DÜMF Mühendislik Dergisi

web: http://dergipark.gov.tr/dumf

Araştırma Makalesi / Research Article

Meta-Genomik Gen Analizi için Filtre Tasarımı

İbrahim SAVRAN*
Karadeniz Teknik Üniversitesi, Bilgisayar Mühendisliği Bölümü, Trabzon
savran@ktu.edu.tr ORCID:, 0000-0002-4155-0485 Tel: (462) 377 4346

Esra ERDEN
Karadeniz Teknik Üniversitesi, Bilgisayar Mühendisliği Bölümü, Trabzon
esraeerden@gmail.com ORCID: 0000-0003-3204-7535

Geliş: 13.12.2018, Revizyon: 27.05.2019, Kabul Tarihi: 28.05.2019

Öz

Bazı virüsler hariç tüm canlılarda kalıtımın temel yapısını nükleik asitler oluşturmaktadır. Bilim insanları kalıtım üzerine çeşitli çalışmalar yapmıştır. Yapılan araştırmalar ve deneyler sonucunda bilginin depolanması, bilgiye ulaşma ve bilginin analizinin rahat bir şekilde yapılmasına gerek duyulmuştur. Bunun için biyolojinin bilgisayarlar üzerinde kullanımı 1960'lı yıllarda başlamıştır. Sonrasında bilgisayar bilimleri geliştikçe biyolojinin gelişmesi ve araştırma alanları da artmıştır. Gelişmeler arttıkça biyolojinin büyük çaplı deneyler, makro molekül yapıları, DNA ve RNA gibi alanlarda araştırmaları daha çok olmaya başlamıştır. Son 50 yıl boyunca, hastalıklardaki artış bilim insanlarını DNA dizilimleri üzerinde çalışma yapmaya yönlendirmiştir. 1970'lerin başlarında ilk DNA dizileri üniversite araştırmacıları tarafından bulunmuştur. Daha sonralardan Sanger ve arkadaşları geliştirdikleri dizileme yöntemi ile insan genomunu dizilemeyi başarınca Yeni Nesil Dizileme kavramı ortaya çıkmıştır. Yeni Nesil Dizileme yöntemi kullanılarak çok büyük boyuttaki gen sekanslarına dizileme yapılmıştır. Ancak gen sekanlarının uzunlukları çok büyük olduğu için dizileme yapılırken hatalar ile karşılaşılır. Ayrıca ideal DNA dizileme daha hızlı ve kolay çözüme ulaşmalıdır. Bu çalışmada gen sekanslamadaki hataları azaltma üzerinde çalışılmıştır. Bir filtre yardımı ile büyük boyuttaki gen sekanlarını benzerlik açısına göre birkaç kümeye ayırılmıştır. Ayrılan bu dizilerde doğru sıralamalar elde edilmeye çalışılmıştır.

Anahtar Kelimeler: *DNA dizileme; Biyoinformatik; Yeni nesil dizileme; Meta-genomik gen; Sonek ağacı; Filtre,*

* Yazışmaların yapılacağı yazar

DOI: 10.24012/dumf.496688

Giriş

Canlılarda nesilden nesile geçen bilgileri içeren ve canlının birçok özelliğini belirleyen kalıtımı sağlayan temel yapı taşı nükleik asitlerdir. Nükleik asitler bir fosfat, bir deoksiriboz veya riboz bir de azot içeren bir kimyasal bileşiktir. Bir DNA genomu üzerindeki Adenin, Guanin, Timin ve Sitozin bazlarının sıralarının belirlenme işlemine *dna dizileme* denilmektedir. DNA parçalarına ait bu nükleotid sıralarının belirlenmesi; DNA bölgesinin hangi proteini kodlayabileceğine ilişkin bilgilerin elde edilmesinde, genomik DNA dizisi ile tamamlayıcı DNA'ya ait dizi bilgilerinin karşılaştırılarak ekson ve intronların ortaya çıkarılmasında, DNA dizi analizi ile gen aktivitesini kontrol eden bölgelerin tanımlanmasında, spesifik DNA dizilerinin belirlenmesi ile evrimsel akrabalık ilişkilerinin tanımlanmasında kullanılmaktadır (Türktaş, 2011). DNA dizileme biyolojik işlemleri ve araştırmaları çok hızlandırmıştır. Öyle ki 3 milyar DNA baz çiftine sahip olan insan genomu bile DNA dizileme yöntemlerinin gelişmesi ile dizilenmiştir.

1866 yılında, Mendel'in bezelye üzerinde melezleme çalışmaları yapmasıyla kalıtımın ilkelerini ortaya çıkarılmıştır. 1944 yılında Oswald T. Avery ve arkadaşları DNA'nın genetik bilgiyi aktardığına dair ilk kanıtı bakterilerle yaptığı çalışmalardan elde etmişlerdir. Çalışmada zatürre hastalığına neden olan bakteriler kullanılmış-tır. Canlı kapsülsüz bakteriler fareye enjekte edildiğinde fare zatürreye yakalanmamış ve yaşama-ya devam etmiştir (Kushner ve Samols,2011). 1953 yılında Watson ve Crick DNA'nın çift sarmal yapısını keşfetmişlerdir ve diğer benzer araştırmalar nükleik asit dizileme sistemlerinin kökenini oluşturmuştur (Pray, 2008).

İlk zamanlarda dizileme çalışmaları çok uzun sürmekte ve çok zahmetli bir iş olmuştur. RNA dizilemesi, baz dizilemesinin ilk basamağı olmuştur. İlk DNA dizileri 1970'lerin başlarında üniversite araştırmacıları tarafından iki-boyutlu kromatografiye dayanan zahmetli yöntemlerle elde edilmiştir. Otomatik analizle çalışan boyatabanlı dizileme yöntemlerinin gelişimiyle (Olsvik vd., 1993) DNA dizilemesi çok daha kolaylaşmış ve birkaç büyüklük mertebesi hızlanmıştır (Petterson vd.,2009).

1973 yılında, zahmetli bir yöntem olan Wandering-Spot Analiz yöntemi kullanılarak 24 bazın dizisi yayınlanmıştır. Ancak 1975 yılında, Frederic Sanger ve arkadaşlarının *Zincir sonlandırma yöntemi* ya *da Sanger dizileme yöntemi* geliştirilmesi ile daha güvenilir, daha kolay ve daha hızlı olmasından dolayı bu yöntem daha çok kullanılmaya başlanmıştır (Sanger vd., 1977). Floresan boya ile işaretli dideoksi nükleotid trifosfatlar (ddNTP)'rın kullanıldığı Sanger yöntemiyle çok sayıda örnek aynı anda dizilenmektedir. Her çalıştırılışında 400-800 arası baz uzunluğuna sahip DNA dizileri yüksek doğrulukla okunabilmektedir. Bu yöntem günümüze kadar en çok kullanılan DNA dizileme yöntemi olmuştur (Bentley vd., 2008), (Stein, 2004). Sanger dizileme (Zincir sonlandırma) ve floresan tabanlı elektroforez teknolojileri kullanılarak insan DNA dizisinin büyük çoğunluğu tanımlanmıştır.

1986 yılında ilk yarı otomatik DNA dizileme makinesini bulunmuştur ve bilgisayarların kullanılmaya başlanmasıyla günümüzde DNA dizilemesi son derece hızlı bir şekilde gerçekleştirilmiştir.

1990 yılında çeşitli kuruluşların, sağlık örgütlerinin ve 16 ülkenin katılımıyla İnsan Genom Projesi'ne resmi olarak başlanmıştır (Bentley vd., 2008). İnsan genomu projesi ile insan haploit genomuna ait 3,3 milyar nükleotit baz dizisinin belirlenmesi ile genomdaki mevcut bütün genlerin tespit edilmesi amaçlanmıştır. Çalışmanın ilk yıllarında insan genomuyla ilgili büyük ilerlemeler kaydedilmiştir. İnsan genom projesi DNA dizilemenin daha da gelişmesini sağlamıştır. Proje kapsamında çalışılan ve ilk tamamlanan insan genom dizilemesi, 10 yıllık bir süre sonunda yaklaşık 3 milyar dolarlık bir maliyetle tamamlanmıştır (Stein, 2004). İnsan genom projesinin bitirilmesi ile birlikte yeni nesil dizileme olarak adlandırılan masif paralel dizileme yöntemleri geliştirilmeye başlanmıştır. Yeni Nesil Dizileme yönteminde kullanılan ilk Next generation sequencing (NGS) cihazı ise

2005 yılında kullanıma sunulmuştur (Margulies vd., 2005).
Sanger Dizileme Yöntemi insan genomu projesi, belirli hayvan ve bitki genomlarını başarılı bir şekilde tamamlanmasında kullanılmıştır (Ulutin, 2005). İnsan genomu projesinde karşılaşılan zorluklardan dolayı daha hızlı, daha ucuz, daha doğru sonuçlar üretebilen bir DNA dizileme yöntemi olan Yeni Nesil Dizileme (YND) bulunmuştur. Sanger dizileme tekniği ile genom dizileme projeleri uzun zaman alır iken günümüz dizileme yaklaşımları ile kısa sürede (bir hafta gibi bir süre) tamamlanabilmektedir. Bu yöntemle elde edilen bir mikrobiyal genom dizisi araştırmacılara başka hiçbir deneysel yöntem ile elde edilemeyecek kadar zengin ve özgün bilgi sağlamaktadır. Örneğin 4.6 Mb'lık E.coli genomu tek bir okuma ile tamamlanabilmektedir. Yapılan bir çalışmada E.coli genomu dört kere, her bir koşmada 400.000 okuma yapılarak de novo dizilenmiş ve sekanslamalann %99.997 ila %99.999 arasında doğrulukla yapıldığı tespit edilmiştir (Margulies vd., 2005).

Dizileme çalışmalarından elde edilen bilgiler, biyoloji ile bilgisayarın birlikte kullanımından meydana gelmiş olan biyoinformatik alanının ortaya çıkmasını sağlamıştır. Biyoinformatiğin amaçlarından ilki verileri, araştırmacıların kolaylıkla ulaşabileceği şekilde düzenlemek ve yeni veriler üretildikçe hızlı bir şekilde kaydetmektir. Biyoinformatiğin bir diğer amacı verilerin anlamlı duruma gelmesini sağlayan araçlar ve kaynaklar geliştirmektir.

Mikroçiplerdeki ilerlemeler sayesinde milyonlarca nükleotite sahip genomlar arasında benzerlik ilişkileri kurulabilmiştir ve bu genomların karakterizasyonu yapılabilmiştir. 5.368 baz çifti büyüklüğündeki bakteriyofaj fx174 genomu sekanslanan ilk viral genomdur (Sanger, 1977).

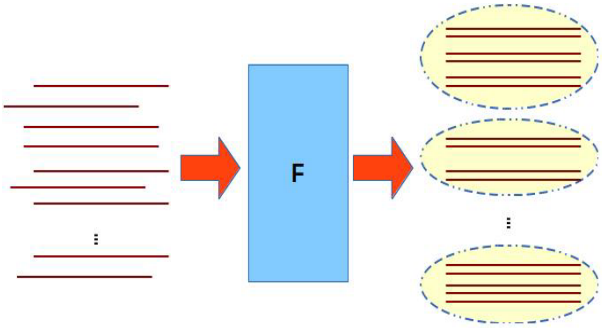
Yeni nesil dizileme yönteminin çok fazla olumlu yanı olmasına rağmen büyük boyuttaki verilerin analizleri, değerlendirmesi ve depolanmasında sorunlar ortaya çıkmıştır (Üstek, 2011). Sorunların çözümlenerek yeni nesil dizileme yönteminin başarılı sonuçlar üretmesi için gelişmiş biyoenformatik araçlarına ihtiyaç duyulmuştur. Bu araçlar canlının DNA dizilimlerini okuyup analiz eder. Gen sekansları çok büyük uzunlukta olmalarından kaynaklı olarak hesaplama açısından pahalıdır, DNA dizilimlerini okumak zordur ve okurken çeşitli hatalar ortaya çıkmaktadır.

Bu çalışmada hata durumlarının olmaması için bir filtre modülü tasarımı gerçekleştirilmiştir. Filtre modülü ile çok büyük uzunluktaki veriler, benzerliklerine göre kabaca gruplandırılmıştır. Daha sonra bu gruplar arasında hizalama aşaması devreye girecektir. Kısacası filtre modülü yeni nesil dizilemedeki verileri belirli bir eşiğe göre kabaca gruplamaktır.

Materyal ve Yöntem

Filtre Tasarımı

Filtre, meta-genom gen dizilerinin benzerliklerine göre kabaca gruplanmasını sağlayan bir modüldür. Meta-genom gen dizisi öncelikle filtre yardımı ile gruplanır, sonrasında hizalama modülüne gönderilir.



Şekil 1. Filtre modülü.

Sonek ağacı ve PaCE filtresi

Sonek ağacı bir dizideki verilere erişmek için sonek dizisinin kullanan, bir kümeleme aracıdır. Bir kökten başlayıp yapraklara ayrılarak diziyi kümelere ayırmaktadır. PaCE filtresi ise dizideki verileri kümeleme yaparken sonek ağaçlarını kullanan bir yazılımdır.

a) Sonek ağacı

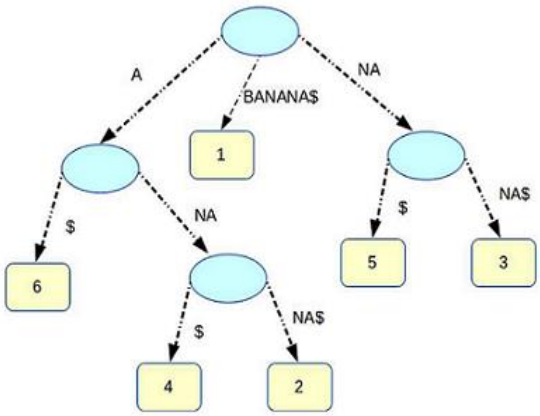
Bilgisayar bilimlerinde en yüksek erişim hızını sağlayan veri yapıları ağaçlardır ve ağaçlar bu özelliklerini hiyerarşik yapılarına borçludur (Cormen, 1989). Sonek ağacında bir tane kök düğüm bulunmaktadır ve tüm arama işlemleri kök düğümden başlamaktadır. Kök düğümden yaprak düğüme giden yolda karşılaşacağımız alt düğümlerin birleşimi bir soneki temsil etmektedir. Hangi boyutta olursa olsun, tüm sonekler için ağaçta bir yol oluşturulur.

Sonek, bir dizide bir karakterden başlayıp dizinin sonuna kadar devam eden bir alt dizidir. Sonekler bir ağaca öyle şekilde yerleştirilirler ki birbirine benzer karakterler ile başlayan iki sonek, sonek ağacı boyunca aynı yolu izlerler. Yol, kök düğümünden başlar ve sonekler arasında bir farklılık oluşuncaya kadar aşağı doğru ilerler. Farklılaşmanın başladığı noktadan itibaren soneklerin her biri ayrı yol izlerler (İnan, 2003).

Biyoinformatikte de sıklıkla kullanılan sonek ağacı kelime işleme algoritmalarından biridir. DNA dizileri çok büyük uzunlukta olduklarından analizlerinin elle yapılması mümkün değildir. Sonek ağacı kullanılarak hızlı bir şekilde dizilerin birbirleri ile uyumlu olup olmadığı kontrol edilmektedir.

n uzunluğunda bir S stringinin sonek ağacının özellikleri aşağıdaki gibidir.

- Köke sahip bir ağaçtır ve yönlüdür.
- 1-n arasında yaprağa sahiptir.
- Kök olmayan her ara düğüm en az 2 yaprağa sahiptir.
- Bir düğümden çıkan kenarlar farklı karakterler ile başlar.



Şekil 2. "BANANA" dizesi için sonek ağacı örneği.

Manber ve Myers, 1990 yılında sonek ağacının verilerini dizilemek için bir algoritma önermiştir (Manber ve Myers, 1990). Bu algoritmada en uzun ortak önek (LCP) dizisi ve O($\boldsymbol{n \log n}$) karmaşıklığı hesaplanmıştır. LCP dizisi, sıralı sonek dizisindeki ardışık eklerden en uzun öneklerin uzunluklarını tutmaktadır. Sonek dizileri genellikle LCP dizilerine ihtiyaç duymaktadır ve sonek dizileri ile sonek ağaçları arasında ilişkisel olarak bir bağlantı bulunmaktadır. Sonek dizisi hesaplandıktan sonra LCP dizisi, en uzun ortak önekleri belirlemek için sözcüksel olarak ardışık ekleri karşılaştırarak oluşturulur.

Tablo 1. "banana" sonek dizisi ve LCP dizisi.

i	Sonek	Sonek	A(i)	LCP(i)
0	banana$	$	6	
1	anana$	a$	5	0
2	nana$	ana$	3	1
3	ana$	anana$	1	3
4	na$	banana$	0	0
5	a$	na$	4	0
6	$	nana$	2	2

Sonek ağacı, en yüksek erişim hızını sağladığı için büyük bir öneme sahip olmaktadır. Sonek ağacının önemli avantajları olmasına rağmen bir diğer yandan dezavantajları da bulunmaktadır. Ağacın fazla yer kaplaması, kötü bellek yerleşimi ve ağacın dengesiz yapısı dezavantajları oluşturmaktadır. Bu gibi sebeplerden ötürü ağacın oluşturulma süreci çok uzun sürmektedir.

b) PaCE filtresi

PaCE filtresi, gen dizilerini kümeleme yapmak için sonek ağacını kullanan, açık kaynaklı ve MPI tabanlı olan bir yazılımdır. Büyük ölçekli verilerinin hızlı bir şekilde kümelenmesini sağlamak için, paralel bilgisayarlarda verilerin kümelenmesi için bir yazılım programı olan PaCE filtresi kullanılmaktadır (Kalyanaraman, 2003). Aynı zamanda hızından dolayı, farklı parametrelerle çoklu çalışmaların yapılmasını sağlamaktadır ve biyologlara gen sekans verilerini daha iyi analiz etmek için bir araç sağlamaktadır.

Skew Algoritması, Ko ve Aluru'nun (KA) Algoritması

Hem skew algoritması hem Ko-Aluru'nun algoritması sonekler üzerinde çalışmaktadır. Büyük boyuttaki sonek dizilerini alt dizilere bölerek işlem yapmaktadırlar.

a) Skew algoritması

2003 yılında Karkkainen and Sanders (Kärkkäinen ve Sanders, 2003) sonek ağaçlarının yapısını kullanarak bir string için bir sonek dizisi oluşturan optimal bir algoritma oluşturmuşlardır. Bu algoritma s*kew algortiması* olarak adlandırılmıştır. O(n) karmaşıklığına sahiptir ve n tane ögeyi sıralamak için için ise O($n \log n$)) karmaşıklığa sahiptir. Sonek dizisi atomik ögeleri dizilemek için indirgendiğinden algoritma verimli sıralamanın kullanılabildiği her modelde kullanılabilir. Skew algoritması önceki lineer zamanlama algoritmalarından çok daha basittir (Kärkkäinen ve Sanders, 2003].

Algoritma bir tamsayı alfabesine($\sum$) sahiptir ve bu alfabe üzerinde çalışmaktadır. Genel olarak algoritmanın amacı sonekleri bölmektir. "banana" stringine skew algoritması aşamaları uygulanmıştır. Tablo 1' de ve Tablo 2'de skew algoritmasının 1.ve 2. aşamaları sonucunda oluşan alt diziler gösterilmiştir. Tablo 3'te ise 1. ve 2. Tablolarda elde edilen alt dizelerin birleştirilmiş hali gösterilmektedir.

Tablo 2. Sonek dizisi alt dizisi 1.

i	sonek
-	-
5	a$
-	-
1	anana$
-	-
4	na$
2	nana$

Tablo 3. Sonek dizisi alt dizisi 2.

i	sonek
6	$
-	-
3	ana$
-	-
0	banana$
-	-
-	-

Tablo 4. Alt dizileri birleştirme.

i	sonek
6	$
5	a$
3	ana$
1	anana$
0	banana$
4	na$
2	nana$

Algoritma 1: Skew algoritmasının doğrusal zamanlı alfabeler üzerindeki sonek dizisi.

1. *i mod3 = 0* pozisyonundaki kısa soneklerden başlanarak sonekler art arda sıralanır.
2. *i mod3 ≠ 0* pozisyonunda, 1.adımdaki soneklere göre kalan sonekler sıralanır.
3. 1. ve 2. adımdaki sonek dizileri birleştirilir.

$$G^{\neq 0} = \begin{Bmatrix} (1,'anana\$'), (2,'nana\$'), \\ (4,'na\$'), (5,'a\$') \end{Bmatrix}$$

$$G^{=0} = \begin{Bmatrix} (0,'banana\$'), (3,'ana\$'), \\ (6,'\$') \end{Bmatrix}$$

Skew algoritmasının 1. ve 2. Adımları sonucunda oluşan diziler gösterilmektedir. Daha sonrasında bu iki adım sonucunda bulunan diziler birleştirilmektedir.

s	**b**	**a**	**n**	**a**	**n**	**a**	**$**
Type	L	S	L	S	L	L	S/L

Pos	0	1	2	3	4	5	6

bucket	**$**	**a**	**a**	**a**	**b**	**n**	**n**
Step-2	6	5	3	1	0	2	4
Sorted Order	6	5	3	1	0	4	2

b)Ko veAluru'nun Algoritması

Çok yakın zamanda, hem zaman hem de uzay üzerindeki araştırmalar daha verimli ek yapı dizisi algoritmaları (SACA'lar), büyük ölçekli uygulamalar için, web arama ve biyolojik genom veri tabanları gibi sonek dizileri yapılarının artan ihtiyacı nedeniyle giderek daha hızlı bir arayışa dönüşmüştür. Büyük veri kümelerinin büyüklüğü genellikle milyarlarca karakterde ölçülür. Şimdiye kadar elde edilen en son sonuçlar arasında en hızlı lineer SACA algoritması Ko ve Aluru'nun KA algoritmasıdır (Kim vd., 2003).

Ko ve Aluru'nun algoritması komşunun son eki ile bir sonraki sonekin söz dizimsel sıralamasına dayanmaktadır. Algoritma, S ve L tipinde iki adet sonek dizisinin etiketlenmeye başlanması ile başlamaktadır. Sınıflandırma şu şekilde yapılır: eğer $\boldsymbol{suff_i < suff_{i+1}}$ ise $\boldsymbol{suff_i}$ S sınıfında bir sonek, eğer $\boldsymbol{suff_{i+1} < suff_i}$ ise $\boldsymbol{suff_i}$ L sınıfında bir sonektir. En son ek S/L olarak etiketlenir. S tipi soneklerin konumları stringi bir dizi alt dizeye ayırır. Bu alt dizelerin her biri, tüm alt dizeler arasındaki sırayla değiştirir ve yeni bir dize oluştururlar. Yeni dizenin sonekleri daha sonra tekrar sıralanır. Sonek dizisi, tüm *S* tipi soneklerin sözlüksel kurallarını verir. Diğer tüm soneklerinin sırası bu sıradan çıkarılabilir.

Algoritma 2. Ko – Aluru'nun algortiması.

$suff_{n-1} = S/L$
for i = n-2 down to n
if $s_i < s_{i+1}, suff_i$ *S tipinde.*
if $s_i > s_{i+1}, suff_i$ *L tipinde.*
if $s_i = s_{i+1}, suff_i$ $suff_{i+1}$ *tipinde.*
end for

Ko ve Aluru'nun algoritması, 3 özyinelemeli adımdan oluşmaktadır. İlk adımda daha karmaşık kodlama sözcüklerine sahip olan S substringleri daha küçük dizilere indirgenir. Böyleye büyük problem küçük parçalara ayrılmış olur. İkinci adımda küçük dizilere ayrılmış problemlerin sonek dizileri tekrarlı olarak hesaplanır. Son adımda özyinelemeli seviyeye göre indirgenmiş problemin soneklerinin sırası, indirgenmiş problemin sonek dizisine dizi bitene kadar yazılır. Ko-Aluru'nun algoritmasının aşamaları aşağıda detaylıca anlatılmıştır (Ko ve Aluru, 2003).

Oluşan alt dizeler arasında sıralamayı doğru şekilde yapmak için veriler yer değiştirir. B, dizenin tüm son eklerini içeren bir dizi olsun. C, S tipi tüm son eklerin sıralanmış bir dizisi olsun. C'yi kullanarak, tüm soneklerin sırasını aşağıdaki gibi hesaplanmaktadır.

1. Dizinin tüm soneklerini B'deki ilk karakterlere göre gruplandırılmaktadır.
2. C dizisi taranır. Taramada karşılaşılan her sonek için, C dizisinin içindeki geçerli ucuna taşınır ve geçerli ucu bir konum sola doğru ilerletilir. Bu adımdan sonra, tüm tip S sonekleri B'deki doğru konumlarındadır.
3. Eğer her bir B girişi için, eğer $suff_{Bi-1}$ bir L tipi sonek ise, geçerli dizideki biriktirme yeri geçerli ön kısmına getirilir ve biriktirme yeri önünü birer birer ilerletilir. Bu adımın sonunda, B, S'nin sonek dizisidir.

Uygulama ve Başarımlar

Bu çalışmada, filtre tasarımı gerçekleştirilmiştir. Filtre modülünde, t stringi kümülatif bir sonek dizisinin hesaplanması için bir çekirdek görevi uygulamaktadır. t stringi, S kümesinde s_i stringlerin bileşiminden oluşmaktadır. t stringinin sonek dizileri (S,A) oluşturulduktan sonra, girişler filtre çekirdeği için hazır durumdadır. Ancak bunun yanında toplam t stringinin kopyalanması ile bir s_i dizisini eşleştirmesini önlemek için çekirdeğe yönelik ilave bir adım atılmalıdır. Bu ilave durum da Algoritma 3 ile çözülmektedir (Savran, 2014).

Algoritma 3. Filtre algoritması2.

$Input: S = \{s_0, s_1, \dots, s_{n-1}\}$
$t = \varepsilon$
for $s = s_0 : s_{n-1}$ *do*

t = t + s
end for
Compute suffix array of S A
A= FilterKernel(S A, S)
Align(S,A)

Tüm diziler için, dizilerin karşılık gelen kısımlarına uygulanmaktadır. Denklem(1), denklem(2) ve denklem(3) birbiri ile çakışma olasılığı olan gen dizileri gösterilmiştir. Denklem (1)'deki gen dizisi ile denklem (2)'deki gen dizisi arasında "TTCC" bazları, denklem (1)'deki gen dizisi ile denklem (3)'teki gen dizi arasında ise "CAT" bazları çakışmaktadır.

$$s_j = ..TTCCCAT.. \quad (1)$$

$$s_i = ..ACCTTCC \ldots \quad (2)$$

$$s_{i+1} = ..CATTG.. \quad (3)$$

s_j dizesi, hem s_i dizesi hem de s_{i+1} dizesi ile örtüşen "TTCCCA"' alt dizesine sahiptir. Sonuç olarak çakışma durumunun ortadan kalkması için dizilerin verileri arasına "#" sembolü eklenir ve olası bir çatışma olma durumu ortadan kalkar. Denklem (4)'te ise çakışma durumunun ortadan kalması için yapılanlar gösterilmiştir.

$$T=s_0 + \# + s_1 + \# + \cdots \# + s_{n-1} \quad (4)$$

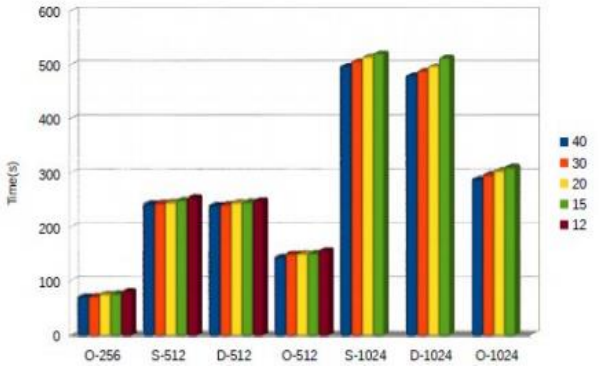
Sonuçlar ve Tartışma

Filtre modülü, hizalama prosedürüne gönderilecek olan gen dizilerini üretmektedir. Filtreden çıkan sonuçlar paralel programlamada kullanılan PaCE filtresinden çıkan sonuçlar ile karşılaştırarak verimlilik derecesi ölçülmektedir. PaCE filtresini seçmemizin nedeni sonek ağacını kullanması ve açık kaynaklı MPI tabanlı bir filtre olmasıdır.

Meta-genom veri seti örnekleri, 13 tane benzer olan Bacillus genus ve 12 tane benzer olmayan Protebacteria Phylum olarak toplamda 25 tam bakteri veri setinden oluşmaktadır. Bu verileri NCBI veri tabanından alınmıştır. NCBI veri tabanı 1988 yılında moleküler biyoloji ve genetik alanında kullanılmak üzere oluşturulmuş halka açık bir veri tabanıdır (NCBI).

Bu veriler, MetaSim cihazıyla yeni nesil dizilemeyi sümüle etmektedir. MetaSim cihazı bir sıralama simülatörüdür (Richter,2008). Verilen genomların bir veri tabanına dayanan MetaSim, farklı seviyelerde bulunan genomların sayısını belirlemektedir ve sonrasında yeni nesil dizileme teknolojisinin bir benzetimini kullanarak bir meta-genom tasarlamaya izin vermektedir.

Filtre modülü üç farklı boyuttaki küme üzerinde test edilmektedir (Savran, 2014). İlki temel kümedir ve sadece bir tane NVIDIA K20 GPU içerir. Burada filtre modülü, minimum eşleşme uzunluğunun azaltılmasından sonra daha fazla zamana ihtiyaç duyar.



Şekil 2. Filtre modülü zamanlaması: 1 NVDIA K20 GPU.

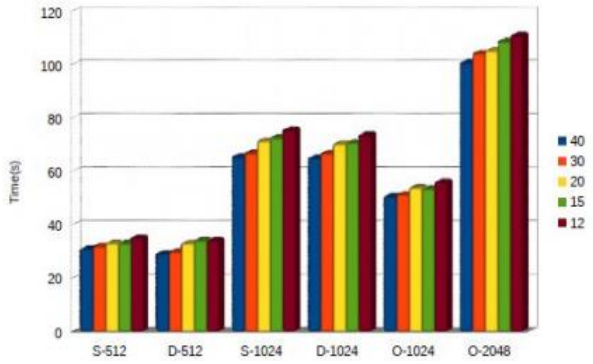
Her bir veri seti için 6X okuma derinliği ve 12X okuma derinliğine karşılık gelen 512K ve 1024K dizileri elde edilmiştir. S-512K, D-512K, S-1024K ve D-1024K şeklinde dört adet test dosyası üretilmiştir.

1 NVIDIA K20 GPU'nun test dosyasının sahip olduğu uzun diziler ile filtre çekirdeğinin gerektirdiği süre Tablo 5'te gösterilmiştir. Eşleşme uzunluğu 15 olarak alındığı zaman, filtre prosedürü (S-1024) için 521.220 sn süre geçmektedir. Filtre, O-1024K test dosyası ortalama 270bp uzunluğunda dizilerden oluşan 311.848s'de işlemi tamamlar.

Tablo 5. 1 NVIDIA K20 GPU zamanlama sonuçları.

File name	40	30	20	15	12
O-256	72.302	73.008	77.115	78.179	82.425
S-512	244.086	245.733	247.115	251.594	255.962
D-512	241.708	243.206	246.760	247.745	250.601
O-512	145.686	151.749	152.332	153.249	158.653
S-1024	498.332	506.403	514.875	521.220	
D-1024	480.538	488.640	497.512	512.953	
O-1024	289.987	298.071	305.200	311.848	

İkinci küme, 10 NVIDIA K20 GPU'dan oluşmaktadır. 10 GPU kümesi /1 GPU kümesinin ortalama performans oranı 7'dir.



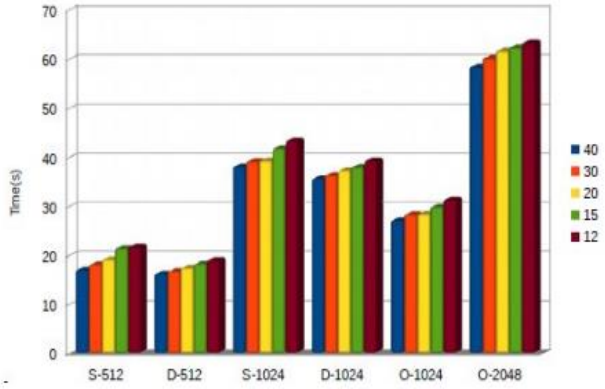
Şekil3. Filtre modülü zamanlaması: 10 NVIDIA K20 GPU.

10 NVIDIA K20 GPU'nun test dosyasının sahip olduğu uzun diziler ile filtre çekirdeğinin gerektirdiği süre Tablo 6'da gösterilmiştir. Eşleşme uzunluğu 15 olarak alındığı zaman, filtre prosedürü (S-1024) için 72.766 sn süre geçmektedir. Filtre, O-1024K test dosyası ortalama 270bp uzunluğunda dizilerden oluşan 53.618 sn'de işlemi tamamlar.

Tablo 6. 10 NVIDIA K20 GPU zamanlama sonuçları.

File name	40	30	20	15	12
S-512	31.098	32.128	33.115	33.182	35.224
D-512	29.319	30.081	33.129	34.305	34.339
S-1024	65.710	66.987	71.440	72.766	75.764
D-1024	65.267	66.879	70.426	70.961	73.812
O-1024	50.865	51.543	53.909	53.618	56.229
O-2048	99.015	102.426	104.367	108.944	109.159

Son olarak üçüncü test kümesinde her özelliği aynı olan 20 GPU kullanılmaktadır. 20 GPU kümesi / 1 GPU kümesi ortalama performans oranı yaklaşık 13,6'dır.



Şekil 4. Filtre modülü zamanlaması: 20 NVIDIA K20 GPU.

20 NVIDIA K20 GPU'nun test dosyasının sahip olduğu uzun diziler ile filtre çekirdeğinin gerektirdiği süre Tablo 7'da gösterilmiştir. Eşleşme uzunluğu 15 olarak alındığı zaman, filtre prosedürü (S-1024) için 41.469 sn süre geçmektedir. Filtre, O-1024K test dosyası ortalama 270bp uzunluğunda dizilerden oluşan 29.941 sn'de işlemi tamamlar.

Tablo 7. 20 NVIDIA K20 GPU zamanlama sonuçları.

File name	40	30	20	15	12
S-512	17.102	18.171	19.245	21.484	21.845
D-512	16.319	16.812	17.529	18.308	19.110
S-1024	38.221	39.242	39.404	41.869	43.443
D-1024	35.716	36.421	37.347	38.110	39.412
O-1024	27.200	28.373	28.442	29.941	31.356
O-2048	56.498	60.256	61.719	61.429	63.423

Kaynaklar

Bentley DR, Balasubramanian S, Swerdlow HP, Smith GP, Milton J, Brown CG, et al. Accurate whole human genome sequencing using reversible terminator chemistry. Nature. 2008;456(7218):53-9.

Cormen, T.H., Leiserson, C.E. ve Rivest, R.L. (1989). Introduction to Algorithms The MIT Press, Boston.

İnan O. (2006). Ardışık tekrarlı DNA dizilerinin optimum düzeyde bulunmasına yönelik programlama çalışmaları.

Kalyanaraman A., Aluru S., Brendel V., Kothari S., Space and time efficient parallel algorithms and software for EST clustering, IEEE Transactions on parallel and distributed systems, 14:1209–1221, 2003.

Kärkkäinen J. Sanders P. Simple linear work suffix array construction, Automata, Languages and Programming, Springer Berlin Heidelberg, 943–955, 2003.

Kim, D.K., Sim, J.S., Park, H., Park, K.: Linear-time construction of suffix arrays. In: Proceedings 14th Annual Symp. Combinatorial Pattern Matching, LNCS 2676, Springer-Verlag. (2003) 186–199.

Ko P. and Aluru S. Space efficient linear time construction of suffix arrays. In Proceedings 14th CPM, LNCS 2676, Springer-Verlag, pages 200–210, 2003.

Kushner I. ve Samols D., Oswald Avery and the pneumococcus.,2011

Manber U., Myers G., Suffix arrays: a new method for on-line string searches. First Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 319-327, 1990.

Margulies M, Egholm M, Airman WE, Attiya S, Bader JS, Bemben LA, et.al.. Genome sequencing in microfabricated high-density picolitre reactors. Nature. 2005 Sep 15;437(7057):376-80.

Olsvik O, Wahlberg J, Petterson B; ve diğerleri. (January 1993). "Use of automated sequencing of polymerase chain reaction-generated amplicons to identify three types of cholera toxin subunit B in Vibrio cholerae O1 strains". J. Clin. Microbiol. 31 (1), s. 22-5.

Pettersson E, Lundeberg J, Ahmadian A (February 2009). "Generations of sequencing technologies". Genomics. 93 (2), s. 105-11.

Pray, L. (2008) Discovery of DNA structure and function: Watson and Crick. Nature Education 1(1):100

Richter D.C., Ott F., Auch A.F., Schmid R., Huson D.H. MetaSim - A Sequencing Simulator for Genomics and Metagenomics. PLoS ONE 3(10): e3373, 2008.

Sanger F, Nicklen S, Coulson AR (December 1977). "DNA sequencing with chain-terminating inhibitors". Proc. Natl. Acad. Sci. U.S.A. 74 (12): 5463–7.

Sanger et al., 1977b. Nucleotid sequence of bacteriophage X174 DNA. Nature 265:687-695.

Savran I., High-performance meta-genomic gene Identification(2014).

Stein LD.,International Human Genome Sequencing Consortium. Finishing the euchromatic sequence of the human genome. Nature. 2004;431(7011):931-45.

Türktaş M., DNA Dizi Analizi. Floresan Temelli Yeni nesil genetik analiz uygulamaları: DNA dizi analizi, moleküler markörler uygulamaları ve çoklu gen anlatım analizleri uygulamalı eğitimi kitapçığı, TÜBİTAK (2011).

Ulutin, T. (2005). Insan genom projesi. Moleküler Hematoloji ve Sitoge netik Alt Komitesi, Temel Moleküler Biyoloji Kursu,70-72.

Üstek D. ,Abacı N. , Sırma S. ,Çakiris A.(2011) Deneysel Tıp Ara¸stırma Enstitüsü Dergisi,1(1),11-18.

Wolpaw R. J., Birbaumer N., McFarland, D.J., Pfurtscheller, G., Vaughan, T.M., (2002). BCI for communication and control, *Clinical Neurophysiology*, 113, 767-791.

NCBI: National Center for Biotechnology Information. http://www.ncbi.nlm.nih.gov Erişim tarihi 20 Kasım 2018.

Filter design (for error correction) in meta-genomic gene analysis

Extended abstract

The use of biology on computers started in the 1960s. Thereafter as computer science develops, the development of biology and research areas have increased. As the developments increase, researches on biology have become more common in areas such as large-scale experiments, macromolecular structures, DNA and RNA. Over the past 50 years, the increase in disease has led scientists to study DNA sequences. In the early 1970s, the first DNA sequences were discovered by university researchers. Later, when Sanger et al. succeeded in sequencing the human genome with the sequencing method developed, the Next Generation Sequencing concept emerged.

Sequences are used to sequencing large-scale gene sequences using the Next Generation Sequencing method. However, because the length of the gene sequence is too large, errors are encountered during sequencing. Also,, the ideal DNA sequencing should reach a faster and easier solution.

In this study, a filter was designed to reduce errors in gene sequencing. The filter design was made by using suffix trees, PaCE filter, Skew and Ko-Aluru algorithms. The suffix tree has a high access rate. It consists of paths starting from root to the leaves. PaCE filter is a software that uses suffix trees to group gene sequences. Analyze gene sequence easily and quickly. The Skew algorithm creates suffix sequences using the suffix tree structure. It is preferred because it is an optimal structure.

Ko-Aluru's algorithm has a great advantage for sequencing large data sets. The Ko-Aluru algorithm is based on the syntax sequence of the next suffix with the neighbor's suffix.

With the help of the filter, large-scale gene sequences are divided into several groups according to similarity. In this series, the correct sequences are tried to be obtained.

The filter module classifies the sequences to be delivered to the other bioinformatics analysis. The results obtained from the filter are compared with the results obtained from the PaCE filter. The reason we chose the PaCE is that it is an open source MPI based filter.

Examples of the meta-genome data set consist of a total of 25 complete bacterial data sets, 13 of them are from the Bacillus genus. We called this set as "the similar group." 12 genomes are called "dissimilar group" from the Protebacteria Phylum. This genomes are taken from the NCBI database. The NCBI database is a public database (NCBI) created in 1988 for use in molecular biology and genetics.

This data is based on a new generation of arrays with the MetaSim application. MetaSim is a sorting simulator (Richter, 2008). Based on a database of genomes, MetaSim generates the synthetic sequences. The 512K and 1024K datasets correspond to 6x and 12x coverage rate. There are "S" and "D" prefixes corresponds to the similar and the dissimilar set. Four test files have been produced in the form of S-512K, D-512K, S-1024K and D-1024K.

In the test, we utilized NVIDIA K20 GPUs. When the common substring-length is set to 15, the filter procedure required 521.220 s to evaluate (S-1024) file. On the other hand, the at 311,848 s was required to process the O-1024K test file.

Keywords: *DNA sequencing; Biyoinformatic; New generation sequencing; Meta-genomic gene; Suffix tree; Filter design.*