T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİYOBİLİŞİMDE ÖRÜNTÜ TANIMA YÖNTEMLERİNİN İNCELENMESİ VE ÖRNEK UYGULAMA

YÜKSEK LİSANS TEZİ

Bilgisayar Müh. Sedat ALAN

Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ

Tez Danışmanı : Yrd. Doç. Dr. Nilüfer YURTAY

Şubat 2009

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİYOBİLİŞİMDE ÖRÜNTÜ TANIMA YÖNTEMLERİNİN İNCELENMESİ VE ÖRNEK BİR UYGULAMA

YÜKSEK LİSANS TEZİ

Bilgisayar Müh. Sedat ALAN

Enstitü Anabilim Dalı	:	**Bilgisayar ve Bilişim Mühendisliği**
Enstitü Bilim Dalı	:	**Bilgisayar ve Bilişim Mühendisliği**

Bu tez 11/02/2009 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.

.....	.....	.....
Jüri Başkanı	**Üye**	**Üye**
Prof. Dr. Emin GÜNDOĞAR	Yrd. Doç. Dr. Kürşat AYAN	Yrd. Doç. Dr. Nilüfer YURKAN

ÖNSÖZ

Biyoloji bilimindeki gelişmeler sonucunda hücre ve DNA yapısının çözülmesiyle ortaya çıkan veriler, bunların depolanması ve analiz edilebilmesi için bilgisayar bilimlerine gereksinim duyulması yeni bir bilim dalı olan biyobilişimi (biyoinformatik) ortaya çıkarmıştır. Önümüzdeki yıllarda özellikle insan DNA'ları üzerinde yapılan araştırmalardan ortaya çıkan ve depolanan veriler üzerinde yapılacak analizlerle, hastalıklara karşı yeni ilaçlar geliştirilmesi, kalıtsal hastalıkların yok edilmeleri konularında büyük aşama kaydedilecektir. Bitki ve hayvan DNA'ları üzerlerinde yapılacak araştırmalar ile daha fazla verim alınabilirliği ile ilgili ciddi adımlar atılacağı tahmin edilmektedir.

Bu tez çalışmasında bana her türlü desteği sağlayan aileme ve tez çalışması boyunca tüm yardımlarından dolayı tez danışmanım sayın Yrd. Doç. Dr. Nilüfer Yurtay'a teşekkürleri bir borç bilirim.

İÇİNDEKİLER

ŞEKİLLER LİSTESİ

ÖZET

Anahtar Kelimeler : Biyobilişim, Örüntü Tanıma, Moleküler biyoloji, Örüntü Tanıma Algoritmaları, DNA

Bu tez çalışmasında moleküler biyoloji ve bilgisayar bilimlerinin birleşmesi ile ortaya çıkan biyobilişim, örüntü tanıma ve moleküler biyoloji alanlarında bilgiler verilmiştir.

Biyobilişimde örüntü tanıma algoritmaları anlatılmış, algoritmaların kullanım alanlarıyla ilgili bilgiler verilmiş, örneklerle açıklanmaya çalışılmıştır.

Sonuç bölümünde biyobilişim ve geleceği ile bazı öngörülerde bulunulmuş ve yapılan araştırmalar sonucunda ortaya çıkacak yeni bilgiler ışığında insanlığa faydalarından bahsedilmiştir.

PATTERN RECOGNITION METHODS IN BIOINFORMATICS AND SAMPLE APPLICATION

SUMMARY

Keywords : bioenformatics, pattern recognition, molecular biology, pattern recognition algorityms, dna

This thesis contains information about pattern recognition, molecular biology and bioenformatics which is a combination of molecular biology and computer science.

Pattern recognition algorityms of bioenformatics are explained. The areas that they are used are described with examples.

In the conlcusion part some predictions are made about the future of bioenformatics and the benefits that will occur as a result of new information which will be achieved by ongoing research projects.

BÖLÜM 1. GİRİŞ

DNA ikili sarmal yapısının 1953 yılında James Watson ve Francis Crick tarafından ortaya çıkarılmasından bu yana moleküler biyolojide çok büyük çapta gelişmeler yaşanmıştır [8]. Bu süreçte bilgisayar bilimleri alanında da büyük ilerlemeler ve gelişmeler yaşanmış ve bu iki bilim dalının kesiştiği alanda biyobilişim adıyla yeni bir bilim dalı ortaya çıkmıştır.

Bu yeni ortaya çıkan bilimin ilgilendiği en önemli noktalar DNA, RNA ve protein sıralanmasının incelenmesi ve bu sıralamalardaki örüntülerin ortaya çıkarılmasıdır. DNA bir canlının yapı taşıdır ve taşıdığı bilgilerin ortaya çıkarılması yeni ilaçların bulunmasından doğadaki canlı türleri arasındaki benzerliklerin keşfine kadar birçok konuda yararlı bilgiler sunabileceği için büyük bir öneme sahiptir.

İlk kısımda örüntü tanıma açıklaması yapılacak bunun yanında bazı biyolojik terimler ve açıklamalarından bahsedilecektir. DNA ve RNA'nın yapısı ve üstlendiği bilgiler anlatılacaktır. Daha sonra biyolojik sıralamalar üzerinde örüntü tanıma için kullanılan bazı algoritmalardan bahsedilecek ve algoritmalar açıklanmaya çalışılacaktır.

BÖLÜM 2. ÖRÜNTÜ TANIMA

2.1. Örüntü Tanıma Nedir

Örüntü, ilgilenilen varlıklar ile ilgili gözlenebilir veya ölçülebilir bilgilere verilen addır. Gerçek dünyadaki bu örüntüler, genellikle ilgilenilen verilerin nicel tanımlama şekilleridir. Örüntü tanıma, insanların çeşitli ses, görüntü ve benzeri tüm örüntülerin biçimsel şekillerinden çıkardıkları dilsel şekillendirmedir. Aslında, örüntü tanıma bilimin, mühendisliğin ve günlük hayatın geniş bir alanındaki etkinlikleri kapsamaktadır. Örüntü tanıma uygulamalarını insanların yaşantısında da görebiliriz: hava değişimin algılanması, binlerce çiçek, bitki, hayvan türünü tanımlama, kitap okuma, yüz ve ses tanıma gibi bulanık sınırlara sahip birçok etkinlikte örüntü tanıma kullanılır. İnsan örüntü tanıması, geçmiş tecrübelere dayalı öğrenme esaslıdır. Böylece, insanlar pratikte karşılaştığı örüntü tanıma olaylarını tecrübeleri ışığında değerlendirebilme yeteneğine sahiptirler. Belirli bir sesi tanımak için kullanılan kuralları tanımlamak mümkün değildir. İnsanlar bu işlemlerin birçoğunu oldukça iyi yapmalarına rağmen, bu işlemleri daha ucuz, iyi, hızlı ve otomatik olarak makinelerin yapmasını arzularlar. Örüntü tanıma, böyle akıllı ve öğrenebilen makineleri gerçekleştirmek için, çok boyutlu bir mühendislik disiplinidir. Örüntü tanıma olayını şu şekilde irdeleyebiliriz: Aralarında ortak özellik bulunan ve aralarında bir ilişki kurulabilen karmaşık işaret örneklerini veya nesneleri bazı tespit edilmiş özellikler veya karakterler vasıtası ile tanımlama veya sınıflandırmadır. Bu bağlamda, örüntü tanımanın en önemli amaçları; bilinmeyen örüntü sınıflarına belirli bir şekil vermek ve bilinen bir sınıfa ait olan örüntüyü teşhis etmektir. Örüntü tanıma tekniklerinin uygulamaları birçok mühendislik, tıp, askeri ve bilim alanına açıktır. Bunlardan bazıları; ses tanıma, EEG sınıflama, DTMF haberleşme işaretlerini tanıma ve radar hedef sınıflama, biyomedikal kontrol, veri madenciliği verilebilir. Örüntü tanıma olarak bilinen bu uygulamalar, makine öğrenmesi, örüntü sınıflandırma,

ayrım analizi ve nitelik tahmini gibi isimlerle de anılmaktadır. Örüntü tanıma kavramı, Şekil 2.1. de gösterildiği gibi üç önemli birimden oluşmaktadır:

2.1.1. İşaret / görüntü işleme

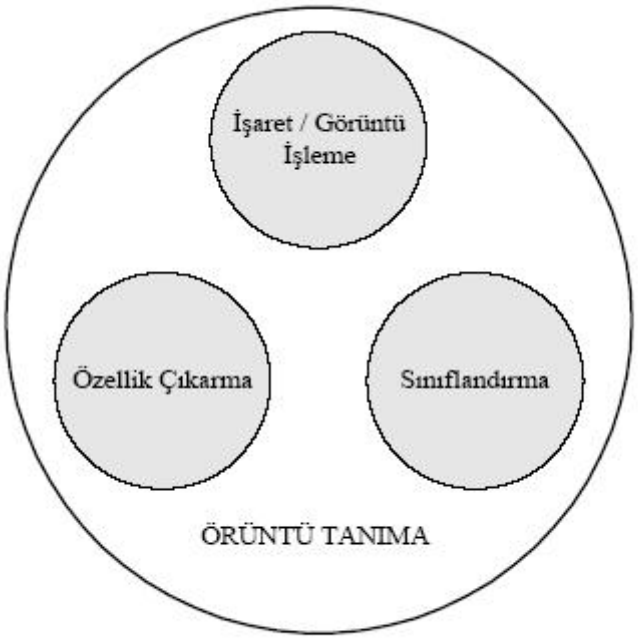
Ön işlem aşamasıdır. İşaret veya görüntünün filtre edildiği, çeşitli dönüşüm ve gösterim teknikleri ile işlendiği, bileşenlerine ayrıldığı veya
modellendiği kısımdır.

2.1.2. Özellik çıkarma

İşaret ve görüntünün veri boyutunun indirgendiği ve tanımlayıcı anahtar özelliklerinin tespit edildiği ve aynı zamanda normalizasyona tabii tutulduğu aşamadır. Sistemin başarımında en etkili rolü oynar.

2.1.3. Sınıflandırma

Çıkarılan özellik kümesinin indirgendiği ve formüle edildiği tanımlayıcı karar aşamasıdır.



Şekil 2.1. Örüntü tanıma kavramı.

2.2 Örüntü Tanıma Sistemleri

Örüntü tanıma sistemleri gözlenen veya ölçülen verileri tanımlanmasında birçok uygulamanın merkezinde yer alır. Şekil 2.2' de yaygın olarak kullanılan genel anlamda örüntü tanıma sistemi verilmiştir. Algılayıcılar, herhangi bir anda mümkün olan birçok doğal durumlardan biri olabilen bazı fiziksel işlemleri ölçerler. Aşağıdaki blok diyagramın en önemli görevlerinden biride, elde edilen ölçümlerin hepsinden oluşan giriş uzayından daha az boyutta özellik çıkartmaktır. Sonunda, sınıflandırıcının rolü örüntüyü özelliklerine göre kategorize ederek uygun sınıflara kaydetmektir.

Şekil 2.2. Örüntü tanıma sistemi.

Mevcut örüntü tanıma sistemleri üç grupta toplanmaktadır:

2.2.1. İstatistiksel örüntü tanıma

İstatistiksel örüntü tanıma yöntemin de, sınıflama algoritmaları istatistiksel analiz üzerine kurulmuştur. Aynı sınıfa ait örüntüler, istatistiksel olarak tanımlanan benzer karakteristiklere sahiptirler. Bu yöntemde, özellik olarak nitelendirilen karakteristik ölçümler giriş örüntü örneklerinden çıkarılır. Her örüntü bir özellik vektörü ile tanımlanır. Genelde sınıflandırıcıyı oluşturan karar ve sınıflandırma yöntemleri üzerinde önemle durulur. Sınıflandırıcı tasarımı, ölçümler ve olasılıklar gibi işlenebilir örüntü bilgilerini birleştirmeyi esas alır. Böylece sınıflama, giriş veri uzayının olasılık yoğunluk fonksiyonlarının tahmini üzerine kurulu bir istatistiksel yapıdır. İstatistiksel örüntü tanıma Bayes Karar Teorisi üzerine kurulmuş olup, uzun bir geçmişe sahiptirler.

2.2.2. Yapısal örüntü tanıma

Yapısal (geometriksel, kural dizilim) örüntü tanıma yaklaşımın da, verilen bir örüntü, şekilsel yapıdan temel karakteristik tanımlanmaya indirgenir. Çoğu zaman, örüntülerden çıkarılan bilgi yalnızca özellikler kümesinin sayısal değerlerinden değildir. Özelliklerin birbirine bağlanması veya aralarındaki karşılıklı ilişki, tanımlamayı ve sınıflandırmayı kolaylaştıran önemli yapısal bilgiye sahiptir. Bir başka deyişle örüntünün işlenmemiş halinden elde edilen tanımlayıcı biçimsel sentaks veya bunların sentezinden çıkarılan gramer ile tanımlama gerçekleşir. Örneğin, örüntünün köşe sayısı, kenar açıları vb. Genel olarak yapısal yöntemde daha basit alt örüntüler karışık örüntülerin hiyerarşik tanımlamalarını formüle eder. Yapısal yöntemde her örüntü, bileşenlerinin bir kompozisyonu olarak ele alınır. Şekil 2.4’de bir yapısal örüntü tanıma sistemi görülmektedir.

Şekil 2.3 Yapısal örüntü tanıma sistemi.

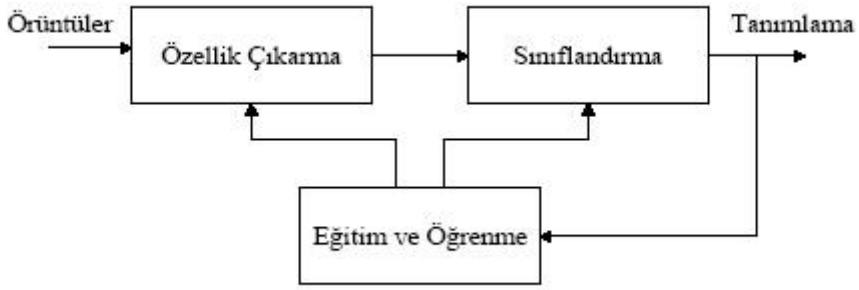
Yapısal örüntü tanıma yönteminde çeşitli birimler arasındaki ilişki çok büyük önem taşır ve gerçek tanımada kullanılan bazı şekilsel notasyonlar tarafından belirtilir. Örneğin, ekrandaki bir masayı tanıma, "köşelerinden eşit uzunlukta bacaklar tarafından desteklenen yatay bir dikdörtgen yüzey" gibi yapısal tanımlamayı temel alarak gerçekleştirilebilir. Bu yöntemde, çevre uzunluğu, alan, ağırlık merkezi, eylemsizlik momenti ve Fourier tanımlayıcıları gibi genel özellikleri kullanır. Otoregresif model, poligonsal yaklaşım ve zincir kodları yapısal örüntü tanıma yöntemine örnek olarak verilebilir.

2.2.3. Akıllı örüntü tanıma

Örüntü tanıma sistemi, daha önceden öğrendiklerini tutabilecek bir hafızaya sahip, çıkarım, genelleme ve belirli bir hata toleransı ile karar verebilme yeteneklerini içermekte ise bu sistem akıllı örüntü tanıma sistemi olarak değerlendirilir. Şekil 2.5.' de böyle akıllı ve öğrenebilen makinaları gerçekleştirmeye yönelik örüntü tanıma yaklaşımı verilmiştir.

Akıllı örüntü tanıma yaklaşımları, öğrenme tabanlı olup, karar aşamasında geçmiş tecrübelerinden sonuç üretmektedirler. Günümüzde, öğrenmeli örüntü tanıma algoritmaları yapay sinir ağ merkezli olarak gelişmektedir ve bu doğrultuda çalışmalar yoğunluktadır. YSA yaklaşımları istatistik yaklaşıma karşı belirleyici olarak ifade edilebilir. Çünkü öğrenme algoritmaları örüntü sınıflarının istatistiksel özellikleri hakkında hiçbir şey kullanmamaktadır. Bununla birlikte, istatistiksel ve YSA örüntü tanıma yaklaşımları şekil ve amaç olarak çok benzer olup, hatta YSA

'nın geleneksel istatistiksel örüntü tanımanın bir uzantısı olarak ifade edilen görüşlerde bulunmaktadır.



Şekil 2.4. Akıllı örüntü tanıma yaklaşımı.

BÖLÜM 3. BİYOBİLİŞİM

3.1. Biyobilişim Nedir

1960'larda başlayan bilgisayar uygulamalarının biyolojide kullanılması girişimi, her iki alandaki teknolojik gelişime paralel olarak hızla ilerlemiş ve böylelikle ortaya çıkan "BİYOİNFORMATİK" dalı bugün en popüler akademik ve endüstriyel sektörlerin başına geçmiştir [5].

Bilgisayarların moleküler biyolojide kullanımı üç boyutlu moleküler yapıların grafik temsili, moleküler dizilimler ve üç boyutlu moleküler yapı veritabanları oluşturulması ile başlamıştır. Kısa sürede çok yüksek miktarlarda veri üreten, endüstri düzeyinde gen ekspresyonu, protein-protein ilişkisi, biyolojik olarak aktif molekül araştırmaları, bakteri, maya, hayvan ve insan genom projeleri gibi biyolojik deneylerin doğurduğu talep sonucunda, bu alandaki bilişim uygulamaları neredeyse takip edilemez bir hızda gelişmiştir. Biyoinformatik dalının ayrı bir (disiplinlerarası) bilim dalı olarak tanınması da son 10 yılda gerçekleşmiştir.

Biyoinformatik genel olarak biyolojik problemlerin çözümünde bilişim teknolojilerinin kullanılması olarak tanımlanabilir. En dar tanımı ile; Genomik sekansları destekleyen biyolojik veritabanlarının oluşturulması ve işletilmesi, en geniş tanımı ile; Mevcut tüm bilgisayar uygulamalarının biyolojik problemlerin çözümünde kullanılması olarak anlaşılır. Biyoinformatik modern biyolojinin iki temel bilgi akışını kapsar [6].

1. Genetik bilgi akışı: Bir organizmanın DNA'sı incelenerek özelliklerinin belirlenmesinden, incelenen bu organizma türünün oluşturduğu toplulukların

karakteristik özelliklerine kadar olan bilgi akışı. Elde edilen DNA bilgisi tekrar genetik havuzun tanımlanması için kullanılır.

2. Deneysel bilgi akışı: Biyolojik olaylar gözlenerek elde edilen enformasyon, açıklayıcı modeller ile tarif edilir, daha sonra bu modellerin doğruluğu yeni deneyler ile test edilir.

Son yirmi yılda temel biyolojik araştımaların klinik tıp uygulamaları ve klinik tıp bilgi sistemleri üzerindeki etkisi daha da belirleyici olmuş ve bugün yeni kuşak epidemiyolojik, tanı, teşhiş ve tedavi amaçlı modüllerin ortaya çıkmasına yol açmıştır. Biyoinformatik çalışmalar temel bilimsel araştırmalara yönelik görünmekle beraber önümüzdeki on yıl içinde kilinik bilişim için vazgeçilmez olacaktır. Örneğin hastaların medikal formlarında giderek artan bir sıklıkla DNA dizilim bilgileri yer almaya başlayacaktır. Bugün ABD'de bazı sigorta şirketleri, risk primleri belirlenirken mevcut genetik tarama test sonuçlarını talep edebilmektedir. Biyoinformatik araştırmalar için geliştirilen algoritmaların çok yakında klinik bilişim sistemlerine entegre olması beklenmektedir.

Bu alanı kısaca tanımlamanın bir yolu da, biyoinformatik araçların kullanıldığı genel araştırma konularını özetlemek olabilir:

Metodolojik çalışmalar:

1. DNA sıra ve dizilimi araştırmaları
2. Protein sıra ve dizilimi araştırmaları
3. Makromoleküler yapıların (DNA, RNA, protein) üç boyutlu dizilim araştırmaları
4. Küçük moleküllerin(potansiyel terapötik maddeler, aktif peptidler, ribozimler, vs) ligandlarıyla etkileşiminin araştırılması
5. Heterojen biyolojik veritabanlarının entegrasyonu
6. Biyolojik enformasyonun paylaşımının kolaylaştırılması
7. Bilgisayar ile otomize edilmiş veri analizi ve iletimi
8. Etkileşimde bulunan gen ürünleri için bilgi ağları oluşturulması

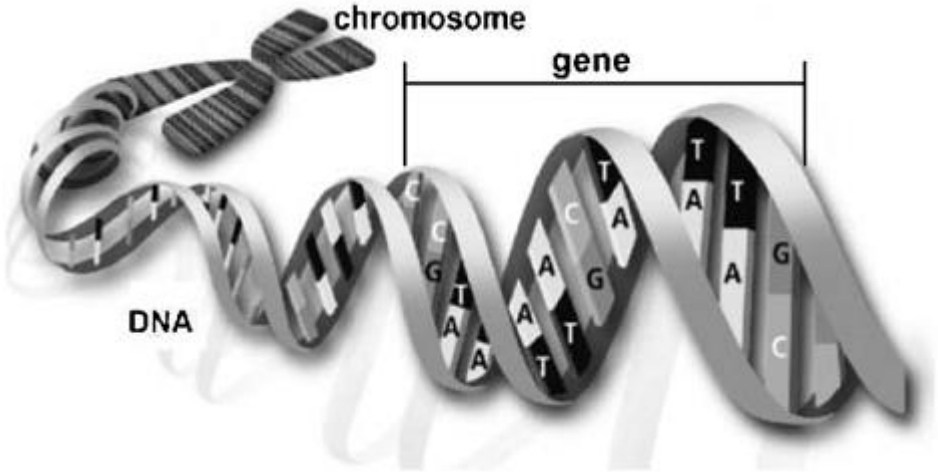
9. Kimyasal reaksiyonlardan hücrelerarası iletişime kadar pek çok biyolojik faaliyet sürecinin simülasyonu
10. Büyük çaplı biyolojik deneylerden (GENOM projeleri gibi) çıkan sonuçların analizi

Biyolojik çalışmalar

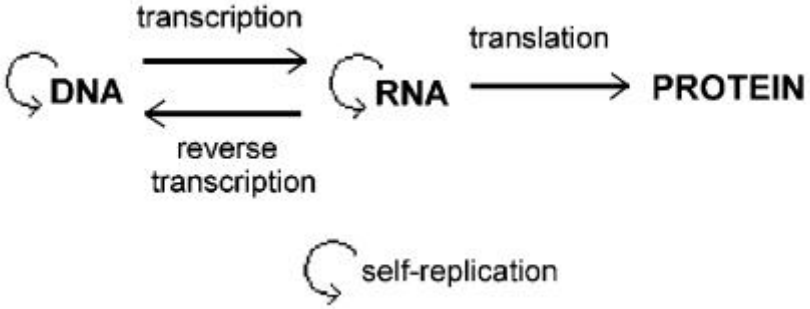
1. Protein yapı ve fonksiyonun belirlenmesi
2. Herhangi bir biyolojik fonksiyonu arttıran ya da engelleyen küçük moleküllerin tasarlanması
3. Karmaşık genetik fonksiyon ya da regülasyon faaliyetlerinin tanımlanması
4. Tıbbi ya da endüstriyel amaçlı yeni makromoleküller üretmek
5. Genetik faktörlerin, hastalık yatkınlığına etkilerini ortaya çıkarmak

3.1. Dna Nedir

DNA kalıtımın temelidir. Küçük nükleoitit denilen molekülerden oluşan bir polimerdir. Bu dört temelde ayırt edilebilir: Adenin(A), sitozin(C), guanin(G), ve timin(T). Bir DNA dizilimi böylece bu dört alfabenin dizilimlerinden oluşmuştur.(A, C, G, T) DNA genellikle iki kıyı şeklinde oluşur ve bu iki kıyı temellerinde birbirlerini tamamlayıcıdırlar. Örneğin, Hidrojen bağında A ile T'nin eşleşmesi ve G ile C'nin eşleşmesi. Çift kıyılı DNA boşlukta ünlü çift helezonu oluşturur. (Şekil 3.1) Eşleştirme mekanizması DNA'nın bir kıyısının ters tamamlayıcı kıyının üretimini üslenmesine izin verir, böylece DNA'nın nasıl çoğaldığı açıklanmaktadır.



Şekil 3.1. DNA'nın yapısı

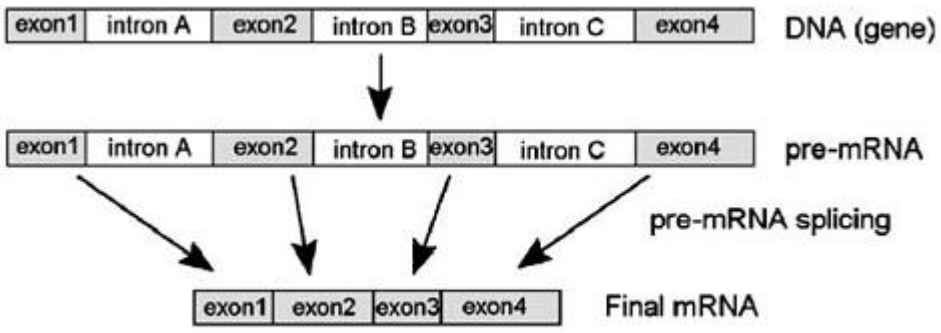


Şekil 3.2 Hücre içersinde bilgi akışı

DNA bir organizmanın işlevi için gerekli genetik bilgileri taşır. Bir hücre içindeki bilgi akışı Şekil 3.2'deki diyagramda özetlenmiştir. Şematikte DNA'dan protein sentezine olan ve transkripsiyon denilen işleme kadar olan adımı görmekteyiz. Transkripsiyon DNA'daki bilgileri RNA denilen kopyalara kopyalar. Eğer DNA dizilimindeki bir segment bir protein kodlarsa(DNA diziliminde kodlama bölgesine uygun) RNA'ya bir haberci ya da mRNA denilir. Transkripsiyonda A,C,G,T olan DNA nükleotitleri RNA nükleotidlerine uyarlanır. RNA moleküllerinde U (urasil) T'in (timin) G, C,A. yerini alır. Bilgi akışındaki son adım ise çevirim işlemidir. mRNA'da kodlanan bilgi, protein oluşturan amino-asitlerin kesin sıralamasının belirlenmesinde kullanılır. Proteinler 20 değişik amino asitten oluşan alfabenin polipeptid zincirlerdir .Genetik kod üç temelli bir koddur.Üç ardışık RNA

nükleotidin 20 amino asitten birini kodladığı ya da çeviriyi durdurduğu ardışık kodonlardan oluşur.

Transkripsiyon işlemi Prokaryot hücrelerde (örneğin: bakteri) ve ökaryot hücrelerde farklıdır. Prokaryot hücrelerde RNA polimerazı mRNA transkriptini doğrudan DNA şablonundan üretir Ökaryotlarda ise DNA dizilimindeki genler devamlı değildir fakat bunun yerine kodlanan bölgelere(protein kodu olan exonlar) ve kodlanmayan bölgelere(intronlar) ayrılmışlardır. RNA çekirdeğe kopyalanır ve sonrasında post-transkripsiyonel modifikasyona(örn: ön-mRNA eklenmesi) uğrar. Burada intronlar ayrılır ve kalan exonlar son mRNA'yı oluşturmak üzere birleşirler(Şekil 3'e bakınız)Bu sonrasında çevirim sırasındaki protein sentezi için kullanılır. Böylece DNA diziliminin içine konulanlar transkripsiyonun başlamasını yada sona erdirilmesi kontrol eden spesifik alt dizilimlerdir. Bu destekleyiciler, güçlendiriciler, susturucular, sonlandırıcılar gibi alt dizilimler gen tanımının düzenleyicileridirler. Ökaryot DNA dizilimindeki diğer ilgi çekici dizilimler ise kodlama bölgeleri(exonlar), kodlanmayan bölgeler(intronlar ve intergenik bölgeler) eklenen sinyaller ya da ekleme alanları ve stop kodonları içermeyen DNA dizileridir. Alternatif bir ekleme ile bazı exonlar son mRNA'yı sıralama yaparken çıkarılabilirler.



Şekil 3.3. Ökaryot hücresi içindeki kalıtım süreci.

3.2. Rna Nedir

RNA'lar ribonukleotitlerinbirbirlerine bağlanması ile meydana gelen tek zincirli nukleik asitlerdir. DNA molekülleri ile kıyaslandığı zaman boyları daha kısadır. Hemen hemen bütün hücrelerde bol olarak bulunmaktadırlar. DNA'nın protein üretimindeki işlevini yerine getirebilmesi için bir "ara molekül"e ihtiyaç vardır. Bu işlevi yüklenen Ribonükleik asit ,nükleotidlerin ard arda yerleşmesiyle birleşmiş tek diziden oluşan (DNA nın tek sarmal zincirinden biri gibi) yüksek kaliteli moleküldür. DNA molekülleri büyük oranda hücre çekirdeğinde bulunurken, RNA'lar hücre içine yayılmış durumdadırlar. DNA genellikle çift şeritli sarmal yapıda iken, RNA tek şerittlidir. Ancak, tek şeritli DNA ve çift şeritli RNA moleküllerine de rastlanılmaktadır.

3.2.1. Messenger rna (mrna)

DNA'da saklı bulunan genetik bilginin, protein yapısına aktarılmasında kalıplık görevi yapan aracı bir moleküldür. DNA molekülünde lokalize çözülme ile kopyası çıkarılan moleküllerdir. Haberci ribonükleik asit olarak adlandırılan m-RNA, DNA'dan özel bir polipeptit yapısına çevrilmiş kimyasal enformasyonu taşır**.** m-RNA nükleotidlerin tek sarmalından ve bir DNA boyundaki kalıptan oluşmuştur. Üzerindeki nükleotid dizisi, DNA sarmallarından bir tanesiyle eşleşir. Polipeptit molekülü, DNA'dan ayrılarak m-RNA halinde ribozomlara yapışır, Buralarda da gelen mesaja uygun proteinler üretilir. Bu şekilde birleştirilmiş RNA molekülü, tıpkı bir fotoğrafın pozitifi ve negatifi gibi kalıtım mesajının karşı tip halindeki eşidir. Bu mesaj daha sonra sitoplazmada ribozomlar sayesinde çözülebilecek ve taşıyıcı RNA sayesinde amino asit birleşimi için kullanılacaktır.

3.2.2. Ribozomal rna (r-rna)

RNA'lar ribozomların ana yapısal elementi olup yaklaşık olarak ribozom ağırlığının % 65'ini teşkil ederler. Ribozomal RNA; ribozomlar sitoplazma içine dağılmış küresel yapılardır. Proteinler ve r-RNA denen özel bir RNA çeşidinden oluşurlar. Türe göre ribozomun %40 ila %60ını bu moleküller meydana getirir. Ribozomların rolü haberci RNA da yazılı genetik kodu çözmektir. Prokaryotik hücrelerde 3 çeşit,

ökaryotik hücrelerde ise 4 çeşit rRNA bulunmaktadır. Bunlara ilave olarak ökaryotik hücrelerde iki çeşit RNA daha bulunmaktadır. Bunlardan birincisi heterojen nuklear RNA (hnRNA)'lardır. Bunlar ökaryotik hücrede sentezlenen ve prosese uğramamış öncül mRNA molekülleridir. İkincisi ise küçük nuklear (snRNA)'dır ve yine öncül mRNA moleküllerinin prosese uğraması esnasında ortaya çıkmaktadırlar.

3.2.3 Transfer rna (t-rna)

t-RNA'lar da ribonukleotidlerin polimerize olması ile meydana gelmiş, çok kıvrımlar gösteren ve tek zincirli yapıya sahip bir RNA çeşididir. Biçimi 3 yapraklı yonca yaprağı ve molekülün iki ucundan oluşan bir ‘’Sap’’ biçimidir. Zincirde yer alan ribonukleotid sayısı 70 ile 99 arasında, molekül ağırlığı ise 23.000 ile30.000 dalton arasında değişmektedir. Doğada yer alan 20 aminoasitin her biri için en az bir tRNA molekülü bulunmaktadır. tRNA'lar adaptörlük görevi yaparak bir uçlarına bağladıkları amino asiti, ribozoma tutunmuş mRNA'nın taşıdığı kodono göre polipeptid zincirine dizerler. RNA-M tarafından belirlenen özgül bir dizide, bir polipeptit molekülüne bağlanacak aminoasitleri yerleştirme işlevini yapar; uygun aminoasitleri ribozomlara taşır. Yani haberci RNA moleküllerinin taşıdığı mesajın çevirisiyle ilgilidir.

İşte RNA molekülleri 20 çeşit aminoasitin çeşitli sıra ve sayıda dizilimini oluşturarak protein dediğimiz yapıları oluşturma mekanizmasının yani protein sentezinin başrolünü oynar.

3.4. Dna Dizilim Analizi

Yüksek iş-çıkarımlı dizilim metotları ve DNA mikro dizilim teknolojisi gibi moleküler biyoloji ve gen araştırmalarındaki son gelişmeler, büyük miktarda eşine daha önce rastlanmamış bilgi ortaya çıkarmıştır. İnsan Biyolojisi Projesinin tamamlanması sonucunda, tamamlanmış insan biyolojisinden elde edilen bilgilerin tıp ve biyoloji alanlarında kullanabilinecek olması araştırma çevrelerinde büyük ilgi uyandırmıştır. Bu devasa ölçüdeki bilgilerin verimli olarak bilgisayar ortamında kullanılabilmesi ise, şu andaki en büyük sorundur. Genlerin fonksiyonunu, yapısını ve oynadıkları dominant rolleri aydınlatmadaki en önemli iki teknoloji DNA dizilim analizi ve DNA mikro-dizilim bilgi analizidir. DNA dizilim analizi 20 yıldır hatta

geniş ölçekli dizilim tekniklerinden bile önce incelenmektedir. Yinede son yıllardaki hızlı bilgisayarların ve algoritmaların gelişmesi ile momentumuna ulaşmış ve güncel çevrimiçi veritabanlarının mevcudiyeti, bu büyük ölçüdeki dizilim bilgilerini saklamaktadır. Bu veritabanları ayrıca araştırmacıların kendi araştırmalarını başka kişiler ile paylaşmasını ya da onların araştırmalarına ulaşmalarını sağlamaktadır. Bir moleküler biyolog bilinmeyen bir DNA dizilimi ile karşılaştığında ilk görevi, büyük umumi dizilim veritabanlarındaki benzer dizilimleri araştırmak olmalıdır. Bunu yaparak biyolog diğer araştırmacılar tarafından elde edilen bilgilerle, bilinmeyen dizilimin olası fonksiyonunu yada yapısını öğrenebilir, bu da onu daha spesifik ve amaçlı bir analize ya da deneyime götürebilir [6].

BÖLÜM 4. BİYOBİLİŞİMDE ÖRÜNTÜ TANIMA ALGORİTMALARI

4.1. Tekrarlanma Bulma

Pek çok genetik hastalık uzun kromozal bölgelerin kopması, ikileşmesi ve yeniden düzenlenmesi ile ilişkilidir. Bunlar büyük çapta bir genomik yapıyı etkileyen dramatik olaylardır ve milyonlarca nükleotidi kapsayabilir. Örneğin, DiGeorge sendromu 22 numaralı insan kromozomundaki 3 Mb'lık bir kopma ile ilişkilidir ve genelde bağışıklık sistemi sorunları ve kalp yetmezliği neden olur. Bu büyüklükte bir kopmada önemli genlerin kaybı olasıdır ve bu durum hastalığa yol açar.

Genomik yapıdaki bu tür dramatik değişimler çoğu kez DiGeorge sendromunda olduğu gibi kopan bölümü yandan destekleyen çok benzer bir dizi çiftini gerektirir. Bu benzer diziler DNA'da bir tekrarlanma oluşturur ve bu, genomdaki tüm tekrarlanmaları bulmak için gereklidir [1].

DNA'daki tekrarlanmalar pek çok evrimsel sır barındırır. Çoğu genomlardaki yüksek sayıda tekrarlanmalar çarpıcı ve hala açıklanamamış bir olgudur. Örneğin, tekrarlanmalar insan genomunun yaklaşık %50'sine tekabül eder. İnsan genomundaki tekrarlanmaları arayan algoritmaların 3 milyar nükleotidlik bir genomu analiz etmesi gerekir ve ikinci dereceden dizi hizalama algoritmaları bu iş için çok yavaştır. Kesin tekrarlanmaları bulmak için kullanılacak en basit yöntem, her l-mer için l-mer'in genomik DNA dizisinde bulunduğu tüm konumları tutan bir tablo oluşturmaktır. Böyle bir tablo, her birinin 0 ve M arasında bir sayı kadar konum tuttuğu 4^l adet kutu içerecektir. Burada M, genomdaki en genel l-mer'in bulunma sıklığıdır. Her kutudaki ortalama eleman sayısı $n/4^l$ 'dir, n burada genomun uzunluğunu belirtir. Pek çok uygulamada l parametresi 10 ila 13 arasında değişir, bu nedenle bu tablo kontrol edilemeyecek kadar büyük değildir. Bu listeleme yaklaşımı

l uzunluğundaki tüm tekrarlanmaların kolaylıkla bulunmasını sağlasa da, böyle kısa tekrarlanmalar DNA analizleri için yeterince ilginç değildir. Biyologlar bunun yerine uzun azami, yani sola ya da sağa genişletilemeyen tekrarlanmalara ilgi duyarlar. Önceden belirlenmiş bir L parametresinden daha uzun olan azami tekrarlanmaları bulmak için, l uzunluğundaki her kesin tekrarlanma L'den daha uzun bir tekrarlanmada gömülü mü değil mi anlamak amacıyla sola ya da sağa uzatılmalıdır. Tipik olarak l << L olduğundan genişletilecek olan tekrarlanmaların sayısı, bulunacak azami tekrarlanmalardan çok daha fazladır. Örneğin, 4.6 milyon nükleotidden oluşan koli basili bakterisi genomunun l = 12 uzunluğunda milyonlarca tekrarlanmaları vardır, ama sadece 8000 civarında L = 20 veya daha fazla uzunlukta azami tekrarlanmaları bulunur.[1] Bu nedenle, Tekrarlanma Bulma'nın bu yaklaşımındaki işin çoğu, kısa tekrarlanmaları gereksizce genişletmeye çalışarak boşa harcanır.

4.2. Komut Tablosu

Bir listeden tüm benzersiz öğeleri listeleme sorununu göz önünde bulundurmak gereklidir.

Kopya Yok etme Sorunu:
Tamsayı listesindeki tüm benzersiz girdileri bulun.

Girdi: n tamsayılarının bir a listesi
Çıktı: tüm kopyaları yok ederek a listesi

Örneğin, (8,1,5,1,0,4,5,10,1) listesinde 1 ve 5 öğeleri birçok kere tekrarlanmıştır. Böylece sonuçta ortaya çıkan liste (8,1,5,0,4,10) veya (0,1,4,5,8,10)'dur. Listedeki öğelerin sırası fark etmemektedir. Liste çoktan düzenlemiş ise kolaylıkla çaprazlanabilir ve eş ardıl öğeler kaldırılabilir. Bu yaklaşım, n boyutunda listeyi düzenlemek için
O (n log n) süre ve ardından düzenlenen versiyonu çaprazlamak için O(n) süre gerektirir [2].

Bir başka yaklaşım ise b dizisine yönelen a dizisi öğelerini kullanmaktır. Yani, 0 ve 1lerden oluşan başka bir b dizini yaratabiliriz; i a içinde ise bi =1(her seferinde) ve aksi şekilde bi =0. a= (8,1,5,1,0,4,5,10,1) için b0, b1, b4, b5, b8 ve b10 öğeleri 1'e ve b= (1,1,0,0,1,1,0,0,1,0,1)'de olduğu gibi b'nin diğer öğeleri 0'a eşittir. b oluştuktan sonra aşağıdaki KOPYAYOKETME algoritmasında olduğu gibi sadece b'yi çaprazlamak Kopya Yok etme sorununu ortadan kaldırır.

```
Kopyayoketme(a, n)
m ← a'nın en büyük öğesi
i için ← m 0
     bi ← 0
i için ← n 0
   baı ← 1
i için ← m 0
   eğer bi =1
            çıktı i
geri dönüş
```

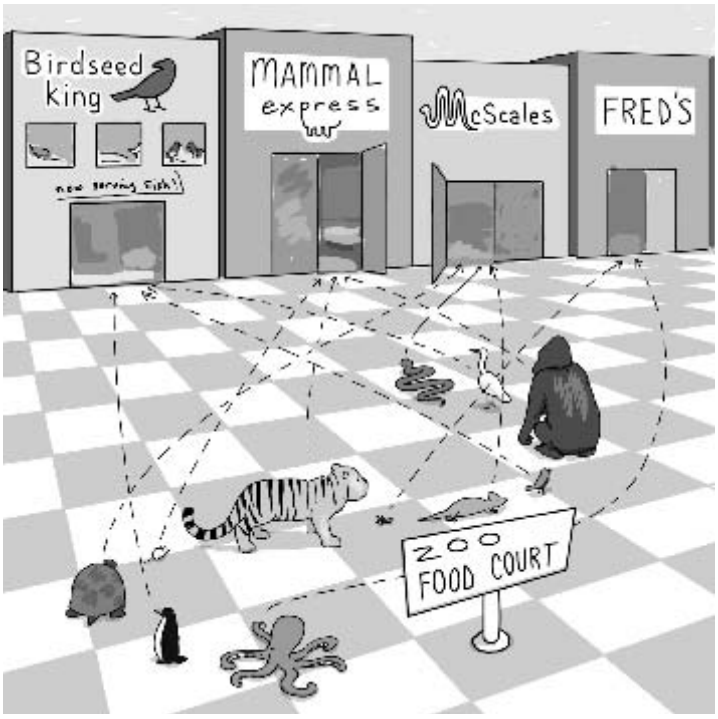
b dizini uzunluğuna oranlı zaman içinde yaratılır ve çaprazlanabilir, böylece bu algoritma b uzunluğunda çizgiseldir.3 Yüzeysel olarak, n'de çizgisel olmayan bir O (n log n) algoritmasından daha iyidir. Ancak, a'daki girdilerin isteğe bağlı olarak geniş olması zorluğu vardır. Eğer a=(1,5,1023) ise b'de 1023 öğe vardır. Bir listenin 1023 öğe ile yaratılması ve çaprazlanması etkili değildir, hatta muhtemel değildir.

Bu sorunu çözmek için herhangi bir tamsayıda işleyen ve o tamsayıyı küçük bir çerçevede bazı tamsayılara eşleyen bir fonksiyon belirleyebiliriz. (şekil 4.1'de şematik olarak gösterilmiştir) Örneğin, b listesinin 1000 girdiden daha az girdi içerdiğini varsayalım. Herhangi bir tamsayıyı alırsak (hatta 1023 bile olur) ve 1 ila 1000 arasında bir yere eşlersek o zaman bu "kutulama" stratejisini etkili şekilde uygulayabiliriz. Bu eşleme olayını uygulayan fonksiyona genellikle komut fonksiyonu adı verilir ve b listesine komut tablosu denir. Komut fonksiyonu –h-hesaplanması basit ve tamsayı değerlidir. 4 h(x)'yi |b|'nin b boyutunda olduğu yerde

x/|b|'nin tamsayı kalıntısına almak basit bir komut fonksiyonudur. Genellikle, b boyutu bilgisayar hafızasına yerleştirilmek için seçilir. Kopya Yok etme sorunu için komut tablosu b, a'daki aj öğesi için eğer h(aj)=i ise bi=1 kuralına göre, aksi takdirde bi=0 olarak yapılır. |b|=10 ise h(1)=1, h(5)=5 ve h(1023)=0 olur. a=(1,5,1023), b=(1,1,0,0,1,0,0,0,0,0) dizini ile sonuçlanır. Ancak, bu yaklaşım ai ve aj farklı öğeleri aynı kutuda yani h(ai) = h(aj) içinde çökebileceğinden kopya yok etme için hemen işe yaramaz [2].

İdeal şekilde, a dizinindeki farklı tamsayıların b'deki farklı tamsayılara eşleşmesini isteriz. Ancak a'daki benzersiz değer sayısı b uzunluğundan büyük ise bu açıkça imkansızdır. a'daki benzersiz değer sayısı b uzunluğundan küçük olsa dahi h komut fonksiyonunu farklı kutulara eşleyecek şekilde tasarlamak risklidir. Bunun yanı sıra, farklı her girdi için h'yi tekrar tasarlamak zorunlu değildir. Örneğin, eğer h(x)=x/1000 ise o zaman, 3, 1003, 2003 ve devamı çarpışır ve aynı kutuya düşer. Çarpışma ile başa çıkmanın genel bir tekniği zincirlemedir. Aynı kutuya yığılan öğeler genellikle bağlantılı bir listede düzenlenirler (şekil 4.2) Aynı kutuya yığılan öğelerin başka analizlerinde kopya öğeler ortaya çıkarılmıştır ve Kopya Yok etme sorunu için hızlı bir algoritmaya yön vermiştir.

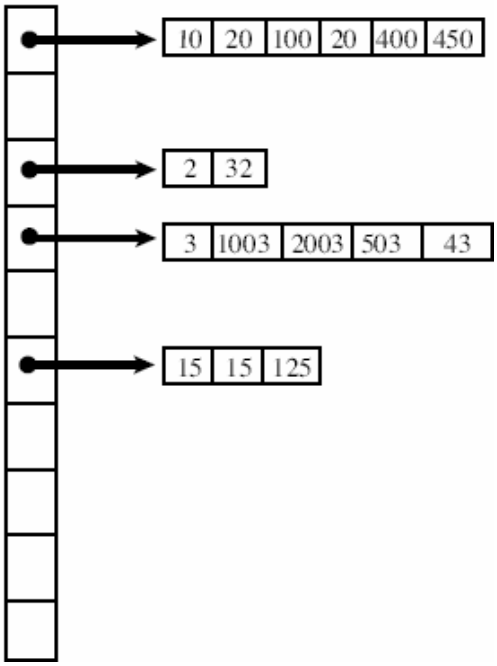
Bu bölümde listelerden kopya tamsayıları yok etme ile uğraşırken, komutlama olgusunu başka sorunlara doğru büyütmek nispeten kolaydır. Örneğin, bir genomda l-merlerin belirli tekrarlarını bulmak amacıyla genomu bir l-mer listesi olarak algılar ve komut tablosuna güveniriz.



x	h(x)
Penguen	1
Ahtapot	4
Kaplumbağa	3
Fare	2
Yılan	3
Balıkçıl	1
Kaplan	2
İguana	3
Maymun	2
Çekirge	4
Serçe	1

Şekil 4.1. Komutlama Örneği

Şekil 4.1 Bir komutlama resmi: kuşlar Birdseed King'de yiyorlar [h(x)=1], memeliler Mammal Express'de yiyorlar [h(x)=2], sürüngenler, yılanlar ve kaplumbağalar McScales'da yiyorlar [h(x)=3] ve diğer tüm hayvanlar Fred's'de yiyorlar [h(x)=4].



Şekil 4.2 Komut tablosunda zincirleme.

Komut tablosunda görebileceğimiz en fazla öğe sayısı vardır. Bu nedenle çarpışma ile ilgili hiçbir stratejiye gerek duymadık. Diğer yandan, büyük değerlerde belirli l tekrarları bulmak isteseydik (diyelim ki l=40), o zaman 440 girdilik bir komut tablosu oluşturamazdık. Onun yerine, uygun bir komutlama fonksiyonu tasarlayıp daha küçük boyda bir tablo kullanabilirdik.

4.3. Tam Örüntü Eşleme

Bilinen bir dizilim için dizilimler veri tabanı araştırmak biyo bilişimde genel bir sorundur. Bir örüntü dizgesine p=p1….pn ve daha uzun bir metin dizgesine t=t1…tm dendiği göz önünde tutulursa Örüntü Eşleme sorunu p örüntüsünün t metinindeki tüm oluşumlarını bulmaktır [3].

Örüntü Eşleme Sorunu:
Bir örüntü ve metini göz önünde tutarak metinde tüm örüntü oluşumlarını bulunuz.

Girdi: Örüntü p= p1….pn ve t=t1…tm

Çıktı: i'de başlayan t alt dizgesi n harfi p ile örtüşecek şekilde tüm konumlar $1 \leq i \leq m - n + 1$ (diğer bir deyişle ti…ti+n-1 = p1… pn).

Örneğin, eğer t= ATGGTCGGT ve p= GGT ise Örüntü Eşleme sorununu çözen algoritma 3 ve 7 konumları olur.

i konumunda başlayan t'den n uzunluğu alt dizgesini belirtmek için ti = ti…ti+n-1 simgesini kullanacağız. ti = p ise o zaman metinde örüntünün oluşumunu buluruz. Artan bir sırada i'nin tüm muhtemel değerlerini kontrol ederek soldan sağa doğru n uzunluğunun penceresini kaydırarak ve p örüntüsü oluştuğunda pencerenin yerini not ederek gerçekte t'yi tarıyoruz. Örüntü Eşleme sorununu çözmek için kullanılan kava kuvvet bir algoritma tam olarak bunu gerçekleştirir.

```
örüntüeşleme (p ,t)
n ← p örüntüsü uzunluğu
m ←t metni uzunluğu
i için ← m – n +1 'e 1
    eğer ti ←– = p ise
        çıktı i
```

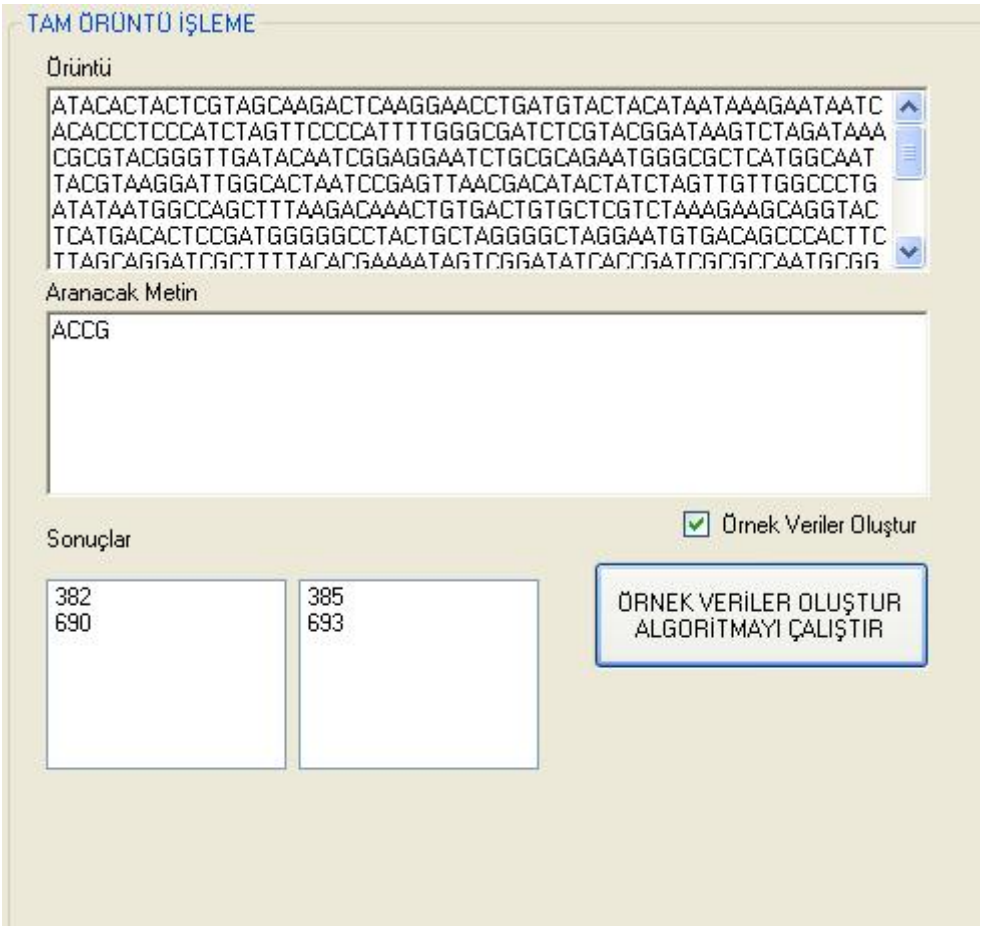
Her i konumunda örüntüeşleme (p, t), ti = p1, ti+1 = p2 vb olup olmadığını test ederek p'nin pencere içinde olup olmadığını doğrulamak için n kadar işleme gerek duyabilir. Tipik bir örnek olmak üzere, örüntü eşleme zamanının çoğunu i konumunun metinde oluşmadığını ortaya çıkarmaya harcar. Bu test p'nin i konumunda oluşmadığını gösteren tek işlemden oluşur (eğer ti ≠ p1). Ancak bu testi yapmak için olabildiğince çok n işlemine gerek duyarız. Bu nedenle, örüntü eşleme en kötü işlem süresi O (nm) olarak hesaplanabilir. Bu tür bir en kötü durum senaryosu t = AAAA…AAAAA 'da p= AAAT aranarak görülebilir. m 1 milyon ise araştırma yaklaşık 5 milyon işlem yapılmasına sebep olmakla kalmaz aynı zamanda hiçbir çıktı vermeyerek hayal kırıklığı yaratır.

Örüntü eşleme işlem süresi "rasgele" metinde örüntü bulunması ile değerlendirilebilir. Öncelikle, ilk yapılan testin uyuşmaz olma ihtimali yüksektir (t1'i p1 ile karşılaştırma). Bu nedenle p'nin geri kalan n – 1 hafrlerini kontrol etme sorunu çıkar. Genelde, örüntünün ilk harfinin i konumunda metinle eşleşmesi olasılığı

1/A'dır iken eşleşmeyeceği olasılığı A-1/A'dır. Benzer şekilde, örüntünün ilk iki harfinin metinle eşleşmesi olasılığı 1/A2 iken birinci harfin eşleşip ikinci harfin eşleşmemesi olasılığı A-1/A2'dir. örüntü eşleme'nin t1'de başlayan p'den n karakterleri arasından sadece j ile eşleşme olasılığı j için 1 ila n -1 arasında A-1/Aj'dir. j arttıkça bu sayının daraldığından örüntü eşleme'nin uzun testler uygulama olasılığı oldukça küçüktür. Bu nedenle, rasgele şekilde bütünleşik olarak üretilen bir metinle beraber sunulduğunda örüntü eşleme'nin p'yi kontrol ederken uygulaması gereken iş miktarı olumsuz en kötü durum senaryosu olan O(nm) işlem süresindense O(m)'ye daha yakındır. 1973 yılında Peter Weiner, her tür metin ve örüntüde Örüntü Eşleme sorununu çizgisel O(m) süresinde çözen son ek ağacı adı verilen dahice bir veri yapısı keşfetmiştir. Şaşırtıcı olan ise Weiner'in sonuçlarına göre Örüntü Eşleme sorununun karmaşıklığının yanında örüntünün boyutu hiç önemli değildir. Son ek ağaçlarını tanımlamadan önce örüntüleri çoğaltmak için Örüntü Eşleme sorununu yaratmada kullanılan anahtar sözcük ağaçlarını tanımlayacağız.

4.3.1. Örnek uygulama

Örnek programda rasgele bir örüntü ve aranacak metin yaratılarak algoritma işletilmektedir. Sonuçlar kısmına örüntü içinde aranan metin bulunduğunda başlangıç ve bitiş karakterleri yazılmaktadır.



Şekil 4.3. Tam örüntü işleme örnek uygulama ekran görüntüsü.

4.4. Anahtar Sözcük Ağaçları

Çoklu Örüntü Eşleme Sorunu:
Bir örüntü ve metini göz önünde tutarak metinde tüm örüntü oluşumlarını bulunuz.

Girdi: k örüntüleri p1, p2, …, pk ve t = t1 …tm serisi
Çıktı: $1 \leq j \leq k$ için pj örüntüsü ile i konumunda başlayan t alt dizgesi örtüşecek şekilde tüm konumlar $1 \leq i \leq m$

Örneğin, t= ATGGTCGGT ve p1= GGT, p2= GGG, p3= ATG, p4= CG ise Çoklu Örüntü Eşleme sorununu çözen algoritma sonucu 1, 3, 6 ve 7 konumları olur [4].

Tabiî ki, k örüntülü Çoklu Örüntü Eşleme sorunu k tekli Örüntü Eşleme sorununa indirgenebilir ve O(knm) sürede çözülebilir. Burada n, Örüntü Eşleme algoritmasının

k uygulamaları ile en uzun k örüntüsünün uzunluğudur. Eğer Örüntü Eşleme çizgisel bir örüntü eşleme algoritması ile yer değiştirirse Çoklu Örüntü Eşleme sorunu sadece O(km) sürede çözülebilir. Ancak, N'nin p1, p2, … , pk örüntülerinin toplam uzunluğu olduğu yerde O(N+m) sürede bu sorunu çözmek için daha hızlı bir yol vardır.

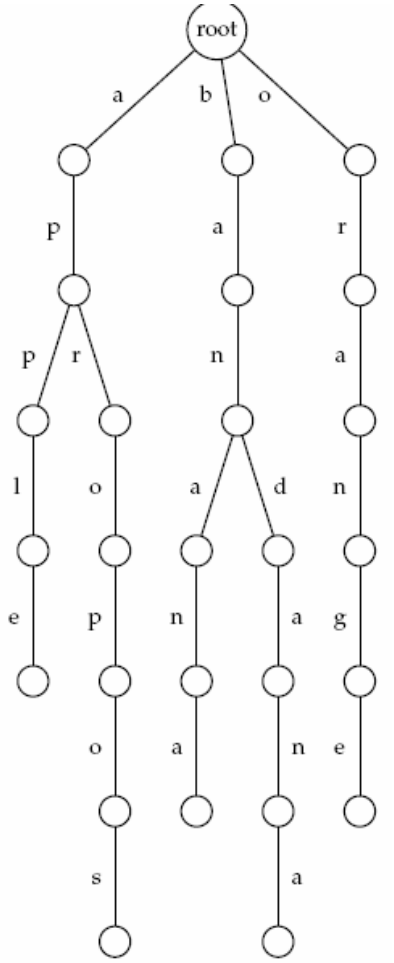
1975 yılında Alfred Aho ve Margaret Corasick, Çoklu Örüntü Eşleme sorununu çözmek için anahtar sözcük ağacı veri yapısını kullanmayı önermiştir. Apple, apropos, banana, bandana ve orange (elma, alakalı, muz, bandana ve portakal) örüntüleri serisi için anahtar sözcük ağacı şekil 4.3'te gösterilmiştir. Daha resmi olmak gerekirse, p^1, p^2, …, p^k örüntü serisinin anahtar sözcük ağacı, serideki hiçbir örüntünün başka bir örüntünün ön eki olmadığı basitliğini varsayan aşağıdaki şartlara uygun köklü etiketli bir ağaçtır:

* Ağacın her kenarı alfabenin bir harfi ile etiketlenmiştir,
* Aynı noktadaki herhangi iki kenar belirgin etiketlere sahiptir,
* Örüntü serisindeki tüm p^i ($1 \leq i \leq k$) örüntüleri kökten yaprağa kadar bir yolda hecelenir.

Açıkça görülüyor ki, N'nin tüm örüntülerin toplam uzunluğu olduğu yerde anahtar sözcük ağacı en çok N köşe noktasına sahiptir, ancak bu sayı azabilir. Anahtar sözcük ağacı, ilk j örüntüleri için anahtar sözcük ağacını j+1 örüntüleri için anahtar sözcük ağacına düzenli şekilde genişleterek O(N) sürede oluşturulabilir.

Anahtar sözcük ağacı, p^1, p^2, …, p^k serisinde metnin belli bir i konumunda başlayan metinle eşleşen bir örüntü olup olmadığını bulmak için kullanılabilir. Bunu yapmak için şekil 4.4'te gösterildiği üzere her aşamada nereye hareket edeceğine karar vermek için metnin t_i t_i+_1 t_i+_2 harfleri kullanılarak anahtar sözcük ağacı çaprazlanabilir. Bu işlem ya bir yaprakta sonlanır -ki bu durumda yaprak tarafından temsil edilen bir örüntüye eşleme vardır- ya da yaprağa ulaşmadan sekteye uğrar –ki bu durumda da i konumunda başlayan hiçbir eşleşme yoktur. En uzun örüntü uzunluğu n ise, anahtar sözcük ağacını oluşturmak ve ardından metinde arama yapmak için kullanmak için Çoklu Örüntü Eşleme sorunu O(N + nm) sürede

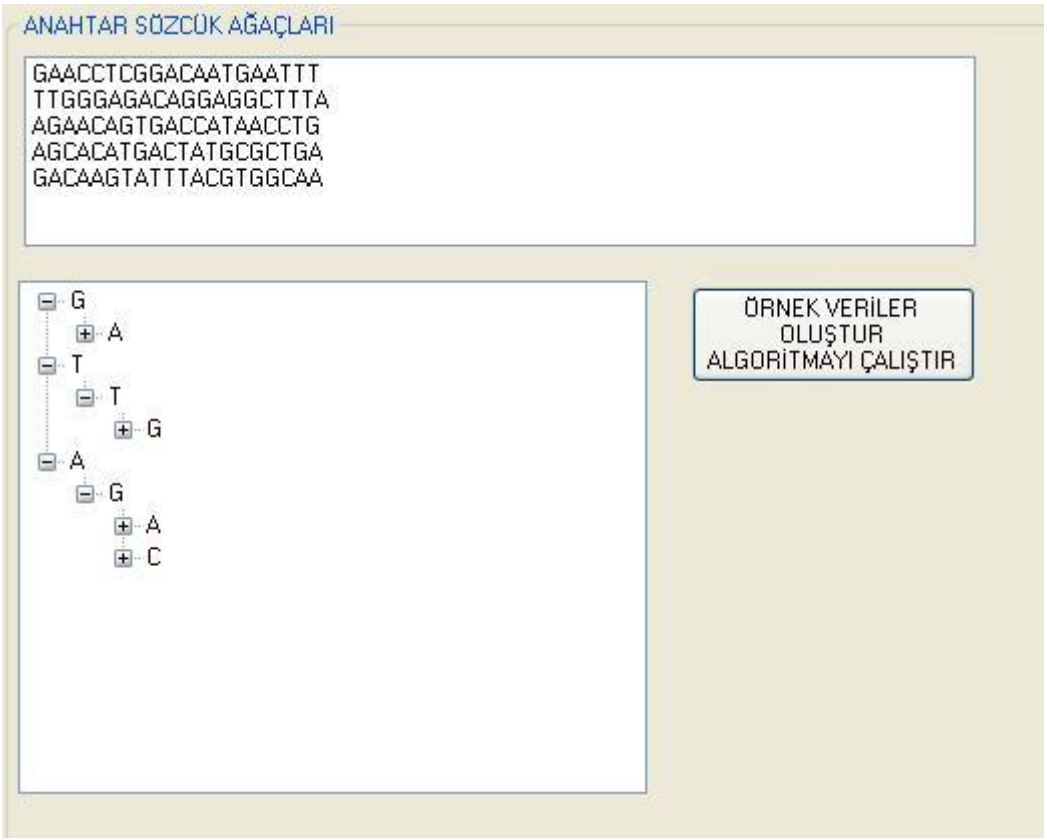
çözülebilir. Henüz ayrıntılarını vermediğimiz Aho-Carosick algoritması ayrıca Çoklu Örüntü Eşleme sorununun işlem süresini O(N + m)'ye düşürür.



Şekil 4.4. apple, apropos, banana, bandana ve orange anahtar sözcük ağacı

4.4.1. Örnek uygulama

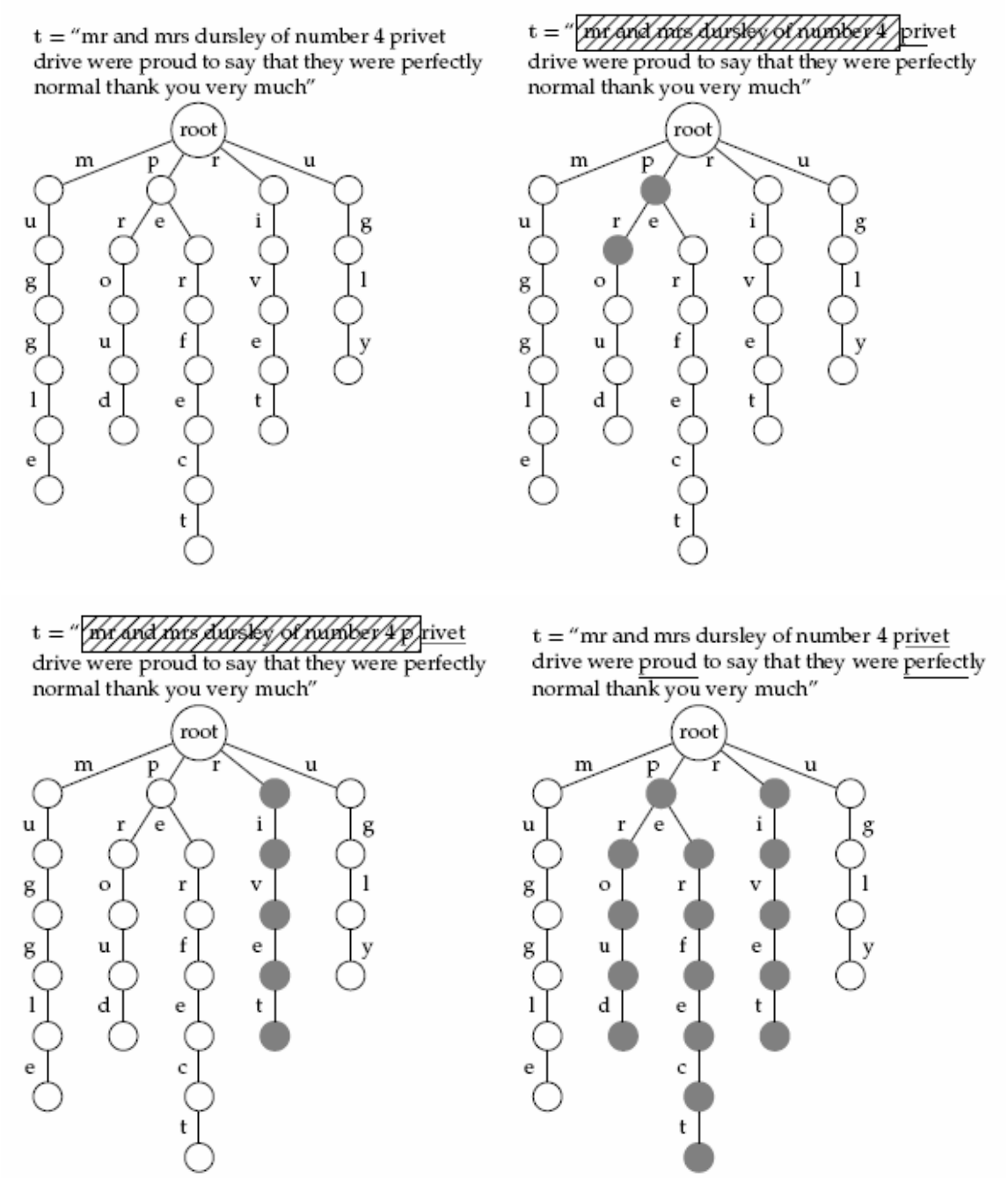
Örnek programda rasgele örüntüler oluşturularak bu örüntülerden anahtar sözcük ağacı oluşturulmaktadır. Oluşturulan sözcük ağacı, kendisini oluşturan örüntüler içinde arama yapmayı kolay hale getirmektedir.



Şekil 4.5. Anahtar sözcük ağaçları örnek uygulama ekran görüntüsü

4.5. Son Ek Ağacı

Son ek ağacı sayesinde metnin uzunluğuna bağlı kalınmaksızın sadece O(n) süre kullanılarak n uzunluğunda herhangi bir örüntünün metinde oluşup oluşmadığını görecek şekilde metin ön işleme tabii tutulur. Yani, Science bültenindeki tüm konular içinden kocaman bir kitap yaratmışsanız p'nin uzunluğuna orantılı şekilde süre içinde p örüntüsü araştırabilirsiniz. Şimdiye kadar yazılan tüm kitapları birleştirirseniz p'yi araştırmak aynı süreyi kapsayacaktır. Tabiî ki, son ek ağacını oluşturmak için gereken süre iki metin için farklıdır.



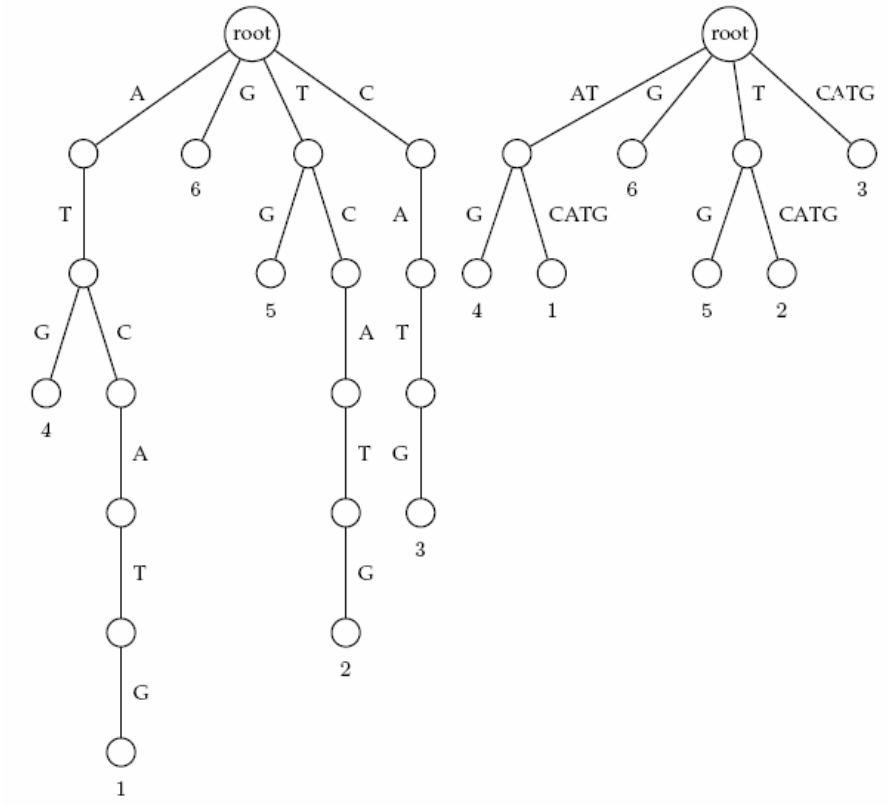
Şekil 4.6. proud, perfect, ugly, rivet, muggle anahtar sözcüklerini "mr and mrs dursley of number 4 privet drive were proud to say that they were perfectly normal thank you very much" metninde arama

m uzunluğunda bir metnin son ek ağacı O(m) sürede tamamlanabilir ve Örüntü Eşleme sorununda çizgisel bir O(n+m) algoritmasına yön verir: t için O(m)sürede son ek ağacı oluşturmak ve ardından O(n) süre gerektiren p'nin ağaçta oluşup oluşmadığını kontrol etmek.

Şekil 4.5 (a)'da AT-CATG: G, TG, ATG, CATG, TCATG ve ATCATG dizgelerinin altı son ekinin de anahtar sözcük ağacı gösterilmektedir. Şekil 4.5 (b)'de olduğu gibi dallandırılmayan tüm köşe noktalarını tek bir kenara çökerterek bir metnin son ek ağacı son eklerinin anahtar sözcüklerinden elde edilebilir. Son ek ağacı oluşturmak için anahtar sözcük yapısı kullanılabilmesine rağmen kendisi son ek ağacı yapısı için etkili bir algoritma üzerine konumlanmaz. Aslında, k örüntüleri için anahtar sözcük ağacı N'nin tüm bu örüntülerin toplam uzunluğu olduğu yerde O(N) sürede yapılabilir. m uzunluğu metninin, uzunluğu 1 ila m arasında değişen m kadar son eki vardır; bu nedenle bu son eklerin toplam uzunluğu $1 + 2 + \ldots + m = m(m+1)/2$, yani metinde ikinci derece uzunluktur. O(m) sürede son ek oluşturma olayına Weiner'in katıkısı anahtar sözcük oluşturma aşamasının tümünü atlamaktadır.

t= $t_1 \ldots t_m$ metninin son ek ağacı aşağıdaki koşulları yerine getiren (1 ila m arasında numaralandırılmış) m yapraklı köklü bir etiketli ağaçtır:
Her kenar metnin alt dizgesi ile etiketlendirilmiştir,
Her dahili tepe noktası (muhtemelen kök hariç) en az 2 çocuğa sahiptir,
Aynı tepe noktasının herhangi iki kenarı farklı bir harfle başlar,
t metninin her son eki kökten yaprağa kadar bir yolda hecelenir.

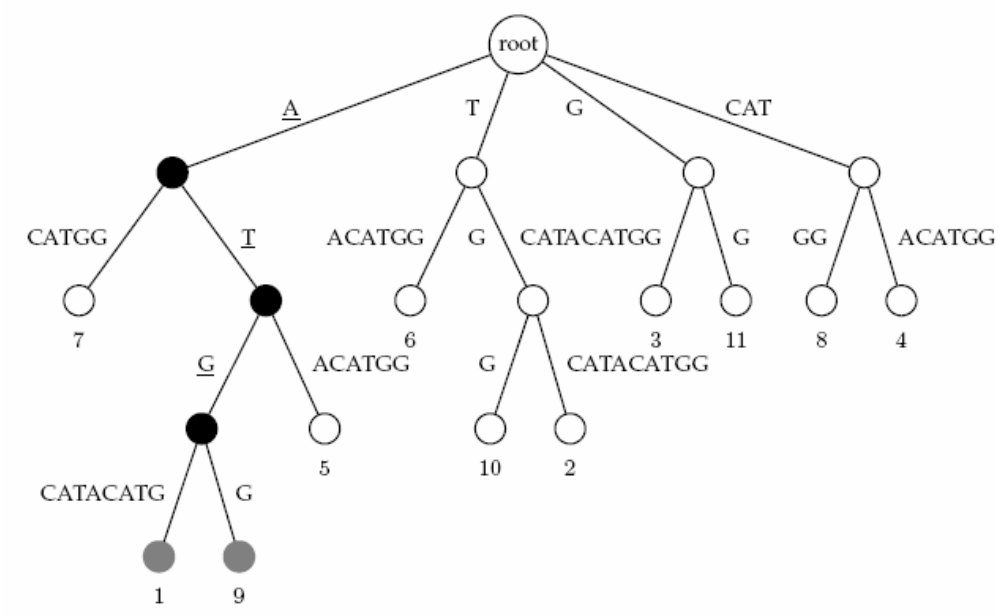
Son ek ağaçları örüntü eşleme için derhal hızlı bir algoritmaya yönlenir. p örüntüsünün son ek ağacı içerisinde düğümünü ya tam düğüm denilen p'nin bütün karakterleri eşleşene kadar ya da yarı düğüm denilen eşleşmenin mümkün olmamasına kadar T'de benzersiz bir yol boyunca p'den karakterlerin eşleşmesi olarak tanımlıyoruz. Örüntünün düğümü tamamlanmış ise T'nin köşe noktasında veya kenarında sona erer. p-eşleşme yapraklarını, o köşe noktasının yada kenarın soyundan gelen tüm yapraklar olarak tanımıyoruz. Ağaçtaki her p-eşleşme yaprağı t'de p oluşumuna karşılık verir. Tam düğüm ve p-eşleşme yapraklarının örneği şekil 4.6'da verilmiştir.



Şekil 4.7 (a) anahtar sözcük ağacı

Şekil 4.7 (b) son ek ağacı

Sonekağacıörüntüeşleme (p, t)
t metni için son ek ağacı oluşturun
son ek ağacı yoluyla p örüntüsünü düğümleyin
düğümleme tamamlanmış ise
    çıktı ağaçtaki tüm p-eşleşme yapraklarının konumu
ya da
    çıktı "örüntü metnin hiçbir yerinde oluşmamaktadır"



Şekil 4.8 ATG örüntüsünü ATGCATACATGG metni için son ek ağacı yoluyla düğümleme.

Ağaçtaki gri noktalarda görüldüğü üzere ATGCATACATGG ve ATGG son eklerinin her ikisi de eşleşir (p-eşleşme yaprakları). Her p-eşleşme yaprağı p'nin oluştuğu her yerde metin içindeki bir konuma karşılık gelir.

Sonekağacıörüntüeşleme'nin t'de kesin p oluşumu aradığını vurgulamaktayız. Aşağıda kesin olmayan eşleşme için birkaç algoritma açıklayacağız.

4.6. Buluşsal Benzerlik Arama Algoritmaları

Bundan yirmi yıl önce yeni dizilmiş kansere sebebiyet veren gen ile normal büyüme ve gelişme gösteren gen arasındaki benzerlikleri bulmak şaşırtıcı bir olaydı. Günümüzde ise GenBank veri tabanının büyüklüğü göz önüne alınırsa yeni dizilmiş gen için eşleşme bulamama daha da şaşırtıcıdır. Ancak, GenBank veritabanını aramak yirmi yıl önce olduğu kadar basit değildir. Yukarıda bahsi geçen son ek ağacı hızlı olmasına rağmen bir veri tabanında sadece kesin gen oluşumlarını bulur, benzer ya da yaklaşık oluşumları bulmaz. 10^{10} boyutunda bir veri tabanında 10^3 uzunluğunda bir gene yaklaşık bir eşleşme bulmaya çalışırken ikinci dereceden dinamik programlama algoritmaları çok yavaş kalabilir (Smith-Waterman bölgesel hiza algoritması gibi). Her biri 3 x 10^9 nükleotidli olan iki memeli genomu arasındaki benzerlikleri bulmak ise çok daha zordur. 1990ların başından itibaren biyologlar

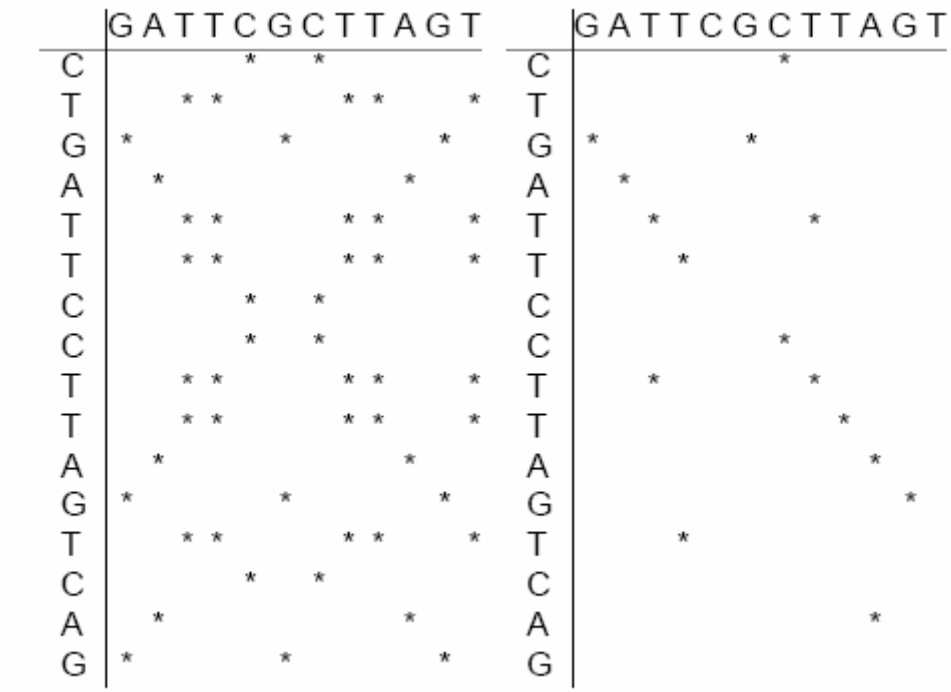
ikinci dereceden dizilim hizası algoritmalarına alternatif olarak hızlı buluşsal yöntemler kullanmaktan başka yola başvuramamıştır [7].

Moleküler biyolojide hızlı veri tabanı araması yapan birçok buluşsal yöntem aynı *filtreleme* fikrini kullanır. Filtreleme, iyi bir hizalamanın kısa eş veya büyük oranda benzer parçacıklar içerdiği gözlemine dayanır. Bu nedenle, örneğin komut tablosu veya son ek ağacı kullanılarak kısa kesin eşlemeler aranabilir ve bu kısa eşlemeler ilerideki araştırmalar için başlangıç olabilir. 1973 yılında Donald Knuth, tek uyuşmaz ile farklılaşan dizgelerin birinci veya ikinci yarı ile tam olarak eşleşmesi gerektiği gözlemine dayanan tek uyuşmazlı bir örüntü eşleme yöntemi önermiştir. Örneğin, tek izin verilebilir hatalı 9-merleri eşleme, kesin 4-mer eşlemelerini yaklaşık 9-mer eşlemelerine genişleterek kesin 4-mer eşlemeye indirgenebilir. Bu durum bize tüm konumlardaki çiftlerin çoğunu kapsayan ortak 4-mer paylaşmayan konumları filtreleme fırsatı sağlar. 1985 yılında bilgisayarlı moleküler biyolojide filtreleme fikri FASTA algoritmasında David Lipman ve Bill Pearson tarafından kullanılmıştır. BLAST'da daha da geliştirilmiştir ve artık moleküler biyolojide yaygın bir veri tabanı arama motoru haline gelmiştir.

Biyologlar sıklıkla *nokta matris* formunda iki dizilim arasındaki benzerlikleri tanımlarlar. Nokta matris, her girdisi ya 0 ya 1 olan ve şekil 9.7'de gösterildiği gibi ilk dizilimde *i* konumu ve ikinci dizilimde *j* konumu arasında *(i, j)* konumunda bazı benzerlikleri belirten basit bir matristir. Benzerlik kriterleri değişkendir. Örneğin, birçok nokta matrisi, ilk dizilimdeki *i* konumu ile ikincideki *j* konumu arasındaki hizada en çok *k* uyuşmazlı *t* uzunluğu eşlemelerine dayalıdır. Ancak, şans eseri benzerliklerden biyolojik açıdan alakalı benzerlikleri ayırma konusunda hiçbir kriter mükemmel değildir. Biyolojik uygulamalarda, ses genellikle gerçek benzerliği gizler ve biyolojik açıdan alakalı benzerlikleri filtrelemeden sesi nokta matristen ayırma sorunu ortaya çıkar.

Bir dizilim ve veri tabanı ile şekillenen *l*-mer konumları bu dizgeler arasında benzerliklerin dahili bir nokta matris temsilciğini oluşturur. Popüler FASTA protein veri tabanı arama motoru, nokta matris yaratmak için *l* uzunluğunda kesin eşlemeler

kullanır. Bu nokta matristen 1'lerin yoğun olduğu bazı köşegenler seçer ve bu köeşegende daha uzun süre işlenmek üzere 1 işlemlerini gruplar.



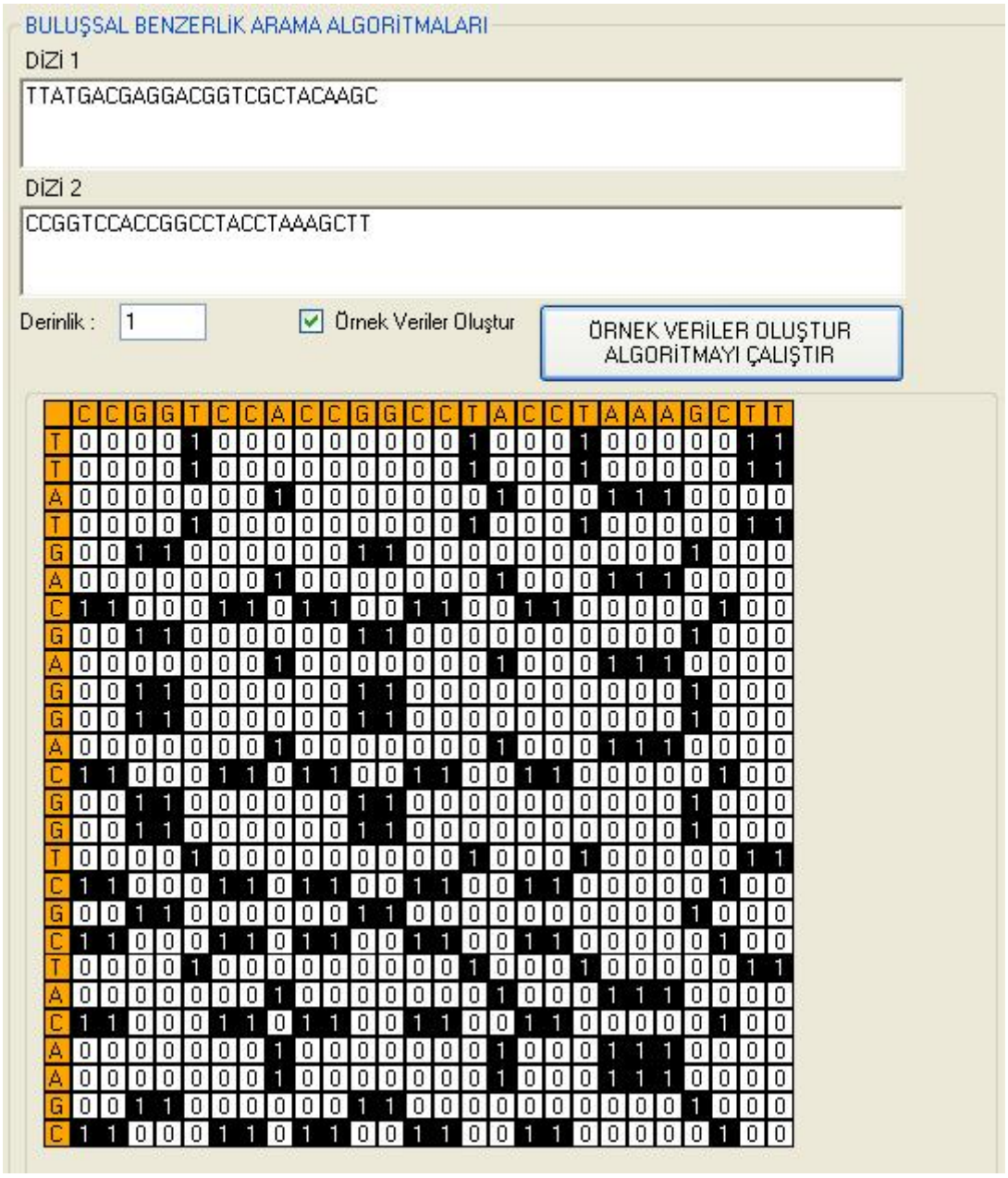
Şekil 4.9. CTGATTCCTTAGTCAG ve GATTCGCTTAGT dizgeleri için iki nokta matrisi.

İlk matristeki (i, j) konumundaki nokta, birinci dizilimdeki i nükleotidinin ikincideki j nükleotidi ile eşleştiğini ortaya koyar. İkinci matristeki (i, j) konumundaki nokta, birinci dizilimdeki i dinükleotidi ile ikincideki j dinükleotidinin aynı olduğunu, diğer bir deyişle birinci dizilimdeki i ve (i + 1) sembollerinin ikinci dizilimdeki j ve (j + 1) sembolleri ile eşleştiğini ortaya koyar. İkinci matriste, birinci dizilimdeki GATTCCTTAGT ile ikinci dizilimdeki GATTCGCTTAG alt dizgeleri arasında yaklaşık bir eşleşmeye işaret eden 9 noktalı nerdeyse köşegen bir işlem görülür. Aynı köşegen işlem birinci matriste de mevcuttur ancak birçok suni nokta ile değiştirilmiştir.

4.6.1. Örnek Uygulama

Örnek programda rasgele 2 örüntü oluşturulmakta ve derinlik parametresine göre algoritma örüntüler üzerinde işletilmektedir. Derinlik parametresi arttıkça

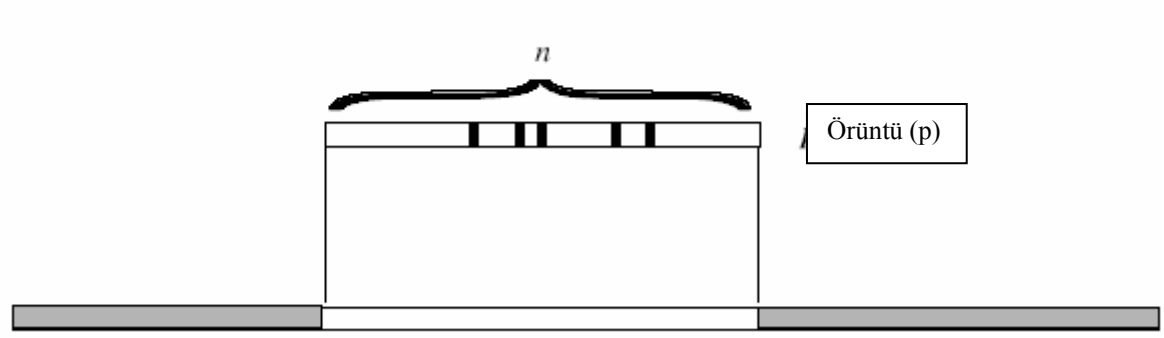
algoritmaya göre oluşan matristeki “1” değerleri azalacaktır. Buluşsal benzerlik arama algoritması derinlik parametresine göre iki örüntünü benzerliklerini bize göstermektedir.



Şekil 4.10. Buluşsal benzerlik arama algoritmaları örnek uygulama ekran görüntüsü.

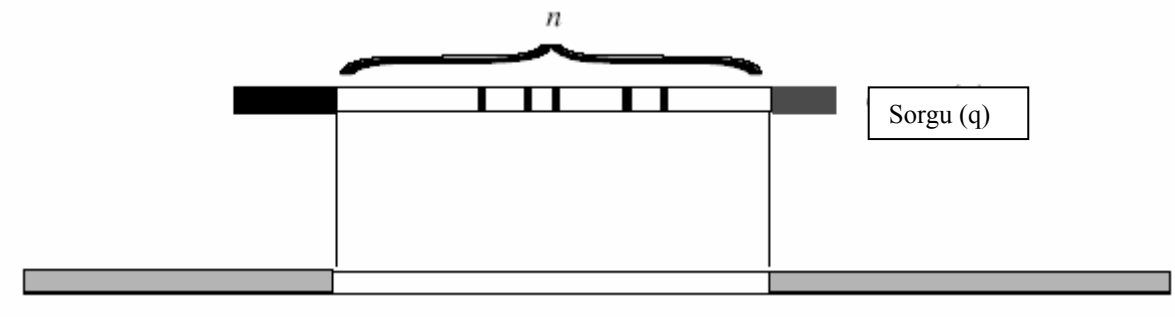
4.7. Yaklaşık Örüntü Eşleme

Bilgisayarlı moleküler biyolojide bir metindeki örüntünün yaklaşık eşlemelerini bulmak diğer bir deyişle k’li metindeki tüm alt dizgeleri bulmak önemli bir sorundur.



metin (t)

Şekil 4.11. (a) Yaklaşık Örüntü Eşleme



metin (t)

Şekil 4.11. (b) Sorgu Eşleme

Yaklaşık Örüntü Eşleme sorunu ile Sorgu Eşleme sorunu arasındaki fark şudur: Yaklaşık Örüntü Eşleme sorunu metne karşı n uzunluğunun tüm örüntülerini eşlerken Sorgu Eşleme sorunu metne karşı sorguda n uzunluğunun tüm alt dizgelerini eşler.

Yaklaşık Örüntü Eşleme Sorunu:
Bir metinde tüm yaklaşık örüntü oluşumlarını bulunuz.

Girdi: $p = p_1p_2 \ldots p_n$ örüntüsü, $t = t_1t_2 \ldots t_m$ metni, k parametresi ve maksimum uyuşmazlık sayısı

Çıktı: $t_it_{i+1} \ldots t_{i+n-1}$ ve $p_1p_2 \ldots p_n$ en çok k uyuşmazlığına sahip olacak şekilde tüm konumlar $1 \leq i \leq m - n+1$ (diğer bir deyişle, $d_H(t_i, p) \leq k$).

Yaklaşık örüntü eşleme için naif kaba kuvvet algoritması O(nm) sürede işler. Ardından gelen algoritma, p'nin k kadar uyuşmazlıktan daha fazla oluşmadığı yerde t'deki tüm konumları çıkarır.

```
Yaklaşıkörüntüeşleme (p, t, k)
1       n ← p örüntüsü uzunluğu
2       m← t metni uzunluğu
3       i için ← m- n +1'e 1
4           uzaklık ←0
5           j için ← n'ye 1
6               T_{i+j-1} ≠ p_j ise
7                   uzaklık ← uzaklık + 1
8           uzaklık ≤ k ise
9                   çıktı i
```

1985 yılında Gadi Landau ve Udi Vishkin O(km) en kötü durumda işleme süreli yaklaşık dizge eşleme için bir algoritma bulmuşladır. Bu algoritma bilinen en kötü senaryoyu sağlamasına rağmen uygulamada en iyisi değildir. Buna bağlı olarak, birçok filtreleme temelli yaklaşımın, en kötü senaryo işlem süreleri daha kötü olmasına rağmen uygulamada daha iyi işlem süreleri vardır.

Sorgu Eşleme sorunu, Yaklaşık Örüntü Eşleme sorununu geneller. İki sorun arasındaki fark şekil 9.8'de gösterilmiştir. k uyuşmazlı sorgu eşleme sorgunun bir kısmı ile metnin bir kısmı arasındaki Hamming uzaklığını sınırlamak için bir t metni, bir q sorgu dizilimi ve bir k parametresi içerir. Ayrıca elimizde eşlemenin ortalama uzunluğu olan n parametresi bulunmaktadır. Sorgu Eşleme sorunu, tüm bir dizge (p) ile t'den n uzunluğunun tüm alt dizgeleri arasında yaklaşık eşlemeler aramak yerine sorgudan n uzunluğunun herhangi bir alt dizgesi ile metinden n uzunluğunun diğer bütün alt dizgeleri arasında yaklaşık eşlemeler arar.

Sorgu Eşleme Sorunu:

Metinle yaklaşık olarak eşleşen tüm sorgu alt dizgelerini bulunuz.

Girdi: Sorgu $q= q_1 \ldots q_p$, metin $t= t_1 \ldots t_m$ ve tam sayılar n ve k.

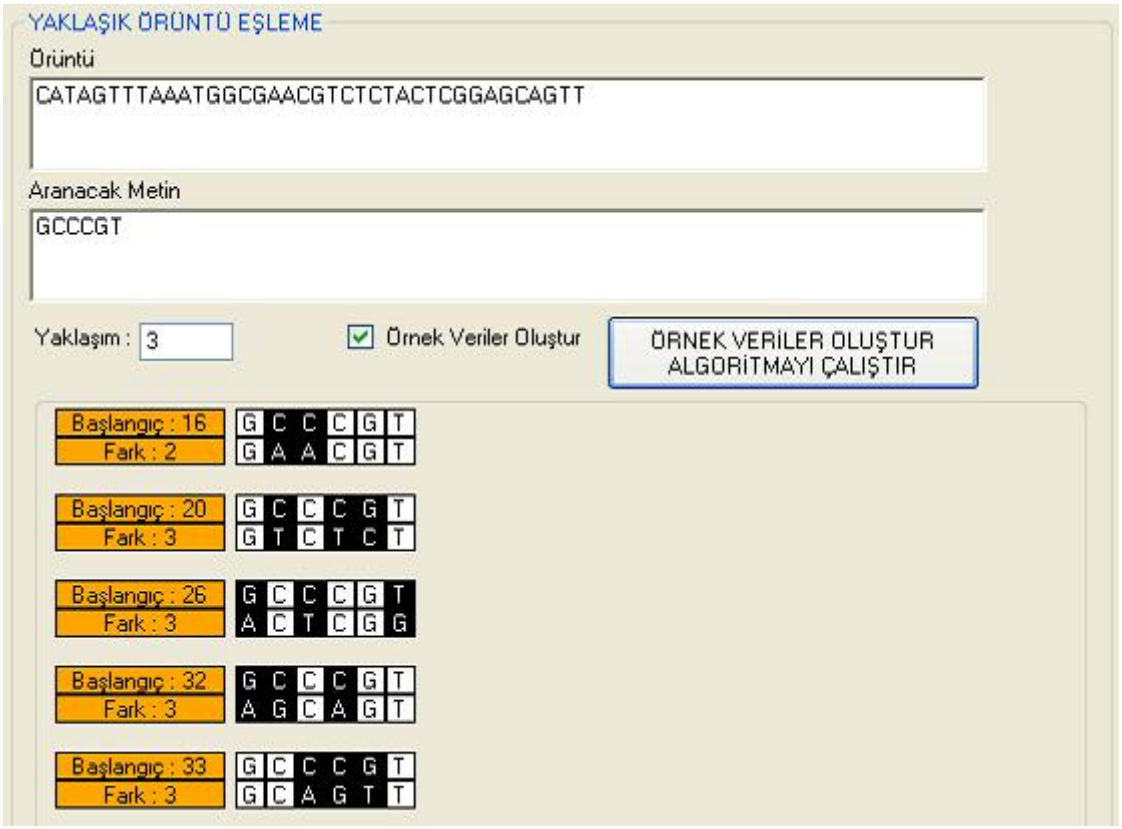
Çıktı: i'de başlayan q'nun n harfli alt dizgeleri en çok k uyuşmazlı j'de başlayan t'nin n harfli alt dizgeleri ile yaklaşık olarak eşlemesi şeklinde $1 \leq i \leq p - n + 1$ ve $1 \leq j \leq m - n + 1$ olduğu yerde tüm (i, j) konum çiftleri.

p=n iken Sorgu Eşleme sorunu, k uyuşmazlı Yaklaşık Dizge Eşleme sorununa dönüşür.

Yaklaşık sorgu eşleme için filtreleme algoritmaları kullanmak iki aşamalık bir işlemdir. İlk aşamada, metin içinde sorguya benzer konum serisi önceden seçilir. İkinci aşamada k'dan çok uyuşmazlı potansiyel eşlemeler reddedilerek her potansiyel konum doğrulanır. Potansiyel eşlemelerin sayısı az ve potansiyel eşleme doğrulaması çok yavaş değil ise bu yöntem birçok girdide uygulamada artan hızlı bir sorgu eşleme algoritması kazandırır.

4.7.1. Örnek uygulama

Örnek programda rasgele bir örüntü ve aranacak metin yaratılarak yaklaşım parametresine göre algoritma örüntü ve aranacak metin üzerinde çalıştırılmaktadır. Aranacak metin, örüntü içerisinde arandığında bulunan farkların sayısı yaklaşım parametresinden küçük veya eşit ise bulunan kısım sonuçlar bölümünde görünmektedir.



Şekil 4.12. Yaklaşık örüntü eşleme örnek uygulama ekran görüntüsü

BÖLÜM 5. İNSAN GENOMU PROJESİ

5.1. Giriş

26 Haziran 2000 tarihinde Clinton ve Blair'in video konferansında insan DNA'sının %97' sinin şifresinin çözüldüğünü açıklandı. Bu insanlık için yeni bir dünyanın kapısını açan bir sonuçtu. Amerikan Enerji Bakanlığı ve Ulusal Sağlık Enstitüsünün beraberce yürüttüğü bu projede amaç tüm insanlarda bulunan genetik materyalin haritasını çıkarmaktır [6].

Bütün insanlar, vücutlarındaki trilyonlarca hücrede anne ve baba'dan 23'er çift (toplam 46 adet) kromozomda genetik materyalleri (genom) taşımaktadır. İnsan hücrelerinde yarısı anneden yarısı babadan olmak üzere 6 milyar baz dizisi bulunmasına rağmen, ancak 1000 bazdan biri diğer kişilerden farklıdır. DNA dizisinin %99.9'u insanlarda aynıdır. Kişiler arasındaki fark % 0.1'lik kısımdan meydana gelir.

İnsan genomu projesinde oldukça ileri teknoloji kullanılmasına rağmen daha hızlı ve kesin sonuçlar veren yöntemlerin eklenmesi gerekmektedir. Şu anda haritalama ve dizileme metodları kullanılmaktadır. Haritalama aşamasında kromozomlar tanımlanabilecek en küçük parçaya ayrılır, daha sonra kromozomdaki dizilişine uygun olarak sıralanır. Haritalama bitince bu fragmanların DNA dizisini çözmeye sıra gelir.

Projenin tamamlanacağı zamanı vermek çok güçtür. Kuşkusuz insan genomu projesindeki gelişmeleri, diğer teknolojik gelişmelerle de beraber düşünmek gerekmektedir. Bilgisayar teknolojisinde ve biyobilişimde gelişmeler olmasaydı belki bugünkü noktaya daha geç ulaşılacaktı. İnsan Genomu Projesini diğer

bilimlerle paralel olduğunu düşünülürse, 20-25 yıl daha süreceğini tahmin edilmektedir.

5.2. İnsan Genomu Projesinin Tarihçesi

1953: James Watson ve Francis Crick DNA'nın çift sarmal yapısını keşfettiler.

1972: Paul Berg ve arkadaşları ilk Rekombinant DNA molekülünü yarattılar.

1980:Massachusetts Teknoloji Enstitüsü'nden David Botstein,Stanford Üniversitesinden Ronald Davis , ve Utah Üniversitesi'nden Mark Skolnick ile Ray White, tüm insanların genom haritasının oluşturulmasına yarayacak bir yöntem önerdiler.

1991: Ulusal Sağlık Enstitüsü'nden D. Craig Venter, özel genlerin bulunmasına yarayan bir yöntemi kamuya sundu.Caltech'ten Mel Simon ve arkadaşları klonlama için BAC ürettiler.ABD ve Fransa'dan ekipler ilk fiziksel kromozom haritalarını tamamladılar.ABD'li ve Fransız ekipler,fare ve insanın genetik haritalarını tamamladılar.

1998: 2001 yılına dek insan genomunun "geçerli bir taslağını" oluşturmak hedeflendi.Bu arada taslağa son şeklinin verilmesi ile ilgili tarih 2005 yılından 2003'e alındı.

1999: Ulusal Sağlık Enstitüsü, ilk taslağın tamamlanma tarihini yeniden öne alarak,bu kez 2000 baharı olarak belirledi.

2000: Beyaz Saray'da yapılan bir törenle,İnsan Genomu Projesi ve Celera,ortaklaşa olarak insanın genetik diziliminin ilk taslağının tamamlandığını ilan ederek.çalışmaların süreceğini belirttiler**.**

5.3. İnsan Genomu Projesinin Gelecekteki Yararları Nedir?

1. İnsan Genomu Projesinin sonuçları genlerin fonksiyonlarını ve hastalıklara hangi fonksiyon bozukluklarının yol açtığının anlaşılmasını sağlayacaktır. Böylece tedavi hastalıkların önceden kesin olarak teşhisine ve kaynağın ortadan kaldırılarak muhtemel zararların önlenmesine yönelik olacaktır.
2. Bu proje sonuçlarını bugün ve gelecekteki kullanım alanları şu şekilde sıralanabilir: Moleküler tıp, mikrobial genetik, hastalık risk faktörlerinin belirlenmesi, bioarkeoloji, adli tıp, tarım ve hayvancılık.
3. İnsan Genomu Projesi sayesinde kanserin genetik temelleri giderek daha çok aydınlatılır duruma gelecektir. İnsan Genomu Projesi bu genlerin dolayısıyla bu genlerin ürünü olan proteinlerin yapılarını ortaya çıkardıkça kanserin tanısı gen düzeyinde olabilecek, dna düzeyinde tedavisi mümkün hale gelebilecek.
4. İnsan DNA'sındaki kişisel bilgilerin edinilmesi sonucu kişiye özel ilaç döneminin başlayabilecektir.
5. 2025 yılında genetik marketler olabilecektir. Gen transferi yaptırmak isteyenler kök hücre değişikliğiyle göz rengini değiştirmeyi düşünenler bu marketlere gidebilecekler.
6. Bakteri mantar, tek hücreli hayvanlar gibi mikroorganizmaların genomlarının deşifre edilmesiyle, yeni alternatif biyoyakıt kaynakları keşfedilebilecektir.
7. Her türlü cinayetlerde ve adli vakalarda, failin geride bıraktığı hücre örneklerinden gerçek suçluyu belirlemede büyük ilerlemeler olacaktır.

BÖLÜM 6. SONUÇLAR VE ÖNERİLER

Tüm dünya genelinde biyoenformatik araştırmaları oldukça yeni denebilecek 15-20 yıllık bir geçmişe sahiptir. Ancak son 5 yıl içinde çok hızlı ve önemli gelişmelere tanık olunmaktadır. Önceleri sadece birkaç araştırma merkezinde değişik disiplinlerden araştırıcıların eğitimi ile başlayan bu bilim dalı şimdilerde birçok üniversite programında disiplinler arası bir kimlikle akademik derece veren öğretim programlarına dönüştürülmektedir.

Biyoenformatiğe sadece genetik bilginin analizi konusunda bakılması da yanlış olur, gerçekte tüm düzeylerde biyolojik bilgi için biyoenformatik geçerli bir kavramdır. Tek bir bölüm, fakülte ve hatta üniversite bu yeni alandaki insan ihtiyacını karşılamayabilir. Aslında temel konular için fakülteler ve bölümler oluşurken yeni mühendislik alanları için disiplinler arası programlar uygulamasına gidilmelidir. Bu durum, dünya'da ve özellikle ABD'de de gittikçe egemen olmaya başlayan bir yaklaşımdır. Bilim dallarında görülen hızlı ilerleme ve artık bir önceki döneme göre geometrik şekilde katlanarak artan büyük hacimdeki bilgiyi yönetmek ve işlemek geleneksel yöntemlerle mümkün olamamaktadır. İşte böyle bir noktada enformatik veya biyoenformatik gibi gereksinimlerin son derece doğal bir zorlamayla oluşan yeni bilimlerin ve uygulamalı mühendisliklerin önüne konulan bürokratik engellerin kaldırılarak yeni ihtiyaçlar doğrultusunda gelişmesine olanak tanınması gerekir. Üstelik Enformasyon Çağı'nın henüz başında olduğumuz unutulmamalıdır.

KAYNAKLAR

[1] A. Aho, J. Hopcroft, and J. Ullman. Data Structures and Algorithms. Addison-Wesley, Boston, 1983.

[2] A.V. Aho andM.J. Corasick. Efficient stringmatching: an aid to bibliographic search. Communication of ACM, 18:333–340, 1975.

[3] AlanWee-Chung Liewa, HongYanb, MengsuYangd Pattern recognition techniques for the emerging field of bioinformatics

[4] B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, and J. Watson. Molecular Biology of the Cell. Garland Publishing, New York, 1994.

[5] D.W. Mount, Bioinformatics—Sequence and Genome Analysis, Cold Spring Harbor Laboratory Press, New York, 2001.

[6] Neil C. Jones, Pavel A. Pevzner An Introduction to Bioinformatics Algorithms.

[7] T.F. Smith, M.S. Waterman, Identification ofcommon molecular subsequences, J. Mol. Biol. 147 (1981)

[8] Ulaş Baran Baloğlu Dna sıralarındaki tekrar örüntülerin ve potansiyel motiflerin veri madenciliği yöntemiyle çıkarılması 2006

ÖZGEÇMİŞ

Sedat Alan, 30.08.1984 de Bursa'da doğdu. İlk öğretimi Emek ilk öğretim okulunda, liseyi Bursa Anadolu Lisesinden okudu. 2002 yılında liseden mezun olarak aynı yıl Sakarya Üniversitesi Bilgisayar Mühendisliği Bölümünü kazandı. 2006 yılında mezun olarak, Sakarya üniversitesi Bilgisayar ve Bilişim Mühendisliğinde yüksek lisansa başladı. 2007 yılında işe başladığı Teknolojix Bilişim Ürünleri A.Ş.'de sistem geliştirme uzmanı olarak görevine devam etmektedir.