Research Article/Araştırma Makalesi

GAZİ
JOURNAL OF ENGINEERING SCIENCES

GPU Performance of Alignment Step in Next Generation Sequencing Analysis

Hilal Akarkamçı[*a], Gülistan Özdemir Özdoğan[b]

Submitted: 05.07.2024 Revised: 20.08.2024 Accepted: 22.08.2024 doi:10.30855/gmbd.0705A17

ABSTRACT

As the amount of biological data has increased, it has become difficult to process it effectively. This has brought the discipline of bioinformatics to the forefront and increased the need for the development of relevant tools. A sensitive data analysis process is required to make sense of this large amount of data produced by next generation sequencing technique. The most costly step in this process is the alignment step. One of the most effective techniques to reduce this cost is the use of a graphics processing unit. In this study, the performances of the CPU-based Burrows-Wheeler aligner and the GPU programming version BarraCUDA tools in the alignment step were compared in terms of alignment rates and computation times for different datasets. In the study, total runtime of these tools was also examined, as well as the runtime of the alignment sub-steps when using one or more GPUs. While there is a similarity in the alignment rates of the tools used in each data set, it has been observed that there is a significant time benefit in data of different sizes through GPU supported BarraCUDA. As a result, with the use of GPU in the alignment step, approximately 5 times acceleration was achieved in single-end data and approximately 9 times in paired-end data.

Keywords: CUDA, GPU, Alignment, Next generation sequencing, BarraCUDA

[a,*] Ankara Yıldırım Beyazıt University, Computer Engineering, 06010 - Ankara, Türkiye
Orcid: 0000-0003-4787-105X
e mail: kaya.hilal79@gmail.com

[b] Ankara Yıldırım Beyazıt University, Computer Engineering, 06010 - Ankara, Türkiye
Orcid: 0000-0001-8221-8473

*Corresponding author: kaya.hilal79@gmail.com

Yeni Nesil Dizileme Analizinde Hizalama Adımının GPU Başarımı

ÖZ

Biyolojik verilerin miktarının artmasıyla birlikte, bu verilerin etkin bir biçimde işlenebilmesi güçleşmiştir. Bu durum, biyoinformatik disiplinini ön plana çıkarmış ve ilgili araçların geliştirilmesine olan ihtiyacı artırmıştır. Yeni nesil dizileme tekniği ile üretilen büyük miktardaki verinin anlamlandırılabilmesi için hassas bir veri analizi süreci yürütülmelidir. Bu süreç içerisinde en yüksek maliyetli adım, hizalama adımıdır. Bu maliyeti azaltan en etkili tekniklerden birisi de, grafik işlem biriminin kullanılmasıdır. Bu çalışmada; hizalama adımında CPU ile çalışan Burrows-Wheeler hizalayıcı ve GPU programlama versiyonu olan BarraCUDA araçlarının performansları, farklı veri setleri için hizalama oranları ve hesaplama zamanları açısından karşılaştırılmıştır. Çalışmada ayrıca, bu araçların toplam çalışma zamanlarının yanı sıra, hizalama alt adımlarının bir veya birden fazla GPU kullanıldığındaki çalışma zamanları da incelenmiştir. Her bir veri setinde kullanılan araçların hizalama oranlarında benzerlik görülmekle birlikte, GPU destekli BarraCUDA aracılığıyla farklı büyüklükteki verilerde zaman açısından önemli bir yarar sağlandığı görülmüştür. Sonuç olarak, hizalama adımında GPU kullanımı ile tek uçlu verilerde yaklaşık 5 kat, çift uçlu verilerde ise yaklaşık 9 kat hızlanma elde edilmiştir.

Anahtar Kelimeler: CUDA, GPU, Hizalama, Yeni nesil dizileme, BarraCUDA

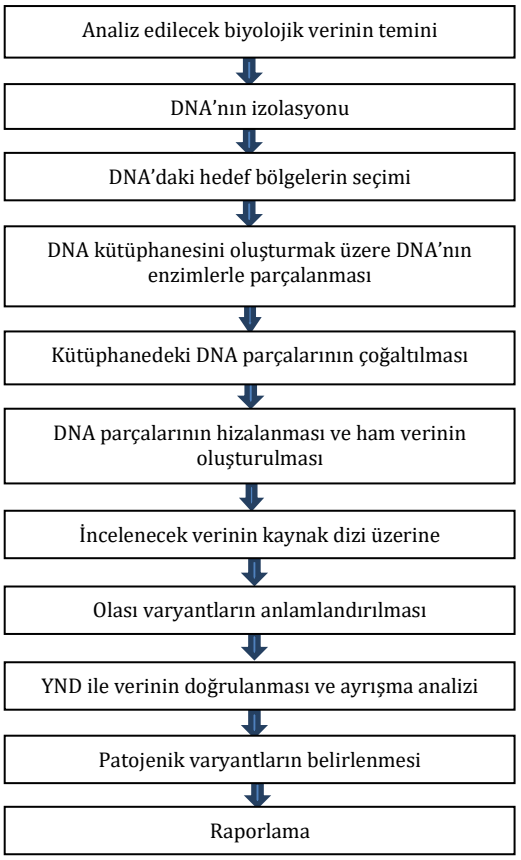
To cite this article: H. Akarkamçı and G. Özdemir Özdoğan, "GPU Performance of Alignment Step in Next Generation Sequencing Analysis," *Gazi Journal of Engineering Sciences,* vol. 10 no. 2, pp. 432-442, 2024. doi:10.30855/gmbd.0705A17

1. Giriş (Introduction)

Genom çalışmalarında, insanda bulunan genlerin tümünün haritalandırılmasını ve anlaşılmasını amaçlayan İnsan Genom Projesi (İGP) önemli bir yere sahiptir. Bu projeyle insan genomu üzerindeki baz çiftlerinin diziliminin anlaşılması ve genlerin tanımlanması hedeflenmiştir. İGP Projesinin tamamlanması; kanserin sebepleri, genetik rahatsızlıkların tanı ve tedavisi, yeni ilaçların üretilmesi, genlerin fonksiyonelliğinin araştırılması ve biyoinformatik disiplinin gelişmesi gibi konularda büyük gelişmelere yol açmıştır [1]. Diğer taraftan, 1990 yılında başlayıp 2003 yılında tamamlanan İGP, hem zaman hem de bütçesi itibariyle tarihin en yüksek maliyetli projelerinden birisi olmuştur. Bu projedekine benzer ancak daha düşük maliyetli yeni bir yaklaşımın geliştirilmesi amacıyla, 2004 yılında Ulusal İnsan Genomu Araştırma Enstitüsü tarafından yeni bir proje başlatılmıştır. Bu yeni proje ile geliştirilen teknoloji, DNA/RNA'nın paralel olarak dizilenmesi esasına dayandığından, Yeni Nesil Dizileme (YND), Masif Paralel Dizileme (MPD) ya da İkinci Nesil Dizileme (İND) olarak isimlendirilmiştir.

YND teknolojisine göre okumalar, DNA'nın birden fazla parçaya ayrılması ile üretilen verinin üzerinde çeşitli analizler yapılarak anlamlandırılır [2]. YND veri analizi olarak bilinen bu süreç, Şekil 1'de detaylıca gösterilmekte olup temel olarak kalite değerlendirmesi, hizalama, varyant çağırma, varyantı anlamlandırma ve görselleştirme adımlarından oluşur.

Bu adımların arasında en yüksek maliyetli olan, üst üste gelen çok sayıda küçük dizi parçacığının birleştirilerek orijinal dizinin elde edildiği hizalama adımıdır. Bu adımın maliyetini azaltabilmek için farklı teknolojileri kullanan araçlar geliştirilmiştir. Bu araçlara verilecek örneklerin başında, Grafik İşlem Birimi (GPU) gelir. GPU, aynı anda çok sayıda iş parçacığını çalıştırabilmesi sayesinde hizalama adımında tercih edilen bir araç olmuştur [3,-5]. OpenCL [6], UPC++ [7], MIC [8] hizalamada kullanılan diğer araçlara verilebilecek örneklerdendir.



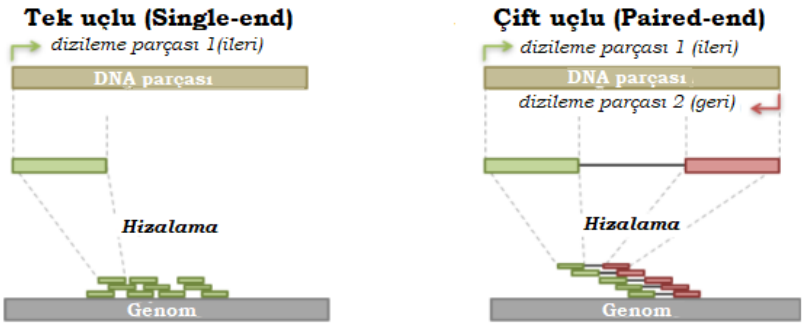
Şekil 1. YND (Yeni Nesil Dizileme)'nin iş akışı şeması
(Workflow diagram of NGS (Next Generation Sequencing))

Önceki yıllarda GPU kartlarına kolayca ulaşılamadığından dolayı bu yönde geliştirilen araç ve metotların biyoinformatik alanında çalışanlar tarafından pek kullanılamadığı belirtilse de [9], GPU'nun

daha kolay ulaşılabilir hale gelmesi, GPU ile geliştirilen hizalama araçlarının daha fazla araştırmacı tarafından kullanılmasını sağlamıştır. GPU için geliştirilen hizalama araçlarından bazıları; BarraCUDA [10], SOAP3-dp [11], CUSHAW2-GPU [12] ve nvBowtie [13] 'dir. Çeşitli araştırmalarda [14,15,4], GPU destekli hizalama araçlarının performansları incelenerek araştırmacılara farklı bir bakış açısı sunmaya çalışılırken, YND veri analizinde hangi aracın en etkili olduğunu belirtmek mümkün olmamaktadır.

Birçok yüksek verimli dizileme çalışmasında, çift uçlu (ÇU) DNA verisi üretilir. Çift uçlu okumalar, tek uçlu (TU) verilere göre çeşitli avantajlar sağlamakla birlikte, tek uçlu DNA okuma işlem akışında küçük değişikliklerle elde edilebilirler. Çift uçlu DNA okuma işlemi, kısa okumaların referans genomuna güvenilir şekilde eşlenmesinde ve yapısal varyasyonların aşağı akış (downstream) analizlerinde kolaylık sağlayan ekstra konumsal bilgiyi içerme avantajına sahiptir [16,17].

Sunulan bu çalışma kapsamında, hizalama araçlarından birisi olan BWA [18,19] ve GPU versiyonu olan BarraCUDA[10] üzerinde durulacaktır. BarraCUDA'da, GPU'nun sağladığı süre avantajı, CPU versiyonuna benzer hizalama oranı ve birden fazla GPU ile çalışabilme özelliği bulunmaktadır. Buna göre; çalışma kapsamında, BWA ve BarraCUDA, Şekil 2'de analizi görülen tek uçlu ve çift uçlu farklı okuma uzunluklarına sahip gerçek veriler üzerinde çalıştırılmıştır. Ayrıca, bu araçların hizalama oranlarının ve çalışma zamanlarının karşılaştırılmasıyla birlikte GPU'nun sağladığı avantajlar incelenmiştir.



Şekil 2. Tek uçlu ve çift uçlu verilerin analizi
(Analysis of single-end and paired-end data)

Bu çalışmanın amacı, GPU aracının performansının farklı özelliklere sahip veriler üzerinde izlenmesi ve toplam çalışma zamanına ek olarak, hizalama alt adımlarının çalışma zamanlarının da ayrı ayrı incelenerek performanslarının karşılaştırılmasıdır. BarraCUDA üzerine yayınlanan temel makalede [10], hizalama alt araçları tek bir GPU üzerinde incelenirken, bu çalışmada hizalama araçlarının birden fazla GPU üzerindeki başarım durumları da incelenerek sonuçları paylaşılmıştır.

2. Materyal ve Metot (Materials and Methods)

YND'de, çalışılan genomun çok sayıda küçük parçaya ayrılarak bir sıralayıcı cihazı ile çeşitli adımlardan geçirilmesiyle bu parçalara ait nükleotid (Adenin–A, Sitozin–C, Guanin–G, Timin–T) dizileri belirlenir. Her bir parçaya ait A, C, G, T bazlarından oluşan nükleotid dizisine okuma denir. Her bir okuma, baz çifti anlamına gelen bp ile ifade edilir. Okuma uzunluğu, kullanılan dizileme platformuna bağlıdır. Sıklıkla kullanılan YND platformları Illumina ve IonTorrent'tir.

Hizalama, çalışılan genoma ait YND ile üretilmiş çok sayıda okumanın birleştirilerek bu genomun DNA diziliminin belirlenmesi sürecidir. İnsanda olduğu gibi karmaşık genomların hizalanmasında, çalışılan genom için kabul görmüş bir referans genoma ihtiyaç duyulur. Hizalama işlemi, YND veri analizi içinde en temel adımdır ve bu sürecin doğru, verimli ve aynı zamanda hızlı olabilmesi, veri analizinin diğer adımları için son derece önemlidir. Ancak, YND ile üretilen çok sayıda okumanın referans genoma eşleştirilmesi zor bir süreçtir. Bunda, okumaların uzunluğunun kısa olması bir etmendir. Diğer bir etmen ise, çalışılan genom üzerinde varyant denilen farklılıkların bulunmasıdır [20]. Ayrıca, okumaların olası başlangıç konumlarının belirlenmesi yoğun hesaplama gerektiren bir süreçtir [6].

Kısa okumaların dizilenmesi için kullanılan hizalama algoritmaları, kullandıkları indeksin oluşturulma şekline göre hash tablosu ve sonek ağacı kullananlar olmak üzere ikiye ayrılır [1,2]. Bunların ilki, daha yavaş ve hassas çalışırken, ikincisi hafızayı daha etkin kullanarak daha hızlı çalışır. Hizalama algoritmaları içinde en bilinen ve sıklıkla kullanılan BWA algoritması, sonek dizisini Burrows-Wheeler

Transform (BWT) ile birleştiren bir indeks yapısı olan FM-index kullanır. BWA'nın hizalama süreci, okuma uzunluğuna göre üç farklı algoritmadan oluşur. Bunlar, 100 bp uzunluğuna kadar olan okumalar için 'backtrack', daha uzun okumalar için geliştirilen 'SW' ve 'mem' algoritmalarıdır. 'backtrack' algoritması, 'aln' ve 'samse/sampe' olmak üzere iki alt adımdan oluşur. 'aln' ile okumaların sonek dizisindeki koordinatları belirlenirken, 'samse/sampe' ile de bu koordinatların referans genom üzerinde dönüşümleri yapılarak standartlaşmış bir metin dosyası olan SAM formatında yazdırılır. Burada, çalışılan genoma ait dizinin tek uçtan ya da her iki uçtan okunmasına bağlı olarak kullanılan algoritma 'samse' ya da 'sampe' olarak değişir.

GPU, sahip olduğu çok sayıda çekirdek ile aynı anda birçok işi bir arada yürütebilen ve bu yönüyle paralel işlemler için uygun olan bir işlemci birimidir. CPU ile karşılaştırıldığında, daha yüksek bir verimlilik, yüksek hızlanma oranları ve düşük bellek gecikmesi gibi avantajlar sağlamaktadır [20]. GPU'nun sağladığı yüksek hızlanma oranının temel sebebi, tekli işlem çoklu veri hesaplama modeli ile aynı komutun işlemci üzerinde bulunan çok sayıda iş parçacığı ile paralel olarak çalıştırılmasıdır. CUDA ise, NVIDIA tarafından sunulan GPU programlamada kullanılan bir hesaplama platformudur. GPU öncelikle grafik işleme için kullanılsa da, zamanla yoğun hesaplama içeren diğer problemlerde de kullanılmaya başlanmıştır. Genel amaçlı GPU (GPGPU) programlama olarak bilinen bu yaklaşımın yaygınlaşmasında, GPU maliyetlerinin düşmesi ve NVIDIA'nın CUDA platformu gibi çeşitli paralel hesaplama platformlarının geliştirilmesi etkili olmuştur. Bir CUDA programı iki bileşenden oluşur [21]. Sunucu adı verilen kısım, CPU üzerinde çalışan kısımdır. Diğeri ise, kernel denilen daha küçük, ama hesaplama işini üstlenen ve GPU üzerinde çalışan kısımdır. 'kernel' çalıştırılırken ihtiyaç duyulan veri, önce GPU'nun belleğine kopyalanmalıdır. Çıktı verileri de, GPU'nun belleğinden CPU'ya yüklenir.

BarraCUDA, BWA algoritmasına dayalı bir GPGPU dizi hizalama aracıdır. Her bir okumanın referans genoma hizalanması diğer okumalardan bağımsız bir süreç olduğundan [6], BarraCUDA burada verinin paralelleştirilmesi yaklaşımını kullanır. Temelde şu şekilde çalışır: İlk olarak, hizalama adımının girdi verileri olan indekslenen referans genom ve okumalar, diskten GPU'nun belleğine yüklenir. Daha sonra, bir GPU kerneli oluşturularak, okumaların GPU üzerindeki işlemcilere dağıtılmasıyla hizalama sürecinin paralel olarak gerçekleştirilmesi sağlanır. 'kernel' tamamlandığında ise, hizalama sonuçları GPU'dan diske aktarılır [10].

BarraCUDA aracının doğruluk ve hız açısından performansı, CPU üzerinde iş parçacığı ile çalışan BWA versiyonu ile karşılaştırılmıştır [10]. İlgili çalışmada, BWA ile benzer bir hizalama oranına sahip olduğu belirtilmiş ve hız açısından da incelendiğinde, tek GPU ile elde edilen verimin 6 iş parçacığı kullanılarak 'aln' adımında elde edilen verime eşdeğer olduğu paylaşılmıştır. Birden fazla GPU kullanımının da CPU'dan daha iyi ölçeklenebilirlik sağladığı belirtilmiştir.

2.1. Veri setleri (Datasets)

Tek uçlu dizileme, genellikle çift uçlu dizilemeye kıyasla daha kısa okuma uzunlukları üretir. Yapılacak analiz için uzun okumalara ihtiyaç olduğunda veya hedef bölgenin tekrarlayan diziler içermesi durumunda, belirsizlikleri çözme ve okumaları doğru şekilde hizalama yeteneği nedeniyle çift uçlu dizileme verisi, tek uçlu dizilemeye göre daha avantajlı olabilir [22].

Sunulan çalışmada, farklı özelliklerde 12 gerçek veri seti kullanılmıştır. Veriler ve verilere ait özellikler Tablo 1'de sunulmuştur. Veri setleri, 35, 36, 37, 51, 65, 76, 100 ve 101 olmak üzere farklı okuma uzunluklarına sahiptir. Bunun yanı sıra, Tablo 1'de görüldüğü gibi, her verinin okuma sayısı da farklıdır ve ÇU veriler için toplam okuma sayısı kullanılmıştır. Çalışmada kullanılan veriler, NCBI'nin SRA veri deposundan veri indirmek için kullanılan "fastq-dump 2.9.0" aracı ile indirilmiştir.

Verilerin seçiminde, Illumina platformu tarafından dizilenen DNA verisi olmaları, farklı okuma sayısı ve okuma uzunluklarına sahip olmaları dikkate alınmıştır. Bunlar dışında, BarraCUDA aracı geliştirilirken 1 GPU için kullanılan ERR003014 ve SRR032215 etiketli iki verinin 2 GPU'daki yaklaşımını inceleyebilmek adına bu iki veri de çalışmaya eklenmiştir. Ayrıca, CUDA tabanlı geliştirilen hizalama araçlarından olan Cushaw, CUSHAW2-GPU ve nvBowtie araçlarının test edildiği ERR000589, SRR211279 ve ERR161544 verileri de çalışmaya eklenerek BarraCUDA üzerindeki performanslarının incelenmesi amaçlanmıştır. Farklı sayıda GPU kullanımının performansını inceleyebilmek için, literatür araştırmasında ulaşılabilen tek uçlu veriler kullanılmıştır. SRR622457 gibi diğer verilere kıyasla daha büyük bir veri de çalışmaya eklenerek hesaplama ortamının disk alanı, hafıza, her bir yürütme için atanan süre gibi kaynaklarının imkân verdiği ölçüde bu verinin de performansı incelenmiştir.

Tablo 1. Kullanılan veri setlerinin özellikleri (Properties of the used datasets)

Veri adı	Kısaltma	TU/ÇU	Okuma uzunluğu (bp)	Toplam okuma sayısı
SRR070994	VS1	TU	35	6810249
SRR071052	VS2	TU	36	7853651
SRR10696340	VS3	TU	51	8448534
SRR5559128	VS4	TU	65	16821658
ERR003014	VS5	ÇU	37	22673582
ERR000589	VS6	ÇU	51	4279572
SRR032215	VS7	ÇU	76	28291390
SRR211279	VS8	ÇU	100	50937050
ERR251661	VS9	ÇU	100	96723198
ERR161544	VS10	ÇU	100	148223280
SRR622461	VS11	ÇU	101	184918908
SRR622457	VS12	ÇU	101	2873647546

2.2. Analiz (Analysis)

Her bir veri seti, hem CPU hem GPU üzerinde test edilmiştir. CPU üzerindeki yürütmeler BWA 0.7.17 aracı ile 20 adet çift çekirdeğe sahip makineler üzerinde; GPU üzerindeki yürütmeler ise, BarraCUDA 0.7.0r107 aracı ile Nvidia P100 GPU kartları üzerinde gerçekleştirilmiştir. BWA, CPU üzerinde çalıştırılırken farklı sayıda iş parçacığı (1, 8, 16, 24, 32, 40), BarraCUDA'da ise farklı sayıda GPU (1, 2, 3, 4) kullanılmıştır.

Analizler sırasında kullanılan referans insan genomu hg38, diğer bir ifadeyle GRCh38 genomudur. Hizalama için öncelikle referans genomun indekslenmesi gerekir. İndeks oluşturma süreci, BWA ve BarraCUDA için ayrı ayrı olmak üzere, bir kez çalıştırılmış olup, bu süreç sonuçlara katılmamıştır. Hizalama sürecinin iki alt adımda gerçekleştiğinden ve verinin TU ya da ÇU olmasına bağlı olarak 'samse' ya da 'sampe' alt adımı kullanıldığından Bölüm 2'de bahsedilmişti. BWA aracının çalışma prensibine göre, 'aln' adımı farklı sayıda iş parçacıkları ile çalıştırılabilirken, 'samse/sampe' adımında, birden fazla iş parçacığı desteklenmemektedir. BarraCUDA aracında ise, 'aln' adımında GPU kullanılabilirken, sadece 'sampe' adımında, BWA aracından farklı olarak, birden fazla iş parçacığı kullanılabilmektedir. BarraCUDA, TU verilerinde istenilen sayıda GPU'yu, ÇU verilerinde ise 1 GPU ya da her bir veri bir GPU üzerinde olmak üzere 2 GPU'yu desteklemektedir.

Analizler sırasında çalışma zamanı olarak 'aln' ve 'samse/sampe' adımlarının çalışma zamanlarının toplamı alınmıştır. Her yürütme üç kez tekrarlanarak sonuçların ortalaması alınmıştır. BWA aracında kullanılan iş parçacığı sayısı, BarraCUDA'da ise kullanılan GPU sayısı dışında diğer parametreler için varsayılan değerler kullanılmıştır.

3. Bulgular ve Tartışma (Findings and Discussion)

Yapılan analizler sonrasında TU ve ÇU verilere dair elde edilen sonuçlar, Tablo 2 ve Tablo 3'te sunulmuştur. Buna göre, hem TU hem de ÇU verilerde CPU ve GPU üzerinde beklenildiği gibi [10] birbirine çok yakın hizalama oranlarının elde edildiği görülmüştür. Çalışma zamanları incelendiğinde ise, TU verilerde BWA aracı ile farklı iş parçacıklarında çalışma zamanının azaldığı görülürken, BarraCUDA ile de farklı sayıda GPU kullanıldığında çalışma zamanının azaldığı görülmektedir.

Sistemdeki GPU sayısı arttıkça, çalışma zamanı da azalmaya devam etmektedir. Çalışma kapsamında ulaşılabilen TU veriler nispeten daha küçük veriler olup, ilerleyen çalışmamızda BarraCUDA'nın daha büyük TU verileri üzerindeki performansı da incelendiğinde, GPU sayısının artırılmasının benzer şekilde çalışma zamanında azalma sağlaması beklenmektedir.

Tablo 2. Tek uçlu verilerin analiz sonuçları (Analysis results of single-end data)

			BWA						BarraCUDA			
	BWA	BarraCUDA	1İP	8İP	16İP	24İP	32İP	40İP	1GPU	2GPU	3GPU	4GPU
Veri Seti	Hizalama Oranı (%)	Hizalama Oranı (%)	Zaman (dk.)	Zaman (dk.)	Zaman (dk.)	Zaman (dk.)	Zaman (dk.)	Zaman (dk.)	Zaman (dk.)	Zaman (dk.)	Zaman (dk.)	Zaman (dk.)
SRR070994	97,68	97,46	26,24	3,96	2,61	2,37	2,31	2,28	1,88	1,00	0,71	0,58
SRR071052	93,62	93,38	16,71	4,12	2,84	2,85	2,70	2,65	2,19	1,14	0,81	0,63
SRR10696340	97,96	97,63	24,76	7,31	3,74	3,51	3,30	3,22	2,36	1,24	0,87	0,68
SRR5559128	98,13	97,71	69,44	11,91	8,97	7,72	7,26	6,93	5,45	2,63	1,81	1,41

Tablo 3'te görülen ÇU veriler incelendiğinde de benzer bir durum görülmekte olup, bu verilerin daha fazla okuma sayısına ve okuma uzunluğuna sahip olması nedeni ile burada çalışma zamanlarında ciddi bir azalış gözlemlenmektedir. CPU kullanımında maksimum İP sayısı ile elde edilen çalışma zamanı, SRR622457 etiketli büyük veride tek bir GPU kullanımında %54 oranında azalmaktadır. Bu oranın, verinin boyutu küçüldükçe arttığı görülmektedir. Dolayısıyla, veri büyüklüğü arttıkça azalmanın devam etmesi beklenir.

Tablo 3. Çift uçlu verilerin analiz sonuçları (Analysis results of paired-end data)

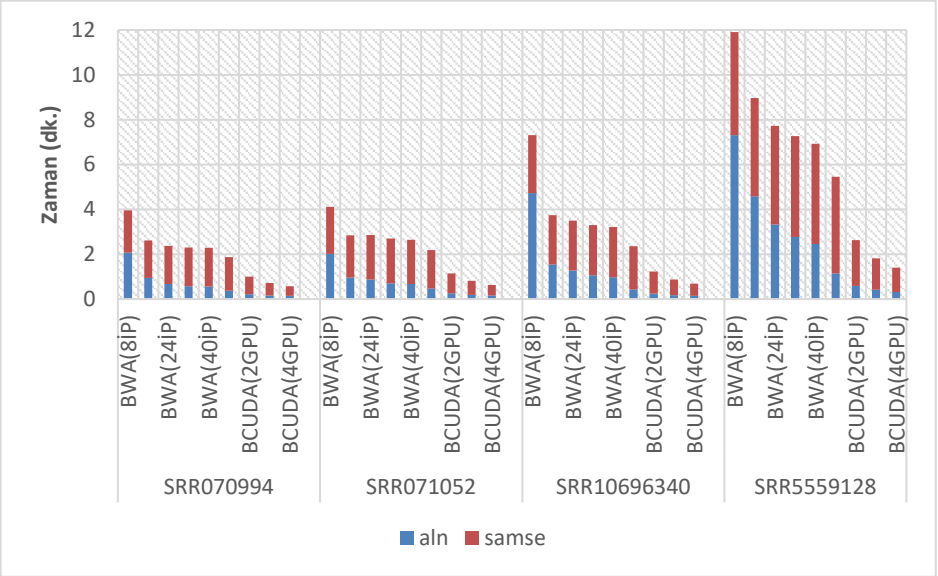
			BWA						BarraCUDA	
	BWA	BarraCUDA	1İP	8İP	16İP	24İP	32İP	40İP	1GPU	2GPU
Veri Seti	Hizalama Oranı (%)	Hizalama Oranı (%)	Zaman (dk.)	Zaman (dk.)	Zaman (dk.)	Zaman (dk.)	Zaman (dk.)	Zaman (dk.)	Zaman (dk.)	Zaman (dk.)
ERR003014	94,93	96,17	82,21	51,11	45,38	44,70	44,57	44,12	6,86	4,77
ERR000589	97,83	97,94	116,25	50,81	43,12	41,80	40,93	40,56	5,92	5,08
SRR032215	88,94	89,05	146,99	40,32	29,53	26,86	25,22	24,57	6,64	6,16
SRR211279	97,67	97,61	301,66	87,45	53,47	46,87	43,58	41,45	14,03	12,50
ERR251661	96,01	96,04	820,89	206,71	118,63	101,73	93,39	88,65	34,46	27,72
ERR161544	98	97,89	829,59	206,78	151,77	133,54	124,52	120,08	49,48	40,53
SRR622461	89,66	89,7	1037,53	232,93	170,70	146,73	135,45	128,02	56,07	48,33
SRR622457	89,75	89,81	16600,01	4193,25	2535,65	2139,22	1951,12	1911,53	880,45	779,76

Çalışma kapsamında, hizalama alt araçları olan 'aln' ve 'samse/sampe' adımının detaylı zaman analizleri yapılmıştır. Buna göre, TU verilerinden elde edilen sonuçlar Şekil 3'te sunulmaktadır. Daha önce belirtildiği gibi, BWA aracında 'aln' adımı birden fazla iş parçacığı ile çalıştırıldığından çalışma zamanlarının azaldığı Şekil 3'te mavi renkli olarak, 'samse' adımının ise birden fazla iş parçacığını desteklememesi sebebi ile çalışma zamanlarının sabit kaldığı Şekil 3'te kırmızı renkli olarak görülmektedir. BarraCUDA'nın sonuçları incelendiğinde, 'aln' adımında tek bir GPU'nun katkısının bile maksimum iş parçacığından elde edilen sonuçtan daha iyi olduğu görülmektedir. GPU sayısı arttığında ise, çalışma zamanındaki azalma devam etmektedir.

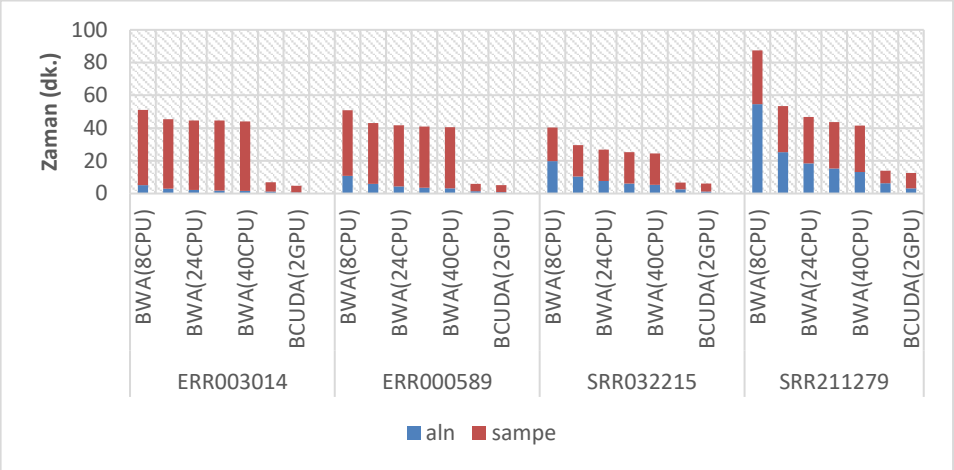
BarraCUDA'nın 'samse' adımında, GPU sayısı kadar bölümlenen verinin aynı sayıda CPU üzerinde çalışmasıyla, Şekil 3'te kırmızı renkli olarak gösterildiği gibi zamanda kısmı bir azalış olduğu anlaşılmaktadır.

Hizalama alt araçlarının ÇU verilerdeki sonuçları ise verilerin büyüklüğü ve ölçeklendirme göz önüne alınarak sırasıyla Şekil 4a, Şekil 4b ve Şekil 4c'de sunulmaktadır. Buna göre Şekil 4a, Şekil 4b ve Şekil 4c'de, BWA sonuçlarında yine 'aln' adımında farklı iş parçacıklarının çalışma zamanını azalttığı mavi renkli olarak, CPU üzerinde yürütülen 'sampe' adımının süresinin ise her iş parçacığı için aynı olduğu kırmızı renkli olarak görülmektedir. BarraCUDA'da 'aln' adımında maksimum iki GPU kullanılmıştır. 'aln' adımı incelendiğinde, her iki grafikte de GPU kullanımının maksimum iş parçacığı ile elde edilen çalışma zamanını azalttığı ve bu azalışın iki GPU kullanımında da devam ettiği gözlemlenmektedir.

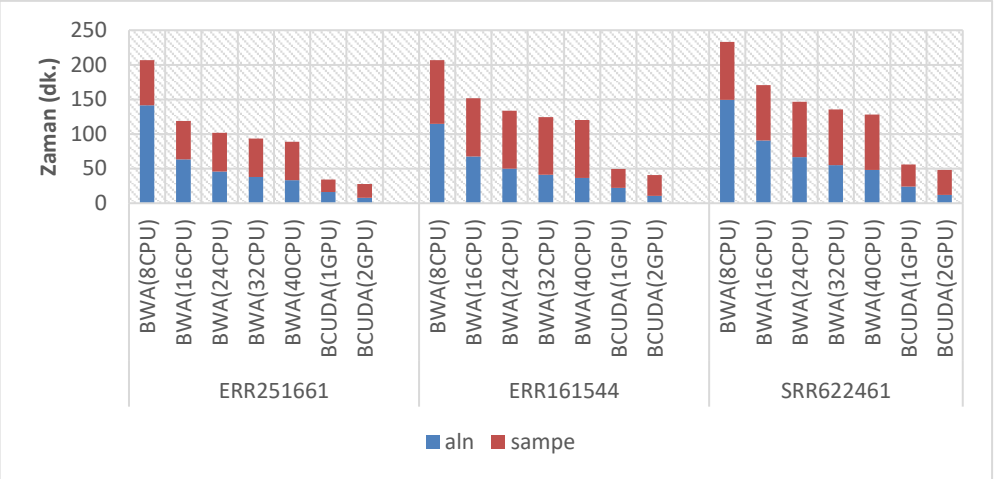
BarraCUDA'ya 'sampe' adımında BWA'dan farklı olarak iş parçacığı desteği eklendiğinden, Şekil 4a, Şekil 4b ve Şekil 4c'de kırmızı renkli olarak ifade edildiği haliyle, 'sampe' adımının süresinin CPU versiyonuna göre hayli azaldığı görülmektedir.



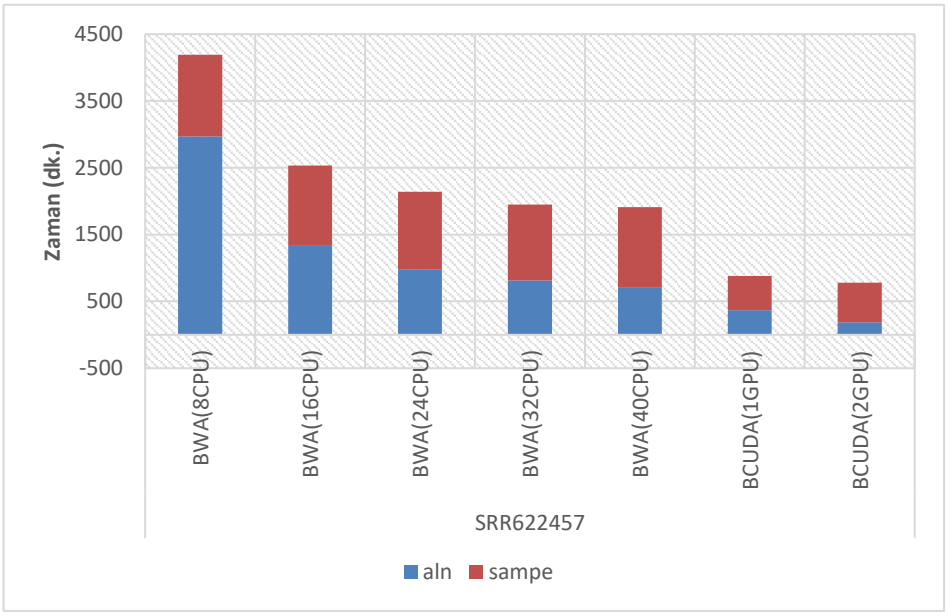
Şekil 3. Tek uçlu verilerin çalışma zamanı grafiği
(Runtime graph of single-end data)



Şekil 4 (a). Çift uçlu verilerin çalışma zamanı grafiği – 1 (ERR003014, ERR000589, SRR032215, SRR211279)
(Runtime graph of paired-end data– 1 (ERR003014, ERR000589, SRR032215, SRR211279))

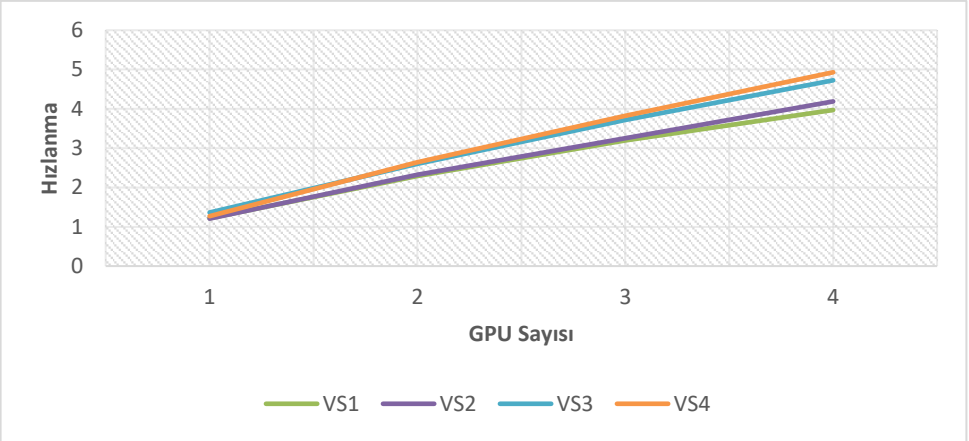


Şekil 4 (b). Çift uçlu verilerin çalışma zamanı grafiği – 2 (ERR251661, ERR161544, SRR622461)
(Runtime graph of paired-end data - 2 (ERR251661, ERR161544, SRR622461))

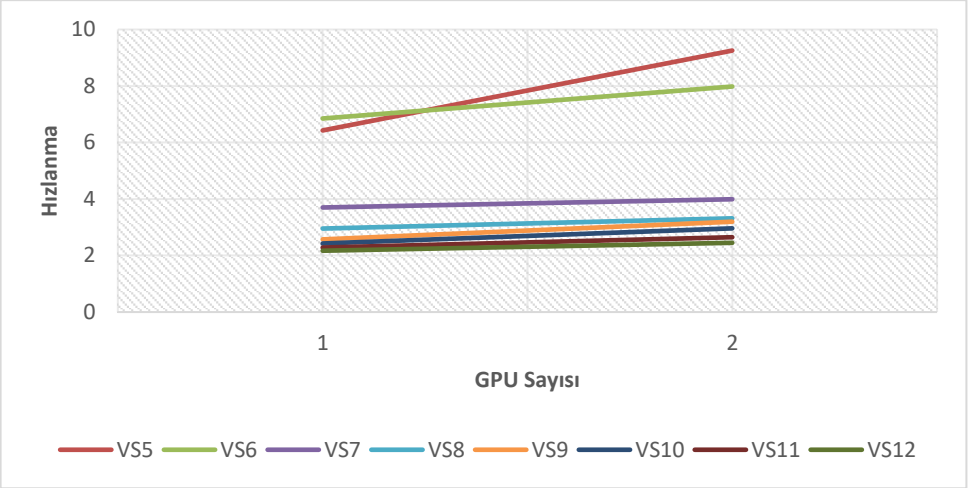


Şekil 4 (c). Çift uçlu verilerin çalışma zamanı grafiği - 3 (SRR622457)
(Runtime graph of paired-end data - 3 (SRR622457))

Hizalama sürecinin maliyetli ve verilerin büyük olmasından dolayı burada elde edilen sayıların nicel değerinin önemi olması ile birlikte, GPU kullanımının gerçekte ne kadar bir katkı sağladığını gösterebilmek adına Şekil 5(a) ve Şekil 5(b)'de hızlanma grafikleri de incelenmiştir. Bunun için, sistemdeki maksimum iş parçacığı sayısı olan 40 iş parçacığının çalışma zamanı referans alınmıştır. Buna göre, maksimum 4 GPU kullanılan TU verilere ilişkin hızlanma grafiği Şekil 5a'da sunulmaktadır. Beklenildiği gibi, GPU kullanımı ile hızlanmadaki artışın, doğrusal bir davranış gösterdiği görülmektedir. Tek GPU kullanımındaki veriler için 1,2-1,4 kat bir artış görülürken, sistemdeki maksimum 4 GPU kullanımında 4-5 kat civarında bir hızlanma olduğu belirlenmiştir. Maksimum 2 GPU kullanılan ÇU verilere ilişkin hızlanma grafiği ise Şekil 5b'de sunulmaktadır. Şekil 5b incelendiğinde, ÇU verilere ilişkin hızlanma grafiğindeki küçük veriler üzerinde 1 GPU kullanıldığında yaklaşık 6-7 kat oranında hızlanma görülürken, bu hızlanma 2 GPU'da 8-9 kat civarında olmaktadır. Sistemde çalışılan en büyük verinin hızlanması incelendiğinde, 1 GPU'da 2 kat olan hızlanmanın, 2 GPU ile 2,5 kata kadar çıktığı görülmektedir.



Şekil 5 (a). Tek uçlu verilerin hızlanma grafiği
(Speedup graph of single-end data)



Şekil 5 (b). Çift uçlu verilerin hızlanma grafiği
(Speedup graph of paired-end data)

4. Sonuçlar ve Tartışma (Results and Discussion)

Yeni nesil dizileme ile elde edilen büyük miktarda verinin analizinde en maliyetli adım olan hizalama sürecindeki bu hesaplama maliyetini azaltabilmek için birçok çalışma yapılmaktadır. Bu çalışmada, literatürde kabul gören BWA ve onun CUDA ile geliştirilmiş versiyonu olan BarraCUDA araçlarına yönelik hizalama oranları ve hesaplama zamanları karşılaştırması sunulmaktadır. CPU üzerinde çalışan BWA aracı ile GPU üzerinde çalışan BarraCUDA aracının farklı özellikteki veri setleri üzerindeki çalışma performansları incelenerek GPU kullanımının hizalama sürecine maliyet açısından katkısı incelenmiştir. Literatürdeki diğer CUDA araçlarının test edildiği veri setleri de dâhil edilerek, farklı verilerde BarraCUDA aracının performansı incelenerek sonuçları analiz edilmiştir. Buna göre, BarraCUDA'nın BWA ile beklenildiği gibi benzer bir hizalama oranına sahip olması yanında, çalışma zamanında ciddi bir azalmaya sebep olduğu görülmüştür. Literatürde GPU kullanan hizalama araçları incelendiğinde çoğu çalışmada, zaman analizinde sadece toplam çalışma zamanı ele alınmıştır. Bu çalışma ile toplam çalışma zamanına ek olarak, bir ve birden fazla GPU üzerinde hizalama alt adımlarının ayrı ayrı zaman analizleri de sunulmuştur.
Tek uçlu ve çift uçlu veriler üzerinde gerçekleştirilmiş çalışmada [23], gen düzeyinde okuma analizi için maliyet etkin bir yaklaşım arayan araştırmacılara, daha uzun tek uçlu bir okuma verisi yerine kısa çift uçlu verileri tercih etmeleri tavsiye edilmiştir.

Görüntü işleme benzeri veri işleme kapasitesi yüksek görevleri paralel hale getirmeye yardımcı olmak için genel amaçlı CPU'larla karşılaştırıldığında, GPU'lar uzmanlaşmış tekli işlem çoklu veri mimarisi yönünde gelişme sağlamış olup, CPU'lardan daha basit işlem çekirdekleri bulunmaktadır. Örneğin,

daha basit kontrol mantığına, genellikle dallanma tahmini veya ön getirme ve küçük çekirdekli belleğe sahiptirler. Daha basit bilgi işlem çekirdekleri, GPU'ların bir çipe genel amaçlı bir CPU'dan çok daha fazla çekirdek sığdırmasına olanak tanır. GPU mimarileri, dallanma koşulu az olan veya veri bağımlılığı olmayan iş yüklerinde son derece iyi performans gösterir. Ek olarak, GPU mimarilerinde, bellek yapıları yüksek hızlı veri akışını desteklemek üzere özelleştirilmiştir [24].

Gen dizilimi çalışmaları da uzun okumalar içeren, yüksek veri işleme kapasitesine sahip uygulamalardır. Geleneksel CPU tabanlı yöntemler, genom dizilerinin artan hacmi ve karmaşıklığıyla mücadele eder ve bu da veri işleme ve analizinde önemli gecikmelere yol açar [25]. COVID-19 pandemisinin yakın zamanda gösterdiği gibi, insanlar arasında bulaşıcı olan hastalıkların bulaşma modellerini anlamlandırmak için, büyük ölçekli patojen genomik verilerin analizi hayati öneme sahiptir. Ham dizi verilerini analize hazır varyantlara işlemek için kullanılan mevcut yöntemler ölçek açısından yeterince performanslı sonuçlar vermemektedir ve hastalık kontrolü için hızlı gözetim çabalarını ve epidemiyolojik araştırmaları yavaşlatmaktadır [26]. Gerçek hayatta hızlı çözümler üretilmesi gereken geniş pandemik süreçlerde, hızlı çözüm üretmek açısından GPU hızlandırmalı genomik çalışmaların başarılı sonuçlar vermesi hayati önem arz etmektedir.

GPU katkısının daha açık anlaşılabilmesi için geliştirdiğimiz çalışmadaki hızlanma değerleri incelendiğinde, tek uçlu verilerde hızlanma değerlerinin 5 kata kadar, çift uçlu verilerde ise 9 kata kadar çıktığı görülmüştür. GPU sayısına göre doğrusal bir davranış gösteren hızlanma artışı görülmektedir. Çalışmanın genel olarak literatürle uyumlu sonuçlar vermesi, ayrıca önceki çalışmalardan farklı olarak birden fazla GPU üzerinde denenmesinin çalışma zamanında azalma sağlaması, farklı tür veriler üzerindeki dizi hizalamasında GPU kullanımını teşvik edici niteliktedir.

Elde edilen doğrusal hızlanma davranışı ile sonraki çalışmalarda, daha fazla okuma sayısına sahip tek uçlu ve çift uçlu veriler de GPU katkısı ile araştırılabilecektir. Çalışmada oluşturulan ardışık düzen, açık bir kütüphane olarak diğer araştırmacılarla paylaşılabilecektir. Ayrıca bu çalışmadan elde edilen deneyimlerle ilerleyen çalışmalarda, bulaşıcı hastalıkların genetik nedenlerinin anlaşılmasına yardımcı olmak üzere farklı veri setleri ve vaka çalışmaları üzerinde genomik varyantların anlamlandırılması konusunda GPU hızlandırmalı hesaplama çalışmaları gerçekleştirilebilecektir.

Teşekkür (Acknowledgment)

Bu araştırmada yer alan tüm nümerik hesaplamalar TÜBİTAK ULAKBİM, Yüksek Başarım ve Grid Hesaplama Merkezi'nde (TRUBA kaynaklarında) gerçekleştirilmiştir.

Çıkar Çatışması Beyanı (Conflict of Interest Statement)

Yazarlar tarafından herhangi bir çıkar çatışması bildirilmemiştir.

Kaynaklar (References)

[1] G. Özdemir Özdoğan, "Retinoblastom hastalığında yeni nesil dizileme veri analizi ile bir ardışık düzenin geliştirilmesi (Development of a pipeline with next-generation sequencing data analysis in retinoblastoma disease)," Ph.D. dissertation. Ankara Yıldırım Beyazıt University, Ankara, Türkiye, 2020.

[2] G. Özdemir Özdoğan and H. Kaya, "Next-generation sequencing data analysis on pool-seq and low-coverage retinoblastoma data," *Interdisciplinary Sciences, Computational Life Sciences*, vol. 12, no. 3, pp. 302–310, Sep. 2020. doi:10.1007/s12539-020-00374-8

[3] M. Nobile, P. Cazzaniga, A. Tangherloni, and D. Besozzi, "Graphics processing units in bioinformatics, computational biology and systems biology," *Brief Bioinform.*, vol. 18, no. 5, pp. 870-885, Sep. 2017. doi:10.1093/bib/bbw058

[4] S. Pawar, A. Stanam, and Y. Zhu, "Evaluating the computing efficiencies (specificity and sensitivity) of graphics processing unit (GPU)-accelerated DNA sequence alignment tools against central processing unit (CPU) alignment tool," *Journal of Bioinformatics and Sequence Analysis,* vol. 9, no. 2, pp. 10-14, July 2018. doi:10.5897/JBSA2018.0109

[5] Y. Liu, J.-Y. Li, Y.-Q. Mao, X.-L. Wang and D.-S. Zhao, "A literature evaluation of CUDA compatible sequence aligners," *in Proceedings of the International Conference on Bioinformatics Models, Methods and Algorithms (Bioinformatics-2013)*, P. Fernandes, J. Solé-Casals, A. L. N. Fred, and H. Gamboa, Eds. Spain: Scitepress, Feb. 2013, pp. 268-271. [Online]. Available: https://dblp.org/db/conf/biostec/bioinformatics2013.html. [Accessed: April 19, 2023].

[6] X. Zhao, C. Liu, and G. Tan, "Implementation of short read alignment algorithm in OpenCL on Xeon Phi coprocessor," *in IEEE 17th International Conference on High Performance Computing and* Communications, HPCC 2015. New York, NY, USA, Aug. 24-26, 2015, IEEE, 2015, pp. 1633-1636. [Online]. Available: IEEE Xplore, https://ieeexplore.ieee.org/abstract/document/7336403. [Accessed: May 20, 2023].

[7] J. González-Domínguez, Y. Liu and B. Schmidt, "Parallel and scalable short-read alignment on multi-core clusters using UPC++," *PLoS One*, vol. 11, no. 1, Jan. 2016. doi:10.1371/journal.pone.0145490

[8] R. Luo, J. Cheung, E. Wu, H. Wang, S.-H. Chan, W.-C. Law and G. He, "MICA: A fast short-read aligner that takes full advantage of Many Integrated Core Architecture (MIC)," *BMC Bioinformatics*, vol. 16, no. 7, p. S10, Apr. 2015. doi:10.1186/1471-2105-16-S7-S10

[9] P. Liu, A. Hemani, K. Paul, C. Weis, M. Jung and N. Wehn, "3D-Stacked many-core architecture for biological sequence analysis problems," *Int J. Parallel Prog*., vol. 45, pp. 1420–1460, Apr. 2017. doi:10.1007/s10766-017-0495-0

[10] P. Klus, S. Lam, D. Lyberg, M. Cheung, G. Pullan and I. McFarlane, "BarraCUDA - a fast short read sequence aligner using graphics processing units," *BMC Research Notes*, vol. 5, no. 2, Jan. 2012. doi:10.1186/1756-0500-5-27

[11] R. Luo, T. Wong, J. Zhu, C.-M. Liu, X. Zhu, E. Wu and L.-K. Lee, "SOAP3-dp: Fast, accurate and sensitive GPU-based short read aligner," *PLoS One*, vol. 8, no. 5, May 2013. doi:10.1371/journal.pone.0065632

[12] Y. Liu and B. Schmidt, "CUSHAW2-GPU: empowering faster gapped short-read alignment using GPU computing," *IEEE Design and Test of Computers*, vol. 31, no. 1, pp. 31 – 39, Febr. 2014. doi:10.1109/MDAT.2013.2284198

[13] "NVBIO: nvBowtie," [Online]. Available: https://nvlabs.github.io/nvbio/nvbowtie_page.html. [Accessed: April 19, 2023].

[14] A. Manconi, A. Orro, E. Manca, G. Armano and L Milanesi, "A tool for mapping Single Nucleotide Polymorphisms using Graphics Processing Units," *BMC Bioinformatics*, vol. 15, no. 1, p. S10, Jan. 2014. doi:1471-2105/15/S1/S10

[15] F. Buntara, B.-S. Lee, R. Purbojati and C. Zhou, "Is GPUs ready to boost genomic alignment computation," *in 2019 International Conference on Innovative Trends in Computer Engineering,* ITCE, 2019. Egypt, February 02-04, 2019, IEEE, 2019, pp. 130-135. [Online]. Available: IEEE Xplore, https://ieeexplore.ieee.org/document/8646637. [Accessed: May 21, 2024].

[16] A. Shrestha and M. Frith, "An approximate bayesian approach to mapping paired-end DNA reads to a reference genome," *Bioinformatics*, vol. 29, no. 8, pp. 965–972, April 2013. doi:10.1093/bioinformatics/btt073

[17] "Advantages of paired-end and single-read sequencing - Illumina," [Online]. Available: https://www.illumina.com/science/technology/next-generation-sequencing/plan-experiments/paired-end-vs-single-read.html. [Accessed: August 7, 2024].

[18] H. Li and R. Durbin, "Fast and accurate short-read alignment with burrows-wheeler transform," *Bioinformatics*, vol. 25, no. 14, pp. 1754-60, May 2009. doi:10.1093/bioinformatics/btp324

[19] H. Li and R. Durbin,, "Fast and accurate long-read alignment with burrows-wheeler transform," *Bioinformatics*, vol. 26, no. 5, pp. 589-95, March 2010. doi:10.1093/bioinformatics/btp698

[20] A. Al Kawam, "Towards the next generation of clinical decision support: Overcoming the integration challanges of genomic data and electronic health records," Ph.D. dissertation. Graduate and Professional Studies of Texas A&M University, Texas, USA, 2018.

[21] M. Schatz, C. Trapnell, A. Delcher and A. Varshney, "High-throughput sequence alignment using graphics processing units," *BMC Bioinformatics*, vol. 8, pp. 474, Dec. 2007. doi:10.1186/1471-2105-8-474

[22] "Single-read vs. paired-end sequencing – CD Genomics," [Online]. Available: https://www.cd-genomics.com/resource-single-read-vs-paired-end-sequencing.html. [Accessed: July 22, 2024].

[23] A. H. Freedman, J. M. Gaspar, and T. B. Sackton, "Short paired-end reads trump long single-end reads for expression analysis," *BMC Bioinformatics*, vol. 21, no. 149, Apr. 2020. doi:10.1186/s12859-020-3484-z

[24] M. Qasaimeh, K. Denolf, J. Lo, K. Vissers, J. Zambreno, and Jones, Phillip, "Comparing energy efficiency of CPU, GPU and FPGA implementations for vision kernels," *in IEEE International Conference on Embedded Software and Systems,* ICESS, 2019, June 02-03, 2019, Las Vegas, NV, USA, 2019, pp. 1-8. [Online]. Available: IEEE Xplore, https://ieeexplore.ieee.org/document/8782524. [Accessed: Aug. 20, 2024].

[25] K. R. Franke and E. L. Crowgey, "Accelerating next generation sequencing data analysis: an evaluation of optimized best practices for genome analysis toolkit algorithms," *Genomics Inform*, vol. 18, no. 1, Mar. 2020. doi:10.5808/GI.2020.18.1.e10

[26] G. Carpi, L. Gorenstein, T.T. Harkins, M. Samadi, and P. Vats, "A GPU-accelerated compute framework for pathogen genomic variant identification to aid genomic epidemiology of infectious disease: a malaria case study." *Brief Bioinform*., vol. 23, no. 5, Sep. 2020. doi:10.1093/bib/bbac314