VERİ BİLİMİ DERGİSİ
www.dergipark.gov.tr/veri

DNA Dizilerinin Graf Benzetim Yolu İle Karşılaştırılması

Cantekin ÇELİKHASI[1]*, Adem ULU[2], Ahmet SAYAR[2]

[1]*Kocaeli Üniversitesi, Teknoloji Fakültesi, Bilişim Sistemleri Mühendisliği, Kocaeli*
[2]*Kocaeli Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği, Kocaeli*

Özet

Bu çalışmanın amacı farklı DNA örneklerindeki dizilimlerin benzerlik oranını hesaplamaktır. DNA verilerinin büyüklüklerinin getirdiği işlem hacmini kısaltmak ve performansı arttırmak için,DNA dizileri sıkıştırılıp motif çıkarımı yapıldı. Bunun için LZ Sıkıştırma algoritması kullanıldı. Elde edilen motiflerden her biri sadece bir graf düğümünü ifade ederken bu motiflerin sırası da düğümlerin komşuluklarını ifade edecek şekilde grafa dönüştürüldü. Her bir DNA'dan elde edilen graf, DNA'lardaki benzerliği bulabilmek için kullanıldı. Graflar üzerinden benzerlik oranını hesaplayan algoritmalardan, her iki Komşu Eşleşme ve Kosinüs Benzerliği metotlarını karşılaştırdık. Buna göre Kosinüs Benzerliği'nin Komşu Eşleşme'ye göre daha performanslı olduğunu gösterdik.

Anahtar Kelimeler: *LZ Sıkıştırma, Komşu Eşleşme, Kosinüs Benzerliği, Graf Benzerliği, DNA örüntü keşfi*

Makale Bilgisi
Başvuru:
27/11/2018
Kabul:
24/12/2018

Comparison of DNA Patterns with Graf Simulation

Abstract

The aim of this study is to calculate the similarity ratio of the sequences in different DNA samples. In order to increase the performance and shorten the transaction volume of the DNA data, DNA sequences were compressed and DNA motifs were obtained. For this, the LZ Compression algorithm was used. While each of the obtained motifs was only a graph node, the order of these motifs was transformed into graphs expressing the edge of the nodes. The obtained graph from each DNA was used to find the similarity in DNA. According to the algorithms calculating the similarity ratio on the graphs, we compared both Neighbor Matching and Cosinus Similarity methods and then we showed that Cosinus Similarity is more efficient than Neighbor Matching.

Keywords: *LZ Compression, Neighbor Matching, Cosinus Similarity, Graph Similarity, DNA pattern exploration*

* İletişim e-posta: cantek41@gmail.com

1 Giriş

Genomlardan okunan nükleotid parçalarını, okunarak birleştirilmesi ile elde edilen DNA dizilimlerini, algoritmalar kullanılarak anlamlı motiflere çevrilmesi yaygın bir yöntemdir [1].

Farklı DNA'lar arasındaki ilişkiyi kurmak çalışmamızın ana konusunu oluşturmaktadır. DNA dizilimleri genellikle çok uzun olabilmektedir ve bu nedenden, ham DNA dizilimleri üzerinden yapılan çalışmalar performans açısından kötü sonuç vermektedir. Bu durum da araştırmacıları DNA karşılaştırmaları için yöntemler bulmaya yöneltmiştir. Bu çalışma Lampel-Ziv algoritması ile DNA dizilimlerinin graflara çevirme işlemi ile [2] üretilen grafları, benzetim algoritmaları aracılığıyla DNA karşılaştırma işlemlerini kolaylaştırmak hedeflenmiştir.

Uygulama sonucunda elde edilen bulgular, türlerin birbirleri arasındaki benzerlik ilişkisini DNA parçacıkları tespit etmeye çalışan biyologlar, ya da bu alanda çalışan uzmanlara kolaylık sağlayacaktır. Bu çalışmada, veri seti olarak gerçek insan ve şempanze DNA verileri kullanılmıştır [3].

Çalışmanın geri kalanı; 2.bölümde literatür taraması yapılmıştır. 3.bölümde de DNA dizilerinin motiflerinin elde edilebilmesi için Lampel-Ziv algoritmasından bahsedilmiştir. 4. bölümde ise elde edilen motiflerin Komşuluk Eşleştirme ve Cosinus Benzerliği (Komşu Eşleşme ve Kosinüs Benzerliği) algoritmaları ile benzerlik hesaplamaları yapılmış ve son olarak da 5. bölümde elde edilen sonuçlar yorumlanmıştır.

2 Mevcut Çalışmalar

DNA motiflerini keşfetmek için birçok çalışma yapılmış ve bununla ilgili yöntemler geliştirilmiştir. Bu çalışmalardan bazıları aşağıda sunulmuştur.

İlk inceleyeceğimiz yöntem Multiple EM for Motif Elicitation (MEME-Maximum Benzeti ile Motif Çıkarma) Bu algoritma çok bilinen ve başarılı bir motif arama algoritmasıdır. Bu algoritmanın işleyişi kısaca şu şekilde anlatılır: Dizilerdeki yeni, uyuşmayan motifleri (tekrarlanan, sabit uzunlukta desenler) keşfeder. MEME değişken uzunluklu desenleri iki veya daha fazla ayrı motife ayırır [4].

Diğer bir algoritma Gibbs Sampler olarak verilebilir. Bu çalışmada Genom projeleri ve diğer sıralama çabaları tarafından zengin bir protein ve DNA dizisi verileri üretilebilmektedir. DNA dizisi içerisindeki, tekrar eden motifleri bulmak için "yerel çoklu hizalama" probleminin temel alan bir algoritma geliştirilmiştir. Bu algoritma, N-lineer zamandaki N dizileri için optimize edilmiş bir yerel hizalama modelini bulur ve aynı anda birden fazla desen ve model tekrarının eş zamanlı olarak tespit edilmesine ve optimize edilmesine izin verir [5].

Ulaş Baran BALOĞLU tarafından önerilen yöntem de ise; Genetik algoritma kullanılarak motiflerin bir popülasyonun bulunması ve sonrasında, veri madenciliği kullanılarak motiflerin uygunluğunun değerlendirilmesi yapılır. Bu sayede dizi hizalama araçlarını kullanmaya gerek kalmadan sadece veri madenciliği kullanılarak potansiyel motiflerin ve tekrarlı örüntülerin veriden çıkartılması sağlanır [6] .

3 DNA motiflerinin elde edilmesi

DNA dizilerinin benzerliklerini bulabilmek için ilk yapacağımız iş, DNA motiflerinin elde edilmesidir. DNA dizilimlerindeki motifleri elde edip graf çıkarmak çalışmamızı kolaylaştıracaktır. DNA dizilimlerinden motif çıkarmanın ilk adımı:

DNA dizilimleri katar olduğu için, katar üzerinde sıkıştırma işlemi yapan A. Lempel ve J. Ziv tarafından geliştirilen LZ algoritması kullanılmıştır [2]. Bu algoritma, sıkıştırılacak olan verideki karakter katarlarını bir sözlüğe girerek, bu karakter katarı daha sonra tekrar ettiğinde onun sözlük sırasını kodlar [7].

$$V = \bigcup_{i=1}^{n-k} \{S[i - i + k - 2]\} \quad (1)$$

$$E = \bigcup_{i=1}^{n-k} \{S[i - i + k - 2], S[i + 1, i + k - 1]\} \quad (2)$$

k = dizilim uzunluğu +1 E= kenarlar kümesi

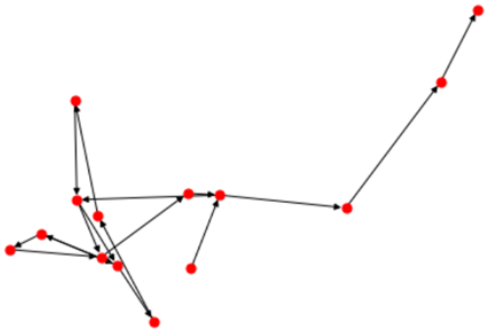
V= düğümler kümesi S=sözlük kümesi

i= indis

```
Pseudo code: LZ Algoritması
1   tmparray = []
2   while True:
3   if i >= n:
4       if tmp:
5           tmparray.append(tmp)
6           break
7       tmp = tmp + katar[i]
8       if not tmp in tmparray:
9          tmparray.append(tmp)
10         tmp = ""
11      i += 1
12  return tmparray
```

Pseudo code: LZ algoritması, bu algoritmanın en kötü durum analizi O(n)'dir.

Örnek: S="gatcacaggtctatcaccctattaa" bu dizilimde LZ algoritması uyguladığında aşağıdaki node ve edge kümelerini bulur ve bu veriler neticesinde Şekil 1 deki grafı oluşturmuş oluruz.



Şekil 1. Örnek DNA diziliminin grafı

Node ve edge (kenar) setleri de örnek DNA için aşağıdaki şekilde elde edilir.

V={'g', 'a', 't', 'c', 'ac', 'ag', 'gt', 'ct', 'at', 'ca', 'cc', 'cta', 'tt', 'aa' }

E={ [('g', 'a'), ('a', 't'), ('t', 'c'), ('c', 'ac'), ('ac', 'ag'), ('ag', 'gt'), ('gt', 'ct'), ('ct', 'at'), ('at', 'ca'), ('ca', 'cc'), ('cc', 'cta'), ('cta', 'tt'), ('tt', 'aa')]}

4 Motiflerin karşılaştırılması

İki farklı DNA diziliminin LZ algoritması yardımı ile üretilmiş graflarına G_A ve G_B olarak isimlendirilir. Elde edilen grafların karşılaştırılması için Komşu Eşleşme ve Kosinüs Benzerliği algoritmaları aşağıdaki şekilde uygulanır.

4.1 Komşu Eşleşme

Bu metot Mladen Nikolic tarafından 2012 yılında geliştirilmiştir. Bu algoritma kısaca: i ∈ G_A ve j ∈ G_B (i ve j birer node); i, j'nin komşu düğümleri ile eşleştirile biliniyorsa benzer olduğu sonucuna varmaktadır.

Indegree (Gelinen Komşular derecesi - id()) hesaplaması aşağıda gösterilmiştir.

$$s_{in}^{k+1}(i,j) \leftarrow \frac{1}{m_{in}} \sum_{l=1}^{n_{in}} x^k \qquad (3)$$

$$m_{in} = max\big(id(i), id(j)\big) \quad n_{in} = min\big(id(i), id(j)\big) \qquad (4)$$

S_{in}(i,j) indegree benzerliği hesaplanırken (1), nodeların gelinen komşu benzerliklerinin toplamının, min indegree ye oranın alınır.

Outdegree (Gidilen Komşular derecesi - od()) hesaplaması aşağıda gösterilmiştir.

$$s_{out}^{k+1}(i,j) \leftarrow \frac{1}{m_{out}} \sum_{l=1}^{n_{out}} x^k \qquad (5)$$

$$m_{out} = max\big(od(i), od(j)\big) \quad n_{out} = min\big(od(i), od(j)\big) \qquad (6)$$

S_{out}(i,j) outdegree benzerliği hesaplanırken (5), nodeların gidilen komşu benzerliklerinin toplamının, min outdegree ye oranın alınır.

Farklı iki node un benzerliğinin hesaplanması aşağıdaki şekilde sunulmuştur.

$$x_{(i,j)}^{k+1} \leftarrow \frac{s_{in}^{k+1}(i,j) + s_{out}^{k+1}(i,j)}{2} \qquad (7)$$

Burada her nodelerin benzerlik hesabının yapılabilmesi için; S_{in}(i,j) Indegree (gelinen komşular) komşuların toplamının toplam indegree'ye oranı ve aynı şekilde S_{out}(i,j) outdegree (gidilen komşular) için de hesap yapılarak elde edilen sonucun ortalaması ile x(i,j) nodeların benzerliğini hesaplamış oluruz [8] .

Pseudo code: Komşu Eşleşme Algoritması

```
1   for i in self.graph1.nodes():
2       for j in self.graph2.nodes():
3           sin(i,j)=FnNodesMatch(id(i), id(j))
4           sout(i,j)= FnNodesMatch (od(i),od(j))
5           X(i,j)= mean(sin(i,j) , sout(i,j))
6   FnNodesMatch(ii,jj):
7           for i in ii:
8               for j in jj:
9                   x[i,j]=benzelik_fn(i,j)
10  return sum(x(i,j))return tmparray
```

Komşu Eşleşme Bu algoritmaya bakarak en kötü durum analizini yapmak istediğimizde:

FnNodesMatch() kod parçacığını $(n-1)^2$ den $O(n^2)$, onu çağıran ana kod ise n^2 olarak çalıştığı için $O(n^4)$ olarak hesaplanır (matrislerin boyut i,j olsun n =i+j olur).

4.2 Kosinüs Benzerliği

İki vektör arasındaki açı ve bu vektörlerin iç çarpımının çarpılmasıyla hesaplanan uzaklık; G_A ve G_B arasındaki kosinüs benzerliği olarak formüle edilir ve aşağıdaki gibidir [9],[10].

$$sim_{cos}(i,j) = \frac{dot(i,j)}{\|i\|\|j\|} = \frac{\sum_{k=1}^{n} w_{ik} \times w_{jk}}{\sqrt{\sum_{k=1}^{n} w_{ik}^2 \times \sum_{k=1}^{n} w_{jk}^2}} \qquad (8)$$

```
Pseudo code: Cosinus Benzerliği Algoritması
1    for i,j in A,B:
2        tmp[ii] ← i.dot(j) / (norm(i) * norm(j))
3    return np.mean(tmp)
```

Cosinus Benzerliği, bu algoritmaya bakarak karmaşıklığını O(n) olarak görmekteyiz.

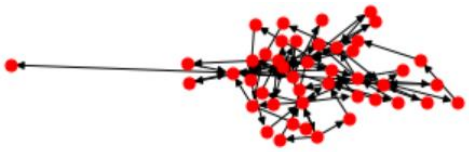
5 Deney ve Ölçümler

Önerdiğimiz algoritmanın örnek DNA verileri üzerinde çalıştığını göstermek için aşağıda verilen H (insan) ve C (şempanze) verilerini kullandık.

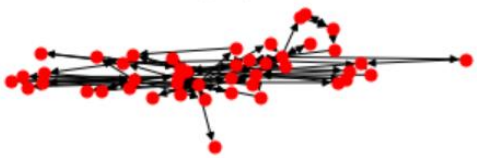
H=gatcacaggtctatcaccctattaaccactcacgggagctctccatgcatttggtattttcgtctgggggtatgcacgcgatagcattgcgagacgctggagccggagcaccctatgtc [11]

C=gtttatgtagcttacccccttaaagcaatacactgaaaatgtttcgacgggtttatatcaccccataaacaaacaggtttggtcctagcctttctattagctcttagtaagattacacat [3]

Şekil 2 de H-DNA parçacığının, Şekil 3'de de C-DNA parçacığının graflar dizilimi verilmiştir.



Şekil 2. Örnek İnsan DNA diziliminin grafı



Şekil 3. Örnek Şempanze DNA diziliminin grafı

Hx=gatcacaggtctatcaccctattaaccactcacgggagctctccatgcatttg

Cx=gtttatgtagcttacccccttaaagcaatacactgaaaatgtttcgacgggtt

Sentetik veri olarak da elimizdeki reel dataların 53 karakterlik parçalarını kullanarak Hx ve Cx dizlimlerini elde ettik.

DNA parçacığı için öncelikle LZ algoritması ile graflara dönüştürdük, sonra elde edilen grafları iki algoritmayı kullanarak benzerlik oranların hesaplayarak aşağıdaki sonuçları elde ettik.

Tablo 1. Algoritma ve Benzerlik oranları karşılaştırılması

Diziler	Algoritma	O	Benzerlik Oranı	Diziler
H,C	NB	n^4	0.005097313	H,C
H,C	Cos	n	0.185079555	H,C
C,C	NB	n^4	0.922945978	C,C
C,C	Cos	n	1.0	C,C
H,Hx	NB	n^4	0.000269661	H,Hx
H,Hx	Cos	n	0.827119610	H,Hx
C,Cx	NB	n^4	6.84376e-05	C,Cx
C,Cx	Cos	n	0.916209019	C,Cx

6 Sonuç ve Yorumlar

H (insan) DNA örneği, C (şempanze) DNA örneği sonuçları Tablo 1 de görülmektedir ki, farklı DNA örnekleri üzerinden Komşu Eşleşme düğümlerin benzer komşuluk bağıntısı üzerinden hesaplama yaptığı için daha düşük bir oran vermektedir.

Kosinüs Benzerliği ise grafların komşuluk matrisleri üzerinden hesaplama yaptığın için daha yüksek bir oran vermektedir.

Veriler neticesinde, Komşu Eşleşme ve Kosinüs Benzerliği algoritmaları arasında büyük bir fark elde ettik. Ne var ki Cosinus benzerliğinin karmaşası daha az olduğunda daha uzun dizilimlerde daha performanslı ve doğru çalışacaktır.

Yapılacak daha detaylı çalışmalar neticesinde elde edilen başarım ile belli başlı hastalık ve mutasyonları motifler halinde tespit edip hastalık şüphesi olan DNA dizilimleri içinde benzetim oranı alınarak bir yoruma varılabilmektedir.

Kaynaklar

[1] Shamir R, *Bioinformatics for Biologists*, Cambridge University Press. California, USA, 2014.

[2] Sülü M, "Graf Tabanlı Biyolojik Dizilerde Örüntü Keşfi", *Elektrik-Elektronik Bilgisayar Sempozyumu*. Elazığ, 2011.

[3] GenBank, "Pan paniscus mitochondrial DNA, complete sequence". https://www.ncbi.nlm.nih.gov/nuccore/D38116 (26.7.2016).

[4] Elkan C, Bailey TL ,"Fitting a mixture model by expectation maximization to discover motifs in biopolymers", *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, pp. 28-36, 1994.

[5] Lawrence CE, Altschul SF, Boguski MS, Liu JS, Neuwald AF, Wootton JC, "Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment", *Science*, p. 262, 1993.

[6] Baloğlu UB, "DNA Sıralarındaki Tekrarlı Örüntülerin Ve Potansiyel Motiflerin Veri Madenciliği Yöntemiyle Çıkarılması", *Fırat Üniversitesi*, Elazığ, Türkiye, 2006.
[7] Lempel A, Ziv J, "On the Complexity of Finite Sequences", *IEEE Transactions on Information Theory*, p. 22, 1976.
[8] Nikolic M, "Measuring Similarity of Graph Nodes by Komşu Eşleşme", *IOS Press*, pp. 865-878, 2012.
[9] M Gallo, "Implementing and Understanding Cosine Similarity". https://masongallo.github.io/machine/learning,/python/2016/07/29/cosine-similarity.html (29.07.2016).
[10] Madylova A, "Kosinüs Benzerliğini Kullanarak Belgeler Arası Anlamsal Benzerliği Kavramsal Sözlüğe Dayalı Hesaplama Yöntemi", İTU, İstanbul, Türkiye 2009.
[11] Anderson B, "H.Sapiens mitochondrial genome", https://www.ncbi.nlm.nih.gov/nuccore/V00662. (14.7.2016).