

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GENETİK TANI TEMELLİ YENİ NESİL DİZİLEME
VERİLERİNİN İŞLENMESİ VE RAPORLANMASININ
OTOMASYONU

Oğuzhan KALYON

YÜKSEK LİSANS TEZİ
Moleküler Biyoloji ve Genetik Anabilim Dalı
Moleküler Biyoloji ve Genetik Programı

Danışman
Prof. Dr. Muhammed Hamza MÜSLÜMANOĞLU

Şubat, 2020

TÜRKİYE CUMHURİYETİ
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

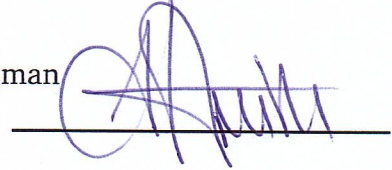
GENETİK TANI TEMELLİ YENİ NESİL DİZİLEME VERİLERİNİN
İŞLENMESİ VE RAPORLANMASININ OTOMASYONU

Oğuzhan KALYON tarafından hazırlanan tez çalışması 11.02.2020 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Moleküler Biyoloji ve Genetik Anabilim Dalı Moleküler Biyoloji ve Genetik Tezli Yüksek Lisans Programı **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Prof. Dr. Muhammed Hamza
MÜSLÜMANOĞLU
Yıldız Teknik Üniversitesi
Danışman

Jüri Üyeleri

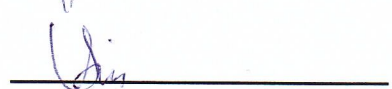
Prof. Dr. Muhammed Hamza MÜSLÜMANOĞLU, Danışman
Yıldız Teknik Üniversitesi



Dr. Öğr. Üyesi Emel ORDU, Üye
Yıldız Teknik Üniversitesi



Dr. Öğr. Üyesi Ümmühan DEMİR, Üye
İstanbul Medeniyet Üniversitesi



Danışmanım Prof. Dr. Muhammed Hamza MÜSLÜMANOĞLU sorumluluğunda tarafımda hazırlanan Genetik Tanı Temelli Yeni Nesil Dizileme Verilerinin İşlenmesi ve Raporlanmasının Otomasyonu başlıklı çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Oğuzhan KALYON
imza

*Değerli aileme
arkadaşlarıma
ve hocalarıma adanmıştır*

TEŞEKKÜR

Yüksek lisans öğrenimi gördüğüm sürede ve tez çalışmam esnasında yol gösterip değerli bilgileriyle çalışmalarına katkıda bulunan, genetik bilimine ve hayata bakışına tecrübeleriyle çok farklı yaklaşımlar katan değerli hocalarım sayın Prof.Dr. Muhammed Hamza MÜSLÜMANOĞLU ve Dr.Öğr.Üyesi Alper YILMAZ'a, çalışma süresince maddi ve manevi desteklerinden ötürü hocalarım sayın Doç.Dr.Emre TEPELİ'ye, Doç. Dr. Ahmet ULUDAĞ'a ve Dr. Hanifi AKTAŞ'a, teşvik ve yardımlarıyla her daim yanımda olan değerli arkadaşım Nur Sena ULUSKAN'a, ve eğitim hayatım boyunca üzerimde emeği geçen tüm hocalarıma, ayrıca bu süreçte maddi ve manevi koşulsuz olarak yanımda olan çok değerli aileme teşekkürü bir borç bilirim. . .

Oğuzhan KALYON

İÇİNDEKİLER

KISALTMA LİSTESİ	vii
ŞEKİL LİSTESİ	ix
TABLO LİSTESİ	x
ÖZET	xi
ABSTRACT	xiii
1 Giriş	1
1.1 Literatür Özeti	1
1.2 Tezin Amacı	2
1.3 Hipotez	2
2 Genel Bilgiler	3
2.1 Mutasyon	3
2.1.1 Mutasyon Tipleri	3
2.2 Moleküler Genetik Analiz Yöntemleri	7
2.2.1 Hedef PCR	8
2.2.2 Sanger Dizileme	8
2.2.3 Panel veya Yolak Dizileme	8
2.2.4 Tüm Ekzom Sekanslama (WES) ve Tüm Genom Sekanslama (WGS)	8
2.3 Yeni Nesil Dizileme Verilerinin Analizi	9
2.3.1 Ham Verinin Kalite Değerlendirmesi ve Temizlenmesi	9
2.3.2 Genoma Eşleme	10
2.3.3 Varyant Tespiti	11
2.3.4 Tespit Edilen Varyantların Anotasyonu	11
2.4 İş Akışı Yönetim Sistemi	13
2.5 YND Çalışma ve Analizlerindeki Zorluklar	14
3 Materyal ve Metod	15
3.1 Kullanılan Sisteme Genel Bakış	15

3.2	İş Akışı ve Sistem	15
3.2.1	Yapılandırma Dosyası	15
3.2.2	Snakefile Dosyası ve Kurallar	18
4	Sonuç ve Tartışma	29
4.1	Bulgular	29
4.1.1	Anotasyon Raporu	29
4.1.2	Kalite Kontrol Raporu	31
4.1.3	IGV Raporu	32
4.2	Tartışma	33
4.3	Sonuç	34
	Kaynakça	35
	Tezden Üretilmiş Yayınlar	40

KISALTMA LİSTESİ

aCGH	Array-Karşılaştırmalı Genomik Hibridizasyon (Microarray-Based Comparative Genomic Hybridization)
ACMG	Amerikan Tıbbi Genetik ve Genom Koleji (American College of Medical Genetics and Genomics)
DNA	Deoksiribo Nükleik Asit (deoxyribonucleic Acid)
ExAC	Ekzom Toplama Konsorsiyumu (The Exome Aggregation Consortium)
Fastq	Hızlı Adaptif Büzüşme Eşiği Algoritması – Kalite skoru (Fast Adaptive Shrinkage Threshold algorithm – Quality Score)
FISH	Floresan In Situ Hibridizasyon (Fluorescence In Situ Hybridization)
GATK	Genom Analizi Araç Seti (Genome Analysis Toolkit)
gnomAD	Genom Toplama Veritabanı (Genome Aggregation Database)
HGMD	İnsan Gen Mutasyonu Veri Tabanı (The Human Gene Mutation Database)
HGVS	İnsan Genom Varyasyon Topluluğu (Human Genome Variation Society)
HPO	İnsan Fenotip Ontolojisi (Human Phenotype Ontology)
IGV	Entegre Genomik Görüntüleyici (Integrated Genomics Viewer)
JSON	JavaScript Nesnesi Gösterimi (JavaScript Object Notation)
MLPA	Çoklu Ligasyona Bağlı Prob Amplifikasyonu (Multiplex Ligation-dependent Probe Amplification)
NIH	Ulusal Sağlık Enstitüleri (National Institutes of Health)
OD	Otozomal Dominant
OMIM	İnsanlarda Mendelyan Kalıtım - Çevrimiçi (Online Mendelian Inheritance in Man)
OR	Otozomal Resesif

PCR	Polimeraz Zincir Reaksiyonu (polymerase Chain Reaction)
RNA	Ribo Nükleik Asit (Ribonucleic Acid)
SNP	Tek Nükleotid Polimorfizmi (Single Nucleotide Polymorphism)
SVD-WG	Dizi Varyant Açıklaması Çalışma Grubu (Sequence Variant Description Working Group)
VEP	Değişken Etki Tahmini (Variant Effect Predictor)
VCF	Varyant Belirleme Formatı (Variant Caller Format)
WES	Tüm Ekzom Sekanslama (Whole Exome Sequencing)
WGS	Tüm Genom Sekanslama (Whole Genome Sequencing)
XBD	X-Bağlı Dominant
XBR	X-Bağlı Resesif
YAML	Yine Başka Bir Biçimlendirme Dili (Yet Another Markup Language)
YND	Yeni Nesil Dizileme

ŞEKİL LİSTESİ

Şekil 2.1 A) Yanlış Anlamlı Mutasyon B) Anlamsız Mutasyon	4
Şekil 2.2 A) İnsersiyon B) Delesyon	5
Şekil 2.3 A) Duplikasyon B) Çerçeve Kayması Mutasyonu	6
Şekil 2.4 Okuma Sonlarına Doğru Gerçekleşen Düşük Kaliteyi Gösteren Kalite Kontrol Rapor Örneği	10
Şekil 2.5 Referans Genoma Eşlenen Okumalardan Varyantların Tespiti	12
Şekil 2.6 YND ile Genotiplendirme	12
Şekil 3.1 Sistemin Genel Şeması	16
Şekil 3.2 GATK En İyi Uygulaması İş Akışı	22
Şekil 4.1 Snakemake İle Gerçekleştirilen Örnek Bir Çalışmanın Süresi ve İşlem Sayısı	30
Şekil 4.2 Örnek Kalite Kontrol Rapor İçeriği	32
Şekil 4.3 Örnek IGV Raporu	33

TABLO LİSTESİ

Tablo 2.1 Eşleme Algoritmalarının Karşılaştırılması	10
Tablo 3.1 Yapılandırma Dosyası	17
Tablo 3.2 Snakefile Dosyası	18
Tablo 3.3 Başlangıç Kuralı	19
Tablo 3.4 Kalite Kontrol Kuralı	20
Tablo 3.5 Genoma Eşleme Kuralı	21
Tablo 3.6 Varyant Tespiti Kuralı (GATK)	23
Tablo 3.7 Varyant Tespiti Kuralı (freebayes)	24
Tablo 3.8 Varyant Filtreleme Kuralı	25
Tablo 3.9 Anotasyon Kuralı (SnEff)	25
Tablo 3.10 Anotasyon Kuralı (VEP)	26
Tablo 3.11 Annotasyon Çıktı Dosyasının Düzenlenme Kuralı	26
Tablo 3.12 Kural Dosyasının Çalışması İçin Gerekli Komut Dosyası	27
Tablo 3.13 Tespit Edilen Varyantların Görüntü Raporu Kuralı	28

Genetik Tanı Temelli Yeni Nesil Dizileme Verilerinin İşlenmesi ve Raporlanmasının Otomasyonu

Oğuzhan KALYON

Moleküler Biyoloji ve Genetik Anabilim Dalı

Yüksek Lisans Tezi

Danışman: Prof. Dr. Muhammed Hamza MÜSLÜMANOĞLU

Yeni Nesil Dizileme (YND) teknolojisi, hızlı ve doğru tanıya olanak sağlayan genetik araştırmaları önemli ölçüde güçlendirmiştir. Bununla birlikte, üretilen verilerin artması ve analiz basamaklarının karmaşık hale gelmesine bağlı olarak analizler zorlaşmış ve hataya açık hale gelmiştir. Ayrıca, yapılan analizlerin tekrarlanabilirliği ve geriye dönük olarak denetlenmesi imkansız hale gelmektedir. Söz konusu zorluklara çare olarak yazılım geliştiriciler tarafından kullanılan iş akışı yönetimi programları bulunmakla beraber bu programların biyologlar tarafından kullanımı oldukça zordur. Ancak, Python tabanlı iş akışlarının yönetimini sağlayan Snakemake, sayısız biyoinformatik yazılımın kurulumunu veya yapılandırılmasını kolay ve zahmetsiz hale getirirken, kullanım açısından daha kolaydır. Bu çalışmada, YND verilerinde Snakemake yardımıyla varyant analizini programlama gerektirmeden kolaylaştırmak ve daha kullanışlı raporlar elde etmek hedeflenmiştir. Bu amaçla, GATK'ın tavsiye edilen varyant tespiti uygulamaları kullanılarak oluşturulan mevcut bir Snakemake iş akışı modifiye edilmiş, ENSEMBL Varyant Etki Predictor (VEP) ile varyantlar değerlendirilmiş ve Web tabanlı, interaktif son rapora Integrative Genome Viewer (IGV) yazılımı entegre edilmiştir. Bu yaklaşım tekrarlanabilir ve taşınabilir bir çözüm sunmaktadır, bu yüzden kişisel bir dizüstü bilgisayarda veya büyük bir sunucuda kolayca uygulanabilir. Snakemake paralel işleri yürütebildiğinden, çok işlemcili bilgisayarlarda SNP analizi paralel şekilde daha hızlı tamamlanabilir. Daha da önemlisi, bir iş akışı yaklaşımı kullanmak çok sayıda analiz gerçekleştirilirken insan kaynaklı hataları büyük ölçüde önleyecektir. Sonuç olarak, biyoloji veya ilgili alanlarda uzmanlaşmış bir kullanıcı, programlama bilgisi gerekmeden, geliştirdiğimiz

iş akışını kullanarak bir çok ham YND verisini işleyebilir ve kullanışlı, interaktif varyant raporları ile daha doğru, çabuk ve tekrarlanabilir sonuçlara ulaşabilir.

Anahtar Kelimeler: Yeni nesil dizileme, Snakemake, genetik test

ABSTRACT

Automation of Genetic Diagnosis Based Next Generation Sequencing Data Processing and Reporting

Oğuzhan KALYON

Department of Molecular Biology and Genetics

Master of Science Thesis

Advisor: Prof. Dr. Muhammed Hamza MUSLUMANOGLU

Next Generation Sequencing (NGS) technology has significantly boosted genetic research allowing rapid and accurate diagnosis. However, due to vast amount of data and complex workflow steps, the variant calling from NGS data is difficult and susceptible to errors. Furthermore, the reproducibility and auditing of the whole process is almost impossible. There are workflow management programs used by software developers to address these challenges, but these programs are difficult to use by biologists. However, Snakemake, a Python-based workflow manager, facilitates installation and configuration of numerous bioinformatics software and with minimum programming skills. This study aims to simplify the variant analysis with interactive reports using Snakemake in NGS data without writing scripts or codes. For this purpose, an existing Snakemake workflow created using GATK's recommended variant detection applications has been modified. We added annotations from ENSEMBL Variant Effect Predictor (VEP) and interactive reports via Integrative Genome Viewer (IGV) software into the final report. This approach offers a reproducible and portable solution, so it can be easily implemented on a personal laptop or a large server. Since Snakemake can run parallel jobs, SNP analysis can be completed faster in parallel on multiprocessor computers. More importantly, using a workflow approach will greatly avoid human-induced errors when analyzing lots of samples. As a result, a user specialized in biology or related fields can process a lot of raw NGS data using our workflow without programming knowledge, and achieve more accurate, quick and reproducible results with useful, web-based interactive variant reports.

Keywords: Ngs, Snakemake, genetic test

1.1 Literatür Özeti

Organizmalardaki gelişim ve fonksiyonel süreçlerin büyük bir kısmı genler tarafından kontrol edilmektedir. Kodlanan genetik bilginin değişimi veya hasara uğraması mutasyon olarak adlandırılmaktadır. Genetik hastalıklar temelde tek gen hastalıkları, kromozomal bozukluklar ve multifaktöryel hastalıklar olmak üzere üç ana grupta sınıflandırılmaktadır. Günümüzde kromozomal anomalileri saptamak ve kopya sayısı değişimlerini değerlendirmek amacı ile genellikle konvansiyonel karyotipleme, Array-Karşılaştırmalı Genomik Hibridizasyon (aCGH), Floresan In Situ Hibridizasyon (FISH) ve Çoklu Ligasyona Bağlı Prob Amplifikasyonu (MLPA) yöntemleri kullanılmaktadır. Bunun yanı sıra nadir ve yaygın nokta mutasyonlarının analizi için genellikle; hedef Polimeraz Zincir Reaksiyonu (PCR), Sanger Dizileme, Panel veya Yolak Dizileme (Pathway Sequencing), Tüm Ekzom Sekanslama (WES) ve Tüm Genom Sekanslama (WGS) yöntemleri kullanılmaktadır. Ayrıca Yeni Nesil Dizileme (YND) dizileme teknolojileri metilasyon paterni, alternatif ekzon-intron kesimi, küçük RNA, gen ifadesi ve genomdaki yeniden düzenlenmeler gibi birçok genetik ve biyolojik süreçlerin analizlerinde kullanılmaktadır. Bu analizler metabolomik, proteomik, transkriptomik gibi alanların gelişmesini oldukça hızlandırmıştır. Omik analizler, hücredeki süreçlerin bütünsel olarak anlaşılmasında, patolojik durumların etiyolojisinin aydınlatılmasında ve hedef moleküllerin keşfinde oldukça önemli hale gelmiştir. Özellikle Mendel kalıtmalı hastalıkların araştırılması için gerçekleştirilen YND verilerinin analizleri ham verinin kalite değerlendirmesi ve temizlenmesi, genoma eşleme, varyant tespiti ve tespit edilen varyantların anotasyonu basamaklarından oluşmaktadır. Bu işlemler iş akışı yönetim programları ile otomatikleştirilebilir ve tekrarlanabilirlik sağlanabilir. YND analizlerinde kütüphane hazırlama veya dizileme sırasında meydana gelen hatalardan ve genomun yapısından kaynaklı, örneğin yüksek dizi homolojisi gösteren bölgeler, zorluklar yaşanmaktadır. Ayrıca algoritmaların yazılım bilgisi gerektirmesi ve birçok aşamanın olması da YND analizlerini zorlaştırmaktadır.

1.2 Tezin Amacı

Gelişen teknolojiler ile birlikte üretilen bilgi miktarı her geçen sene daha da artmaktadır ve “büyük veri” çağı yaşanmaktadır. Dizileme teknolojileri gibi teknolojilerin gelişmesiyle birlikte biyolojik verilerin de büyüklüğü artmış, işlenmesi ve anlamlandırılması zor hale gelmiştir. Biyoinformatik araçların gelişmesi, ayrıca yapay zeka, makine öğrenmesi ve derin öğrenme gibi kavramların biyolojik bilimlerle entegre olması verilerin işlenmesini kolaylaştırmaktadır. Halihazırda elde edilen dizileme verilerinin temizlenmesi, referans genoma eşlenmesi ve varyant analizleri ile ilgili birçok program mevcuttur. Yazılım bilgisi olmayan bir araştırmacının elde ettiği yeni nesil dizileme sonuçlarını teknik destek almadan şahsi bilgisayarında analiz etmesi oldukça zordur. Bu problemi çözmek maksadıyla, özellikle varyant analiz için birden fazla aşama gerektiren analizlerin kolayca gerçekleştirilebilmesi için iş akışı hazırlanması amaçlanmaktadır.

1.3 Hipotez

Yeni Nesil Dizileme işlemlerinin ucuzlaması ve birçok platformun geliştirilmesinden sonra genetik tanı amacı ile kullanımı oldukça artmıştır. Bunun yanında platformlardan elde edilen ham verilerin işlenmesi için geliştirilen algoritmaların sayısı da bu gelişmelere bağlı olarak artmıştır. Ancak algoritmaların kullanılması için kodlama bilgisi gerekmektedir. Ayrıca her bir yazılımın kurulumu, birbiri ile olan uyumu yapılan analizleri daha da güç hale getirmektedir. Özellikle genetik tanı amaçlı çalışmalarda birden fazla verinin aynı anda işlenmesi ve hızlı bir şekilde analizlerin tamamlanması gerekmektedir. Ayrıca analizler süresince verilerin çok olması nedeni ile tam otomatik olarak, manuel olarak müdahale gerektirmeden, çalışması önemlidir. Manuel işlem basamağı arttıkça insan kaynaklı hatalar da artacaktır. Çalışmamızdaki önermemiz; Python tabanlı Cconda ortamında Snakemake ile iş akışı oluşturulup kullanılması halinde birçok ham verinin aynı anda hızlı bir şekilde, programlama bilgisi gerektirmeden ve hataya imkan vermeden analizlerinin yapılması mümkün hale gelecektir.

2 Genel Bilgiler

2.1 Mutasyon

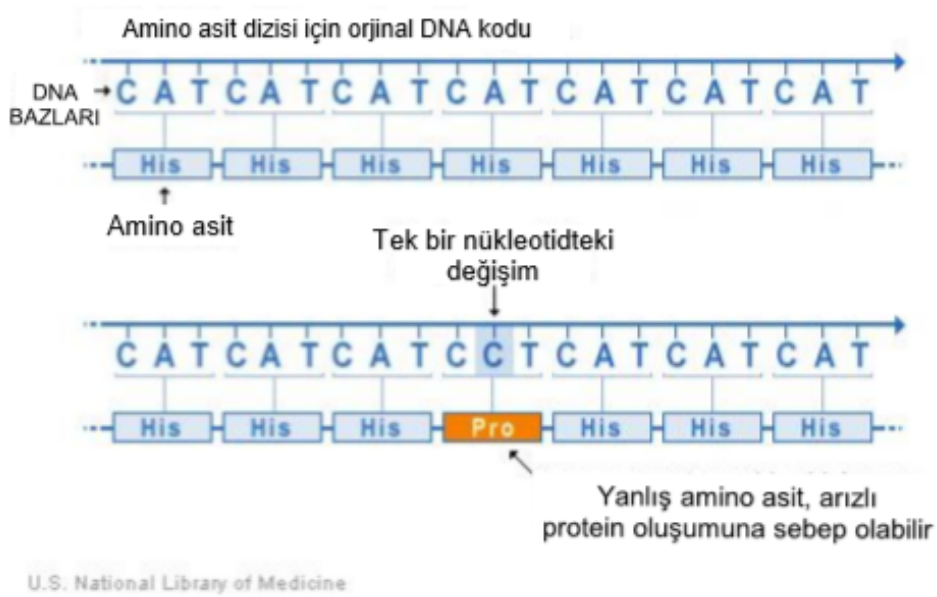
Organizmalardaki gelişim ve fonksiyonel süreçlerin büyük bir kısmı genler tarafından kontrol edilmektedir. Kodlanan genetik bilginin değişimi veya hasara uğraması mutasyon olarak adlandırılmaktadır. DNA dizisindeki değişimler popülasyonda sık görülüyorsa polimorfizm adını almaktadır. Mutasyonlar, genler tarafından kodlanan proteinin yapısında değişime veya gen ifadesinde bir azalmaya neden olmaktadır. Organizmanın dış görünümü, gelişimi ve davranışları fenotip olarak adlandırılırken taşıdığı genler ve çevresel etkileşimi genotip olarak tanımlanmaktadır. Organizmalardaki temel genetik farklılıklardan biri her bir kromozomun tek mi yoksa iki kopyamı olarak taşınmasıyla ilgilidir. Tek kopya taşınanlar haploid, iki set olarak bulunanlar diploid olarak adlandırılır. Basit tek hücreli canlıların birçoğu haploid iken kompleks yapıdaki çok hücreli organizmalar diploiddir. Diploid organizmalardaki genlerin aynı lokusa yerleşmiş alternatif kopyaları alel olarak adlandırılır. Diploid organizmalarda her iki alelde meydana gelen genetik değişimler homozigot, tek bir alelde meydana gelen genetik değişimler ise heterozigot olarak adlandırılmaktadır. Ayrıca tek kopya bulunmasına rağmen fenotipik olarak etkisini gösteren aleller dominant olarak tanımlanır. Dominant alelin varlığında fenotipte etkisi görülmeyen aleller resesif olarak adlandırılır ve resesif alellerin fenotipik olarak ortaya çıkabilmeleri için homozigot durumda, yani iki kopya halinde bulunmaları gerekir. Resesif mutasyonların genellikle kodlanan proteinlerin fonksiyonlarını azaltıcı, dominant mutasyonların ise fonksiyonlarını artırıcı etki gösterdiği bildirilmiştir. [1]

2.1.1 Mutasyon Tipleri

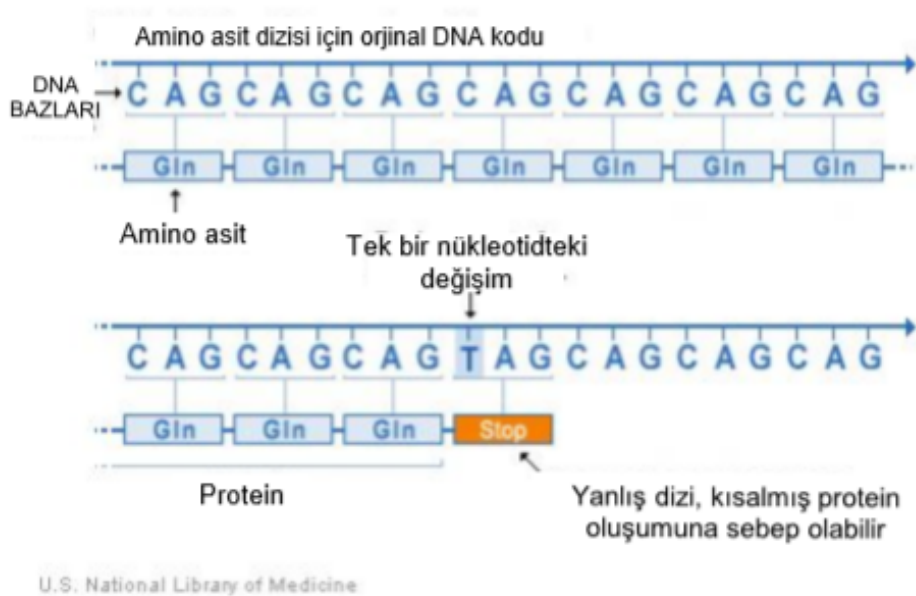
Genlerin DNA dizileri birden fazla mekanizma ile değişebilir. Mutasyonlar nasıl ve nerede gerçekleştiklerine bağlı olarak insan sağlığı üzerinde etki gösterirler. Kodlanan proteindeki etkilerine göre veya sebep oldukları değişimlere göre mutasyonları aşağıdaki gibi özetleyebiliriz:

- Yanlış Anlamlı Mutasyon: DNA dizisindeki bir bazlık değişim ile kodlanan bir amino asidin değişikliğe uğradığı mutasyon çeşididir (Şekil 2.1A).
- Anlamsız Mutasyon: DNA dizisindeki bir bazlık değişim ile prematür durdurma kodonu oluşmasına dolayısıyla fonksiyon görmeyen kısa bir protein üretimine neden olmaktadır (Şekil 2.1B).

A Yanlış Anlamlı Mutasyon



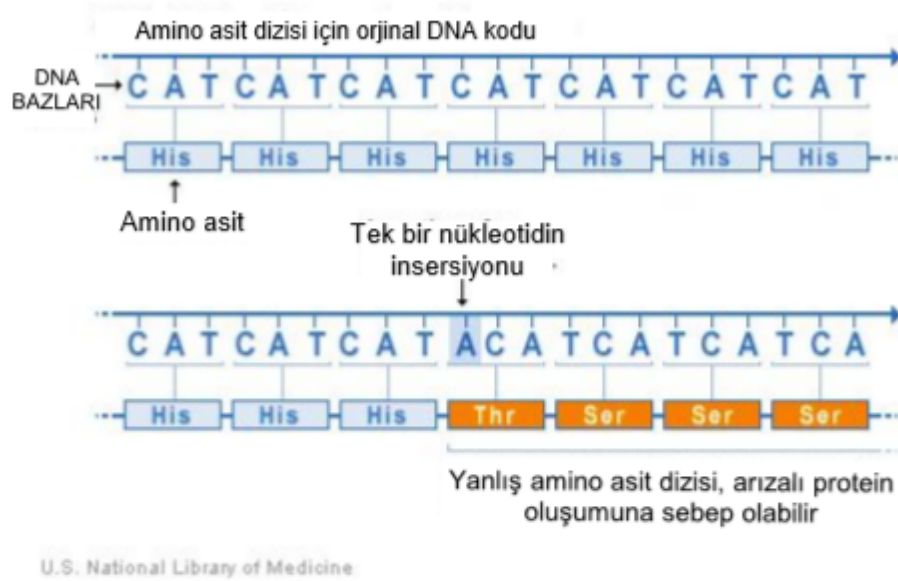
B Anlamsız Mutasyon



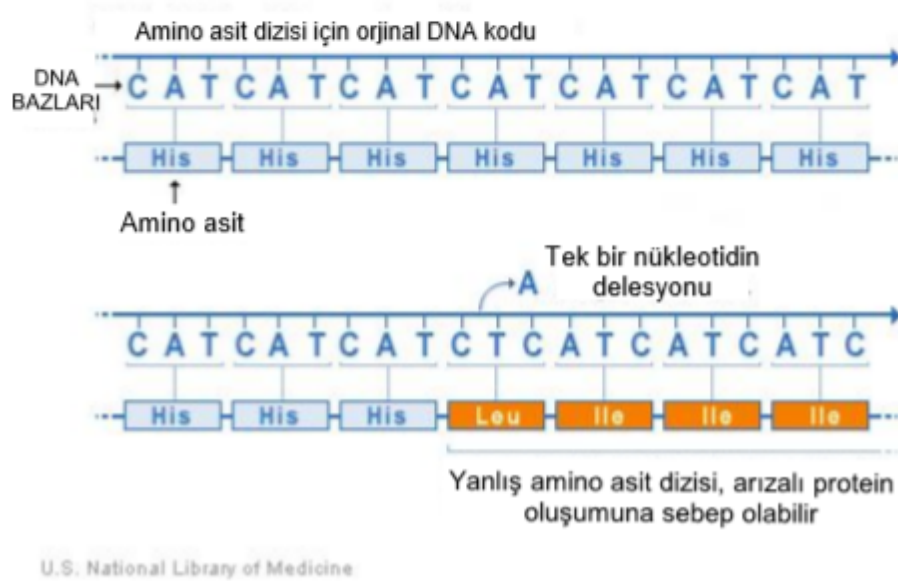
Şekil 2.1 A) Yanlış Anlamlı Mutasyon B) Anlamsız Mutasyon
NIH veri tabanından modifiye edilerek alınmıştır.[2]

- İnsersiyon: DNA dizisine bir DNA parçasının eklenmesiyle oluşan DNA dizisindeki artış fonksiyonel olarak işlevsel olmayan protein üretimine neden olmaktadır (Şekil 2.2A).
- Delesyon: DNA dizisindeki çok sayıdaki veya tek bazlık kayıpların kodlanan proteinin fonksiyonunda değişme yol açtığı mutasyon çeşididir (Şekil 2.2B).

A İnsersiyon Mutasyonu

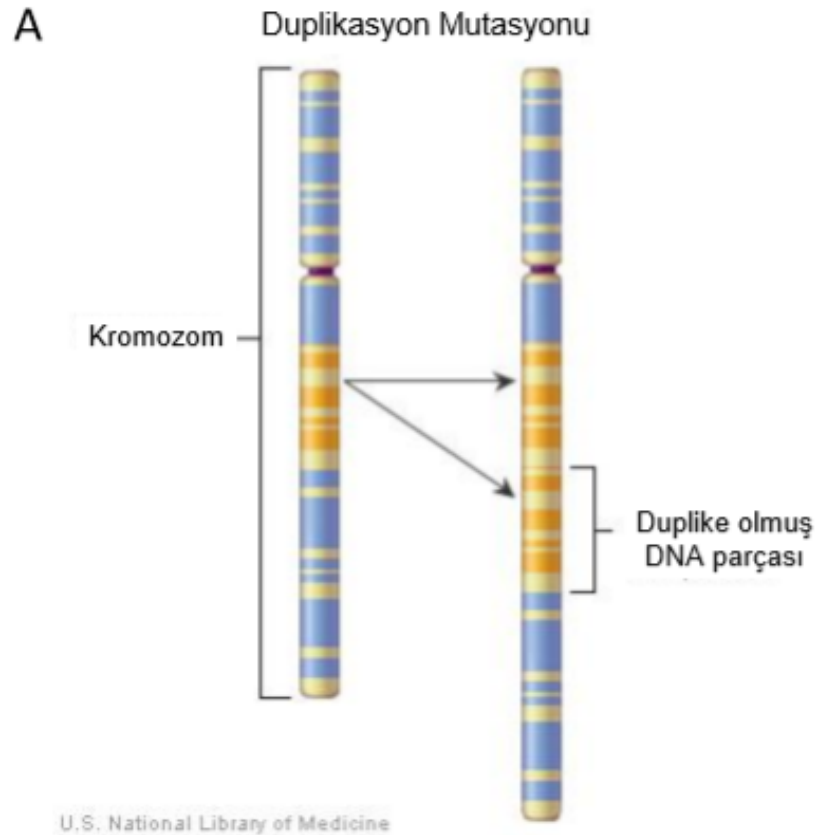


B Delesyon Mutasyonu

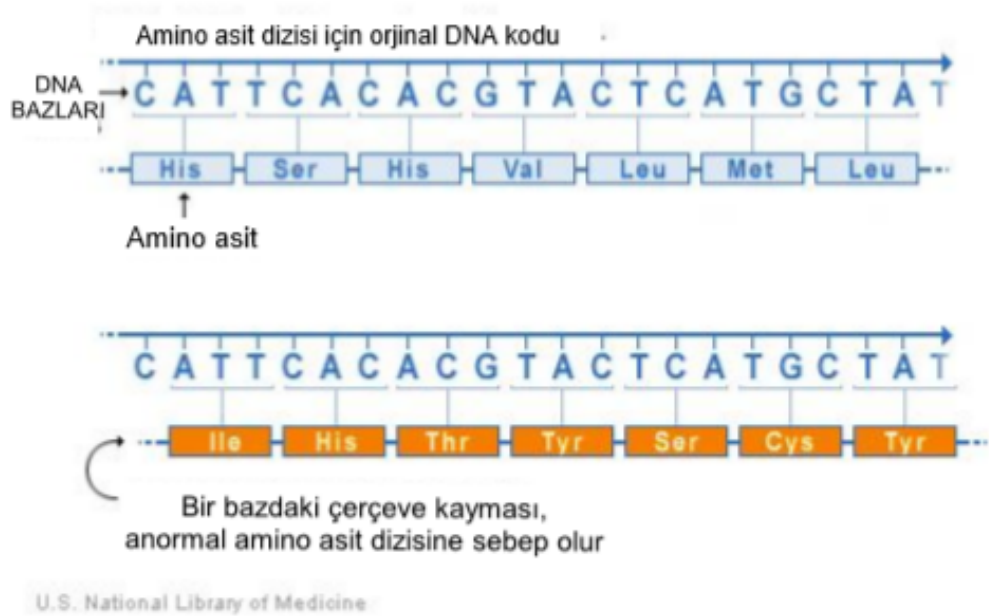


Şekil 2.2 A) İnsersiyon B) Delesyon
NIH veri tabanından modifiye edilerek alınmıştır. [2]

- Duplikasyon: Herhangi bir DNA parçasının anormal olarak genomda birden fazla bulunma durumudur (Şekil 2.3A).



B Çerçeve Kayması Mutasyonu



Şekil 2.3 A) Duplikasyon B) Çerçeve Kayması Mutasyonu
NIH veri tabanından modifiye edilerek alınmıştır.[2]

- Çerçeve Kayması Mutasyonu: DNA dizisindeki artış veya azalmaya bağlı olarak genin okuma çerçevesinin değiştiği ve genellikle fonksiyonel olmayan protein üretimiyle sonuçlanan mutasyon türüdür. İnsersiyon, delesyon veya duplikasyon sonucunda oluşabilir (Şekil 2.3B).
- Ekzon İntron Kesim Bölgesi Mutasyonları: Ekzon intron kesim bölgesinde meydana gelen değişimlerdir. RNA kesimini etkileyerek fonksiyonel olmayan protein üretimine yol açar.[2]

2.2 Moleküler Genetik Analiz Yöntemleri

Genetik hastalıklar temelde tek gen hastalıkları, kromozomal bozukluklar ve multifaktöryel hastalıklar olmak üzere üç ana grupta sınıflandırılmaktadır. Tek gen veya Mendelian hastalıkları genin DNA dizisindeki değişimler sonucunda oluşan ve otozomal resesif (OR), otozomal dominant (OD), X-bağlı resesif (XBR), X-bağlı dominant (XBD) veya Y-bağlı (holandrik) kalıtım gösterebilen hastalıklardır. Kromozomdaki sayısal ve yapısal değişimlere bağlı olarak oluşan hastalıklar kromozomal hastalıklar olarak adlandırılır. Hastalığa sebep olan mutasyonların tespiti için moleküler ve sitogenik yöntemler kullanılmaktadır. Doğru tanı hastaların tedavileri ve uygun genetik danışma için oldukça önemlidir. Genetik testler nadir hastalıkların yanı sıra multifaktöryel hastalıkların aydınlatılması ve kişisel tıp amacıyla kullanılmaktadır. Günümüzde klinik genetik testler aşağıda listelenen amaçlar için kullanılmaktadır:

- Yeni Doğan Taraması : Fenilketonüri, anemi ve kistik fibroz gibi resesif genetik hastalıkların taranması amacıyla uygulanmaktadır.
- Tanı Testleri: Semptomatik bireylerin tanılarının konfirmasyonu amacı ile yapılmaktadır.
- Taşıyıcılık Testleri: Asemptomatik bireylerde bir veya birden fazla resesif gen mutasyonlarının taranması olarak tanımlanmaktadır.
- Tahmin Testleri: Kalıtsal olan ve genetik değişimlerin neden olduğu erişkin yaş başlangıçlı hastalıkların taranması amacıyla yapılmaktadır.
- Farmakogenetik: Genomda ilaç yanıtı ve kullanım dozu ile ilişkilendirilmiş değişimlerin analizlerine dayanır.[3]

Günümüzde kromozomal anomalileri saptamak ve kopya sayısı değişimlerini değerlendirmek amacı ile genellikle konvansiyonel karyotipleme,

Array-Karşılaştırmalı Genomik Hibridizasyon (aCGH), Floresan In Situ Hibridizasyon (FISH) ve Çoklu Ligasyona Bağlı Prob Amplifikasyonu (MLPA) yöntemleri kullanılmaktadır. Bunun yanı sıra nadir ve yaygın nokta mutasyonların analizi için genellikle aşağıdaki yöntemler kullanılmaktadır;

2.2.1 Hedef PCR

Amplifikasyonun ardından restriksiyon enzim kesimi, hibridizasyon işlemleri veya proba dayalı gerçek zamanlı PCR yöntemleri ile alel spesifik mutasyon analizleri mümkündür.

2.2.2 Sanger Dizileme

Klinik genetik çalışmalarda altın standart olarak kabul gören yöntem kapiller elektroforez tekniği ile gen spesifik dizilemeyi mümkün kılar.

2.2.3 Panel veya Yolak Dizileme

YND yöntemlerinin gelişmesi ile birlikte aynı metabolik yolda zenginleşen veya aynı fenotipe sebep olan genlerin aynı anda multipleks PCR yöntemi ile amplifikasyonu ve dizilemesine dayalı yöntem genetik tanıyı zaman ve maliyet açısından daha etkili hale getirir. [4]

2.2.4 Tüm Ekzom Sekanslama (WES) ve Tüm Genom Sekanslama (WGS)

Gelişen teknoloji ve düşen maliyet ile kullanımı yaygın hale gelen YND platformları aynı anda genomda kodlama yapan tüm bölgelerin analizine veya genomun tamamının analizine olanak sağlamaktadır. Özellikle WES'in Mendel kalıtmalı hastalıkların tanısındaki önemi yapılan bir çok analiz ile ortaya konulmuştur. Klinik genetik çalışmalarında özellikle açıklanamayan fenotiplerin ortaya konulmasında yaygın olarak kullanılan WES'in ilerleyen yıllarda maliyet düşüşü ile birlikte yerini WGS'e bırakacağı tahmin edilmektedir. WGS çalışmaları ile birlikte intronik bölgelerdeki genetik değişimlerin fenotipe etkisinin daha net ortaya konulabileceği düşünülmektedir. Yapılan WES çalışmalarının kalitesinin platformlar arasında farklılık göstermesi ve bir çok limitasyondan ötürü saptanan değişimlerin Sanger dizileme ile confirmasyonunun yapılması gerekmektedir. Ayrıca hastaların klinik bulguları ile uyumlu olduğu düşünülen aday genetik değişimlerin biyolojik akrabalarda analizlerinin yapılması ve fenotip genotip korelasyon çalışması aday mutasyonların değerlendirilebilmesi açısından oldukça önemlidir. [5][6]

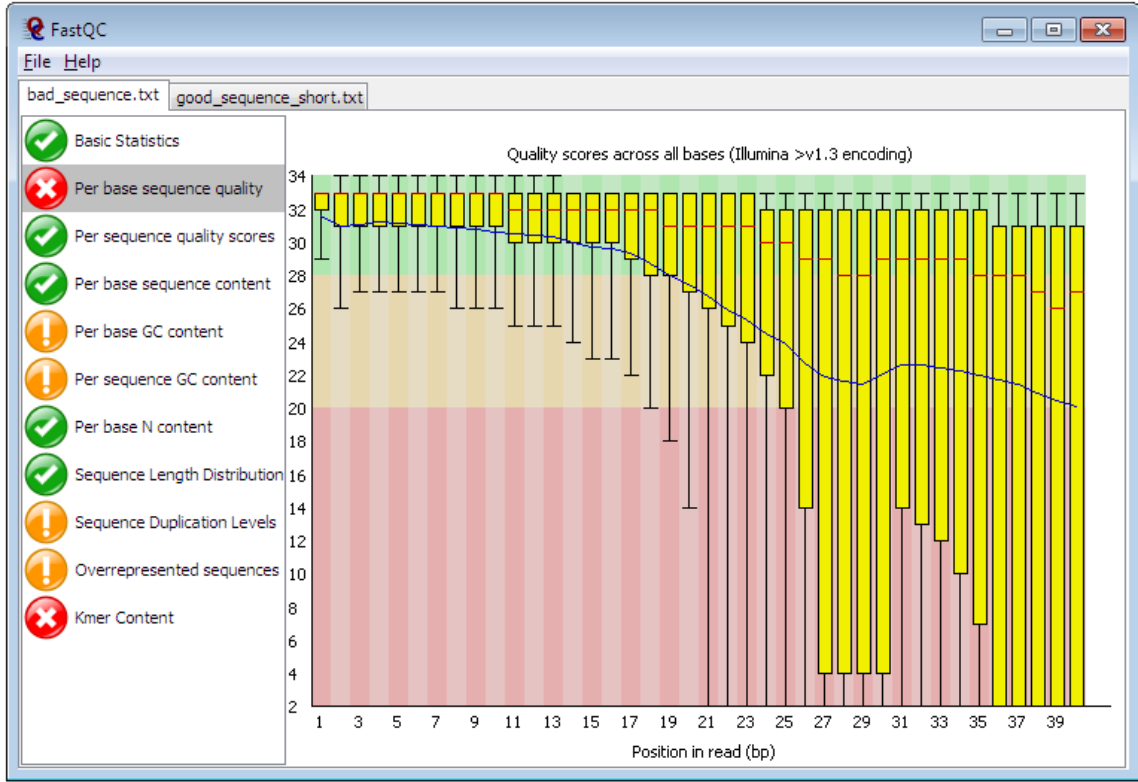
Ayrıca YND dizileme teknolojileri metilasyon paterni, alternatif ekzon-intron kesimi, küçük RNA, gen ifadesi ve yeniden düzenlenmeler gibi birçok genetik ve biyolojik olguların analizlerinde kullanılmaktadır. Bu analizler metabolomik, proteomik, transkriptomik gibi alanların gelişmesini oldukça hızlandırmaktadır. Omik analizleri hücredeki süreçlerin bütünsel olarak anlaşılmasında, patolojik durumların etiolojisinin aydınlatılmasında ve hedef moleküllerin keşfinde oldukça önemlidir.

2.3 Yeni Nesil Dizileme Verilerinin Analizi

Platformdan elde edilen ham veriler bir dizi basamaklardan geçirilerek genetik değişimler doğru bir şekilde elde edilmeye çalışılır. Elde edilen verilerin kalite analizlerinin yapılması, gerekli temizliklerin yapılması ve aşamaların kontrollü bir şekilde ilerletilmesi doğru sonuçlar elde etmek açısından oldukça önemlidir. Ayrıca dizilenen genom bölgesinde değişim saptanan bölgenin okuma sayısının en az 15x olması sonucun doğruluğu açısından önemlidir. [7]

2.3.1 Ham Verinin Kalite Değerlendirmesi ve Temizlenmesi

Dizileme sırasında meydana gelebilen kimyasal prosesler, makine hataları, deneysel hatalar gibi nedenlerle okuma kaliteleri olumsuz etkilenmektedir. Örneğin illumina platformlarından elde edilen verilerde okuma sonlarına doğru kalitenin düştüğü bilinmektedir (Şekil 2.4’de örnek bir kalite değerlendirme raporu gösterilmiştir). Elde edilen Fastq verilerinin kalite analizi gereklidir ve bunun için birçok algoritma geliştirilmiştir. Bunlardan bazıları FaQCs [8], FastQC [9], FASTX-Toolkit [10] ve PRINSEQ-lite [11] gibi algoritmalarlardır. Bu yazılımların birbirinden üstün veya zayıf yanları vardır ve literatürde bu programları 3’ son kesimi, 5’ son kesimi, k-mer içeriği, farklı filtreleme özellikleri bakımından karşılaştıran çalışmalar bulunmaktadır [8]. Yapılacak analizin niteliğine göre daha üstün özellikleri olan algoritmanın seçilmesi ve kullanılması tavsiye olunmaktadır. Yapılan kalite değerlendirmesinin ardından düşük kaliteli okumalar ve dizileme için kullanılan adaptörler uzaklaştırılmalıdır. Adaptör temizliği aşaması içinde birçok algoritma geliştirilmiştir, bunlardan önemli olanlardan bazıları Cutadapt [12], Trimmomatic [13] ve ConDeTri [14] algoritmalarıdır. Geliştirilen algoritmaların hangi deneysel çalışmalarda hangisinin daha iyi sonuç verdiğine dair birbiri ile ayrıntılı karşılaştırmaları literatürde yapılan çalışmalar ile ortaya konulmuştur [15]. Özetle analiz edilcek YND verilerinin deneysel süreçleri ve elde edilmek istenen çıktılar dikkate alınarak kalite kontrolü yapılması ve uygun bir algoritma ile gerekli temizliklerin yapılması verilerden üretilcek bilgilerin doğruluk ve hassaslığı açısından önemlidir.



Şekil 2.4 Okuma Sonlarına Doğru Gerçekleşen Düşük Kaliteyi Gösteren Kalite Kontrol Rapor Örneği

2.3.2 Genoma Eşleme

Yapılan dizileme sonrası elde edilen okumaların değerlendirilebilmesi için referans genoma eşleme yapılması her bir okumanın referans genomdaki yerinin belirlenmesi gereklidir. Oldukça kritik öneme sahip bu aşama için pek çok algoritma geliştirilmiş olup bunlardan en çok kullanılanları BWA [16], STAR [17], Bowtie 2 [18] ve TopHat'dir [19]. Birbirinden farklı özellik, avantaj ve dezavantajlara sahip olan algoritmalarından en uygun olanı, analizi yapılan veri, kullanılan bilgisayarın teknik özellikleri ve dizileme yapılan platform gibi birçok parametre göz önüne alınarak seçilmelidir. Literatürde yapılan karşılaştırma çalışmaları ile algoritmaların sınıflandırılmaları ortaya konulmuş süre, doğruluk ve hafıza kullanımı gibi birçok parametre açısından değerlendirilmişlerdir[20]. İlgili birkaç parametre ve farklı genom eşleme algoritmaları için sonuçlar Tablo 2.1 'de gösterilmiştir.

Tablo 2.1 Eşleme Algoritmalarının Karşılaştırılması [21]

	STAR	BWA-MEM	Segemehl	Bowtie2	BWA
Doğru Eşleme	%99,3	%98,9	%99,9	%98,7	%98,6
Hafıza Kullanımı	28,12 Gb	5,75 Gb	70,05 Gb	3,77 Gb	3,85 Gb
Süre	0,99sn	186,97sn	318,72sn	87,91sn	166,31sn

Genoma eşlemenin ardından özellikle varyant analizlerinde eşleme sonrası duplike

okumaların uzaklaştırılması ve baz kalite skorunun yeniden kalibrasyonu gibi doğruluğu arttıran işlemler uygulanır[22]. YND analizi için kütüphane hazırlanması sırasında PCR artefaktı olarak tanımlanan aynı zincirden çoğalan kirlilikler ortaya çıkmaktadır. Sonuçları etkilememesi için Picard [23] ile duplike okumaların uzaklaştırılması önerilmektedir [24]. Ayrıca bilinen insersiyon delesyon bölgelerinin etrafında lokal yeniden eşleme ve baz kalite skorunun yeniden kalibrasyonu varyant analizlerinin güvenilirliğini artırdığı literatürde yapılan çalışmalar ile ortaya konulmuştur [24]. Bu prosesler varyant analizi için geliştirilen GATK [25] algoritması ile uygulanabilir.

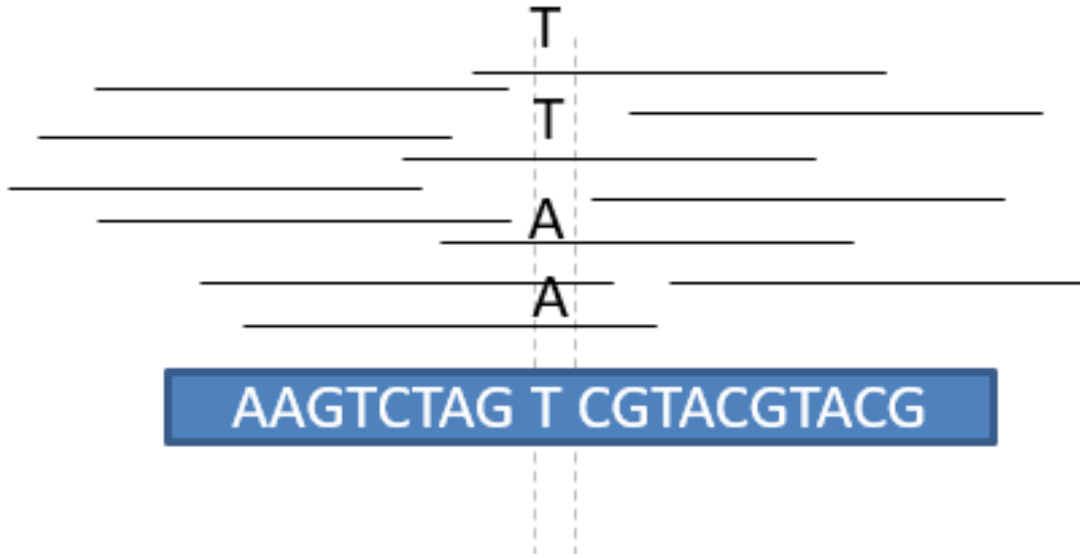
2.3.3 Varyant Tespiti

Dizilenen ve referans genoma eşlenen okumalar yapılan deneysel çalışmanın amacına göre analiz edilir. Mendel kalıtmalı hastalıkların tanısında kullanılan hedef dizileme, tüm ekzom ve genom dizileme verilerinin analizinde varyant listesi oluşturulur. Referans genoma eşlenen her bir okumanın koordinat başına farklı olan bazlar tespit edilir (Şekil2.5). Diğer analiz basamaklarında olduğu gibi varyant tespiti için algoritmalar geliştirilmiştir. Ayrıca YND bazlı varyant tespiti genotiplendirmeye (heterozigot/homozigot) olanak sağlamaktadır (Şekil2.6). Bunlardan en yaygın olarak kullanılanlar GATK [25], SAMtools [16], freebayes [26] ve DeepVariant [27] olarak gösterilmektedir. Literatürde adı geçen algoritmalar farklı platformdan üretilmiş test verileri ile karşılaştırma analizlerine tabi tutulmuş literatürde çokça yayın yapılmıştır. Günümüzde halen devam etmekte olan sistematik analizler en doğru platform ve iş akışını oluşturarak büyük verilerin güvenli analizini sağlamaya odaklanmaktadır [28, 29].

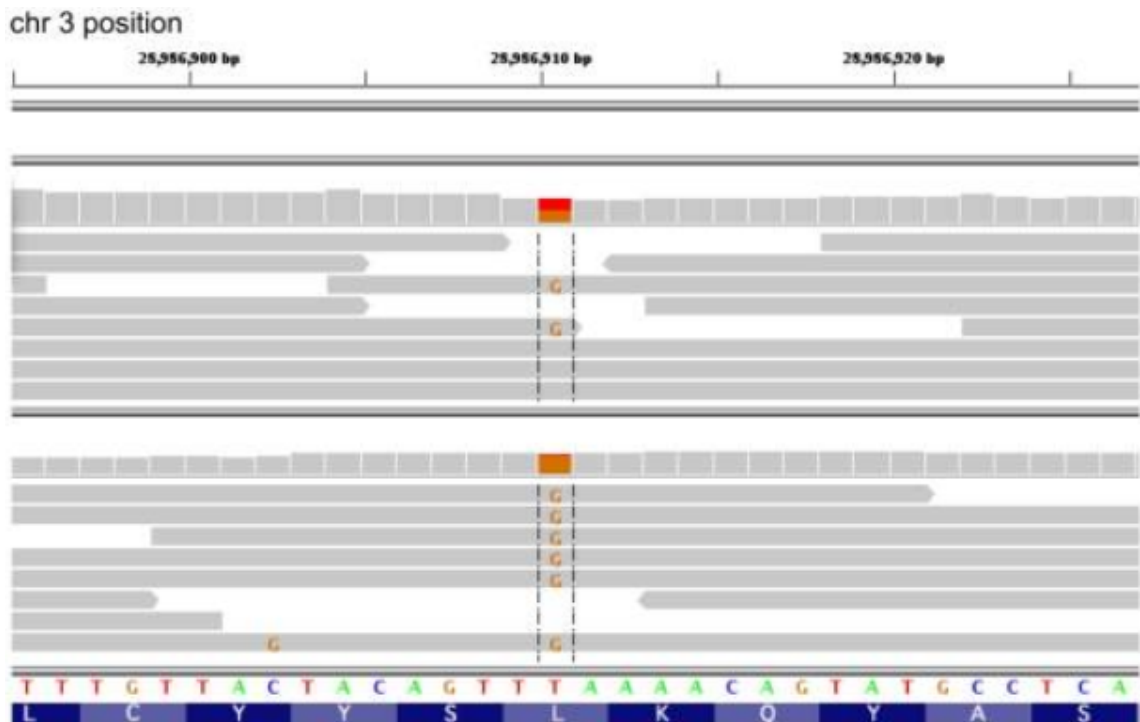
2.3.4 Tespit Edilen Varyantların Anotasyonu

YND çalışmalarından elde edilen genetik değişimlerin değerlendirip sınıflandırılması gerekmektedir. Değişimlerin değerlendirilebilmesi için;

- Değişimin transkriptte nerede meydana geldiği (ekzonik, intronik, ekzon-intron kesim bölgesi)
- Değişimin amino asit değişimine sebep olup olmadığı
- Aynı tipte mutasyonların ilgili gende tanımlı olup olmadığı
- Toplumda görülme sıklığı
- Değişimlerin etkilerini öngören in siliko algoritmaların değişimi nasıl sınıflandırdığı



Şekil 2.5 Referans Genoma Eşlenen Okumalardan Varyantların Tespiti



Şekil 2.6 YND ile Genotiplendirme

YND analizleri genotiplendirmeye de olanak sağlar. Üst panoda görüldüğü gibi değişim oranı yaklaşık yarı yarıya ise heterozigot, alt panoda görüldüğü gibi tüm okumalarda değişim gözleniyorsa homozigot olarak değerlendirilir.

gibi birçok parametre oldukça önemlidir. Genetik değişimlerin değerlendirmek amacı ile ACMG (American College of Medical Genetics and Genomics) [30] gibi klavuzlar literatürde bildirilmiştir. Saptanan değişimlerin sınıflandırıldığı ve ilişkili olduğu fenotiplerin toplandığı Clinvar [31], HGMD (The Human Gene Mutation

Database) [32] ve OMIM (Online Mendelian Inheritance in Man) [33] gibi yaygın kullanılan veritabanları mevcuttur. Yapılan toplum sıklığı çalışmalarında sağlıklı bireylerde gözlenen değişimlerin ve alel frekanslarının depolandığı ExAC (The Exome Aggregation Consortium) [34] ve gnomAD (The Genome Aggregation Database) [35] gibi veri tabanları mevcuttur. Ayrıca literatürde tanımlanmamış değişimlerin etkilerini öngören in siliko algoritmalar geliştirilmiştir. Bu algoritmalar değişimlerin fenotipik etkilerini tahmin etmek için oldukça önemlidir. 29 farklı algoritmanın tüm potansiyel eşanlamlı olmayan tek nükleotid varyantlarının tahmin verisini tutan dbNSFP gibi [36] veritabanları mevcuttur. Bir diğer önemli nokta saptanan değişimlerin uluslararası standartlarda raporlanmasıdır. Standardizasyonu ve konsensusu sağlamak amacı ile DNA, RNA ve protein düzeyinde meydana gelen değişimlerin raporlanması amacı ile SVD-WG (Sequence Variant Description Working Group) tarafından yazım terminoloji kuralları yayınlanmıştır [37]. Analiz için gerekli anotasyon bilgilerini derleyen, varyantları kategorize eden sınıflandıran ve önemli bilgileri sunan açık kaynak kodlu ANNOVAR [38], VEP [39] ve SnpEff [40] gibi algoritmalar geliştirilmiştir.

2.4 İş Akışı Yönetim Sistemi

Biyoinformatikte büyük ölçekli veri analizleri, birçok işlemin veya algoritmanın sırayla yürütülmesini içerir. Bu akış sırasında, bir adımın çıktısı olan dosyalar bir sonraki adımı beslemek durumundadır. Çok basamaklı bir analizde, özellikle çok örnek işlenmesi durumunda, hangi adımın tamamlandığı, hangi örnek için hangi adımın yapılması gerektiği gibi kararları manuel olarak vermek neredeyse imkansızdır ve hata yapma ihtimalini artırır. İş akışı yönetim sistemleri her bir örnek için her bir adım/algoritmanın çalışmasını otomatikleştirmeyi sağlar ve böylelikle tekrarlanabilirliği mümkün kılar. İş akışı programlarından Biopipe [41], Taverna [42], Galaxy [43], GeneProf [44] ve PegaSys [45] gibi sistemler öğrenimi kolay ve grafik arayüzleri sayesinde kullanımları kolaydır. Ruffus [46], GXP Make [47], Bpipe [48], NextFlow [49] ve Snakemake [50] grafik arayüz olmaksızın metin tabanlı düzenlemeler ile yönetilebilir. Snakemake sistemi GNU Make inşa sisteminden ilham alınarak geliştirilmiştir. Snakemake oldukça esnek bir sistemdir, girdi ve çıktı dosyaları iyi tanımlanmış uygulamalar kolayca sisteme eklenebilir. İş akışının tanımlandığı Snakefile dosyası ve Python kurulumu yapıldıktan sonra sistem kolayca çalıştırılabilir, bu sayede, bir cihazda geliştirilen bir iş akışı başka bir cihazda aynen çalıştırılabilir, yani taşınabilir. İş akışında herhangi bir değişiklik yapmadan paralel işlemleri gerçekleştirmek için işlemci sayısı kullanılan cihaza uygun şekilde ayarlanabilir ve işlemler paralel olarak yapılabilir.

2.5 YND Çalışma ve Analizlerindeki Zorluklar

YND yaklaşımları, çok sayıda genin aynı anda uygun maliyetli bir şekilde test edilmesini sağlar. Özellikle klinik kullanımlarda hedef bölge dizileme ve ekzom sekanslama ön plana çıkmaktadır. Hedef bölge dizileme genellikle belirli bir fenotiple ilişkili gen setlerinin panel analizlerine dayanır. Temel kısıtlamalardan birisi belirlenen genler dışındaki bölgelerin analiz edilemeyeşidir. Diğer taraftan tüm ekzonların analizini mümkün kılan tüm ekzom dizileme çalışmaları hedef bölge dizilemesi veya diğer testler ile tanısı konulamamış hastalar için oldukça önemlidir. Ancak unutulmamalıdır ki tüm ekzom sekanslama yöntemi, yöntem kısıtlamalarına bağlı olarak klinik öneme sahip genlerin ancak yaklaşık yüzde 85'ini efektif olarak analiz edebilmektedir. Ayrıca yapılan YND çalışmalarında tespit edilen genetik değişimlerin değerlendirmesi, özellikle hakkında fonksiyonel çalışma bulunmayan nadir mutasyonlar, oldukça zahmetlidir ve segregasyon çalışmaları gerektirmektedir. Belirtilmesi gereken diğer noktalardan birisi de psödogenlerin analizi sırasında yaşanan zorluklardır. Yüksek dizi homolojisi gösteren genlerden her ne kadar hepsi fonksiyonel olmasada PCR sırasında dizi benzerliğinden ötürü psödogenlerde çoğaltılmakta ve dizilenmektedir. Elde edilen verilerde saptanan genetik değişimlerin fonksiyonel bölgede mi fonksiyonel olmayan bölgede mi olduğu belirlenememekte ileri moleküler yöntemler ile çalışma gerektirmektedir. Bir diğer kısıtlama genomdaki tekrar bölgeleri ve homopolimer bölgelerdir. Genoma eşlenen okuma verilerinde özellikle tekrar bölgelerinde yanlış pozitif insersiyon veya delesyonlar görülebilir, Sanger dizileme yöntemi ile konfirmasyonunun yapılması gereklidir. Dizileme maliyetlerinin düşmesi ile giderek büyüyen genomik verinin analizi için kullanılan algoritmalarında sayısı artmaktadır. Algoritmalar ile ilgili olarak her birinin dosya formatları farklılık göstermekte hatta algoritmaların yayınlanan sürümleri arasında dahi farklılıklar olmaktadır. Birçok algoritma grafik arayüz ile kullanılmamakta Linux iMAC gibi sistemlerde terminal de komut yazarak işlem gerektirmektedir. Gerek kurulum gerekse kullanımdaki bu zorluklar sürümler arasındaki farklar ve algoritmaların kısıtlamaları ile birleşince analizler oldukça güç hale gelmektedir. Ayrıca büyük verilerin analizi yüksek kapasitede bellek ve fazla sayıda işlemci gerektirmektedir. İş akışındaki bazı basamaklardaki işlemlerin paralel olarak gerçekleştirilmesi gerekmektedir. Dolayısıyla hazırlanan iş akışının kullanıldığı bilgisayarın özelliklerine bağlı olarak yönetilmesi gerekir. Snakemake sistemi kullanılan algoritmaların kurulumu ve yönetimi konusunda oldukça kolaylık sağlamaktadır.

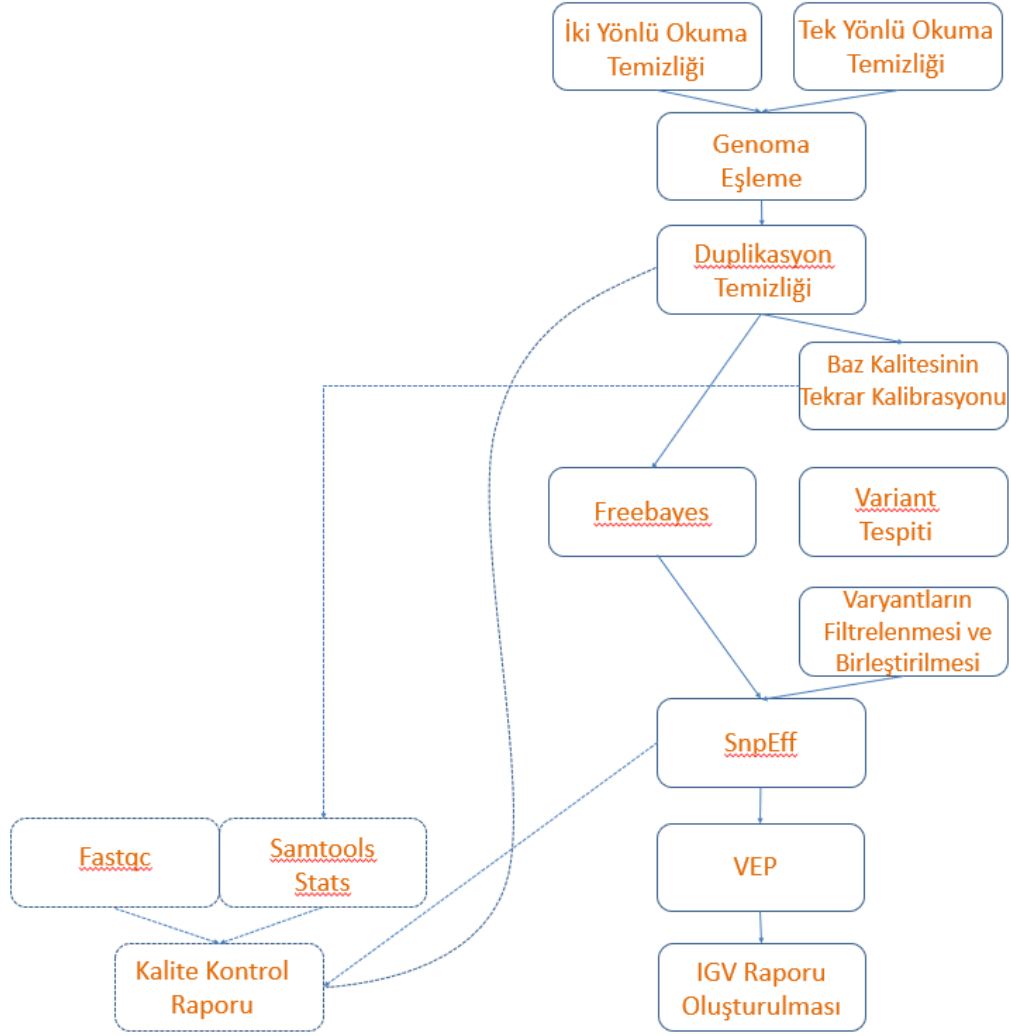
3.1 Kullanılan Sisteme Genel Bakış

Büyük ölçekli biyolojik veri analizleri birçok adım içermektedir ve sıralı bir şekilde yürütülmeleri gerekmektedir. Bu sıralı işlemlerin manuel olarak gerçekleştirilmesi oldukça zor ve hataya açık olup tekrarlanabilirliğini sağlamak mümkün olmamaktadır. İş akışı yönetim sistemleri iş akışındaki dosyaların, işlemlerin, girdi ve çıktıların tanımlanıp kullanılacak algoritmalara da ortam sağlayarak bu işlemlerin otomatikleşmesine ve tekrarlanabilir olmasına katkı sağlamaktadır. Tez çalışmamızda kullandığımız Snakemake herhangi kurulu bir paket veya girdi ve çıktı dosya türü iyi tanımlanmış internet tabanlı uygulamalar ile çalışabilmektedir. Snakemake tamamen taşınabilirdir çünkü 'Snakefile' adlı konfigürasyon dosyasında iş akışının çalışması için gerekli bütün bilgiler mevcuttur ve Python kurulu herhangi bir ortamda çalıştırılabilir. Bir iş akışı 'Snakefile' dosyasında Python diline benzer bir dil ile tanımlanır. Bu dosya girdi dosyalarından çıktı dosyalarının nasıl oluşturulduğunu gösteren kurallardan oluşur. Birden çok dosya işlenmesi durumunda girdi dosyalarının adlarını teker teker girilmesine gerek kalmadan, bir klasörün içindeki bütün dosyaların girdi olarak işleneceği kural olarak verilebilir[50]. Yaptığımız çalışmada YND analizi için geliştirilmiş Snakemake ile oluşturulmuş bir sistem taslak olarak kullanıldı [51] ve modifiye edilerek birçok özelliği değiştirildiği gibi analizler için kritik olarak değerlendirilen basamaklar eklendi. Ayrıca sistem oluşturulurken analiz sırasında kullanıcının isteğine göre analizlerin rahatça gerçekleştirilebilmesi için ilgili parametreler yapılandırma dosyasına taşındı. Sistemin genel görünümü ve iş akışı Şekil 3.1 de gösterilmiştir.

3.2 İş Akışı ve Sistem

3.2.1 Yapılandırma Dosyası

Yapılandırma dosyası iş akışındaki kullanıcı tarafından karar verilmesi gereken parametrelerin tutulduğu dosyadır. YAML veya JSON formatındaki yapılandırma



Şekil 3.1 Sistemin Genel Şeması

dosyası yapılan analizin basamaklarının ve ilgili parametrelerinin kolayca değiştirilebilmesini sağlar. Kullanılan referans genom, bilinen varyantların olduğu veri tabanı, veri temizliği parametreleri, varyant tespiti parametreleri bu dosyadan kolayca değiştirilebilir. Ayrıca örnek isimleri 'sample.tsv' dosyasında belirtilerek aynı anda birden fazla örnek analiz edilebilir. Tablo 3.1'deki örnek yapılandırma dosyasındaki parametreler, kuralların açıklandığı bölümde tarif edilecektir.

Tablo 3.1 Yapılandırma Dosyası

```
1 samples: samples.tsv
2 ref:
3   name: GRCh37.75
4   genome: data/ref/index_hg19/hg19.fa
5   known-variants: data/ref/common_all_20180423.vcf.gz
6   genome-vep: data/ref/GRCh37-vep-db/homo_sapiens/96_GRCh37/
   Homo_sapiens.GRCh37.75.dna.primary_assembly.fa
7   vepdir: data/ref/GRCh37-vep-db
8
9 filtering:
10  vqsr: false
11  hard:
12    snvs:
13      "QD < 2.0 || FS > 60.0 || MQ < 40.0 || MQRankSum <
-12.5 || ReadPosRankSum < -8.0"
14    indels:
15      "QD < 2.0 || FS > 200.0 || ReadPosRankSum < -20.0"
16
17 processing:
18  remove-duplicates: true
19  # restrict-regions: captured_regions.bed
20
21 params:
22  gatk:
23    HaplotypeCaller: ""
24    BaseRecalibrator: ""
25    GenotypeGVCFs: ""
26    VariantRecalibrator: ""
27  picard:
28    MarkDuplicates: "REMOVE_DUPLICATES=true"
29  trimmomatic:
30    pe:
31      trimmer:
32        - "LEADING:3"
33        - "TRAILING:3"
34        - "SLIDINGWINDOW:4:15"
35        - "MINLEN:36"
36    se:
37      trimmer:
38        - "LEADING:3"
39        - "TRAILING:3"
40        - "SLIDINGWINDOW:4:15"
41        - "MINLEN:36"
42 vcf-columns:
43   ann: [""]
44   csq: [""]
```

3.2.2 Snakefile Dosyası ve Kurallar

3.2.2.1 Snakefile Dosyası

Snakefile dosyası iş akışının basamaklarının belirlendiği sistemdeki kuralların ve çıktı dosyalarının tanımlandığı dosyadır. Ana kural dosyasının içeriği Tablo 3.2’de verilmiştir. Kural dosyaları modüler şekilde kullanılabilir, ana kural dosyası başka kural dosyalarını çağırıp kullanabilir (Tablo 3.2).

Tablo 3.2 Snakefile Dosyası

```
1
2 include: "rules/common.smk"
3
4 ##### Target rules #####
5
6 rule all:
7     input:
8         expand("parse-vcf/{sample}.ann.csq.tsv", sample=
9             samples.index)
10
11 ##### Modules #####
12
13 include: "rules/mapping.smk"
14 include: "rules/calling.smk"
15 include: "rules/filtering.smk"
16 include: "rules/stats.smk"
17 include: "rules/qc.smk"
18 include: "rules/annotation.smk"
19 include: "rules/vep.smk"
20 include: "rules/igv.smk"
21 include: "rules/vepparse.smk"
```

3.2.2.2 Kurallar

- Başlangıç Kuralı: Sistemin çalışması için gerekli girdi dosyaları ve girdilerin içeriklerinin düzenlendiği kurallardır. İş akışının başlaması için gerekli konfigürasyon dosyası ve 'samples.tsv' dosyası gibi dosyalar konfigürasyon dosyasındaki kullanıcının seçimleri doğrultusunda tanımlanır ve iş akışı başlatılır. Tez için geliştirilen iş akışı yönetim sisteminin başlangıç kuralı Tablo3.3’de gösterilmiştir.

Tablo 3.3 Başlangıç Kuralı

```
1 import pandas as pd
2 from snakemake.utils import validate
3
4 report: "../report/workflow.rst"
5 configfile: "config.yaml"
6
7 samples = pd.read_table(config["samples"]).set_index("
    sample", drop=False)
8
9 units = pd.read_table(config["units"], dtype=str).
    set_index(["sample", "unit"], drop=False)
10 units.index = units.index.set_levels([i.astype(str) for i
    in units.index.levels]) # enforce str in index
11
12 contigs = pd.read_table(config["ref"]["genome"] + ".fai",
13                          header=None, usecols=[0], squeeze
    =True, dtype=str)
14
15
16 def get_fastq(wildcards):
17     fastqs = units.loc[(wildcards.sample, wildcards.unit)
18     , ["fq1", "fq2"]].dropna()
19     if len(fastqs) == 2:
20         return {"r1": fastqs.fq1, "r2": fastqs.fq2}
21     return {"r1": fastqs.fq1}
22
23 def is_single_end(sample, unit):
24     return pd.isnull(units.loc[(sample, unit), "fq2"])
25
26
27 def get_read_group(wildcards):
28     return r"-R '@RG\tID:{sample}\tSM:{sample}\tPL:{
    platform}'".format(
29         sample=wildcards.sample,
30         platform=units.loc[(wildcards.sample, wildcards.
    unit), "platform"])
```

- Kalite Kontrol Kuralı: Kalite kontrol algoritmalarının çalışmasını girdi ve çıktı dosyalarının düzenlendiği kural dosyasıdır. Fastqc ile oluşturulan raporda dizi uzunluk dağılımı, dizi duplikasyon seviyeleri, dizi başına ortalama kalite dağılımı, ortalama GC içeriği, ortalama kalite skorları ve adaptör içeriği gibi kritik bilgiler raporlanır ve diğer uygulamalardan gelen bilgiler multiqc ile birleştirilerek tek bir rapor oluşturulur. Ayrıca programların kurulması Conda

yardımla proje klasörü içinde gerçekleşir ve programın nasıl çalıştırılacağına dair komut satırı da hazır olarak alınır. Kalite kontrol kuralı Tablo 3.4’de gösterilmiştir.

Tablo 3.4 Kalite Kontrol Kuralı

```
1 rule fastqc:
2     input:
3         unpack(get_fastq)
4     output:
5         html="qc/fastqc/{sample}-{unit}.html",
6         zip="qc/fastqc/{sample}-{unit}.zip"
7     wrapper:
8         "0.27.1/bio/fastqc"
9
10
11 rule samtools_stats:
12     input:
13         "recal/{sample}-{unit}.bam"
14     output:
15         "qc/samtools-stats/{sample}-{unit}.txt"
16     log:
17         "logs/samtools-stats/{sample}-{unit}.log"
18     wrapper:
19         "0.27.1/bio/samtools/stats"
20
21
22 rule multiqc:
23     input:
24         expand(["qc/samtools-stats/{u.sample}-{u.unit}.",
25             "txt",
26             "qc/fastqc/{u.sample}-{u.unit}.zip",
27             "qc/dedup/{u.sample}-{u.unit}.metrics.txt"
28         ],
29             u=units.itertuples()),
30         "snpeff/all.csv"
31     output:
32         report("qc/multiqc.html", caption="../report/
33 multiqc.rst", category="Quality control")
34     log:
35         "logs/multiqc.log"
36     wrapper:
37         "0.33.0/bio/multiqc"
```

- Genoma Eşleme Kuralı: Genoma eşleme için gerekli iş akışının düzenlendiği girdi ve çıktı dosyalarının düzenlendiği kuralları içerir. Kalite kontrol raporu oluşturulan ham veriler konfigürasyon dosyasında kullanıcı tarafından belirtilen

değerlere göre öncelikle trimmomatic tarafından temizlenir. Ardından BWA ile genomla eşleme gerçekleştirilir ve kullanıcı isteğine bağlı olarak duplike okumalar Picard ile uzaklaştırılır. Yine kullanıcının isteğine bağlı olarak baz kalitesine göre bir yeniden kalibrasyon yapılır ve samtools ile bam dosyalarının indeksleri oluşturulur. Genomla eşleme kuralı Tablo 3.5’de gösterilmiştir.

Tablo 3.5 Genomla Eşleme Kuralı

```

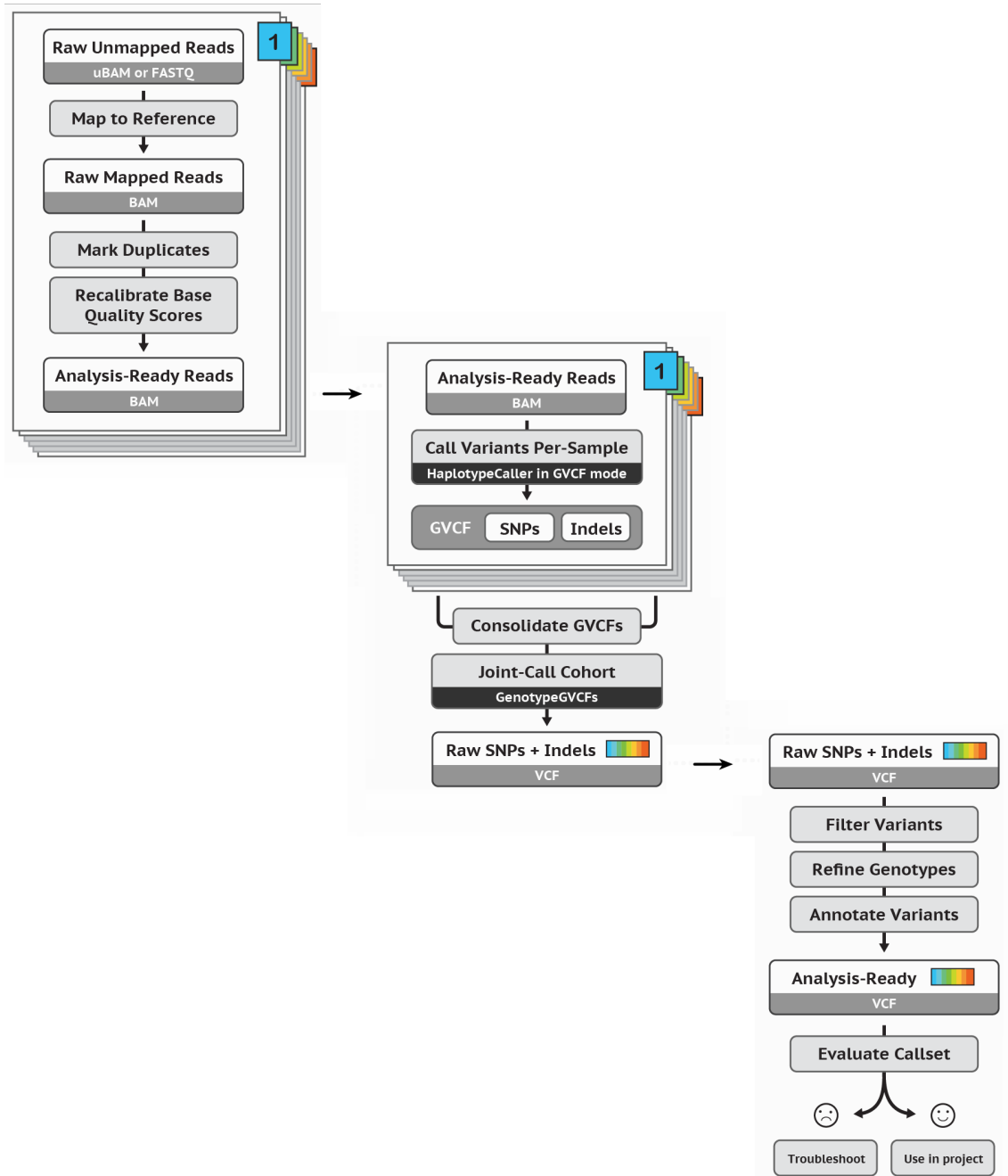
1 rule map_reads:
2     input:
3         reads=get_trimmed_reads
4     output:
5         temp("mapped/{sample}-{unit}.sorted.bam")
6     params:
7         index=config["ref"]["genome"],
8         extra=get_read_group,
9         sort="samtools",
10        sort_order="coordinate"
11    threads: 8
12    wrapper:
13        "0.27.1/bio/bwa/mem"
14 rule mark_duplicates:
15     input:
16         "mapped/{sample}-{unit}.sorted.bam"
17     output:
18         bam=temp("dedup/{sample}-{unit}.bam"),
19         metrics="qc/dedup/{sample}-{unit}.metrics.txt"
20     params:
21         config["params"]["picard"]["MarkDuplicates"]
22     wrapper:
23         "0.26.1/bio/picard/markduplicates"
24 rule recalibrate_base_qualities:
25     input:
26         bam=get_recal_input(),
27         bai=get_recal_input(bai=True),
28         ref=config["ref"]["genome"],
29         known=config["ref"]["known-variants"]
30     output:
31         bam=protected("recal/{sample}-{unit}.bam")
32     params:
33         extra=get_regions_param() + config["params"]["gatk"]
34         ["BaseRecalibrator"]
35     wrapper:
36         "0.27.1/bio/gatk/baserecalibrator"

```

- Varyant Tespiti Kuralları:

Varyant tespiti için gerekli dosyalar ve iş akışının düzenlendiği kural dosyalarıdır.

Sık kullanılan GATK algoritması veya freebayes algoritması konfigrasyon dosyasında kullanıcının hangisini kullanmak istediğine bağlı olarak çalışır. GATK ile varyant tespiti için GATK en iyi uygulaması olarak bildirilen iş akışı (Şekil3.2) uygulanmıştır (Tablo3.6). Opsiyonel olarak eklenen freebayes algoritması için ise varsayılan olarak en az 20 derinlik ve değişen baz sayısı 5 olarak belirlenmiştir. Kullanıcının isteğine göre parametreleri değiştirmesi mümkündür (Tablo3.7).



Şekil 3.2 GATK En İyi Uygulaması İş Akışı

Tablo 3.6 Varyant Tespiti Kuralı (GATK)

```
1 rule call_variants:
2     input:
3         bam=get_sample_bams ,
4         ref=config["ref"]["genome"],
5         known=config["ref"]["known-variants"],
6         regions="called/{contig}.regions.bed" if config["
processing"].get("restrict-regions") else []
7     output:
8         gvcf=protected("called/{sample}.{contig}.g.vcf.gz
")
9     log:
10        "logs/gatk/haplotypcaller/{sample}.{contig}.log"
11    params:
12        extra=get_call_variants_params
13    wrapper:
14        "0.27.1/bio/gatk/haplotypcaller"
15 rule genotype_variants:
16     input:
17         ref=config["ref"]["genome"],
18         gvcf="called/{sample}.{contig}.g.vcf.gz"
19     output:
20         vcf=temp("genotyped/{sample}.{contig}.vcf.gz")
21     params:
22         extra=config["params"]["gatk"]["GenotypeGVCFs"]
23     log:
24         "logs/gatk/genotypegvcfs{sample}.{contig}.log"
25     wrapper:
26         "0.27.1/bio/gatk/genotypegvcfs"
27 rule merge_variants:
28     input:
29         vcf=expand("genotyped/{sample}.{contig}.vcf.gz"
, contig=contigs)
30     output:
31         vcf="genotyped/{sample}.vcf.gz"
32     log:
33         "logs/picard/merge-genotyped{sample}.log"
34     wrapper:
35         "0.27.1/bio/picard/mergevcfs"
```

Tablo 3.7 Varyant Tespiti Kuralı (freebayes)

```
1 rule samtools_index:
2     input:
3         "dedup/{sample}-1.bam"
4     output:
5         "dedup/{sample}-1.bam.bai"
6     wrapper:
7         "0.37.1/bio/samtools/index"
8
9 rule freebayes:
10    input:
11        ref=config["ref"]["genome"],
12        samples="dedup/{sample}-1.bam",
13        index= "dedup/{sample}-1.bam.bai"
14    output:
15        "filtered/{sample}.vcf"
16    log:
17        "logs/freebayes/{sample}.log"
18    params:
19        extra="-C 5 --min-coverage 20 ",
20        chunksize=10000000
21    threads: 4
22    wrapper:
23        "0.36.0/bio/freebayes"
```

- Varyant Filtreleme Kuralı: Tespit edilen varyantların filtreleme işlemlerinin tanımlandığı kural dosyasıdır (Tablo3.8).
- Annotasyon Kuralları: Tespit edilen varyantların değerlendirmesine ilişkin veri tabanlarındaki bilgilerin toplanması ile ilgili iş akışını düzenleyen kural dosyalarıdır. Literatürde sıkça adı geçen SnpEff ve VEP algoritmalarının her ikisinde iş akışı yönetim sistemine eklenmiş çalışmaları için gerekli komut ve girdi çıktı dosyaları Tablo3.9 (SnpEff) ve Tablo3.10 (VEP) gösterildiği üzere oluşturulmuştur.

Tablo 3.8 Varyant Filtreleme Kuralı

```
1 rule select_calls:
2     input:
3         ref=config["ref"]["genome"],
4         vcf="genotyped/{sample}.vcf.gz"
5     output:
6         vcf=temp("filtered/{sample}.{vartype}.vcf.gz")
7     params:
8         extra=get_vartype_arg
9     log:
10        "logs/gatk/selectvariants/{sample}.{vartype}.log"
11    wrapper:
12        "0.27.1/bio/gatk/selectvariants"
13 rule hard_filter_calls:
14     input:
15         ref=config["ref"]["genome"],
16         vcf="filtered/{sample}.{vartype}.vcf.gz"
17     output:
18         vcf=temp("filtered/{sample}.{vartype}.
19         hardfiltered.vcf.gz")
20     params:
21         filters=get_filter
22     log:
23        "logs/gatk/variantfiltration/{sample}.{vartype}.
24        log"
25    wrapper:
26        "0.27.1/bio/gatk/variantfiltration"
```

Tablo 3.9 Anotasyon Kuralı (Snpeff)

```
1 rule snpeff:
2     input:
3         "filtered/{sample}.vcf.gz",
4     output:
5         vcf=report("annotated/{sample}.vcf.gz", caption="
6         ../report/vcf.rst", category="Calls"),
7         csvstats="snpeff/{sample}.csv"
8     log:
9        "logs/snpeff{sample}.log"
10    params:
11        reference=config["ref"]["name"],
12        extra="-Xmx6g"
13    threads: 4
14    wrapper:
15        "0.27.1/bio/snpeff"
```

Tablo 3.10 Anotasyon Kuralı (VEP)

```
1 rule veptest:
2     input:
3         vcf="annotated/{sample}.vcf.gz",
4         genome=config["ref"]["genome-vep"],
5         vepdir=config["ref"]["vepdir"]
6     output:
7         "vep/{sample}.vep.vcf"
8     conda:
9         "../envs/vep.yaml"
10    threads: 4
11
12    shell:
13        "vep --verbose --offline --everything --fa {input
.genome} --gencode_basic --cache --dir {input.vepdir}
--fork 4 -i {input.vcf} --vcf -o {output}"
```

- Annotasyon Çıktı Dosyasının Düzenlenme Kuralı: Her bir varyant için elde edilen ve çıktı vcf dosyasının bir kolonunda tutulan bilgilerin düzenlenmesini içeren kural dosyasıdır. Tablo3.11’de belirtildiği gibi R dilinde yazılmış basit komutlar yardımıyla VCF formatındaki dosya işlenmiştir.

Tablo 3.11 Annotasyon Çıktı Dosyasının Düzenlenme Kuralı

```
1 rule strip_vcf_comments:
2     input: "vep/{sample}.vep.vcf"
3     output: temp("parse-vcf/{sample}.anno.vcf.nocomment")
4     shell:
5         "sed -e 's/^#//' {input} > {output}"
6 rule extract_ann_csq:
7     input: "parse-vcf/{sample}.anno.vcf.nocomment"
8     output: "parse-vcf/{sample}.ann.csq.tsv"
9     params:
10        ann= config["vcf-columns"]["ann"],
11        csq= config["vcf-columns"]["csq"],
12        sample_name = lambda w: w.sample
13        #sample={{sample}}
14    conda: "../envs/tidyverse.yml"
15    script:
16        "../extract-ann-csq-vcf.R"
```

Kural dosyasının çalışması için gerekli komut dosyası ise Tablo 3.12’de gösterilmiştir.

Tablo 3.12 Kural Dosyasının Çalışması İçin Gerekli Komut Dosyası

```
1 filename <- snakemake@input[[1]]
2 contents <- read_tsv(filename, comment="#")
3 sample_name <- snakemake@params$sample_name
4 main_cols <- c("CHROM", "POS", "ID", "REF", "ALT", "QUAL", "
  FILTER")
5 ann_cols <- snakemake@params$ann
6 csq_cols <- snakemake@params$csq
7 cols_to_pick <- c(main_cols, ann_cols, csq_cols)
8 ann_col_headers <- c("")
9 csq_col_headers <- c("")
10 zip_last_columns <- . %>%
11 mutate(temp=map2(strsplit(FORMAT, ":"), strsplit((!!sym(
  sample_name)), ":"), ~paste0(.x, ":", .y))) %>%
12 unnest(temp) %>%
13 separate(temp, into=c("key", "value"), sep=":") %>%
14 spread(key, value) %>%
15 select(-FORMAT, -(!!sym(sample_name))) %>%
16 mutate(GT_text=case_when(GT == "0/1" ~ "Het", GT == "1/1
  " ~ "Homo", TRUE ~ "Other"))
17 pick_expand_ann_csq <- . %>%
18 separate_rows(INFO, sep=";") %>%
19 separate(INFO, c("key", "value"), sep="=") %>%
20 filter(key=="ANN" | key=="CSQ") %>%
21 spread(key, value) %>%
22 { select(., id_snp, CSQ) %>%
23   separate_rows(CSQ, sep=",") %>%
24   separate(CSQ, sep="\\|", into=csq_col_headers) ->>
  csq_raw } %>%
25 select(-CSQ) %>%
26 separate_rows(ANN, sep=",") %>%
27 separate(ANN, sep="\\|", into=ann_col_headers) %>%
28 full_join(csq_raw, by=c("id_snp", "allele", "feature_id
  " = "feature"))
29 parsed_pool <- contents %>%
30 mutate(id_snp=row_number()) %>%
31 zip_last_columns() %>%
32 pick_expand_ann_csq() %>%
33 group_by(id_snp) %>%
34 mutate(id_eff=row_number()) %>%
35 ungroup() %>%
36 mutate(ann_id=paste("ANN", id_snp, id_eff, sep=".")) %>%
37 select(one_of(cols_to_pick))
38 write_tsv(parsed_pool, snakemake@output[[1]])
```

- Tespit Edilen Varyantların Görüntü Raporu: Tespit edilen varyantların interaktif olarak İGV ile görüntülenmesi için gerekli raporu oluşturan işlemleri düzenleyen kural dosyasıdır (Tablo3.13).

Tablo 3.13 Tespit Edilen Varyantların Görüntü Raporu Kuralı

```
1 rule igv_report:
2     input:
3         fasta=config["ref"]["genome"],
4         vcf="vep/{sample}.vep.vcf",
5         tracks=["recal/{sample}-1.bam"]
6     output:
7         "igv/{sample}.igv-report.html"
8     params:
9         extra=""
10    log:
11        "logs/{sample}.igv-report.log"
12    wrapper:
13        "0.35.1/bio/igv-reports"
```

4.1 Bulgular

Materyal metod bölümünde behsedilen iş akışı ile üç farklı çıktı dosyası üretilir. Birincisi her bir varyantın anote edildiği dosya, diğeri kalite kontrol raporu diye belirtilen ham verinin kalitesi ile ilgili bilgilerin bulunduğu dosya sonuncu olarak ise IGV raporu dosyasıdır. Özellikle varyant analizlerinde ham verilerin değerlendirilebilmesi için kritik olan bu dosyaların otomatik olarak oluşturulması hata oranını düşürmektedir. Geliştirdiğimiz iş akışının pratikliğini ve süresini test etmek maksadıyla örnek olarak 12 ham veri dosyası (toplam 150 milyon okuma) kullanılmış ve işlenmiştir. Ham veri dosyaları bir klasöre taşındıktan sonra Snakemake ayar dosyasında ayarlar girilmiş ve iş akışı başlatılmıştır. Snakemake, her bir dosya için 9 kural/basamağın çalıştırılacağını ve toplamda 109 iş/basamak çalıştıracağını belirtmiştir. Dosyaların analizi sırasında herhangi bir manuel müdahale yapılmamış ve yaklaşık 60 dakika sonra analiz tamamlanmıştır. Bu örnek, 10 CPU ve 32 GB hafızası olan bir bilgisayarda gerçekleştirilmiştir. İşlem sayısı ve süresi Şekil 4.1’de özetlenmiştir. Ayrıca iş akışı sırasında herhangi bir kesinti oluştuğunda ve sonrasında tekrar çalıştırıldığında iş akışı kaldığı yerden devam edebilmektedir, işlemler baştan başlatılmamaktadır.

4.1.1 Anotasyon Raporu

Her bir genetik değişim için bir çok veri tabanından değişimin değerlendirilebilmesi için bilgi toplanmakta ve tablo halinde kullanıcıya sunulmaktadır. Toplamda her bir varyant için yaklaşık 80 farklı bilgi toplanmakta ve değişimin değerlendirilebilmesi için kullanıcıya raporlanmaktadır. Bunlardan bazı önemli olanlarını açıklayacak olursak;

- Genetik Değişimin Koordinatı ve Değişim: Rapor dosyasında ilk sırada bu bilgiler gelir Her bir değişim için referans genoma göre kromozom ve koordinat bilgisi ayrıca referanstaki baz ve hangi baza dönüşüm olduğu bilgisi tutulur. Bu her bir değişim için sabittir ve başka nerede sorgu yapılacak olunursa olunsun bu

```
Provided cores: 10
Rules claiming more threads will be scaled down.
Job counts:
```

count	jobs
1	all
12	extract_ann_csq
12	freebayes
12	map_reads
12	mark_duplicates
12	samtools_index
12	snpeff
12	strip_vcf_comments
12	trim_reads_pe
12	veptest
109	

```
[Tue Dec 17 16:00:16 2019]
```

```
[Tue Dec 17 17:00:37 2019]
```

```
Finished job 0.
```

```
109 of 109 steps (100%) done
```

```
Complete log: /home/test/Desktop/snakemake/.snakemake/
```

Şekil 4.1 Snakemake ile Gerçekleştirilen Örnek Bir Çalışmanın Süresi ve İşlem Sayısı
12 Dosya için çalıştırılan iş akışı her bir kuralı her bir dosya için çalıştırır. 10 işlemci kullanıldığında tüm iş akışı yaklaşık 1 saatte tamamlanmaktadır.

bilgiler kullanılacaktır.

- **Genotip:** Her bir değişim için genotip bilgisi rapor dosyasında kullanıcıya sunulur. Bu bilgi genetik değişimlerin değerlendirilmesi için oldukça önemlidir. Çünkü genlerin kalıtım modeline göre değerlendirme yapılması gerekmektedir. Örneğin, resesif kalıtılan bir gendeki heterozigot değişim fenotip oluşturmazken dominant bir gende meydana gelen heterozigot bir değişim fenotip oluşturabilir.
- **Okuma Derinliği ve Değişim Oranı:** Her bir değişim başına kaç okuma düştüğü ve referans ile değişim oranı raporda verilmektedir. Burada iki şey önemlidir. Birincisi okuma derinliği arttıkça saptamanın doğruluğu artmaktadır. Dolayısıyla değişim değerlendirilirken bu bilgi oldukça önemlidir. İkincisi ise somatik mutasyonlarda ve psödogene sahip genlerdeki değişimlerde değişim oranları varyantların değerlendirilmesi için önem arz etmektedir.
- **Gen Adı ve Transkript:** Her bir değişimin meydana geldiği gen ismi son raporda kullanıcıya sunulmaktadır. Bu bilgi birçok açıdan önemli olsa da en kritik nokta genotip-fenotip ilişkisinin kurulmasıdır. Değişimin hangi gende meydana

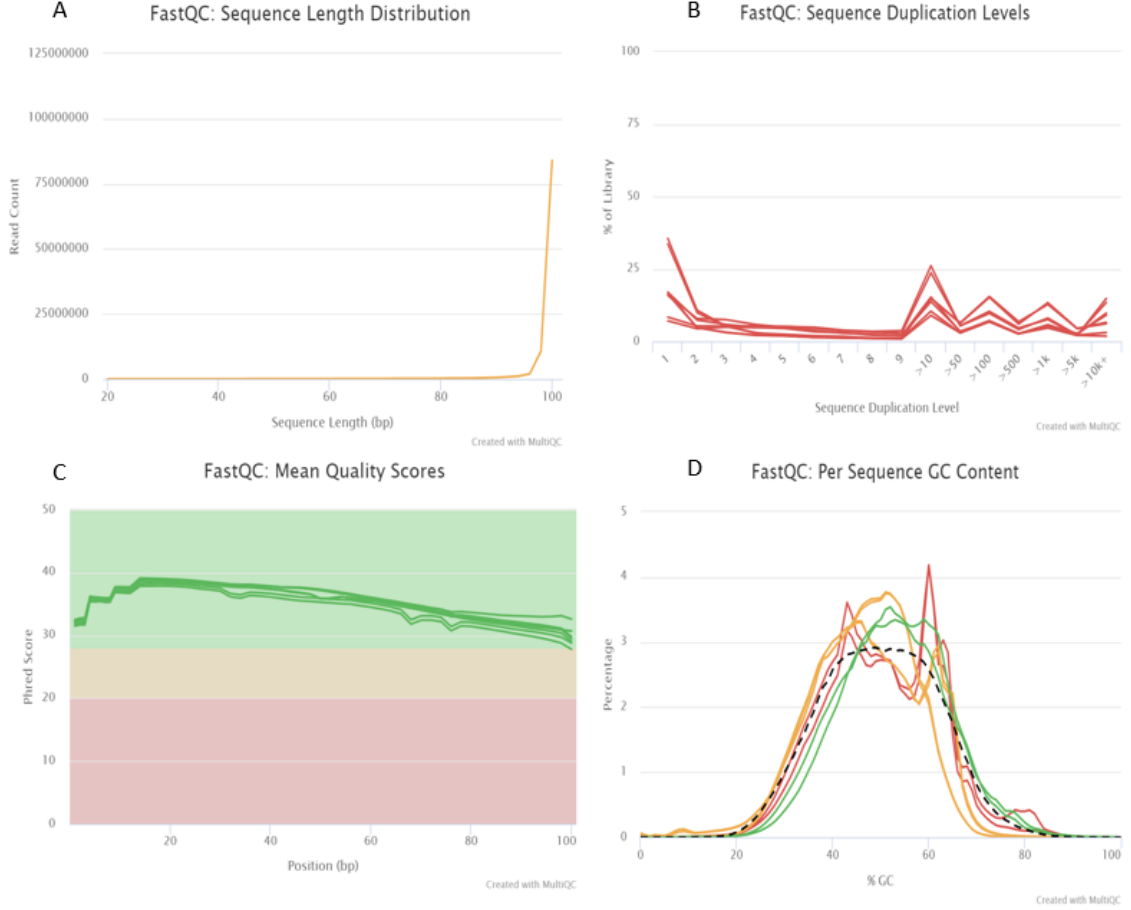
geldiğinin bilinmesi analizin en kritik adımlarındandır. Transkript bilgisi de en az gen ismi kadar önemlidir. Bu yüzden, geliştirdiğimiz iş akışında, her bir değişimin meydana geldiği genin her bir transkripti için verilen bilgiler ayrı satıra basılarak raporlanmaktadır. Böylece kullanıcıya kendi çalıştığı transkripte göre analiz imkanı sunulmaktadır.

- Değişimin Meydana Geldiği Ekzon: Saptanan genetik değişimin her bir transkript için hangi ekzonda veya intronda meydana geldiği bilgisi de raporda belirtilir. Bu bilgi, özellikle manuel olarak bam dosyasından yapılacak manuel kontrolleri kolaylaştırmaktadır.
- HGVS Yazımı: Uluslararası olarak kabul görmüş, genetik değişimlerin yazılımı için kurulan konsensusun belirlediği varyant yazım kurallarına göre saptanan her bir genetik değişim her bir transkripte göre ayrı ayrı rapora eklenir.
- Clinvar Veri Tabanı ve Literatür Bilgisi: Yaygın olarak kullanılan genetik değişimlerin klinik etkilerinin literatür bilgisine dayanılarak sınıflandırıldığı veri tabanı olan clinvar veri tabanından eğer saptanan genetik değişimle ilgili girdi mevcutsa rapora eklenmektedir. Ayrıca, değişimlerle ilgili yapılan bilimsel çalışmaların PubMed numaraları da kullanıcılara değişimleri efektif olarak değerlendirilebilmeleri için sunulmaktadır.
- Proteindeki Domain Bilgisi: Genetik değişimin meydana geldiği koordinatın kodlanan proteinde hangi domaine denk geldiğinin bilgisi de raporda sunulan bir bilgidir. Bilgi, değişimin protein üzerine fonksiyonel etkilerinin değerlendirilmesi için önemlidir.
- Alel Frekansı: Saptanan değişimlerin yapılan toplum sıklığı çalışmalarında sağlıklı insanlarda ne kadar görüldüğü bilgisi ilgili veritabanlarından alınarak kullanıcılara sunulmaktadır. Bu bilgi bir değişimin değerlendirmesi açısından önemlidir çünkü toplumda sık görülen değişimlerin fenotipik etkilerinin ya da hastalık sebebi olmasının ihtimali azdır.

4.1.2 Kalite Kontrol Raporu

Analiz edilcek YND verilerinin deneysel süreçleri ve elde edilmek istenen çıktılar dikkate alınarak kalite kontrolü yapılması ve uygun bir algoritma ile gerekli temizliklerin yapılması verilerden üretilcek bilgilerin doğruluğu ve hassaslığı açısından önemlidir. Oluşturulan iş akışı, kullanıcı verisinin analizine başlamadan önce kalite kontrolü yapabilmesini sağlayan kalite kontrol dosyasını otomatik olarak oluşturmaktadır. Ayrıca birden fazla dosyanın analizinde de tek bir kalite kontrol dosyası oluşturulmakta ve her bir örneğin bilgileri tek raporda toplanmaktadır.

Oluşturulan raporda dizi uzunluk dağılımı, dizi duplikasyon seviyeleri, dizi başına ortalama kalite dağılımı, ortalama GC içeriği, ortalama kalite skorları ve adaptör içeriği gibi önemli bilgiler kullanıcıya sunulmaktadır. Şekil 4.2’de bazı örnek raporlar gösterilmiştir.

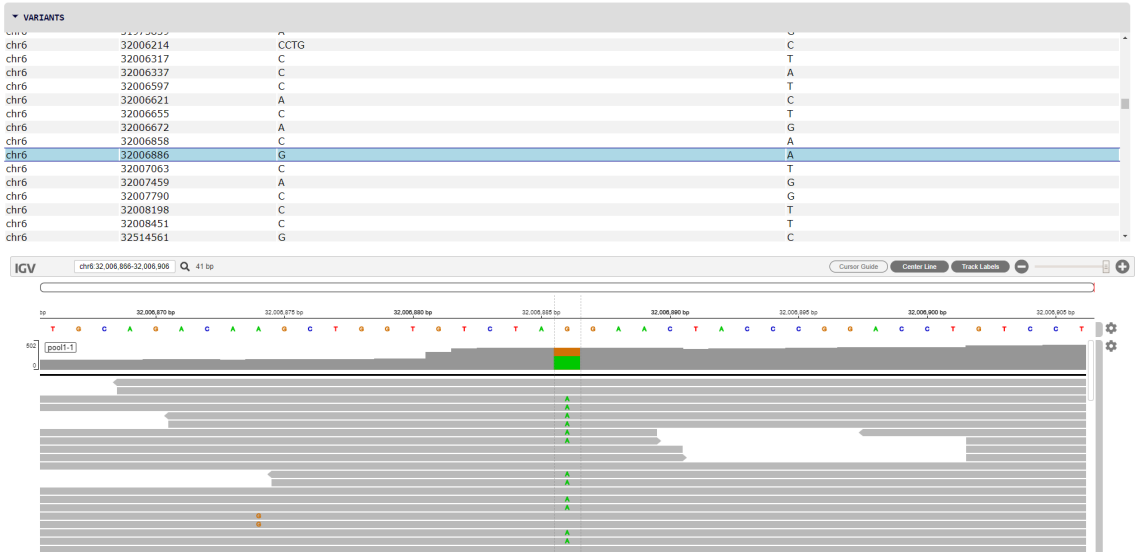


Şekil 4.2 Örnek Kalite Kontrol Rapor İçeriği

A) Dizi Uzunluk Dağılımı B) Dizi Duplikasyon Seviyeleri C) Ortalama Kalite Skorları
D) Ortalama GC İçeriği

4.1.3 IGV Raporu

İş akışına dahil edilen ve otomatik olarak oluşturulan bu dosya kullanıcının analiz sırasında işini oldukça kolaylaştırmaktadır. Özellikle varyant analizlerinde her ne kadar oluşturulan anotasyon dosyasından değişimler değerlendirilebilse de IGV görüntüsü ile interaktif olarak yapılan analizler yöntem kısıtlamaları nedeniyle oluşacak hataları en aza indirmektedir. Saptanan her bir değişimin interaktif olarak IGV görüntüsünün sunulduğu rapor dosyasının bir örneği şekil 4.3’te gösterilmiştir.



Şekil 4.3 Örnek IGV Raporu

4.2 Tartışma

Yeni Nesil Dizileme verilerinin analizi için açık kaynak kodlu bir çok algoritma geliştirilmiştir. Bu algoritmalar kullanılarak gerçekleştirilen analizler vakit almakta ve kodlama bilgisi gerektirmektedir. Her bir algoritmanın girdi ve çıktı dosyaları farklılık göstermekte ve hatta algoritmaların sürümleri arasında bile önemli farklar olmaktadır dolayısıyla otomasyonun sağlanması gerekmektedir. Kısıtlı sayıda olsa da literatürde açık kaynak kodlu iş akışları yayınlanmıştır [52]. Ayrıca ticari olarak da özellikle varyant analizlerini gerçekleştirmek üzere iş akışı sistemleri geliştirilmiştir. Ancak bunlar geliştiricilerin belirlediği algoritmalar ve parametreler doğrultusunda çalışmaktadır. Biz çalışmamızda literatürde varyant analizlerinde efektif olduğu ortaya konulan algoritmaları opsiyonel olarak kullanıcılara sunmakla birlikte analiz parametrelerini değiştirebilme imkanı sunduk. Literatürde yapılan çalışmalar ile iş akışı sistemleri birbirleri ile karşılaştırılmış farkları ortaya konulmuştur [53]. İş akışı oluştururken kullanılan Snakemake iş akışı oluşturmak için geliştirilen algoritmalar arasında manipülasyonu kolay olan ve girdi çıktı dosyalarının kolayca tanımlanabildiği bir algoritmadır, Nextflow [49] gibi algoritmalar ile karşılaştırıldığında kullanımı oldukça kolaydır. Geliştirilen iş akışına ileri çalışmalar ile ACMG kılavuzuna göre her bir varyantın anote edilmesi varyant değerlendirmesi için anlamlı olacaktır. Literatürde yapılan çalışmalarda bu kılavuza göre değerlendirme için algoritmalar geliştirilmiştir [54] ve iş akışına eklenmesi planlanmaktadır. Bunun yanı sıra ham veriden elde edilen varyantların değerlendirilmesi sırasında gen fenotip ilişkisinin bilinmesi varyantların değerlendirilmesi esnasında kritik öneme sahiptir. OMIM ve HPO (Human Phenotype Ontology) [55] gibi veri tabanlarından gen fenotip ilişkilerinin indirilip her bir gen için en son oluşturulan anotasyon dosyasında

eklenmesi ilerleyen çalışmalar için planlanmaktadır. İlerleyen çalışmalarda eklenmesi düşünülen bir diğer özellik ise ham verilerden genomdaki kopya sayısı değişimlerinin analizlerine imkan sağlayabilmektir. Bunun için farklı teknikler ile geliştirilmiş birçok algoritma mevcuttur [56]. En kullanılabilir olanın seçilip iş akışına eklenmesi özellikle kopya sayısına bağlı fenotiplerin ortaya konulması açısından önem taşımaktadır. Ayrıca geliştirilen sistemin Windows işletim sisteminde denemelerinin yapılması ve gerekli optimizasyonların sağlanması kullanılabilirliğini arttıracaktır.

4.3 Sonuç

Yapılan çalışmada Illumina YND platformundan elde edilen ham verilerin özellikle mutasyon tespiti çalışmalarında analizinin otomasyonu iyileştirilmiştir. Ayrıca geliştirilen sistem birden çok aşama gerektiren ve örnek sayısı arttıkça karmaşılaşan işlemleri otomatik hale getirmiştir. Bu tez çalışmasında YND analizleri için kullanılan algoritmaların belirli sürümleri otomatik olarak kurulmuş böylelikle sürüm farklarından kaynaklanan uyumsuzluklar en aza indirilmiştir. Ayrıca, girdi/çıkı dosyaları ve yapılacak işlemler tek bir dosyada belirtilmiş ve böylelikle tek bir kod ile analiz yapılabilir hale gelmiştir. Sonuç olarak özellikle yazılım bilgisi gerektiren analizler moleküler biyologlar için kolay hale gelmiştir. Kullanıcı tarafından değiştirilmesi muhtemel parametreler bir ayar veya yapılandırma dosyasında toplanmış böylelikle kullanıcının algoritma çalıştırmak için kurallar dosyasında program bilgisi gerektiren değişiklik yapmasına gerek kalmamıştır. Bu parametreler şu şekilde sıralanabilir; veri temizliği parametreleri, genoma eşleme parametreleri ve ilgilenilen anotasyon kaynakları. Özellikle belirtilmesi gereken bir diğer nokta ise varyant tespiti için geliştirilen ve altın standart olarak gösterilen GATK ve Freebayes algoritmalarının her ikisi de sisteme eklenmiş, kullanıcı istediği algoritmayı ayar dosyasında belirterek kullanabilmektedir. Ayrıca tespit edilen değişimlerin anotasyonu için tek bir algoritma kullanılmamış sık kullanılan SnpEff ve VEP algoritmalarının her ikisi de varyant anotasyonu için iş akışına eklenmiştir.

Özetle, analiz edilcek ham veriler örnek dosyasında tanımlanıp işlemler başlatıldıktan sonra verilerin kalite kontrol raporu, anote edilmiş varyant listesi ve interaktif analizi sağlayan IGV raporu sistemden alınmaktadır. Ayrıca manuel analiz yapmak isteyen kullanıcılar için bam dosyaları tutulmakta gerekli klasörlerden alınıp kullanılabilir hale getirilmektedir. Yapılacak ileri çalışmalar ile de oluşturulan algoritma daha kullanılabilir hale getirilmelidir.

- [1] Z. S. Lodish H Berk A, *Mutations: Types and Causes*. New York: W. H. Freeman, 2000.
- [2] NIH. (). What kinds of gene mutations are possible? [Online]. Available: <https://ghr.nlm.nih.gov/primer/mutationsanddisorders/possiblemutations> (visited on 10/10/2019).
- [3] J. Sequeiros, M. Paneque, B. Guimarães, E. Rantanen, P. Javaher, I. Nippert, J. Schmidtke, H. Kääriäinen, U. Kristoffersson, and J.-J. Cassiman, “The wide variation of definitions of genetic testing in international recommendations, guidelines and reports,” *Journal of Community Genetics*, vol. 3, no. 2, pp. 113–124, Feb. 2012.
- [4] S. H. Katsanis and N. Katsanis, “Molecular genetic testing and the future of clinical genomics,” *Nature Reviews Genetics*, vol. 14, no. 6, pp. 415–426, May 2013.
- [5] A. Kiezun, K. Garimella, R. Do, N. O. Stitzel, B. M. Neale, P. J. McLaren, N. Gupta, P. Sklar, P. F. Sullivan, J. L. Moran, C. M. Hultman, P. Lichtenstein, P. Magnusson, T. Lehner, Y. Y. Shugart, A. L. Price, P. I. W. de Bakker, S. M. Purcell, and S. R. Sunyaev, “Exome sequencing and the genetic basis of complex traits,” *Nature Genetics*, vol. 44, no. 6, pp. 623–630, May 2012.
- [6] J. O. Kitzman, M. W. Snyder, M. Ventura, A. P. Lewis, R. Qiu, L. E. Simmons, H. S. Gammill, C. E. Rubens, D. A. Santillan, J. C. Murray, H. K. Tabor, M. J. Bamshad, E. E. Eichler, and J. Shendure, “Noninvasive whole-genome sequencing of a human fetus,” *Science Translational Medicine*, vol. 4, no. 137, 137ra76–137ra76, Jun. 2012.
- [7] K. Song, L. Li, and G. Zhang, “Coverage recommendation for genotyping analysis of highly heterologous species using next-generation sequencing technology,” *Scientific Reports*, vol. 6, no. 1, Oct. 2016.
- [8] C.-C. Lo and P. S. G. Chain, “Rapid evaluation and quality control of next generation sequencing data with FaQCs,” *BMC Bioinformatics*, vol. 15, no. 1, Nov. 2014.
- [9] R. M. Leggett, R. H. Ramirez-Gonzalez, B. J. Clavijo, D. Waite, and R. P. Davey, “Sequencing quality assessment tools to enable data-driven informatics for high throughput genomics,” *Frontiers in Genetics*, vol. 4, 2013.
- [10] G. Hannon. (). Fastx-toolkit, [Online]. Available: http://hannonlab.cshl.edu/fastx_toolkit.. (29.10.2019).
- [11] R. Schmieder and R. Edwards, “Quality control and preprocessing of metagenomic datasets,” *Bioinformatics (Oxford, England)*, vol. 27, no. 6, pp. 863–864, Mar. 2011, PMID: 21278185, ISSN: 1367-4811.

- [12] M. Martin, “Cutadapt removes adapter sequences from high-throughput sequencing reads,” *EMBnet.journal*, vol. 17, no. 1, p. 10, May 2011.
- [13] A. M. Bolger, M. Lohse, and B. Usadel, “Trimmomatic: A flexible trimmer for illumina sequence data,” *Bioinformatics*, vol. 30, no. 15, pp. 2114–2120, Apr. 2014.
- [14] L. Smeds and A. Künstner, “ConDeTri - a content dependent read trimmer for illumina data,” *PLoS ONE*, vol. 6, no. 10, M. J. Donlin, Ed., e26314, Oct. 2011.
- [15] C. D. Fabbro, S. Scalabrin, M. Morgante, and F. M. Giorgi, “An extensive evaluation of read trimming effects on illumina NGS data analysis,” *PLoS ONE*, vol. 8, no. 12, J.-S. Seo, Ed., e85024, Dec. 2013.
- [16] H. Li and R. Durbin, “Fast and accurate short read alignment with burrows-wheeler transform,” *Bioinformatics*, vol. 25, no. 14, pp. 1754–1760, May 2009.
- [17] A. Dobin, C. A. Davis, F. Schlesinger, J. Drenkow, C. Zaleski, S. Jha, P. Batut, M. Chaisson, and T. R. Gingeras, “STAR: Ultrafast universal RNA-seq aligner,” *Bioinformatics*, vol. 29, no. 1, pp. 15–21, Oct. 2012.
- [18] B. Langmead and S. L. Salzberg, “Fast gapped-read alignment with bowtie 2,” *Nature Methods*, vol. 9, no. 4, pp. 357–359, Mar. 2012.
- [19] C. Trapnell, L. Pachter, and S. L. Salzberg, “TopHat: Discovering splice junctions with RNA-seq,” *Bioinformatics*, vol. 25, no. 9, pp. 1105–1111, Mar. 2009.
- [20] R. Lindner and C. C. Friedel, “A comprehensive evaluation of alignment algorithms in the context of RNA-seq,” *PLoS ONE*, vol. 7, no. 12, S. L. Salzberg, Ed., e52403, Dec. 2012.
- [21] C. Otto, P. F. Stadler, and S. Hoffmann, “Lacking alignments? the next-generation sequencing mapper segemehl revisited,” *Bioinformatics*, vol. 30, no. 13, pp. 1837–1843, Mar. 2014.
- [22] M. A. DePristo, E. Banks, R. Poplin, K. V. Garimella, J. R. Maguire, C. Hartl, A. A. Philippakis, G. del Angel, M. A. Rivas, M. Hanna, A. McKenna, T. J. Fennell, A. M. Kernysky, A. Y. Sivachenko, K. Cibulskis, S. B. Gabriel, D. Altshuler, and M. J. Daly, “A framework for variation discovery and genotyping using next-generation DNA sequencing data,” *Nature Genetics*, vol. 43, no. 5, pp. 491–498, Apr. 2011.
- [23] *Picard toolkit*, <http://broadinstitute.github.io/picard/>, 2019.
- [24] S. Tian, H. Yan, M. Kalmbach, and S. L. Slager, “Impact of post-alignment processing in variant discovery from whole exome data,” *BMC Bioinformatics*, vol. 17, no. 1, Oct. 2016.
- [25] A. McKenna, M. Hanna, E. Banks, A. Sivachenko, K. Cibulskis, A. Kernysky, K. Garimella, D. Altshuler, S. Gabriel, M. Daly, and M. A. DePristo, “The genome analysis toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data,” *Genome Research*, vol. 20, no. 9, pp. 1297–1303, Jul. 2010.
- [26] E. Garrison and G. Marth, *Haplotype-based variant detection from short-read sequencing*, 2012. arXiv: 1207.3907 [q-bio.GN].

- [27] R. Poplin, P.-C. Chang, D. Alexander, S. Schwartz, T. Colthurst, A. Ku, D. Newburger, J. Dijamco, N. Nguyen, P. T. Afshar, S. S. Gross, L. Dorfman, C. Y. McLean, and M. A. DePristo, "A universal SNP and small-indel variant caller using deep neural networks," *Nature Biotechnology*, vol. 36, no. 10, pp. 983–987, Sep. 2018.
- [28] M. Kumaran, U. Subramanian, and B. Devarajan, "Performance assessment of variant calling pipelines using human whole exome sequencing and simulated data," *BMC Bioinformatics*, vol. 20, no. 1, Jun. 2019.
- [29] J. Chen, X. Li, H. Zhong, Y. Meng, and H. Du, "Systematic comparison of germline variant calling pipelines cross multiple next-generation sequencers," *Scientific Reports*, vol. 9, no. 1, Jun. 2019.
- [30] S. Richards, N. Aziz, S. Bale, D. Bick, S. Das, J. Gastier-Foster, W. W. Grody, M. Hegde, E. Lyon, E. Spector, K. Voelkerding, and H. L. Rehm, "Standards and guidelines for the interpretation of sequence variants: A joint consensus recommendation of the american college of medical genetics and genomics and the association for molecular pathology," *Genetics in Medicine*, vol. 17, no. 5, pp. 405–423, Mar. 2015.
- [31] M. J. Landrum, J. M. Lee, G. R. Riley, W. Jang, W. S. Rubinstein, D. M. Church, and D. R. Maglott, "ClinVar: Public archive of relationships among sequence variation and human phenotype," *Nucleic Acids Research*, vol. 42, no. D1, pp. D980–D985, Nov. 2013.
- [32] P. D. Stenson, M. Mort, E. V. Ball, K. Shaw, A. D. Phillips, and D. N. Cooper, "The human gene mutation database: Building a comprehensive mutation repository for clinical and molecular genetics, diagnostic testing and personalized genomic medicine," *Human Genetics*, vol. 133, no. 1, pp. 1–9, Sep. 2013.
- [33] M. Johns Hopkins University (Baltimore.). Online mendelian inheritance in man, omim® , [Online]. Available: <https://omim.org/>. (29.11.2019).
- [34] M. Lek, K. J. Karczewski, E. V. Minikel, K. E. Samocha, E. Banks, T. Fennell, A. H. O'Donnell-Luria, J. S. Ware, A. J. Hill, B. B. Cummings, T. Tukiainen, D. P. Birnbaum, J. A. Kosmicki, L. E. Duncan, K. Estrada, F. Zhao, J. Zou, E. Pierce-Hoffman, J. Berghout, D. N. Cooper, N. Deflaux, M. DePristo, R. Do, J. Flannick, M. Fromer, L. Gauthier, J. Goldstein, N. Gupta, D. Howrigan, A. Kiezun, M. I. Kurki, A. L. Moonshine, P. Natarajan, L. Orozco, G. M. Peloso, R. Poplin, M. A. Rivas, V. Ruano-Rubio, S. A. Rose, D. M. Ruderfer, K. Shakir, P. D. Stenson, C. Stevens, B. P. Thomas, G. Tiao, M. T. Tusie-Luna, B. Weisburd, H.-H. Won, D. Yu, D. M. Altshuler, D. Ardissino, M. Boehnke, J. Danesh, S. Donnelly, R. Elosua, J. C. Florez, S. B. Gabriel, G. Getz, S. J. Glatt, C. M. Hultman, S. Kathiresan, M. Laakso, S. McCarroll, M. I. McCarthy, D. McGovern, R. McPherson, B. M. Neale, A. Palotie, S. M. Purcell, D. Saleheen, J. M. Scharf, P. Sklar, P. F. Sullivan, J. Tuomilehto, M. T. Tsuang, H. C. Watkins, J. G. Wilson, M. J. Daly, and D. G. MacArthur, "Analysis of protein-coding genetic variation in 60, 706 humans," *Nature*, vol. 536, no. 7616, pp. 285–291, Aug. 2016.

- [35] K. J. Karczewski, L. C. Francioli, G. Tiao, B. B. Cummings, J. Alföldi, Q. Wang, R. L. Collins, K. M. Laricchia, A. Ganna, D. P. Birnbaum, L. D. Gauthier, H. Brand, M. Solomonson, N. A. Watts, D. Rhodes, M. Singer-Berk, E. M. England, E. G. Seaby, J. A. Kosmicki, R. K. Walters, K. Tashman, Y. Farjoun, E. Banks, T. Poterba, A. Wang, C. Seed, N. Whiffin, J. X. Chong, K. E. Samocha, E. Pierce-Hoffman, Z. Zappala, A. H. O'Donnell-Luria, E. V. Minikel, B. Weisburd, M. Lek, J. S. Ware, C. Vittal, I. M. Armean, L. Bergelson, K. Cibulskis, K. M. Connolly, M. Covarrubias, S. Donnelly, S. Ferriera, S. Gabriel, J. Gentry, N. Gupta, T. Jeandet, D. Kaplan, C. Llanwarne, R. Munshi, S. Novod, N. Petrillo, D. Roazen, V. Ruano-Rubio, A. Saltzman, M. Schleicher, J. Soto, K. Tibbetts, C. Tolonen, G. Wade, M. E. Talkowski, B. M. Neale, M. J. Daly, and D. G. M. and, "Variation across 141, 456 human exomes and genomes reveals the spectrum of loss-of-function intolerance across human protein-coding genes," Jan. 2019.
- [36] X. Liu, C. Wu, C. Li, and E. Boerwinkle, "dbNSFP v3.0: A one-stop database of functional predictions and annotations for human nonsynonymous and splice-site SNVs," *Human Mutation*, vol. 37, no. 3, pp. 235–241, Jan. 2016.
- [37] J. T. den Dunnen, R. Dalgleish, D. R. Maglott, R. K. Hart, M. S. Greenblatt, J. McGowan-Jordan, A.-F. Roux, T. Smith, S. E. Antonarakis, and P. E. T. and, "HGVS recommendations for the description of sequence variants: 2016 update," *Human Mutation*, vol. 37, no. 6, pp. 564–569, Mar. 2016.
- [38] K. Wang, M. Li, and H. Hakonarson, "ANNOVAR: Functional annotation of genetic variants from high-throughput sequencing data," *Nucleic Acids Research*, vol. 38, no. 16, e164–e164, Jul. 2010.
- [39] W. McLaren, L. Gil, S. E. Hunt, H. S. Riat, G. R. S. Ritchie, A. Thormann, P. Flicek, and F. Cunningham, "The ensembl variant effect predictor," *Genome Biology*, vol. 17, no. 1, Jun. 2016.
- [40] P. Cingolani, A. Platts, L. L. Wang, M. Coon, T. Nguyen, L. Wang, S. J. Land, X. Lu, and D. M. Ruden, "A program for annotating and predicting the effects of single nucleotide polymorphisms, SnpEff," *Fly*, vol. 6, no. 2, pp. 80–92, Apr. 2012.
- [41] S. Hoon, "Biopipe: A flexible framework for protocol-based bioinformatics analysis," *Genome Research*, Jul. 2003.
- [42] T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Senger, M. Greenwood, T. Carver, K. Glover, M. R. Pocock, A. Wipat, and P. Li, "Taverna: A tool for the composition and enactment of bioinformatics workflows," *Bioinformatics*, vol. 20, no. 17, pp. 3045–3054, Jun. 2004.
- [43] J. Goecks, A. Nekrutenko, J. Taylor, and T. G. Team, "Galaxy: A comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences," *Genome Biology*, vol. 11, no. 8, R86, 2010.
- [44] F. Halbritter, H. J. Vaidya, and S. R. Tomlinson, "GeneProf: Analysis of high-throughput sequencing experiments," *Nature Methods*, vol. 9, no. 1, pp. 7–8, Dec. 2011.
- [45] S. P. Shah, D. Y. He, J. N. Sawkins, J. C. Druce, G. Quon, D. Lett, G. X. Zheng, T. Xu, and B. F. Ouellette, *BMC Bioinformatics*, vol. 5, no. 1, p. 40, 2004.

- [46] L. Goodstadt, “Ruffus: A lightweight python library for computational pipelines,” *Bioinformatics*, vol. 26, no. 21, pp. 2778–2779, Sep. 2010.
- [47] K. Taura, T. Matsuzaki, M. Miwa, Y. Kamoshida, D. Yokoyama, N. Dun, T. Shibata, C. S. Jun, and J. Tsujii, “Design and implementation of GXP make – a workflow system based on make,” in *2010 IEEE Sixth International Conference on e-Science*, IEEE, Dec. 2010.
- [48] S. P. Sadedin, B. Pope, and A. Oshlack, “Bpipe: A tool for running and managing bioinformatics pipelines,” *Bioinformatics*, vol. 28, no. 11, pp. 1525–1526, Apr. 2012.
- [49] P. D. Tommaso, M. Chatzou, E. W. Floden, P. P. Barja, E. Palumbo, and C. Notredame, “Nextflow enables reproducible computational workflows,” *Nature Biotechnology*, vol. 35, no. 4, pp. 316–319, Apr. 2017.
- [50] J. Koster and S. Rahmann, “Snakemake—a scalable bioinformatics workflow engine,” *Bioinformatics*, vol. 28, no. 19, pp. 2520–2522, Aug. 2012.
- [51] *Dna-seq-gatk-variant-calling*, <https://github.com/snakemake-workflows/dna-seq-gatk-variant-calling>, 2019.
- [52] J. L. Causey, C. Ashby, K. Walker, Z. P. Wang, M. Yang, Y. Guan, J. H. Moore, and X. Huang, “DNAP: A pipeline for DNA-seq data analysis,” *Scientific Reports*, vol. 8, no. 1, May 2018.
- [53] J. Leipzig, “A review of bioinformatic pipeline frameworks,” *Briefings in Bioinformatics*, bbw020, Mar. 2016.
- [54] Q. Li and K. Wang, “InterVar: Clinical interpretation of genetic variants by the 2015 ACMG-AMP guidelines,” *The American Journal of Human Genetics*, vol. 100, no. 2, pp. 267–280, Feb. 2017.
- [55] S. Köhler, S. C. Doelken, C. J. Mungall, S. Bauer, H. V. Firth, I. Bailleul-Forestier, G. C. M. Black, D. L. Brown, M. Brudno, J. Campbell, D. R. FitzPatrick, J. T. Eppig, A. P. Jackson, K. Freson, M. Girdea, I. Helbig, J. A. Hurst, J. Jähn, L. G. Jackson, A. M. Kelly, D. H. Ledbetter, S. Mansour, C. L. Martin, C. Moss, A. Mumford, W. H. Ouwehand, S.-M. Park, E. R. Riggs, R. H. Scott, S. Sisodiya, S. V. Vooren, R. J. Wapner, A. O. M. Wilkie, C. F. Wright, A. T. V.-v. Silfhout, N. de Leeuw, B. B. A. de Vries, N. L. Washington, C. L. Smith, M. Westerfield, P. Schofield, B. J. Ruef, G. V. Gkoutos, M. Haendel, D. Smedley, S. E. Lewis, and P. N. Robinson, “The human phenotype ontology project: Linking molecular biology and disease through phenotype data,” *Nucleic Acids Research*, vol. 42, no. D1, pp. D966–D974, Nov. 2013.
- [56] M. Zhao, Q. Wang, Q. Wang, P. Jia, and Z. Zhao, “Computational tools for copy number variation (CNV) detection using next-generation sequencing data: Features and perspectives,” *BMC Bioinformatics*, vol. 14, no. S11, Sep. 2013.

Tezden Üretilmiş Yayınlar

İletişim Bilgileri: ozyyk61@gmail.com

Konferans Bildirisi

1. Kalyon, O., Yılmaz, A., Muslumanoglu M.H. (2019). “AUTOMATION OF GENETIC TESTING AND REPORTING WITH SNAKEMAKE” HIBIT’19, 17-19 October 2019, İzmir