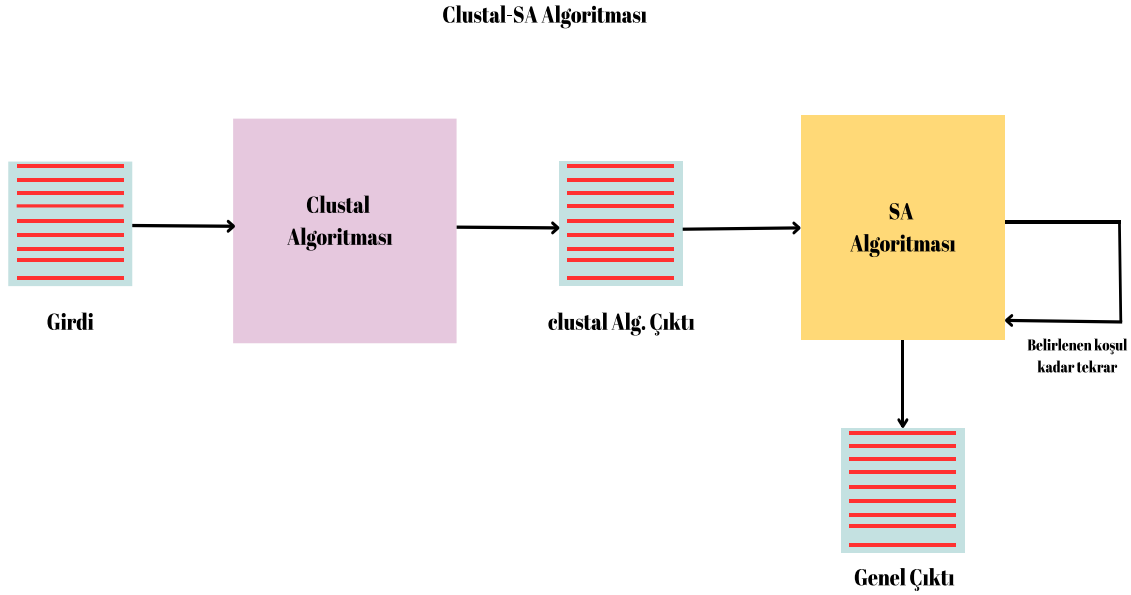
Şekil 7. Benzetimli tavlama algoritmasının sözde kodu

Önerilen Meta-sezgisel tabanlı MSA algoritması: Clustal-SA

Bu çalışmada, Clustal ve Benzetimli Tavlama (Simulated Annealing-SA) algoritmalarının iyi yönleri birleştirilerek hibrit bir MSA algoritması önerilmiştir. Clustal-SA algoritması, dizi setine Clustal algoritması uygulanması ile elde edilen hizalamanın SA algoritmasına girdi olarak verilmesi ve SA algoritmasının yeni bir hizalama üretmesi ile oluşmaktadır. Şekil 8'de Clustal-SA algoritmasının akış şeması gösterilmektedir.

Clustal algoritması bellek kullanımı ve çalışma hızı açısından avantajlı olduğu için tercih edilmiştir. Clustal algoritmasının aşamalı yaklaşımın sonucu olan açgözlü yaklaşımından dolayı meydana gelecek hizalama hatalarından kurtulmak için SA algoritmasından yararlanılmıştır. Clustal-SA algoritmasında uygulanan işlem adımları şu şekildedir:

1. Dizi veri seti üzerinde ilk olarak Clustal algoritması uygulanmaktadır.
2. Uygulama sonucunda elde edilen çoklu dizi hizalama çıktısı SA algoritmasına girdi olarak verilmektedir.
3. SA algoritması ile girdi olarak verilen hizalama üzerinde mutasyon işlemleri (boşluk ekleme, boşluk silme) uygulanmaktadır. Hizalama üzerinde uygulanan mutasyon işlemlerinin sayısı SA algoritmasının parametre değerlerine göre değişmektedir.
4. SA algoritması tamamlandığında elde edilen hizalama Clustal algoritmasının çıktısıdır.
5. Clustal-SA algoritması sonucunda elde edilen hizalama, Clustal algoritması uygulanarak elde edilen hizalama ile hizalama skoru ve eşleşen sütun sayısına göre bir karşılaştırılma yapılmaktadır.



Şekil 8. Önerilen Clustal-SA algoritması akış şeması

Kullanılan veri seti ve özellikleri

Bu çalışmada, geliştirilen algoritmaların doğruluk ve hızlarını test etmek, MSA probleminin çözümü için sık tercih edilen algoritmalar ile başarılarını karşılaştırmak için Cicer reticulatum kloroplast geni NCBI sitesinden alınarak kullanılmıştır. Bu gen üzerinde belirli indisler arasında bulunan genler seçilmiştir. Cicer reticulatum kloroplast geninin nasıl kullanıldığı aşağıda detaylı bir şekilde açıklanmaktadır.

Cicer reticulatum kloroplast geni için NCBI sitesinden bu genin referans veri setinden başlangıç ve bitiş indisleri belli olan bazı genler seçilmiştir. Referans veri setinde CDS başlığı ile genlerin ismi

belirtilmektedir. Genlerin ismi ile genlerin başlangıç ve bitiş adresleri de verilmektedir. Başlangıç ve bitiş indisleri verilen genler referans veri seti içerisinden elde edilmiştir. Daha sonra bu genlerin her birinin tamlayanı alınmıştır. (A'yı T'ye, T'yi A'ya, C'yi G'ye, G'yi C'ye dönüştürme işlemi) ve tamlayanı alınan genler ile kendi elde ettiğimiz Cicer reticulatum kloroplast geni üzerinde sezgisel bir yaklaşıma dayalı eşleştirme yapılmıştır.

Sezgisel yaklaşımda tamlayanı(complement) alınan genin son 10 veya ilk 10 karakterlerinin eşleştiği indisler baz alınarak yaklaşık eşleştirme yapılmıştır. Eğer son 10 veya ilk 10 karakter tam olarak eşleşmiyorsa k-uyuşmazlık(mismatch) sayısı 1'er azaltılarak yaklaşık dizi eşleştirme gerçekleştirilmiştir. Buradaki temel amaç, tamlayanı alınan gen bölgesini doğrudan dizi hizalama işlemine tabi tutmadan önce yaklaşık dizi eşleştirme işlemini gerçekleştirerek olası eşleşebilecek indisleri ortaya çıkarabilmektir.

Son k adet veya ilk k adet dizi eşleşme elde edildiğinde eşleşmenin olduğu indisin başlangıç indisine tamlayanı alınan genin başlangıcı hizalanarak dizi hizalama gerçekleştirilir ve skor elde edilir. Bunlar, son k ve ilk k karakterin eşleştiği tüm yaklaşık eşleşmelerin elde edildiği indisler için tek tek yaparak dizi hizalama ve skor hesaplama işlemi gerçekleştirilmiştir.

İlk 10 ve son 10 nükleotid eşleşmesi olmadığında 1'er azaltarak ilk ve son 9,8,7, ... gibi olası tüm nükleotid eşleşmelerine bakılmıştır. İlk 10,9, 8... nükleotid eşleşmesi gerçekleştiği durumda eşleşmenin başladığı indis baz alınarak tamlayan uzunluğu kadar bir gen parçası alınmıştır. Son 10,9, 8... nükleotid eşleşmesi gerçekleştiğinde alınan nükleotid dizisinin son elemanı son indis alınarak geriye doğru tamlayanı uzunluğu kadar gidilip aranan genin uzunluğu kadar gen parçası elde edilmiştir. Cicer reticulatum kloroplast geni için NCBI sitesinden bu genin referans veri setinden başlangıç ve bitiş indisleri belli olan "rps12", "pbf1", "rps18" genleri seçilmiştir.

Veri setinin elde edilmesi

Cicer reticulatum kloroplast geninden alt veri setlerinin hizalama işlemi için nasıl elde edildiği aşağıdaki gibidir. Örn. *CDS: gene" rps12"* geni için;" rps12" genine ait "trnN-GUU" isimli 917. . . 990 indisleri arasında bulunan gen ve genin tamlayanı aşağıda verilmiştir.

Gen:TCTCCCCAAATAGGATTTGAACCTACGACCAATCGGTTAACAGCCGACCGCTCTACCACTGAGCTACTGAGGAA

Tamlayan:AGAGGGGTTTATCCTAAACTTGGATGCTGGTTAGCCAATTGTCGGCTGGCGAGATGGTGACTCGATGACTCCTT

• İlk 10 alt dizi: AGAGGGGTTT

– İlk 8 eşleşen AGAGGGGT: (indis aralığı 72959-73032)

AGAGGGGTATTCTCCTATATTTTTTTTGTATCATTTTGGCGGCATGGCCGAGTGGTAAGGCGGGGGACTGCAAA

• Son 10 alt dizi: ATGACTCCTT

– İlk 9 eşleşen ATGACTCCT: (indis aralığı 80163-80236)

AGCTCGCGCAGCTGCTGCAGGATTTGAAAAAGGAATTGATCGAGATTTTGAGCCTGTTCTTTCC**ATGACTCCT**C

"trnN-GUU" genin tamlayanının ilk 10 nükleotidinde eşleşme gerçekleşmemiştir. Bu nedenle nükleotid sayısı 1 azaltılarak eşleşmeler kontrol edilmiştir. İlk 8 nükleotidin eşleşmesinin sağlandığı indis konumları bulunmuştur. Son 10 nükleotidinde son 9 nükleotid alındığında bir eşleşme olmuştur. Bu eşleşme sonucu elde edilen indisler belirtilmiştir. İlk ve son eşleşme sonucunda elde edilen gen blokları tamlayan ile hizalanmıştır.

Yukarıda örneği verilen işlemler her bir gen ve tamlayanı için ayrı ayrı uygulanmıştır. Çizelge 1'de elde edilen genler, genlerin dizi uzunlukları, dizi sayıları ve her bir gen için dizi benzerlikleri verilmişidir. Gen benzerlikleri Hamming mesafe algoritması kullanılarak hesaplanmıştır.

Cicer reticulatum kloroplast veri setinden elde edilen genlere ek olarak uzunlukları aynı olmayan Opuntia adlı veri setinde geliştirilen algoritmaların performansını test etmek için kullanılmıştır. Dizi seti içerisinde 8 adet DNA dizisi bulunmaktadır. Veri setindeki en uzun dizi boyutu 902, en kısa dizi boyutu ise 893'tür. Gen dizisi benzerliği ise %63'tür.

Çizelge 1. Veri setlerinden elde edilen genler ve özellikleri

Elde Edilen Gen	Gen İsmi	Gen Uzunluğu	Dizi Sayısı	Benzerlik (%)
	trnN-GUU	82	3	28
rps12	rrn5	143	4	27
	rrn4.5	124	3	30
pbf1	psbT	118	3	26
rps18	rpl33	242	11	33

BULGULAR VE TARTIŞMA

Bu bölümde, geliştirilen Clustal-SA algoritmasındaki parametrelerin değerlerinin değiştirilmesiyle elde edilen skorlar detaylı bir şekilde incelenmiştir. Farklı parametre kombinasyonları kullanılarak elde edilen yüksek skorlar, geliştirilen algoritmanın başarısını değerlendirmek amacıyla kullanılmıştır. Bu skorlar, dizi hizalamada yaygın olarak kullanılan ClustalW programının performansıyla karşılaştırılmıştır.

Clustal-SA algoritması, Python programlama dili ile kodlanmış ve deneyler sırasında elde edilen sonuçlar için süre ve skor değerleri hesaplanmıştır. Bu hesaplamalar için 30 tekrarın standart sapması (SD) ve ortalaması (AVG) alınmıştır. Ayrıca, hizalama skorları SP (Çiftlerin Toplamı) amaç fonksiyonu kullanılarak belirlenmiştir.

Elde edilen sonuçlar, farklı parametre setleriyle yapılan denemelerin optimal skorları belirlemede etkili olduğunu göstermektedir. Yüksek skor elde edilen parametreler, geliştirilen Clustal-SA algoritmasının başarısını artırmada önemli bir rol oynamaktadır.

Bu parametre optimizasyonu ve karşılaştırma çalışmaları, geliştirilen algoritmanın biyolojik verilerin analizi ve dizi hizalama probleminin çözümü için etkili bir araç olduğunu vurgulamaktadır. Elde edilen sonuçlar, Clustal-SA algoritmasının biyoinformatik alanındaki potansiyelini desteklemekte ve gelecekteki çalışmalara yönlendirme sağlamaktadır.

Ayrıca, MSA algoritmalarının performansını değerlendirmek için seçilen metrikler arasında biyolojik doğruluk, eşleşen sütun sayısı ve çalışma süresi önemli bir yer tutmaktadır. Bu metrikler, algoritmaların gerçek dünya biyolojik verilerle uyumlu hizalamalar üretme yetenekleri, hizalama kalitesi ve uygulama süreleri hakkında değerli bilgiler sağlamaktadır.

SA Algoritması Parametre Optimizasyonu

Bu çalışmada kullanılan SA algoritmasının performansını etkileyen önemli faktörler, başlangıç sıcaklığı ve soğuma oranıdır. Bu parametreler, SA algoritmasının iterasyon sayısını belirlerken çalışma süresi, başlangıç sıcaklığı arttığında artmakta ve soğuma oranı arttığında ise çalışma süresi azalmaktadır.

Başlangıç sıcaklığı ve soğuma oranı değerlerini belirlemek için farklı kombinasyonlar denendi. Başlangıç sıcaklığı sırasıyla 50.000, 100.000 ve 150.000 olarak seçildi. Tüm veri setlerini değerlendirdiğimizde, başlangıç sıcaklığı değeri 100.000 olarak belirlendi. Bu değer seçildiğinde, 50.000 başlangıç sıcaklığına göre hizalama skoru başarısı genelde arttı. Başlangıç sıcaklığı 150.000 olarak seçildiğinde ise genelde başarısız bir performans görüldü. Çizelge 2'de Clustal-SA

algoritmasında, SA algoritmasının başlangıç sıcaklığı parametre değerlerinin değiştirilmesi sonucu elde edilen skor, sütun eşleşme sayısı ve çalışma süresi sonuçları gösterilmiştir.

Soğuma oranı için ise 0,003 ve 0,005 değerleri denenmiş, tüm veri setlerini değerlendirdiğimizde soğuma oranı 0,003 olarak seçilmiştir. Bu değer seçildiğinde, 0,005 değerine göre hizalama skoru başarısı ve eşleşen sütun sayısı değerlerine göre Clustal-SA algoritması daha iyi hizalama performansı göstermiştir. Çizelge 3'te Clustal-SA algoritmasında, SA algoritmasının soğuma oranı parametre değerlerinin değiştirilmesi sonucu elde edilen skor, sütun eşleşme sayısı ve çalışma süresi sonuçları gösterilmiştir.

Sonuç olarak, önerilen hibrit Clustal-SA algoritmasında, başlangıç sıcaklığı ve soğuma oranı parametre değerleri sırasıyla 100.000 ve 0,003 olarak belirlenmiştir. Bu değerler, yapılan karşılaştırmalar sonucunda en iyi performansı sağlamıştır ve önerilen algoritmanın dizi hizalama problemine etkili bir çözüm sunduğunu göstermektedir.

Çizelge 2. SA algoritmasında başlangıç sıcaklığının faklı değerler almasıyla skor ve eşleşen sütun sayısının (Sütun E.) (ortalama ve standart sapma ile) değişimi

Parametre	Veri Seti		Başlangıç Sıcaklığı=50.000			Başlangıç Sıcaklığı=100.000			Başlangıç Sıcaklığı=150.000		
			Skor	Sütun E.	Süre (sn)	Skor	Sütun E.	Süre(sn)	Skor	Sütun E.	Süre(sn)
Başlangıç sıcaklığı	trnN-GUU_rps12	AVG	889,06	15	4,72	915,333	17	5,55	908,6	18	6,28
		SD	5,650	0,498	0,05	25,577	0,94	0,03	11,35	0,92	0,06
	rrn4.5_rps12	AVG	1200,66	21	5,82	1194,0	21	6,77	1194,0	21	7,50
		SD	2,981	0	0,016	0	0	0,02	0	0	0,04
	rrn5_rrps12	AVG	2188,0	11,23	9,32	2254,93	10	11,22	2253,46	9	12,11
		SD	10,70	1	0,019	41,56	0,88	0,12	30,56	0,94	0,03
	psbT_pbf1	AVG	1240,4	21	6,068	1254,66	22	7,26	1237,66	21	7,95
		SD	11,55	0,47	0,031	24,37	1,62	0,03	27,48	0,87	0,08
	psl33_rps18	AVG	30085,86	8	60,77	30066,0	7	71,54	30022,53	7	77,54
		SD	94,83	0,47	0,38	78,12	0,33	0,17	28,09	0,44	0,25
	Opuntia	AVG	194472,0	871	94,49	194472	871	110,791	194472,0	871	121,01
		SD	0	0	0,46	0	0	0,37	0	0	0,89

Çizelge 3. SA algoritmasında soğuma oranının faklı değerler almasıyla skor ve eşleşen sütun sayısının (Sütun E.) (ortalama ve standart sapma ile) değişimi

	Veri Seti		Soğuma Oranı = 0,003			Soğuma Oranı = 0,005		
			Skor	Sütun E.	Süre(sn)	Skor	Sütun E.	Skor
Soğuma Oranı	trnN-GUU_rps12	AVG	915,33	17	5,55	908,13	17	3,31
		SD	25,577	0,94	0,03	20,88	1,39	0,02
	rrn4,5_rps12	AVG	1194,0	21	6,77	1194,0	21	4,10
		SD	0	0	0,02	0	0	0,027
	rrn5_rrps12	AVG	2254,93	10	11,22	2165,06	9	6,75
		SD	41,56	0,88	0,12	15,38	0,74	0,02
	psbT_pbf1	AVG	1254,66	22	7,26	1228,46	20	4,28
		SD	24,37	1,62	0,03	20,51	0,65	0,01
	psl33_rps18	AVG	30066,0	7	71,54	29996,2	7	42,72
		SD	78,12	0,33	0,17	80,76	0,48	0,09
	Opuntia	AVG	194472	871	110,79	194472,0	871	66,43
		SD	0	0	0,37	0	0	0,20

Performans Karşılaştırması: Clustal-SA algoritması, clustalW ve muscle

Uygun parametreler belirlendikten sonra, önerilen Clustal-SA algoritmasının performansını değerlendirmek amacıyla elde edilen hizalama skoru, sütun eşleşme sayısı ve çalışma süresi değerleri, çoklu dizi hizalama (MSA) problemlerinin çözümünde sıklıkla kullanılan ClustalW ve MUSCLE programlarıyla karşılaştırılmıştır.

Bu karşılaştırma, Clustal-SA'nın belirli parametreler altında nasıl performans gösterdiğini anlamamıza yardımcı olurken, aynı zamanda ClustalW ve MUSCLE gibi yaygın olarak kullanılan alternatif yöntemlerle karşılaştırarak Clustal-SA'nın etkinliğini ve uygunluğunu değerlendirmemize

olanak sağlar. Bu değerlendirme, çoklu dizi hizalama problemlerine yönelik en uygun çözümü belirlerken önemli bir rehber sağlar.

Çizelge 4'te, ClustalW ile önerilen Clustal-SA algoritması arasındaki karşılaştırma detaylı olarak sunulmuştur. Skor, eşleşen sütun ve çalışma süresi (ortalama ve standart sapma ile) başarı metriklerine göre karşılaştırma yapılmıştır. Bu karşılaştırma, her iki algoritmanın performansını ayrıntılı bir şekilde incelememize olanak sağlar ve en uygun hizalama aracını seçerken önemli bir rehber sağlar.

Çizelge 4. Clustal-SA algoritması, ClustalW ve MUSCLE programının skor, eşleşen sütun (sütun Eş.) ve çalışma zamanı (ortalama ve standart sapmayla) başarı metriklerine göre performanslarının karşılaştırılması

Veri Seti		Clustalw			MUSCLE			Clustal-SA		
		Skor	Sütun Eş.	Süre(sn)	Skor	Sütun Eş.	Süre(sn)	Skor	Sütun Eş.	Süre(sn)
trnN-GUU_rps12	AVG	844	15	0,008	918	21	0,094	915,333	17	5,55
	SD	0	0	0	0	0	0	25,577	0,94	0,03
rrn4.5_rps12	AVG	1194	21	0,007	1208	19	0,108	1194	21	6,77
	SD	0	0	0	0	0	0	0	0	0,02
rrn5_rrps12	AVG	2044	8	0,007	2350	14	0,133	2254,93	10	11,22
	SD	0	0	0	0	0	0	41,56	0,88	0,12
psbT_pbf1	AVG	1046	17	0,006	1000	23	0,047	1254,66	22	7,26
	SD	0	0	0	0	0	0	24,37	1,62	0,03
psl33_rps18	AVG	29780	7	0,029	28252	7	0,298	30066	7	71,54
	SD	0	0	0	0	0	0	78,12	0,33	0,17
Opuntia	AVG	194472	871	0,293	194632	871	0,771	194472	871	110,791
	SD	0	0	0	0	0	0	0	0	0,37

Önerilen Clustal-SA algoritması, ClustalW çoklu dizi hizalama programına göre trnN-GUU_rps12, rrn5_rps12, psbT_pbf1 ve psl33_rps18 veri setlerinde hizalama skoru ve sütun eşleşmesi performans değerlendirme kriterlerine göre daha iyi bir performans göstermiştir. Ancak, Rrn4.5_rps12 ve Opuntia veri setlerinde her iki algoritma da aynı hizama skoru ve sütun eşleşme sayısını sağlamıştır.

Önerilen Clustal-SA algoritması, ClustalW çoklu dizi hizalama programına göre trnN-GUU_rps12, rrn5_rps12, psbT_pbf1 ve psl33_rps18 veri setlerinde daha iyi bir hizalama skoru ve daha fazla sütun eşleşmesi sağlamıştır. Ancak, Rrn4.5_rps12 ve Opuntia veri setlerinde her iki algoritma da benzer hizalama skoru ve sütun eşleşme sayısı elde etmiştir.

MUSCLE programına göre, Clustal-SA algoritması psbT_pbf1 ve psl33_rps18 veri setlerinde daha yüksek bir hizalama skoru sağlamıştır. Ayrıca, rrn4.5_rps12 veri setinde daha iyi bir sütun eşleşmesi elde edilmiştir. Bununla birlikte, trnN-GUU_rps12, rrn4.5_rps12 ve psl33_rps18 veri setlerinde MUSCLE'ın daha başarılı olduğu gözlemlenmiştir, ancak Clustal-SA algoritması çok benzer sonuçlar vermiştir.

Bu durum, Clustal-SA'nın belirli veri setlerinde MUSCLE'dan daha iyi performans gösterebileceğini, ancak genel olarak her iki algoritmanın da benzer hizalama kalitesi sağladığını göstermektedir. Bu bulgular, belirli veri setleri veya gereksinimlere bağlı olarak en uygun hizalama aracını seçerken dikkate alınabilir.

Çalışma süresi performansına göre, ClustalW programı, MUSCLE ve Clustal-SA algoritmalarına kıyasla her bir veri seti için en iyi sonucu vermiştir. Bu durumun temel nedeni, ClustalW programında kullanılan Clustal algoritmasının açgözlü yaklaşımı benimsemesidir. Açgözlü yaklaşım, hizalama yaparken mevcut durumu değerlendirerek bir sonraki en iyi adımı seçer, bu da genellikle daha hızlı çalışma süresi sağlar.

Bu sonuçlar, Clustal-SA algoritmasının belirli veri setlerinde ClustalW ve MUSCLE'a kıyasla daha iyi performans gösterdiğini, ancak çalışma süresi açısından ClustalW'nin daha etkili olduğunu

göstermektedir. İleriki çalışmalarda bu algoritmaların avantajları ve dezavantajları daha ayrıntılı bir şekilde incelenebilir.

SONUÇ

Bu çalışma, biyolojik dizilerin analizi ve yorumlanması için Clustal ve Simulated Annealing (SA) algoritmalarını birleştiren bir hibrit yaklaşım olan Clustal-SA algoritmasının performansını değerlendirmiştir. Çalışmanın ana hedefi, çoklu dizi hizalama probleminin çözümü için daha etkili bir çözüm sunmaktır.

Çalışmanın sonuçları, Clustal-SA algoritmasının belirli veri setlerinde ClustalW programından daha iyi hizalama skoru ve sütun eşleşmesi performansı sergilediğini göstermektedir. Ayrıca, MUSCLE programına göre bazı veri setlerinde daha başarılı olduğu veya benzer sonuçlar verdiği belirlenmiştir.

Bu sonuçlar, önerilen hibrit algoritmanın, belirli gen dizilerinin analizi konusunda güvenilir sonuçlar elde etmede etkili olduğunu ortaya koymaktadır.

Ayrıca, farklı uzunluktaki dizi veri setlerinde elde edilen benzer sonuçlar, Clustal-SA algoritmasının geniş bir genetik çeşitlilik içeren veri setlerine uygulanabilirliğini vurgulamaktadır.

Çalışmanın gelecekteki yönlendirmeleri arasında, Clustal-SA algoritmasındaki SA algoritmasının mutasyon işlemlerinin optimize edilmesi ve farklı dizi uzunluklarına sahip veri setlerinde daha iyi performans elde etmek için dikkatlice ayarlanması önerilmektedir. Ayrıca, çalışma süresini azaltmak için paralel programlama kullanılması da göz önünde bulundurulmalıdır.

Sonuç olarak, bu çalışma biyoinformatik alanında otomatikleştirilmiş ve etkili çoklu dizi hizalama yöntemleri üzerine odaklanmıştır. Önerilen Clustal-SA algoritması, genetik veri analizi ve karşılaştırmalarında daha hassas sonuçlar elde etme potansiyeline sahiptir.

TEŞEKKÜR

Bu çalışma Ege Üniversitesi BAP koordinatörlüğü tarafından FOA-2020-20981 kodlu “Cicer ve Lens türlerinin kloroplast DNA dizilerinin Next Generation Sequencing ile sekanslanması ve genom organizasyonlarının belirlenerek karşılaştırmalı genom analizlerinin gerçekleştirilmesi” isimli proje kapsamında desteklenmiştir.

Çıkar Çatışması

Yazarlar herhangi bir çıkar çatışmasının olmadığını beyan eder.

Yazar Katkısı

Araştırma ve makale için fikir ya da hipotezin oluşmasında Hatice Erdirik ve Hasan Bulut katkı sağlamıştır. Sonuçlara ulaşmak için yöntemleri araştıran ve planlayan Hatice Erdirik’tir. Proje ve makalenin organizasyonu ve seyrinin gözetimi ve sorumluluğu Abdullah Ammar Karcıoğlu’ndadır. Veri toplama adımında Muhammed Bahattin Tanyolaç katkı sağmıştır. Verilerin işlenmesi ve analizi konusunda Abdullah Ammar Karcıoğlu katkı sağlamıştır. Çalışma sonucu elde edilen sonuçların analizi ve yorumlanması kısmında Hatice Erdirik, Abdullah Ammar Karcıoğlu ve Hasan Bulut katkı sağlamıştır.

KAYNAKLAR

Aarts, E. H., & van Laarhoven, P. J. (1987). Simulated annealing: a pedestrian review of the theory and some applications. In Pattern recognition theory and applications (pp. 179-192). Springer Berlin Heidelberg. Doi: 10.1007/978-3-642-83069-3_15

Aktan, M. N., & Bulut, H. (2022). Metaheuristic task scheduling algorithms for cloud computing environments. Concurrency and Computation: Practice and Experience, 34(9), e6513. Doi: 10.1002/cpe.6513

Botta, M., & Negro, G. (2010). Multiple sequence alignment with genetic algorithms. In Computational Intelligence Methods for Bioinformatics and Biostatistics: 6th International Meeting, CIBB 2009, Genoa, Italy, October 15-17, 2009. Springer Berlin Heidelberg. Doi: 10.1007/978-3-642-14571-1_15

Bucak, İ. Ö., & Uslan, V. (2011). Sequence alignment from the perspective of stochastic optimization: a survey. Turkish Journal of Electrical Engineering and Computer Sciences, 19(1), 157-173. Doi: 10.3906/elk-1002-410

Chen J.T, Chao J.N, Liu H., Yang F.L., Zou Q. & Tang F.R, (2023) WMSA 2: a multiple DNA/RNA Sequence alignment tool implemented with accurate progressive mode and a fast win-win mode combining the center star and progressive strategies, Briefings in Bioinformatics, Volume: 24, Issue:4, Doi :10.1093/bib/bbad190

Cohen, J. (2004). Bioinformatics—an introduction for computer scientists. ACM Computing Surveys (CSUR), 36(2), 122-158. Doi: 10.1145/1031120.1031122

Diamantis, S., & Anna, C. (2005). Comparison of multiple sequence alignment programs. National and Kapodistrian university of Athens.

Doğan, H., & Otu, H., (2014) Multiple Sequence Alignment Methods: Objevtive Function, Springer Protocols, Chepter 3, 44-85p

Edgar, R.C., (2004), MUSCLE: multiple sequence alignment with high accuracy and high throughput, Nucleic Acids Research, 200, vol.32, No.5, DOI: 10.1093/nar/gkh340

Edgar, R.C., Batzoglou, S., (2006) Multiple sequence alignmnet, Current Opinion in Structural Biology 2006, 16:368–373, Elsevier

Haque, W., Aravind, A., & Reddy, B. (2009, March). Pairwise sequence alignment algorithms: a survey. In Proceedings of the 2009 conference on Information Science, Technology and Applications (pp. 96-103). Doi: 10.1145/1551950.1551980

Karcıoğlu, A. A., & Bulut, H. (2022). DNA sekansları için q-gram hash karşılaştırmasına dayalı çoklu kesin dizi eşleştirme algoritması. Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 38(2), 875-888. Doi: 10.17341/gazimmfd.951157

Karcioglu, A. A., & Bulut, H. (2021). Improving hash-q exact string matching algorithm with perfect hashing for DNA sequences. Computers in Biology and Medicine, 131, 104292. Doi: 10.1016/j.compbiomed.2021.104292

Karcioglu, A. A., & Bulut, H. (2021). The WM-q multiple exact string matching algorithm for DNA sequences. Computers in Biology and Medicine, 136, 104656. Doi: 10.1016/j.compbiomed.2021.104656

Karcioglu, A. A., & Bulut, H. (2022). q-frame hash comparison based exact string matching algorithms for DNA sequences. Concurrency and Computation: Practice and Experience, 34(9), e6505. Doi: 10.1002/cpe.6505

Lee, Z. J., Su, S. F., Chuang, C. C., & Liu, K. H. (2008). Genetic algorithm with ant colony optimization (GA-ACO) for multiple sequence alignment. Applied Soft Computing, 8(1), 55-78. Doi: 10.1016/j.asoc.2006.10.012

Likic, V. (2008). The Needleman-Wunsch algorithm for sequence alignment. Lecture given at the 7th Melbourne Bioinformatics Course, Bi021 Molecular Science and Biotechnology Institute, University of Melbourne, 1-46.

Luscombe, N. M., Greenbaum, D., & Gerstein, M. (2001). What is bioinformatics? An introduction and overview. Yearbook of medical informatics, 10(01), 83-100. Doi: 10.1055/s-0038-1638103

Major Differences, Difference between Global and Local Sequence Alignment, https://www.majordifferences.com/2016/05/differencebetween-global-and-local.html, Access Date: 28.12.2022.

Mirjalili, S. (2019). Evolutionary algorithms and neural networks. In Studies in computational intelligence (Vol. 780). Berlin/Heidelberg, Germany: Springer.

Omar, M.F., Salam, R.A., Abdullah, R. & Rashid, N.A (2004). Multiple Sequence Alignment Using Optimization Algorithms, International Journal of Computational Intelligence Volume 1 Number 2

Pais, F. S. M., Ruy, P. D. C., Oliveira, G., & Coimbra, R. S. (2014). Assessing the efficiency of multiple sequence alignment programs. Algorithms for molecular biology, 9, 1-8. Doi: 10.1186/1748-7188-9-4

Paruchuri T., Kancharla G.R. & Dara S. (2023). Solving multiple sequence alignment problems by using a swarm intelligent optimization based approach, International Journal of Electrical and Computer Engineering (IJECE), Vol. 13, No. 1, February 2023, pp. 1097-1104

Russel and Norvig. (2010). Artificial intelligence: a modern approach, Global Edition. Harlow, Essex, England: Pearson Educatio.

Smith, T. F., & Waterman, M. S. (1981). Identification of common molecular subsequences. Journal of molecular biology, 147(1), 195-197.