

İÇİNDEKİLER

2.....	BİOLANG TANITIMI
8.....	SEVDA SAĞLAMTAŞ
10.....	EYLÜL NAZ ÇELİK
13.....	ZEYNEP KARATAŞ
18.....	DEFNE ÇİFÇİ
21	RANA SABİHA ÇEVİK
25.....	SENA ZEYNEP GÖRGÜN
28.....	SUDE BAŞALAN
31.....	BERFİN ZÜMRA KARACAKAYA
34.....	KAYNAKÇA

BİOLANG TANITIMI

1. BİOLANG'un Tanıtımı

BİOLANG, biyoinformatik, genetik mühendisliği ve biyolojik veri analizi alanlarına yönelik olarak tasarlanmış yenilikçi bir programlama dilidir. BİOLANG, özellikle biyolojik verilerin işlenmesi, genetik analizlerin yapılması ve hastalık tahminleri gibi biyoteknolojik uygulamalar için optimize edilmiştir. Genetik verilerin yönetilmesi, analiz edilmesi ve biyolojik modellerin oluşturulması süreçlerini kolaylaştırmak amacıyla geliştirilen BİOLANG, biyoloji ve bilgisayar bilimlerini birleştiren güçlü bir altyapıya sahiptir. Bu dil, biyolojik verilerin işlenmesi için gerekli tüm temel işlevleri içeren kapsamlı kütüphanelerle donatılmıştır.

BİOLANG, genetik ve biyolojik araştırmalar için yüksek verimlilik sağlayan, sadeleştirilmiş ve kullanıcı dostu bir syntax yapısına sahiptir. Bu sayede biyolojik verilerle çalışan bilim insanları ve araştırmacılar, karmaşık genetik hesaplamaları daha hızlı ve etkili bir şekilde gerçekleştirebilmektedir. BİOLANG'ın sunduğu temel avantajlar arasında biyolojik veri yönetimi, hastalık tahminleri, genetik analizler ve biyolojik modelleme yer alır.

2. BİOLANG'un Avantajları

2.1. Hedef Alan Odaklı Yapı

BİOLANG, biyoinformatik ve genetik mühendisliği alanlarına özgü ihtiyaçları karşılamak amacıyla özel olarak geliştirilmiş kütüphanelere sahiptir. Bu kütüphaneler, genetik analizlerin temel bileşenleri olan DNA dizilimlerinin işlenmesi, genetik mutasyonların tespiti, hastalık sınıflandırma ve prediktif analizler gibi işlemleri destekler. Örneğin:

- **BİODATA:** Biyolojik veritabanlarıyla bağlantı kurarak biyolojik veri setlerini kolayca yükleyip işleyebilir.
- **GENOLİB ve PHENOLİB:** Genetik ve fenotipik verilerin analizini yaparak genetik riskleri hesaplar.
- **PREDICTIVE_DISEASE:** Genetik veriler kullanarak hastalık tahminleri yapar.
- **AB0.MAP ve DNA.MAP:** Genetik verilerin haritalanmasını sağlar ve DNA dizilimlerinden biyolojik veriler elde edilmesini kolaylaştırır.
- **BIOFLOW:** Genetik verilerin analizinden elde edilen verilerin işlenmesine yardımcı olan bir kütüphanedir.

2.2. Yüksek Performans ve Optimizasyon

BİOLANG, büyük biyolojik veri setleriyle çalışırken yüksek performans sağlar. Genetik analizler için özel olarak optimize edilmiş algoritmalar, özellikle büyük veri setleriyle çalışırken performans kaybını engeller. Bunun yanı sıra, dilin paralel işleme yetenekleri sayesinde veri analiz süreçleri hızlandırılabilir ve zaman tasarrufu sağlanabilir. BİOLANG, genetik verilerle çalışırken zaman ve kaynak kullanımı açısından verimliliği artıran bir altyapı sunar.

2.3. Kullanıcı Dostu Syntax

BİOLANG, biyolojik verilerin işlenmesi için geliştirilen basit ve anlaşılır bir syntax yapısına sahiptir. Genetik analizlerde kullanılan karmaşık hesaplamalar, BİOLANG dilinde doğal dil benzeri bir yapıyla ifade edilebilir. Bu, kullanıcıların yazılım geliştirme sürecinde karşılaştıkları zorlukları minimize eder ve bilim insanlarının daha verimli bir şekilde çalışmasını sağlar. Örneğin, DNA dizilimlerini analiz etmek ve mutasyonları tespit etmek gibi işlemler BİOLANG dilinde oldukça sade bir şekilde yapılabilir.

2.4. Entegre Biyolojik Veri Yönetimi

BİOLANG, biyolojik veritabanlarıyla kolayca entegre olabilen bir yapıya sahiptir. Veriler, BİODATA kütüphanesi ile hızlıca yüklenebilir ve genetik analizler için uygun hale getirilebilir. Ayrıca, hastalık tahminleri ve genetik analizler, PHENOLIB ve PREDICTIVE_DISEASE kütüphaneleri ile yapılabilir. Bu özellikler, araştırmaların daha hızlı ve etkili bir şekilde yapılmasına olanak tanır.

2.5. Esnek Kütüphane Desteği

BİOLANG, genetik analizlere yönelik esnek kütüphane desteği sunar. Bu kütüphaneler, araştırmacıların ihtiyaç duyduğu özel hesaplamaları yapabilmelerine olanak tanır. Genetik mutasyonları analiz etmek, hastalıkları sınıflandırmak ve genetik riskleri hesaplamak gibi işlemler için optimize edilmiş araçlar içerir.

3. BİOLANG ile Python Karşılaştırması

Özellik	BİOLANG	Python
Hedef	Biyoinformatik, Genetik Mühendisliği	Genel Amaçlı

Özel Kütüphaneler	BİODATA, GENOLİB, PHENOLİB, PREDICTIVE_DISEASE, AB0.MAP, BIOFLOW, DNA.MAP	Biopython, NumPy, Pandas, vb.
Performans	Genetik işlemler için optimize edilmiş	Genel amaçlı, genetik işlemler için ek
Okunabilirlik	Doğal dil benzeri syntax	Yüksek okunabilirlik, fakat genetik işlemler için daha
Veri	Entegre biyolojik veri analizi	Ek kütüphanelerle
Topluluk	Yeni, sınırlı topluluk	Geniş bir topluluk ve kaynak ekosistemi
Esneklik	Genetik ve biyoinformatik alanları	Çok esnek, ancak genetik

4. BİOLANG'ın Dezavantajları

- **Topluluk ve Ekosistem:** BİOLANG henüz gelişme aşamasındadır ve Python gibi büyük bir kullanıcı topluluğuna sahip değildir. Bu, dilin geniş bir kütüphane ekosistemine ve destek altyapısına sahip olmamasına yol açmaktadır.
- **Genel Kullanım:** BİOLANG, biyoinformatik ve genetik mühendisliği odaklı bir dil olduğu için, genel yazılım geliştirme ve diğer alanlarda esneklik açısından sınırlı kalabilir.
- **Kütüphane ve Genişletilebilirlik:** BİOLANG, genetik analizlere yönelik zengin bir kütüphane desteği sunsa da, genel yazılım geliştirme ihtiyaçlarına yönelik kütüphane desteği sınırlıdır.

5. BİOLANG'ın Geleceği

BİOLANG, biyoinformatik ve genetik mühendisliği alanlarında önemli bir boşluğu doldurabilir ve biyoteknolojik araştırmalar için güçlü bir araç olabilir. Gelecekte, BİOLANG'ın şu alanlarda gelişmesi beklenmektedir:

- **Makine Öğrenimi Entegrasyonu:** BİOLANG'a makine öğrenimi algoritmaları entegre edilerek genetik veriler üzerinde daha sofistike modeller yapılabilir.
- **Bulut Entegrasyonu:** Genetik analizler için büyük veri setleriyle çalışılacak bulut tabanlı platformlar geliştirilebilir.
- **Küresel Kullanım:** BİOLANG, üniversitelerde biyoinformatik eğitimi için standart bir araç haline gelebilir ve global biyoteknoloji topluluğunda daha geniş bir yer edinebilir.

6. Etik İlkeler

BIOLANG, genetik verilerin hassasiyetini göz önünde bulundurarak aşağıdaki etik kurallar çerçevesinde geliştirilmiştir:

- **Gizlilik ve Güvenlik:** Kullanıcı verileri şifrelenerek saklanır ve yalnızca yetkili kullanıcıların erişimine izin verilir.
- **Veri Kullanımı ve Etik Dışılık:** Genetik verilerin etik dışı kullanımlarını engellemek amacıyla güvenlik protokollerine sahiptir.
- **Açık Kaynak:** BIOLANG, araştırma ve eğitim amacıyla ücretsiz kullanım sağlar ve genetik biliminin ilerlemesine katkı sunar.

7. Gelecekteki Gelişmeler

BIOLANG, başlangıçta biyolojik verilerin işlenmesi ve genetik analizler için tasarlanmış olsa da, gelecekte daha geniş kapsamlı işlevler eklenmesi planlanmaktadır. Veri görselleştirme, veri tabanı yönetimi gibi alanlarda yeni özellikler eklenecek ve bu özellikler BIOLANG'ı daha güçlü ve kapsamlı bir dil haline getirecektir.

8. Sytnax Yapısı:

<code>continue</code>	<code>skipStep</code>	Döngüdeki bir adımı atlar. Örneğin: <code>skipStep</code>
<code>==</code>	<code>matches</code>	Eşitlik kontrolü yapar. Örneğin: <code>if (geneSequence matches "ATCG")</code>
<code>!=</code>	<code>notMatches</code>	Eşitsizlik kontrolü yapar. Örneğin: <code>if (geneSequence notMatches "GCAT")</code>
<code>typeof</code>	<code>natureOf</code>	Bir değişkenin türünü sorgular. Örneğin: <code>natureOf geneSequence</code>
<code>instanceof</code>	<code>isKindOf</code>	Bir nesnenin türünü kontrol eder. Örneğin: <code>isKindOf mutation</code>
<code>return</code>	<code>yield</code>	Fonksiyondan değer döndürür. Örneğin: <code>yield mutationType</code>

Ekstra Özellikler:

Geleneksel Terimler	Yeni Syntax Yapıları	Açıklama
<code>Math.random()</code>	<code>geneRandom()</code>	Rastgele genetik bir değer üretir. Örneğin: <code>geneRandom()</code>
<code>Array.isArray()</code>	<code>isGeneList()</code>	Bir dizinin genetik liste olup olmadığını kontrol eder. Örneğin: <code>isGeneList(geneList)</code>

<code>import</code>	<code>link to</code>	Dış kütüphaneleri veya veritabanlarını bağlamak için kullanılır. Örneğin: <code>link to mutationDatabase</code>
<code>try</code>	<code>observe</code>	Hata kontrolü yapmak için kullanılır. Örneğin: <code>observe geneSequence</code>
<code>catch</code>	<code>respond</code>	Hata meydana geldiğinde yapılacak işlemi belirler. Örneğin: <code>respond("Mutation data not found")</code>
<code>finally</code>	<code>alwaysDo</code>	Hata sonrası yapılacak işlemleri belirtir. Örneğin: <code>alwaysDo cleanup()</code>
<code>new</code>	<code>create</code>	Yeni bir nesne oluşturmak için kullanılır. Örneğin: <code>create geneSequence()</code>
<code>delete</code>	<code>remove</code>	Nesne veya öğe silmek için kullanılır. Örneğin: <code>remove geneSequence</code>
<code>array.push()</code>	<code>addTo(geneList)</code>	Diziye yeni eleman ekler. Örneğin: <code>addTo(geneList, "C789")</code>
<code>array.pop()</code>	<code>removeLast(geneList)</code>	Diziden son öğeyi çıkartır. Örneğin: <code>removeLast(geneList)</code>
<code>array.length</code>	<code>geneList.length()</code>	Dizi uzunluğunu alır. Örneğin: <code>geneList.length()</code>
<code>break</code>	<code>exitLoop</code>	Döngüyü bitirir. Örneğin: <code>exitLoop("Gene mutation detected")</code>

<code>int</code> , <code>float</code>	<code>number</code> , <code>decimal</code>	Sayı türleri, sayısal değerleri ifade eder. Örneğin: <code>number mutationCount = 5</code>
<code>string</code>	<code>geneSequence</code> , <code>trait</code>	Metin türü için daha biyolojik terimler kullanılır. Örneğin: <code>geneSequence = "ATCG"</code>
<code>let</code> , <code>var</code> , <code>const</code>	<code>geneSequence::</code> , <code>trait::</code>	Değişken tanımlamak için kullanılan yapılar. Örneğin: <code>trait:: dominantTrait = "Brown Eyes"</code>
<code>array</code>	<code>geneList</code> , <code>mutationSet</code>	Diziler, genetik liste veya mutasyon seti olarak ifade edilir. Örneğin: <code>geneList = ["A123", "B456"]</code>
<code>&&</code> , <code>`</code>		<code>`</code>
<code>==</code>	<code>equals</code>	Karşılaştırma yapmak için kullanılır. Örneğin: <code>if (geneSequence equals "ATCG")</code>
<code>!=</code>	<code>notEquals</code>	Eşitsizlik kontrolü için kullanılır. Örneğin: <code>if (trait notEquals "Short Hair")</code>
<code>class</code>	<code>trait</code>	Nesne tanımlamak için biyolojik terimler kullanılır. Örneğin: <code>trait EyeColor</code>
<code>object</code>	<code>geneticTrait</code>	Nesneler yerine genetik özellikler tanımlanır. Örneğin: <code>geneticTrait mutation1</code>

<code>if</code>	<code>is?</code>	Koşul kontrolü için kullanılır. Örneğin: <code>is? geneSequence == "ATCG"</code>
<code>else</code>	<code>otherwise</code>	<code>else</code> durumunu ifade eder. Örneğin: <code>otherwise express("No match found")</code>
<code>else if</code>	<code>elseIs</code>	Alternatif koşulları kontrol eder. Örneğin: <code>elseIs mutation == "A123"</code>
<code>for</code>	<code>for (each: ...)</code>	Dizileri veya koleksiyonları dolaşmak için. Örneğin: <code>for (each: geneList)</code>
<code>while</code>	<code>while (condition)</code>	Koşul doğru olduğu sürece döngü çalıştırır. Örneğin: <code>while (mutationCount < 10)</code>
<code>function</code>	<code>function</code>	Fonksiyon tanımlamak için kullanılır. Örneğin: <code>function calculateRisk()</code>
<code>return</code>	<code>return</code>	Fonksiyondan değer döndürür. Örneğin: <code>return riskScore</code>
<code>print</code>	<code>express</code>	Çıktı vermek için kullanılır. Örneğin: <code>express("Gene Sequence: " + geneSequence)</code>
<code>break</code>	<code>exitLoop</code>	Döngüyü sonlandırmak için kullanılır. Örneğin: <code>exitLoop("No more mutations")</code>
<code>continue</code>	<code>skipStep</code>	↓ güde bir adımı atlamak için kullanılır. Örneğin: <code>skipStep</code>

SEVDA SAĞLAMTAŞ

1. DNA Replikasyonu (DNA_Rep)

Amaç:

Verilen bir DNA dizisinin replikasyonunu gerçekleştirir ve yeni bir tamamlayıcı DNA dizisi döndürür.

Girdiler:

dna_sequence: Bir DNA dizisini temsil eden string. Örneğin, "ATCGGTA".

Çıktı:

Tamamlayıcı DNA dizisi. Örneğin, "TAGCCAT".

İşleyiş:

- Her bir nükleotid, karşılık gelen tamamlayıcı nükleotid ile değiştirilir:
 - A (Adenin) -> T (Timin)
 - T (Timin) -> A (Adenin)
 - C (Sitozin) -> G (Guanin)
 - G (Guanin) -> C (Sitozin)
- Tamamlayıcı dizi döndürülür.

Avantajları:

- Basit ve hızlı bir şekilde DNA dizisinin tamamlayıcı dizisini üretir.
- Genetik analiz ve biyoinformatik projelerinde kullanılabilir.

Dezavantajları:

- Girdi dizisinin geçerliliği kontrol edilmezse hata verebilir.
- Yalnızca DNA dizilerini işler, RNA dizileri için uygun değildir.

2. JSON Dosyasına Veri Kaydetme (save_data_to_json)

Amaç:

Verilen veriyi bir JSON dosyasına kaydeder.

Girdiler:

data: JSON formatında kaydedilecek veri.

file_path: Dosyanın kaydedileceği yol.

Çıktı:

Veri başarıyla dosyaya kaydedilir.

İşleyiş:

1. Verilen veri JSON formatına dönüştürülür.
2. Belirtilen dosya yoluna yazılır.
3. İşlem başarılıysa dosya kaydedilir.

Avantajları:

1. Veriyi JSON formatında saklayarak yapılandırılmış veri kaynağı sağlar.
2. JSON formatı birçok programlama diliyle uyumludur.

Dezavantajları:

1. Dosya yazma sırasında hata oluşursa veri kaydedilemez.
2. Büyük veri setlerinde performans düşebilir.

3. İlaç Yan Etkilerini Hesaplama (med_side_effect_risk)

Amaç:

Belirli bir genotip ve ilaç bilgisine dayalı olarak metabolizma hızını belirler ve risk seviyesini döndürür.

Girdiler:

genotip: Kullanıcının genotip varyasyonu (örneğin, "*1", "*2", "*3").

med: İlaç adı veya ilgili CYP450 enzim grubu (örneğin, "CYP2D6").

Çıktı:

Genotip ve ilaç bilgilerine dayalı olarak risk seviyesi (düşük, orta veya yüksek).

İşleyiş:

1. CYP450 gen varyasyonları ve bunların karşılık gelen metabolizma hızları tanımlanır.
2. Kullanıcının genotip ve ilaç bilgileri karşılaştırılarak metabolizma hızı belirlenir.
3. Metabolizma hızına bağlı olarak risk seviyesi belirlenir ve döndürülür.
4. Eğer ilaç veya genotip veritabanında yoksa hata mesajı döndürülür.

Avantajları:

1. Farmakogenetik analiz için kullanılabilir.
2. Kişiselleştirilmiş tedavi planları ve ilaç risk değerlendirmesi sunar.
3. Genetik varyasyonlara bağlı olarak yan etkilerin risk seviyesini değerlendirir.

Dezavantajları:

1. Veritabanında bulunmayan ilaç ve genotipler için sonuç vermez.
2. İlaç ve genotip bilgileri doğru girilmezse yanlış sonuçlara yol açabilir.

EYLÜL NAZ ÇELİK

BLOOD_PROBABİLİTY

Fonksiyonun Tanıtımı:

Anne ve babanın kan grubu ile Rh faktörüne dayalı olarak, bir çocuğun olası kan gruplarını ve Rh faktörlerini tahmin etmek için tasarlanmıştır.

Özellikleri:

Kullanıcıdan yalnızca anne ve babanın kan grupları istenir. Bu sayede kullanımı kolaydır. Genetik kombinasyonlar ile bu fonksiyon anne ve babadan gelen alleller doğrultusunda bilimsel çıktılar üretir. Ve tüm kan grupları ve Rh faktörleri desteklenir.

İşleyişi:

Kullanıcı anne ve babanın kan gruplarını ve Rh faktörlerini programa girer. Genetik kombinasyonlar yoluyla çocuğun alabileceği olası kan gruplarını ve Rh faktörlerini hesaplar. Sonuçlar, ekranda olası kan grupları ve Rh faktörleri olarak görüntülenir.

Fonksiyonun Avantajı ve Dezavantajı:

Kolay anlaşılır bir fonksiyondur, karmaşıklık içermez. Biyoloji temelli yazılımlara giriş yapmak isteyenler için ideal yapıdadır. Fonksiyonun dezavantajı ise yalnızca kan grubu tahmin etmesidir ve veri setleri içermemesidir. Bu nedenle büyük ölçekli genetik analizler için performans sorunu yaşanabilir.

SAVE_DATA_TO_CSV

Fonksiyon Tanıtımı:

Bu fonksiyon, bir sözlük listesi şeklinde verilen verileri CSV dosyasına kaydeder. Özellikle sağlık verileri vb. bilgilerin kaydedilmesi için kullanılır.

Özellikleri:

Sözlükteki anahtarlar ile CSV dosyasının başlıkları otomatik yazılır. Yazdırma sırasında oluşan hatalar observe-except blokları ile bildirilir ve farklı dil karakter setlerini desteklemek için utf-8 yapısı kullanılır.

İşleyişi:

Sözlük listesi (data) ve hedef CSV dosyasının adı (output_filepath) parametre olarak alınır. CSV yazımı için Python'da bulunan csv.DictWriter sınıfı kullanılır. writeheader() ile başlıklar yazılır, ardından writerows() ile tüm veriler output.csv dosyasına kaydedilir.

Fonksiyonun Avantaj ve Dezavantajları:

Fonksiyon, Python'da sık kullanılan sözlük listesi formatını destekler. Bu formatta verileri hazırlamak daha kolaydır. Kod, dosya yazma sırasında oluşabilecek hataları yakalayarak kullanıcıya bilgi verir. Bu, daha dayanıklı bir uygulama sağlar. Sözlük anahtarlarını otomatik olarak CSV dosyasının başlıkları olarak kullanır. Kullanıcının başlıkları manuel olarak tanımlamasına gerek yoktur. utf-8 kodlaması sayesinde Türkçe karakterler ve diğer özel karakterlerle uyumludur.

Fonksiyonun dezavantajları ise yalnızca sözlük listesi formatında veri kabul eder. Veri başka bir formatta (örneğin, pandas DataFrame veya numpy array) ise önce dönüştürme yapılması gerekir. Diğer kütüphanelerde, eksik verileri doldurma (imputation), veri türü dönüştürme veya dosya formatı doğrulama gibi gelişmiş özellikler sunulur. Bu fonksiyon yalnızca temel veri kaydetme işlemini gerçekleştirir. CSV'nin başlıkları doğrudan sözlük anahtarlarından alınır. Kullanıcı başlıkların sırasını veya adını değiştirmek isterse manuel işlem yapması gerekir.

CALCULATE_CF_RISK

Fonksiyon Tanıtımı:

Bu fonksiyon, kistik fibrozis gibi otozomal resesif genetik hastalıklar için risk analizi yapar. Anne ve babanın genetik durumlarına (sağlıklı, taşıyıcı veya hasta) göre, çocuğun hastalığa yakalanma ve taşıyıcı olma ihtimalini hesaplar. Kod, her kombinasyona göre hastalık ve taşıyıcılık risklerini değerlendirir.

Özellikleri:

Kapsamlı Durum Analizi: Anne ve baba sağlıklı, taşıyıcı veya hasta olabilen 9 farklı kombinasyonu değerlendirir.

Doğrudan Kullanıcı Girdisi: Kullanıcı, anne ve babanın sağlık durumunu girer; kod otomatik olarak riski hesaplar.

Kolay Anlaşılır Çıktılar: Çocuğun hasta olma ve taşıyıcı olma ihtimalleri ayrı ayrı yüzdelik olarak belirtilir.

İşleyişi:

Anne ve babanın sağlık durumları sorulur (sağlıklı, taşıyıcı, hasta). Anne ve babanın durumu fonksiyonda tanımlı kombinasyonlarla eşleştirilir. Kod, her kombinasyon için hasta olma (%disease_risk) ve taşıyıcı olma (%carrier_risk) oranlarını hesaplar. Çocuğun genetik hastalık ve taşıyıcılık riski yüzdesel olarak belirtilir.

Fonksiyonun Avantajı ve Dezavantajı:

Bu fonksiyon, temel biyoloji ve genetik risk analizleri için basit bir çözüm sunar. Kullanımı kolay ve açıktır. Eğitim için de kullanılabilecek düzeydedir. Ancak daha geniş kapsamlı analizler, görselleştirme gibi gereksinimler için yetersizdir.

ZEYNEP KARATAŞ

1) *Kalp Atış Hızına Göre Sağlık Durumunu Analiz Eden Fonksiyon(heartRateStatus):*

Amaç: Kullanıcının yaşı ve dinlenme halindeki kalp atış hızını değerlendirerek "Düşük", "Normal" veya "Yüksek" risk seviyesini belirler.

Biolang:

Avantajlar:

- *Hedef Odaklılık:* Yaş gruplarına ve kalp atış hızı kategorilerine özel analizler önceden tanımlanmıştır.
- *Hata Kontrolü:* Hatalı veri girişi için otomatik mekanizmalar ve açıklayıcı çıktılar (express komutu) sunar.
- *Sağlık Uzmanları İçin Tasarlanmıştır:* Bilimsel olmayan kullanıcılar bile rahatça analiz yapabilir.

Dezavantajlar:

- *Sabite Dayalı:* Yaş grupları ve kalp atış sınırları gibi sabit parametreler esneklikten yoksundur; farklı standartlara uyum sağlaması zordur.
- *Genişletilebilirlik:* Çoklu veri kaynakları veya büyük veri ile çalışmak için sınırlı kapasiteye sahiptir.

Python:

Avantajlar:

- *Veri Görselleştirme:* matplotlib ve seaborn gibi kütüphanelerle kalp atış hızını grafikler ve dağılım analizleri ile görselleştirebilir.
- *Gelişmiş Analiz:* Yaş, cinsiyet ve diğer parametrelerin birlikte analiz edilmesi için karmaşık algoritmalar geliştirilebilir.
- *Büyük Veri ve Hız:* Büyük veri kümelerinde paralel işleme ve hız optimizasyonu sağlar.

Dezavantajlar:

- Biolang kadar kullanıcı dostu değildir ve hata kontrolü manuel yapılmalıdır.
- Yeni başlayanlar için daha fazla efor gerektirir.

2) *Kan Şekeri Seviyesini Analiz Eden Fonksiyon(glucose_level_analysis)*

Amaç: Kan şekeri seviyesini açlık ve tokluk durumlarına göre analiz ederek kullanıcının sağlık durumunu ("Düşük Kan Şekeri", "Normal", "Prediyabet", "Diyabet") sınıflandırır.

Biolang:

Avantajlar:

- *Sağlık alanına özgü:* Fonksiyonlar spesifik kategoriler (örneğin, "Normal", "Prediyabet") ve hata kontrol mekanizmalarıyla doğrudan sağlık verileri üzerinde çalışacak şekilde tasarlanmıştır.
- *Kullanıcı Dostu:* Kodlama bilmeyen sağlık uzmanları için sade ve anlaşılır syntax (is?, elsels, yield).
- *Hızlı Sonuçlar:* Küçük veri kümelerinde hızlı analiz sağlar; hata kontrolü ve sonuç üretimi entegre edilmiştir.

Dezavantajlar:

- *Esneklik Eksikliği:* Sağlık alanına özel sabit mantık yapıları, kullanıcıların yeni kategoriler veya özel analiz eklemesini zorlaştırır.
- *Veri İşleme Limitasyonu:* Büyük veri kümeleri üzerinde analiz yapmak veya veri madenciliği algoritmaları geliştirmek için uygun değildir.

- *Performans*: Paralel işlemler ve optimizasyon seçenekleri eksik.

Python:

Avantajlar:

- *Geniş Kütüphane Desteği*: pandas, numpy ve matplotlib gibi güçlü kütüphanelerle verilerin hızlı işlenmesi ve görselleştirilmesi mümkün.
- *Esneklik*: Farklı kategoriler ve özel mantık yapıları eklemeye olanak tanır.
- *Makine Öğrenmesi Entegrasyonu*: Diyabet analizi gibi alanlarda tahminsel modeller oluşturmak için scikit-learn gibi kütüphanelerle çalışabilir.

Dezavantajlar:

- Yeni başlayanlar için daha dik bir öğrenme eğrisi vardır.
- Biolang kadar net kategorik yapılar sunmadığından, özel bir yapı oluşturmak zaman alır.

3) *X Kromozomuna Bağlı Hastalık Riskini Analiz Eden Fonksiyon(x_linked_disease_risk):*

Amaç: Kullanıcının cinsiyetine ve genetik mutasyon durumuna göre hastalık riskini belirler.

Biolang:

Avantajlar:

- *Spesifik Çözüm*: Cinsiyet ve genetik mutasyon gibi biyomedikal analizlere odaklanır.

- *Kolay Veri Yükleme:* Genetik veriler için hazır veri yükleme fonksiyonları sunar (`load_data_from_csv`).
- *Doğrudan Hedefe Ulaşım:* Kullanıcıdan cinsiyet ve genetik durumu alarak hızlı şekilde sonuç üretir.

Dezavantajlar:

- *Kapsam Sınırlılığı:* Daha karmaşık genetik analizler, örneğin gen dizilimi veya derin öğrenme algoritmaları, yapılamaz.
- *Özelleştirme Eksikliği:* Yeni genetik kategoriler veya analiz yolları eklemek zordur.

Python:

Avantajlar:

- *Genel Kullanım:* Genetik analizler dışında birçok biyomedikal ve veri bilimi alanında da kullanılabilir.
- *Kütüphane Zenginliği:* biopython veya numpy gibi araçlarla genetik veri analizi yapılabilir.
- *Makine Öğrenmesi:* Genetik mutasyon riskini tahmin etmek için makine öğrenmesi modelleri entegre edilebilir.

Dezavantajlar:

- Biolang kadar kullanıcı dostu bir yapı sunmaz, daha fazla kodlama gerektirir.
- Belirli bir soruna hızlı çözüm sağlama konusunda Biolang kadar etkili değildir.

Sonuç

- **Biolang Avantajları:**

- Sağlık ve biyomedikal alanlar için özelleştirilmiş yapılar ve sade syntax ile kullanıcı dostudur.
- Hızlı prototip oluşturma ve önceden tanımlı analiz kategorileri ile doğrudan çözümler sunar.
- Hatalı girişlerin engellenmesi için güçlü mekanizmalar içerir.

- **Python Avantajları:**

- Geniş kütüphane desteği ve esnek yapısıyla daha kapsamlı ve karmaşık işlemleri destekler.
- Makine öğrenmesi ve veri görselleştirme gibi modern veri analizi araçlarıyla entegre çalışabilir.
- Çok yönlü kullanım imkanıyla her sektöre uygulanabilir.

- **Genel Kıyaslama:**

- **Biolang**, dar bir uzmanlık alanı için optimize edilmiş hızlı çözümler sunarken, **Python** her alanda esneklik ve geniş bir ekosistem avantajı sağlar. Biolang'ın sade ve basit yapısı yeni başlayanlar için avantajdır; Python ise ileri seviye ihtiyaçlara cevap verir.

DEFNE ÇİFÇİ

olasi_kan_grubu fonksiyonu

- **Avantajlar:**

1. Kullanıcı Dostu: Bilgileri interaktif olarak alır ve kolay kullanım sağlar.
2. Doğru Hesaplama: Kan grubu ve Rh faktörlerini doğru şekilde tahmin eder.
3. Risk Farkındalığı: Kan uyuşmazlığı riskini belirterek kullanıcıyı bilgilendirir.
4. Eğitici: Genetik kombinasyonları anlamaya yardımcı olur.
5. Geliştirilebilir: Ek analizler veya özellikler kolayca eklenebilir.

- **Dezavantajlar:**

1. Sınırlı Kapsam: Klinik detaylar (antikor seviyesi, tedavi seçenekleri) dahil değildir.
2. Kullanıcı Hatalarına Duyarlı: Yanlış girişler sonuçları etkileyebilir.
3. Derin Genetik Analiz Eksik: Sadece temel genetik ihtimalleri değerlendirir.
4. Özel Durumları Kapsamaz: Nadir kan grupları veya mutasyonları analiz edemez.

dna_to_mrna fonksiyonu

- **Avantajları**

1. Basit ve Kullanışlı: DNA'yı mRNA'ya dönüştürmede kolaylık sağlar.
2. Hata Yönetimi: Geçersiz girişlerde anlamlı hata mesajları verir.

3. Modülerlik: Başka projelerde yeniden kullanılabilir.
4. Geliştirmeye Açık: Ek biyolojik doğrulamalar veya protein sentezi özellikleri kolayca eklenebilir.

- **Dezavantajları**

1. Biyolojik Doğrulama Yok: DNA'nın biyolojik geçerliliğini kontrol etmez.
2. Performans: Çok uzun DNA dizileri için optimize edilmemiştir.
3. Temel İşlevsellik: Sadece mRNA üretir, protein sentezi yapmaz.
4. Kapsamlı Değil: Sarmal yapı veya kodon özelliklerini değerlendirmez.

load_data_from_sqlite fonksiyonu

- **Avantajlar**

1. Basit ve okunabilir: Kod, SQLite veritabanından veri çekme işlemini kolay ve anlaşılır bir şekilde gerçekleştiriyor.
2. Hata yönetimi: try-except bloğu ile hata durumlarında kullanıcıya bilgi veriyor ve programın çökmesini engelliyor.
3. Esneklik: query parametresi sayesinde istenilen SQL sorgusu çalıştırılabilir.
4. Pandas entegrasyonu: pd.read_sql_query kullanılarak veri, direkt olarak pandas DataFrame'e dönüştürülüyor ve analiz için hazır hale geliyor.

.

1. Bağlantının açık kalması riski: Hata oluşması durumunda bağlantının kapatılmaması (örn. `conn.close()` öncesinde bir hata olursa) potansiyel kaynak sızıntısına yol açabilir.

- Çözüm: `with` bloğu kullanılarak bağlantının otomatik olarak kapatılması sağlanabilir.

2. Sınırlı hata ayıklama: Hata mesajları genel düzeyde ve detaylı bilgi sağlamıyor. Daha fazla loglama yapılabilir.

3. Tek sorgu işleme: Aynı anda birden fazla sorgu çalıştırılmak istenirse bu fonksiyon uygun değil.

4. Dış bağımlılık: Pandas kütüphanesine bağımlı olması, bu fonksiyonun çalıştırılması için ekstra kütüphane kurulumunu gerektirir.

RANA SABİHA ÇEVİK

Calcule_cancer_risk:

Bu fonksiyon, meme kanseri riskini tahmin etmek için tasarlanmıştır. İki ana faktöre dayanarak çalışır: genetik durum (BRCA1 veya BRCA2 mutasyonlarının varlığı) ve aile geçmişi (birinci derece akrabalarda kanser öyküsü olup olmadığı).

1. Genetik Durumun Etkisi:

- BRCA1 mutasyonu varsa, temel riske %45 eklenir.
- BRCA2 mutasyonu varsa, temel riske %40 eklenir.

2. Aile Geçmişinin Etkisi:

- Ailede birinci derece akraba öyküsü varsa, riske ek olarak %15 daha eklenir.

Başlangıçtaki temel risk, genel popülasyondaki ortalama meme kanseri riski olan %12'dir. Fonksiyon, bu temel riskin üzerine yukarıdaki faktörlere göre risk eklemeleri yapar ve hesaplanan değeri %100'ü geçmeyecek şekilde sınırlandırır.

Bu sayede, kullanıcı genetik durumu ve aile geçmişi bilgilerinden yola çıkarak kişisel kanser riskini yaklaşık bir oranla öğrenebilir.

Search_pattern:

Bu fonksiyon, bir DNA dizisinde belirli bir genetik motifin (örneğin bir mutasyon ya da gen dizisi) kaç kez geçtiğini ve geçtiği pozisyonları tespit etmek için tasarlanmıştır.

Fonksiyonun Ana Özellikleri:

1. Girdi Kontrolü:

- Kullanıcıdan bir DNA dizisi ve aranacak bir genetik desen (örüntü) alır.

- Eğer DNA dizisi veya aranan desen boşsa, kullanıcıya anlamlı bir hata mesajı döndürür.

2. Harf Duyarlılığı Ayarı:

- Varsayılan olarak büyük/küçük harfe duyarlıdır.
- Ancak case_sensitive parametresi False olarak ayarlandığında, DNA dizisini ve aranan deseni büyük harfe çevirerek arama yapar.

3. Arama İşlemi:

- DNA dizisi içinde, verilen genetik desenin kaç kez geçtiğini bulur.
- Ayrıca, aranan desenin her bir başlangıç pozisyonunu liste şeklinde kaydeder.

4. Sonuç Raporlama:

- Eğer desen DNA dizisinde bulunamazsa, kullanıcıya "Dizi bulunamadı" mesajı verir.
- Eğer bulunursa, tekrar sayısını ve tüm başlangıç pozisyonlarını içeren bir mesaj döndürür.

3.Load_data_from_excel

Bu fonksiyon, verilen bir Excel dosyasındaki verileri yükler. Şu işlevleri yerine getirir:

1. **Dosya Kontrolü:** Dosyanın varlığını ve doğru formatta olup olmadığını kontrol eder.
2. **Veri Yükleme:** Belirtilen sayfadan (veya varsayılan olarak ilk sayfadan) veriyi yükler.
3. **Hata Yönetimi:** Dosya veya sayfa hataları durumunda uygun mesajlar gösterir.
4. **Bilgi Çıktıları:** Veriyi başarıyla yükledikten sonra, veri hakkında özet bilgiler (satır, sütun sayısı, ilk birkaç satır) sunar.

5. Sonuç: Veriyi döndürür, hata durumunda None döner.

Bu fonksiyon, Excel dosyasındaki sağlık verilerini kolayca yüklemek ve ön izlemek için kullanılabilir.

Avantaj-Dezavantaj-Etik durumları

Avantajlar:

- 1. Bilimsel Yaklaşım ve Esneklik:** Programlama dili genetik veriler, biyolojik analizler ve sağlık verileri üzerinde bilimsel bir yaklaşım sergiler. Modüler yapıları sayesinde çeşitli parametreler ve faktörler eklenebilir.
- 2. Kullanıcı Dostu ve Anlaşılır:** Karmaşık analizleri basitleştirerek kullanıcı dostu sonuçlar sunar. İleri düzey tıbbi bilgi gerektirmez, böylece geniş bir kullanıcı kitlesi tarafından erişilebilir.
- 3. Hata Yönetimi ve Güvenlik:** Giriş hatalarını ve veri eksikliklerini ele alarak doğru analizler sağlar. Ayrıca, kullanıcıların yanlış veri girmesini engellemek için hata kontrolleri içerir.
- 4. Önleyici ve Psikolojik Etkiler:** Yüksek risk durumlarında kullanıcıların erken önlem almasına yardımcı olur. Ayrıca, sonuçlar genellikle belirli bir sınırdan tutulur, bu da aşırı stres ve panik yaratma riskini azaltır.

Dezavantajlar:

- 1. Sınırlı Faktörler ve Veri Kapsamı:** Genetik mutasyonlar ve sağlık verileri üzerinde odaklanırken çevresel ve yaşam tarzı faktörlerini göz ardı edebilir. Bu, gerçek riskleri tam anlamıyla yansıtmayabilir.
- 2. Performans ve Veri İşleme:** Büyük veri setleriyle çalışırken fonksiyonların performansı düşebilir. Özellikle genetik analizler ve büyük sağlık verileri için işlem süreleri uzayabilir.
- 3. Eksik Veri Kontrolü:** Verilerin doğruluğunu denetlemez. Bu durum eksik veya hatalı veri girişi yapılması halinde yanıltıcı sonuçlara yol açabilir.

Etik İlkeler:

- 1. Gizlilik ve Güvenlik:** Genetik veriler ve saėlık bilgileri son derece hassastır. Bu verilerin güvenli bir şekilde saklanması ve yalnızca yetkilendirilmiş kişilerce erişilmesi gerektiğı vurgulanmalıdır.
- 2. Şeffaflık ve Yanıltıcı Sonuçlar:** Sonuçların tahmini olduğuna dair şeffaflık sağlanmalı ve kullanıcılar biyolojik anlamlar çıkarırken dikkatli olmalıdır. Ayrıca, yazılımın tıbbi danışmanlık yerine geçmeyeceğı belirtilmelidir.
- 3. Erişim Eşitliğı ve Güvenlik:** Veri ve analizlere yalnızca yetkilendirilmiş kişilerin erişebilmesi, aynı zamanda genetik testlere erişimi olmayan kişilerin dışlanmaması önemlidir.

Bu faktörlerin dikkate alınması, fonksiyonların verimli, güvenli ve etik bir şekilde kullanılmasını sağlayacaktır.

Biolang Programlama Dilinin Fonksiyonları, Etik İlkeleri ve Diğer Dillere Göre Avantajları

- Bu dil, genetik analizleri basitleştirirken, aynı zamanda sağlık verilerini işlemek ve analiz etmek için gerekli araçları sunmaktadır. Biolang dilindeki bir başka üç temel fonksiyon—phenotype_probability, load_data_from_csv ve mrna_to_protein—sağlık bilimlerinde kullanılan biyolojik verilerin işlenmesini daha verimli hale getirmektedir. Bu yazıda, her bir fonksiyonun açıklamaları yapılacak, dilin etik ilkeleri değerlendirilecek ve Biolang'ın diğer programlama dillerine kıyasla sağladığı avantajlar ile potansiyel dezavantajlarını ele alacağım.

1. Fonksiyonlar:

a. phenotype_probability Fonksiyonu:

Genetik özelliklerin fenotipe yansıması, genetik bilimlerin temel taşlarından biridir. phenotype_probability fonksiyonu, anne ve babanın genetik kombinasyonlarına göre çocuklarının sahip olacağı fiziksel özelliklerin (fenotip) olasılıklarını hesaplamaktadır. Genetik özelliklerin dominant ve resesif allellerle taşınması prensibi, bu fonksiyon ile simüle edilerek fenotiplerin istatistiksel olasılıkları ortaya konmaktadır. Biolang dilinde yazılan bu fonksiyon, biyolojik verilerin anlaşılmasını sağlayan bir araçtır.

Fonksiyon, anne ve babanın genetik kodlarını alarak, bu kodlardan çocuklarının alabileceği olası genotipleri hesaplar. Daha sonra, her genotip için dominant veya resesif özellikler belirlenerek, her bir fenotipe ait olasılıklar hesaplanır.

Bu fonksiyon, genetik danışmanlık, araştırma ve sağlık genetiği alanlarında önemli bir araçtır. Genetik hastalıkların taşıyıcılarını ve genetik çeşitliliği anlamada kullanıcılara yardımcı olur.

b. load_data_from_csv Fonksiyonu:

Biyoteknolojik ve genetik çalışmaların çoğu büyük veri setleri gerektirir. load_data_from_csv fonksiyonu, kullanıcıların genetik ve fenotip verilerini CSV formatında yüklemelerini sağlar. Bu fonksiyon, veri analizi yapmak isteyen biyoteknologlar ve genetik araştırmacılar için oldukça kullanışlıdır. Sağlık verilerinin doğru bir şekilde yüklenmesi, verilerin güvenli bir biçimde işlenmesine olanak tanır.

Bu fonksiyon, genetik araştırmalar için veri setlerinin yönetilmesini kolaylaştırır. Veri analizi süreçlerinin hızlandırılmasını sağlar ve kullanıcıların sağlıkla ilgili verileri verimli bir şekilde işlemelerine olanak tanır.

c. mrna_to_protein Fonksiyonu:

Genetik mühendislik ve biyoteknoloji alanlarında mRNA dizilerinin amino asit zincirine dönüştürülmesi kritik bir adımdır. mrna_to_protein fonksiyonu, mRNA dizisinin amino asit zincirine dönüştürülmesini sağlar ve bu süreçte protein üretimi hakkında bilgi verir. Fonksiyon, mRNA dizisinin üçlü kodonlara bölünerek karşılık gelen amino asitleri belirler ve proteinin doğru şekilde oluşmasını sağlar.

Bu fonksiyon, biyoteknolojik araştırmalarda genetik mühendislik uygulamalarında ve hastalıkların moleküler düzeyde incelenmesinde kritik bir role sahiptir.

2. Etik İlkeler:

Biolang, sağlık ve genetik verileri işlerken etik ilkelere son derece önem verir. Dilin tasarımı, kullanıcıların verilerini gizli ve güvenli bir şekilde işleme prensibi esas alınmıştır. Genetik verilerin doğru ve etik bir şekilde analiz edilmesi, biyolojik araştırmaların güvenilirliğini artırır ve sağlıkla ilgili doğru sonuçlar elde edilmesini sağlar. Ayrıca, Biolang, kullanıcıların genetik verilerini yalnızca izniyle işler ve anonimleştirilmiş verilerin paylaşılmasını teşvik eder.

3. Biolang'ın Avantajları ve Dezavantajları:

Avantajları:

- **Sağlık ve Genetik Alanlarında Özel Olarak Tasarlanmış:** Biolang, genetik veri analizleri için optimize edilmiş fonksiyonlara sahiptir. Bu, sağlık bilimleri, biyoteknoloji ve genetik mühendislik alanlarında çalışmak isteyen kullanıcılar için büyük bir avantaj sağlar.
- **Veri Yükleme ve Analiz Kolaylığı:** CSV formatındaki veri setlerinin kolayca yüklenebilmesi, genetik verilerin hızlı bir şekilde işlenmesini sağlar.
- **Kapsamlı ve Kullanıcı Dostu:** Biolang, kullanıcıların karmaşık biyolojik ve genetik analizleri kolayca yapabilmesini sağlar. Fonksiyonlar, anlaşılabilir bir dil yapısına sahiptir ve veri analizi süreçlerini hızlandırır.

SUDE BAŞALAN

Yeni Programlama Dili Projesi: Fonksiyonlar, Etik İlkeler ve Karşılaştırma

Bu proje kapsamında, biyoloji temelli bir programlama dili tasarlıyoruz. Dilin temel amacı, biyolojik süreçleri analiz etmek, genetik verileri işlemek ve bilim insanlarına hızlı, kullanışlı çözümler sunmaktır. Benim projedeki görevim, `find_mutations`, `hastalık_sınıflandırma` ve `save_data_to_excel` fonksiyonlarını tasarlamak ve bu fonksiyonların dilimizin özelliklerine uygunluğunu açıklamaktır.

1. Fonksiyonların Açıklamaları

- **Find_mutations (mutasyon bulma):**
Bu fonksiyon, referans DNA dizisi ile bir hastanın DNA dizisini karşılaştırarak mutasyonları tespit eder. Mutasyonların pozisyonları, referans nükleotit ve değişen nükleotit bilgileri kullanıcıya sunulur. Bu işlem, genetik analizlerde kritik öneme sahiptir; özellikle genetik hastalıkların tespitinde veya ilaç geliştirme süreçlerinde kullanılır. Fonksiyon hem veri çekilerek (örneğin bir genetik veri tabanından alınan bilgilerle) hem de kullanıcı girdisiyle çalışabilir. Bu esneklik, dilimizin pratikliğini artırır.
- **Disease_cat (hastalık riski sınıflandırma):**
Bu fonksiyon, bireylerin genetik verilerine dayanarak, belirli mutasyonların hastalık risklerini hesaplar. Örneğin, BRCA1 mutasyonunun meme kanseri riskini artırdığı bilimsel olarak biliniyor. Bu tür analizlerle bireyler, olası sağlık risklerini önceden öğrenebilir. Veri hem kullanıcının girdiği bilgilerle hem de API aracılığıyla genetik veri sağlayan kaynaklardan alınabilir.
- **save_data_to_excel:**
Biyolojik analizlerden elde edilen sonuçların kaydedilmesi, raporlama süreçlerinde önemlidir. Bu fonksiyon, elde edilen verileri Excel formatında dışa aktarır. Böylece araştırmacılar, verileri kolayca paylaşabilir veya arşivleyebilir.

2. Etik İlkeler

Yeni bir programlama dili geliştirirken etik değerler büyük önem taşır.

- **Veri Gizliliği:** Genetik bilgiler son derece kişisel ve hassastır. Dilimizin veri işleme ve saklama yöntemleri, kişisel verilerin korunmasını garanti etmelidir.
- **Adil Erişim:** Bu dil, herkesin kullanabileceği şekilde erişilebilir olmalı; yalnızca büyük kuruluşlar değil, bağımsız araştırmacılar da yararlanabilmelidir.
- **Yanıltıcı Sonuçlardan Kaçınma:** Dilin algoritmaları, yanlış veya eksik bilgiye dayalı sonuç üretmemelidir. Kullanıcıya, analiz sonuçlarının kesinliği ve doğruluk oranı açıkça belirtilmelidir.

3. Dilin Avantaj ve Dezavantajları

Avantajlar:

- **Kullanıcı Dostu:** Biyoloji odaklı bir dil olduğu için biyologların kolayca kullanabileceği bir yapı sunuyor.
- **Veri İşleme Gücü:** Büyük veri setleri üzerinde çalışmak için optimize edilmiş araçlar içeriyor.
- **Özelleştirilebilirlik:** Hem manuel girişlerle hem de API kullanımıyla veri işleyebilir.

Dezavantajlar:

- **Yeni Kullanıcılar için Öğrenme Eğrisi:** Diğer dillerden farklı syntax yapısı nedeniyle kullanıcıların dili öğrenmesi zaman alabilir.
- **Biyolojiye Özel Alan:** Genel amaçlı bir dil olmaması, kullanım alanını sınırlandırabilir.
- **Kütüphane Eksikliği:** Başlangıç aşamasında mevcut diller kadar geniş bir kütüphane desteği olmayabilir.

- **Yeni Bir Dil Olarak Gelişim Aşamasında:** Biolang, henüz gelişim aşamasında bir dil olduğu için, daha geniş kütüphanelere ve fonksiyonlara ihtiyaç duyulabilir. Sağlık bilimlerinde daha derinlemesine analizler için henüz yeterli kaynak bulunmamaktadır.
- **Büyük Veri İşleme Kapasitesi Sınırlı:** Biolang, büyük ölçekli genetik veri setlerini işlerken bazı performans zorlukları yaşayabilir. Veri işleme hızını artırmak için ek geliştirmelere ihtiyaç duyulabilir.

BERFİN ZÜMRA KARACAKAYA

Fonksiyon Özellikleri:

1.Diyabet Risk Hesaplama

Fonksiyon Özellikleri

- **Amaç:**
Genetik, yaşam tarzı ve biyolojik faktörlere göre diyabet riskini yüzdesel olarak hesaplar.
- **Girdiler:**
 - Genetik risk faktörleri: Risk çarpanlarını içeren bir liste.
 - Yaşam tarzı faktörü: Stres, uyku düzeni gibi faktörler.
 - Yaş, BMI, Aktivite düzeyi, Diyet kalitesi.
- **Çıktı:**
Maksimum %100 ile sınırlı normalize edilmiş bir risk yüzdesi.

İşleyiş

1. Başlangıç risk değeri:
Fonksiyon 1.0 ile başlar ve kullanıcı girdilerine göre güncellenir.
2. Genetik ve yaşam tarzı etkileri:
Bu faktörler risk değerine çarpan olarak eklenir.
3. Yaşın etkisi:
Yaş 45 ve 60 olduğunda risk artar (sırasıyla 1.2 ve 1.5 kat).
4. BMI:
Sağlıksız BMI (18.5'ten düşük veya 25'ten yüksek) risk çarpanını artırır.
5. Aktivite ve diyet:
Fiziksel aktivite ve diyet kalitesi risk değerini azaltır.

Avantajları

- Kapsamlı Hesaplama:
Genetik, biyolojik ve yaşam tarzı faktörlerini dikkate alır.
- Esneklik:
Kullanıcı verilerine göre uyarlanabilir.
- Normalize Sonuç:
Sonuçlar %100 ile sınırlıdır, kolayca anlaşılır.

Dezavantajları

- Veri Doğruluğu:
Kullanıcıdan alınan yanlış veriler yanıltıcı olabilir.
- BMI'nin Sınırlamaları:
Kas ve yağ oranı dikkate alınmaz, bu da sonucu etkileyebilir.
- Eksik Faktörler:
Sigara, alkol gibi önemli faktörler modele dahil edilmemiştir.

Sonuç: Bu fonksiyon, kişisel sağlık değerlendirmesi yaparak diyabet riskini tahmin eder. Ancak daha güvenilir sonuçlar için ek faktörlerin dahil edilmesi ve doğru veri girişi gereklidir.

2.Kodon-Aminoasit Eşleştirme

Fonksiyon Özellikleri

- **Amaç:**
Bu fonksiyon, DNA/RNA kodonları ile karşılık gelen amino asitleri eşleştirir ve bir sözlük olarak döndürür.

- **Girdiler:**
 - Fonksiyon herhangi bir giriş almaz. Kodon gruplarının sabit bir kümesi fonksiyona dahil edilmiştir.
- **Çıktı:**
 - **codon_to_amino**: Kodonları amino asitlerle eşleştiren bir sözlük. Örneğin, "AUG" kodonu "Methionine" (Metiyonin) amino asidine karşılık gelir.

İşleyiş

1. **Kodon Gruplarının Tanımlanması:**

Kodon grupları, her bir amino asit için bir liste içerir. Başlangıç kodonu ("AUG") ve durdurma kodonları ("UAA", "UAG", "UGA") da dahil edilmiştir.
2. **Kodondan Amino Asite Dönüşüm:**

Her bir kodon, ilgili amino asit ile eşleştirilir. Bu eşleştirmeler bir sözlük (dictionary) olarak saklanır.
3. **Fonksiyonun Çıktısı:**

Fonksiyon, tüm eşleştirmeleri içeren **codon_to_amino** sözlüğünü döndürür.

Avantajları

- **Hızlı Eşleştirme:**

Kodonlar ve amino asitler arasındaki eşleştirme hızlı bir şekilde yapılır, çünkü tüm eşleştirmeler önceden tanımlıdır.
- **Kapsamlı ve Düzenli:**

Tüm temel amino asitler ve durdurma kodonları için eşleştirmeler içerir.
- **Genişletilebilir:**

Eğer yeni kodonlar veya yeni amino asitler eklenmesi gerekirse, sadece **codon_groups** içerisine yeni değerler eklenebilir.

Dezavantajları

- **Esneklik Sınırlamaları:**

Fonksiyon sadece sabit bir kodon grubunu içerdiğinden, yeni genetik kodlar veya nadir amino asitler için esneklik sağlanamaz.
- **Daha Karmaşık Veriler İçin Yetersiz:**

Eğer kullanıcı özel bir biyolojik veritabanı kullanmak istiyorsa veya belirli türler arasında farklılıkları hesaba katmak gerekirse, fonksiyon yetersiz kalabilir.
- **Statik Veri:**

codon_groups sabit ve değiştirilemez, bu da fonksiyonu daha dinamik verilerle çalışmaya uygun hale getirmeyi zorlaştırır.

Sonuç: **codon_to_amino_acid_map** fonksiyonu, kodonların doğru bir şekilde amino asitlere eşleştirilmesini sağlar. Ancak, fonksiyonun sınırlı esneklik ve genişletilebilirlik özellikleri bulunmakta. Gelişmiş genetik analizler için daha dinamik ve detaylı eşleştirmeler yapılması gerekebilir.

3. API Veri Fonksiyonu

Fonksiyon Özellikleri

Amaç:

API'den JSON formatında veri alır, bu veriyi bir DataFrame'e dönüştürür ve DNA dizisini mRNA dizisine çevirir.

- **Girdiler:**
 - **api_url**: Veri çekilecek API'nin URL adresi.
- **Çıktı:**

- o DataFrame: API'den alınan veriyi içeren bir Pandas DataFrame.
- o mRNA dizisi: DNA dizisinin mRNA'ya dönüştürülmüş hali.

İşleyiş

1. API'den Veri Alma:

Fonksiyon, belirtilen api_url'den veri almak için HTTP GET isteği gönderir.

2. Veri Doğrulama:

- o Eğer API'den gelen cevap başarılıysa (HTTP durum kodu 200), veriler JSON formatında alınır ve bir DataFrame'e dönüştürülür.
- o Eğer API isteği başarısız olursa, hata mesajı yazdırılır.

3. DNA'dan mRNA'ya Dönüşüm:

- o API'den alınan her bir genetik diziyi (sequence) alır ve DNA to mRNA dönüşümü yapılır.
- o DNA dizisinde bulunan "T" harfleri "U" harfleriyle değiştirilir.

4. Sonuç:

- o Dönüştürülmüş mRNA dizisi kullanıcıya yazdırılır.

Avantajları

- Veri Entegrasyonu:
API'den doğrudan veri alıp işleme yapabilme özelliği, farklı veri kaynaklarıyla kolayca çalışabilmeyi sağlar.
- Otomatik Veri İşleme:
Alınan genetik veriler doğrudan işlenir ve sonuç mRNA dizisi olarak elde edilir, bu da otomatik veri işleme sürecini hızlandırır.
- Hata Yönetimi:
API'den veri alınırken oluşabilecek hatalar için sağlam bir hata yönetimi mekanizması içerir. Bu, fonksiyonun güvenilirliğini artırır.

Dezavantajları

- API Bağımlılığı:
Fonksiyon, dış bir API'ye bağlı olduğundan, API'nin çalışmaması veya erişilememesi durumunda işlevselliği bozulur.
- Veri Türü ve Yapılandırma:
API'den alınan verinin yapısı beklenmedik olursa, bu veri işlenmeden önce manuel müdahale gerekebilir.
- Hız Sorunları:
API'ye yapılan her istek ağ hızına bağlıdır, bu da veri çekme sürecini yavaşlatabilir.

KAYNAKÇA

<https://biyoinformatik.net/python-kutuphaneleri>

[https://www.youtube.com/live/IFw_-0iE4pA?
si=I9EPHLMYLy07xcJp](https://www.youtube.com/live/IFw_-0iE4pA?si=I9EPHLMYLy07xcJp)

[https://academic.oup.com/biometmethods/pages/cfp-artificial-intelligence?
utm_source=google&utm_medium=cpc&utm_campaign=
86942c0wb-8696edgb3-Tl-Gb-Fb-Cb-
Ai%20%20541230%20%20Srch%20BIOMAP%20CFP%20AI
%20GLB%20EN%20WebV%201124&utm_id=2193315709
1&gad_source=1&gbraid=0AAAAApCzloeGwrqIPIMTey12
kShZf0eva&gclid=CjwKCAiA9bq6BhAKEiwAH6bqoMKKAY1
sBv1KWv2eSlAJWI1r8yH90uMXzQUEsyqla1b3BcTR8A60Q
xoCSkoQAvD_BwE](https://academic.oup.com/biometmethods/pages/cfp-artificial-intelligence?utm_source=google&utm_medium=cpc&utm_campaign=86942c0wb-8696edgb3-Tl-Gb-Fb-Cb-Ai%20%20541230%20%20Srch%20BIOMAP%20CFP%20AI%20GLB%20EN%20WebV%201124&utm_id=21933157091&gad_source=1&gbraid=0AAAAApCzloeGwrqIPIMTey12kShZf0eva&gclid=CjwKCAiA9bq6BhAKEiwAH6bqoMKKAY1sBv1KWv2eSlAJWI1r8yH90uMXzQUEsyqla1b3BcTR8A60QxoCSkoQAvD_BwE)