T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GEN AĞI ÇIKARIM ALGORİTMALARI İÇİN EN UYGUN İLİŞKİ KESTİRİMCİLERİNİN BELİRLENMESİ

ZEYNEB KURT

DOKTORA TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ PROGRAMI

DANIŞMAN
PROF. DR. NİZAMETTİN AYDIN

İSTANBUL, 2013

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

GEN AĞI ÇIKARIM ALGORİTMALARI İÇİN EN UYGUN İLİŞKİ KESTİRİMCİLERİN BELİRLENMESİ

Zeyneb KURT tarafından hazırlanan tez çalışması 16.12.2013 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda **DOKTORA TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Prof. Dr. Nizamettin AYDIN

Yıldız Teknik Üniversitesi

Eş Danışman

Doç. Dr. Gökmen ALTAY

Bahçeşehir Üniversitesi

Jüri Üyeleri

Prof. Dr. Nizamettin AYDIN

Yıldız Teknik Üniversitesi _____

Doç. Dr. Duran ÜSTEK

İstanbul Üniversitesi _____

Yrd. Doç. Dr. Gökhan BİLGİN

Yıldız Teknik Üniversitesi _____

Prof. Dr. Zümray DOKUR ÖLMEZ

İstanbul Teknik Üniversitesi _____

Doç. Dr. Banu DİRİ

Yıldız Teknik Üniversitesi _____

ÖNSÖZ

Tez çalışmamı gerçekleştirirken benden hiçbir zaman desteklerini esirgemeyen, daima cesaretlendiren ve yönlendiren danışmanlarım Doç. Dr. Gökmen ALTAY'a ve Prof. Dr. Nizamettin AYDIN'a teşekkürlerimi sunarım. Ayrıca tez izleme komitemde bulunan hocalarım Doç. Dr. Duran ÜSTEK'e ve Yrd. Doç. Dr. Gökhan BİLGİN'e de değerli yorumları ve yönlendirmeleri için teşekkürlerimi sunarım.

Hayatımda her anımda maddi-manevi destekçilerim ve ilk öğretmenlerim olan anneme, babama; ayrıca eşime, kardeşlerime ve yakın arkadaşlarıma da teşekkürlerimi sunarım.

Aralık, 2013

Zeyneb KURT

İÇİNDEKİLER

Sayfa

KISALTMA LİSTESİ

ANOVA	Analysis of Variance
ARACNE	Accurate Cellular Networks
AUC-PR	Area under the Curve Precision-Recall
BGA	Bayesian Gen Ağı
BS	B-spline
BS2	Eğri Derecesi 2 Olan B-spline Kestirimcisi
BS3	Eğri Derecesi 3 Olan B-spline Kestirimcisi
C3NET	Conservative Causal Core network
CD	Copula Dönüşümü
CLR	Context Likelihood or Relatedness
CS	Chao-Shen
CSEF	EF Ayrıklaştırma Tekniği ile Kullanılan Chao-Shen (CS) Kestirimcisi
CSEG	EG Ayrıklaştırma Tekniği ile Kullanılan Chao-Shen (CS) Kestirimcisi
ÇD	Çapraz Doğrulama
ÇYK	Çekirdek Yoğunluk Kestirimcisi
DNA	Deoxyribonucleic Acid
EB	En büyük Benzerlik
EBK	En Büyük Bilgi Katsayısı
EF	Eş Frekans
EG	Eş Genişlik
EKKB	En küçük Kareler Karşılıklı Bilgi
EÜS	En iyi Üst Sınır
FP	False Positive (Hatalı Pozitif)
FN	False Negative (Hatalı Negatif)
GAÇ	Gen Ağı Çıkarımı
GDAÇ	Gen Düzenleyici Ağ Çıkarım
GGM	Grafiksel Gauss Modelleme
GNI	Gene Network Inference
HG	Hedef Genler
HHG	Heller, Heller, Gorfine
KB	Karşılıklı Bilgi
KEYK	K-En Yakın Komşuluk
KKB	Koşullu Karşılıklı Bilgi

KKB-1	Birinci Dereceden Koşullu Karşılıklı Bilgi
KPK	Kısmi Pearson Korelasyonu
KPK-1	Birinci Dereceden KPK
KPK-N	n'nci Dereceden KPK
MI	Mutual Information
mKor	Mesafe Korelasyonu
MM	Miller-Madow
MMEF	EF Ayrıklaştırma Tekniği ile Kullanılan Miller-Madow (MM) Kestirimcisi
MMEG	EG Ayrıklaştırma Tekniği ile Kullanılan Miller-Madow (MM) Kestirimcisi
MRNET	Maximum Relevance/Minimum Redundancy Network
NSB	Nemenman, Shafee, Bialek
OKH	Ortalama Karesel Hata
PKD	Pearson Korelasyon Katsayısı Değeri
RelNet	Relevance Networks
SKK	Spearman Korelasyon Katsayısı
TF	Transkripsiyon Faktörleri
TP	True Positive (Gerçek Pozitif)
VİE	Veri İşleme Eşitsizliği

ŞEKİL LİSTESİ

Sayfa

ÇİZELGE LİSTESİ

Sayfa

ÖZET

GEN AĞI ÇIKARIM ALGORİTMALARI İÇİN EN UYGUN İLİŞKİ KESTİRİMCİLERİNİN BELİRLENMESİ

Zeyneb KURT

Bilgisayar Mühendisliği Anabilim Dalı

Doktora Tezi

Tez Danışmanı: Prof. Dr. Nizamettin AYDIN

Eş Danışman: Doç. Dr. Gökmen ALTAY

Gen Ağı Çıkarımı (GAÇ) yöntemleri, biyoenformatik alanında önemli bir yere sahiptir. Hücrelerdeki moleküllerin birbirleri ile olan ilişkilerini incelememize olanak sağlayan GAÇ uygulamalarının kullanım alanları oldukça yaygındır. Örneğin; gen ikilileri veya protein gibi gen ürünü ikilileri arasındaki genom-genişliğindeki ilişkileri tespit etmek, ilaç üretiminde ana bileşeni seçmek, düzenleyen (regulator) ve düzenlenen genlerin etkileşimlerini gözlemlemek, vb çalışmalarda kullanılırlar.

GAÇ ya da gen ağı ters mühendisliği, gen ifade değerlerinden yola çıkılarak genlerin ilişkilerini tahmin etmek olarak düşünülebilir. Gen ifade değerlerini içeren biyolojik veri kümelerinin çok-boyutlu olması ve deneysel süreçlerden veya hesaplama işlemlerinden kaynaklanan gürültüler içermesi gibi sebeplerden Gen Ağı Çıkarımı (GAÇ) işlemi oldukça güçleşir. GAÇ algortimalarının en önemli aşaması, veri kümesindeki değişkenler arasındaki ilişki skorlarının kestirimidir. Bu aşama doğru bir şekilde gerçekleştirilemez ise, hangi GAÇ algoritması kullanılırsa kullanılsın, Gen Ağı Çıkarımı sonucu hatalı olur. Dolayısıyla, korelasyon ya da ilişki kestirimi aşaması, GAÇ algoritmalarının en önemli adımıdır. Ancak hâlihazırda mevcut GAÇ algoritmaları ile kullanılmasının uygun olacağı düşünülen, ortak kabul görmüş tek bir kestirimci yoktur.

Bu çalışmada GAÇ algoritmalarında kullanılması olası olan birçok farklı kestirimci incelenmiştir. Ayrıca kestirimciler çeşitli açılardan sınıflandırılarak bu alanda çalışan

araştırmacıların bilgisine sunulmuştur. Kestirimcilerin arasından, farklı GAÇ algoritmalarında kullanılmasının en uygun olacağı ve en iyi performansı vereceği düşünülen yöntemler belirlenmiştir. Yüksek performans sergileyeceği düşünülen kestirimciler, çeşitli GAÇ algoritmaları kullanılarak karşılaştırılmış ve değerlendirme sonuçları verilmiştir. Bunlara ilaveten bu çalışmada, ilişki kestirimcileri arasında belli parametrelere bağımlı olan yöntemlerin parametre seçimi incelenmiş, çeşitli öneriler sunulmuştur. Bu bağlamda literatürde bulunan ilişki-tabanlı, entropi-tabanlı ve doğrudan karşılıklı bilgi tabanlı 27 farklı kestirimci incelenmiş, aralarından en iyi performansı vereceği düşünülen 12 tanesi ve türevleri uygulanmak üzere seçilmiştir. Kestirimciler karşılaştırılırken daha sağlıklı değerlendirme yapabilmek amacıyla bir değil, üç farklı GAÇ algoritması kullanılmıştır. Bu algoritmalar Accurate Cellular Networks (ARACNE), Relevance Networks (RelNet) ve Conservative Causal Core Network (C3NET)'tir. Ayrıca, çalışmanın sonucunda kestirimcilerin gerçeklenmesini içeren bir R yazılım paketi hazırlanmıştır. Sonuç olarak bu çalışmanın, GAÇ algoritmaları ile çalışan araştırmacılar için kestirimcilerin hangi sınıfa dahil olduğunu ve hangi kestirimcinin hangi parametreler ile kullanılması gerektiğini söyleyen rehber niteliğinde bir çalışma olması hedeflenmiştir.

Anahtar Kelimeler: Gen ağı çıkarımı, ilişki kestirimcileri, entropi kestirimi, karşılıklı bilgi kestirimi

ABSTRACT

DETERMINING THE MOST SUITABLE CORRELATION ESTIMATORS FOR GENE NETWORK INFERENCE ALGORITHMS

Zeyneb KURT

Department of Computer Engineering

PhD. Thesis

Adviser: Prof. Dr. Nizamettin AYDIN

Co-Adviser: Assoc. Prof. Dr. Gökmen ALTAY

Gene network inference (GNI) algorithms have a significant role in bioinformatics research area. Those algorithms provide us to explore the vast amount of the interactions among the molecules in the cell. Application areas of GNI algorithms are very wide; such as discovering a genome-wide interactions and associations among the genes and gene-products (e.g. proteins); determining the main target of a drug in pharmacological studies; observing the interaction of the regulator and regulated genes, and so on.

GNI or in other words reverse engineering of gene networks is the process of predicting the interactions of the genes by using microarray gene expression values. GNI process is a challenging process because of the current very large-scale biological datasets and the noise caused by the experimental and computational processes. In almost all GNI algorithms the main process is to estimate the association scores among the variables of the dataset. If this step is not correctly fulfilled then the ultimate inference process becomes erroneous for whichever the GNI algorithm is used. Therefore, this is the most crucial process of any GNI algorithms. Nonetheless, there is no commonly accepted estimator to compute association scores that is used with the current GNI algorithms.

In this study, we investigate almost all of the available estimators that might be used in the GNI algorithms. Estimators are classified according to different point of views and presented to the researchers by this study. Among the reviewed estimators, we determine the most suitable and the best performing estimators for various GNI algorithms. We achieved this by comparing the inference performance of the estimators over different GNI algorithms, which are selected as representatives of many others. For that purpose, from the literature, we first reviewed correlation-based estimators and then entropy-based and finally direct MI estimator approaches to estimate Mutual Information; in the context whether they are used in genomics datasets or not. So far we reviewed 27 different estimators and identified 12 of the promising methods and their derivatives for effective usage in GNI. We expect this study to assist many researchers before using those estimators in their own GNI studies. We compared the estimators on three popular GNI algorithms; those are called Accurate Cellular Networks (ARACNE), Relevance Networks (RelNet) and Conservative Causal Core Network (C3NET). At the end of the study, we prepared an R software package which includes the determined estimators. Since we provide the most prominent estimators out of the study, any of the algorithm that wants to use our proposed estimators, needs to reassess the inference performance before replacing their estimators. Ultimately, this study aimed to be a reference guide for the usage of the estimators in GNI algorithms.

Key words: Gen network inference, correlation-based estimators, entropy-based estimators, mutual information estimators

BÖLÜM 1

GİRİŞ

Gen Ağı Çıkarım (GAÇ) algoritmaları biyoenformatik, genetik, farmakoloji, vb alanlarda önemli bir yere sahiptir ve sıklıkla kullanılırlar. Genler arasındaki ilişkiyi bir ağ şeklinde sunarak araştırmacıların görselleştirilen ağ bilgisinden faydalanmalarını sağlarlar. GAÇ algoritmalarının en temel adımı gen ikilileri arasındaki ilişki skorlarının tespit edilmesidir. Bu amaçla çeşitli etkileşim veya ilişki kestirimcisi yöntemler kullanılır. Bu çalışmada GAÇ algoritmaları ile kullanılması olası kestirimciler incelenmiş ve en iyi olduğu düşünülen yöntemler karşılaştırılarak değerlendirme yapılmıştır. Ayrıca, yöntemler çeşitli açılardan sınıflandırılmıştır. Kestirimcilerin bir kısmı için temel parametre seçimi incelenmiş ve bunlara dair öneriler getirilmiştir. Bunlara ilaveten, veri kümelerine uygulanan bir ön işlem olan Copula Dönüşümü'ne (CD) göre kestirimcilerin değerlendirilmeleri yapılmıştır. Bu tez çalışmasının nihayetinde, literatürden seçilen en ümit verici kestirimcilerin gerçeklenmesini içeren bir yazılım paketi hazırlanmış ve bu paket, bu alanda çalışan araştırmacıların kullanımına sunulmuştur. Kestirimcilerin çıkarım performansı değerlendirilirken Accurate Cellular Networks (ARACNE) [1], Relevance Networks (RelNet) [2] ve Conservative Causal Core Network (C3NET) [3] GAÇ algoritmaları kullanılmıştır.

Bu bölümde literatür özeti, tezin amacı ve tezin literatüre katkısı verilmiştir. Literatür özetinde, tez kapsamında gerçeklenen kestirimcilerin seçim sebepleri detaylı bir şekilde verilmiştir.

1.1 Literatür Özeti

İki rastgele değişken arasındaki ilişki veya etkileşim skoru, bu iki değişkenin birbirlerine olan bağımlılık derecelerini ölçer. Rastgele değişkenler arasındaki ilişki skoru kestirimcileri ekonomi [4]-[6], sinyal işleme [7]-[11], telekomunikasyon [12]-[13], astrofizik [14]-[15], meteoroloji [16], istatistik ve biyoenformatik [1]-[3], [17]-[21] gibi çeşitli çalışma alanlarında kullanılmaktadır.

Bu tez çalışmasında Gen Ağı Çıkarımı (GAÇ) algoritmalarında kullanılan ve kullanılması olası olan tüm ilişki kestirimcileri incelenmiştir. GAÇ algoritmaları biyoenformatik alanında önemli bir yere sahiptir. Genler veya protein gibi gen ürünleri arasındaki genom-genişliğindeki bağlantıların gösteriminde sıklıkla GAÇ algoritmaları kullanılmaktadır [1]-[3], [17]-[21]. GAÇ algoritmalarının örneğin farmakolojik çalışmalarda kullanımı, daha güvenli ve daha düşük maliyetli ilaç üretimine yardımcı olabilmektedir. Bu algoritmaların literatürde sıklıkla kullanıldığı uygulamalar:

- İlgilenilen genlerin fonksiyonlarının tespitine yardımcı olmak,
- Düzenleyen (regulating) ve düzenlenen (regulated) genleri tespit edebilmek,
- İlgilenilen hastalığa karşı üretilecek ilaçların etken maddesini tespit edebilmek, vb.

şeklinde özetlenebilir. GAÇ veya gen ağları ters mühendisliği uygulamaları, mikroçip (microarray) veri analizinden elde edilen gen ifade değerleri veri kümelerini kullanarak genlerin genom genişliğindeki bağlantılarını ortaya koyar. Bu uygulamalar, biyolojik veri kümelerinin büyük boyutlu ve gürültülü olması gibi sebeplerden olumsuz etkilenebilmektedir. Gen Ağı Çıkarımı sürecinde gen ikilileri arasındaki etkileşim skorlarının etkin ve doğru bir şekilde kestirilebilmesi gerekmektedir. Eğer bu adımda etkileşim skorları doğru bir şekilde tespit edilemezse, hangi GAÇ algoritması kullanılırsa kullanılsın ağ çıkarım performansı düşük ve çıkarım işlemi başarısız olur. Dolayısıyla etkileşim skoru kestirim işlemi, GAÇ algoritmalarındaki en önemli adımdır.

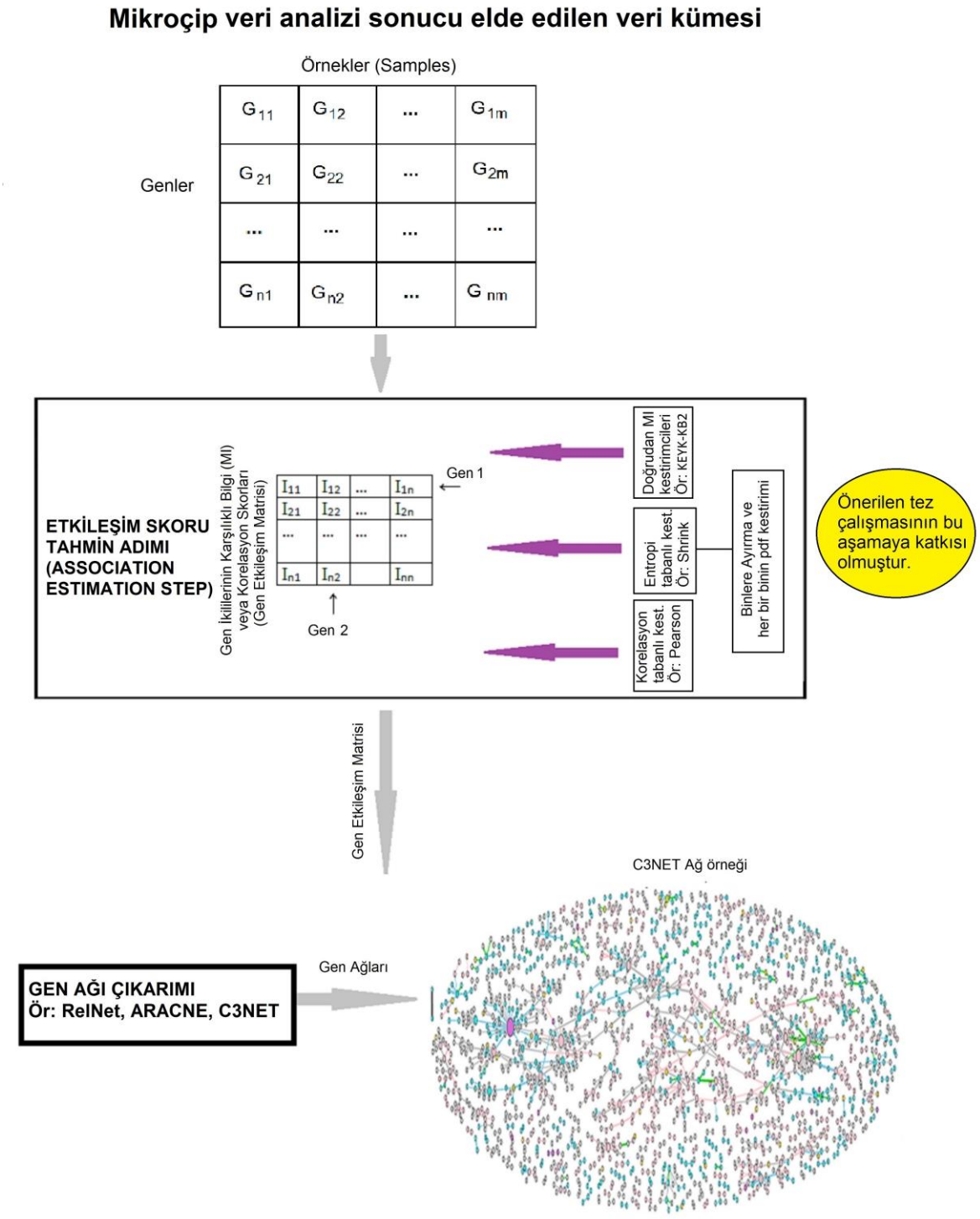
Şekil 1.1'de GAÇ uygulamalarının genel blok diyagramı verilmiştir. Öncelikle gen ifade değerlerini içeren veri kümesi girdi olarak alınır. Gen ifadelerini içeren veri kümesi bir matriste saklanmaktadır. Bu matristeki her bir satır bir gene, her bir sütun da bir örneğe karşılık gelmektedir. Gen ifade değerleri, DNA (Deoxyribonucleic acid) mikroçip

veri analizi deneylerinden yola çıkılarak elde edilir. Lazer ışığı ile uyarılan mikroçip'in her bir gözündeki (spot) renk değerleri görüntü işleme yöntemleri kullanılarak sayısal değerlere (gen ifade değerleri) dönüştürülür. Bu tez çalışmasında mikroçip veri analizi işlemi gerçeklenmemiş, literatürde sıklıkla kullanılan mevcut gen ifade veri kümeleri girdi olarak kullanılmıştır. Mikroçip veri analizi aşamasını GAÇ uygulamalarının ilk adımı olarak kabul edersek, ikinci adımda gen ifade değerlerini içeren matris kullanılarak tüm gen ikilileri arasındaki etkileşim skorları tahmin edilir. Bu aşama etkileşim kestirimi (association estimation) şeklinde isimlendirilir. Bu tez çalışmasında, etkileşim kestirimi aşaması detaylı bir şekilde incelenmekte ve kestirimciler için çeşitli öneriler getirilmektedir. Etkileşim skorları tahmin edilirken korelasyon tabanlı, entropi-tabanlı veya doğrudan karşılıklı bilgi bulan yöntemlerden herhangi biri kullanılabilir. Entropi-tabanlı yöntemler kullanılacaksa, öncelikle veri kümesine ayrıklaştırma işlemi uygulanmalıdır. İkinci adımın sonucunda kare bir matris olan gen etkileşim matrisi elde edilir. Bu matris son aşama olan GAÇ algoritmasına verilerek sonuç gen ağı elde edilir.

Bu çalışmada GAÇ algoritmalarından ziyade, bu algoritmaların temel adımı olan etkileşim tahmini aşamasına odaklanılmıştır. Bu bağlamda korelasyon tabanlı, entropi tabanlı veya doğrudan karşılıklı bilgi bulabilen olmak üzere 27 farklı etkileşim kestirimcisi incelenmiştir. Bilgimiz dâhilinde bu kadar çok kestirimcinin bir arada incelendiği başka bir çalışma bulunmamaktadır.

İlişki veya etkileşim skoru kestirimcilerinin incelendiği az sayıda çalışma mevcuttur [17], [18], [22]-[24]. Var olan çalışmalar da az sayıda kestirimciyi içermekte ve incelemektedir. Bu çalışmaların arasından sadece birkaç tanesi mikroçip veri analizinden elde edilen gen ifade veri kümelerini kullanarak kestirimcilerin karşılaştırılmasını gerçekleştirir [17], [18]. İncelenen kestirimcilerin literatüre dayanan karşılaştırmaları Çizelge 1.1'de verilmiştir. Bu çizelgede ilk kolon ve ilk satırdaki numaraların her biri bir kestirimciyi temsil etmektedir. Hangi numaranın hangi kestirimciye karşılık geldiği çizelgenin ikinci kolonunda görülmektedir. Çizelgede *i*'nci satır ve *j*'nci kolondaki hücrede ">" işareti bulunması, *i*'nci kestirimcinin *j*'nci kestirimciden daha yüksek performanslı ilişki skoru tahmini yapabildiği anlamına gelir. Bu hücrede "<" işareti bulunması *i*'nin *j*'den daha düşük performansla tahmin yapabildiğini gösterir. "≈" işareti ise iki kestirimcinin tahmin performanslarının yaklaşık

aynı olduğunu gösterir. Bu bilgiler ışığında, aşağıda Çizelge 1.1'den de faydalanarak literatür incelemesi ve değerlendirmesi detaylı bir şekilde verilmiştir. Literatür incelemesi verilirken birbirleri ile ilişkili ve çeşitli çalışmalarda bir arada değerlendirilmiş olan yöntemler alt başlıklarda incelenmiştir.



Şekil 1. 1 GAÇ uygulamalarının blok diyagramı

Çizelge 1. 1 Literatürden İncelenen 27 Kestirimcinin Karşılaştırılması

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
1	PKD							<	<	<		<	<	>*	>*			<	<	<		<		<	<	<	<	
2	Bayes1			>	<	<								>	>		<							<	<	<	<	
3	Bayes2		<		<	<								>	>		<							<	<	<	<	
4	Bayes3		>	>		<								>*4,<*3	>*4,<*3		<											<*4,>*3
5	Bayes4		>	>	>									>	>		<							<	<	<	<	<
6	Edgeworth							<						>*6					>		>*6							
7	EKKB	>					>											>				>						
8	KPK1	>								<								>						>*2				
9	KPKn	>							>									>						>				
10	Jackknife																						<					
11	SKK	>											>*	>*	>*			>		>					<	<	<	
12	Kendall	>																										
13	EB	<*	<	<*4,>*3	<	<*6						<*			<													<*4,>*3,>*
14	MM	<*	<	<*4,>*3	<							<*		>									<					<*4,>*3,>*
15	ANOVA	>																										
16	CS		>	>	>	>																						≈
17	BS	>							<	<								>	>*5			>	>	<				
18	ÇYK	>						<				<							<*5		<	>			<	<	<	
19	KEYK entr.	>																			<	<						
20	KEYK-KB1						<*6					<						>	>	>		<			<	<	<	
21	KEYK-KB2	>						<												>	>							
22	EÜS										>			>	>		<	<	<									
23	KKB1	>							<*2	<							>	>										
24	EBK	>										>						>	>		>							
25	mKor	>										>						>	>		>							
26	HHG	>										>						>	>		>							
27	Çekilme		>	>*4,<*3	>	>							>*4,<*3,<*3	>*4,<*3,<*		≈												

1.1.1 EB, MM, Pearson, Spearman, Çekilme (Shrinkage) ve Schürmann-Grassberger Kestirimcilerinin Analizi

Olsen vd. [17] ile Simoes ve Streib [18], çalışmalarında çeşitli kestirimcileri GAÇ uygulamalarındaki çıkarım performanslarına göre karşılaştırmışlardır. Ayrıca, veri kümesi ayrıklaştırma tekniklerinin çıkarım performansına etkilerini de incelemişlerdir. Ayrıklaştırma teknikleri bu tez çalışmasının Bölüm 3.2'sinde verilmiştir, detaylı bilgi için bu bölüm incelenebilir. [17]'deki çalışmada Eş Frekans (EF) ayrıklaştırma tekniğinin Eş Genişlik (EG) tekniğinden daha etkin olduğu ifade edilmiştir. [18]'de ise EG tekniği ile elde edilen sonuçların daha başarılı olduğu belirtilmiştir. Bu tez çalışması gerçeklenirken de, veri kümesine dair dağılım bilinmezken veri kümesini eşit veya benzer frekanslı hücrelere bölmenin, dolayısıyla EF tekniğini kullanmanın, performansa daha olumlu etkisinin olabileceği düşünülmüştür. Bununla birlikte, bu tez çalışmasında da ayrıklaştırma tekniklerinin performansa etkisini gözlemleyebilmek için her iki teknik de kullanılarak değerlendirme yapılmıştır.

[17]'deki çalışmada değerlendirmesi yapılan yöntemler Pearson korelasyon katsayısı değeri (PKD), Spearman korelasyon katsayısı (SKK), En büyük Benzerlik (EB, Maximum Likelihood-ML), Miller-Madow (MM), James-Stein çekilme (shrinkage) kestirimcileridir. [17]'de beş kestirimcinin çıkarım performansı, üç GAÇ algoritması kullanılarak değerlendirilmiştir. Kullanılan GAÇ algoritmaları: Accurate Cellular Networks (ARACNE [1]), Context Likelihood or Relatedness (CLR [25]), ve Maximum Relevance/Minimum Redundancy Network (MRNET [26]) şeklindedir. [17]'de, SynTReN [27] isimli simulatör kullanılarak 12 sentetik veri kümesi üretilmiş, veri kümelerinin bir kısmına yapay gürültü eklenmiştir. Veri kümelerindeki gen sayıları {100 ,200, 300} değerlerinden birini, örnek sayıları ise {50, 100, 200, 300} değerlerinden birini almıştır. MRNET GAÇ algoritması ile SKK kestirimci kombinasyonu ve CLR GAÇ algoritması ile PKD kestirimci kombinasyonunun en iyi çıkarım performansını verdiğini belirtmişlerdir [17]. Gürültüsüz veri kümesinde MRNET ve SKK kombinasyonu, CLR ve PKD kombinasyonundan daha başarılı olduğu ve ilk kombinasyonun ikincisinden daha düşük sapmaya (bias) sahip olduğu belirtilmiştir. İkinci kombinasyonun ise gürültülü veride daha düşük varyanta (variant) sahip olduğu gözlendiği için daha sağlıklı (robust) sonuç verdiği öne sürülmüştür [17]. Gürültülü ve eksik değerlerin olduğu veri kümeleri

kullanıldığında PKD ve SKK kestirimcilerinin en yüksek F-skor sonuçlarını verdiği, eksik değerler varken PKD'nin SKK'dan daha başarılı olduğu, verinin eksiksiz ama gürültülü olduğu durumda ise SKK'nın PKD'den daha başarılı olduğu belirtilmiştir. Gürültüsüz veri kümeleri için ise F-skor cinsinden en iyi performansı, EF ayrıklaştırma tekniği ile kullanılan MM ve EB kestirimcilerinin MRNET ve ARACNE GAÇ algoritmaları ile verdiği belirtilmiştir [17]. Sentetik veri kümeleri ile yapılan deney senaryolarının birçoğu için korelasyon tabanlı kestirimciler olan PKD ve SKK yöntemlerinin en iyi sonuçları verdiği gözlendiği için rastgele değişkenlerin (genlerin) ilişkilerinin lineer olmama ve monoton olmama durumlarının ihmal edilebileceği öne sürülmüş, etkin ilişki skoru kestirimi için lineer ve monoton ilişkileri hesaba katmanın yeterli olduğu iddia edilmiştir [17]. Bu kestirimciler bir R [26] yazılım paketinde sunulmuş, Kendall Tau korelasyon katsayısı yöntemi de bu pakette bulunmaktadır ancak [17]'de bu yöntemle ilgili bir sonuç bildirilmemiştir. Sonuç olarak veri kümesi gürültüsüz iken kestirimcilerin en iyiden kötüye doğru performans sıralaması: MM > EB > SKK > PKD > James-Stein çekilme şeklindedir [17]. Veri kümesinin gürültülü ve tam olması durumunda başarım sıralaması: SKK > PKD > MM > EB > James-Stein çekilme şeklindedir [17]. Çizelge 1.1'de ismi geçen kestirimcilerin belirttiği hücrede bulunan "*" işareti veri kümesindeki gürültü mevcudiyetine göre başarımın değiştiğini simgelemektedir. *Biyolojik veri kümelerinin gürültülü olduğu hesaba katıldığında SKK ve PKD yöntemlerinin ümit verici olduğu görüldüğü için bu tez çalışmasında incelenecek kestirimcilere bu yöntemler dâhil edilmiştir.*

[18]'deki çalışmada da Karşılıklı Bilgi (KB) kestirimcileri ve veri ayrıklaştırma tekniklerinin GAÇ algoritmalarının başarımına etkileri incelenmiştir. MM, EB, Schürmann-Grassberger (Bayesian 3) ve James-Stein çekilme yöntemleri karşılaştırılmıştır. Ayrıklaştırma tekniği olarak EF, EG, ve global EG yaklaşımları kullanılmıştır. Ayrıklaştırma yöntemini değiştirmenin performans üzerinde kestirimcileri değiştirmekten daha çok etkili olduğunu öne sürmüşlerdir. MM kestirimcisinin EG ve global EG ayrıklaştırma teknikleri ile kullanıldığında en iyi çıkarım performansını sergilediği belirtilmiştir. EB, Schürmann-Grassberger ve James-Stein çekilme yöntemlerinin sırayla MM'yi takip ettiği belirtilmiştir [18]. EB ve MM ampirik (empirical) tabanlı yöntemlerdir. Veri kümesindeki her bir hücrenin olasılık dağılımı,

hücrelerdeki veri örneklerinin adedinden elde edilir. MM sapmayı (bias) hesaba katan tek yöntemdir, böylelikle sapmadan kaynaklı hatayı düzeltebilmektedir. EB ve MM'den farklı olarak James-Stein çekilme ve Schürmann-Grassberger yöntemleri her bir hücredeki verilerin olasılık tahminini düzeltmeye çalışmaktadır. Bu olasılık dağılımları kullanılarak bireysel ve birleşik (joint) entropiler, buradan da KB skoru elde edilir. [18]'deki çalışmanın deneysel sonuçlarına göre dört KB skor kestirimcisinin performans sıralaması MM > ML > Schürmann-Grassberger (Bayes 3) > James-Stein çekilme şeklindedir. Bu sonuçlar [17]'deki çalışmanın sonuçları ile uyumludur. [18]'de veri kümesinin gürültü durumuna göre inceleme yapılmamıştır. Yine sentetik veri kümeleri kullanılmış, örnek sayısı {50, 100, 200, 500, 1000} değerlerinden biri olarak seçilmiştir [18]. *[17] ve [18]'in her ikisinde de MM kestirimcisi ümit verici olarak görüldüğü için bu tez çalışmasında da MM incelenmek üzere seçilmiştir*. Son olarak [18]'deki kestirimcilerin tümü verinin tek bir olasılık yoğunluğuna sahip olduğunu varsaymaktadır, ancak gerçek biyolojik veri kümeleri için bu durum gerçekçi değildir.

1.1.2 Çekilme, EB, MM, Bayes 1, Bayes 2, Bayes 3, Bayes 4, Chao-Shen ve NSB Kestirimcilerinin Analizi

Hausser ve Strimmer GAÇ algoritmalarında genler arasındaki etkileşim skorlarını bulmak için yeni bir tür James-Stein çekilme kestirimcisi önermişlerdir [23]. Deneylerinde gerçek biyolojik veri kümesi kullanmadan önce çeşitli veri üretimi ve örnekleme senaryoları ile yapay veri kümeleri üretmişlerdir. Örnek sayılarını {10, 30, 100, 300, 1000, 3000, 10000} değerlerinden biri olarak seçmişlerdir. Yapay veri kümelerini kullanarak 9 farklı kestirimciyi karşılaştırmışlardır. Daha sonra en iyi sonuç verdiğini öne sürdükleri James-Stein çekilme yöntemini kullanarak 9 örnek ve 102 genden oluşan genomik bir veri kümesi için gen etkileşim ağını elde etmişlerdir. [23]'te kullanılan 9 kestirimci: James-Stein çekilme, Jeffreys öncülü (Bayesian 1), Bayes-Laplace öncülü (Bayesian 2), Schürmann-Grassberger (Perks öncülü veya Bayesian 3), Minimax öncülü (Bayesian 4), EB, MM, Chao-Shen (CS) ve NSB (Nemenman, Shafee, Bialek) kestirimcileri şeklindedir. Bu 9 kestirimci Ortalama Karesel Hata (OKH) ve entropi tahmininin sapmasına göre karşılaştırılmış, James-Stein çekilme yönteminin diğerlerine fark attığı belirtilmiştir [23]. Sonuçları inceleyecek olursak [23]:

- Örnek sayısı fazla iken tüm 9 kestirimci daha başarılı tahminler yapmaktadır.
- NSB, CS ve çekilme (shrinkage) kestirimcileri benzer performanslar sergilemişler ve en tutarlı ve doğru tahmini bu yöntemlerin yaptığı belirtilmiştir. Tüm senaryolar ve örnek sayıları için bu yöntemlerin en düşük OKH değerlerini verdiği gözlenmiştir [23].
- NSB'nin CS ve çekilme yöntemlerine göre hesaplama karmaşıklığının daha fazla olduğu ve deneylerde örneğin çekilme yönteminin NSB'den 1000 kat daha hızlı olduğu belirtilmiştir. Dolayısıyla CS ve çekilme yaklaşımları NSB'ye nazaran daha çok tercih edilebilirdir [23].
- Birçok senaryo için CS'nin neredeyse hiç sapmasının olmadığı (unbiased) gözlenmiştir [23]. Ayrıca CS'nin gerçeklenmesi diğerlerinden kolay olduğu için GAÇ çalışmalarında kullanıma daha uygun olduğu düşünülerek CS bu tez çalışmasına dâhil edilmiştir.
- EB ve MM'in büyük örnek sayıları ile dahi pekiyi sonuç vermediği öne sürülmüştür [23].
- Bayes tabanlı (Bayesian) yöntemlerin EB'den daha iyi ya da daha kötü çıkabildiği, Bayesian yöntemlerde başarımın öncül olasılık seçimine ve örneklemeye bağlı olduğu belirtilmiştir [23].
- Bayes-Laplace (Bayesian 2) ve Jeffreys (Bayesian 1) öncülü yöntemlerinin EB'den biraz daha iyi performans sergilediği gözlenmiştir [23].
- Minimax ve Perks Bayesian kestirimcilerin birçok senaryo için EB, MM, Bayesian 1 ve Bayesian 2 yöntemlerinden daha başarılı olduğu belirtilmiştir [23].

Sonuç olarak [23]'teki çalışma için başarım sıralaması iyiden kötüye doğru: James-Stein çekilme (shrinkage) ≈ Chao-Shen (CS) ≈ NSB > Minimax öncülü (Bayes 4) > Schürmann-Grassberger (Perks öncülü-Bayes 3) > Jeffreys öncülü (Bayes 1) > Bayes-Laplace öncülü (Bayes 2) > MM > EB şeklinde verilebilir. NSB hesaplama yükünün büyüklüğünden dolayı tercih edilmemiştir. *CS en iyi performans veren yöntemlerden birisi olduğu için ve gerçeklenmesi kolay olduğu için bu tez çalışmasında incelenen kestirimcilere dâhil edilmiştir.* Ancak çekilme (shrinkage) ve Schürmann-Grassberger (Bayes 3)

yöntemlerinin EB ve MM yöntemlerinden daha başarılı olduğunu söyleyen sonuçlar [17] ve [18]'in sonuçları ile çelişmektedir. [17] ve [18]'de EB ve MM'nin çekilme yönteminden her koşulda daha iyi performans sergilediği gözlenmiştir. Bize göre, bu çalışmaların sonuçlarındaki farklılıklar şunlardan kaynaklanıyor olabilir: [17] ve [18]'deki sonuçlar, doğrudan KB tahmin başarımına dayanmamakta, gerçek gen ağı ile GAÇ algoritmasının çıkardığı gen ağının farkından elde edilen F-skor başarımına dayanmaktadır. [23]'te kestirimciler değerlendirilirken yapay veri kümeleri için gerçek KB skorları ile tahmin edilen KB skorları arasındaki farka bakılmıştır. Ayrıca bu çalışmalarda kullanılan veri kümeleri birbirlerinden farklıdır. Çizelge 1.1'deki "*3" işareti çalışma [18]'e göre olan performans değerlendirmesini, "*4" işareti ise [23]'e göre olan performans değerlendirmesini ifade etmektedir. Yukarıda açıklaması verildiği üzere, "*3" ve "*4" birbirinin tersidir.

1.1.3 EB, MM, Jackknife ve EÜS Kestirimcilerinin Analizi

Paninski çalşmasında EB, MM, Jackknife, En iyi Üst Sınır (EÜS, Best Upper Bound-BUB) kestirimcilerini, merkezi limit teoremi tutarlılığı, sapma (bias) ve varyans gibi çeşitli istatistikî açılardan incelemiştir [28]. Kestirimciler doğaları gereği entropi veya KB kestirimi esnasında tahmin sapmasını ve varyansını minimize etmeye çalışırlar, [28]'de de kestirimcilerin bu değerleri minimize etmedeki başarıları incelenmiştir. Kestirimcilerin $N << m$, $N >> m$ ve $N \sim m$ (N: örnek sayısı, m: hücre sayısı olmak üzere) durumları için güven aralıkları incelemesi de yapılmıştır. MM'nin bir türevi olan EÜS yönteminin genel olarak diğerlerinden daha başarılı olduğu belirtilmiş, MM'nin başarısız olduğu $N \sim m$ durumunda EÜS'nin daha başarılı olduğu öne sürülmüştür [28]. Ancak gen ikilileri arasındaki KB skoru bulunurken $N \sim m$ durumu değil, $N >> m$ durumu geçerlidir. *Bu nedenle bu tez çalışmasına EÜS yöntemini dâhil etmenin gerekli olmadığı, MM'yi seçmenin yeterli olduğu düşünülmüştür*. [28]'daki çalışmanın deneylerinde EÜS yöntemi sinirbilimsel (neuroscientific) veri kümesi ile de kullanılmıştır.

1.1.4 B-spline, ÇYK ve EÜS Kestirimcilerinin Analizi

Daub vd. GAÇ algoritmalarında kullanılmak üzere B-spline (BS) yöntemini önermişler, bu yöntemin başarımının eğri derecesine göre değiştiğini belirtmişlerdir [19]. B-spline

kestirimcisini Çekirdek Yoğunluk Kestirimcisi (ÇYK, Kernel Density Estimator-KDE) ve EÜS kestirimcisi ile karşılaştırmışlar, deneylerde 2 büyük ölçekli gen ifade veri kümesi kullanmışlardır. KB kestirim performansına dayalı bir değerlendirme yapılmıştır. İlk veri kümesinde 5345 gen ve 300 örnek, ikincisinde 22608 gen ve 102 örnek bulunmaktadır [19]. Performans sıralaması: B-spline (eğri derecesi=3) > ÇYK > B-spline (eğri derecesi=1) > EÜS şeklinde gözlenmiştir. Çizelge 1.1'deki "*5" işareti başarım sıralamasının eğri derecesi ile değiştiğini ifade etmektedir. [19]'da ÇYK'nın hesaplama karmaşıklığının B-spline'dan 10^4 kez daha fazla olduğu belirtilmiştir [19]. Dolayısıyla ÇYK'nın karmaşıklığı veri kümesine sınır konulmasına sebep olabilir. *Yine de B-spline kestirimcisi ile beraber ÇYK yöntemi de bu tez çalışmasında kullanılmak üzere seçilmiştir.* [19]'da ÇYK'nın hangi parametrelerle kullanıldığı belirtilmemiştir, bu çalışmada en uygun parametre seçimi ile ilgili incelemeler de yapılmıştır. Ayrıca B-spline'daki en uygun eğri derecesi seçimi de bu tez çalışmasında incelenmiştir [29].

1.1.5 B-spline, Entropi-tabanlı KEYK, KEYK-KB$^{(1)}$, KEYK-KB$^{(2)}$, PKD ve ÇYK Kestirimcilerinin Analizi

Kraskov vd. K-en Yakın Komşuluk (KEYK) yaklaşımına dayanan ve doğrudan KB skoru kestiren iki yöntem önermişlerdir, bunları KEYK-KB$^{(1)}$ ve KEYK-KB$^{(2)}$ şeklinde isimlendirmişlerdir [30]. Bu iki kestirimciyi entropi-tabanlı KEYK yöntemi ile karşılaştırmışlardır. Entropi-tabanlı yöntem, diğerlerinden farklı olarak KB skorunu dolaylı yoldan elde eder. Deneylerde maya'ya (yeast) ait, 6000 gen ve 300 örnek içeren gen ifade veri kümesi kullanılmıştır. Gen ikilileri için bireysel ve birleşik entropilerin tahmin hatasından kaynaklanan sapmanın KB$^{(1)}$ ve KB$^{(2)}$ yöntemleri için söz konusu olmadığını ve bu yöntemlerin daha başarılı tahminler yaptığını belirtmişlerdir [30]. [30]'da Performans sıralaması KEYK-KB$^{(1)}$ > KEYK-KB$^{(2)}$ > Entropi-tabanlı KEYK şeklindedir. Numata vd. de KEYK-KB$^{(2)}$ kestirimcisi ile entropi-tabanlı KEYK ve PKD yöntemlerini yapay veri kümelerindeki tahmin sapmaları açısından karşılaştırmıştır [31]. Deneyler esnasında öncelikle basit fonksiyonel ilişkilerin olduğu küçük veri kümeleri kullanılmış, daha sonra 43 örnek ve 181 rastgele değişken içeren metabolomik bir veri kümesi kullanılmıştır. Üç yöntem arasından en iyi performansı ve en düşük sapmayı doğrudan KB skoru kestirebilen KEYK-KB$^{(2)}$ kestirimcisinin verdiği

belirtilmiştir [31]. Dolayısıyla [31]'e göre başarım sıralaması: KEYK-KB$^{(2)}$ > entropi-tabanlı KEYK > PKD şeklindedir.

Papana vd. da çeşitli karmaşıklıktaki zaman serilerinden üretilen simule edilmiş veri kümelerini kullanarak ÇYK ve KEYK yöntemlerini karşılaştırmışlardır [32]. KEYK'ın daha istikrarlı (stable), yönteme özel parametre seçiminden daha az etkilenir olduğunu ve hesaplama yükünün daha az olduğunu belirtmişler, KEYK'in ÇYK'den daha tercih edilir olduğunu öne sürmüşlerdir [32]. Dolayısıyla, *bu tez çalışmasında incelenen kestirimciler arasına [30]-[32] çalışmalarında başarılı olduğu belirtilen KEYK-KB$^{(2)}$ yöntemini de dâhil ettik.*

1.1.6 PKD, KPK-1, KPK-n, B-spline ve KKB-1 Kestirimcilerinin Analizi

Bu tez çalışmasında, koşullu ilişkileri kullanarak doğrudan olmayan (indirect) bağımlılıkları eleyen kestirimciler de incelenmiştir. de la Fuente vd. üçüncü derece Kısmi Pearson Korelasyon (KPK) katsayısını ve daha küçük dereceden KPK'ları incelemiştir [33]. [33]'teki çalışmanın amacı dolaylı bağımlılıkları eleyerek rastgele değişkenler arasındaki korelasyon skorlarını doğru bir şekilde elde edebilmektir. 781 adet genden oluşan bir veri kümesi kullanarak KPK kestirimcilerini karşılaştırmışlardır. İkinci ve üçüncü dereceden KPK'ların sonuçlarının birbirine yakın olduğu ve güvenli ağ çıkarımı için ikinci dereceden KPK seçiminin uygun ve yeterli olduğu belirtilmiştir [33]. Çakır vd. da PKD, birinci dereceden KPK (KPK-1), n'nci dereceden KPK (KPK-n) katsayıları, B-spline, B-spline'a dayanan birinci dereceden Koşullu Karşılıklı Bilgi (KKB-1) kestirimcilerini karşılaştırmıştır [34]. Bu beş kestirimci iki gerçek mikroçip veri kümesi (E.Coli bakterisi ve S. Cerevisiae mayası) kullanılarak ve üç farklı koşul değişikliğine (enzimatik, içsel ve çevresel değişikliklere) göre karşılaştırılmıştır [34]. Çalışmada metabolom verisi kullanılarak metabolik ağ çıkarımı yapılmıştır. Kendi çalışmalarından önce literatürde metabolom verisi ile detaylı bir çalışma yapılmadığını öne sürmüşlerdir [34]. Çalışmadaki deneysel sonuçlara göre:

- Çevresel değişiklik koşulu dışında her durumda n'nci dereceden KPK (KPK-n) en iyi sonuçları vermiştir. Buradaki *n*: gen ya da rastgele değişken adedini ifade etmektedir.

- Çevresel değişiklik koşulu altında, KKB-1 yöntemi en iyi sonucu vermiştir.
- Koşulsuz skorlar, bir başka deyişle PKD ve B-spline KB kestirimcileri en kötü sonuçları vermiştir.
- Kullanılan veri kümelerinde doğrusal olmayan (non-linear) ilişkilerin bulunmadığını, çünkü doğrusal olan (PKD, KPK-1, KPK-n) ve olmayan (B-spline, KKB-1) kestirimcilerin sonuçlarının benzer olduğunu belirtmişlerdir.
- Koşullu yöntemler (KPK-1, KPK-n ve KKB-1) dolaylı ve dolayısıyla hatalı bağlantıları ortadan kaldırdıkları için çıkarım performanslarının yüksek olduğu belirtilmiştir. (Dolaylı bağlantılar ile ilgili bilgiler bu tez çalışmasında Bölüm 3.1'de mevcuttur)
- KPK-1 ve KKB-1 yöntemlerinin koşullu olmayan karşılıklarına (PKD ve B-spline) göre önemli oranda daha başarılı sonuçlar verdiği belirtilmiştir. KPK-n'den sonra birçok senaryo için KPK-1 ikinci, KKB-1 üçüncü; birkaç senaryo için ise KKB-1 ikinci, KPK-1 üçüncü iyi kestirimci olarak gözlenmiştir.

Sonuç olarak [34]'e göre kullanılan veri kümeleri için koşul değişikliklerine göre inceleme yapıldığında genel olarak başarım sıralaması: KPK-n > KPK-1 > KKB-1 > B-spline > PKD şeklindedir. Ancak yukarıda belirtildiği gibi senaryoların birkaç tanesi için ikinci (KPK-1) ve üçüncü kestirimci (KKB-1) yer değiştirebilmektedir. *Bu sonuçlara göre, bu tez çalışmasında incelenecek kestirimciler arasına KPK-n de eklenmiştir.* "*2" işareti bu durumu göstermektedir.

1.1.7 EBK, PKD, SKK, KEYK, ÇYK, mKor ve HHG Kestirimcilerinin Analizi

Reshef vd. En Büyük Bilgi Katsayısı (EBK, Maximal Information Coefficient-MIC) yöntemini önerdikleri çalışmalarında, bu yöntemi PKD, SKK, ÇYK ve [30]'da önerilen KEYK kestirimcisi ile karşılaştırmışlardır [24]. Bunların arasından EBK'nin en kararlı ve güvenilir yöntem olduğunu belirtmişlerdir. Deneylerinin sonuçlarına göre başarım sıralaması: EBK > SKK > KEYK > ÇYK > PKD şeklindedir. Deneylerde dört farklı gerçek veri kümesi ve çeşitli yapay veri kümeleri kullanmışlardır. Gerçek veriler: sağlık, beysbol, genomik ve büyük-ölçekli insan mikrobiyota veri kümeleridir. Yapay veri kümelerini de çeşitli fonksiyonel ilişkiler (doğrusal, ikinci dereceden, üstel, vb fonksiyonlar) ve farklı gürültü seviyeleri kullanarak üretmişlerdir [24]. [24]'teki çalışma üzerinde çeşitli eleştiri

ve yorumlar bulunmaktadır [35]. Bunların birçoğu mesafe korelasyonu (mKor, distance correlation-dCor) ve HHG (Heller, Heller, Gorfine) kestirimcilerinin EBK'dan daha başarılı olduğunu öne sürmüştür.

Szekely vd. tarafından önerilen mKor kestirimcisi verilen iki rastgele değişken arasında bağımsızlık olup olmadığını kontrol eder [36]. [36]'da genomik veri kümesi kullanılmamış, Monte Carlo simulasyonundan üretilen yapay veri kümeleri kullanılmıştır. GAÇ algoritmalarının bir kısmında etkileşim skorları elde edildikten sonra birbirinden bağımsız olduğu düşünülen genler arasındaki bağlantılar kopartılır. Bize göre bu ağı budama işlemi mKor metriği ile gerçekleştirilebilir.

Heller vd. tarafından önerilen HHG yöntemi de rastgele vektörler arasında bağımsızlık olup olmadığını kontrol eder [37]. Çalışma [37]'nin amacı tutarlı ve çok-boyutlu yeni bir bağımsızlık test metriği önermektir. HHG, simüle edilmiş veri kümeleri kullanılarak EBK ve mKor metrikleri ile karşılaştırılmıştır. Genomik veri kümesi deneylerinde kullanılmamıştır [37]. [24]'te dengelilik (equitability) özelliğinin güç (power) özelliğinden daha önemli olduğu, EBK'nin da dengeli bir yöntem olduğu belirtilmiştir. Ancak [37]'de dengelilik özelliğinin değişkenler arasındaki ilişkiyi tespit etmekte bir işe yaramadığı, yarasa bile EBK'nın sadece veri kümesi gürültüsüz iken dengeli olabildiği belirtilmiştir. Veri kümesinin gürültüsüz olması ise gerçekçi olmayan bir durumdur. Ayrıca EBK'nın sadece 1-boyutlu rastgele değişkenler için kullanılabilir olduğu, mKor ve HHG'nin ise çok-boyutlu rastgele vektörler için de kullanılabildiği öne sürülmüştür [35], [37].

Bunlara ilaveten, Tibshirani ve Simon da farklı fonksiyonel ilişkileri olan yapay veri kümeleri kullanarak EBK, PKD ve mKor'u karşılaştırmışlardır [35]. mKor'un EBK'dan daha güçlü olduğu, bazı durumlarda PKD'nin bile EBK'dan daha güçlü olabildiği öne sürülmüştür [35]. [37]'deki gibi bir yöntemin gücü (power) düşükse dengelilik özelliğinin önemli olmadığı; ayrıca mKor'un EBK'dan daha güçlü olmasının yanı sıra hesaplama karmaşıklığının daha az olduğu belirtilmiştir.

EBK'yi önerenler [35] ve [37]'deki eleştirilere cevap vermişlerdir [35]. Buna göre EBK'nın büyük boyutlu veri kümeleri için de kullanılabilir olduğunu, örneğinde kullandıkları veri setlerinden birinin 7000 değişkenli olduğunu belirtmişlerdir. Ayrıca

mKor ve HHG'nin EBK'dan daha güçlü olabileceğini kabul etmişlerdir ancak bu iki yöntemin değişkenler arasında sadece istatistiksel bir ilişki bulunup bulunmadığını kontrol edebildiğini belirtmişlerdir. Ancak EBK'nın ilişki skoru da verebildiğini öne sürmüşlerdir. Ancak mKor [36] ve HHG [37] çalışmalarından görüldüğü kadarıyla sadece bağımlılık olup olmadığı değil, bu bağımlılığa dair skorlar da elde edilebilmektedir.

Sonuç olarak literatürde rapor edilen deneysel sonuçlara göre performans sıralaması: HHG > mKor > EBK > SKK > KEYK > ÇYK > PKD şeklinde düşünülebilir. *Bu tez çalışmasında HHG ilişki skoru kestirimcisi de incelenecek ve gerçeklenecek kestirimcilere dâhil edilmiştir.*

1.1.8 GAÇ Uygulamaları için bir ANOVA Kestirimcisinin Analizi

Küffner vd. spesifik bir GAÇ uygulaması olan Gen Düzenleyici (Regulatory) Ağ Çıkarım (GDAÇ) algoritmaları için ANOVA'ya (Analysis of variance) dayanan bir ilişki skoru kestirimcisi önermişlerdir [38]. ANOVA'nın PKD'den daha başarılı bir metrik olduğunu öne sürmüşlerdir. Ancak bu çalışma, Transkripsiyon Faktörleri (TF-Transcription Factors) ile Hedef Genler (HG, Target Genes-TG) arasındaki ilişkiyi içeren ve ağda sadece TF'leri düzenleyici olarak kabul eden özel bir GAÇ uygulamasıdır [38]. Deneylerinde beş farklı büyük ölçekli veri kümesi kullanılmıştır. Veri kümelerinden yapay veriler, E.Coli bakterisi ve S.Cerevisiae mayasını içeren üç tanesi DREAM5'ten [39], [40] alınmış, yine E.Coli bakterisi ve S.Cerevisiae mayasını içeren ikisi de M3D [41] veritabanından alınmıştır. Bu beş veri kümesindeki gen sayıları ve örnek sayıları sırasıyla {1643, 4511, 4297, 5950, 6572} ve {805, 805, 907, 536, 904} değerlerinden birini almakatadır. ANOVA yöntemi, fazladan TF ve HG'lerin bilgisini gerektiren özel bir GAÇ türü (GDAÇ) olduğu için bu tez çalışmasına uygulanabilir görülmemiştir ve gerçeklenecek kestirimciler arasına dâhil edilmemiştir.

1.1.9 PKD, Edgeworth, EB, KEYK, ÇYK ve EKKB Kestirimcilerinin Analizi

Hulle, Edgeworth açılımını kullanarak çok-boyutlu entropi hesaplamasını önermiştir [42]. Bu yöntemi hem tek-boyutlu hem çok-boyutlu veri kümeleri için KB tahmini başarısına göre KEYK yöntemi ile karşılaştırmıştır. Verinin sahip olduğu dağılım Gauss iken, çok-boyutlu veri kümeleri için Edgeworth'un KEYK'dan daha iyi KB tahmini

yapabildiğini, ancak veri modeli Gaussian'a yakın değilken Edgeworth'un pek başarılı bir tahmin yapamadığını belirtmiştir. Ayrıca KEYK'nın hesaplama karmaşıklığının Edgeworth'unkinden daha fazla olduğunu öne sürmüştür. Buna ilaveten, Edgeworth yöntemi KB tahmini için standartlaştırılmış kümülantlar kullandığı için entropi tahmin sapmasının sıfıra yaklaştığını iddia etmiştir [42].

Suzuki vd. da Edgeworth yöntemini EB kestirimcisi ile karşılaştırmış, verinin gerçek dağılımı Gauss'a yakınken Edgeworth'un daha iyi bir tahmin yapabildiğini belirtmişlerdir [43]. Ancak verinin dağılımı Gauss'a yakın değilken Edgeworth'un EB'den daha düşük performans sergilediğini gözlemlemişler. Dolayısıyla, Edgeworth'un başarımı ile veri kümesinin normal dağılıma yakınlığı arasındaki ilişki bu çalışmada da aynı şekilde gözlenmiştir [43]. Sonuç olarak, veri kümesindeki dağılım Gauss'a yakın ise performans sıralaması Edgeworth > KEYK ve Edgeworth > EB şeklindedir; değilse sıralama Edgeworth < KEYK ve Edgeworth < EB şeklindedir. Çizelge 1.1'deki "*6" işareti, sıralamanın veri kümesi dağılımının normal dağılıma yakınlığına göre değiştiğini göstermektedir.

Suzuki vd. bir başka çalışmalarında gen ikilileri arasındaki etkileşim skorunu bulmak için En küçük Kareler Karşılıklı Bilgi (EKKB, Least Squares Mutual Information-LSMI) yöntemini önermişlerdir [44]. Bu yöntemi PKD, KEYK, ÇYK ve Edgeworth yöntemleri ile karşılaştırmışlardır. KEYK'nın ÇYK'dan daha başarılı olduğunu ancak KEYK'taki komşu sayısı olan *k* parametresini uygun seçmek gerektiğini, bunun bir sorun olduğunu belirtmişlerdir. Deneylerinde 2 mikroçip veri kümesi kullanmışlardır [44]. Her iki veri kümesinde de kayıp değerler yerine ortalama ifade değerleri konulmuştur. EKKB yönteminin diğer yöntemlere göre üç avantajının olduğunu belirtmişlerdir [44]:

- EKKB ÇYK'dan farklı olarak yoğunluk tahmini ile uğraşmak zorunda değildir.
- EKKB'da model seçimine olanak sağlar ancak KEYK gibi yöntemler bunu sağlayamaz.
- EKKB'de veri kümesinin dağılımı ile ilgili herhangi bir varsayım yapılmasına gerek yoktur. Ancak Edgeworth yönteminde veri kümesinin Gauss dağılıma yakın olması gerekmektedir ki başarılı bir şekilde KB kestirimi yapılabilsin.

Sonuç olarak çalışma [44] de göz önüne alındığında, başarıma dair EKKB > ÇYK, EKKB > KEYK, EKKB > Edgeworth sıralamaları elde edilir. *Bu bağlamda, EKKB'nin GAÇ uygulamalarında ilişki skoru kestirimi için iyi bir tercih olacağı düşünülmüş ve bu tez çalışmasında kullanılacak kestirimcilere dâhil edilmiştir.*

1.1.10 Genel Analiz ve Literatür Özeti Sonucu

Bu son alt başlıkta, kullanılan veri kümeleri ve başarım metrikleri de hesaba katılarak ve önceki analizler göz önüne alınarak GAÇ uygulamarında en iyi performansı vereceği düşünülen kestirimciler verilmiştir.

İncelenen çalışmaların birçoğunda karşılaştırma için öncelikle sentetik veri kümeleri kullanılmıştır. Daha sonra bu çalışmaların bir kısmında değerlendirme için gerçek biyolojik veri kümeleri kullanılmıştır. Çalışmaların bir kısmında başarım metriği olarak gerçek KB ve kestirimcinin tahmin ettiği KB değerleri arasındaki hata farkı kullanılmıştır [23], [24]. Bir kısmında ise başarım metriği, gerçek gen ağı ile çıkartılan gen ağı arasındaki hata farkına dayandırılmıştır [17], [18]. İncelenen çalışmalardaki bu veri kümesi ve performans değerlendirme kriterlerinin farklılıkları, kestirimcilerin performans sıralamasında değişikliklere sebep olabilmektedir. Veri kümesindeki örnek sayısının değişimi bile aynı kestirimcinin performansında farklılaşmaya sebep olabilmektedir [45]. Yine de bu tez çalışmasında sunulan başarım sıralamaları yöntemlerle ilgili bir ön bilgi verebilmektedir. Bu bilgiler ışığında, literatüre dayanarak önceki alt bölümlerde verilen değerlendirme ve karşılaştırma sonuçları bu alt bölümde özetlenmiştir.

F-skor metriği ve SynTReN simulatörü ile üretilen gürültüsüz sentetik veri kümeleri kullanıldığında 5 kestirimcinin sıralaması: MM > EB > SKK > PKD > Çekilme (shrinkage) şeklindedir. Aynı metrik ve gürültülü sentetik veri kümeleri kullanıldığında ise sıralama: SKK > PKD > MM > EB > Çekilme (shrinkage) şeklinde olmaktadır [17]. Gerçek genomik veri kümeleri gürültülü olduğu için *SKK, GAÇ uygulamalarında kullanıma uygun bir yöntem olarak görülmektedir*. Bununla beraber doğrusal ilişkiyi ölçebilen bir korelasyon metriği olan PKD'nın biyolojik veri kümelerindeki davranışını görmek için *PKD'yi de bu tezde kullanmak üzere seçtik*.

Çalışma [18]'de de sentetik veri kümeleri kullanılmıştır. 4 farklı KB tabanlı tekniğin başarım sıralaması: MM > EB > Schürmann-Grassberger (Bayesian 3) > Çekilme şeklinde elde edilmiştir. Başarım değerlendirme metriği olarak Hassasiyet-Geri çekilme eğrisi altındaki alan (Area under the curve Precision-Recall, AUC-PR) kullanılmıştır. [18]'de veri kümesinde gürültü olma/olmama durumuna göre bir değerlendirme yapılmamıştır. *MM, [18]'de ve farklı çalışmalarda [17],[28] etkin bir yöntem olarak gözlendiği için bu tez çalışmasında da gerçeklenmek ve kullanılmak üzere seçilmiştir*.

Bir başka çalışmada sentetik veri kümeleri üretilerek ve gerçek KB ve tahmin edilen KB değerleri arasındaki farkların OKH kriteri kullanılarak 9 kestirimci karşılaştırılmıştır [23]. [23]'teki sıralamanın [17] ve [18]'dekinden farklı olmasının sebebi [23]'te kullanılan başarım değerlendirme kriterinin diğer çalışmalardakinden farklı olmasıdır. James-Stein Çekilme yöntemi 9 kestirimci arasında en iyi performans sergileyenlerden birisi olarak gösterilmiş, çalışmanın son kısmında bu kestirimci gerçek bir genomik veri kümesi ile kullanılarak GAÇ uygulamaları için ümit verici bir yöntem olduğu öne sürülmüştür [23]. CS kestirimcisi ve NSB, Çekilme yöntemiyle beraber en iyi sonuçları vermişlerdir. CS KB tahmini için ümit verici bir yöntem olarak görülmektedir çünkü performansı NSB'yle benzer olmasına rağmen hesaplama karmaşıklığı NSB'ninkinden önemli oranda düşüktür, ayrıca gerçeklenmesi NSB'ye nazaran daha kolaydır. *Dolayısıyla CS bu tez çalışmasında kullanılacak yöntemlere dahil edilmiştir*.

Bir başka çalışmada EÜS kestirimcisi 3 farklı kestirimci ile sapma ve varyans gibi çeşitli istatistiksel açılardan karşılaştırılmıştır [28]. EÜS'nin EB ve Jackknife yöntemlerinden daha iyi olduğu, örnek sayısının hücre sayısına yaklaşık eşit olduğu durumlarda MM kestirimcisinden de daha iyi olduğu belirtilmiştir [28]. Ancak bu durum, entropi-tabanlı yöntemleri kullanan GAÇ algoritmaları için geçerli değildir, çünkü hücre sayısı örnek sayısından çok daha küçük seçilmektedir. Bu çalışmanın dışında, B-spline kestirimcisi EÜS ve ÇYK ile karşılaştırılmış, B-spline'ın bütün eğri dereceleri için EÜS'den daha başarılı sonuçlar verdiği belirtilmiştir [19]. Dolayısıyla EÜS iyi bir tercih olarak görülmemektedir. Eğri derecesi 2'den büyük olduğunda B-spline ÇYK'dan daha iyi performans sergilemektedir. Ancak ÇYK, Copula Dönüşümü (CD) denilen fazladan bir ön işleme ihtiyaç duyarken, B-spline'ın CD'ye ihtiyaç duymadığı; ayrıca ÇYK'nın B-

spline'dan 10^4 kat daha fazla hesaplama karmaşıklığı olduğu belirtilmiştir [19]. *Bu tez çalışmasında hem B-spline hem ÇYK yöntemleri kullanılmak üzere seçilmiştir.*

KEYK-KB$^{(2)}$ yöntemi sapma içermediği ve bir örnek ve bu örneğin k'ncı komşusu üzerindeki dağılımı daha doğru bir şekilde ifade edebildiği için entropi-tabanlı KEYK kestirimcisinden ve KEYK-KB$^{(1)}$ kestirimcisinden daha iyi performans vermektedir [30]. [30]'daki performans metriği KB'nin sistematik hatası, yani gerçek ve tahmin edilen KB'ler arasındaki farktır. [*30]-[32]'deki sonuçlar KEYK-KB$^{(2)}$'nin GAÇ uygulamalarında başarılı bir yöntem olabileceğini gösterdiği için bu yöntem de bu teze dâhil edilmiştir*.

Dolaylı bağlantıların elendiği çalışmalar da bu çalışmada incelenmiştir. Yüksek dereceli KPK'larının kısmi olmayan karşılıklarından daha başarılı oldukları belirtilmiştir [33]-[34]. [34]'te KPK-n'nin birçok deneyde PKD, KPK-1, KKB-1 ve B-spline'dan daha başarılı olduğu belirtilmiştir. *Dolayısıyla KPK-n bu tezde incelenen yöntemlere dâhil edilmiştir*.

[24]'te etkileşim skorlarını tahmin etmek üzere EBK yöntemi önerilmiştir. Çeşitli fonksiyonel ilişkiler içeren yapay veri kümeleri kullanılarak yapılan deneylerde EBK'nın SKK, KEYK, ÇYK ve PKD'ye fark attığı bildirilmiştir. Birçok çalışmada mKor ve HHG'nin önemli oranda EBK'dan daha iyi olduğu belirtilmiştir [35]-[37]. Ayrıca HHG mKor'a fark atmıştır [37]. *Dolayısıyla HHG ümit verici bir yöntem olarak görülmüş ve bu tez çalışmasında kullanılması kararlaştırılmıştır*.

[44]'te EKKB yöntemi önerilmiş, iki farklı mikroçip veri kümesi kullanılarak PKD, ÇYK, KEYK ve Edgeworth ile karşılaştırılmıştır. EKKB'nın bu 4 yönteme fark attığı ve bunlardan daha avantajlı olduğu belirtilmiştir: EKKB, ÇYK'nın aksine yoğunluk tahminine ihtiyaç duymaz. EKKB, KEYK'ın tersine model seçimine izin verebilir. EKKB, Edgeworth'un aksine veri kümesinin dağılımı ile ilgili bir kısıtı sağlamak zorunda değildir (Edgeworth için veri normale yakın dağılım göstermeli). *Dolayısıyla EKKB makul bir seçim olarak görülmüş ve bu tezde gerçeklenmek üzere seçilmiştir*.

Bunların dışında, Bayesian Gen Ağı (BGA) uygulamaları, bu tez çalışmasının kapsamının dışında olmasına rağmen incelenmiştir. Çekilme, PKD ve yüksek dereceli KPK yöntemlerinin BGA ile kullanılabildiği gözlenmiştir [46], [47]. [48]'de BGA'ların genelleştirilmiş doğrusal regrasyon, olasılıksal yapay sinir ağları, olasılıksal karar ağaçları, sözlük modelleri vb. gibi olasılıksal sınıflandırma ve regrasyon modelleri ile

çıkarılabildiği belirtilmiştir. BGA'lar, etkin olarak Gen Düzenleyici Ağlardaki (GDA) yönlü bağlantıların çıkartılmasında kullanılabilirler. Statik ve dinamik olmak üzere iki temel gruba ayrılırlar. Dinamik BGA'lar Markov sürecinin olduğunu varsayar, dolayısıyıla her bir değişkenin (gen) kendinden önce gelen değişkenlere bağlı olduğunu kabul eder [47]. Ancak bu çalışmada yönlü bağlantıların olduğu ağ çıkarımı hedeflenmemiştir ve süreç Markovian değildir. Dolayısıyla BGA'ları bu tez çalışmasında detaylı incelenmemiştir ve daha önce belirtildiği gibi, bu çalışmanın kapsamı dışındadır. Bu çalışmada, yönsüz bağlantıların olduğu ARACNE [1], RelNet [2] ve C3NET [3] gibi GAÇ algoritmalarında kullanılan etkileşim kestirimcileri incelenmiştir. Ayrıca yönsüz ağ çıkarımı yapan GAÇ'ların aksine BGA'lar büyük ölçekli veri kümeleri için etkin olarak kullanılamaz.

Literatür incelemesinin sonucunda, GAÇ algoritmaları ile kullanılmaya uygun olan en ümit verici kestirimciler belirlenmiştir. Seçilen kestirimciler:

- Pearson Korelasyon Değeri (PKD) Katsayısı
- Spearman Korelasyon Katsayısı (SKK)
- n'nci dereceden Kısmi Pearson Korelasyonu (KPK-n)
- Heller, Heller, Gorfine (HHG) korelasyonu
- Miller-Madow (MM) kestirimcisi
- Chao-Shen (CS) kestirimcisi
- B-Spline (BS) kestirimcisi
- Çekirdek Yoğunluk Kestirimcisi (ÇYK)
- Doğrudan KB kestirimi yapabilen K-en Yakın Komşuluk (KEYK-KB$^{(2)}$) kestirimcisi
- En küçük Kareler Karşılıklı Bilgi (EKKB) kestirimcisi

şeklindedir. Ayrıca, sırayla PKD ve SKK'nın türevleri olan Pearson Tabanlı Gaussian (PTG) ve Spearman Tabanlı Gaussian (STG) da incelenecek yöntemlere eklenmiştir.

Bundan sonraki alt bölümlerde tezin amacı ve literatüre katkısı verilmiştir.

1.2 Tezin Amacı

Bu çalışmada GAÇ algoritmalarının temel adımı olan etkileşim skoru tahmini aşaması detaylı olarak incelenmiştir. Korelasyon tabanlı, entropi tabanlı veya doğrudan karşılıklı bilgi tabanlı olmak üzere 27 farklı etkileşim skoru kestirimcisine dair detaylı incelemeler ve aralarından hangilerinin hangi gerekçelerle kullanılmak üzere seçildiği literatür incelemesi bölümünde verilmiştir. Sadece genom bilimi alanında değil, çeşitli alanlardaki kestirimcilerin kullanıldığı çalışmalar da incelenmiş ve çıkarımlar yapılmıştır. Hâlihazırda literatürde bu kadar çok kestirimcinin bir arada incelendiği bir başka çalışma bulunmamaktadır. Deneylerde kestirimcilerin GAÇ algoritmalarındaki çıkarım performansını değerlendirebilmek amacıyla daha önce belirtildiği gibi ARACNE [1], RelNet [2] ve C3NET [3] GAÇ algoritmaları kullanılmıştır.

Tez çalışması kapsamında, herhangi bir GAÇ algoritması ile kullanılması durumunda en yüksek başarıyı vermesi olası olan kestirimciler belirlendikten sonra gerçeklenmiştir. Böylelikle bu alanda çalışan araştırmacıların kendi GAÇ algoritmalarını değerlendirirken kestirimci faktörünü de göz önünde bulundurmalarını, kullandıkları kestirimcilerden daha başarılı kestirimcilerin olabileceğini görmelerini ve önerilen kestirimcileri kendi uygulamalarına kolaylıkla uyarlayabilmelerini sağlamak amaçlanmıştır. Bu amaçla kestirimciler gerçeklendikten sonra bir R yazılım paketi oluşturulmuş ve sunulmuştur. Yazılım paketinde, kestirimciler çok çekirdekli bilgisayarlarda paralel kodlama mantığı ile de çalışabilecek şekilde gerçeklenmiş, böylece kullanıcıların zamandan tasarruf etmesi hedeflenmiştir. Ayrıca, belli kestirimcilerin hiper-parametre (Ör: KEYK'da komşu sayısı, ÇYK'da çekirdeğin bant genişliği, B-spline'da eğri derecesi, vb) seçimleri üzerinde incelemeler yapılmıştır. Böylece araştırmacıların, kestirimcilerden yüksek başarım alabilmelerini sağlamak hedeflenmiştir. İncelenen kestirimcilerin detaylı sunumu ve açıklaması verilmiş, bazı kestirimciler için açıklayıcı örnekler sunulmuştur. Bunlara ilaveten, kestirimciler farklı bakış açılarına göre (parametrikliğe ve ayrıklaştırmaya göre) sınıflandırılmış ve araştırmacıların veri kümelerinin veya uygulamalarının gerektirdiği şekilde kestirimci tercihi yapmalarını kolaylaştırmak amaçlanmıştır.

Son olarak, daha önceki çalışmalardan farklı olarak bu çalışmada kestirimcilerin başarımı üzerinde sık kullanılan bir ön işlem olan Copula Dönüşümünün (CD) etkisini incelemek amaçlanmış ve bununla ilgili sonuçlar raporlanmıştır.

1.3 Orijinal Katkı

Bu tez çalışmasında daha önce belirtildiği gibi GAÇ algoritmalarının temel adımı olan ilişki skoru kestirimi aşamasında kullanılabilecek yöntemler detaylı bir şekilde incelenmiştir. Çalışmanın literatüre katkıları aşağıdaki şekilde özetlenebilir:

- Bilgimiz dâhilinde ilişki skoru kestirimi aşamasında kullanılabilecek yöntemlerin sayıca ve nitelik olarak bu kadar detaylı incelendiği başka bir çalışma litertürde bulunmamaktadır.
- Literatürde, GAÇ algoritmaları ile kullanılan kestirimcilerin sınıflandırmasının yapıldığı az sayıda çalışma mevcuttur [22]. [22]'de az sayıda kestirimcinin doğrusallığa ve parametrikliğe göre sınıflandırması yapılmıştır. *Bu tez çalışmasında ise daha çok sayıda kestirimcinin doğrusallığa, parametrikliğe ve ayrıklaştırma tekniğine ihtiyacı olup olmamasına göre sınıflandırılması yapılmıştır*. Böylece, bu alanda çalışan araştırmacıların veri kümesine veya uygulamasına göre en uygun kestirimciyi seçmesini sağlamak hedeflenmiştir.
- GAÇ kullanımında kullanımı en uygun olacağı düşünülen kestirimciler hem R hem de C++ dilleri ile gerçeklenmiş, C++ kodları R'dan çağrılabilir hale getirilerek herkese açık bir R yazılım paketi oluşturulmuştur (Bölüm 5). Yöntemlerin C++ kodları çok çekirdekli bilgisayarlarda paralel de çalıştırılabilmeye olanak sağlayacak şekilde yazılmıştır. *Bilgimiz dâhilinde hem seri hem de paralel programlama mantığı ile gerçeklenen ve bu kadar çok sayıda kestirimciyi içeren bir R paketi hâlihazırda bulunmamaktadır*. Bu tez, bu yönüyle özgün bir çalışmadır.
- Yöntemlerin bir kısmı için en uygun parametre seçimi incelenmiş, hazırlanan paket programı kullanacak kişiler için bu parametreler ile ilgili öneriler getirilmiştir. KEYK yöntemindeki komşu sayısının, ÇYK'da çekirdek bant genişliğinin ve B-spline'daki eğri derecesinin belirlenmesi detaylı olarak incelenmiştir. B-spline'ın eğri derecesi daha önce beşinci dereceye kadar [19] incelenmişken bu çalışmada 10. dereceye

kadar inceleme ilerletilmiştir. Ayrıca CD ön işleminin B-spline'daki en uygun eğri derecesine etkisi de incelenmiştir. Bu tez çalışması B-spline ile ilgili literatürde bulunmayan bu eksiklikleri gidermiştir.

- Ayrıklaştırmaya ihtiyaç duyan yöntemler için ayrıklaştırma esnasında veri kümesinin bölüneceği en uygun hücre sayısının ne olacağı incelenmiş, bununla ilgili detaylar Bölüm 6.8'de verilmiştir. *Bilgimiz dâhilinde bu parametrenin en uygun değerinin incelenmesi daha önce literatürde yapılmamıştır*. Bu tez çalışması bu yönüyle de bir ilktir.
- Son olarak sık kullanılan bir ön işlem olan CD tekniğinin yöntemlerin başarımı üzerindeki etkisi detaylı olarak incelenmiştir. Yine bilgimiz dâhilinde daha önce bu ön işlemin kestirimci performansı üzerindeki etkisi incelenmemiştir. Bu tez çalışması bu incelemesi ile de bir ilktir.

Bu çalışma 7 bölümden oluşmaktadır. Bölüm 2'de incelenen 27 kestirimciye dair yapılan çeşitli sınıflandırmalar verilmiştir. Literatür incelemesi sonucunda seçilen kestirimcilerin anlatımı Bölüm 3'te, bu çalışmada kullanılan üç farklı GAÇ algoritmasının anlatımı ise Bölüm 4'te yer almaktadır. Tez çalışması kapsamında oluşturulan yazılım paketi Bölüm 5'te tanıtılmış; Bölüm 6'da ise kestirimcilere dair detaylı deneysel sonuçlar ve çıkarımlar sunulmuştur. Son bölümde ise çalışmanın sonuçları ve ilerideki çalışmalar bulunmaktadır.

BÖLÜM 2

ETKİLEŞİM KESTİRİMCİLERİNİN SINIFLANDIRILMASI

Giriş bölümünde belirtildiği gibi, bu çalışmada etkileşim skoru kestirimcileri özelliklerine ve çeşitli bakış açılarına göre sınıflandırılmışlardır. Etkileşim skoru kestirimcilerinin sınıflandırıldığı az sayıda çalışma mevcuttur [22]. [22]'de etkileşim kestirimcilerinin parametrik olup olmamasına göre sınıflandırma yapılmıştır. Bu tez çalışması kapsamında ise kestirimciler doğrusallığa, parametrikliğe ve ayrıklaştırma yaklaşımına göre sınıflandırılmıştır. Bu sınıflandırmaların amacı, GAÇ uygulamaları alanında çalışan araştırmacıların veri kümesi dağılımına ve diğer ihtiyaçlarına göre en uygun kestirimcileri seçebilmelerini sağlamaktır.

Etkileşim kestirimcileri doğrusal olup olmamalarına göre sınıflandırılırken, iki rastgele değişken (iki gen) arasındaki işlevsel ilişkinin doğrusal olup olmadığı dikkate alınmıştır. Parametrikliğe göre sınıflandırma yapılırken ise her bir rastgele değişkenin (genin) veya rastgele değişken ikilileri arasındaki ilişkinin belli ve bilindik bir dağılıma sahip olup olmadığı dikkate alınmıştır. Bunların dışında, yöntemler ayrıklaştırmaya ihtiyaç duyup duymamalarına göre, ayrıklaştırmaya ihtiyaç duyan yöntemlerin de ayrıklaştırma tekniğine göre sınıflandırması yapılmıştır. Kestirimcilerin herbir bakış açısına göre sınıflandırmaları sırayla Bölüm 2.1, 2.2 ve 2.3'te verilmiştir.

2.1 Etkileşim Kestirimcilerinin Doğrusallığa göre Sınıflandırılması

Kestirimcilerin bir kısmı, rastgele değişken ikilileri veya gen ikilileri arasındaki sadece doğrusal ilişkileri bulabilirken, bir kısmı doğrusal olmayan veya herhangi bir işlevsel model içermeyen ilişkileri de bulabilir. Doğrusal ilişkileri kestirebilen yöntemlerin

gerçeklenmeleri daha kolay ve hesaplama karmaşıklıkları daha düşüktür. Bu nedenle doğrusal olmayan (non-linear) yöntemlerin yerine tercih edilebilir.

Doğrusal olmayan yöntemler ise hesaplama karmaşıklıkları doğrusal yöntemlerden daha fazla olmasına rağmen doğrusal olmayan türdeki ilişkileri de bularak iki değişken arasındaki bağımlılık derecesini gösteren skoru elde edebilirler.

Giriş bölümünde detaylı incelemesi verilen 27 etkileşim kestirimciden üç tanesi doğrusal kestirimcilerdir ve gen ikilileri arasındaki sadece doğrusal ilişkilerin skorunu bulabilir. Bu kestirimciler:

- Pearson Korelasyon Katsayısı Değeri (PKD),
- 1'nci dereceden Kısmi Pearson Korelasyonu (KPK-1) ve
- n'nci dereceden Kısmi Pearson Korelasyonu (KPK-n) şeklindedir.

Bu üç kestirimci, gen ikilileri arasındaki ilişkinin doğrusal olmaması durumunda ilişki skorunu "0" olarak elde ederler. Skorun 0 olması iki gen arasında doğrusal bir ilişki olmadığını gösterir ancak buradan bu iki genin arasında hiçbir ilişki olmadığı çıkarımını yapmak doğru değildir. Çünkü bu iki gen arasında doğrusal olmayan bir ilişki de bulunabilir. Dolayısıyla doğrusal etkileşim kestirimcileri gen ikilileri arasındaki etkileşimleri bulurken yetersiz kalmaktadırlar. Bu alanda çalışan araştırmacılar uygulamalarına ve veri kümelerine en uygun yöntemi seçerken bu durumu göze almalılardır.

Doğrusal kestirimciler, gen ikilileri arasındaki ilişkinin bilindik bir dağılıma (doğrusal ilişkiye) sahip olduğunu varsaydıkları için aynı zamanda parametrik yöntemler sınıfına da girerler. Her doğrusal kestirimci aynı zamanda parametrik kestirimci olmak zorundadır. Ancak her parametrik kestirimci doğrusal yöntem olmak zorunda değildir.

2.2 Etkileşim Kestirimcilerinin Parametrik Olmalarına göre Sınıflandırılması

Bu çalışmada, etkileşim kestirimcileri parametrik olma özelliğine göre de sınıflandırılmıştır. Çizelge 2.1'de bu sınıflandırma sonucunda her bir kestirimcinin hangi sınıfa dahil olduğu gösterilmiştir.

Çizelge 2. 1 Kestirimcilerin parametrikliğe göre sınıflandırılması

Parametrik olan	**Parametrik olmayan**	**Yarı-Parametrik olan**
PKD; Bayesian 1 (Jeffreys' öncülü); Bayesian 2 (Bayes-Laplace öncülü); Bayesian 3 (Perks öncülü, Schürmann Grassberger) Bayesian 4 (Minimax öncülü); Edgeworth Kestirimcisi; EKKB Kestirimcisi; 1. dereceden KPK (KPK-1); n. dereceden KPK (KPK-n); Jackknife Kestirimcisi *	SKK; Kendall Tau Korelasyon Katsayısı; En büyük Benzerlik (EB); Miller-Madow (MM); ANOVA katsayısı; Chao-Shen (CS); B-spline (BS); Çekirdek Yoğunluk Kestirimcisi (ÇYK); K-en Yakın Komşuluk (KEYK) Entropi Kestirimcisi; KEYK doğrudan KB Kestirimci-1; KEYK doğrudan KB Kestirimci-2; En iyi Üst Sınır (EÜS); 1. dereceden Koşullu KB (KKB-1); En büyük Bilgi Katsayısı (EBBK); Mesafe Korelasyonu (mKor); Heller, Heller, Gorfine (HHG); Jackknife Kestirimcisi *	James-Stein Çekilme Kestirimcisi
* Jackknife yöntemi, çapraz doğrulama (cross validation)'da kullanılan kestirimci yönteme bağlı olarak parametrik ya da non-parametrik olabilir.		

Parametrik yöntemler, ilgilenilen rastgele değişkenlerin (genlerin) örnek dağılımının bilindiğini veya değişkenler arasındaki ilişki modelinin belli ve bilindik bir dağılıma sahip olduğunu varsayar. Örneğin Edgeworth yöntemi veri kümesinin dağılımının Gaussian olduğunu varsayan parametrik bir yöntemdir [42]. Parametrik yöntemlere bir başka örnek de gen ikilileri arasındaki ilişkinin doğrusal olduğunu varsayan PKD kestirimcisidir.

Parametrik yaklaşımlar veri kümeleri ile ilgili çok sayıda varsayım ve kısıtlamaya sahip olabilir ancak bu sayede araştırmacılara veri kümesi ile ilgili daha detaylı bilgi sağlayabilirler. Dağılımın parametreleri veri kümesindeki örnekler kullanılarak elde

edilebilir. Veri kümesi ile ilgili varsayımlar yapmanın dışında, rastgele değişkenler arasındaki ilişki ile ilgili de varsayımlar yapılabilir. Bu varsayımlar değişkenler arasındaki ilişki modelinin parametrelerine dair daha detaylı bilgi verirler ancak aynı zamanda bu ilişki modelinin çok çeşitlenememesi, esnek olamaması ve kısıtlı kalmasına sebep olurlar. Parametrik yöntemler Çizelge 2.1'de en sol sütunda listelenmiştir.

Parametrik olmayan (non-parametric) yöntemler, dağılımdan bağımsız istatistikler olarak da adlandırılabilir [22]. İlgilenilen rastgele değişkenlerin (genlerin) dağılımı ile ilgili bir bilgi yoksa veya veri kümesinin dağılımı bilindik bir fonksiyon ailesinden gelmiyorsa, ilişki skoru kestirimi için parametrik olmayan yöntemler kullanılmalıdır. Parametrik olmayan yöntemler veri kümesi ile ilgili daha az bilgi içerdiği için parametrik yöntemlere göre daha başarısız tahminler yapabilirler. Veri kümesi dağılımı dışında, rastgele değişkenler arasındaki ilişkinin dağılımı da bilinmiyorsa yine parametrik olmayan yöntemler kullanılmalıdır. Parametrik olmayan yöntemlerin kullanılması durumunda iki rastgele değişken arasındaki ilişkinin belli bir fonksiyon ailesinden gelme ya da belli bir modele sahip olma zorunluluğu olmadığı ve daha esnek olduğu için skoru bulunacak ilişki ile ilgili bir kısıtlama yapılmamış olur. Çizelge 2.1'de orta sütunda parametrik olmayan yöntemler listelenmiştir.

Yarı parametrik yöntemler, parametrik olan ve olmayan yöntemlerin birtakım özelliklerine aynı anda sahiptir. Bu yöntemler parametriklik özelliğine veri kümesi dağılımının veya ilişki modelinin belli bir oranda bilinebileceğini varsaymalarından dolayı sahiptir. Bu yöntemlerin parametrik olma veya olmama özelliklerinden hangisinin daha baskın olacağı bir ağırlık parametresi ile ayarlanabilmektedir. Çizelge 2.1'de de görüldüğü üzere incelenen 27 kestirimciden sadece James-Stein çekme (shrinkage) yöntemi yarı parametrik yöntemler sınıfına girmektedir.

Sonuç olarak değişkenlerin veri kümesi dağılımının veya ilişki modelinin bilinmesi durumunda parametrik yöntemlerle başarılı tahminler yapılabilir. Ancak veri kümesi dağılımı bilinmiyorsa veya bilindiği varsayılan modelden (fonksiyondan) uzaklaşıyorsa parametrik yöntemler başarılı tahminler yapamaz, bunların yerine parametrik olmayan yöntemler kullanılmalıdır.

2.3 Etkileşim Kestirimcilerinin Ayrıklaştırma Tekniğine göre Sınıflandırılması

Bölüm 2.1 ve 2.2'de verilen sınıflandırmalara alternatif olarak kestirimciler, veri kümesini ayrıklaştırma ya da hücrelere ayırma işlemine ihtiyaç duyup duymadıklarına göre de sınıflandırılabilirler. Kestirimcilerin bu yaklaşıma göre sınıflandırılmaları Çizelge 2.2'de verilmiştir.

Çizelge 2. 2 Kestirimcilerin ayrıklaştırma tekniklerine göre sınıflandırılması

Korelasyon Tabanlı Kestirimciler			**KB Tabanlı Kestirimciler**			
			Hücre olasılıkları kullanan "Entropi Tabanlı Kestirimciler"			**Direkt KB Kestirebilen Yöntemler**
Sadece doğrudan bağlantıları bulabilen kestirimciler	**Dolaylı (Indirect) bağımlılıkları da eleyen kestirimciler**		**Her bir hücrede bir tek örnek içeren kestirimciler**	**Her bir hücrede çok sayıda örnek içeren kestirimciler**		
PKD; SKK; Kendall Tau; ANOVA; mKor; HHG;	KPK-1; KPK-n		ÇYK; KEYK Entropi Kestirimcisi; Edgeworth Kestirimcisi	EB; MM; Jeffreys' öncülü; Bayes-Laplace öncülü; Perks öncülü (Schürmann-Grassberger); Minimax öncülü; CS; James-Stein çekme; EÜS; Jackknife; KKB-1; BS; EBBK		KEYK direkt KB Kestirimci 1; KEYK direkt KB Kestirimci 2; EKKB Kestirimcisi

PKD ve SKK gibi korelasyon tabanlı yöntemler iki rastgele değişken arasındaki ilişki skorunu bulmak için veri kümesinin ayrıklaştırılmasını gerektirmez. Veri kümesinin tüm örneklerini kullanarak ikinci momentleri (varyansları) elde eder, buradan da ilişki

skorunu doğrudan bulurlar. PKD sadece doğrusal ilişkileri bulabilen parametrik bir yöntem iken SKK doğrusal olmayan ilişkileri de bulabilir. Ancak SKK da sadece monoton olan ilişkileri bulabilir, monoton olmayan ilişkilerin skorunu tespit etmede başarısız kalır. Dolayısıyla korelasyon tabanlı yöntemlerin başarımı, veri dağılımı doğrusallıktan ve monotonluktan uzaklaştıkça azalır, doğrusallığa veya monotonluğa yaklaştıkça artar. Bunun dışında korelasyon tabanlı yöntemlerin bir kısmı dolaylı (indirect) bağımlılıkların elenebilmesine olanak sağlar. Korelasyon tabanlı yöntemleri iki temel gruba ayırabiliriz:

- Sadece doğrudan (direct) olan doğrusal veya monoton ilişkilerin skorunu bulabilen yöntemler (Bknz. Çizelge 2.2)
- Doğrusal veya monoton ilişkilerin skorunu bulmanın yanısıra dolaylı bağımlılıkları eleyebilen yöntemler (Bknz. Çizelge 2.2)

Korelasyon tabanlı yöntemlerin haricinde, rastgele değişkenler (genler) arasındaki etkileşim, Karşılıklı Bilgi (KB) metriği ile de ölçülebilir. İlişki skoru olarak KB metriğini kullanan etkileşim skoru kestirimcilerinin bir kısmı rastgele değişkenlere dair entropi hesaplamasını gerektirir. Bir veri kümesi, farklı bölgelerde farklı dağılım özellikleri göstermeye meyilli ise entropi-tabanlı kestirimciler ile KB skoru tahmini yapılmalıdır. Entropi-tabanlı yöntemler, rastgele değişkenlerin örneklerini içeren veri kümesini çok sayıda hücreye bölerek ayrıklaştırır. Bu işlem fazladan bir yük getirdiği gibi entropi tahmin hatasından kaynaklanan bir sapmaya (bias) da sebep olabilir. Yine de veri kümesi dağılımının homojen olmadığı veya tam olarak bilinmediği durumlarda kullanılması en uygun yöntemler, entropi-tabanlı KB skoru kestirimcileridir. Entropi tabanlı yöntemlerin bir çoğu histogram yöntemine dayanır. Histogram tabanlı yöntemler ve ayrıklaştırma teknikleri Bölüm 3.2'de detaylı bir şekilde incelenmiştir. Entropi tabanlı kestirimciler ayrıklaştırma işlemi sonucunda her bir hücreye düşen örnek sayısına bakılarak iki grup altında toplanabilir:

- Her bir hücre çeşitli sayıda örnek içerebilir. Böylece her bir hücrenin olasılık dağılımı bu hücredeki örneklere bakılarak elde edilir (Bknz. Çizelge 2.2).
- Her bir hücre tek bir örnek içerir, her bir örnek ve belli komşuları üzerindeki olasılık dağılımları toplanarak sonuç etkileşim skoru elde edilir (Bknz. Çizelge 2.2).

Korelasyon tabanlı ve entropi tabanlı kestirimciler haricinde, ilişki skorunu entropiye ihtiyaç duymadan doğrudan elde edebilen “direkt KB skoru kestirimcileri” de bulunmaktadır. Bu kestirimcilerin entropi tahmin hatasından kaynaklanan sapmayı içermedikleri için entropi tabanlı yöntemlerden daha başarılı olduğu öne sürülmüştür [30]. Dolayısıyla KB tabanlı kestirimcileri, entropi tabanlı olan ve direkt KB skoru bulan yöntemler olmak üzere iki temel gruba ayırabiliriz. Bu ayrıma dair ağaç yapısı Çizelge 2.2’de görülmektedir.

Tez çalışmasının bu bölümünde kestirimcilerin çeşitli bakış açılarına göre sınıflandırmaları verilmiştir. Böylelikle GAÇ uygulamaları alanında çalışan araştırmacıların çalışmalarına ve veri kümelerine en uygun kestirimciyi belirlemelerine yardımcı olmak amaçlanmıştır.

BÖLÜM 3

ETKİLEŞİM SKORU KESTİRİMCİLERİ

Bu bölümde etkileşim skoru kestirimcilerinin detaylı anlatımı verilmiştir. Kestirimcilerin tanımı ve işlem adımlarının anlatımından önce entropi, Karşılıklı Bilgi (KB), doğrudan ve dolaylı bağlantı şeklindeki temel kavramların tanımı Bölüm 3.1'de verilmiştir.

Kestirimcilerin sınıflandırılması bölümünde de belirtildiği gibi KB skoru bulan yöntemlerin bir kısmı entropi tabanlıdır ve veri kümesinin ayrıklaştırılmasına ihtiyaç duymaktadır. Ayrıklaştırma süreci ve bu süreçte kullanılabilecek teknikler Bölüm 3.2'de anlatılmıştır.

Kestirimcilerin tanımları sırayla Bölüm 3.3'ün alt başlıklarında verilmiştir. İncelenen kestirimciler:

- Pearson Korelasyon Değeri (PKD) Katsayısı
- Pearson Tabanlı Gauss (PTG) Kestirimcisi
- Spearman Korelasyon Katsayısı (SKK)
- Spearman Tabanlı Gauss (STG) Kestirimcisi
- n'nci dereceden Kısmi Pearson Korelasyonu (KPK-n)
- Heller, Heller, Gorfine (HHG) korelasyonu
- Miller-Madow (MM) kestirimcisi
- Chao-Shen (CS) kestirimcisi
- B-Spline (BS) kestirimcisi
- Çekirdek Yoğunluk Kestirimcisi (ÇYK)

- Doğrudan KB kestirimi yapabilen K-en Yakın Komşuluk (KEYK-KB$^{(2)}$) kestirimcisi
- En küçük Kareler Karşılıklı Bilgi (EKKB) kestirimcisi

şeklindedir.

3.1 Entropi, Karşılıklı Bilgi (KB), Doğrudan ve Dolaylı Bağlantı

Entropi, rastgele değişkenlerin belirsizliğini ölçen bir kavramdır. Fizik [49]-[50], kimya [50]-[51], haberleşme [12]-[13] ve istatistik gibi çeşitli çalışma alanlarında; metin veya görüntü sıkıştırma [52], doğal dil işleme [53], işaret işleme [7]-[11], örüntü segmentasyonu ve sınıflandırması [54]-[55] gibi çok çeşitli uygulamalarda kullanılmaktadır.

Entropisi ölçülecek rastgele değişkenin öngörülemezliği arttıkça entropisi de artar. Ancak öngörülemezlik azaldıkça yani rastgele değişkenin durumu daha bilinir oldukça entropi azalır. Dolayısıyla bir rastgele değişkenin durumu kesin olarak biliniyorsa bu değişkenin entropisi 0 olur. Bir X rastgele değişkeni ile ilgili Shannon entropi tanımı:

$$H(X) = -\sum_{x_i \in X} P(X = x_i) \log\left(P\left(X = x_i\right)\right) \tag{3.1}$$

ile verilir.

Rastgele değişkenlerin entropisini elde edebilmek için veri kümesinin ayrıklaştırılarak hücrelere bölünmesi gerekmektedir. Ayrıklaştırma işleminin sonucunda her bir i hücresinin olasılık dağılımı $P\left(X = x_i\right)$ olmak üzere Bağıntı (3.1) kullanılarak entropi elde edilir. Ayrıklaştırma işlemi ve kullanılabilecek teknikler Bölüm 3.2'de verilmiştir.

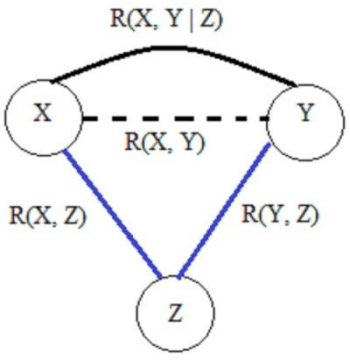
İki rastgele değişken arasındaki Karşılıklı Bilgi (KB) skoru, bu iki değişkenin birbiriyle olan bağımlılığını, ilişkisini ve etkileşimini ölçer. $H(X)$ ve $H(Y)$ bireysel entropi, $H(X,Y)$ birleşik entropi olmak üzere KB (Mutual Information - MI) skoru (3.2)'den elde edilebilir:

$$\begin{aligned} MI(X,Y) &= H(X) + H(Y) - H(X,Y) \\ &= -\sum_{x_i \in X} \sum_{y_j \in Y} P(X = x_i, Y = y_j) \log \frac{P(X = x_i, Y = y_j)}{P(X = x_i) P(Y = y_j)} \end{aligned} \tag{3.2}$$

$H(X,Y)$ birleşik entropisi, $P(X=x_i,Y=y_j)$ birleşik olasılık dağılımı kullanılarak elde edilir:

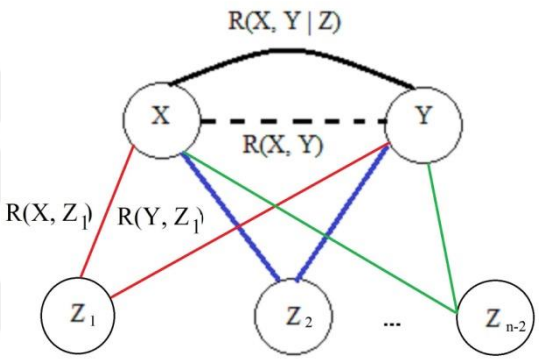
$$H(X,Y)=-\sum_{x_i\in X}\sum_{y_j\in Y}P(X=x_i,Y=y_j)\log\left(P(X=x_i,Y=y_j)\right) \quad (3.3)$$

İki rastgele değişken arasındaki etkileşim skoru, korelasyon katsayıları ile elde edilirken sistemdeki diğer değişkenlerin bu iki değişkenin korelasyonu üzerinde birtakım etkileri bulunabilir. Bu etkiler, gerçekte birbiriyle doğrudan bağlantısı olmayan iki değişken arasında bir korelasyon bulunduğu yanılgısına sebep olabilir. Bu durumda iki değişken arasındaki dolaylı, bir başka deyişle doğrudan olmayan (indirect) bağlantıların tespit edilerek elenmesi gerekmektedir. Bu durumu bir örnek üzerinden açıklayalım: Şekil 3.1'de verildiği gibi, X ve Z rastgele değişkenleri arasında doğrudan bir bağlantı olduğunu kabul edelim. Y ve Z arasında da doğrudan bir bağlantı olduğunu; ancak X ve Y arasında doğrudan bir bağlantı olmadığını varsayalım. X ve Y arasında doğrudan bir bağlantı bulunmamasına rağmen ikisi de Z ile ilişkide oldukları için bu iki değişken arasında yüksek bir korelasyon skoru elde etmek olasıdır. Bu durumda, bu korelasyonun Z'den kaynaklandığı, ve X - Y arasında doğrudan bir ilişki bulunmadığı tespit edilerek X - Y arasındaki bağlantı kopartılmalıdır. Bu amaçla lüteratürde kısmi korelasyon katsayısı bulan yöntemler [33], [34] önerilmiştir. Kısmi korelasyon katsayıları, X ile Y'nin Z ile ilişkide bulunmayan kısımlarının korelasyon katsayısını elde eder. Bu elde edilen değer belli bir eşik değerinin altında ise X ile Y arasında doğrudan bir bağlantı olmadığı kabul edilir.



Şekil 3. 1 X ve Y rastgele değişkenleri, Z üzerinden dolaylı ilişkidedir

Örneğin X ve Y arasındaki tüm olası birinci dereceden kısmi korelasyon ilişkileri Şekil 3.2'de gösterilmiştir. Birinci dereceden korelasyon katsayısı ile X ve Y arasında doğrudan bir ilişki bulunup bulunmadığı incelenirken, X ve Y arasındaki kısmi korelasyon, bu iki değişken dışındaki tüm değişkenler (Z_i'ler) üzerinde koşullandırılarak ayrı ayrı elde edilir. Üçüncü bir rastgele değişken (Z_i) üzerinde koşullandırılarak elde edilen bu korelasyon değerlerinden herhangi biri belli bir eşik değerinin altında ise X ve Y'nin o üçüncü değişken üzerinden dolaylı olarak birbirine bağımlı olduğu varsayılır ve X - Y arasındaki bağlantı kopartılır. Bir ilişki matrisindeki tüm ilişkiler burada anlatıldığı gibi değerlendirildikten sonra elde edilen sonuç grafiğinin sadece doğrudan ilişkileri içerdiği, dolaylı ilişkileri elediği kabul edilir.



Şekil 3. 2 X ve Y rastgele değişkenlerinin 1. dereceden tüm olası koşullu (kısmi) ilişkileri

Dolaylı ilişkileri elemek için kısmi korelasyon katsayıları haricinde koşullu karşılıklı bilgi (conditional mutual information) skorunu elde eden yöntemler de kullanılabilir. Literatür incelemesi bölümünde verildiği üzere n'nci dereceden kısmi Pearson korelasyonu katsayısı, dolaylı bağlantıları eleyen yaklaşımlar arasında en başarılısı olarak belirtildiği için [34] bu tez çalışmasında da bu yöntem kullanılmıştır.

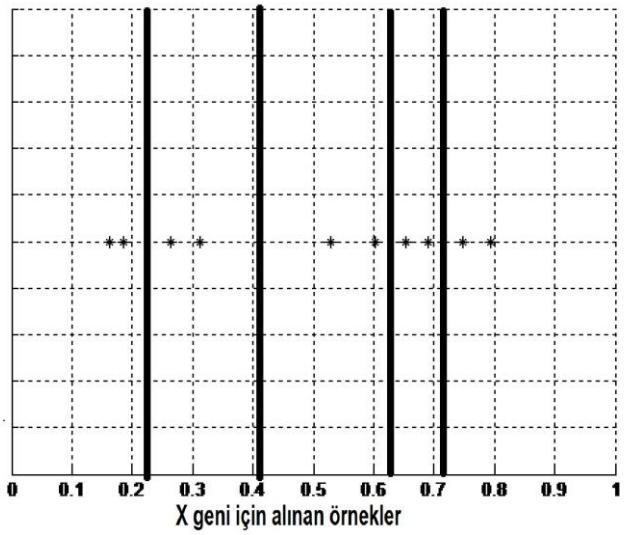
3.2 Ayrıklaştırma İşlemi ve Ayrıklaştırmada Kullanılan Teknikler

KB skoru tahmin eden kestirimcilerin bir kısmı bu skoru doğrudan elde edebilirken, bir kısmı bu skoru bireysel ve birleşik entropilerden bulur. Entropi tabanlı KB skoru kestirimcileri PKD, SKK vb. gibi korelasyon tabanlı kestirimcilerden ve diğer KB skoru kestirimcilerinden farklı olarak veri kümesinin ayrıklaştırılmasını veya hücrelere

bölünmesini gerektirir. Ayrıklaştırma işleminden sonra Bağıntı (3.1)'deki gibi her bir hücrenin olasılık dağılımından entropi; entropiden de Bağıntı (3.2)'deki gibi KB skoru elde edilir. İki temel ayrıklaştırma tekniği bulunmaktadır. Bunlar eş frekans ve eş genişlik ayrıklaştırma yaklaşımlarıdır ve sırayla Bölüm 3.2.1 ve 3.2.2'de verilmişlerdir.

3.2.1 Eş Frekans (EF) Ayrıklaştırma Tekniği

Veri kümesi, Eş Frekans (EF) ayrıklaştırma tekniği ile hücrelere ayrılırken her bir hücre eşit sayıda örnek içerecek şekilde bölünür. Hücrelerin eleman sayısı yaklaşık eşit olur ancak genişlikleri birbirinden farklı olur.



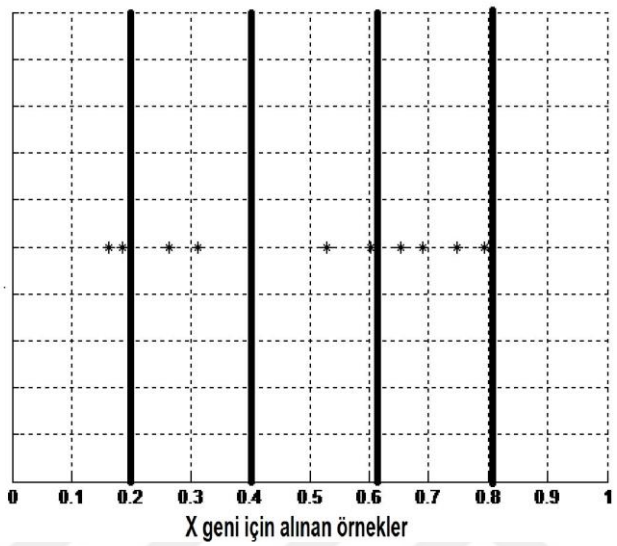
Şekil 3. 3 EF tekniği ile hücrelere ayırma (ayrıklaştırma)

Şekil 3.3'te bir X geni için 10 farklı örnekten alınan normalize edilmiş gen ifade değerlerinin EF tekniği ile ayrıklaştırılma sonucu gösterilmiştir. Normalize edilen gen ifade değerleri {0.16, 0.79, 0.31, 0.53, 0.19, 0.60, 0.26, 0.65, 0.69, 0.75} şeklindedir. Bu küçük veri kümesi ayrıklaştırılırken veri kümesinin 5 hücreye bölünmesi istenmiştir. Ayrıklaştırılma sonucunda her bir hücrede eşit sayıda (ikişer) örnek bulunmaktadır ve hücrelerin genişliği birbirinden farklıdır. Her bir hücredeki örnek sayısından hücrelerin olasılık dağılımı bulunur ve Bağıntı (3.1)'den X geninin bireysel entropisi elde edilir.

3.2.2 Eş Genişlik (EG) Ayrıklaştırma Tekniği

Veri kümesi, EG tekniği ile ayrıklaştırılırken her bir hücrenin eşit genişlikte olacak şekilde bölünmesi sağlanır. Böylece her bir hücreye düşen eleman sayısı değişken

olmaktadır. Şekil 3.3'te EF tekniği kullanılarak ayrıklaştırılan veri kümesinin EG tekniği ile ayrıklaştırılma sonucu Şekil 3.4'te görüldüğü gibidir. Her bir hücrenin genişliği 0.2 birimdir. İlk üç hücrede ikişer örnek, dördüncü hücrede 4 örnek bulunmaktayken, beşinci hücrede hiçbir örnek bulunmamaktadır.



Şekil 3. 4 EG tekniği ile hücrelere ayırma (ayrıklaştırma)

N örnek sayısı olmak üzere, her iki ayrıklaştırma tekniğinde de hücre sayısı çoğunlukla $\sqrt{N}$ olarak alınır [17], [18]. Böylece her bir hücrenin belli sayıda veri örneği içermesi sağlanmaya çalışılır. Bilgimiz dahilinde, literatürde varsayılan hücre sayısı olarak neden bu değerin seçildiğine dair detaylı bir inceleme bulunmamaktadır. Bu tez çalışmasında ayrıklaştırma işlemi için en uygun hücre sayısının incelemesi de yapılmış, $\sqrt{N}$'nin uygun bir varsayılan değer olduğu gösterilmiştir. Bölüm 6.8'de bu incelemenin detayları verilmiştir.

3.3 Etkileşim Skoru Kestirimcilerinin İşlem Adımları

Kestirimcilerin anlatımında sıklıkla karşımıza çıkacak olan entropi, KB, ayrıklaştırma, vb. temel kavramlar Bölüm 3.1 ve 3.2'de verildiğine göre yöntemlerin detaylı tanımına geçilebilir. Bu bölümün alt başlıklarında kestirimcilerin anlatımı verilmiştir.

3.3.1 Pearson Korelasyon Katsayısı Değeri (PKD)

Pearson korelasyon katsayısı, Karl Pearson tarafından önerilen [56] ve iki rastgele değişken arasında doğrusal bir ilişki olduğunu varsayan bir yaklaşımdır. Aslında PKD'ye benzer bir yöntem daha önce Francis Galton tarafından da önerilmiştir [57]. PKD, biyoenformatik alanındaki çok sayıda çalışmada [17], [33], [34], [58]-[64] kullanılmıştır. Korelasyon tabanlı bir yaklaşım olan PKD, sadece doğrusal ilişkilerin skorunu bulabildiği için doğrusal ve parametrik bir yöntem olarak sınıflandırılmıştır.

$\mathrm{cov}(X,Y)$: X ve Y rastgele değişkenlerinin kovaryansı, σ_x: X'in standart sapması, σ_y: Y'nin standart sapması olmak üzere PKD:

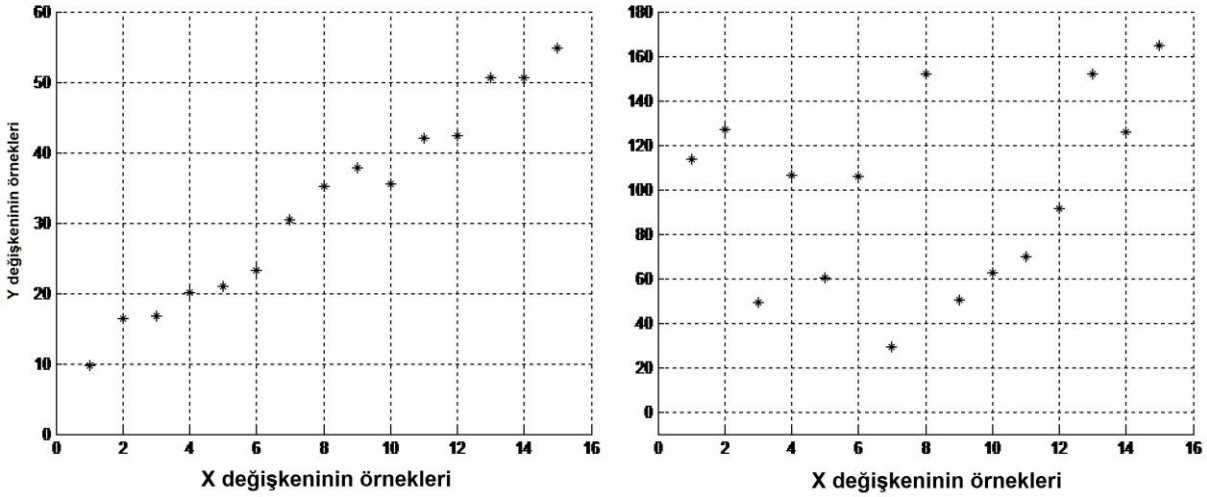
$$\rho = \frac{\mathrm{cov}(X,Y)}{\sigma_x \sigma_y} \tag{3.4}$$

ile elde edilir.

N: X ve Y rastgele değişkenlerinin (genlerinin) örnek sayısı olmak üzere, PKD X ve Y değişkenlerinin x_i ve y_i veri örnekleri kullanılarak Bağıntı (3.5) ile de ifade edilebilir:

$$\hat{\rho} = \frac{N\sum_{i=1}^{N} x_i y_i - \sum_{i=1}^{N} x_i \sum_{i=1}^{N} y_i}{\sqrt{N\sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i\right)^2}\sqrt{N\sum_{i=1}^{N} y_i^2 - \left(\sum_{i=1}^{N} y_i\right)^2}} \tag{3.5}$$

Korelasyon değeri [-1, 1] aralığında değer alır. Korelasyon değerinin 0 olması iki değişkenin veya genin birbirinden kesinlikle bağımsız olması anlamına gelmez. Aralarında doğrusal olmayan bir ilişki de bulunabilir. PKD sadece doğrusal ilişkileri ölçebilen bir yöntem olduğu için doğrusal olmayan ilişkilerin skorlarını tahmin edemez. Bu durumla ilgili bir örnek Şekil 3.5'te verilmiştir. Şekil 3.5 (a)'da verilen örnekte iki rastgele değişken arasında doğrusala yakın bir ilişki olduğu görülmektedir. PKD sonucunda da korelasyon değeri 0.98 olarak elde edilmiştir. Ancak Şekil 3.5 (b)'deki dağılıma bakıldığında değişkenler arasındaki ilişkinin doğrusallıktan uzaklaştığı görülmektedir, PKD ile ilişkinin skoru 0.29 olarak elde edilmiştir.



(a) (b)

Şekil 3. 5 (a) Değişkenler arası ilişki doğrusallığa yakın iken (Korelasyon değeri 0.98); (b) Değişkenler arası ilişki doğrusallıktan uzak iken (Korelasyon değeri 0.29);

3.3.2 Pearson Tabanlı Gaussian (PTG) Kestirimcisi

İki rastgele değişkenin birleşik olasılık dağılımı Gauss fonksiyonu özelliği taşıyor ise korelasyon skoru ile KB skoru ilişkilidir. İki rastgele değişkenin birleşik olasılığı ile ilgili bir varsayım yapıldığı için PTG parametrik yöntemler sınıfına girmektedir.

Gauss dağılımına sahip d boyutlu, σ_x standart sapmalı bir X rastgele değişkeninin entropisi:

$$H(X) = \frac{1}{2}\ln\left\{(2\pi e)^d(\sigma_x)\right\} \tag{3.6}$$

ile elde edilir.

Benzer şekilde, X ve Y değişkenlerinin kovaryansı $\mathbf{C}$ olmak üzere, birleşik entropi Bağıntı (3.7) ile bulunur:

$$H(X,Y) = \frac{1}{2}\ln\left\{(2\pi e)^{2d}\det(\mathbf{C})\right\} \tag{3.7}$$

Bağıntı (3.2), (3.6) ve (3.7) kullanılarak korelasyon değerinden KB skorunu veren ilişki:

$$MI(X,Y) = \frac{1}{2}\log\left(\frac{\sigma_x\sigma_y}{\det(\mathbf{C})}\right) = -\frac{1}{2}\left(1-\rho^2\right) \tag{3.8}$$

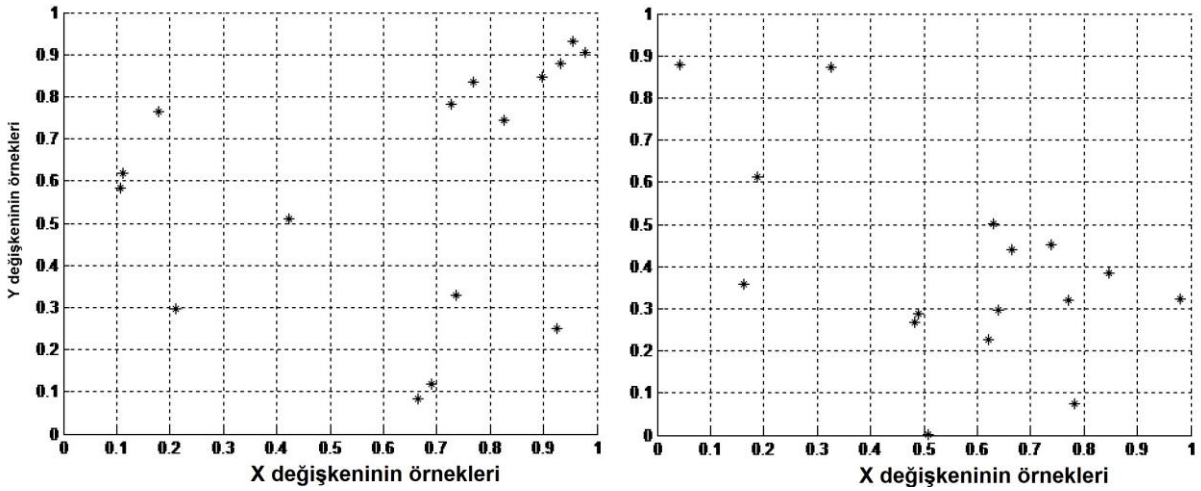
şeklinde elde edilir.

Bu tez çalışmasında Pearson Tabanlı Gaussian ilişki kestirimcisi de kullanılmıştır.

3.3.3 Spearman Korelasyon Katsayısı (SKK)

Spearman Korelasyon Katsayısı (SKK) yöntemi, PKD'nin özel bir halidir ve biyoenformatik alanında çok sayıda çalışmada kullanılmıştır [17], [34], [65]-[70]. İlişki skoru elde edilirken veri kümesindeki orijinal değerler yerine, verinin küçükten büyüğe sıralanması durumunda sahip olacağı sıra numarası kullanılır. SKK yönteminde Bağıntı (3.5)'teki x_i ve y_i veri örneklerinin orijinal değerleri yerine bu örneklerin sıralama sonucundaki sıra bilgisi kullanılarak ilişki skoru elde edilir.

SKK ile iki değişken arasındaki sadece doğrusal ilişkiler değil, doğrusal olmayan ilişkiler için de ilişki skoru başarılı bir şekilde tespit edilebilir. Veri kümesi ile ilgili bir kısıtlama getirilmediği için SKK parametrik olmayan yöntemler sınıfındadır. Ancak değişkenler arasındaki ilişkinin monoton olması gerekmektedir. Monoton olmayan ilişkiler için (doğrusal olsun/olmasın) ilişki skoru tespiti SKK ile etkin bir şekilde yapılamayabilir.



(a) (b)

Şekil 3. 6 (a) Düşük PKD, orta SKK örneği; (b) Orta PKD, düşük SKK örneği

Aralarında gerçekte doğrusal olmayan bir ilişki olduğu bilinen iki rastgele değişkene dair iki farklı dağılım Şekil 3.6 (a) ve (b)'de verilmiştir. (a)'da PKD düşük düzeyde (0.24), SKK orta düzeydedir (0.55); (b)'de ise PKD orta düzeyde (-0.54) ve SKK düşük düzeydedir (-0.29). Bu değişkenler için dağılımdaki örneklerin durumuna göre PKD ve SKK'nın tahmin başarımı görüldüğü gibi değişmektedir. Dağılım doğrusal yapı

göstermeye yaklaştıkça PKD daha başarılı tahmin yapabilmekte, monoton yapı hakimse de SKK daha başarılı tahminler yapabilmektedir.

3.3.4 Spearman Tabanlı Gaussian (STG) Kestirimcisi

Bölüm 3.3.2'de PTG yöntemi tanımında belirtildiği üzere, iki rastgele değişkenin birleşik olasılık dağılımı Gauss fonksiyonu özelliği taşıyor ise korelasyon skoru ile KB skoru ilişkilidir. Bağıntı (3.8)'deki PKD skoru (ρ) yerine, SKK skoru konularak Spearman Tabanlı Gaussian (STG) kestirimcisinin KB tahmini elde edilebilir. Bu çalışmada STG yöntemi de kullanılmıştır. İki rastgele değişkenin birleşik olasılığı ile ilgili bir varsayım yapıldığı için STG parametrik yöntemler sınıfına girmektedir.

3.3.5 n'nci Dereceden Kısmi Pearson Korelasyon Katsayısı (KPK-n)

Çakır vd. Grafiksel Gauss Modelleme (GGM) veya n'nci dereceden Kısmi Pearson Korelasyon Katsayısı (KPK-n) isimli parametrik ve koşullu olan bir ilişki skoru kestirimcisi önermişlerdir [34]. GGM veya KPK-n yöntemi bir gen etkileşim matrisindeki doğrudan olmayan (dolaylı-indirect) bağlantıların kopartılması için kullanılabilir. Bölüm 3.1'de dolaylı bağlantının ne olduğu ve neden elenmesi gerektiği anlatılmıştır. KPK-n yöntemi iki değişken arasındaki korelasyonu eş zamanlı olarak geri kalan tüm diğer değişkenlere koşullandırarak elde eder. İşlem adımları:

- Öncelikle 0'ncı dereceden (standart) Pearson korelasyon skorlarını barındıran korelasyon matrisi elde edilir.
- PKD ile elde edilen korelasyon matrisinin tersi alınır.
- Tersi alınan matrisin diyagoneli "-1" değerini alacak şekilde normalize edilir. Normalizasyon işlemi [71]'de verilen yaklaşımla gerçeklenir. Buna göre $\mathbf{P}$: Pearson korelasyon değerlerini içeren matris, $\mathbf{\Omega} = \mathbf{P}^{-1} = \omega_{i,j}$: $\mathbf{P}$'nin tersi olan matris olmak üzere; n'nci dereceden kısmi Pearson korelasyon matrisi olan $\mathbf{\Pi}$:

$$\mathbf{\Pi}_{i,j} = -\frac{\omega_{i,j}}{\sqrt{\omega_{i,i} \times \omega_{j,j}}} \tag{3.9}$$

ile elde edilir.

KPK-n kestirimcisi sadece doğrusal ilişki skorlarını elde edebildiği için parametrik yöntemler sınıfına dahildir.

3.3.6 Heller, Heller, Gorfine (HHG) Kestirimcisi

Heller vd. rastgele vektörler arasındaki bağımsızlığı test edebilmek amacıyla yeni bir metrik önermişlerdir [37]. Literatürde değişkenlerin bağımsızlığını test eden yöntemler nadirdir. Değişkenler arasındaki etkileşimi bulabilmek için çoğunlukla bağımlılık testleri gerçeklenir. Bağımsızlık testlerinde sıfır hipotezi (null hypothesis) iki rastgele değişkenin bağımsız olduğunu söylemektedir. HHG de sıfır hipotezinde çok boyutlu iki rastgele değişkenin (X ve Y) bağımsız olduğunu öne sürer. EBK ve mKor gibi yöntemlerden daha başarılı olduğu [35], [37] literatür özeti bölümünde verilmiştir.

X ve Y rastgele değişkenlerinin (x_0, y_0) noktası merkez alınarak (R_x ve R_y) yarıçapındaki dairesel bir bölge ile gösterilebildiğini varsayalım. Merkez noktası ve her iki boyuttaki yarıçaplar kesin olarak bilinemez. Tutarlı ve başarılı bir test istatistiği elde edebilmek için x_0, y_0, R_x ve R_y değerleri uygun bir şekilde seçilmelidir. HHG yöntemi için bu nokta ve yarıçaplar deneme yanılma yaklaşımı ile seçilmiştir. Bu bağlamda, HHG metriği X ve Y değişkenlerinin veri örneklerinin ikili farklarını, yani $\{d_X(x_i, x_j): i, j \in \{1,\ldots,N\}\}, \{d_Y(y_i, y_j): i, j \in \{1,\ldots,N\}\}$ ifadesini kullanmayı gerektirir. Aslında HHG metriği, bu ikili mesafelerin bir fonksiyonudur. $d_X(\cdot,\cdot)$ ve $d_Y(\cdot,\cdot)$ ifadeleri Öklit, Mahalanobis vb. herhangi bir mesafe metriği ile bulunabilir.

I bir gösterge fonksiyonu olmak üzere, bu fonksiyonun içindeki ifade doğru ise fonksiyon değeri 1; ifade doğru değilse fonksiyon değeri 0'dır. Örnek sayısı N olmak üzere, rastgele değişkenlerin gözlemlerinin (örneklerinin) ikili farklarının R_x ve R_y mesafesinden büyük ve küçük olma durumuna göre sınıflara ayrılışı Çizelge 3.1'de verildiği gibidir. Bu çizelge değiştirilmeden [37]'den alınmıştır. Çizelge 3.1'deki örneğin A_{11} ifadesine bakalım. $k \in \{1,\ldots,N\}$ olmak üzere:

$$A_{11} = \sum_{k=1}^{N} I\{d(x_0, x_k) \le R_x\} I\{d(y_0, y_k) \le R_y\} \tag{3.10}$$

ile elde edilir.

Çizelge 3. 1 $I\{d(x_0, x_k) \le R_x\}$ ve $I\{d(y_0, y_k) \le R_y\}$'e göre örnekleri sınıflandırma [37]

	$d(y_0,\cdot) \le R_y$	$d(y_0,\cdot) > R_y$	
$d(x_0,\cdot) \le R_x$	A_{11}	A_{12}	$A_{1\bullet}$
$d(x_0,\cdot) > R_x$	A_{21}	A_{22}	$A_{2\bullet}$
	$A_{\bullet 1}$	$A_{\bullet 2}$	N

Yukarıda belirtildiği gibi tutarlı ve etkin bir test istatistiği tanımlayabilmek için x_0, y_0, R_x ve R_y değerleri uygun bir şekilde seçilmelidir ve bunun için HHG'de deneme yanılma yaklaşımı kullanılmıştır. (x_0, y_0) noktası yerine her bir (x_i, y_i) gözlem ikilisi kullanılır ve her $i \in \{1,\ldots,N\}$ ve $j \in \{1,\ldots,N\}$ için $R_x = d(x_i, x_j)$ hesaplanır. Benzer şekilde aynı ikili için R_y değeri de elde edilir. Dolayısıyla ($k \neq i$ ve $k \neq j$ olmak üzere $k \in \{1,\ldots,N\}$) geri kalan $N-2$ adet nokta için A_{11}, A_{12}, A_{21}, ve A_{22} ifadeleri Çizelge 3.2'deki sınıflandırma ile elde edilir. Çizelge 3.2 de değiştirilmeden [37]'den alınmıştır.

Çizelge 3. 2 $I\{d(x_i, x_k) \le d(x_i, x_j)\}$ ve $I\{d(y_i, y_k) \le d(y_i, y_j)\}$'e göre örnekleri sınıflandırma [37]

	$d(y_i,\cdot) \le d(y_i, y_j)$	$d(y_i,\cdot) > d(y_i, y_j)$	
$d(x_i,\cdot) \le d(x_i, x_j)$	A_{11}	A_{12}	$A_{1\bullet}$
$d(x_i,\cdot) > d(x_i, x_j)$	A_{21}	A_{22}	$A_{2\bullet}$
	$A_{\bullet 1}$	$A_{\bullet 2}$	N

A_{11}, A_{12}, A_{21}, ve A_{22} ifadeleri kullanılarak her bir (i, j) için Pearson'ın chi karesi (chi-square) testi ile $S(i, j)$ skoru edilir:

$$S(i,j) = \frac{(N-2)\{A_{12}(i,j)A_{21}(i,j) - A_{11}(i,j)A_{22}(i,j)\}^2}{A_{1\bullet}(i,j)A_{2\bullet}(i,j)A_{\bullet 1}(i,j)A_{\bullet 2}(i,j)} \tag{3.11}$$

$S(i, j)$ skorundan da HHG test skoru elde edilir:

$$T = \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} S(i, j) . \tag{3.12}$$

$S(i, j)$ skoru büyüdükçe X ve Y değişkenleri arasındaki bağımlılık da artar.

3.3.7 Miller-Madow (MM) Kestirimcisi

Daha önceden belirtildiği gibi KB tabanlı yöntemlerin bir kısmı entropi hesaplanmasını gerektirir. Entropi hesaplaması için de veri kümesinin hücrelere bölünerek ayrıklaştırlıması gerekmektedir. Bölüm 3.2'de ayrıklaştırma süreci ile ilgili detaylı bilgiler verilmiştir.

En büyük Benzerlik (EB - Maximum Likelihood) kestirimcisi entropi tabanlı yöntemlerin temelidir. EB'de ayrıklaştırma işlemi sonucunda her bir hücredeki eleman sayısına bakılarak her bir hücre için bireysel frekanslar elde edilir. Ayrıklaştırılan iki ayrı rastgele değişkenin veri örneklerinin birleşik frekansı da her bir bin için elde edilir. [22]'de EB kestirimcisinin parametrik bir yöntem olduğu belirtilmiştir. Ancak bu yöntem verinin dağılımı veya değişkenler arasındaki ilişkinin türü ile ilgili herhangi bir varsayım yapmadığı için parametrik olmayan bir kestirimci olarak sınıflandırmak daha doğrudur.

EB'de deneysel entropi gözlemlerin olasılık dağılımından elde edilir. Tek bir rastgele değişken için N : örnek sayısı, b : hücre sayısı, n_k : k 'ncı hücredeki örnek adedi olmak üzere deneysel entropi H_{deney}:

$$H_{deney} = -\sum_{k=1}^{b} \left(\frac{n_k}{N} \right) \log \left(\frac{n_k}{N} \right) \tag{3.13}$$

ile elde edilir. H_{deney}, ayrık bir rastgele değişken için EB entropi tahminini vermektedir. Ayrıklaştırma işleminde kullanılan hücre sayısı (b) arttıkça, EB yaklaşımı ile tahmin edilen bu değer (H_{deney}) gerçek entropi değerinden uzaklaşır. Çünkü hücre sayısının artması hücre olasılıklarının seyrek-örneklenmesine (undersampling) sebep olur [17], [18], [23], [28]. Bu da EB kestirimcisinin asimptotik sapmasının (Asymptotic bias) Bağıntı (3.14)'teki değere ulaşmasına sebep olur.

$$sapma\left(H_{deney}\right) = -\frac{b-1}{2N} \tag{3.14}$$

EB yöntemi çeşitli çalışmalarda [17], [18], [23], [28], [43] kullanılmış, ancak Bağıntı (3.14)'teki tahmin sapmasından dolayı önerilmemiştir.

Miller-Madow (MM) kestirimcisi entropi tahmini yaparken Bağıntı (3.14)'teki seyrek-örnekleme sapmasını hesaba katar ve daha başarılı entropi tahmini yapar. Böylece MM, varyansı artırmadan sapmayı azaltmış olur. Her bir kestirimcinin amacı hem sapmayı hem de varyansı en küçük seviyede tutabilmektir. Ancak sapma ve varyans arasında ters orantı bulunmaktadır, biri artarken diğeri azalmaktadır. Bir kestirimcinin karmaşıklığı arttıkça sapma azalır, ancak varyans çok artar. MM ise karmaşıklığı ve varyansı artırmadan sapmayı azaltabilmektedir. Sonuç olarak MM'nin entropi tahmini:

$$H_{mm} = H_{deney} + \frac{b-1}{2N} \tag{3.15}$$

ile gösterilebilir.

Parametrik olmayan bir yöntem olan MM'nin kullanıldığı çok sayıda çalışma bulunmaktadır [72]-[75]. Bu tez çalışmasında MM yöntemi hem EG hem de EF ayrıklaştırma tekniklerine göre ayrı ayrı değerlendirilmiştir.

3.3.8 Chao-Shen (CS) Kestirimcisi

Chao-Shen Kestirimcisi de MM ve EB yöntemleri gibi ayrıklaştırma işlemini gerektirir. Bu yöntem, Horvitz-Thompson kestirimcisi ve EB kestirimcisinin Good-Turing düzeltimi yaklaşımlarını birleştirir. İlk defa [76]'da önerilmiştir. Bağıntı (3.13)'te verilen EB entropi kestirimcisinde (H_{deney}), N : örnek sayısı, n_k : k 'ncı hücredeki örnek adedi olmak üzere her bir hücrenin deneysel olasılığı $\hat{\theta}_k^{deney} = \frac{n_k}{N}$ ile de ifade edilebilir. Bu durumda deneysel entropi kestirimi:

$$H_{deney} = -\sum_{k=1}^{b} \hat{\theta}_k^{deney} \log\left(\hat{\theta}_k^{deney}\right) \tag{3.16}$$

haline gelir.

m_1: Eleman sayısı 1 olan hücrelerin adedi ($n_k = 1$) olmak üzere, k'ncı hücrenin olasılığı için EB kestirimcisinin Good-Turing düzeltimi:

$$\hat{\theta}_k^{GT} = \left(1 - \frac{m_1}{N}\right)\hat{\theta}_k^{deney} \tag{3.17}$$

şeklindedir.

Horvitz-Thompson kestirimcisine göre sonuç entropi tahmini:

$$\hat{H}^{CS} = -\sum_{k=1}^{b} \frac{\hat{\theta}_k^{GT} \log\left(\hat{\theta}_k^{GT}\right)}{\left(1 - \left(1 - \hat{\theta}_k^{GT}\right)^n\right)} \tag{3.18}$$

olur.

Parametrik olmayan bir yöntem olan CS kestirimcisi çok sayıda çalışmada kullanılmıştır [23], [77]-[81]. Bu tez çalışmasında CS yöntemi hem EG hem de EF ayrıklaştırma tekniklerine göre ayrı ayrı değerlendirilmiştir.

3.3.9 B-Spline (BS) Kestirimcisi

B-spline, bağlayıcı fonksiyonların (spline functions) bir çeşitidir ve temel eğri (basis spline) olarak da isimlendirilir. Bağlayıcı fonksiyonlar çeşitli düğüm noktalarından (knot) birbirine bağlanan eğriler olarak düşünülebilir. Her bir bağlayıcı fonksiyonun belli bir derecesi, düzgünlüğü (smoothness) ve B-spline'ların doğrusal kombinasyonu ile tanımlanabilecek alan bölmeleri (domain partition) bulunmaktadır. Düğüm noktaları haricinde, eğrinin nerede ve nasıl değişeceğini gösteren kontrol noktaları da bulunmaktadır. Düğüm noktalarının adedi, kontrol noktalarının adedi ve eğrinin derecesinin toplamı ile elde edilir. Düğüm noktaları, uzayı farklı bölgelere böler ve bu noktaların pozisyonları parametre uzayının eğri uzayına dönüşümünü yakından etkiler [82]. Düğüm noktalarının değerleri azalan sırada olamaz, eşit veya artarak devam edebilir. Kontrol noktalarındaki değişimler dikkate alınarak düğüm noktaları belirlenebilir. Daub vd. KB skorunun nümerik tahmini için B-spline fonksiyonlarının kullanılmasını önermiştir [19]. B-spline kestirimcisi biyoenformatik alanında çeşitli çalışmalarda kullanılmıştır [19], [28], [83]-[85]. B-spline yöntemi de MM ve CS yaklaşımları gibi ayrıklaştırma işlemini gerektirir. Ancak B-spline'ın daha farklı bir

ayrıklaştırma yaklaşımı vardır. MM ve CS'nin EG ve EF ayrıklaştırma teknikleri ile gerçekleştirdiği klasik ayrıklaştırma yaklaşımında her bir veri örneği sadece bir hücreye ait olabilir. Çeşitli gürültülerden dolayı hücrelerin sınırındaki veri noktaları yanlışlıkla komşu hücrelere atanabilir. Bu da, sonuçta yapılan KB skoru tahminini olumsuz etkileyebilir. [19]'da klasik ayrıklaştırma yöntemine bir genelleştirme yapılarak veri örneklerinin aynı anda birden fazla hücreye ait olabilmesi sağlanmıştır. Bir başka deyişle, BS kestirimcisi ile katı-ayrıklaştırma (hard-binning) esnetilerek yumuşak-ayrıklaştırma (soft-binning) yapılmıştır.

x_u örneğinin a_i hücresine üyeliğini gösteren ve klasik ayrıklaştırmada kullanılan Bağıntı (3.19)'teki gösterge fonksiyonu genelleştirilmiş ve çok terimli B-spline fonksiyonları kullanılarak Bağıntı (3.21) ile ifade edilmiştir.

$$\theta_i(x_u) = \begin{cases} 1, & x_u \in a_i \text{ ise} \\ 0, & \text{degilse} \end{cases} \tag{3.19}$$

Bağıntı (3.19)'daki gösterge fonksiyonu kullanılarak ayrıklaştırma sürecinde her bir hücrenin deneysel olasılığı Bağıntı (3.20)'de verildiği gibi elde edilebilir:

$$\hat{p}(a_i) = \frac{1}{N}\sum_{u=1}^{N}\theta_i(x_u) \tag{3.20}$$

Her bir veri örneği ayrıklaştırma sonucunda aynı anda birden fazla hücreye ait olabilir ve ait olduğu her bir i hücresi için B-spline fonksiyonları kullanılarak birer $\tilde{B}_{i,k}$ ağırlığı atanır. Sonuç olarak her bir i hücresi için $\hat{p}(a_i)$ olasılığı Bağıntı (3.21) haline gelir:

$$\hat{p}(a_i) = \frac{1}{N}\sum_{u=1}^{N}\tilde{B}_{i,k}(x_u) \tag{3.21}$$

[19]'da önerilen bu yöntemin sözde kodunu inceleyelim:

Girdiler: $u = 1,\ldots,N$ olmak üzere x_u ve y_u örnekleri; k : B-spline fonksiyonlarının eğri derecesi; $i = 1,\ldots,M_x$ ve $j = 1,\ldots,M_y$ olmak üzere a_i ve b_j hücreleri

Çıktı: X ve Y arasındaki Karşılıklı Bilgi skoru $KB(X,Y)$

İşlem Adımları:

1. X rastgele değişkeni için bireysel entropinin hesaplanması:

   a. $z = (x - x_{\min})\dfrac{M_x - k + 1}{x_{\max} - x_{\min}} + 1$ kullanılarak $\tilde{B}_{i,k}(x) = B_{i,k}(z)$ belirlenir.

   b.Her bir x_u için M_x adet ağırlık katsayısı, $\tilde{B}_{i,k}(x_u)$'den belirlenir.

   c.Tüm x_u örneklerini kullanarak Bağıntı (3.21)'den her bir hücre için $\hat{p}(a_i)$ hesaplanır.

   d. $H(X) = -\sum_{i=1}^{b} \hat{p}(a_i)\log(\hat{p}(a_i))$ entropisi hesaplanır.

2. X ve Y değişkenlerinin birleşik entropisinin hesaplanması:

   a.1(a) ve (b) adımları X ve Y için ayrı ayrı yapılır.

   b.Birleşik olasılıklar $p(a_i, b_j)$, tüm $M_x \times M_y$ adet hücre için elde edilir:

$$p(a_i, b_j) = \frac{1}{N}\sum_{u=1}^{N} \tilde{B}_{i,k}(x_u) \times \tilde{B}_{j,k}(y_u). \tag{3.22}$$

   c.Birleşik entropi $H(X,Y)$ hesaplanır.

3. $KB(X,Y) = H(X) + H(Y) - H(X,Y)$ ile KB skoru elde edilir.

BS kestirimcisinde eğri fonksiyonlarının derecesi bir veri örneğinin aynı anda kaç farklı hücreye dahil olabileceğini göstermektedir. [19]'daki çalışmanın b-spline uyarlaması aşağıda verilmiştir. Buna göre m adet düğüm noktası ($t_1 \le t_2 \le \cdots \le t_m$) varken, k'ncı dereceden bir B-spline eğrisi:

$$S : [t_{k+1}, t_{m-k}] \to R \tag{3.23}$$

şeklinde tanımlanabilir.

B-spline, k'ncı dereceden $B_{i,k}$ eğrilerinin doğrusal kombinasyonudur:

$$S(z) = \sum_{i=1}^{m-k-1} P_i B_{i,k}(z), \quad z \in [t_{k+1}, t_{m-k}] \tag{3.24}$$

Cox-de Boor özyineleme fonksiyonu ile k'ncı dereceden temel (basis) B-spline'lar olan $B_{i,k}$'ler aşağıdaki gibi elde edilebilir:

$$B_{i,1}(z) = \begin{cases} 1, & t_i \le z \le t_{i+1} \quad \text{ise} \\ 0, & \text{degilse} \end{cases} \tag{3.25}$$

ve

$$B_{i,k}(z) = B_{i,k-1}(z)\frac{z - t_i}{t_{i+k-1} - t_i} + B_{i+1,k-1}(z)\frac{t_{i+k} - z}{t_{i+k} - t_{i+1}} \tag{3.26}$$

k'ncı dereceden bir B-spline fonksiyonu için m adet düğüm noktası varsa, $m-k-1$ adet kontrol noktası ve temel (basis) B-spline bulunmaktadır. [19]'da düğüm noktaları tespit edilirken, M : hücre sayısı ve k : eğri derecesi olmak üzere, $M+k$ adet düğüm noktası için Bağıntı (3.27)'deki ifadenin kullanıldığı belirtilmiştir.

$$t_i = \begin{cases} 0, & if \quad i < k \\ i-k+1, & if \quad k \le i \le M-1 \\ M-1-k+2, & if \quad i > M-1 \end{cases} \tag{3.27}$$

Ancak çalışma [19] için gerçeklenen kod incelendiğinde, düğüm noktalarını elde etmek için $i = 1, \ldots, M+k$ olmak üzere, Bağıntı (3.27) yerine (3.28)'in kullanıldığı gözlenmiştir:

$$t_i = i - 1 \tag{3.28}$$

[19]'da eğri derecesinin (k) KB skoru tahmini üzerindeki etkisi incelenmiştir. Eğri derecesi 1'den 5'e kadar birer birer artırılarak bu inceleme yapılmıştır. En büyük ilerlemenin eğri derecesi k'nın 1'den 2'ye geçtiğinde kaydedildiği belirtilmiştir. $k > 3$ iken önemli bir ilerleme olmadığı belirtilmiştir. Hücre sayısını $M \ge k+1$'den başlatıp birer birer artırarak 10'a kadar devam etmişlerdir ve hücre sayısının KB tahmini üzerindeki etkisini de incelemişlerdir. Hücre sayısının makul bir şekilde seçildikten sonra sonuç KB tahmini üzerinde pek bir etkisinin olmadığını öne sürmüşlerdir. Hücre sayısı etkisini incelerken (M) neden 10 gibi küçük bir değerde incelemeyi kestiklerinin sebebini açıklamamışlardır. Bu nedenle bu değerin rastgele seçildiği düşünülmüştür ancak 10 gerçek GAÇ uygulamalarında kullanılmak için oldukça küçük bir hücre sayısıdır.

Bu tez çalışmasında da B-spline kestirimcisi üzerinde k eğri derecesinin etkisi büyük ve küçük boyutlu yapay veri kümeleri kullanılarak incelenmiştir. Eğri derecesi 1'den başlatılıp 10'a kadar artırılmıştır. Ayrıca Copula Dönüşümü (CD) isimli bir ön işlemin eğri dereceleri ve B-spline yönteminin başarımı üzerindeki etkisi de incelenmiştir. CD'nin bilgimiz dahilinde BS kestirimcisinin başarımı üzerindeki etkisi literatürde daha önce

incelenmemiştir. Bununla ilgili bir konferans yayını da tez çalışması kapsamında sunulmuştur [29]. Bu incelemenin deneysel sonuçları Bölüm 6.7'de verilmiştir.

Bunlara ilaveten, kestirimcilerin ortak karşılaştırılması ve değerlendirilmesi esnasında k eğri derecesi 2 ve 3 olarak kullanılmıştır.

3.3.10 Çekirdek Yoğunluk Kestirimcisi (ÇYK)

Çekirdek Yoğunluk Kestirimcisi (ÇYK) biyoenformatik alanında çeşitli çalışmalarda kullanılmıştır [86]-[90]. Bu kestirimcinin mikroçip (microarray) veri analizinden elde edilen gen ifade veri kümelerinin KB skorlarını tahmin etmede kullanımı, ARACNE (Algorithm for the Reconstruction of Accurate Cellular Networks) GAÇ algoritması [1] ile popüler olmuştur. Genlerin örnekleri kullanılarak, herbir gen için bireysel ve her bir gen ikilisi için birleşik olasılık dağılım fonksiyonları ÇYK ile ifade edilmektedir. Bireysel ve birleşik olasılık dağılımları kullanılarak elde edilen KB skoru Bağıntı (3.32)'de verilmiştir.

ÇYK'lar $f(x)$ fonksiyonunu tahmin ederken, bu yoğunluk fonksiyonundan seçilen $x_1, x_2, ..., x_M$ gözlemlerini (örneklerini) kullanır. $K_h(\cdot)$: ölçeklenmiş çekirdek fonksiyonu, $K(\cdot)$: çekirdek fonksiyonu, M : gözlem adedi, h : çekirdek bant genişliği olmak üzere, ÇYK tanımı:

$$\hat{f}_h(x) = \frac{1}{M}\sum_{i=1}^{M} K_h(x - x_i) = \frac{1}{Mh}\sum_{i=1}^{M} K\left(\frac{x - x_i}{h}\right) \tag{3.29}$$

şeklinde olur. En sık kullanılan çekirdek fonksiyonu Gauss fonksiyonudur. ARACNE'de [1] ve bu tez çalışmasında da çekirdek fonksiyonu olarak Gauss fonksiyonu seçilmiştir. Fonksiyonun bant genişliği parametresi olan h, $h > 0$ koşulunu sağlamalıdır. Ölçeklenmiş çekirdek fonksiyonu $K_h(x) = \frac{1}{h}K\left(x/h\right)$ şeklinde de tanımlanabilir. ÇYK'lar MM ve CS gibi histogram tabanlı yöntemlere benzerdir. Histogram tabanlı yöntemlerde veri kümesindeki her bir gözlem noktası için belli bir birim yükseklikte üstüste çizgiler konulur ve bu çizgiler örneklerin dağılımını bulmak için toplanır. Gaussian çekirdek fonksiyonu kullanan ÇYK'da ise her bir örnek için belli bir varyansı olan bir Gauss (normal) fonksiyonu oturtulur. Daha sonra bu normal fonksiyonlar toplanarak veri

kümesi ile ilgili sonuç dağılımı elde edilir. Rastgele değişkenler için histogram yöntemlerine nazaran ÇYK yaklaşımı daha hızlı bir şekilde gerçek olasılık dağılımına yakınsayabilir. Ayrıca, ÇYK ile tahmin edilen olasılık yoğunluk fonksiyonu, histogram yöntemleri ile elde edilen tahminden daha yumuşak (smooth) geçişlere sahiptir. Bu nedenle ÇYK'lar olasılık dağılımı tahmininde daha çok tercih edilirler.

Daha önce belirtildiği gibi çekirdek fonksiyonu olarak standart sapması h olan bir Gauss fonksiyonu kullanılmıştır. Böylece rastgele değişkenlerin bireysel olasılık yoğunluk fonksiyonlarının tahmini $\hat{f}(x)$:

$$\hat{f}_h(x) = \frac{1}{M}\frac{1}{\sqrt{2\pi}h}\sum_i \exp\left(-\frac{(x-x_i)^2}{2h^2}\right) \tag{3.30}$$

olur. $\vec{z}_i \equiv \{x_i, y_i\},\ i = 1...M$ olmak üzere, X ve Y değişkenlerinin birleşik olasılık dağılımı tahmini $\hat{f}(x,y) = f(\vec{z})$:

$$f(\vec{z}) = \frac{1}{M}\frac{1}{2\pi h^2}\sum_i \exp\left(-\frac{(\vec{z}-\vec{z}_i)^2}{2h^2}\right) = \frac{1}{M}\frac{1}{2\pi h^2}\sum_i \exp\left(-\frac{(x-x_i)^2+(y-y_i)^2}{2h^2}\right) \tag{3.31}$$

şeklinde olur.

$\hat{f}(x)$, $\hat{f}(y)$ ve $\hat{f}(x,y)$ olasılık dağılımı tahminleri ÇYK ile bulunduktan sonra KB skoru:

$$KB(X,Y) = \frac{1}{M}\sum_i \log\left(\frac{f(x_i,y_i)}{f(x_i)f(y_i)}\right) \tag{3.32}$$

ile elde edilir.

Bu çalışmada Gauss çekirdek fonksiyonunda kullanılan h bant genişliği parametresinin en uygun değerinin ne olacağı incelenmiştir, Ek-A'te bu inceleme bulunabilir. Bu tez çalışmasında da h için Ek-A'da Bağıntı (A.8)'te verilen ifade kullanılmıştır.

3.3.11 K-En Yakın Komşuluk (KEYK) Yöntemi ile Doğrudan KB Skoru Kestirimi

K-en Yakın Komşuluk (KEYK) yöntemiyle doğrudan KB skoru kestirimi Kraskov vd. tarafından önerilmiştir [30]. Doğrudan KB skoru kestirimine geçmeden önce KEYK yöntemi ile entropi tahmininin nasıl yapıldığını kısaca inceleyelim.

KEYK entropi kestirimcisi literatürde çeşitli çalışmalarda kullanılmıştır [30], [31], [91]-[95]. Bu kestirimci de KB skoru kestirimi için $H(X)$ ve $H(Y)$ bireysel entropileri ve $H(X,Y)$ birleşik entropisinin bulunmasını gerektirir. Bu entropilerden KB skoru:

$$KB(X,Y) = H(X) + H(Y) - H(X,Y) \tag{3.33}$$

ile elde edilir.

Kozachenko and Leonenko, her bir veri örneğinin k en yakın komşusuna dayalı parametrik olmayan bir entropi kestirimcisi önermişlerdir [93].

x_i örneği ve bu örneğin k'ncı komşusu arasındaki mesafeye $\in(i)/2$ diyelim. N : örnek sayısını, $P_i(\in)$: x_i örneği merkezli ve $\in$ çaplı kürenin yoğunluğunu göstermek üzere; x_i ve k'ncı komşusu arasındaki mesafenin olasılık dağılımı olan $P_k(\in)$ Bağıntı (3.34) ile ifade edilir:

$$\begin{aligned} P_k(\in) &= \frac{(N-1)!}{1!(k-1)!(N-k-1)!}\frac{dp_i(\in)}{d\in}p_i^{k-1}\times(1-p_i)^{N-k-1} \\ &= k\binom{N-1}{k}\frac{dp_i(\in)}{d\in}p_i^{k-1}(1-p_i)^{N-k-1} \end{aligned} \quad . \tag{3.34}$$

x_i örneğine $\in(i)/2$ mesafesinden daha yakın $k-1$ adet veri örneği bulunmalıdır. Benzer şekilde x_i örneğine $\in(i)/2$ mesafesinden daha uzak $N-k-1$ adet veri örneği bulunmalıdır.

$\psi(\cdot)$ digamma fonksiyonu olmak üzere; Bağıntı (3.34)'ten $\log p_i(\in)$ ifadesinin beklendik değeri:

$$\begin{aligned} E[\log p_i(\in)] &= \int_0^\infty P_k(\in)\log p_i(\in)d\in = k\binom{N-1}{k}\int_0^1 p^{k-1}(1-p)^{N-k-1}\log p\,dp \\ &= \psi(k)-\psi(N) \end{aligned} \tag{3.35}$$

şeklini alır. $E[\log p_i(\in)]$ ifadesi x_i haricindeki tüm $N-1$ adet örnek için elde edilir. d-boyutlu Öklit uzayında birim kürenin hacmi $c_d = \pi^{d/2}\Big/\left(2^d\,\Gamma\left(1+\frac{d}{2}\right)\right)$ iken; x_i merkezli, $\in/2$ yarıçaplı, d-boyutlu bir S küresi düşünelim. Bu S küresinin hacmi:

$$V_{\in} = c_d \in^d = \frac{\pi^{d/2}(\in/2)^d}{\Gamma\left(1+d/2\right)} \tag{3.36}$$

olur [30], [93]. $\mu(x_i)$: x_i örneği için sabit yoğunluk olmak üzere; Bağıntı (3.36)'dan:

$$p_i(\in) \approx V_{\in}\mu(x_i) = c_d \in^d \mu(x_i) \tag{3.37}$$

elde edilir. Bağıntı (3.35) ve (3.37)'den:

$$\log \mu(x_i) \approx \psi(k) - \psi(N) - dE(\log \in) - \log c_d \tag{3.38}$$

ifadesini buluruz. $H(X) = -\int \mu(x)\log \mu(x)dx$, Bağıntı (3.37) ve (3.38)'i birleştirerek, $H(X)$ bireysel entropisini Bağıntı (3.39)'daki gibi elde edebiliriz:

$$H(X) = -\psi(k) + \psi(N) + \log c_d + \frac{d}{N}\sum_{i=1}^{N} \log \in (i) \tag{3.39}$$

Yukarıda belirtildiği gibi $\psi(\cdot)$ digamma fonksiyonudur, yani gamma fonksiyonunun türevidir:

$$\psi(x) = \Gamma(x)^{-1}\frac{d\Gamma(x)}{dx} \tag{3.40}$$

Bu fonksiyon, $\psi(1) = -\gamma$ ve $\gamma = 0.5772156\ldots$ olmak üzere özyinelemeli bir şekilde elde edilebilir:

$$\psi(x+1) = \psi(x) + 1/x \tag{3.41}$$

Örnek sayısının (N) çok büyük olduğu durumlarda bu fonksiyon aşağıdaki gibi elde edilebilir:

$$\psi(x) = \log x - \frac{1}{2x} \tag{3.42}$$

$H(Y)$ bireysel entropisi de benzer şekilde Bağıntı (3.39)'dan elde edilir. $H(X,Y)$ birleşik entropisini bulmak için rastgele değişenlerin x_i ve y_i örnekleri tek bir $z_i = (x_i, y_i)$ noktası olarak kabul edilir. Daha sonra z_i noktasına en yakın k'ncı örnek aranır. Artık z_i noktası ve bu noktanın k'ncı komşusu arasındaki mesafe $\in(i)/2$ ile ifade edilir. Ayrıca boyut iki ile çarpılmalıdır çünkü $d_z = d_x + d_y$ şeklindedir. Genomik

veri kümelerinin kullanıldığı uygulamalarda $d_z = 2$ olur. Bu şekildeki uygun değişikliklerin ardından $H(X,Y)$ birleşik entropisi de Bağıntı (3.39)'dan elde edilir. En sonunda da KB skoru Bağıntı (3.33)'ten elde edilir.

KEYK entropi kestirimcisi ile ilgili giriş yaptıktan sonra [30]'da önerilen doğrudan KB skoru tahmin eden KEYK yaklaşımlarını inceleyebiliriz. [30]'da entropi tabanlı KEYK kestirimcisinin entropi tahmin hatasından kaynaklanan fazladan sapmaya sahip olduğu, doğrudan KB skoru tahmini yapıldığında kullanıcıların bu sapmadan kurtuldukları belirtilmiştir. Doğrudan KB skoru tahmin eden ilk yöntemi: $KB^{(1)}(X,Y)$ ile, ikincisini: $KB^{(2)}(X,Y)$ ile gösterelim. İlk yaklaşımda X ve Y değişkenlerinin bireysel dağılımı elde edilirken $x_j \le x_i \pm \in(i)/2$ ve $y_j \le y_i \pm \in(i)/2$ koşullarını sağlayan noktalar sayılarak sırasıyla $n_x(i)$ ve $n_y(i)$ değerleri elde edilir. Buradaki $\in(i)/2$: (x_i, y_i) noktasının k'ncı komşusu ile olan mesafesidir. Aslında $\in(i)$ değeri, $\in_x(i)$: x boyutundaki mesafe, $\in_y(i)$: y boyutundaki mesafe olmak üzere bunlardan büyük olanı olarak alınır. Dolayısıyla $\in(i)/2$ mesafesi, x_i ile bu noktanın sabit bir k'ncı komşusu arasındaki mesafe değil, bu nokta ve bu noktanın $[n_x(i)+1]$'nci komşusu arasındaki mesafe haline gelir. Bu durumda Bağıntı (3.39), aşağıdaki hale gelir:

$$H(X) = -\frac{1}{N}\sum_{i=1}^{N}\psi[n_x(i)+1] + \psi(N) + \log c_{d_X} + \frac{d_X}{N}\sum_{i=1}^{N}\log \in(i) \tag{3.43}$$

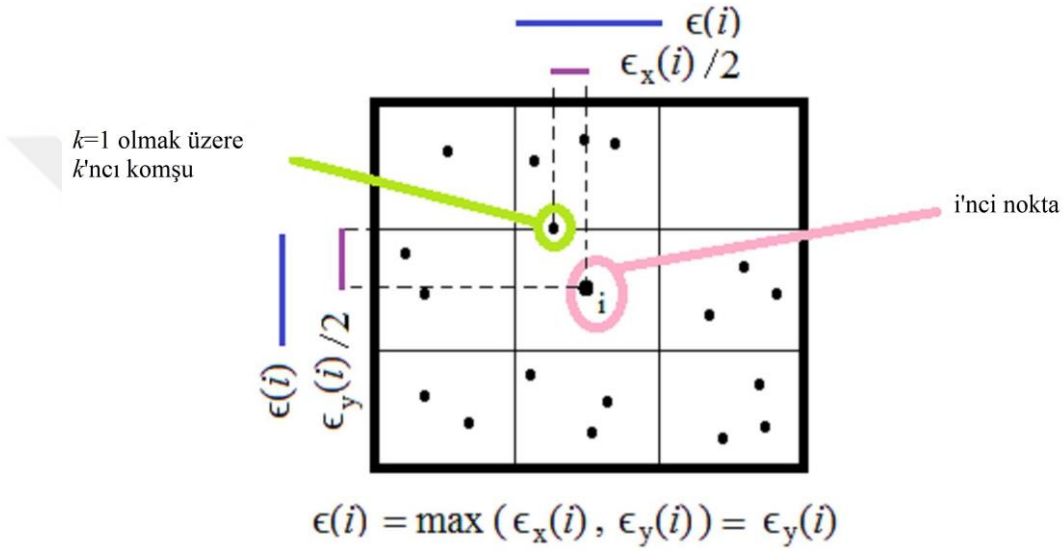
Benzer şekilde, y alt uzayı için de sabit bir k'ncı komşu değil, $y_j \le y_i \pm \in(i)/2$ koşulunu sağlayan $\lfloor n_y(i)+1 \rfloor$'nci komşu ile olan mesafe kullanılır. y boyutu için Bağıntı (3.43)'te yapılan uygun değişiklikler ile $H(Y)$ elde edilir. Bağıntı (3.33)'te yerine konulduğunda, $KB(X,Y)$ skoru doğrudan aşağıdaki ifade halini alır:

$$KB^{(1)}(X,Y) = \psi(k) - \frac{1}{N}\sum_{i=1}^{N}\{\psi[n_x(i)+1] + \psi[n_y(i)+1]\} + \psi(N) \tag{3.44}$$

$\psi(\cdot)$ digamma fonksiyonu, Bağıntı (3.41) ve (3.42)'de verilmiştir.

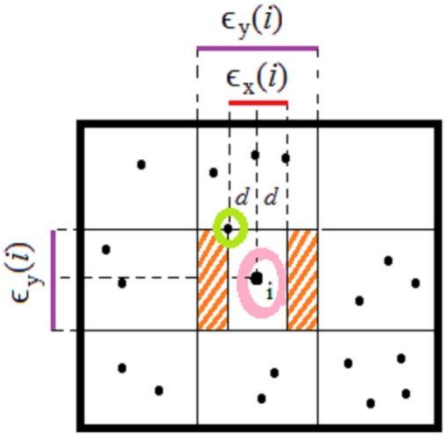
Şekil 3.7'de $n_x(i)$ ve $n_y(i)$ değerlerinin nasıl elde edileceğine dair bir örnek verilmiştir. i'nci nokta pembe daire içine alınmıştır. Bu noktanın $k=1$ olmak üzere k'ncı yakın

komşusu yeşil daire içinde gösterilmektedir. Buna göre x boyutundaki bölgede $n_x(i)=7$ ve y boyutundaki bölgede $n_y(i)=6$ olarak bulunmuştur. $\in(i)$ değeri, $\max\{\in_x(i),\in_y(i)\}$ olarak seçileceği için ve $\in_y(i)$ değeri $\in_x(i)$'ten büyük olduğu için $\in(i)=\in_y(i)$ olur. Bu durumda, x boyutunun komşuluk bölgesi içindeki $n_x(i)$ adet noktanın bir kısmı aslında bu bölgede olmayı hak etmemektedirler. Dolayısıyla $n_x(i)$ değerinin düzeltilmeye ihtiyacı vardır. $KB^{(2)}(X,Y)$ yöntemi ile bu durum düzeltilmiştir [30].



Şekil 3. 7 $KB^{(1)}(X,Y)$ yöntemi ile $n_x(i)$ ve $n_y(i)$ değerlerini bulma

Şekil 3.8'de $KB^{(2)}(X,Y)$ yaklaşımı ile dağılımın nasıl düzeltildiği gösterilmiştir. Buna göre, $KB^{(2)}(X,Y)$ yöntemi sayesinde dikdörtgen şeklindeki turuncu taralı bölgelerdeki noktalar dağılıma dahil edilmeyerek daha başarılı bir tahmin yapılması sağlanmıştır. Eğer $KB^{(1)}(X,Y)$ yönteminin yaptığı gibi, bu taralı bölgelerdeki noktalar da sayılsaydı $H(Y)$ entropi tahmini hatalı olacaktı. Sonuç olarak, $KB^{(2)}(X,Y)$ yöntemi her iki ekseni (x ve y) ortak ve tek bir $\in(i)/2$ mesafesine göre değil, ayrı ayrı $\in_x(i)/2$ ve $\in_y(i)/2$ mesafelerine göre değerlendirmekte ve $n_x(i)$ ve $n_y(i)$ değerlerini sırayla $x_j \le x_i \pm {}^{\in_x(i)}\!/_2$ ve $y_j \le y_i \pm {}^{\in_y(i)}\!/_2$ ifadeleri ile yapmaktadır.



Şekil 3. 8 $KB^{(2)}(X,Y)$ yöntemi ile $n_x(i)$ ve $n_y(i)$ değerlerini bulma

İkinci yöntem olan $KB^{(2)}(X,Y)$'de z_i noktası ve bu noktanın k'ncı en yakın komşusu arasındaki mesafenin olasılık dağılımı olan $P_k(\in)$, 2-boyutlu bir $P_k(\in_x,\in_y)$ dağılımı halini alır. Ayrıca, artık d-boyutlu, z_i merkezli ve $\in/2$ yarıçaplı bir S küresi içindeki dağılımla değil, d-boyutlu bir Öklit uzayı ile, yani (x_i, y_i) merkezli $\in_x \times \in_y$ ebatında dikdörtgen bir bölge ile ilgilenilmektedir. Bu dikdörtgenin yoğunluğuna q_i dersek, $\log q_i$'nin beklenen değeri:

$$E[\log q_i] = \int_0^\infty \int P_k(\in_x,\in_y)\log q_i(\in_x,\in_y) d \in_x d \in_y = \psi(k) - 1/k - \psi(N) \tag{3.45}$$

olur. Buradan da $KB^{(2)}(X,Y)$ yöntemi ile doğrudan KB skorunu aşağıdaki gibi elde ederiz [30]:

$$MI^{(2)}(X,Y) = \psi(k) - \frac{1}{k} - \frac{1}{N}\sum_{i=1}^{N}\{\psi[n_x(i)+1] + \psi[n_y(i)+1]\} + \psi(N) \tag{3.46}$$

[30]'a göre $H(X)$, $H(Y)$ ve $H(X,Y)$ entropilerini ayrı ayrı tahmin etmemize gerek kalmadığı için bunların tahmin hatasından kaynaklanan sapmadan kurtuluruz. Bu nedenle $KB^{(1)}(X,Y)$ ve $KB^{(2)}(X,Y)$ yöntemlerinin entropi tabanlı KEYK'dan daha başarılı tahminler yaptığı belirtilmiştir. Ayrıca $KB^{(2)}(X,Y)$'in $KB^{(1)}(X,Y)$'den daha başarılı tahminler yaptığı da belirtilmiştir [30]. Çünkü $KB^{(1)}(X,Y)$'de $\in(i) = \max\{\in_x(i), \in_y(i)\}$ olduğu kabul edilir. Ancak x ve y eksenlerinin her ikisini de aynı $\in(i)$ mesafesine göre değerlendirmek, komşuların dağılımını araştırırken

eksenlerden birisi için hatalı dağılım tespit etmemize sebep olur. Ancak x ve y eksenlerini ayrı ayrı $\in_x (i)$ ve $\in_y (i)$ mesafelerine göre değerlendirdiğimizde komşuların dağılımını belirlemerken olası hatalardan kurtulmuş oluruz. Sonuç olarak $KB^{(2)}(X,Y)$, $KB^{(1)}(X,Y)$'den daha başarılı bir tahmin yapabilir.

Bu çalışmada da kullanılan $KB^{(2)}(X,Y)$ literatürde çok sayıda araştırmacı tarafından tercih edilmiştir [31], [32], [44], [96]-[98]. Suzuki vd. KEYK yönteminde komşuluk parametresi k'nın seçiminin bir problem olduğunu öne sürmüştür [44]. Ancak Kraskov vd. bu parametrenin ÇYK'daki bant genişliği parametresi olan h gibi seçilebileceğini belirtmiştir [30]. ÇYK'da h parametresi küçük seçilirse istatistiksel hatanın büyüyeceği bu nedenle h'ı veri kümesi saçılımının $1/2$ veya $1/3$'ü olarak seçmek gerektiği belirtilmiş, k'nın da benzer mantıkla yaklaşık $\sqrt{k/N} \approx 0.4$ sağlanacak şekilde seçilebileceği ifade edilmiştir (N : örnek sayısıdır) [30]. Bu tez çalışmasında da k değeri bu öneri doğrultusunda seçilmiştir.

3.3.12 En Küçük Kareler Karşılıklı Bilgi (EKKB) Kestirimcisi

En Küçük Kareler Karşılıklı Bilgi (EKKB) kestirimcisi aslında bir öznitelik seçimi yöntemidir ve KEYK $KB^{(2)}(X,Y)$ yönteminde olduğu gibi doğrudan KB skorunu elde edebilmektedir. EKKB çeşitli uygulamalarda kullanılmıştır [43], [44], [99]-[101]. Suzuki vd. bu özellik seçimi yöntemini gen ikilileri arasındaki etkileşim skorunu ölçmede kullanmışlardır [44]. [44]'te ÇYK'nın naif bir yöntem olduğu ve karmaşıklığı fazla olduğu için pratikte KB kestirimi uygulamalarında kullanımının pek uygun olmadığı belirtilmiştir. KB kestirimi için bir başka alternatif olan KEYK'nın gerçeklenmesinin ÇYK'dan daha kolay olduğu ancak k komşuluk parametresinin seçiminin veri kümesi dağılımına göre adaptif olarak yapılamamasının bu yöntemin dezavantajı olduğunu belirtmişlerdir [44]. [44]'te EKKB'nin, ÇYK'daki gibi yoğunluk tahmini ile veya KEYK'taki gibi k parametresinin seçimi gibi zorluklarla uğraşmadığı için diğer yöntemlerden daha başarılı olabileceği belirtilmiştir. EKKB'de aşağıdaki yoğunluk oranı etkin bir şekilde modellenmelidir:

$$\hat{w}(x,y) = \frac{p(x,y)}{p(x)\,p(y)} \tag{3.47}$$

Karesel kayıba (squared loss) bağımlı olarak KB tanımı aşağıdaki gibidir:

$$KB_{karesel}(X,Y) = \iint \left(\frac{p(x,y)}{p(x)p(y)} - 1 \right)^2 p(x)p(y)dxdy \tag{3.48}$$

EKKB'nin amacı Bağıntı (3.48)'de verilen karesel kaybın tahmin edilmesidir. Bunun için de sadece Bağıntı (3.47)'de verilen yoğunluk oranı kullanılmıştır. Yoğunluk oranı ifadesi olan $\hat{w}(x,y)$'yi Bağıntı (3.48)'de yerine koyarsak:

$$KB_{karesel}(X,Y) = \sum_{i,j=1}^{n} \left(\hat{w}(x_i, y_j) - 1 \right)^2 \tag{3.49}$$

elde ederiz. $\hat{w}(x,y)$ ifadesini modellemek için EKKB yöntemi doğrusal bir denklem sisteminin çözümünü gerektirir. Modelin doğrusal olduğunu varsaydığımız için EKKB yöntemi parametrik bir kestirimcidir. $\hat{w}(x,y)$'yi temsil eden doğrusal model:

$$\hat{w}_a(x,y) = \boldsymbol{\alpha}^{\mathrm{T}} \varphi(x,y) \tag{3.50}$$

şeklinde olur. Buradaki $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_b)$ parametreleri veri kümesindeki örneklerden elde edilir, $\varphi(x,y) = (\varphi_1(x,y), \varphi_2(x,y), ..., \varphi_b(x,y))^{\mathrm{T}}$ parametreleri ise temel (basis) fonksiyonlardır. Temel fonksiyonların her birisi b-boyutlu birer vektördür ve bu vektörlerin elemanları pozitif gerçel sayılardır. Bu fonksiyonların değerleri de $\boldsymbol{\alpha}$ parametreleri gibi x_i ve y_i veri örnekleri kullanılarak elde edilir. $\varphi(x,y)$ fonksiyonu olarak herhangi bir çekirdek fonksiyonu seçilebilir. Bu fonksiyonlara bir örnek Bağıntı (3.57)'de verilmiştir.

Bağıntı (3.50)'deki modelin parametrelerini belirlemek için, C bir sabit olmak üzere, $J(\boldsymbol{\alpha})$ maliyet fonksiyonunun en aza indirgenmesi gerekmektedir:

$$J(\boldsymbol{\alpha}) = J_0(\boldsymbol{\alpha}) - C = \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{H}\boldsymbol{\alpha} - \mathbf{h}^{\mathrm{T}}\boldsymbol{\alpha} \tag{3.51}$$

$\boldsymbol{\alpha}$ parametresini elde edebilmek için Bağıntı (3.51)'den aşağıdaki ifade elde edilir:

$$\tilde{\boldsymbol{\alpha}} = \arg\min_{\mathbf{a} \in R^b} \left[\frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\hat{\mathbf{H}}\boldsymbol{\alpha} - \hat{\mathbf{h}}^{\mathrm{T}}\boldsymbol{\alpha} + \lambda\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\alpha} \right] \tag{3.52}$$

Bağıntı (3.52)'deki toplam teriminin son ifadesi olan $\lambda\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\alpha}$ düzenleme terimidir. $b\times b$'lik $\hat{\mathbf{H}}$ matrisi ve $b\times 1$'lik $\hat{\mathbf{h}}$ vektörü aşağıdaki bağıntılarla elde edilebilir:

$$\hat{\mathbf{H}}=\frac{1}{n^2}\sum_{i,j=1}^{n}\varphi(x_i,y_j)\varphi(x_i,y_j)^{\mathrm{T}} \tag{3.53}$$

$$\hat{\mathbf{h}}=\frac{1}{n}\sum_{i=1}^{n}\varphi(x_i,y_j) \tag{3.54}$$

Yukarıda verilenler doğrultusunda, $\mathbf{I}_b$ birim matris olmak üzere; $\tilde{\boldsymbol{\alpha}}$ parametresi aşağıdaki doğrusal modelden elde edilebilir:

$$\tilde{\boldsymbol{\alpha}}=\left(\hat{\mathbf{H}}+\lambda\mathbf{I}_b\right)^{-1}\hat{\mathbf{h}} \tag{3.55}$$

EKKB yönteminin etkinliği ve performansı $\varphi(x,y)$ temel fonksiyonlarının ve λ düzenleme parametresinin seçimine bağlıdır. Model parametrelerinin seçimi çapraz doğrulama yaklaşımı ile yapılabilir. Çapraz doğrulama (ÇD) için veri kümesi $\{Z_k\}_{k=1}^{K}$ şeklinde K farklı alt gruba ayrılır. Her bir k grubu için yoğunluk oranı tahmini olan $\hat{w}_k(x,y)$ ve $\hat{J}^{K-ÇD}$ maliyeti, k grubu dışında geri kalan $\{Z_j\}_{j\neq k}$ gruplarındaki veri örnekleri kullanılarak elde edilir:

$$\hat{J}^{K-ÇD}=\frac{1}{K}\sum_{k=1}^{K}\hat{J}_k^{K-ÇD}. \tag{3.56}$$

Buradan da maliyet fonksiyonunu en küçülten $\boldsymbol{\alpha}$ parametreleri ve $\varphi(x,y)$ temel fonksiyonları elde edilir. $\varphi(x,y)$ temel fonksiyonları için [44]'te Gauss çekirdek fonksiyonu kullanılmıştır. $\{(u_\ell,v_\ell)\}_{\ell=1}^{b}$ noktaları, $\{(x_i,y_i)\}_{i=1}^{n}$ noktalarından rastgele seçilen b farklı merkez noktası olmak üzere Gauss çekirdek fonksiyonu ile:

$$\varphi_\ell(x,y)=\exp\left(-\frac{\|x-u_\ell\|^2}{2\sigma^2}\right)\delta(y=v_\ell) \tag{3.57}$$

elde edilir. Burada $\delta(\cdot)$ gösterge fonksiyonunun değeri, içindeki "$y=v_\ell$" ifadesi doğru ise 1'dir, doğru değilse 0'dır. Ayrıca [44]'te temel fonksiyonların ve $\boldsymbol{\alpha}$ parametrelerinin adedi olan $b=\min(100,N)$ şeklinde seçilmiştir (N örnek sayısıdır).

Bunun dışında [44]'te Gauss çekirdek fonksiyonundaki bant genişliği parametresi (σ: standart sapma) ve λ düzenleme parametresinin deneme yanılma yoluyla seçilebileceği belirtilmiştir. Ancak bu tez çalışmasında görülmüştür ki 100 gen ve 100 örnekten oluşan küçük bir veri kümesi için bile EKKB yönteminin çalışma süresi 10 saati aşmakta, buna rağmen en kötü başarım sonucunu vermektedir. Çalışma süresinin bu küçük veri kümesi için bile bu kadar uzun olması, deneme yanılma yoluyla parametre optimizasyonunu imkansız kılmaktadır. Bu nedenle bu yöntem deneysel sonuçlar bölümünde detaylı olarak belirtildiği gibi daha büyük ve gerçek veri kümeleri için incelemeden çıkartılmış ve değerlendirmeye alınmamıştır.

BÖLÜM 4

GEN AĞI ÇIKARIM (GAÇ) ALGORİTMALARI

Bu tez çalışmasında, Gen Ağı Çıkarımı (GAÇ) veya gen ağı ters mühendisliği şeklinde isimlendirilen uygulamalarda kullanılan etkileşim skoru kestirimcileri incelenmiştir. Genler arasındaki ikili ilişkilerin bir ağ şeklinde görselleştirildiği ve sunulduğu GAÇ uygulamaları, biyoenformatik alanında sıklıkla kullanılmaktadır [17]-[21]. İlaç üretiminde etken madde tespiti, düzenleyen ve düzenlenen genlerin tespiti, genlerin işlevlerinin incelenmesi gibi çalışmalarda GAÇ uygulamalarından faydalanılmaktadır.

GAÇ uygulamalarının genel işleyişini gösteren blok diyagram giriş bölümünde Şekil 1.1'de verilmiştir. Buna göre, mikroçip veri analizi sonucunda elde edilen gen ifade veri kümeleri GAÇ uygulamalarının girdileridir. Matris şeklinde verilen gen ifade veri kümeleri kullanılarak genler arasında etkileşim bulunup bulunmadığı, varsa bu etkileşimin skorunun ne olduğu etkin bir şekilde tespit edilmelidir. Bu işleme genler arası etkileşim skorunun kestirimi denir ve GAÇ uygulamalarının temelini oluşturur. Bu aşama etkin bir şekilde gerçekleştirilemez ise çalışmada hangi GAÇ algoritması kullanılırsa kullanılsın ağ çıkarım performansı düşük olur. Bu adımda kullanılması olası etkileşim skoru kestirimcileri Bölüm 3'te detaylı bir şekilde verilmiştir. Herbir GAÇ algoritması, kendine has bir yaklaşımla bir önceki adımda bulunan aday gen etkileşimlerini inceler ve bunların arasından elenmesi gerektiğine kanaat getirdiği ilişkileri ağdan eler. Eleme işlemi istatistiksel olarak önemsiz (non-significant) bağlantıların tespitini gerektirir. Bunun için de etkileşim skorlarına dair bir eşik değeri belirlenmelidir. Bu tez çalışmasında bu eşik değeri belirlenirken "permütasyon testi" kullanılmıştır. Bu yaklaşım GAÇ algoritmaları anlatıldıktan sonra, Bölüm 4.4'te detaylı bir şekilde verilmiştir.

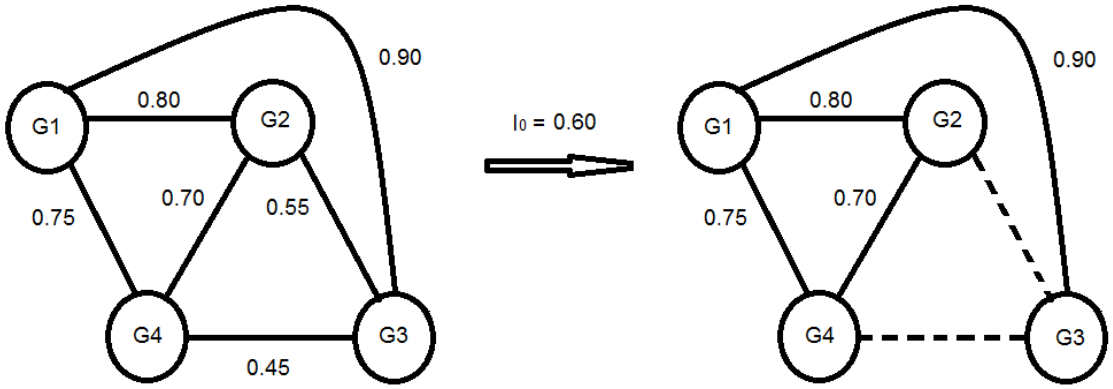
Bu tez çalışmasının amacı daha önceki bölümlerde belirtildiği üzere, farklı GAÇ algoritmalarını incelemekten ziyade, farklı etkileşim skoru kestirimcilerinin GAÇ performansına etkilerini incelemektir. Etkileşim skoru kestirimcilerinin adilane ve genelleştirme yapılabilecek bir şekilde değerlendirilebilmesi amacıyla bu tez çalışmasında gerçeklenen deneylerde çeşitli GAÇ algoritmaları ve çeşitli veri kümeleri kullanılmıştır. Kullanılan GAÇ algoritmaları literatürde popüler olan ve sık kullanılan:

- RelNet (Relevance Networks) [2]
- ARACNE (Accurate Cellular Networks) [1] ve
- C3NET (Conservative Causal Core Networks) [3] yöntemleridir.

Kullanılan GAÇ algoritmalarının üçü de simetrik, yönsüz (undirected) ve KB skoru tabanlıdır. Ayrıca, her biri kendine has bir yaklaşımla ağdaki aday bağlantılardan istatistiksel olarak önemsiz olanlarını eler. Bölüm 4.1, 4.2 ve 4.3'te sırayla RelNet, ARACNE ve C3NET algoritmaları, Bölüm 4.4'te de ilişki skorlarının eşik değerini belirlemede kullanılan permütasyon testi anlatılmıştır.

4.1. Relevance Networks (RelNet)

RelNet, KB tabanlı GAÇ algoritmalarının ilkidir ve diğer KB tabanlı GAÇ algoritmalarının temelini oluşturmaktadır [2]. RelNet'te basit bir işlemle istatistiksel olarak önemsiz olduğu düşünülen aday bağlantılar elenir. Etkileşim kestirimi sonucunda elde edilen ilişki skoru matrisinde bulunan değerler belli bir I_0 eşik değerinin altında ise elenir, değilse ağda bırakılır.



Şekil 4. 1 RelNet GAÇ algoritmasının önemsiz bağlantıları elemesi

Şekil 4.1'de dört genden oluşan küçük bir gen ağı için RelNet'in önemsiz genleri elemesine örnek verilmiştir. Eşik değeri $I_0 = 0.60$ olarak seçilmiş ve bu değerin altındaki skorlar ağdan elenerek sonuç ağı elde edilmiştir. Eşik değeri çeşitli istatistiksel testler ile belirlenebilir. Bu tez çalışmasında, I_0 eşik değeri permütasyon testi kullanılarak belirlenmiştir.

4.2. ARACNE (Accurate Cellular Networks)

ARACNE literatürde sıklıkla kullanılan bir GAÇ algoritmasıdır [1], [17], [18], [34]. Etkileşim skorlarının bulunduğu matris elde edildikten sonra aday bağlantılar arasından istatistiksel olarak önemsiz olanların elenmesi için iki adımdan oluşan işlemler uygulanır.

ARACNE'nin ilk adımında RelNet'in yaptığı gibi belli bir I_0 eşik değerinin altındaki skorlar elenir. İkinci adımında da potansiyel olarak dolaylı (indirect) olduğu düşünülen bağlantıların elenmesi için Veri İşleme Eşitsizliği (VİE - Data Processing Inequality, DPI) işlemi uygulanır. Dolaylı bağlantının ne olduğu ve neden tespit edilerek elenmesi gerektiği Bölüm 3.1'de detaylı bir şekilde anlatılmıştır.

VİE işlemi, g_1 ve g_3 genlerinin doğrudan bağlı olmayıp, g_2 geni üzerinden dolaylı olarak bağlı olmaları durumunda $KB(g_1, g_3) \le \min[KB(g_1, g_2), KB(g_2, g_3)]$ koşulunu sağlamaları gerektiğini varsayar. Bunun için de I_0 eşik değerinden büyük olan her gen üçlüsünü inceler. Gen üçlülerinin ikili kombinasyonları arasından en küçük skorun dolaylı bağlantıdan kaynaklandığını varsayar ve bu skora sahip bağlantıyı ağdan eler. Ancak herbir gen üçlüsü için en küçük skorlu bağlantı ağdan elenirken gerçekte dolaylı bağlantı olmayanlar da yanlışlıkla ağdan elenebilir. Bu nedenle bir tolerans payı bırakılması gerektiği düşünülmüş ve eleme işlemi için Bağıntı (4.1)'de verilen koşulun sağlanıp sağlanmadığına bakılmıştır:

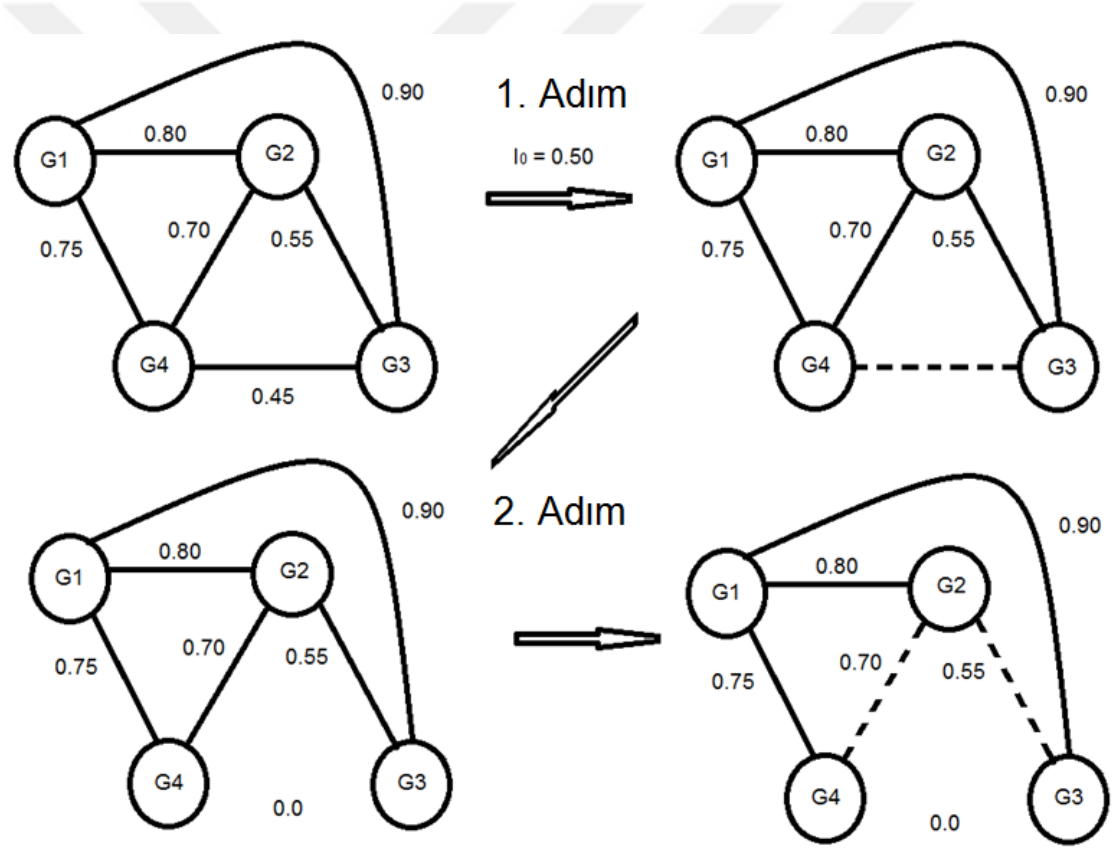
$$KB(g_1, g_3) \le \min[KB(g_1, g_2) \times (1-\tau), KB(g_2, g_3) \times (1-\tau)] \quad (4.1)$$

Buradaki τ: tolerans parametresidir ve [1]'de 0.15 olarak kullanılması tavsiye edilmiştir. ARACNE'nin daha sonraki ve güncel olan versiyonlarında bu parametrenin 0 olarak kullanılması önerilmiştir [102]. Bu tez çalışmasında da $\tau = 0$ olarak alınmıştır.

4.3. C3NET (Conservative Causal Core Networks)

C3NET algoritması da etkileşim skorlarının bulunduğu matrisin önemsiz değerlerini gen ağından budamak için ARACNE gibi iki adımdan oluşan işlemler dizisi uygular [3].

İlk adımda RelNet ve ARACNE'de olduğu gibi belli bir I_0 eşik değerinin altında skora sahip olan bağlantılar elenir. İkinci adımda, potansiyel olarak dolaylı olduğu düşünülen bağlantıların elenmesi amacıyla her bir gen için tüm olası ilişkiler incelenir. Her bir genin sadece en büyük değerli etkileşim skoruna sahip bağlantısı korunur, diğer bağlantıları kopartılır. Tabi bu esnada gen ağının simetrik ve yönsüz yapısının korunmaya devam etmesi gerekmektedir. Şekil 4.2'de C3NET'in işlem adımları dört gen içeren küçük bir ağ için gösterilmiştir.



Şekil 4. 2 C3NET GAÇ algoritmasının önemsiz ve dolaylı bağlantıları elemesi

Şekil 4.2'deki örnekte I_0 eşik değeri 0.5 olarak alınmış, bu değerin altındaki bağlantılar ilk adımda elenmiştir. İkinci adımda da her bir gen için en büyük değerli bağlantılar korunmuş, diğerleri elenmiştir. Örneğin g_1 geni için en büyük skor g_3 geniyle olan bağlantıdır (0.90); g_1 geni için diğerleri elenmiştir. g_2 için en büyük skor g_1 ile olan

bağlantıdır (0.80), g_2 için diğerleri elenmiştir. Ağdaki simetrik yapıyı korumak için g_1-g_2 bağlantısı iki yönlü olarak yeniden kurulmuştur. İncelemeye devam edersek, g_3 için en büyük skor g_1'le olan bağlantıdır (0.90) bu da zaten daha önce bulunmuştu; g_3 için diğerleri elenmiştir. g_4 için en büyük skor g_1'le olan bağlantıdır (0.75); g_4 için diğerleri elenmiştir. Ağdaki simetrik yapıyı korumak için g_1-g_4 bağlantısı iki yönlü olarak yeniden kurulmuştur.

Sonuç olarak C3NET algoritması diğer yöntemlere nazaran daha çok elemeye meyilli bir yapıda olduğu için daha seyrek ancak daha güvenilir bir ağ elde eder. C3NET'in elde ettiği bağlantıların gerçek biyolojik ağda bulunma ihtimali, diğer GAÇ algoritmaların bulduğu bağlantılara nazaran daha yüksektir.

4.4. Permütasyon Testi ile I_0 Eşik Değerini Belirleme

Permütasyon testi I_0 eşik değerini belirlemede kullanılan istatistiksel bir test tekniğidir. Bu yöntem, C3NET GAÇ algoritmasının yazılım paketinde gerçeklenmiş ve kullanılmıştır [3]. Permütasyon testi aşağıdaki sözde kodda tanımlanan işlemleri gerçekleştirerek istatistiksel olarak önemli bağlantıların korunmasını, önemsizlerin elenmesini sağlar:

1. Boş bir S skor vektörü tanımlanır.
2. For (i = 1 to belli_adim_sayisi)
    a. Rastgele değişkenlere (genlere) ait veri örneklerinin bulunduğu veri kümesi matrisindeki değerler rastgele bir şekilde karıştırılır ve yeni veri kümesi elde edilir;
    b. Yeni veri kümesinin KB skorları veya korelasyon skorları bulunur.
    c. Bu skorlar S vektörünün sonuna eklenir.
3. End for
4. S vektörü küçükten büyüğe sıralanır.
5. Bir α önem seviyesi (significance level) değeri belirlenir.

6. S vektörünün boyutu L olmak üzere; bu vektörün $(1-\alpha)L$'ncı elemanı I_0 eşik değeri olarak atanır.

Buna göre permütasyon testinde yapılan işlemleri özetleyecek olursak; belli bir adım sayısı boyunca gen ifade değerlerinin tutulduğu matris rastgele karıştırılarak yeni veri kümeleri elde edilir. Bu veri kümelerinin ilişki skorları bulunup bir vektöre yerleştirilir. Belirlenen istatistiksel bir α önem seviyesine göre bu vektörün belli bir noktasındaki değer ($(1-\alpha)L$ indisindeki değer) I_0 eşik değeri olarak atanır.

Bu tez çalışmasında, veri kümesini karıştırma işlemi için iterasyon sayısı 10; önem seviyesi ise $\alpha = 0.01$ şeklinde seçilmiştir.

BÖLÜM 5

GERÇEKLENEN R YAZILIM PAKETİNİN TANITIMI

Bu tez çalışması kapsamında literatürden seçilen 11 kestirimci ve türevleri R [103] platformunda gerçeklenmiş ve başka araştırmacıların kullanımına hazır hale getirilmiştir. R, açık kaynak kodlu bir platform olup geliştiricilerin oluşturdukları yazılım paketlerini birbirleri ile paylaşmalarına olanak sağlayan bir ortamdır. Yazılım paketi içimde gerçeklenen yöntemler:

- Pearson Korelasyon Katsayısı Değeri (PKD),
- Spearman Korelasyon Katsayısı (SKK),
- Pearson Tabanlı Gaussian (PTG) Kestirimcisi,
- Spearman Tabanlı Gaussian (STG) Kestirimcisi,
- n'nci dereceden Kısmi Pearson Korelasyon Katsayısı (KPK-n),
- HHG (Heller, Heller, Gorfine) Kestirimcisi,
- Miller-Madow (MM) Kestirimcisi
- Chao-Shen (CS) Kestirimcisi
- B-spline Kestirimcisi (BS)
- Çekirdek Yoğunluk Kestirimcisi (ÇYK)
- K-en Yakın Komşuluk doğrudan Karşılıklı Bilgi (KB) Kestirimcisi-2 (KEYK)

kestirimcileri ve bunların türevleridir.

Ayrıca, yazılım paketinde yardımcı işlevler olarak:

- Eş Frekans (EF) ayrıklaştırma yöntemi
- Eş Genişlik (EG) ayrıklaştırma yöntemi
- İstatistiksel olarak önemli olmayan bağlantıların elenmesi işlemi
- Copula Dönüşümü (CD) ön işlemi
- Paralel programlamaya olanak sağlamak için çalışma grubu (cluster) kurulum

işlemleri gerçeklenmiştir.

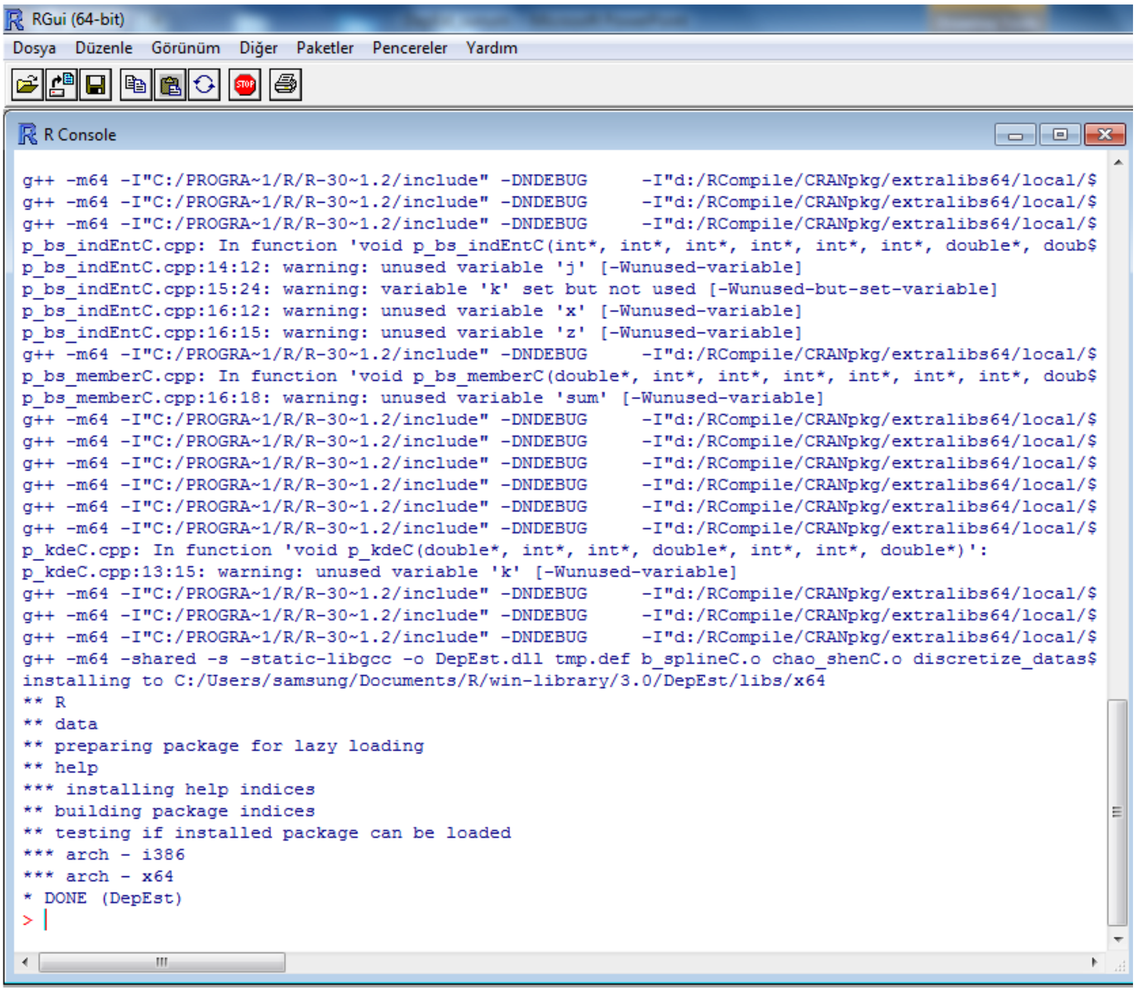
Gerçeklenen yöntemlerden PKD, SKK, KPK-n ve HHG yöntemleri rastgele değişkenler arasındaki ilişki skorunu korelasyon tabanlı bir yaklaşımla elde ederken, diğer yöntemler Karşılıklı Bilgi (KB) tabanlı bir yaklaşım kullanırlar.

R programlama dili yorumlanan (interpreted) bir yapıdadır. Bu nedenle R'ın kendi hazır fonksiyonları hızlı sonuç vermekte iken; kullanıcı-tanımlı fonksiyonlar ve iç içe döngülerin bulunduğu karmaşıklığı fazla olan işlemler oldukça yavaş çalışmakta ve geç sonuç vermektedir. PKD, SKK, PTG, STG ve KPK-n kestirimcileri gerçeklenirken R'ın hazır fonksiyonlarından yararlanıldığı için bu yöntemlerin çalışma süreleri oldukça düşük çıkmaktadır. Bölüm 6'da (deneysel sonuçlar) bulunan sonuç Çizelge'lerinde bu değerler açıkça görülmektedir. Bu beş yöntem haricindeki kestirimciler (BS, CS, ÇYK, HHG, KEYK, MM) kullanıcı tanımlı fonksiyonlar ve döngüler kullanılarak gerçeklenmiş ve çalışma sürelerinin oldukça fazla olduğu gözlenmiştir (Bölüm 6'dan Çizelge 6.1, 6.3 ve 6.5). Bu kestirimcilerin derlenen (compiled) bir yapıda olan C veya C++ gibi bir programlama dili ile gerçeklenip R ortamından çağırıldığı takdirde çalışma sürelerinin azalacağı düşünülmüştür. Bu nedenle bu altı kestirimci C++'da gerçeklenmiştir. Çok çekirdekli bilgisayarların veya birden fazla bilgisayarı içinde barındıran çalışma gruplarının (cluster) biyoenformatik alanında yaygın kullanılması nedeniyle, bu tez çalışmasında da C++'da gerçeklenen altı kestirimci paralel programlamaya uygun şekilde de gerçeklenmiştir. Paralel gerçeklenen kestirimciler dört çekirdekli bir bilgisayarın üç çekirdeği kullanılarak test edilmiş, seri ve paralel çalışma mantığına göre elde edilen çalışma süreleri Bölüm 6'da Çizelge 6.1, 6.3 ve 6.5'te verilmiştir. Dolayısıyla PKD, SKK, PTG, STG ve KPK-n yöntemleri R'da gerçeklenmiş, BS, CS, ÇYK, HHG, KEYK ve MM ise C++'da gerçeklenerek R'dan çağrılmışlardır.

Geliştirilen R yazılım paketinin ismi DepEst (**Dep**endency **Est**imators - Bağımlılık, ilişki Kestirimcileri) şeklinde belirlenmiştir. Paket kurulumu için R ortamında aşağıdaki komutun çalıştırılması gerekmektedir: (Bu işlem bir kez yapılacaktır)

install.packages("PAKETİN_BULUNDUĞU_DİZİN\\DepEst", repos=NULL, type="source")

Böylece paketin kaynak kodundan derleneceğini söylemiş oluruz. Şekil 5.1'de, R'da bu komutun çalıştırılması sonucunda elde edilen ekran görüntüsü verilmiştir.



Şekil 5. 1 Paket kurulumu sonucunda alınan ekran görüntüsü

Artık paket kurulumu yapıldığına göre kurulumun yapıldığı bu bilgisayarda R'ı her açtığımızda *DepEst* kütüphanesini yükleyerek paketteki fonksiyonları kullanabiliriz:

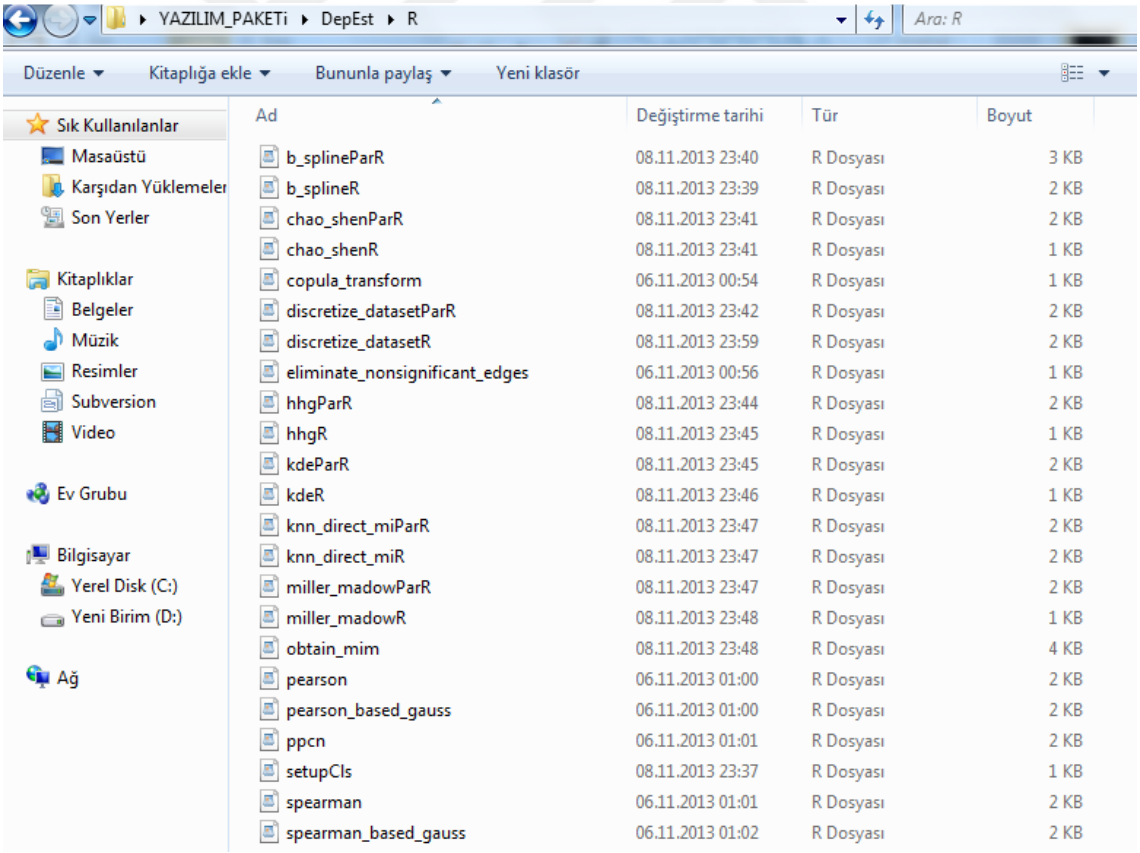
library(DepEst)

Paketin çalışılacak R ortamına yüklenmesi sonucunda alınan ekran görüntüsü Şekil 5.2'de verilmiştir. Paketin bağımlı olduğu diğer kütüphaneler de ortama yüklenmiştir.

```
** testing if installed package can be loaded
*** arch - i386
*** arch - x64
* DONE (DepEst)
> library(DepEst)
Zorunlu paket yükleniyor: minet
Zorunlu paket yükleniyor: infotheo
Zorunlu paket yükleniyor: c3net
Zorunlu paket yükleniyor: igraph
Zorunlu paket yükleniyor: snow
Zorunlu paket yükleniyor: doSNOW
Zorunlu paket yükleniyor: foreach
foreach: simple, scalable parallel programming from Revolution Analytics
Use Revolution R for scalability, fault tolerance and more.
http://www.revolutionanalytics.com
Zorunlu paket yükleniyor: iterators
Zorunlu paket yükleniyor: corpcor
>
```

Şekil 5. 2 Paketin ortama yüklenmesi sonucunda alınan ekran görüntüsü

Pakette bulunan R fonksiyonlarının listesi ve dizin içindeki görünümü Şekil 5.3'te verilmiştir.



Şekil 5. 3 Pakette bulunan R kodlarının dizin içindeki görünümü

Pakette örnek bir gen ifade veri kümesi matrisi bulunmaktadır. Bu veri kümesinin gen ilişkilerini içeren doğrulama gen ağı da pakette mevcuttur. Gen ifade veri kümesi *expdataset*, doğrulama ağı ise *truenetwork* şeklinde isimlendirilmişlerdir. Veri kümesini ve doğrulama ağını R'a yüklemek ve kullanabilmek için aşağıdaki komutlar çalıştırılır:

data(expdataset)

data(truenetwork)

Bu bilgiler ışığında örneğin seri programlama mantığı ile gerçeklenen B-spline kestirimcisini çağıralım:

mim <- b.splineR(expdataset, cop.transform = TRUE, binnum=10, spline.order=2)

İlk argüman gen ifade veri kümesini içeren matristir. İkinci argüman ise veri kümesine CD ön işleminin uygulanıp uygulanmayacağını gösteren boolean tipinde (TRUE/FALSE) bir değerdir. Üçüncü argüman ayrıklaştırma işleminde kullanılacak olan hücre sayısıdır. Son argüman ise BS yöntemindeki eğri derecesini gösteren parametredir. Bu fonksiyon seri kodlama mantığı ile gerçeklenen C++ kodunu çağırarak gen ikilileri arasındaki ilişki skorlarını bulmakta ve bu değerleri kare bir ilişki matrisinde (*mim*) kullanıcıya döndürmektedir.

Aynı veri kümesi için paralel programlama ile gerçeklenen B-spline fonksiyonunu kullanmak istersek aşağıdaki gibi bir kod yazabiliriz:

mim <- b.splineParR(expdataset, cop.transform = TRUE, binnum=10, spline.order=2, num.of.nodes=3, cls.list = NULL)

Yine ilk argüman veri kümesidir. İkinci veri kümesi veriye CD ön işleminin uygulanıp uygulanmayacağını gösterir, üçüncü argüman ayrıklaştırma işleminde kullanılacak hücre adedi, dördüncü argüman yöntemde kullanılacak eğri derecesini belirtir. Beşinci parametre paralel B-spline kodunun kaç çekirdekte çalışacağını göstermektedir. Son parametre ise, paralel kodu çok çekirdekli tek bir bilgisayarda çalıştırmak yerine birden fazla bilgisayarın olduğu bir çalışma grubundaki bilgisayarlarda çalıştırmak istiyorsak bilgisayarların isim listesi bir karakter dizisinde verilir. Paralel kodlama için öncelik *cls.list* parametresindedir. Bu parametre verilmezse veya *NULL* olarak tanımlanırsa paralel kodun çok çekirdekli tek bir bilgisayarda çalıştırılacağı kabul edilir. Paralel bir

kod (Ör: *b.splineParR*) çağırıldığı halde hem bilgisayar isim listesi, hem de kullanılacak çekirdek sayısı belirtilmez ise varsayılan (default) değer olarak, kullanılan bilgisayardaki çekirdek sayısı 1'den fazla ise "çekirdek sayısı - 1" adet çekirdek kullanılır. Örneğin dört çekirdekli bir bilgisayar için kullanılacak çekirdek sayısının varsayılan değeri üçtür.

Bölüm 4'te de anlatıldığı üzere elde edilen gen ilişki matrisindeki (*mim*) istatistiksel olarak önemi olmayan (non-significant) bağlantı skorlarının elenmesi gerekmektedir. Aşağıdaki fonksiyon bu işlemi gerçekleştirmektedir:

mim <- eliminate.nonsignificant.edges(mim)

Doğrulama ağını (*truenetwork*) kullanarak seçilen kestirimcinin RelNet GAÇ algoritmasına göre başarımını değerlendirelim:

relnet <- mim

output.relnet <- checknet(relnet, truenetwork);

RelNet istatistiksel olarak önemsiz bağlantıları elediği için, elemesi yapılmış ilişki matrisi (*mim*) doğrusan *relnet* değişkenine atanmış; C3NET paketinin bir fonksiyonu olan *checknet*() kullanılarak RelNet'in başarımı elde edilmiştir. ARACNE için de değerlendirme yapmak istersek, öncelikle Veri İşleme Eşitsizliği (VİE - Data Processing Inequality, DPI) işleminde kullanılacak *eps* parametresinin değeri belirlenmelidir:

eps=0

aracne.net <- aracne(mim, eps)

output.aracne <- checknet(aracne.net, truenetwork);

C3NET değerlendirmesi için aşağıdaki kod satırları kullanılmalı:

c3net <- c3(mim) # step 2

output.c3net <- checknet(c3net, truenetwork);

Buna göre üç GAÇ algoritması için elde edilen başarım sonuçları Şekil 5.4'te verilmiştir. değerlendirme metriklerinin ne olduğu ve nasıl elde edildikleri Bölüm 6'nın (Deneysel Sonuçlar) girişinde anlatılmıştır.

```
>  data(expdataset)
> mim <-  b.splineR(expdataset, cop.transform = TRUE, binnum=10, spline.order=2)
> mim <- eliminate.nonsignificant.edges(mim)
> data(truenetwork)
> relnet <- mim
> output.relnet <- checknet(relnet, truenetwork);
> eps=0
> aracne.net <- aracne(mim, eps) # step 2
> output.aracne <- checknet(aracne.net, truenetwork);
> c3net <- c3(mim) # step 2
> output.c3net <- checknet(c3net, truenetwork);
> output.relnet
   precision      F-score       recall           TP           FP           FN
3.199200e-02 6.198547e-02 9.922481e-01 1.280000e+02 3.873000e+03 1.000000e+00
> output.aracne
 precision    F-score     recall         TP         FP         FN
 0.4000000  0.4015444  0.4031008 52.0000000 78.0000000 77.0000000
> output.c3net
 precision    F-score     recall         TP         FP         FN
 0.4819277  0.3773585  0.3100775 40.0000000 43.0000000 89.0000000
>
```

Şekil 5. 4 B-spline'ın üç GAÇ algoritmasına göre başarımını gösteren ekran görüntüsü

Paketteki temel fonksiyon olan *obtain.mim*() ile kullanılacak veri kümesi, kestirimci ismi, ön işlem kullanılıp kullanılmayacağı, ismi verilen kestirimcinin paralel mi seri mi çalıştırılacağı, vb. seçenekler belirlenebilir. Fonksiyonun kullanımı, parametre tanımları ve parametrelerin varsayılan (default) değerleri aşağıda verilmiştir:

mim <- obtain.mim(expdataset, estimator="scc", cop.transform=TRUE, num.of.bins=trunc(sqrt(ncol(dataset))), disc.method="eq.freq", bandwidth=0, spline.order=2, num.of.neighbours=0, parallel=0, num.of.nodes=0, cls.list=NULL)

ilk parametre veri kümesidir, ikinci parametre kestirimci yöntemin ismidir. Bu parametrenin varsayılan değeri "scc"dir. *estimator* parametresinin alabileceği değerler:

- "pcc" Pearson Correlation Coefficient (Pearson Korelasyon Katsayısı),
- "scc" Spearman Correlation Coefficient (Spearman Korelasyon Katsayısı),
- "pbg" Pearson-Based Gaussian (Pearson Tabanlı Gaussian),
- "sbg" Spearman-Based Gaussian (Spearman Tabanlı Gaussian)
- "ppcn" n-th Order Pearson Correlation (KPK-n),
- "miller.madow" Miller-Madow Kestirimcisi,
- "chao.shen" Chao-Shen Kestirimcisi,
- "b.spline" B-Spline Kestirimcisi,

- "hhg" Heller,Heller, Gorfine Kestirimcisi,
- "kde" Kernel Density Estimator (Çekirdek Yoğunluk Kestirimcisi),
- "knn" denotes the K-Nearest Neighbourhood (KEYK) Kestirimcisi.

şeklindedir. Üçüncü parametre veri kümesine CD ön işleminin uygulanıp uygulanmayacağını gösterir ve varsayıaln değeri TRUE'dur. Dördüncü parametre ayrıklaştırma işleminde kullanılacak olan hücre sayısıdır, varsayılan değeri örnek sayısının karekköüdür. Beşinci parametre ayrıklaştırma tekniğinin ismidir ve varsayılan değeri Eş Frekans tekniğinin karşılığı olan "*eq.freq*"tir. Altıncı parametre ÇYK yönteminde kullanılacak olan bant genişliği değeri; yedinci parametre BS kestirimcisindeki eğri derecesi; sekizinci parametre ise KEYK yöntemindeki komşu sayısını gösteren değerdir. Dokuzuncu parametre seçilen yöntemin paralel kodunun mu seri kodunun mu çalıştırılacağını gösterir. Değer "0" ise seri kod, "1" ise paralel kod çalıştırılacaktır. Paralel kodlaması gerçeklenmeyen beş yöntemden herhangi biri (PKD, SKK, PTG, STG, KPK-n) için bu parametre 1 seçilirse kullanıcıya uyarı mesajı verilir ve bu yöntemin paralel kodu olmadığı, seri çalıştırılacağı bildirilir. Son iki parametre ise

paralel kodun çok çekirdekli tek bir bilgisayarda mı, çok sayıda bilgisayarın olduğu bir grupta mı çalıştırılacağını belirtir. Öncelik çok sayıda bilgisayarın isim listesini içeren *cls.list* parametresindedir. Bu parametrenin değeri belirtilmediyse paralel kodun çok çekirdekli tek bir bilgisayarda çalıştırılacağı varsayılır. Çekirdek sayısı *num.of.nodes* parametresi ile belirtilir. Belirtilen fonksiyon ve parametreler ile gen ilişki skorlarını içeren matris (*mim*) elde edilir ve kullanıcıya döndürülür.

Pakette bulunan diğer fonksiyonlar, bu fonksiyonların argümanlarının neler olduğu ve argümanların alabileceği değerler her bir fonksiyon için ayrı ayrı dokümante edilmiş ve bir el kitabı (*manual.pdf*) oluşturulmuştur. Bu bölümde birtakım örneklerle paketteki fonksiyonların nasıl kullanılabileceği gösterilmiştir. Ayrıca fonksiyonların listesi (Şekil 5.3'te) verilmiştir. Diğer fonksiyonlar ve detaylı bilgiler için yazılım paketinde bulunan el kitabına bakılabilir.

BÖLÜM 6

DENEYSEL SONUÇLAR

Bu tez çalışmasında GAÇ uygulamalarında kullanılan etkileşim skoru kestirimcilerinin çeşitli açılardan incelemeleri yapılmıştır. Kullanılan etkileşim skoru kestirimcileri:

- Pearson Korelasyon Katsayısı Değeri (PKD),
- Spearman Korelasyon Katsayısı (SKK),
- Pearson Tabanlı Gaussian (PTG) Kestirimcisi,
- Spearman Tabanlı Gaussian (STG) Kestirimcisi,
- n'nci dereceden Kısmi Pearson Korelasyon Katsayısı (KPK-n),
- HHG (Heller, Heller, Gorfine) Kestirimcisi,
- Miller-Madow (MM) Kestirimcisi ve Eş Frekans (EF) Ayrıklaştırma Tekniği (MMEF)
- Miller-Madow (MM) Kestirimcisi ve Eş genişlik (EG) Ayrıklaştırma Tekniği (MMEG)
- Chao-Shen (CS) Kestirimcisi ve Eş Frekans (EF) Ayrıklaştırma Tekniği (MMEF)
- Chao-Shen (CS) Kestirimcisi ve Eş genişlik (EG) Ayrıklaştırma Tekniği (MMEG)
- B-spline Kestirimcisi (eğri derecesi:2) (BS2)
- B-spline Kestirimcisi (eğri derecesi:3) (BS3)
- Çekirdek Yoğunluk Kestirimcisi (ÇYK)
- K-en Yakın Komşuluk doğrudan Karşılıklı Bilgi (KB) Kestirimcisi-2 (KEYK)
- En Küçük Kareler Karşılıklı Bilgi (EKKB) Kestirimcisi

şeklinde 15 farklı yaklaşımdır.

Bu kestirimciler aşağıda belirtildiği gibi çeşitli açılardan değerlendirilmiştir:

- Veri kümesine Copula Dönüşümü (CD) ön işleminin uygulanması durumunda kestirimcilerin çıkarım performanslarının nasıl etkilendiği incelenmiştir,
- Üç farklı GAÇ algoritması kullanılarak kestirimcilerin çıkarım performansı genelleştirilmeye çalışılmıştır,
- Üç yapay ve bir gerçek biyolojik veri kümesi kullanılarak veri kümesinin ve örnek sayısının kestirimcilerin çıkarım performansına etkileri incelenmiştir,
- Kestirimcilerin çıkarım performansları, F-skor ve hassasiyet (precision) ölçütleri kullanılarak değerlendirilmiştir,
- Ayrıklaştırma tekniklerinin entropi-tabanlı yöntemlerin (MM ve CS) performansı üzerindeki etkileri incelenmiştir,
- Ayrıklaştırma işleminde kullanılan en uygun hücre sayısı belirlenmiş (Bölüm 6.8),
- B-spline kestirimcisinin en uygun eğri derecesi belirlenmiştir (Bknz. Bölüm 6.7),

Kestirimcilerin kullanıldığı GAÇ algoritmaları RelNet, ARACNE ve C3NET'tir. Bu algoritmalar Bölüm 4'te anlatılmıştır. Değerlendirmede kullanılan 3 yapay ve bir gerçek biyolojik gen ifade veri kümesi ve bunların doğrulama gen ağları *Escherichia coli* (*E.Coli*) bakterisine aittir. Yapay veri kümeleri SynTReN [27] simülatörü ile üretilmiştir. Üç yapay veri kümesinin tümünde 100 gen vardır. Örnek sayıları ise sırayla 50, 100 ve 1000 şeklindedir. Böylece örnek sayısının çıkarım performansına etkisi gözlemlenebilecektir. Gerçek biyolojik veri kümesi de *E.Coli* bakterisine aittir ve 1146 gen ile 524 örnek içermektedir.

Kestirimcilerin CD ön işleminin kullanılıp kullanılmamasına göre de başarımları değerlendirilmiştir. Bu ön işlemde veri örneklerinin orijinal değerleri yerine sıralama değerleri kullanılır. Öncelikle bu sıralama değerleri (0, 1] aralığında olacak şekilde normalize edilir. Daha sonra normalize edilen değerlerden 0.5 çıkartılır. Böylece CD işlemi uygulanmış veri kümesinin değerleri (-0.5, 0.5] aralığında olur.

Önerilen sistemde kullanılan başarım ölçütleri daha önce de belirtildiği gibi F-skor ve Hassasiyettir. p : Hassasiyet (precision) ve r : Geri Çekilme (recall) olmak üzere F-skor:

$$F = \frac{2pr}{p+r} \tag{6.1}$$

ile elde edilir.

TP: gerçek pozitif (true positive), FP: hatalı pozitif (false positive), FN: hatalı negatif (false negative) olmak üzere p : Hassasiyet ve r : Geri Çekilme skorları Bağıntı (6.2)'den elde edilir.

$$p = \frac{TP}{TP+FP} \text{ ve } r = \frac{TP}{TP+FN} \tag{6.2}$$

Buradaki TP'ler gerçek gen ağında bulunan ve GAÇ uygulaması sonucunda çıkarılan bağlantıların adedi; FP'ler: gerçek ağda bulunmayan ancak GAÇ uygulaması sonucunda çıkarılan bağlantıların adedi; FN: gerçek ağda bulunduğu halde GAÇ uygulamasının çıkartamadığı bağlantıların adedidir.

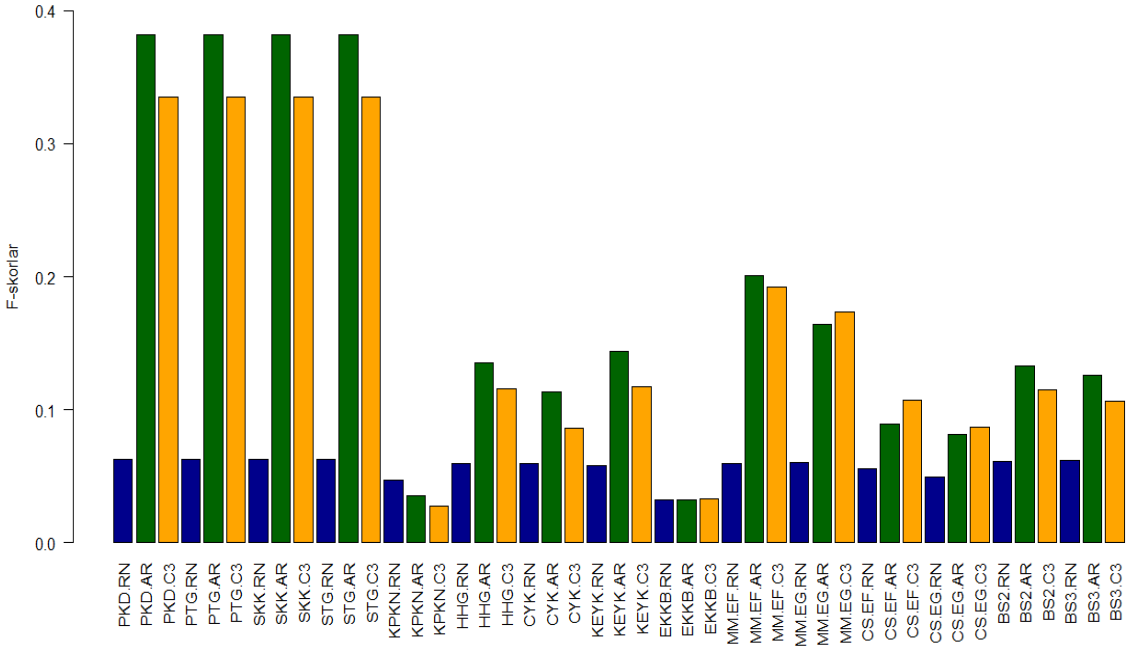
Bölüm 6.1, 6.2 ve 6.3'te sırasıyla ilk üç yapay veri kümesi için CD ön işleminin kullanılması durumunda elde edilen sonuçlar; Bölüm 6.4'te gerçek biyolojik veri kümesi için CD ön işleminin kullanılması durumunda elde edilen sonuçlar; Bölüm 6.5'te CD ön işleminin kullanılması durumunda tüm veri kümeleri için genel bir değerlendirme verilmiştir. Bölüm 6.6'da CD ön işleminin kullanılmaması durumunda tüm veri kümeleri için elde edilen sonuçlar ve ayrıklaştırma tekniklerinin kestirimcilerin performansları üzerindeki değerlendirmeler bir arada verilmiştir. Bölüm 6.7'de B-spline kestirimcisinin en uygun eğri derecesi parametresi incelemesi, 6.8'de ise ayrıklaştırma işlemindeki en uygun hücre sayısı tespiti ile ilgili incelemeler verilmiştir.

6.1. Kestirimcilerin CD Uygulanan İlk Yapay Veri Kümesine göre Karşılaştırılması

Bu tez çalışmasındaki deneylerde kullanılan ilk sentetik veri kümesi 100 gen ve 50 örnek içermektedir ve veri kümesinin gerçek gen ağı daha önce belirtildiği gibi *E.Coli* bakterisinin bir alt ağına aittir. Altay [45]'teki çalışmasında göstermiştir ki GAÇ uygulamalarında sağlıklı bir değerlendirme yapabilmek için veri kümesinde en az 64 adet örnek bulunmalıdır. Bu tez çalışmasında da örnek sayısının kestirimcilerin çıkarım performansına etkisini incelemek istediğimiz için bu değerin altındaki bir örnek sayısı

ile de deney yapılmıştır. [45]'te sunulan sonuçlarla benzer bir durum gözlemiş, örnek sayısının artması durumunda daha başarılı ve genelleştirilebilir sonuçlar elde edilmiştir.

50 örnek içeren bu veri kümesi için CD ön işlemi kullanılması durumunda 15 kestirimcinin üç farklı GAÇ algoritmasına göre elde edilen F-skor değerleri Şekil 6.1 ve Çicelge 6.1'de verilmiştir. Şekil 6.1'de mavi, yeşil ve turuncu çubuklar sırayla RelNet (RN kısaltması), ARACNE (AR) ve C3NET (C3) algoritmalarına karşılık gelmektedir. Ayrıca Çizelgede R kodu, seri ve paralel C++ kodlarının saniye cinsinden çalışma süreleri verilmiştir.



Şekil 6. 1 Birinci sentetik veri kümesine CD ön işlemi uygulanması durumunda kestirimcilerin F-skor ölçütüne göre değerlendirilmesi

CD ön işleminin uygulandığı 50 örnekli birinci sentetik veri kümesinde RelNet GAÇ algoritması kullanılması durumunda F-skor ölçeğine göre elde edilen sonuçlar birbirine yakın ve diğer GAÇ algoritmaları için elde edilen sonuçlardan önemli oranda düşüktür. RelNet diğer GAÇ algoritmalarından farklı olarak dolaylı (indirect) bağlantıları eleyemediği için hatalı pozitif (false positive, FP) bağıntıların adedi yüksektir, bu da F-skoru ciddi bir oranda düşürmektedir. Dolayısıyla, RelNet kullanılması durumunda kestirimciler arasında yapılan karşılaştırma ve değerlendirmeler sağlıklı ve kesin olamamaktadır. Bu durum diğer veri kümeleri için de benzer şekilde gözlenmiştir.

CD işleminin uygulandığı aynı veri kümesi için ARACNE GAÇ algoritmasının kullanılması durumunda en öne çıkan kestirimciler PKD, SKK, PTG ve STG'dir. Bunları MMEF ve MMEG kestirimcileri takip etmektedir. EKKB ve KPKN en kötü performans sergileyen yaklaşımlardır.

CD ön işleminin uygulandığı ilk veri kümesi için C3NET GAÇ algoritmasının kullanılması durumunda da yine aynı dört kestirimci (PKD, SKK, PTG ve STG) en iyi performansı sergilemektedir. Yine MMEF ve MMEG yöntemleri bu 4 kestirimciyi takip etmektedir. En kötü sonuç veren kestirimciler bu GAÇ algoritması için de EKKB ve KPKN'dir.

Çizelge 6. 1 CD kullanılması durumunda birinci veri kümesi için kestirimcilerin F-skorları

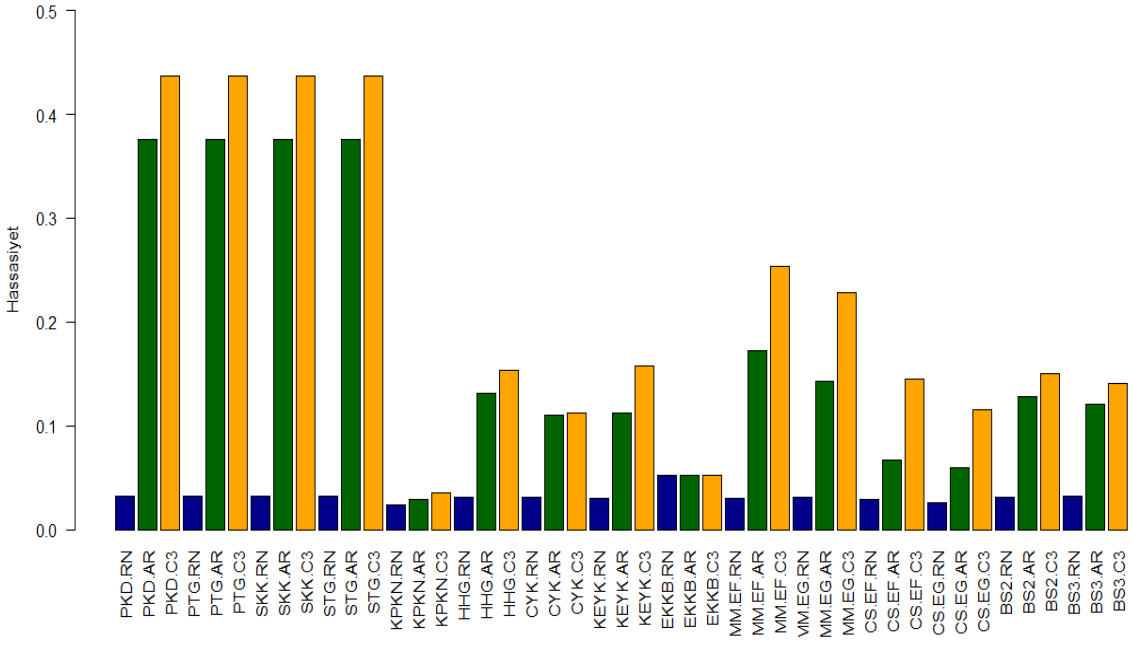
Kestirimci	RelNet F-skoru	ARACNE F-skoru	C3NET F-skoru	R kodu süresi (sn)	Seri C++ kodu süresi	Parallel C++ kodu süresi
PKD	0.0625	0.3817	0.3349	0.01	-	-
PTG	0.0625	0.3817	0.3349	0.04	-	-
SKK	0.0625	0.3817	0.3349	0.02	-	-
STG	0.0625	0.3817	0.3349	0.04	-	-
KPK-n	0.0470	0.0354	0.0279	0.20	-	-
HHG	0.0596	0.1353	0.1159	9911.73	7.65	5.07
ÇYK	0.0596	0.1132	0.0861	164.89	2.20	1.33
KEYK	0.0581	0.1440	0.1171	306.62	1.01	0.68
EKKB	0.0323	0.0323	0.0328	5398.29	-	-
MMEF	0.0593	0.2007	0.1923	2.22	0.05	0.11
CSEF	0.0557	0.0892	0.1073	3.43	0.06	0.11
MMEG	0.0601	0.1645	0.1731	2.04	0.03	0.11
CSEG	0.0497	0.0817	0.0870	4.07	0.05	0.11
BS-2	0.0615	0.1333	0.1148	51.36	0.06	0.13
BS-3	0.0620	0.1259	0.1063	48.50	0.06	0.14

Birinci sentetik veri kümesine CD ön işleminin uygulanması durumunda, 15 etkileşim skoru kestirimcisinin 3 farklı GAÇ algoritmasına göre Hassasiyet skorları dağılımı Şekil 6.2 ve Çizelge 6.2'de verilmiştir. Hassasiyet skoru, çıkartılan ağdaki Gerçek Pozitif (true positive, TP) bağlantıların tüm çıkartılan bağlantılara orantısını veren bir ölçüttür.

Hassasiyet ölçütüne göre RelNet GAÇ algoritmasının kullanılması durumunda elde edilen sonuçlar, RelNet'in F-skor sonuçlarında olduğu gibi birbirine yakın ve diğer GAÇ algoritmaları için elde edilen sonuçlardan önemli oranda düşüktür. Dolayısıyla, RelNet için kestirimcilerin Hassasiyet'e göre karşılaştırılması ve değerlendirilmesi sağlıklı olamamaktadır. Bu durum diğer veri kümeleri için de benzer şekilde gözlenmiştir.

ARACNE algoritmasının kullanılması durumunda, F-skor metriği için öne çıkan aynı 4 kestirimci (PKD, SKK, PTG ve STG), Hassasiyet metriği için de diğer yöntemlere nazaran daha iyi performanslar sergilemektedir. EKKB ve KPKN kestirimcileri de yine en kötü hassasiyet skorlarını veren yöntemlerdir.

C3NET algoritmasının kullanılması durumunda yine aynı 4 kestirimci en başarılı performansı vermektedir. EKKB ve KPKN yöntemleri en düşük hassasiyet skorlarını vermiştir. Bu gözlemler de aynı veri kümesi ve aynı GAÇ algoritması için F-skor sonucu ile uyumlu görülmüştür.



Şekil 6. 2 Birinci sentetik veri kümesine CD ön işlemi uygulanması durumunda kestirimcilerin Hassasiyet ölçütüne göre değerlendirilmesi

50 örnek içeren birinci veri kümesine CD ön işleminin uygulanması durumunda en iyi performans sergileyen kestirimciler F-skor ve hassasiyet ölçütlerinin her ikisine göre de ortaktır. Bunlar PKD, PTG, SKK ve STG yöntemleridir. Her iki performans değerlendirme metriğine (F-skor ve hassasiyet) göre en kötü performans sergileyen yöntemler EKKB ve KPKN'dir. En kötü performans sonuçlarını veren EKKB yönteminde λ ve σ şeklindeki iki parametrenin en uygun şekilde seçilmesi gerekmektedir. Bunun için de zahmetli bir yaklaşım olan deneme yanılma (grid search) yönteminin kullanımı tavsiye edilmiştir [44]. Ancak EKKB'de λ ve σ için deneme yanılma yaklaşımı ile en uygun değerlerin tespiti hesaplama yükünü artırmaktadır. 100 gen ve 50 örnekten oluşan veri kümesi

için 1.5 saat boyunca çalışan EKKB yöntemi, 100 gen ve 100 örnek içeren bir başka veri kümesi için 10 saati aşkın bir süre çalışmaktadır ve bu durum da deneme yanılma yaklaşımı ile parametre optimizasyonunu imkansız kılmaktadır. Daha büyük olan üçüncü sentetik veri kümesi ve gerçek biyolojik veri kümesi için bu çalışma sürelerinin daha da artması, buna rağmen uygun parametre belirlenemediği için düşük performanslar alınması beklenmektedir. Bu nedenlerle EKKB yöntemi ikinci sentetik veri kümesi için de değerlendirilmiş fakat üçüncü sentetik veri kümesi ve gerçek biyolojik veri kümesi için değerlendirilmemiş ve tez çalışması kapsamında değerlendirilecek yöntemlerden ihraç edilmiştir. Ayrıca aynı nedenlerle bu yöntemin seri ve paralel C++ kodları da gerçeklenmemiştir. Son olarak paralel C++ kodları gerçeklenen kestirimciler, 4 çekirdekli bir bilgisayarın 3 çekirdeği kullanılarak test edilmiş, 3 çekirdeğin kullanılması durumunda HHG, ÇYK ve KEYK için çalışma süresinin yarıya yakın azaldığı, MM, CS ve BS için henüz süre açısından kara geçilemediği, seri koda nazaran sürenin arttığı gözlenmiştir.

Çizelge 6. 2 CD kullanılması durumunda birinci veri kümesi için kestirimcilerin Hassasiyet skorları

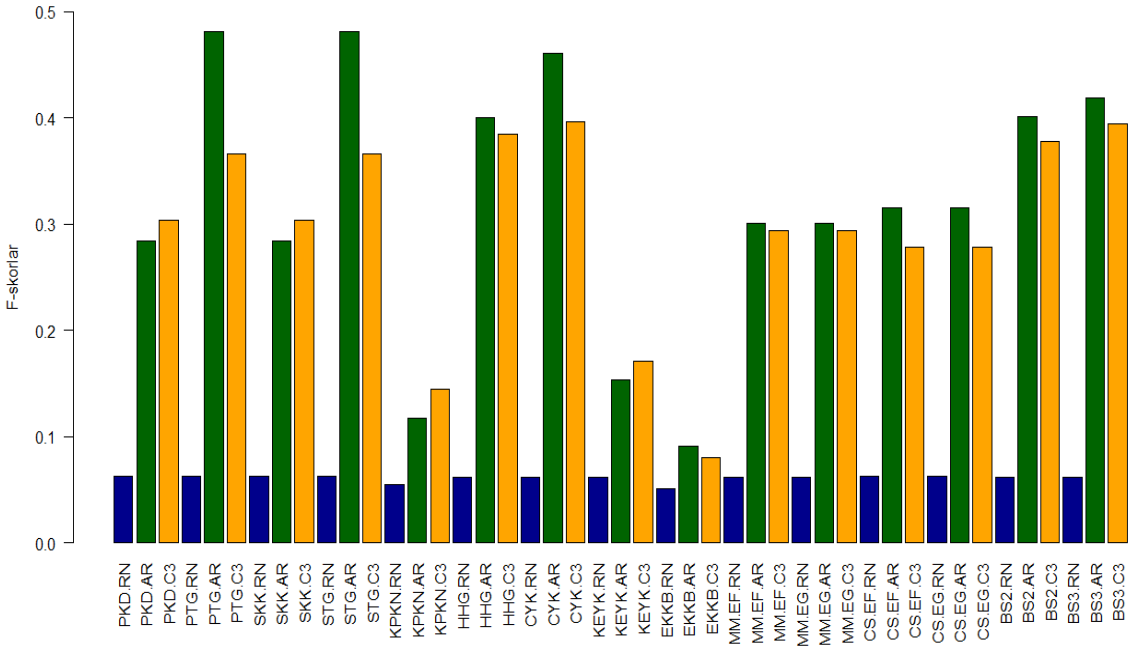
Kestirimci	RelNet Hassasiyeti	ARACNE Hassasiyeti	C3NET Hassasiyeti
PKD	0.0322	0.3759	0.4375
PTG	0.0322	0.3759	0.4375
SKK	0.0322	0.3759	0.4375
STG	0.0322	0.3759	0.4375
KPK-n	0.0243	0.0286	0.0349
HHG	0.0307	0.1314	0.1539
ÇYK	0.0307	0.1103	0.1125
KEYK	0.0300	0.1121	0.1579
EKKB	0.0526	0.0526	0.0526
MMEF	0.0306	0.1722	0.2532
CSEF	0.0287	0.0675	0.1447
MMEG	0.0310	0.1429	0.2279
CSEG	0.0257	0.0592	0.1154
BS-2	0.0317	0.1277	0.1500
BS-3	0.0320	0.1206	0.1410

6.2. Kestirimcilerin CD Uygulanan 2. Yapay Veri Kümesine göre Karşılaştırılması

İkinci sentetik veri kümesi 100 gen ve 100 örnek içermektedir ve gerçek gen ağı *E.Coli* bakterisinin bir alt ağıdır. Daha önce belirtildiği gibi, [45]'teki çalışmada görülmüştür ki

GAÇ uygulamalarında sağlıklı bir değerlendirme yapabilmek için veri kümesinde en az 64 adet örnek bulunmalıdır. Bu nedenle bu veri kümesi için elde edilen sonuçlar bir önceki veri kümesi için elde edilen sonuçlara nazaran daha çok dikkate alınmalıdır ve daha güvenilir olduğu göz önünde bulundurulmalıdır.

Bu veri kümesine CD ön işleminin uygulanması durumunda kestirimcilerin 3 farklı GAÇ algoritması için verdiği F-skor sonuçları Şekil 6.3 ve Çizelge 6.3'te verilmiştir. Bu şekilde de Mavi çizgiler RelNet'e (RN), yeşiller ARACNE'ye (AR), turuncular C3NET'e (C3) karşılık gelmektedir. Ayrıca Çizelge'de kestirimcilerin R kodu, seri ve paralel C++ kodlarının saniye cinsinden çalışma süreleri de verilmiştir.



Şekil 6. 3 İkinci sentetik veri kümesine CD ön işlemi uygulanması durumunda kestirimcilerin F-skor ölçütüne göre değerlendirilmesi

İkinci veri kümesine CD ön işleminin uygulanması durumunda RelNet algoritması için elde edilen sonuçlar birbirine yakın ve diğer GAÇ algoritmaları için elde edilen sonuçlardan önemli oranda düşüktür. Bir önceki alt bölümde de belirtildiği gibi, RelNet diğer GAÇ algoritmalarından farklı olarak dolaylı bağlantıları eleyemediği için hatalı pozitif bağıntıların adedi yüksektir, bu da F-skoru düşürmektedir. Dolayısıyla, RelNet kullanıldığında kestirimciler arasında yapılan karşılaştırma ve değerlendirmeler sağlıklı ve kesin olamamaktadır. Bu durum diğer veri kümeleri için de benzer şekilde gözlenmiştir.

100 örnek içeren ikinci veri kümesine CD ön işlemi uygulanması ve ARACNE algoritması kullanılması durumunda F-skor'a göre değerlendirme yapıldığında 6 kestirimci diğerlerine nazaran öne çıkmaktadır. Bunlardan PTG, STG ve ÇYK daha çok öne çıkmakta; BS2, BS3 ve HHG kestirimcileri de onları çok az bir farkla takip etmektedir. En kötü performanslar EBBK, KPKN ve KEYK yöntemlerinden elde edilmiştir.

Çizelge 6. 3 CD kullanılması durumunda 2. veri kümesi için kestirimcilerin F-skorları

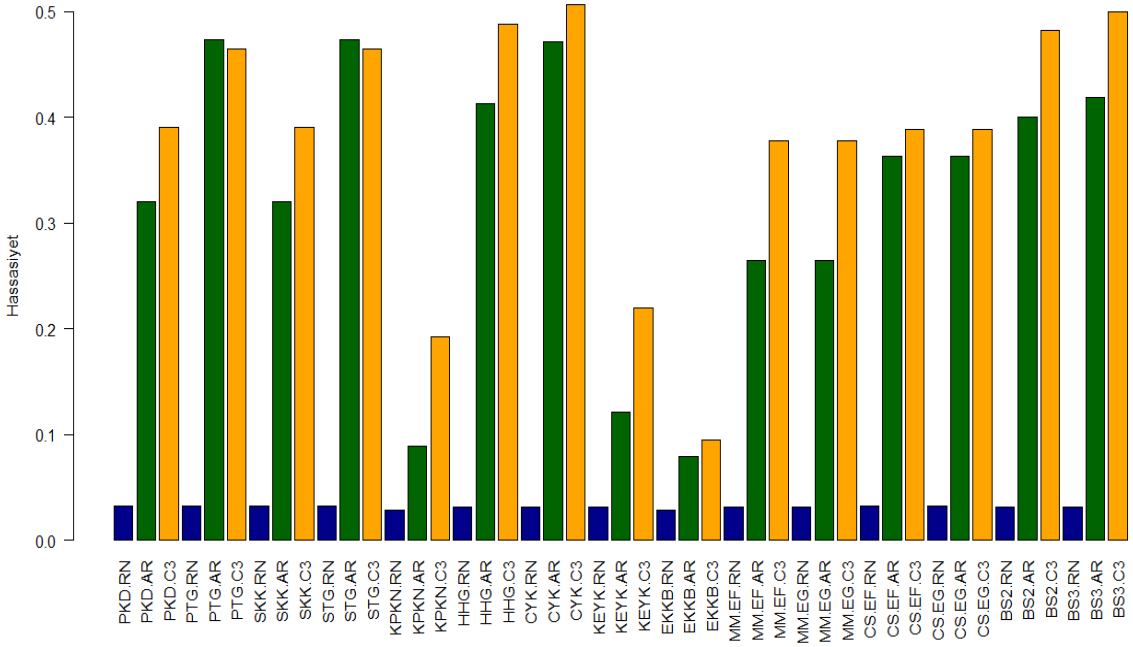
Kestirimci	RelNet F-skoru	ARACNE F-skoru	C3NET F-skoru	R kodu süresi (sn)	Seri C++ kodu süresi	Parallel C++ kodu süresi
PKD	0.0625	0.2838	0.3033	0.02	-	-
PTG	0.0625	0.4809	0.3662	0.03	-	-
SKK	0.0625	0.2838	0.3033	0.03	-	-
STG	0.0625	0.4809	0.3662	0.03	-	-
KPK-n	0.0552	0.1173	0.1449	0.39	-	-
HHG	0.0620	0.4000	0.3850	87919.42	87.49	44.05
ÇYK	0.0620	0.4603	0.3962	563.22	8.46	4.36
KEYK	0.0615	0.1539	0.1706	929.72	7.35	3.50
EKKB	0.0508	0.0906	0.0800	36719.00	-	-
MMEF	0.0614	0.3010	0.2938	3.68	0.06	0.15
CSEF	0.0623	0.3158	0.2786	6.24	0.08	0.17
MMEG	0.0614	0.3010	0.2938	3.70	0.05	0.14
CSEG	0.0623	0.3158	0.2786	6.19	0.09	0.22
BS-2	0.0620	0.4015	0.3774	157.87	0.11	0.74
BS-3	0.0620	0.4186	0.3944	154.98	0.10	0.79

Aynı deneysel düzenekte C3NET algoritmasının kullanılması durumunda F-skora göre aynı 6 kestirimci (PTG, STG, ÇYK, BS2, BS3 ve HHG) öne çıkmaktadır ve başarım değerleri birbirine oldukça yakındır. Bunların arasından ÇYK ve BS3 çok az bir farkla öne çıkmaktadır. En kötü performans sergileyen kestirimciler yine EBBK, KPKN ve KEYK'dir.

Kestirimcilerin üç farklı GAÇ algoritması için Hassasiyet ölçütüne göre değerlendirme sonuçları Şekil 6.4 ve Çizelge 6.4'te verilmiştir. Bölüm 6.1'de belirtildiği gibi Hassasiyet skoru, ağda çıkartılan gerçek pozitif bağlantıların tüm bağlantılara oranı ile elde edilir.

RelNet GAÇ algoritmasının kullanılması durumunda kestirimcilerin hassasiyet skorlarına bakıldığında birbirine yakın ve diğer GAÇ algoritmaları ile elde edilen hassasiyet skorlarından oldukça düşük değerler alındığı görülmektedir. RelNet algoritması F-skor sonucuna göre değerlendirildiğinde de benzer bir sonuç gözlenmiştir. RelNet'in çıkarttığı hatalı pozitif bağlantıların adedi yüksek olduğu için yeterince yüksek örnek

sayısının bulunduğu veri kümeleri için bu algoritmaya bakarak genel bir çıkarım yapmak sağlıklı olmamakta; diğer 2 GAÇ algoritmasının sonuçları daha güvenilir görünmektedir.



Şekil 6. 4 İkinci sentetik veri kümesine CD ön işlemi uygulanması durumunda kestirimcilerin Hassasiyet ölçütüne göre değerlendirilmesi

Çizelge 6. 4 CD kullanılması durumunda ikinci veri kümesi için kestirimcilerin Hassasiyet skorları

Kestirimci	RelNet Hassasiyeti	ARACNE Hassasiyeti	C3NET Hassasiyeti
PKD	0.0322	0.3204	0.3902
PTG	0.0322	0.4737	0.4643
SKK	0.0322	0.3204	0.3902
STG	0.0322	0.4737	0.4643
KPK-n	0.0285	0.0894	0.1923
HHG	0.0320	0.4132	0.4881
ÇYK	0.0320	0.4716	0.5060
KEYK	0.0317	0.1216	0.2195
EKKB	0.0285	0.0796	0.0946
MMEF	0.0317	0.2647	0.3781
CSEF	0.0321	0.3636	0.3889
MMEG	0.0317	0.2647	0.3781
CSEG	0.0321	0.3636	0.3889
BS-2	0.0320	0.4000	0.4820
BS-3	0.0320	0.4186	0.5000

ARACNE algoritmasının kullanılma durumunda Hassasiyet ölçütüne göre en ümit verici kestirimciler PTG, STG, ÇYK, BS2, BS3 ve HHG olarak gözlenmiştir. Bu yöntemler F-skor için de en iyi olduğu gözlenen kestirimcilerdir. Şekil 6.4'te görüldüğü gibi bu 6

kestirimciyi CSEF ve CSEG yöntemleri takip etmektedir. Şekil 6.3'te görüldüğü gibi CS kestirimcisi F-skor sonucuna göre de en iyi 6 kestirimciden hemen sonra gelmesine rağmen CS'nin F-skor sonucu, hassasiyet sonucu kadar en iyi kestirimcilere yaklaşamamıştır. EKKB, KPKN ve KEYK kestirimcileri Hassasiyet skoruna göre de en kötü performansları sergilemiştir.

Aynı deneyde C3NET algoritmasının kullanılması durumunda Hassasiyet ölçütüne göre ÇYK, HHG, BS2, BS3, PTG ve STG en iyi performansları sergileyen kestirimcilerdir. PKD ve SKK da bu 6 kestirimciyi takip etmektedir. EKKB, KPKN ve KEYK yine en kötü performansı veren yöntemlerdir.

İkinci sentetik veri kümesine CD işleminin uygulanması durumunda her iki performans değerlendirme metriğine (F-skor ve Hassasiyet) göre de BS2, BS3, ÇYK, HHG, PTG ve STG kestirimcileri ortak olarak en iyi performansları sergilemişlerdir. En kötü yöntemler de yine her iki metriğe göre ortaktır. Bunlar: EKKB, KPKN ve KEYK şeklindedir. EKKB ilk sentetik veri kümesinde olduğu gibi bu veri kümesi için de en kötü sonuçları vermiş ve çalışma süresi ikinci veri kümesi için 10 saati aşmıştır. Bölüm 6.1'in sonunda değinildiği üzere EKKB'nin iki parametresinin (λ ve σ) en uygun değerleri deneme yanılma (grid search) yöntemi ile tespit edilmelidir. Bu da çalışma süresi bu kadar uzun olan bir yöntem için uygun görülmemiştir. Bu nedenle, EKKB bu tez çalışmasında incelenen kestirimciler arasından ihraç edilmiş, seri ve paralel C++ kodları gerçeklenmemiş; üçüncü sentetik veri kümesi ve gerçek biyolojik veri kümesi için sonuçları verilmemiştir. Son olarak paralel C++ kodları gerçeklenen kestirimciler, yine 4 çekirdekli bir bilgisayarın 3 çekirdeği kullanılarak test edilmiş, 3 çekirdeğin kullanılması durumunda HHG, ÇYK ve KEYK için çalışma süresinin yarıya yakın azaldığı, MM, CS ve BS için henüz süre açısından kâra geçilemediği, seri koda nazaran sürenin arttığı gözlenmiştir.

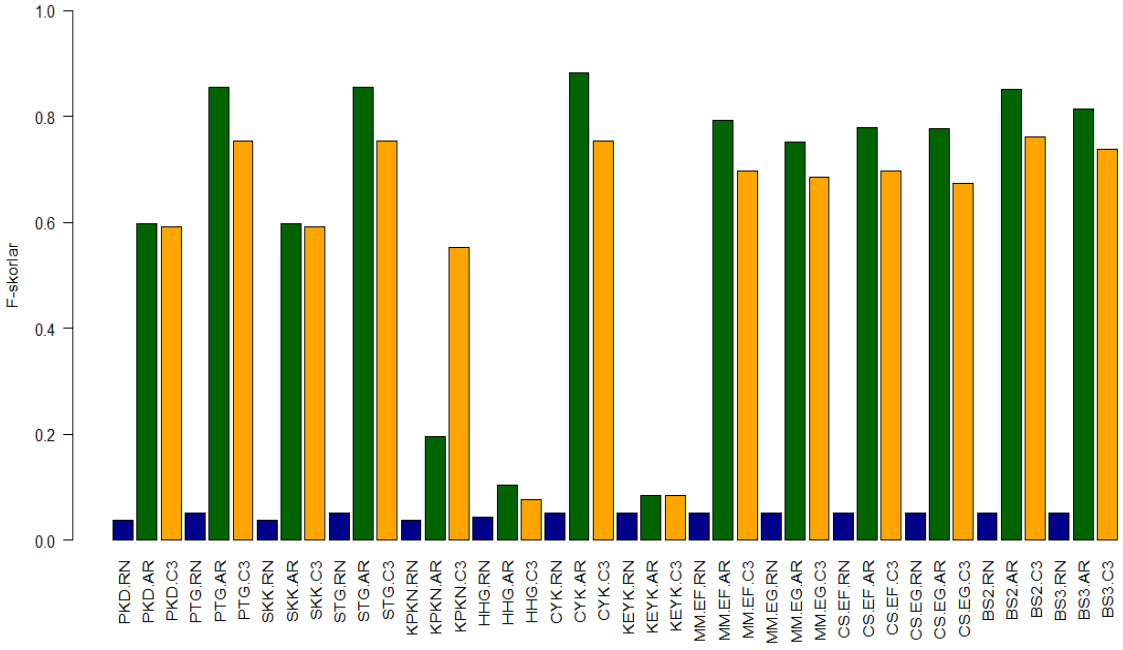
6.3. Kestirimcilerin CD Uygulanan 3. Yapay Veri Kümesine göre Karşılaştırılması

Bu tez çalışması için yapılan deneylerde kullanılan üçüncü sentetik veri kümesinde 100 gen ve 1000 örnek bulunmaktadır. Bu veri kümesinin doğrulama gen ağı da *E.Coli* bakterisinin bir alt ağına aittir. Bu veri kümesine CD ön işleminin uygulanması durumunda 14 farklı kestirimci (EBBK listeden çıkartılmış, Bknz. Bölüm 6.1 ve 6.2) ve üç farklı GAÇ algoritması için F-skor ve Hassasiyet skoruna göre değerlendirme yapılmıştır.

F-skor sonuçları Şekil 6.5 ve Çizelge 6.5'te verilmiştir. Ayrıca Çizelge'de kestirimcilerin R, seri ve paralel C++ kodlarının saniye cinsinden çalışma süreleri de verilmiştir.

RelNet GAÇ algoritması kullanılması durumunda tüm kestirimciler birbirine çok yakın ve diğer GAÇ algoritmalarının kullanılması durumunda elde edilen F-skor sonuçlarından ciddi oranda daha düşük performanslar sergilemişlerdir. Bu durum önceki iki sentetik veri kümesi için de aynen gözlenmiş; belirtildiği üzere RelNet diğer GAÇ algoritmalarından farklı olarak dolaylı bağlantıları eleyemediği için oldukça düşük ve sağlıksız değerlendirme sonuçları vermektedir. Bu nedenle 3. veri kümesi için de RelNet algoritması için kestirimcilerin değerlendirmesi yapılmamıştır.

Üçüncü sentetik veri kümesine CD ön işlemi uygulandığında, ARACNE algoritmasının kullanılma durumunda F-skor ölçütüne göre BS2, BS3, ÇYK, PTG ve STG kestirimcileri en başarılı yöntemler olarak gözlenmiştir. Her iki ayrıklaştırma tekniği (EF ve EG) ile kullanılan MM ve CS yöntemleri yukarıdaki en iyi 5 kestirimciye çok yakın F-skor sonuçları vermektedir. HHG ilginç bir şekilde örnek sayısı arttığında ciddi oranda düşük bir performans sergilemektedir. HHG, KPKN ve KEYK en kötü performansları sergileyen kestirimcilerdir.



Şekil 6. 5 Üçüncü sentetik veri kümesine CD ön işlemi uygulanması durumunda kestirimcilerin F-skor ölçütüne göre değerlendirilmesi

Çizelge 6. 5 CD kullanılması durumunda 3. veri kümesi için kestirimcilerin F-skorları

Kestirimci	RelNet F-skoru	ARACNE F-skoru	C3NET F-skoru	R kodu süresi (sn)	Seri C++ kodu süresi	Parallel C++ kodu süresi
PKD	0.0385	0.5981	0.5909	0.08	-	-
PTG	0.0507	0.8544	0.7543	0.08	-	-
SKK	0.0385	0.5981	0.5909	0.09	-	-
STG	0.0507	0.8544	0.7543	0.09	-	-
KPK-n	0.0375	0.1951	0.5517	0.39	-	-
HHG	0.0434	0.1040	0.0769	-	80950.73	45455.62
ÇYK	0.0507	0.8824	0.7543	53684.7	1093.50	632.75
KEYK	0.0507	0.0850	0.0847	82351.44	2490.64	1437.39
MMEF	0.0507	0.7925	0.6971	30.06	0.19	0.26
CSEF	0.0508	0.7789	0.6971	51.11	0.33	0.36
MMEG	0.0507	0.7512	0.6857	30.20	0.21	0.27
CSEG	0.0508	0.7767	0.6743	50.53	0.33	0.36
BS-2	0.0507	0.8502	0.7614	14459.10	9.89	9.20
BS-3	0.0507	0.8134	0.7386	14489.72	9.56	8.41

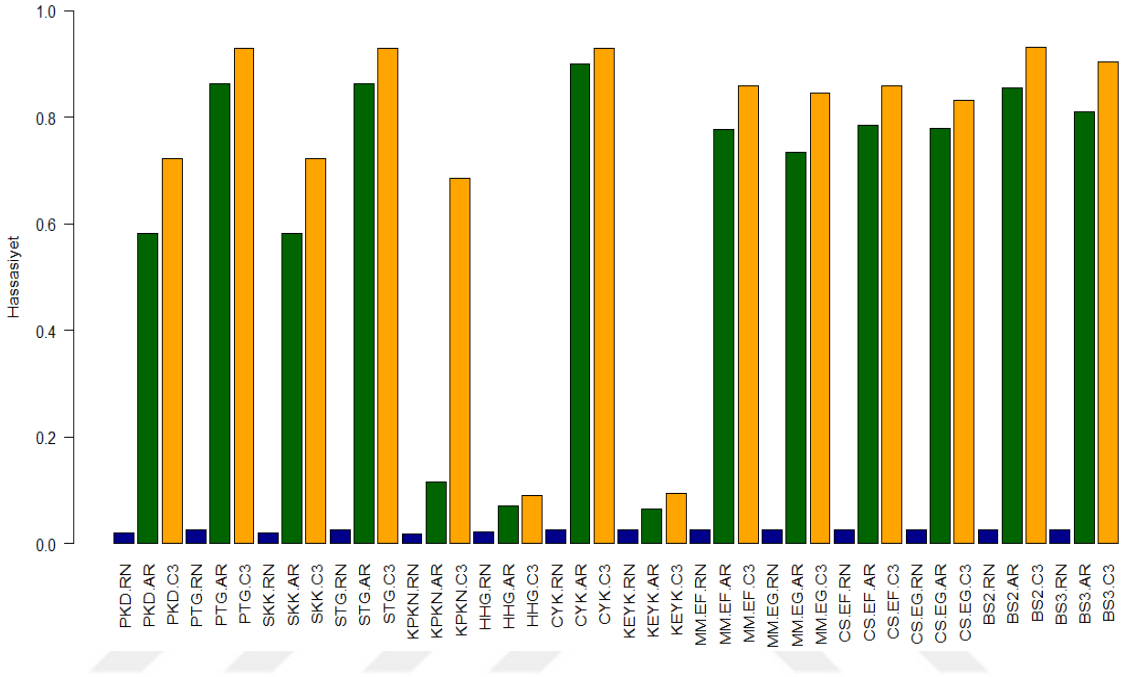
Aynı deneysel koşullarda C3NET algoritmasının kullanılması durumunda, ARACNE algoritması için en iyi performansı sergileyen 5 kestirimci (BS2, BS3, ÇYK, PTG ve STG) yine en başarılı yöntemler olarak gözlenmiştir. Buna ilaveten her iki ayrıklaştırma tekniği (EF ve EG) ile kullanılan MM ve CS kestirimcileri yine bu 5 kestirimciye yakın performans sergilemişlerdir. Aynı şekilde HHG, KPKN ve KEYK yöntemleri en başarısız sonuçları vermişlerdir.

En iyi performans sergileyen 5 kestirimci ve en kötü performansları veren 3 kestirimci ARACNE ve C3NET algoritmaları için aynıdır. CS ve MM kestirimcileri örnek sayısının artması ile daha yüksek performans sergilemişler ve F-skorları en iyi kestirimcilerin F-skor değerlerine yaklaşmıştır.

14 kestirimcinin üç GAÇ algoritması için elde edilen Hassasiyet skorları Şekil 6.6 ve Çizelge 6.6'da verilmiştir.

Şekil 6.6 ve Çizelge 6.6'dan açıkça görüldüğü üzere RelNet GAÇ algoritmasının kullanılması durumunda kestirimcilerin Hassasiyet skorları da F-skor değerlerinde de olduğu gibi birbirine çok yakın ve oldukça düşük değerlerdir. Dolayısıyla RelNet algoritması için kestirimcilerin sağlıklı değerlendirmesi yapılamadığından karşılaştırma sonucu verilmemiştir.

Hassasiyet ölçütüne göre ARACNE algoritmasının kullanılma durumunda BS2, BS3, ÇYK, PTG ve STG kestirimcileri en iyi performansları sergilemektedir. Her iki ayrıklaştırma tekniği (EF ve EG) ile kullanılan MM ve CS kestirimcileri bu 5 kestirimciye çok yakın performanslar sergilemektedir. Bu gözlemler F-skor sonuçlarından edinilen gözlemler ile aynıdır.



Şekil 6. 6 Üçüncü sentetik veri kümesine CD ön işlemi uygulanması durumunda kestirimcilerin Hassasiyet ölçütüne göre değerlendirilmesi

Çizelge 6. 6 CD kullanılması durumunda üçüncü veri kümesi için kestirimcilerin Hassasiyet skorları

Kestirimci	RelNet Hassasiyeti	ARACNE Hassasiyeti	C3NET Hassasiyeti
PKD	0.0197	0.5818	0.7222
PTG	0.0260	0.8628	0.9296
SKK	0.0197	0.5818	0.7222
STG	0.0260	0.8628	0.9296
KPK-n	0.0192	0.1162	0.6857
HHG	0.0222	0.0700	0.0897
ÇYK	0.0260	0.9000	0.9296
KEYK	0.0260	0.0644	0.0941
MMEF	0.0260	0.7778	0.8592
CSEF	0.0261	0.7843	0.8451
MMEG	0.0260	0.7340	0.8592
CSEG	0.0261	0.7789	0.8310
BS-2	0.0260	0.8544	0.9306
BS-3	0.0260	0.8095	0.9028

Hassasiyet ölçütüne göre C3NET algoritmasının kullanılması durumunda yine aynı 5 kestirimci (BS2, BS3, ÇYK, PTG ve STG) en iyi performansları sergilemektedir. Bu 5 kestirimcinin Hassasiyet skorları birbirine oldukça yakındır. MM ve CS kestirimcileri de bu yöntemleri çok yakından takip etmektedir. C3NET kullanılması durumunda da F-skor ve Hassasiyet ölçütlerine göre edinilen gözlemler birbiriyle aynıdır.

Hassasiyet ölçütüne göre genel değerlendirme yapıldığında, BS2, BS3, ÇYK, PTG ve STG kestirimcilerinin en iyi olduğu gözlenmiştir. Bu gözlem aynı veri kümesi için F-skor ölçütünden alınan gözlem sonuçları ile aynı ve uyumludur. Bu beş kestirimciye ek olarak ikinci sentetik veri kümesi için HHG de en başarılı kestirimcilerden biri olarak gözlenmiştir. Diğer kestirimciler örnek sayısının artması ile daha yüksek performanslar sergilemişler ancak HHG bu durumun dışında kalmıştır. HHG, artan örnek sayısı ile ciddi oranda bir performans düşüşü sergilemiştir. Dolayısıyla HHG yönteminin büyük veri kümeleri ile kullanımının uygun olmadığını söyleyebiliriz. Bunlara ilaveten, örnek sayısının artması ile CS ve MM yöntemleri en iyi kestirimcilere yakın performanslar sergilemektedir. Küçük veri kümeleri için de CS ve MM en ümit verici yöntemleri takip etmektedir ancak artan örnek sayısı ile performansları en iyi yöntemlerin performanslarına daha çok yaklaşmaktadır. İlk sentetik veri kümesinde kullanılan örnek sayısı diğer veri kümelerine nazaran daha az olduğu için bu veri kümesinden elde edilen F-skor sonuçları diğer algoritmaların sonuçlarından düşüktür. İlk üç sentetik veri kümelerinin F-skor sonuçlarına bakıldığında artan örnek sayısı ile kestirimci performanslarının da arttığı açıkça gözlenmiştir. Daha önce belirtildiği üzere, örnek sayısının artmasının kestirimcilerin çıkarım performansını artırdığı [45]'te zaten gösterilmiş, çalışma sonucunda en az 64 örnek kullanılması gerektiği tavsiye edilmiştir. Bu nedenle bu tez çalışmasında Bölüm 6.1'de 50 örnekli ilk veri kümesi için verilen sonuçlar bu durum göz önüne alınarak değerlendirilmelidir.

Sonuç olarak Bölüm 6.1, 6.2 ve 6.3'te verilen sonuçlardan yola çıkılarak en iyi performans sergileyen kestirimcilerin her iki metriğe göre, üç veri kümesi için de ortak olduğu gözlenmiştir. Bu kestirimciler BS2, ÇYK, PTG ve STG'dir. Her iki ayrıklaştırma tekniği ile kullanılan CS ve MM kestirimcileri de bu en ümit verici kestirimcileri takip etmektedir. Ayrıca tüm sentetik veri kümeleri için sırayla Çizelge 6.1, 6.3 ve 6.5'te kestirimcilerin R kodu, seri ve paralel C++ kodlarının saniye cinsinden çalışma süreleri,

RelNet, ARACNE ve C3NET GAÇ algoritmalarına göre çıkarım performansları verilmiştir. Çalışma sürelerini de F-skor ölçütü ile beraber değerlendirdiğimizde BS2, PTG ve STG (tabi CS ve MM'nin de) kestirimcilerinin en ümit verici yöntemler olduğu gözlenmiştir. C++ kodları R gerçeklemelerinden çok daha hızlı çalışmasına rağmen, ÇYK yönteminin C++ kodunun çalışma süresi bile diğer yöntemlerin çalışma sürelerinden ciddi oranda çok daha fazladır. Dolayısıyla ÇYK'nın pratik uygulamalarda ve özellikle büyük veri kümeleri ile kullanımı tavsiye edilmemektedir. Bunların dışında Çizelge'lerde PKD, SKK, PTG, STG ve KPK-n yöntemlerinin sadece R kodu için çalışma süreleri mevcuttur çünkü bu yöntemler C++'da gerçeklenmemiştir. Bu yöntemler R'ın kendi fonksiyonlarından faydalanılarak gerçeklendikleri için çalışma süreleri Çizelge'lerde görüldüğü üzere R'da kullanıcı tanımlı fonksiyonlar ile gerçeklenen diğer kestirimcilerin çalışma sürelerinden zaten oldukça düşüktür ve C++'da gerçeklenmelerine ihtiyaç yoktur.

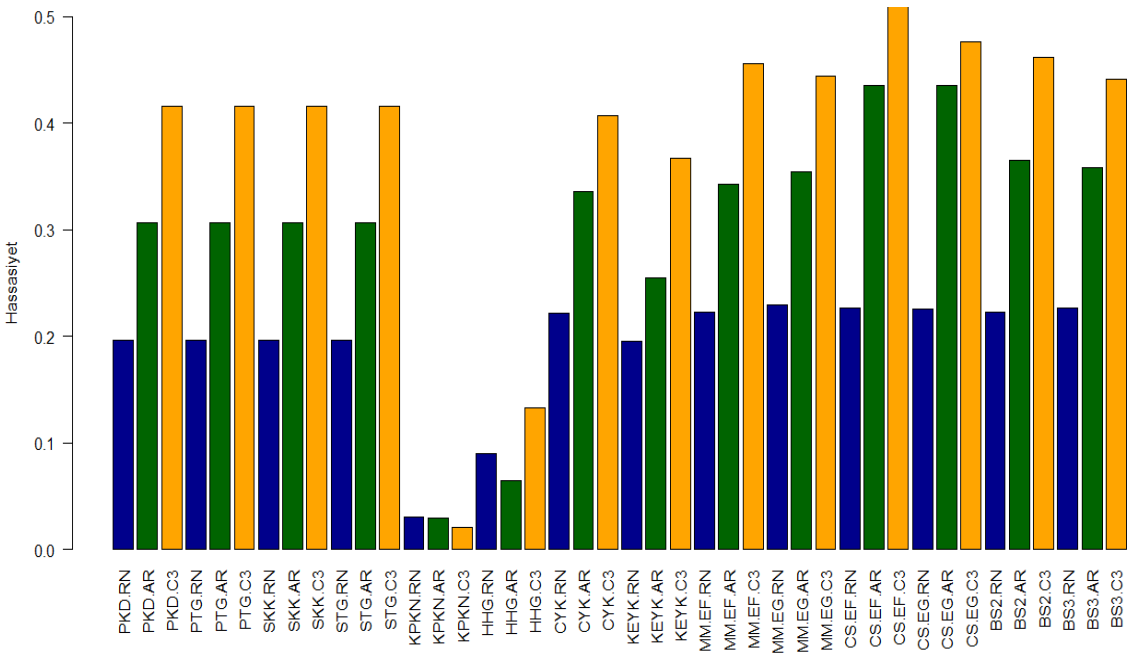
Yine paralel C++ kodları gerçeklenen kestirimciler, 4 çekirdekli bir bilgisayarın 3 çekirdeği kullanılarak test edilmiş; HHG, ÇYK ve KEYK için çalışma süresinin yarıya yakın azaldığı, 1000 örnek içeren üçüncü sentetik veri kümesi ile MM, CS ve BS kestirimcileri için süre açısından kara geçilmeye yeni yeni başlandığı gözlenmiştir.

Son olarak bu tez çalışmasında gerçek bir biyolojik veri kümesi kullanılarak da kestirimciler değerlendirilmiştir. Bununla ilgili sonuçlar Bölüm 6.4'te verilmiştir.

6.4. Kestirimcileri CD Uygulanan Gerçek Biyolojik Veri Kümesiyle Karşılaştırma

Bu bölümde kestirimcilerin üç farklı GAÇ algoritmasına göre çıkarım performansları gerçek bir biyolojik veri kümesi kullanılarak değerlendirilmiştir. *E.coli* bakterisine ait bu veri kümesi Context Likelihood or Relatedness (CLR) GAÇ algoritmasının [25] yazılım paketinden alınmıştır. Gen ifade veri kümesinin doğrulama gen ağındaki bağlantılar çoğunlukla RegulonDB [104] isimli veritabanından alınmıştır. Bu veritabanı, *E.Coli* bakterisine ait genler arasındaki düzenleme (regulation) bilgilerini içermektedir. Kullanılan veri kümesinde 524 mikroçip ve 1146 gen bulunmaktadır. Genler arasında 152 kopyalama faktörü (transcription faktör, TF) bulunmaktadır. Kopyalama faktörü olan genler ve hedef (target) genler arasında 3091 bağlantı bulunmaktadır. Kestirimciler değerlendirilirken Bağıntı (6.2)'de tanımı verilen Hassasiyet (precision) skoru esas alınmıştır. Veri kümesine CD ön işleminin uygulanması durumunda 14

kestirimci ve üç GAÇ algoritmasından alınan F-skor sonuçları da Hassasiyet skorları ile birlikte Çizelge 6.7'de verilmiştir. Yapay veya sentetik veri kümelerinin kullanıldığı çalışmalarda GAÇ uygulamalarını değerlendirmek amacıyla sıklıkla F-skor ölçütü kullanılmasına rağmen, gerçek veri kümeleri eksiksiz olmadığı için ve genler arasında sadece kısıtlı sayıdaki bağlantılar bilinebildiği için F-skor ölçütü yerine Hassasiyet ölçütü kullanılarak değerlendirme yapılmaktadır. Gerçek veri kümelerinin doğrulama gen ağına dair eldeki bilgiler tamamlanmamış olduğu için çıkartılan gerçek pozitif (True Positive, TP) bağlantı adedi düşük olmakta, bu da F-skor değerlerinin oldukça düşük olmasına sebep olmaktadır. Dolayısıyla, F-skor gerçek veri kümeleri için sağlıklı bir değerlendirme sağlayamayabilir. Yine de yukarıda belirtildiği üzere Çizelge 6.7'de kestirimcilerin F-skor sonuçları da Hassasiyet skorları ile beraber sunulmuştur. Ayrıca Bölüm 6.1, 6.2 ve 6.3'teki sonuçlarda gösterildiği üzere sentetik veri kümeleri için Hassasiyet ölçütüne göre edinilen gözlemler, F-skor ölçütüne göre edinilen gözlemlerle doğrudan ilişkili ve uyumludur. Dolayısıyla değerlendirme için Hassasiyet ölçütünün kullanımı yerinde bir tercihtir. Kestirimcilerin gerçek veri kümesi için Hassasiyet skorları Şekil 6.7'de verilmiştir. Daha önce belirtildiği üzere mavi çizgiler RelNet'e (RN), yeşiller ARACNE'ye (AR), turuncular da C3NET'e (C3) karşılık gelmektedir.



Şekil 6. 7 Gerçek biyolojik veri kümesine CD ön işlemi uygulanması durumunda kestirimcilerin Hassasiyet ölçütüne göre değerlendirilmesi

Çizelge 6. 7 CD kullanılması durumunda gerçek biyolojik veri kümesi için kestirimcilerin F-skorları ve Hassasiyet skorları

Kestirimci	RelNet F-skoru	RelNet Hassasiyeti	ARACNE F-skoru	ARACNE Hassasiyeti	C3NET F-skoru	C3NET Hassasiyeti
PKD	0.0906	0.1968	0.0307	0.3068	0.0263	0.4158
PTG	0.0906	0.1968	0.0307	0.3068	0.0263	0.4158
SKK	0.0906	0.1968	0.0307	0.3068	0.0263	0.4158
STG	0.0906	0.1968	0.0307	0.3068	0.0263	0.4158
KPK-n	0.0176	0.0310	0.0150	0.0296	0.0024	0.0213
HHG	0.0508	0.0905	0.0229	0.0652	0.0118	0.1329
ÇYK	0.0963	0.2220	0.0285	0.3358	0.0208	0.4074
KEYK	0.0901	0.1950	0.0293	0.2553	0.0208	0.3667
MMEF	0.0903	0.2229	0.0231	0.3426	0.0196	0.4559
CSEF	0.0906	0.2270	0.0190	0.4348	0.0134	0.5122
MMEG	0.0937	0.2292	0.0250	0.3540	0.0202	0.4444
CSEG	0.0909	0.2254	0.0190	0.4348	0.0128	0.4762
BS-2	0.0919	0.2229	0.0238	0.3654	0.0190	0.4615
BS-3	0.0930	0.2263	0.0238	0.3585	0.0190	0.4412

Gerçek biyolojik veri kümesi için RelNet GAÇ algoritmasının kullanılması durumunda hassasiyet skorlarının ikinci ve üçüncü sentetik veri kümesinden elde edilen Hassasiyet skorlarına nazaran biraz yükseldiği gözlenmiştir, ancak bu değerler hala diğer GAÇ algoritmalarının Hassasiyet skorlarını yakalayamamıştır. HHG ve KPKN dışındaki tüm kestirimcilerin skorları birbirine oldukça yakındır. RelNet algoritması diğer GAÇ algoritmalarından farklı olarak dolaylı bağlantıları etkin bir şekilde eleyemediği için hatalı pozitif (false positive, FP) bağlantıların oranı oldukça yüksektir, bu da performansı diğer algoritmalara nazaran daha çok düşürmektedir.

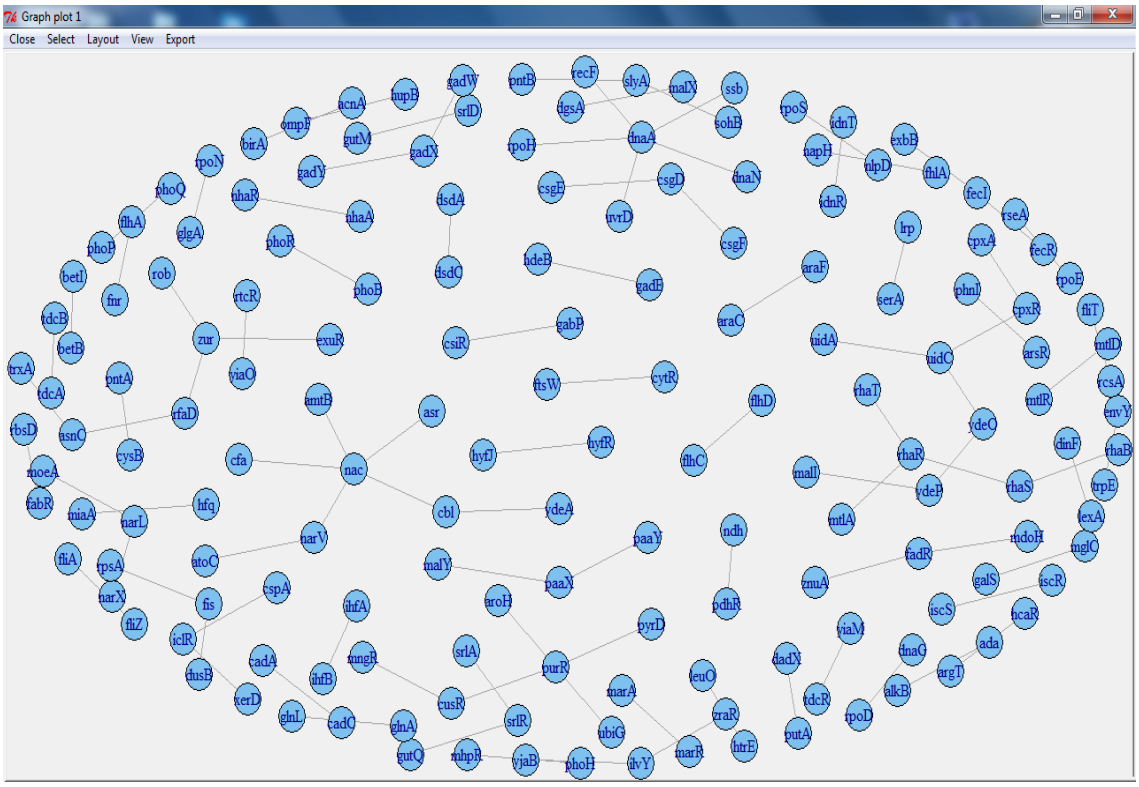
ARACNE algoritmasının kullanılması durumunda CSEF ve CSEG kestirimcilerinin az bir farkla diğer kestirimcilerin önüne geçtiği gözlenmiştir. BS2, BS3, MMEF, MMEG ve ÇYK yakınen CS'yi takip etmektedir. PKD, SKK, PTG ve STG de diğer ümit verici kestirimcilerdir ve ismi geçen en iyi kestirimcilere yakın performanslar sergilemektedirler. Aslında HHG, KPKN ve KEYK haricinde tüm kestirimciler ARACNE algoritması ile başarılı performanslar sergilemişlerdir.

C3NET algoritmasının kullanılması durumunda yine CSEF ve CSEG öne çıkan kestirimciler olarak gözlenmiştir. BS2, BS3, MMEF, MMEG yöntemleri CS'yi yakından takip etmektedir. Dolayısıyla genel bakışta BS, CS ve MM kestirimcilerinin en göze

çarpan yöntemler olduğu söylenebilir. Ayrıca PKD, SKK, PTG, STG ve ÇYK da bu kestirimcilere benzer performanslar sergilemekte ve yakından takip etmektedirler.

Bu veri kümesine CD uygulanması durumunda SKK kestirimcisi ve C3NET algoritması ile elde edilen gen ağı örnek olması açısından Şekil 6.8'de verilmiştir.

Kestirimcilerin tüm veri kümeleri için CD ön işlemi kullanılması durumundaki ortak değerlendirilmesi bir sonraki alt bölümde verilmiştir.



Şekil 6. 8 CD uygulanan gerçek biyolojik veri kümesi için SKK kestirimcisi ve C3NET algoritması ile elde edilen gen ağı

6.5. Kestirimcilerin CD Kullanılan Tüm Veri Kümeleri için Ortak Değerlendirilmesi

Kestirimcileri, hem sentetik hem de gerçek veri kümelerine göre ve her iki performans değerlendirme metriğine (F-skor ve hassasiyet) göre bir arada değerlendirdiğimizde PTG ve STG kestirimcilerinin ortak olarak ön plana çıktığını söyleyebiliriz. BS2, BS3 ve ÇYK da 50 örnek içeren ilk sentetik veri kümesi haricinde, ortak olarak tüm veri kümeleri için öne çıkan diğer yöntemlerdir. Gerçek biyolojik veri kümesi için Hassasiyet ölçütüne göre CSEF ve CSEG yöntemlerinin yukarıda ismi geçen 5 yöntemle beraber öne çıkması okuyucuları yanıltmamalıdır. Çünkü sentetik veri kümelerinin F-skor

sonuçlarında görüldüğü üzere CS yöntemi diğer kestirimcilerin peşisıra gelmiştir ancak onlar kadar yüksek performans sergileyememiştir. Bunun dışında BS, ÇYK, PTG ve STG yöntemleri arasında karar verirken kestirimcilerin Çizelge 6.1, 6.3 ve 6.5'te verilen çalışma süreleri de göz önüne alınmalıdır. ÇYK'nın çalışma süresi diğerlerine nazaran oldukça yüksektir bu nedenle de kullanılması pek tavsiye edilmemektedir. PTG ve STG kestirimcilerinin çalışma süreleri en düşük olduğu için GAÇ uygulamalarında kullanımları idealdir. Ayrıca BS kestirimcisinin de çalışma süresi tercih edilebilir boyuttadır. Dolayısıyla hem her iki performans değerlendirme ölçütüne hem de çalışma süresine bakıldığında en tercih edilebilir kestirimciler BS, PTG ve STG şeklindedir.

En kötü performans sonuçlarını veren EKKB kestirimcisi ilk iki sentetik veri kümesi için değerlendirilmiş, optimize edilecek parametrelerinin seçimi güç olduğu için diğer veri kümeleri için değerlendirmeden çıkartılmıştır. EKKB'yi takiben KPKN kestirimcisi en kötü performansı vermiştir. KPKN yöntemi kendine has bir yaklaşımla dolaylı bağlantıları elemekte, bu esnada gerçek pozitif (true positive, TP) bağlantıları da yanlışlıkla ağdan silmektedir. Bunların dışında örnek sayısının artması ile diğer yöntemlerin aksine HHG yönteminin başarımının düştüğü gözlenmiştir.

Bunların dışında, veri kümesindeki örnek sayısının artışı ile genel olarak kestirimcilerin tümünde (HHG hariç) performans artımı gözlenmiştir.

Ayrıca, kestirimcilerin çalışma sürelerini değerlendirmek için paralel C++ kodu gerçeklenen yöntemler, 4 çekirdekli bir bilgisayarın 3 çekirdeği kullanılarak test edilmiş; tüm veri kümelerinde HHG, ÇYK ve KEYK için çalışma süresinin yarıya yakın azaldığı, küçük veri kümelerinde ise MM, CS ve BS kestirimcileri için süre açısından kara henüz geçilemediği, örnek sayısı arttıkça kara geçilmeye başlandığı gözlenmiştir. Çekirdek sayısının daha fazla olduğu bilgisayarlarda paralel kodlama mantığı ile süre açısından daha fazla kara geçilmesi beklenmektedir.

Yukarıdaki çıkarımlara ilaveten, F-skor metriğine göre GAÇ algoritmaları arasında ARACNE'nin C3NET'ten çok az bir miktar daha iyi performans sergilediği; ARACNE ve C3NET'in birlikte RelNet'ten ciddi oranda daha iyi performanslar sergilediği gözlenmiştir (Şekil 6.1, 6.3 ve 6.5). Hassasiyet ölçütüne bakıldığında ise C3NET'in ARACNE'den daha iyi olduğu, her ikisinin de yine RelNet'ten ciddi bir oranda daha iyi

olduğu gözlenmiştir. Bölüm 4.2'de ARACNE algoritmasında anlatıldığı üzere, τ: tolerans parametresi ARACNE'nin performansını etkileyen önemli bir etmendir. Bu çalışmada τ parametresi 0 olarak alınmıştır.

Son olarak, Şekil ve Çizelge'lerden görüldüğü üzere GAÇ uygulamalarındaki çıkarım performansı üzerinde GAÇ algoritmalarından ziyade etkileşim skoru kestirimcilerinin daha fazla etkisi bulunmaktadır. Zaten unutulmamalıdır ki, bu tez çalışmasının temel amacı kestirimcileri karşılaştırarak değerlendirmektir, GAÇ algoritmalarını karşılaştırmak değildir. Çalışmada farklı GAÇ algoritmalarının kullanılmasının sebebi kestirimcileri farklı durumlara göre adilane bir şekilde değerlendirebilmektir.

CD ön işleminin kullanılmadığı durumlar için kestirimcilerin ortak değerlendirilme sonucu bir sonraki bölümde verilmiştir.

6.6. Kestirimcilerin CD Kullanılmayan Duruma göre Tüm Veri Kümeleri için Ortak Değerlendirilmesi ve Ayrıklaştırma Tekniklerinin Etkilerinin Değerlendirilmesi

Bu bölümde, CD ön işleminin sentetik ve gerçek biyolojik veri kümelerine uygulanmaması durumunda kestirimcilerin çıkarım performansları değerlendirilmiştir. Değerlendirme sentetik veri kümeleri için hem F-skor hem de Hassasiyet skoruna göre; gerçek biyolojik veri kümesi için ise sadece Hassasiyet skoruna göre yapılmıştır. Şekil 6.9 ve 6.10'da 50 örnekli ilk sentetik veri kümesi için kestirimcilerin sırayla F-skorları ve Hassasiyet skorları CD ön işleminin kullanılma ve kullanılmama durumu için birlikte verilmiştir. İkinci sentetik veri kümesine CD uygulama ve uygulamama durumları için kestirimcilerin F-skorları Şekil 6.11'de, Hassasiyet skorları ise 6.12'de görülmektedir. Üçüncü sentetik veri kümesine CD uygulama ve uygulamama durumlarını birlikte gösteren grafikler ise F-skor için Şekil 6.13'te, Hassasiyet skorları için 6.14'te verilmiştir. Gerçek biyolojik veri kümesine CD ön işlemi uygulama ve uygulamama durumları bir arada Hassasiyet ölçütüne göre Şekil 6.15'te verilmiştir. Şekil 6.9 ile 6.15 arasındaki gösterimlerde açık mavi çizgiler ve "RN" kısaltması kestirimcilerin RelNet GAÇ algoritması ile ve CD ön işlemi kullanılmadan değerlendirildiğini; koyu mavi çizgiler ve "RN.C" kısaltması kestirimcilerin RelNet algoritması ve CD ön işlemi kullanılarak değerlendirildiğini göstermektedir. Açık yeşil çizgiler ve "AR" kısaltması kestirimcilerin CD ön işlemi kullanılmadan ve ARACNE GAÇ algoritması ile değerlendirildiğini; koyu

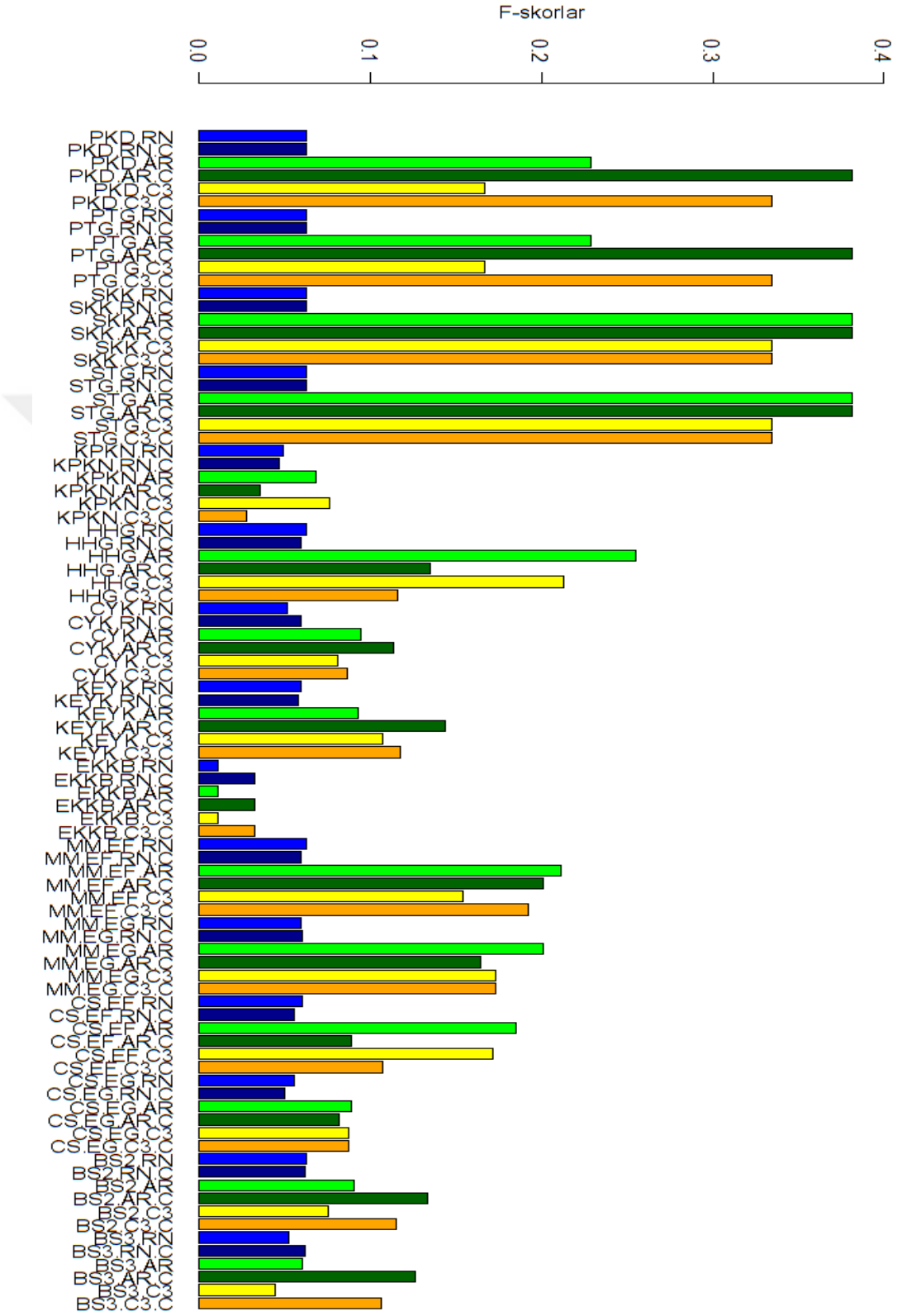
yeşil çizgiler ve “AR.C” kısaltması kestirimcilerin CD ön işlemi kullanılarak ve ARACNE ile değerlendirildiğini belirtmektedir. Sarı çizgiler ve “C3” kısaltması kestirimcilerin CD ön işlemi kullanılmadan ve C3NET GAÇ algoritmasına göre değerlendirildiğini; turuncu çizgiler ve “C3.C” kısaltması ise kestirimcilerin CD ön işlemi kullanılarak C3NET ile değerlendirildiğini ifade etmektedir. Bu şekilleri öncelikle ayrı ayrı, sonra da birlikte ele alarak CD ön işleminin kestirimciler üzerindeki etkisini inceleyelim.

Şekil 6.9 ve 6.10’da ilk sentetik veri kümesinin F-skor ve Hassasiyet ölçütü sonuçlarının birbiri ile uyumlu ve tutarlı olduğu görülmektedir. CD ön işleminin veri kümesine uygulanması durumunda tüm GAÇ algoritmaları için PKD, PTG, SKK ve STG kestirimcilerinin her iki performans metriğine göre de en başarılı kestirimciler oldukları gözlenmiştir. En başarılı yöntemlerden olan PKD ve PTG için CD ön işleminin kullanılmasının bu kestirimcilerin performanslarını ciddi bir oranda artırdığı gözlenmiştir. Bunların haricinde BS2, BS3, ÇYK, KEYK kestirimcileri için de CD ile performansın arttığı gözlenmektedir. Dolayısıyla CD ön işleminin kestirimcilerin performansını genel olarak artıran bir etmen olduğunu söyleyebiliriz. Ancak, SKK ve STG yöntemleri halihazırda örneklerin orijinal değerleri yerine, CD işlemindeki gibi sıralama değerlerini kullandıkları için bu iki yöntem üzerinde CD’nin bir etkisi olmamış; CD işleminin kullanıldığı ve kullanılmadığı durumların başarım sonuçları aynı kalmıştır.

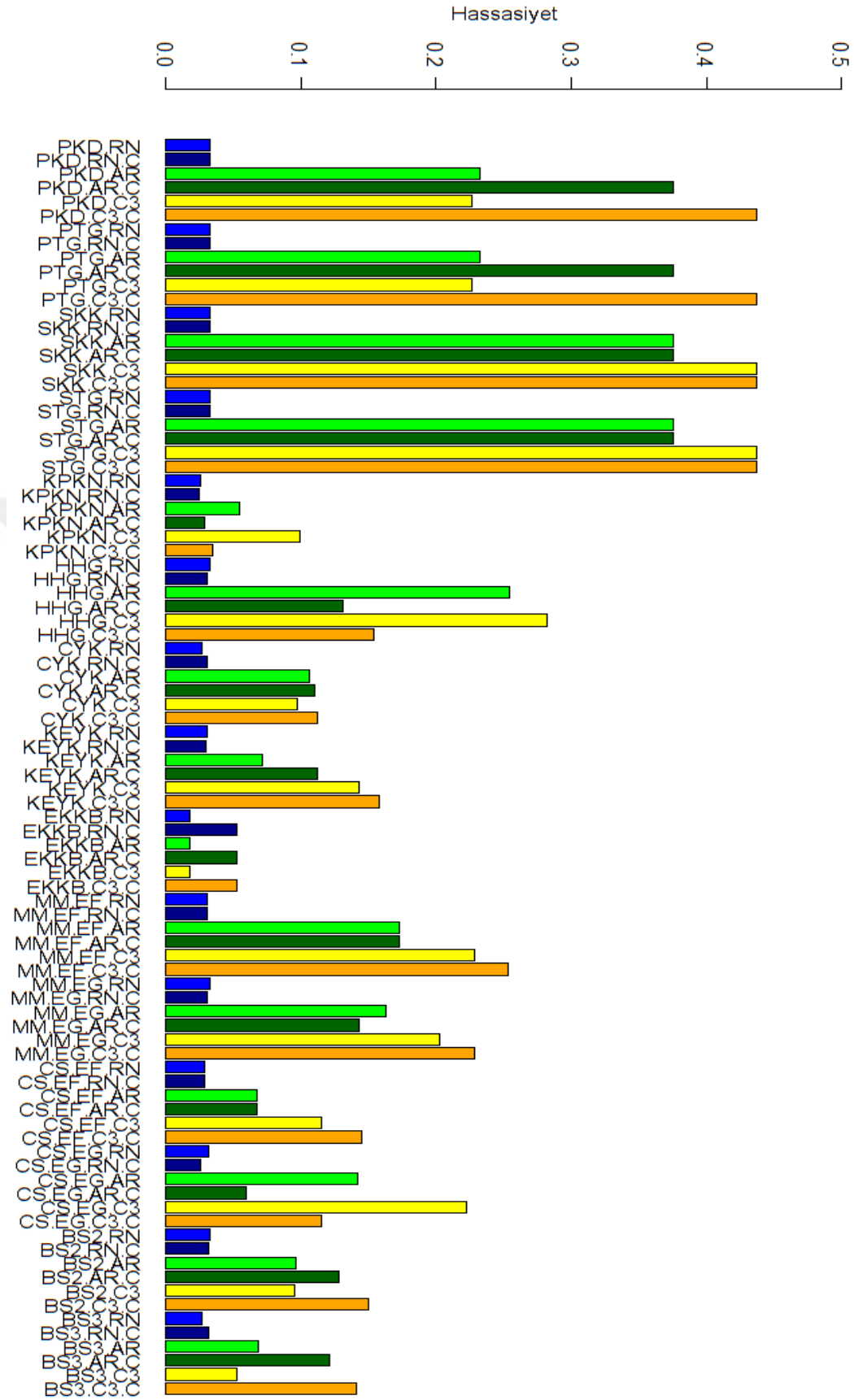
Şekil 6.11 ve 6.12’de ikinci sentetik veri kümesi için de F-skor ve Hassasiyet ölçütü sonuçlarının birbiri ile uyumlu ve tutarlı olduğu görülmektedir. Bu veri kümesi için genel olarak en iyi performans sergileyen yöntemler arasında bulunan BS3, ÇYK ve PTG kestirimcilerinin her iki performans metriğine göre de CD ön işleminin kullanılması durumunda performanslarını ciddi bir oranda artırdığı gözlenmiştir. Bunların haricindeki bazı kestirimcilerin de (örneğin PKD ve HHG) CD işlemi ile performanslarını arttığı gözlenmektedir. Dolayısıyla bu veri kümesi için de CD ön işleminin kestirimcilerin performansını artıran bir etmen olduğunu söyleyebiliriz.

Şekil 6.13 ve 6.14’te üçüncü veri kümesi için verilen F-skor ve Hassasiyet skorlarına bakıldığında yine CD işleminin kestirimcilerin çıkarım performansını genel olarak artırdığı söylenebilir. Bu veri kümesi için kestirimcilerin bir kısmında performans artışı önemli bir oranda iken; bir kısmındaki artış daha önemsizdir. En iyi performans

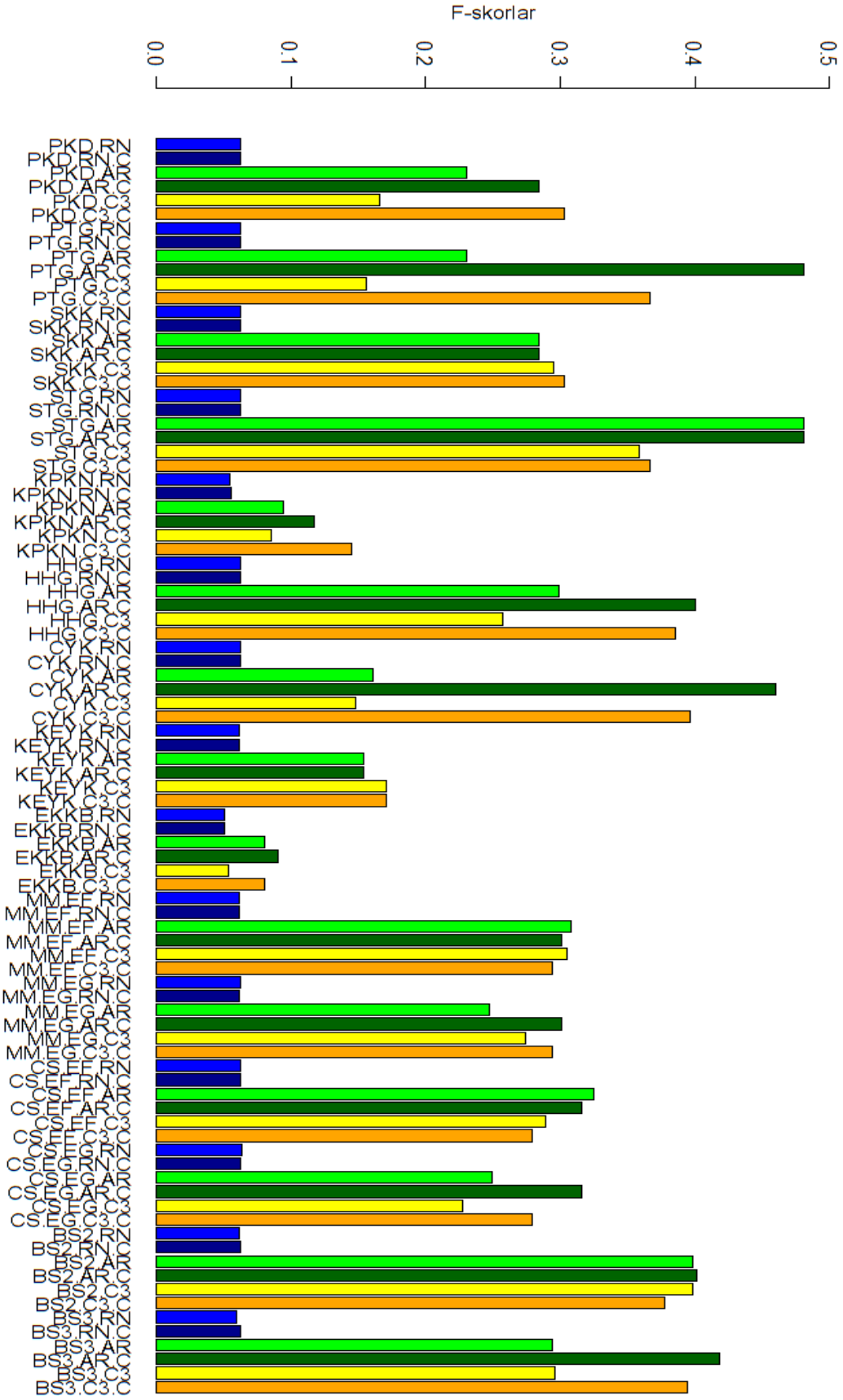
sergileyen yöntemlerden olan BS3, ÇYK ve PTG için CD ile performans artışı önemli orandadır. Bu gözlem sonucu, 2. veri kümesi için elde edilen sonuçlarla da uyumludur.



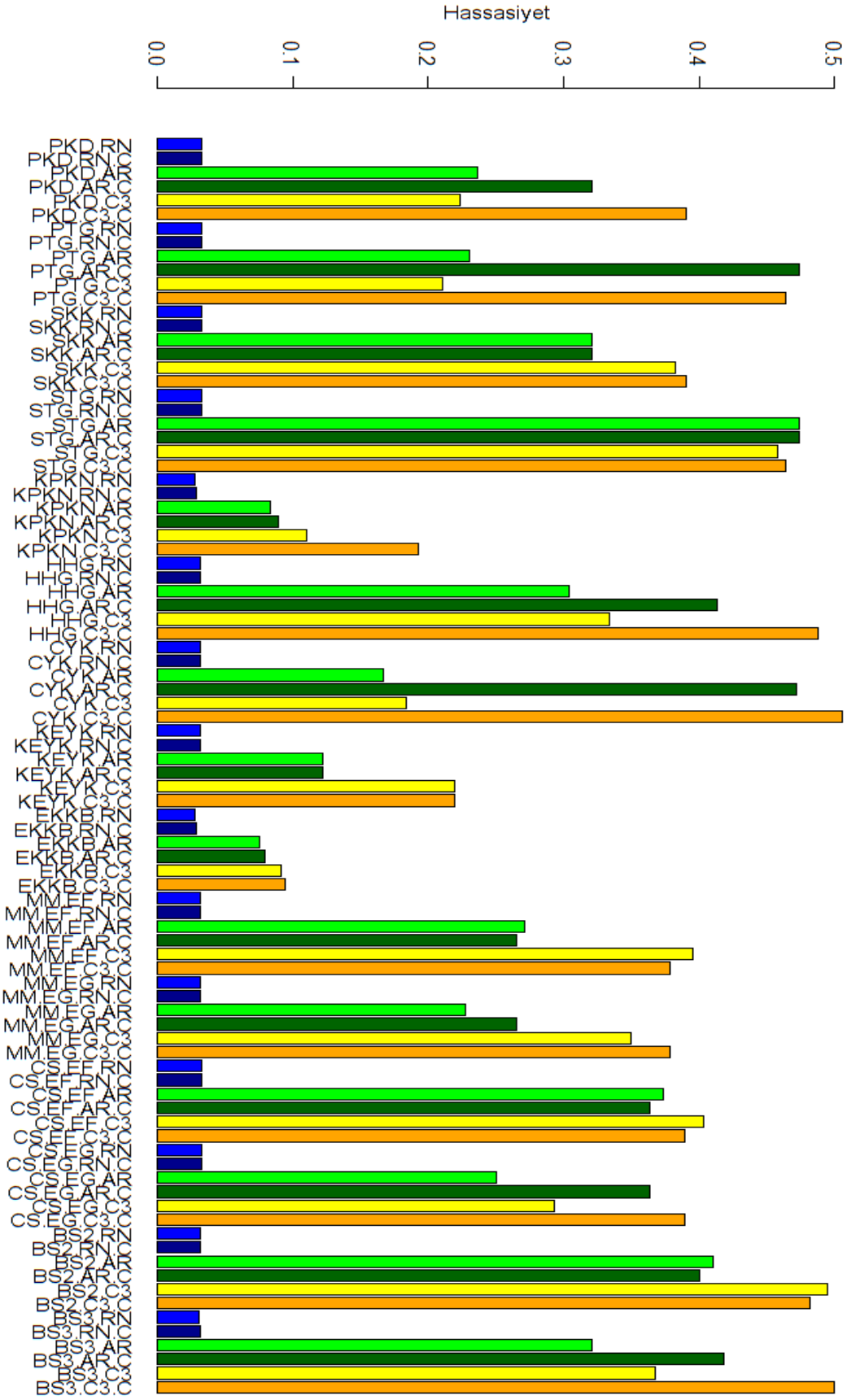
Şekil 6. 9 Birinci sentetik veri kümesi için CD kullanma/kullanmama durumları için gözlenen F-skorlar



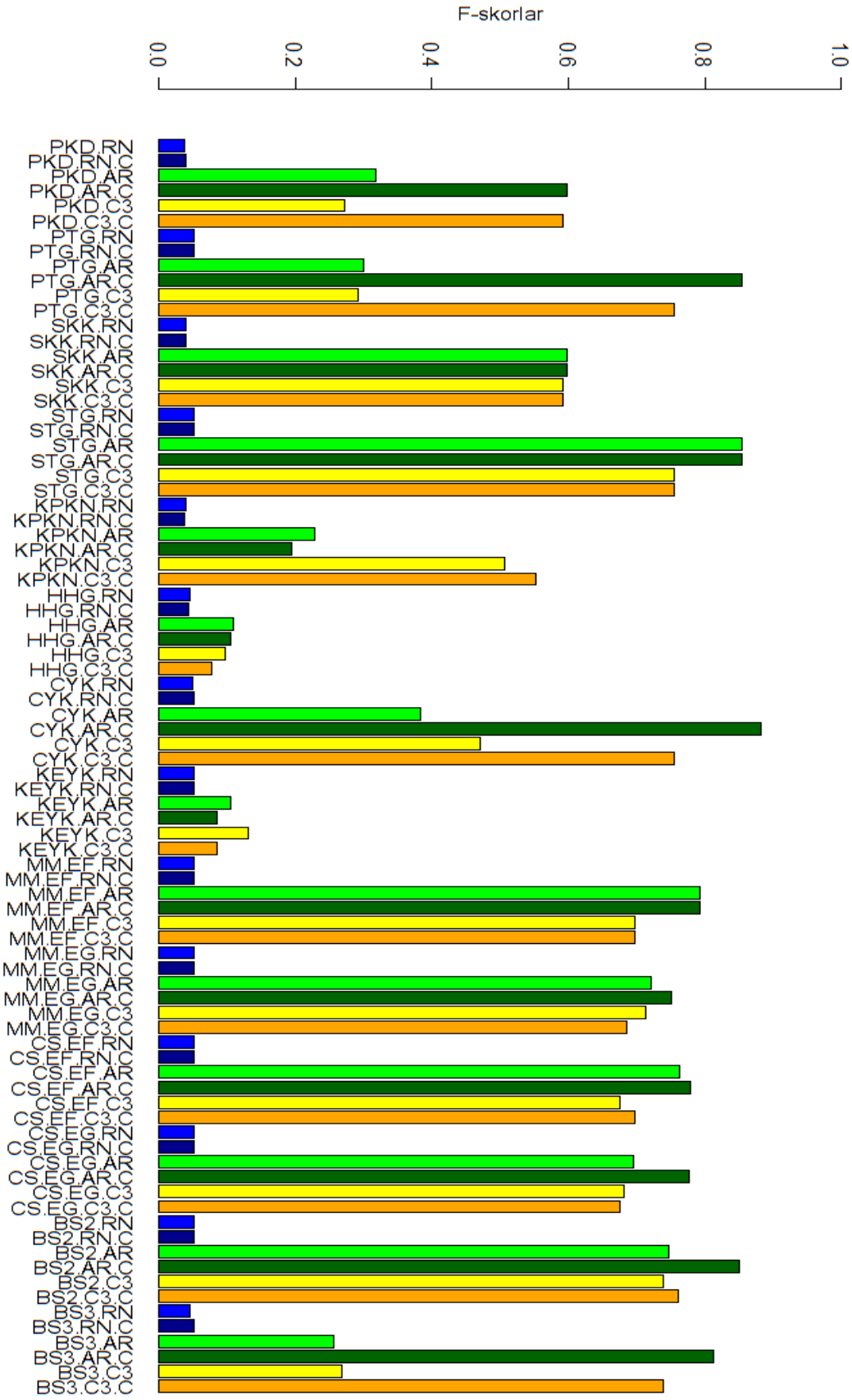
Şekil 6. 10 Birinci sentetik veri kümesi için CD kullanma/kullanmama durumları için gözlenen Hassasiyet skorları



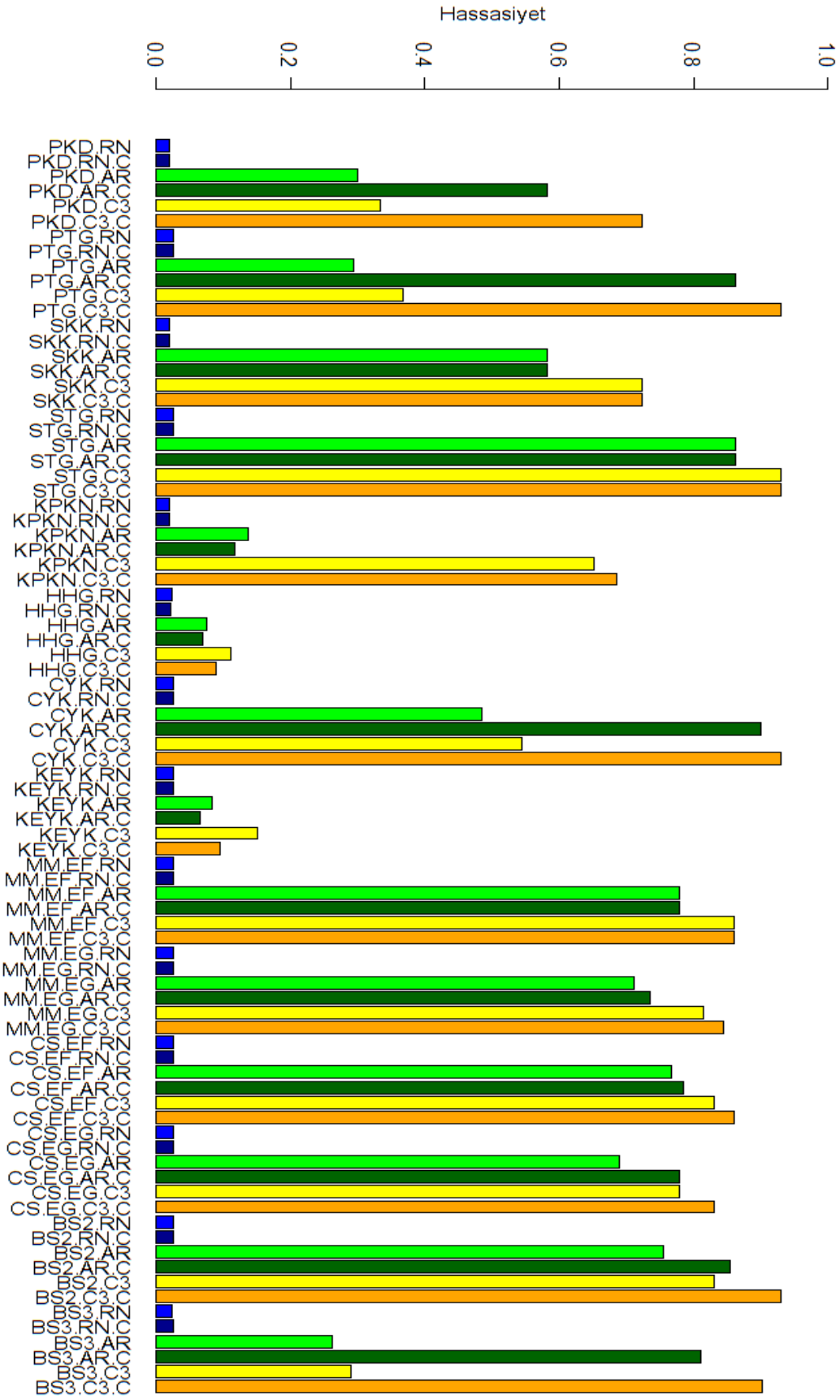
Şekil 6. 11 İkinci sentetik veri kümesi için CD kullanma/kullanmama durumları için gözlenen F-skorlar



Şekil 6. 12 İkinci sentetik veri kümesi için CD kullanma/kullanmama durumları için gözlenen Hassasiyet skorları



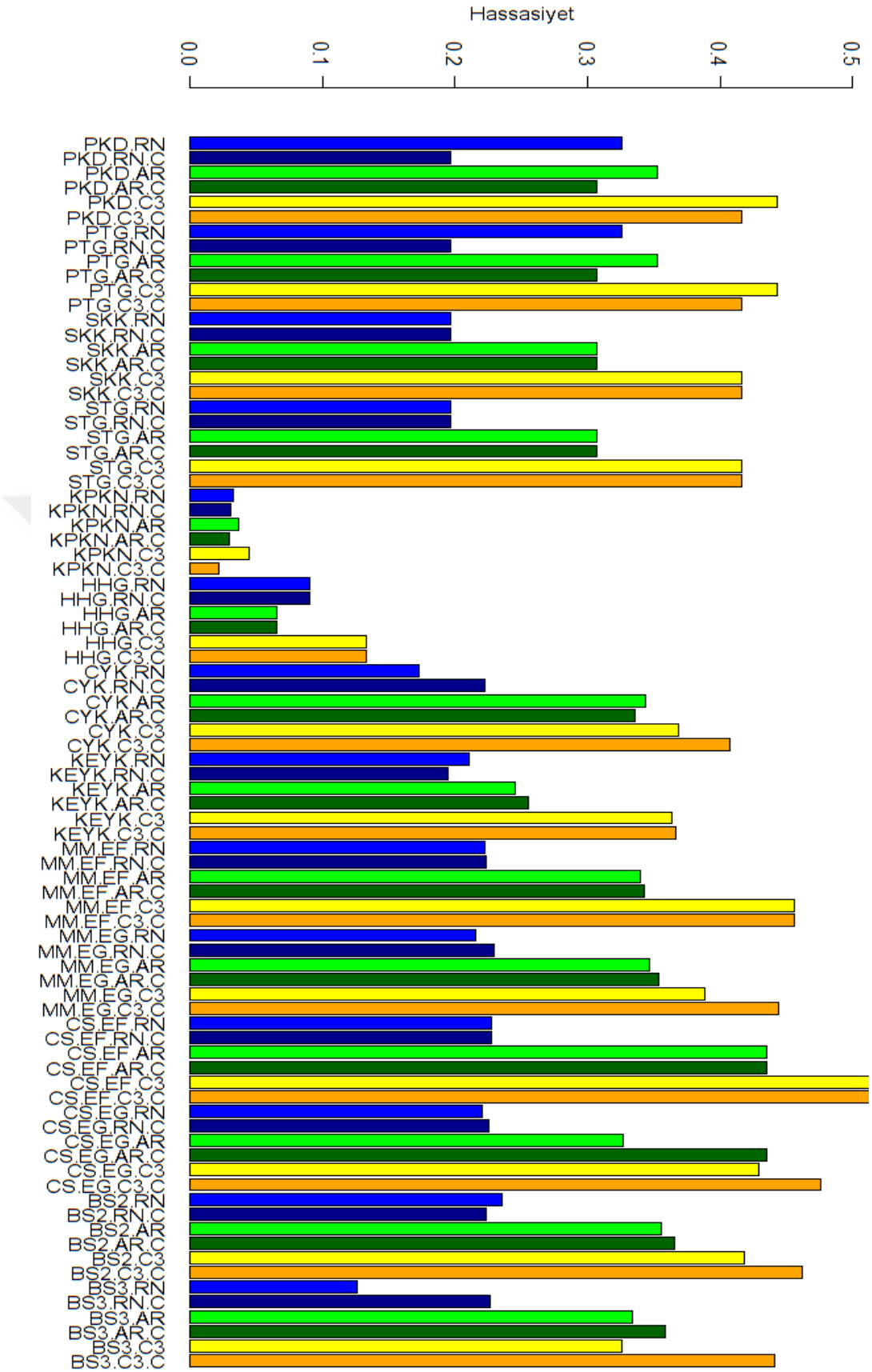
Şekil 6. 13 Üçüncü sentetik veri kümesi için CD kullanma/kullanmama durumları için gözlenen F-skorlar



Şekil 6. 14 Üçüncü sentetik veri kümesi için CD kullanma/kullanmama durumları için gözlenen Hassasiyet skorlar

Gerçek biyolojik veri kümesi için CD ön işleminin kullanılma ve kullanılmama durumlarında kestirimciler için elde edilen Hassasiyet skorları Şekil 6.15'te verilmiştir. Değerlendirme için F-skor metriği yerine Hassasiyet skorunun kullanılma nedeni bir önceki bölümde verilmiştir. Sentetik veri kümeleri için F-skor ve Hassasiyet ölçütlerinin sonuçlarının birbirleri ile uyumlu ve benzer olduğu da halihazırda gösterilmiştir. Dolayısıyla değerlendirme için Hassasiyet ölçütünün kullanılması makul ve uygun görülmektedir. Gerçek biyolojik veri kümesi kullanıldığında kestirimcilerin bir kısmı için CD ön işlemi kullanımının performansı pek değiştirmediği gözlenmesine rağmen, genele bakıldığında bu ön işlemin kestirimcilerin performansını ağırlıklı olarak artırdığı gözlenmiştir. Ancak bu gerçek biyolojik veri kümesi için olan performans artışı, sentetik veri kümeleri için gözlenen performans artışından daha düşük ve daha az göze çarpmaktadır. Bununla beraber, birkaç kestirimci için CD ön işleminin performans düşmesine sebep olduğu da gözlenmiştir. Yine de, daha önce belirtildiği gibi CD ön işleminin genel olarak bu veri kümesi için de performansı artırdığı gözlenmiştir. Sonuçta, CD ön işleminin deneylerde kullanılan tüm veri kümeleri için genel olarak kestirimcilerin performanslarını artıran bir etmen olduğu gözlenmiştir. Dolayısıyla, CD ön işleminin olumlu etkileri araştırmacılar tarafından dikkate alınmalıdır.

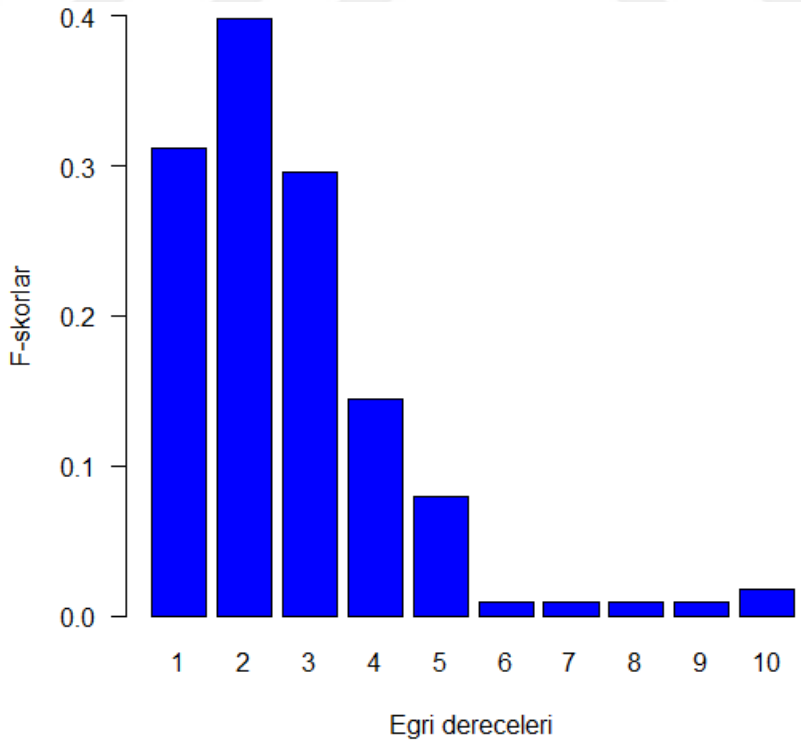
Son olarak, bu tez çalışmasında ayrıklaştırmaya ihtiyaç duyan MM ve CS kestirimcileri için Eş Frekans (EF) ve Eş Genlik (EG) ayrıklaştırma tekniklerinin performansa etkileri de incelenmiştir. N örnek sayısı, $\sqrt{N}$ de hücre sayısı olmak üzere karekökü tam alınabilenen N değerleri için CD ön işleminin kullanıldığı durumlarda veri örneklerinin EF ve EG ayrıklaştırma teknikleri ile hücrelere ayrılma sonucunun aynı olması beklenmektedir. Karekökü tam alınamayan N değerleri için de bu iki ayrıklaştırma tekniğinin benzer sonuçlanması beklenmektedir. Ancak CD ön işleminin kullanılmadığı, bir başka deyişle veri örneklerinin orijinal değerlerinin kullanıldığı durumlarda EF ve EG tekniklerinin ayrıklaştırma sonucunun farklı olması beklenmektedir. Bu nedenle CD ön işleminin kullanılmadığı durumları incelediğimizde (Şekil 6.9-6.15 arasındaki görsellerdeki tüm açık renkli çizgiler) MM ve CS kestirimcileri için özellikle ARACNE ve RelNet algoritmaları için EF tekniğinin çoğunlukla, hatta neredeyse tüm durumlarda EG'den daha başarılı olduğu gözlenmiştir.



Şekil 6. 15 Gerçek biyolojik veri kümesi için CD kullanma/kullanmama durumları için gözlenen Hassasiyet skorları

6.7. B-Spline Kestirimcisi için En Uygun Eğri Derecesi İncelemesi

Bu bölümde B-spline (BS) etkileşim skoru kestirimcisinde kullanılan eğri derecesi parametresinin en uygun değerinin ne olacağı araştırılmıştır. Bu değerlendirme yapılırken CD ön işleminin BS'in çıkarım performansı üzerindeki etkileri de incelenmiştir. Deneylerde iki farklı sentetik veri kümesi ve C3NET GAÇ algoritması kullanılmıştır. Kullanılan veri kümeleri SynTReN [27] simülatörü kullanılarak üretilmiştir ve gerçek ağları *E.Coli* bakterisinin bir alt ağıdır. İlk veri kümesinde 100 gen ve 100 örnek; ikincisinde 100 gen ve 1000 örnek bulunmaktadır. Böylece örnek sayısının BS kestirimcisinin çıkarım performansı üzerindeki etkileri de gözlenebilecektir. Çıkarım performansı, Bağıntı (6.1)'de tanımlanan F-skor ölçütü ile değerlendirilmiştir. BS kestirimcisindeki eğri derecesi 1'den 10'a kadar 1'er artırılarak değerlendirilmiştir. Bu inceleme, bu tezden üretilen bir konferans yayını olarak sunulmuştur [29]. İlk veri kümesi için CD ön işleminin kullanılmaması durumunda TP, FP ve FN bağlantıların adedi ile Hassasiyet (precision), Geri çekilme (recall) ve F-skor ölçütlerinin değerleri Çizelge 6.8'de verilmiştir (TP, FP ve FN tanımları Bölüm 6'nın girişinde verilmiştir). F-skor değerlerinin grafiği de Şekil 6.16'da verilmiştir.



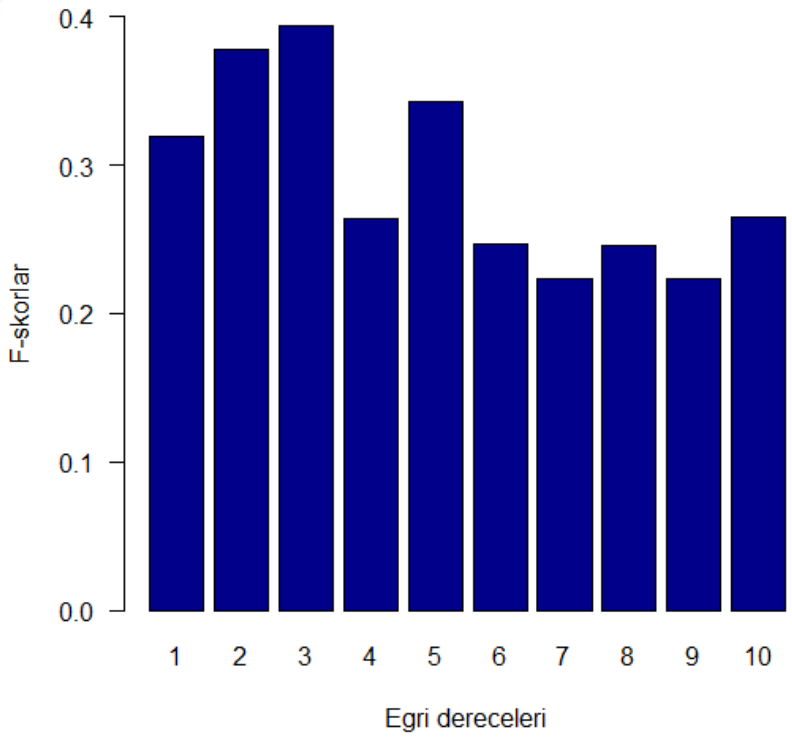
Şekil 6. 16 İlk veri kümesine CD uygulanmaması durumunda BS kestirimcisinin eğri derecesine göre performans değişimi

Çizelge 6. 8 İlk veri kümesine CD uygulanmaması durumunda BS kestirimcisinin eğri derecesine göre performans değişimi

Eğri derecesi	TP	FP	FN	Hassasiyet	Geri Çekilme	F-skor
1	33	50	96	0.3976	0.2558	0.3113
2	43	44	86	0.4943	0.3333	0.3982
3	32	55	97	0.3678	0.2481	0.2963
4	16	76	113	0.1739	0.1240	0.1448
5	9	89	120	0.0918	0.0698	0.0793
6	1	98	128	0.0101	0.0078	0.0088
7	1	98	128	0.0101	0.0078	0.0088
8	1	98	128	0.0101	0.0078	0.0088
9	1	98	128	0.0101	0.0078	0.0088
10	2	97	127	0.0202	0.0156	0.0175

Birinci veri kümesine CD ön işleminin uygulanmaması durumunda BS kestirimcisi en iyi performansı eğri derecesi 2 iken sergilemektedir. Eğri derecesi 4'ten büyük olduğunda BS kestirimcisi ciddi bir performans düşüşü sergilemektedir. Eğri derecesi 1'den 2'ye çıkartıldığında performansın ciddi ve önemli bir şekilde arttığı gözlenmiştir [29].

Birinci veri kümesine CD uygulanması durumu için de farklı eğri derecelerine göre performans değerlendirmesi yapılmış; sonuçları Şekil 6.17 ve Çizelge 6.9'da verilmiştir.

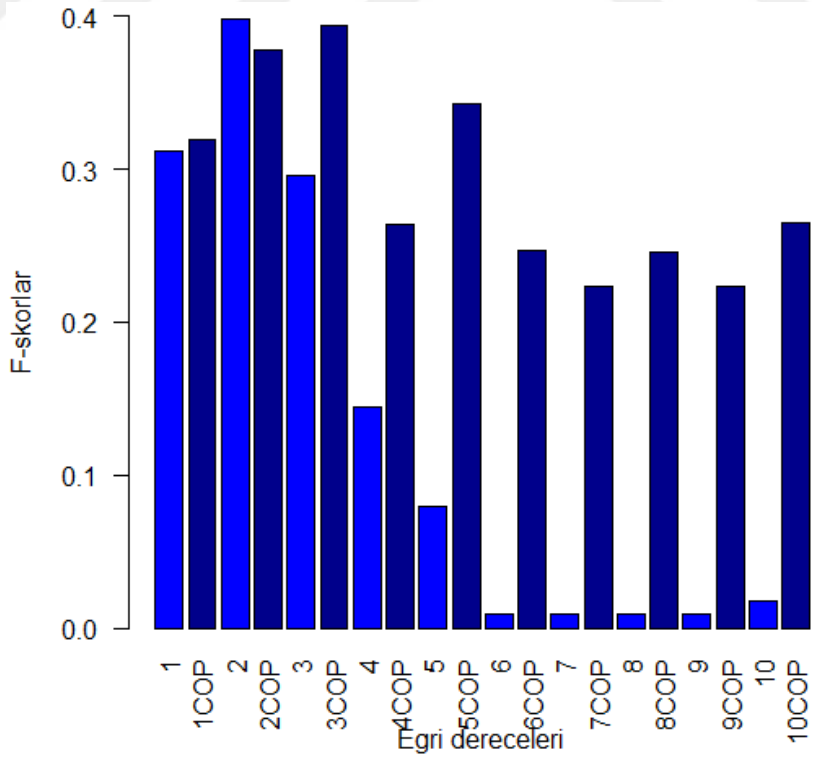


Şekil 6. 17 İlk veri kümesine CD uygulanması durumunda BS kestirimcisinin eğri derecesine göre performans değişimi

Çizelge 6. 9 İlk veri kümesine CD uygulanması durumunda BS kestirimcisinin eğri derecesine göre performans değişimi

Eğri derecesi	TP	FP	FN	Hassasiyet	Geri Çekilme	F-skor
1	34	50	95	0.4048	0.2636	0.3193
2	40	43	89	0.4819	0.3101	0.3774
3	42	42	87	0.5000	0.3256	0.3944
4	28	55	101	0.3374	0.2171	0.2642
5	36	45	93	0.4444	0.2791	0.3429
6	26	56	103	0.3171	0.2016	0.2465
7	24	62	105	0.2791	0.1861	0.2233
8	27	64	102	0.2967	0.2093	0.2455
9	24	62	105	0.2791	0.1861	0.2233
10	29	61	100	0.3222	0.2248	0.2648

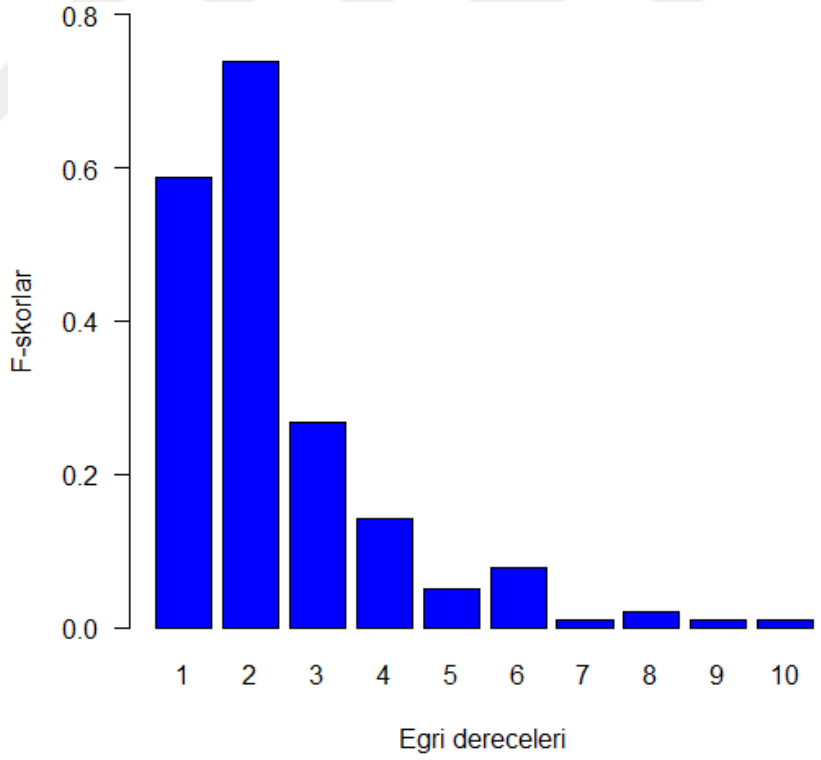
Birinci veri kümesine CD ön işlemi uygulanması durumunda BS kestirimcinin en başarılı performansın eğri derecesi 3 iken (BS3) sergilendiği gözlenmiştir. Ancak eğri derecesinin 2 olduğu durumun (BS2) F-skor sonucu da BS3'ün F-skor'una yakındır. CD ön işleminin kullanılması durumunda, CD'nin kullanılmadığı durumdan farklı olarak eğri derecesi 4'ten büyük olduğunda artık ciddi bir performans düşüşü gözlenmemektedir.



Şekil 6. 18 İlk veri kümesi için BS'in eğri derecesine göre genel değerlendirmesi

Şekil 6.18'de BS kestirimcisinin eğri derecesine göre başarım değişim grafiği CD ön işleminin kullanılma ve kullanılmama durumu için birlikte verilmiştir. Şekildeki "COP" ifadesi, yöntemin CD ile kullanıldığını göstermektedir. Buna göre, birinci veri kümesi için CD ön işleminin kullanımının 1'den 10'a kadar (2 hariç) tüm eğri dereceleri için F-skor cinsinden performansı artıran bir etken olduğu söylenebilir. Eğri derecesinin 2 olduğu durum için de (BS2) zaten CD ön işleminin kullanıldığı ve kullanılmadığı durumların performans sonuçları birbirine yakın olarak elde edilmiştir. Sonuç olarak, CD işleminin kullanıldığı ve kullanılmadığı durumların her ikisi de birlikte değerlendirildiğinde BS2'nin geri kalan eğri derecelerinden genel olarak daha başarılı performans sergilediği gözlenmiştir. Dolayısıyla birinci veri kümesi için BS2'nin uygun bir tercih olduğu söylenebilir.

İkinci veri kümesi için de CD ön işleminin kullanılmaması durumunda TP, FP ve FN bağlantıların adedi ile Hassasiyet, Geri çekilme ve F-skor ölçütlerinin değerleri Çizelge 6.10'da; F-skor değerlerinin grafiği de Şekil 6.19'da verilmiştir.



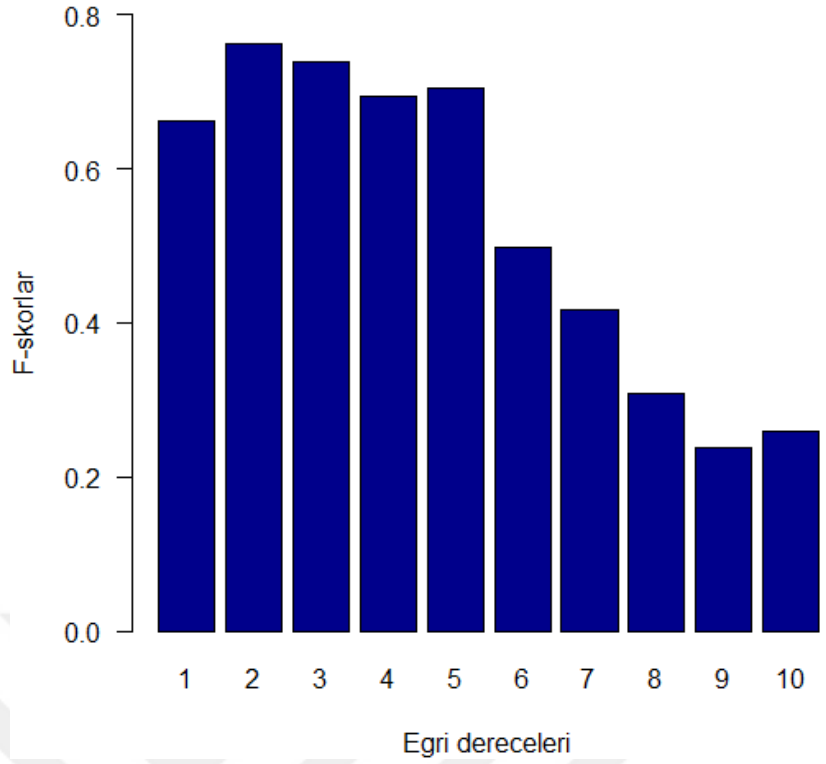
Şekil 6. 19 İkinci veri kümesine CD uygulanmaması durumunda BS kestirimcisinin eğri derecesine göre performans değişimi

Çizelge 6. 10 İkinci veri kümesine CD uygulanmaması durumunda BS kestirimcisinin eğri derecesine göre performans değişimi

Eğri derecesi	TP	FP	FN	Hassasiyet	Geri Çekilme	F-skor
1	55	28	49	0.6627	0.5289	0.5882
2	69	14	35	0.8313	0.6635	0.7380
3	26	64	78	0.2889	0.2500	0.2680
4	14	80	90	0.1489	0.1346	0.1414
5	5	91	99	0.0521	0.0481	0.0500
6	8	91	96	0.0808	0.0769	0.0788
7	1	98	103	0.0101	0.0096	0.0099
8	2	97	102	0.0202	0.0192	0.0197
9	1	98	103	0.0101	0.0096	0.0099
10	1	98	103	0.0101	0.0096	0.0099

İkinci veri kümesinde daha önce de belirtildiği üzere 1000 adet örnek bulunmaktadır. Bu veri kümesi için de CD kullanılmaması durumunda en başarılı tahmin eğri derecesi 2 iken (BS2) elde edilmektedir. BS2'yi eğri derecesinin 1 olduğu durum (BS1) takip etmektedir. Bu iki eğri derecesi birinci veri kümesi için de CD işleminin kullanılmaması durumunda en başarılı tahminleri yapmaktadır. Eğri derecesinin 3 olması (BS3) ilk veri kümesi için daha iyi tahminler yapmış ve BS1'e yakın sonuçlar vermişken; ikinci veri kümesi için BS3, BS1'e ve BS2'ye yakın performans sergileyememiştir. Dolayısıyla ikinci veri kümesi için CD ön işleminin kullanılmadığı durumda eğri derecesi 2 iken en başarılı tahminin yapıldığı gözlenmiştir.

İkinci veri kümesi için CD ön işleminin kullanılması durumunda elde edilen değerlendirme sonuçları Şekil 6.20 ve Çizelge 6.11'de verilmiştir. Bu veri kümesine CD ön işleminin uygulanması durumunda, birinci veri kümesinde de olduğu gibi, tüm eğri derecelerinin ciddi bir performans artışı sergilediği gözlenmiştir. Eğri derecesi 2 olarak alındığında CD'nin kullanıldığı ve kullanılmadığı her iki durum için de en iyi F-skor sonucu alınmıştır. CD kullanıldığında, eğri derecesinin 3 olduğu durumda BS yöntemi en iyi ikinci performansı sergilemiştir. CD ön işleminin kullanıldığı durumda her iki veri kümesi için de BS2 ve BS3'ün sonuçları birbirine yakın olarak elde edilmiştir. Ayrıca eğri derecelerinin birçoğu için örnek sayısının artması, CD'nin kullanılmadığı durumlar için bile F-skor değerlerinin ciddi oranda artmasını sağlamıştır.

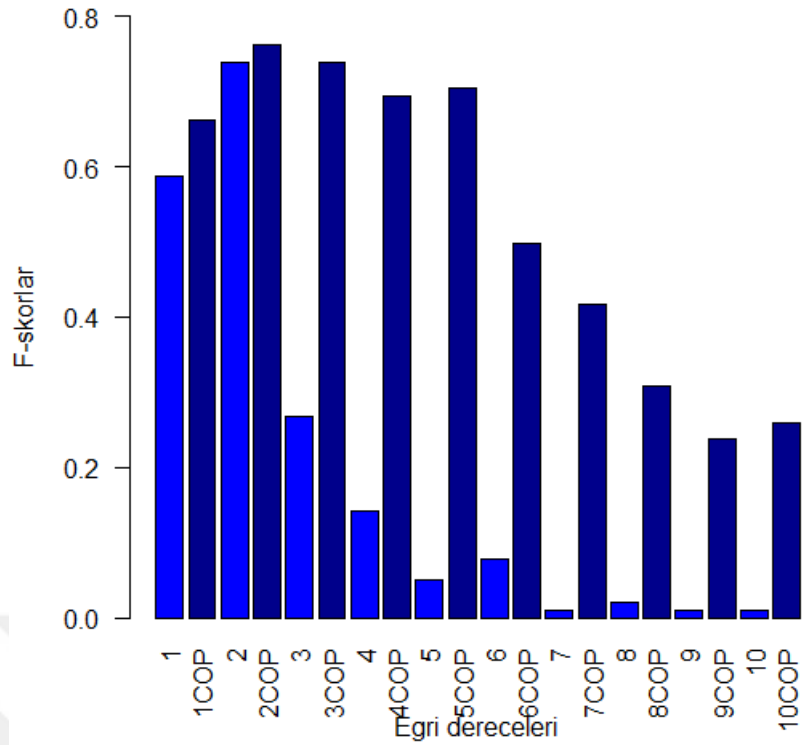


Şekil 6. 20 İkinci veri kümesine CD uygulanması durumunda BS kestirimcisinin eğri derecesine göre performans değişimi

Çizelge 6. 11 İkinci veri kümesine CD uygulanması durumunda BS kestirimcisinin eğri derecesine göre performans değişimi

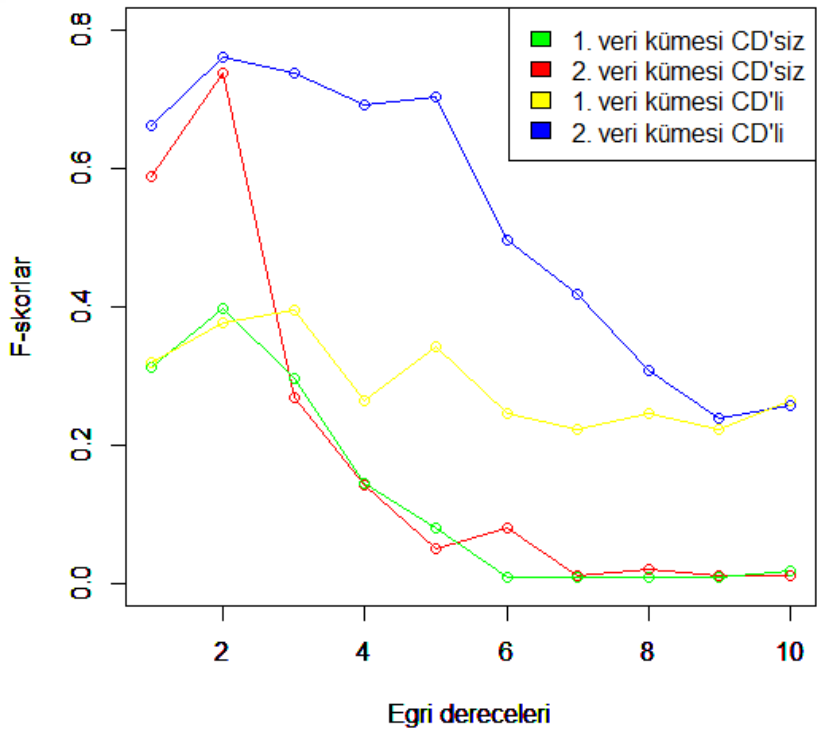
Eğri derecesi	TP	FP	FN	Hassasiyet	Geri Çekilme	F-skor
1	58	13	46	0.8169	0.5577	0.6629
2	67	5	37	0.9306	0.6442	0.7614
3	65	7	39	0.9028	0.6250	0.7386
4	61	11	43	0.8472	0.5865	0.6932
5	62	10	42	0.8611	0.5962	0.7046
6	44	29	60	0.6028	0.4231	0.4972
7	38	40	66	0.4872	0.3654	0.4176
8	28	50	76	0.3590	0.2692	0.3077
9	22	59	82	0.2716	0.2115	0.2378
10	24	58	80	0.2927	0.2581	0.2581

Şekil 6.21'de BS kestirimcisinin eğri derecesine göre başarım değişimi 2. veri kümesine CD ön işleminin uygulanma ve uygulanmama durumu için birlikte verilmiştir. Şekildeki "COP" ifadesi, yöntemin CD ile kullanıldığını göstermektedir. Görüldüğü gibi ikinci veri kümesine CD uygulandığında istisnasız tüm eğri derecelerinin performansı artmıştır.



Şekil 6. 21 İkinci veri kümesi için BS'in eğri derecesine göre genel değerlendirmesi

Her iki veri kümesi için de CD'nin uygulanma ve uygulanmama durumlarında eğri derecesine göre F-skor değerlerinin değişimi Şekil 6.22'de birlikte verilmiştir.



Şekil 6. 22 Her iki veri kümesine göre CD kullanılma/kullanılmama durumuna göre genel değerlendirmesi

Şekil 6.22'de BS kestirimcinin eğri derecesine göre genel başarım değerlendirmesi her iki veri kümesi ve her iki durum için (CD kullanma ve kullanmama) verilmiştir. Buna göre, birçok eğri derecesi için CD işleminin kullanımı ve örnek sayısının artırılması BS kestirimcinin çıkarım performansını artırmaktadır. CD kullanımı, özellikle daha fazla örnek içeren ikinci veri kümesi için eğri derecelerinin performanslarını ciddi bir oranda artırmaktadır. Bu artış, büyük eğri dereceleri için daha da fark edilirdir. Sonuçlara genel olarak bakıldığında, eğri derecesinin 2 olduğu durumun veri kümelerine CD uygulansın veya uygulanmasın en iyi tercih olduğu gözlenmiştir. En iyi performansın alındığı eğri derecesinin 2 olma durumu için CD kullanımının performansa etkisi diğer eğri derecelerine nazaran çok daha azdır.

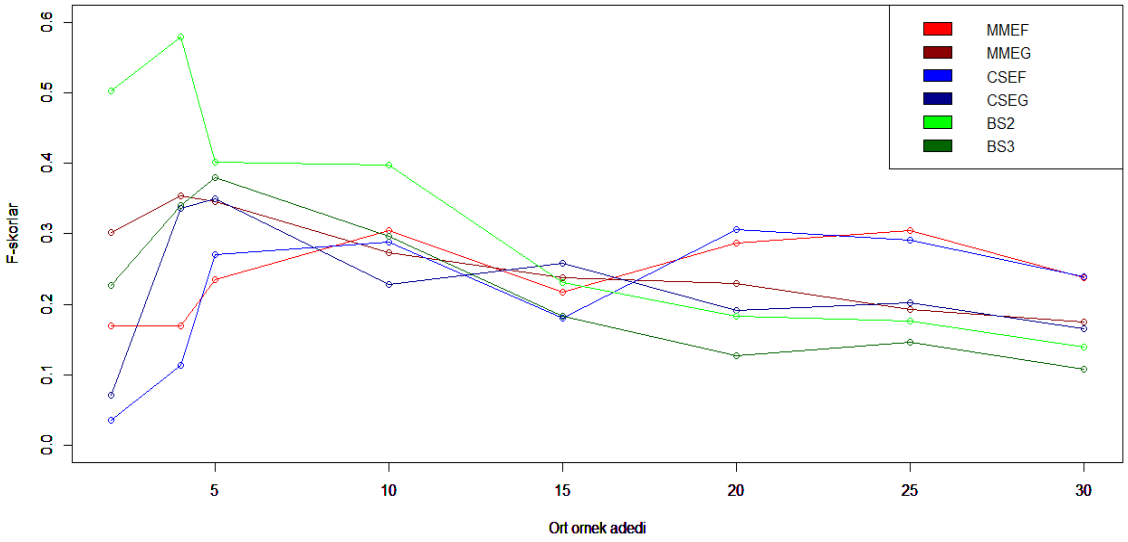
6.8. Ayrıklaştırma İşlemi için En Uygun Hücre Sayısının İncelemesi

Bu tez çalışmasında ayrıklaştırmaya ihtiyaç duyan entropi-tabanlı kestirimciler için örnek sayısına bağlı olarak en uygun hücre sayısı (veya hücre başına düşen en uygun ortalama örnek sayısı) tespit edilmeye çalışılmıştır. Literatürde ayrıklaştırma işlemi için örnek sayısı N olmak üzere; hücre sayısı $\sqrt{N}$ olarak alınmaktadır [17], [18]. Bilgimiz dahilinde literatürde hücre sayısının bu şekilde seçilmesinin sebebi incelenmemiştir.

Bu tez çalışmasında kullanılan kestirimcilerden üçü ayrıklaştırmaya ihtiyaç duymaktadır. Bunlar: B-spline (BS), Chao-Shen (CS) ve Miller-Madow (MM) yöntemleridir. BS kestirimcisi için ayrıklaştırma işlemi B-spline fonksiyonları ile tanımlanan dönüşüm (transfer) fonksiyonları kullanılarak gerçekleştirilir. Bu işlem, MM ve CS için uygulanan ayrıklaştırma yaklaşımından farklı bir süreçtir. BS'deki dönüşüm fonksiyonları ile yapılan ayrıklaştırma işlemi sonucunda elde edilen hücrelerin sınırları, MM ve CS kestirimcilerinin hücre sınırlarından daha farklıdır. Bu nedenle, Bölüm 3.2'de anlatılan Eş Frekans (EF) ve Eş Genişlik (EG) ayrıklaştırma tekniklerinin çıkarım performansı üzerindeki etkisinin incelendiği esas kestirimciler MM ve CS yöntemleridir. Bunlara ilaveten BS kestirimcisinin eğri derecesinin 2 ve 3 olduğu durumlar için de en uygun hücre sayısı incelemesi yapılmıştır. Bölüm 3.3.9'da anlatıldığı üzere, BS'teki eğri derecesi bir örneğin ayrıklaştırma sonucunda aynı anda kaç hücreye üye olabileceğini göstermektedir. En uygun hücre sayısının belirlenmesi için yapılan deneylerde iki sentetik veri kümesi ve C3NET GAÇ algoritması kullanılmıştır. Bu veri kümelerinden ilki

100 gen ve 100 örnek; ikincisi ise 100 gen ve 1000 örnek içermektedir. Veri kümelerinin doğrulama gen ağları *E.Coli* bakterisinin bir alt ağından alınmıştır. Bunlara ilaveten, kestirimcilerin optimal hücre sayısı, CD ön işleminin veri kümesine uygulanma ve uygulanmama durumları için ayrı ayrı incelenmiştir.

100 örnek içeren ilk veri kümesi için ayrıklaştırma sonucunda her bir hücredeki ortalama örnek adedi {2, 4, 5, 10, 15, 20, 25, 30} değerlerinden birini alacak şekilde ayrıklaştırma işlemi yapılmıştır. Dolayısıyla, hücre sayısı sırayla {50, 25, 20, 10, 6, 5, 4, 3} olacak şekilde deneyler yapılmıştır. İlk veri kümesine CD işleminin uygulanmaması durumunda hücre başına düşen ortalama örnek adedine göre MMEF (EF tekniği ile MM), MMEG (EG tekniği ile MM), CSEF (EF tekniği ile CS), CSEG (EG tekniği ile CS), BS2 (eğri derecesi 2 olan BS) ve BS3 (eğri derecesi 3 olan BS) için elde edilen çıkarım performansları Şekil 6.23 ve Çizelge 6.12'de veirlmiştir.



Şekil 6. 23 İlk veri kümesine CD uygulanmaması durumunda her bir hücredeki ortalama örnek adedine göre kestirimcilerin performans değerlendirmesi

Deneylerde kullanılan ilk veri kümesine CD işlemi uygulanmaması durumunda, en iyi F-skor sonucu BS2 kestirimcisi için hücre başına düşen ortalama örnek sayısının 4 olduğu durumda (ort örnek #=4) elde edilmiştir. Ortalama örnek sayısının {10, 20, 25, 30} değerlerinden biri olduğu durumlarda MM ve CS için EF tekniğinin EG'den daha yüksek F-skor sonucu verdiği; ancak MM ve CS için ortalama örnek sayısı sırayla 4 ve 5 iken EG ayrıklaştırma tekniğinin en iyi F-skor sonuçlarını verdiği gözlenmiştir. BS2 ve BS3 için de en iyi performanslar sırayla her bir hücredeki ortalama örnek sayısının 4 ve 5 olduğu

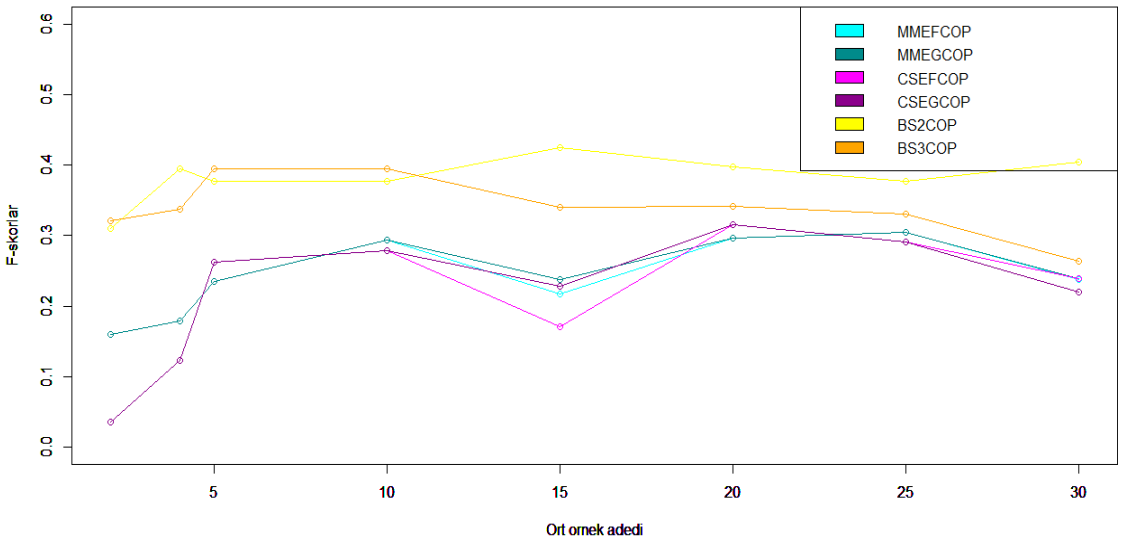
durumlarda elde edilmiştir. Dolayısıyla kestirimciler için genel olarak en iyi sonuçların ortalama örnek sayısı 4 veya 5 iken elde edildiği söylenebilir. Bu da hücre sayısının $2\times\sqrt{N}$ olmasına karşılık gelmektedir. Ancak hücre sayısının $2\times\sqrt{N}$ olduğu ve en başarılı performansların alındığı durumun F-skor değerleri, varsayılan durum olan hücre sayısının $\sqrt{N}$ olduğu durumun F-skor değerlerine yakındır. Ayrıca Şekil 6.23'te görüldüğü üzere, her bir hücredeki ortalama örnek adedine göre performans değişimleri, BS2'nin ortalama örnek sayısının 4 olduğu durum haricinde derin zigzag'lar içermemektedir. Özellikle ortalama örnek sayısı 4'ten fazla iken, kestirimci performansları arasında daha yumuşak geçişler bulunmaktadır. Sonuç olarak, ayrıklaştırma için hücre sayısının $\sqrt{N}$ olarak alınması makul görülmektedir.

Çizelge 6. 12 İlk veri kümesine CD uygulanmaması durumunda her bir hücredeki ortalama örnek adedine göre kestirimcilerin performans değerlendirmesi

Ort Örnek #	**Ölçüt**	**MMEF**	**MMEG**	**CSEF**	**CSEG**	**BS2**	**BS3**
2	**F-skor**	0.1690	0.3014	0.0352	0.0714	0.5023	0.2262
	Hassasiyet	0.2143	0.3667	0.0408	0.0842	0.6111	0.2717
4	**F-skor**	0.1698	0.3535	0.1132	0.3365	0.5794	0.3395
	Hassasiyet	0.2169	0.4419	0.1446	0.4235	0.7294	0.4157
5	**F-skor**	0.2347	0.3458	0.2710	0.3491	0.4019	0.3796
	Hassasiyet	0.2976	0.4353	0.3412	0.4458	0.5059	0.4713
10	**F-skor**	0.3048	0.2736	0.2886	0.2275	0.3982	0.2963
	Hassasiyet	0.3951	0.3494	0.4028	0.2927	0.4943	0.3678
15	**F-skor**	0.2170	0.2381	0.1801	0.2584	0.2315	0.1835
	Hassasiyet	0.2771	0.3086	0.2317	0.3375	0.2874	0.2247
20	**F-skor**	0.2871	0.2300	0.3062	0.1914	0.1835	0.1273
	Hassasiyet	0.3750	0.3000	0.4000	0.2500	0.2247	0.1539
25	**F-skor**	0.3048	0.1923	0.2913	0.2019	0.1759	0.1455
	Hassasiyet	0.3951	0.2532	0.3896	0.2658	0.2184	0.1758
30	**F-skor**	0.2381	0.1748	0.2392	0.1659	0.1389	0.1081
	Hassasiyet	0.3086	0.2338	0.3125	0.2237	0.1724	0.1290

İlk veri kümesine CD ön işleminin uygulanması durumunda kestirimcilerin hücre başına düşen ortalama örnek sayısına göre performans değişimi Şekil 6.24 ve Çizelge 6.13'te verilmiştir. BS2 kestirimcisinin ortalama eleman sayısının 15 olduğu (hücre sayısının 6 olduğu) durumda tüm kestirimciler arasında en iyi performansı sergilediği görülmüştür. BS2'yi BS3 kestirimcisi takip etmektedir. BS3 için en iyi performans ortalama örnek sayısı {5,10} değerlerinden birini aldığında (hücre sayısı 10 veya 20 iken) gözlenmiştir.

Bölüm 6.6'nın sonunda belirtildiği üzere, CD ön işleminin kullanılması EF ve EG tekniklerinin etkilerinin yaklaşık aynı olmasına sebep olur. Dolayısıyla MMEF ve MMEG yöntemleri CD kullanılmasıyla, (ortalama örnek sayısının 15 ve 30 olduğu durumlar haricinde) beklendiği gibi yakın sonuçlar vermişlerdir. Benzer şekilde CSEF ve CSEG yöntemleri de aynı istisnai durumlar haricinde aynı skorlar vermişlerdir. MM ve CS kestirimcileri için en iyi sonuçlar, hücre başına düşen ortalama örnek sayısı sırayla 25 ve 20 iken (başka bir deyişle hücre sayısı sırayla 4 ve 5 iken) elde edilmiştir. Yine MM, CS ve BS için en başarılı sonuçların, varsayılan durum olan hücre sayısı $\sqrt{N}$ iken (ortalama örnek sayısı 10 iken) elde edilen sonuçlara olukça yakın olduğu gözlenmiştir. Dolayısıyla, CD işleminin uygulanmadığı durumda da olduğu gibi, hücre sayısının $\sqrt{N}$ olarak alınmasının makul olduğu gözlenmiştir.

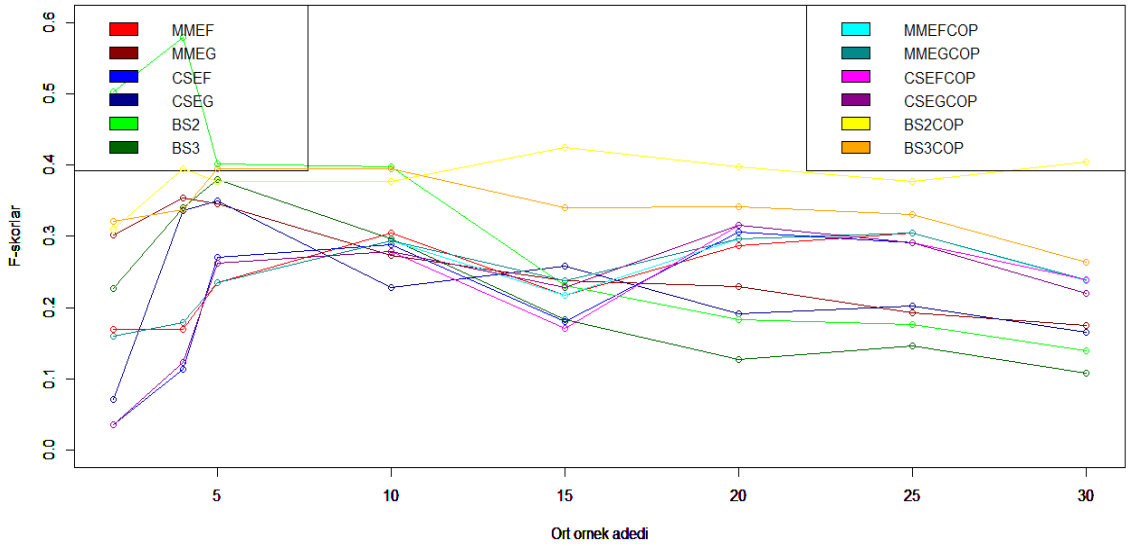


Şekil 6. 24 İlk veri kümesine CD uygulanması durumunda her bir hücredeki ortalama örnek adedine göre kestirimcilerin performans değerlendirmesi

Kestirimcilerin ortalama örnek sayısına göre performans değişimini ilk veri kümesine CD ön işleminin uygulanma ve uygulanmama durumları için birlikte gösteren grafik Şekil 6.25'te verilmiştir.

Çizelge 6. 13 İlk veri kümesine CD uygulanması durumunda her bir hücredeki ortalama örnek adedine göre kestirimcilerin performans değerlendirmesi

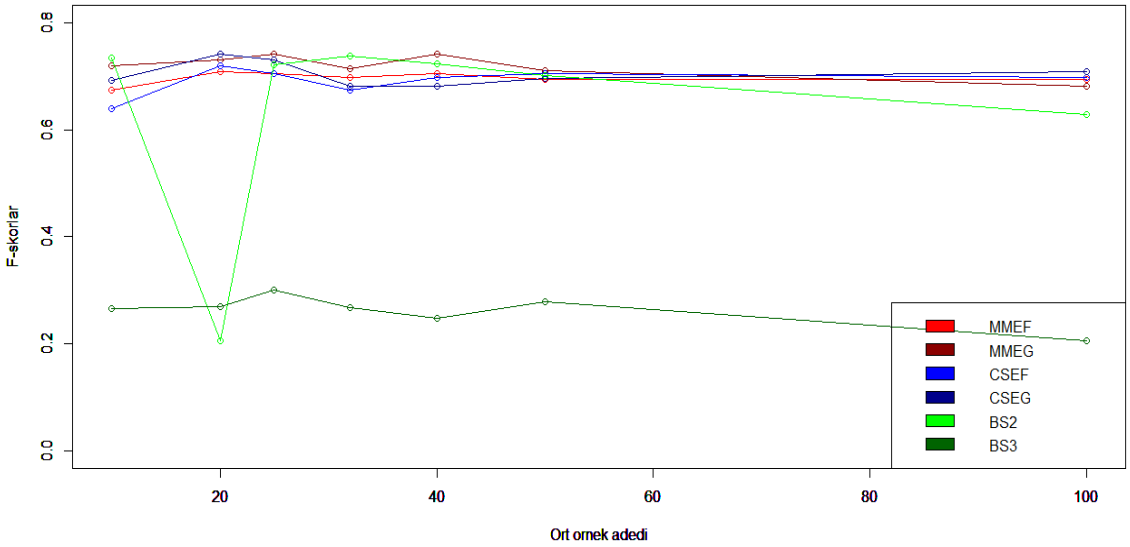
Ort Örnek #	Ölçüt	MMEF	MMEG	CSEF	CSEG	BS2	BS3
2	**F-skor**	0.1596	0.1596	0.0352	0.0352	0.3099	0.3208
	Hassasiyet	0.2024	0.2024	0.0408	0.0408	0.3929	0.4096
4	**F-skor**	0.1792	0.1792	0.1226	0.1226	0.3944	0.3380
	Hassasiyet	0.2289	0.2289	0.1566	0.1566	0.5000	0.4286
5	**F-skor**	0.2347	0.2347	0.2629	0.2629	0.3774	0.3944
	Hassasiyet	0.2976	0.2976	0.3333	0.3333	0.4819	0.5000
10	**F-skor**	0.2938	0.2938	0.2786	0.2786	0.3774	0.3944
	Hassasiyet	0.3781	0.3781	0.3889	0.3889	0.4819	0.5000
15	**F-skor**	0.2170	0.2381	0.1706	0.2275	0.4245	0.3396
	Hassasiyet	0.2771	0.3086	0.2195	0.2927	0.5422	0.4337
20	**F-skor**	0.2967	0.2967	0.3158	0.3158	0.3981	0.3412
	Hassasiyet	0.3875	0.3875	0.4125	0.4125	0.5122	0.4390
25	**F-skor**	0.3048	0.3048	0.2913	0.2913	0.3774	0.3302
	Hassasiyet	0.3951	0.3951	0.3896	0.3896	0.4819	0.4217
30	**F-skor**	0.2381	0.2392	0.2392	0.2201	0.4038	0.2642
	Hassasiyet	0.3086	0.3125	0.3125	0.2875	0.5119	0.3374



Şekil 6. 25 İlk veri kümesine CD uygulanması/uygulanmaması durumları için her bir hücredeki ortalama örnek sayısına göre kestirimci performanslarının değerlendirilmesi

100 gen ve 1000 örnek içeren ikinci sentetik veri kümesi için de ayrıklaştırma sürecinde kullanılan hücre başına düşen ortalama örnek sayısının değerlendirilmesi yapılmıştır. Ayrıca kestirimci performansları yine CD ön işleminin kullanılıp kullanılmamasına göre de değerlendirilmiştir. Bu veri kümesi için hücre başına düşen ortalama örnek sayıları sırayla {10, 20, 25, 32 (1000'in karekökü), 40, 50, 100} değerlerinden biri olarak

seçilmiştir. Dolayısıyla hücre sayıları sırayla {100, 50, 40, 32, 25, 20, 10} şeklindedir. Bu 7 farklı durum için ikinci veri kümesine CD işleminin uygulanmaması durumunda elde edilen performans değerleri Şekil 6.26 ve Çizelge 6.14'te verilmiştir.



Şekil 6. 26 İkinci veri kümesine CD uygulanmaması durumunda her bir hücredeki ortalama örnek adedine göre kestirimcilerin performans değerlendirmesi

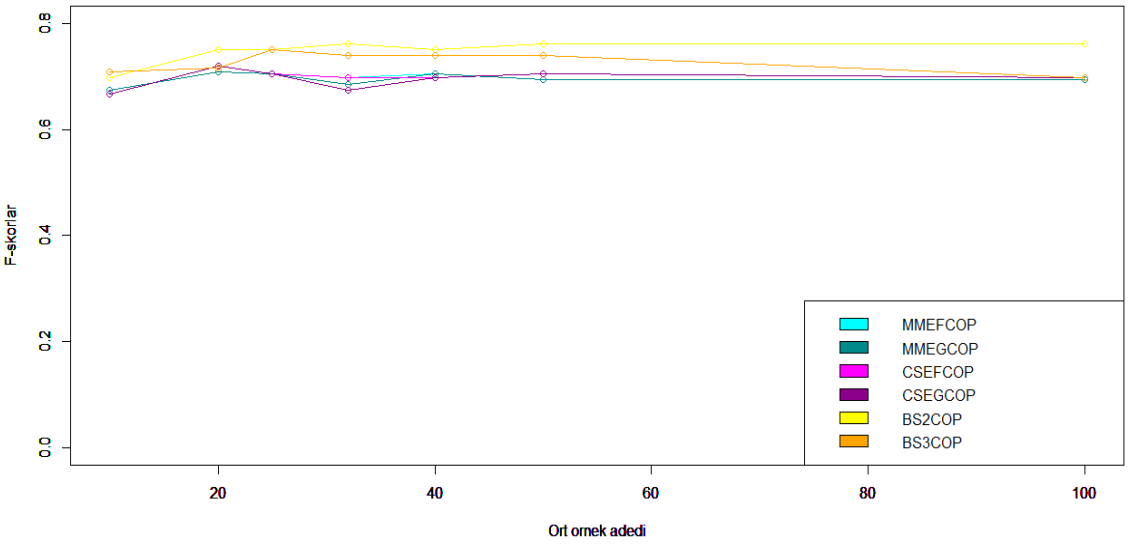
Çizelge 6. 14 İkinci veri kümesine CD uygulanmaması durumunda her bir hücredeki ortalama örnek adedine göre kestirimcilerin performans değerlendirmesi

Ort Örnek #	Ölçüt	MMEF	MMEG	CSEF	CSEG	BS2	BS3
10	**F-skor**	0.6743	0.7196	0.6400	0.6919	0.7340	0.2667
	Hassasiyet	0.8310	0.8000	0.7887	0.7901	0.8214	0.2857
20	**F-skor**	0.7086	0.7312	0.7200	0.7419	0.2051	0.2694
	Hassasiyet	0.8732	0.8293	0.8873	0.8415	0.2198	0.2921
25	**F-skor**	0.7046	0.7419	0.7046	0.7312	0.7204	0.3005
	Hassasiyet	0.8611	0.8415	0.8611	0.8293	0.8171	0.3258
32	**F-skor**	0.6971	0.7135	0.6743	0.6811	0.7380	0.2680
	Hassasiyet	0.8592	0.8148	0.8310	0.7778	0.8313	0.2889
40	**F-skor**	0.7046	0.7419	0.6971	0.6811	0.7234	0.2474
	Hassasiyet	0.8611	0.8415	0.8592	0.7778	0.8095	0.2667
50	**F-skor**	0.6932	0.7097	0.7046	0.6952	0.7021	0.2784
	Hassasiyet	0.8472	0.8049	0.8611	0.7831	0.7857	0.3000
100	**F-skor**	0.6932	0.6809	0.6971	0.7090	0.6283	0.2051
	Hassasiyet	0.8472	0.7619	0.8750	0.7882	0.5769	0.2198

CD ön işleminin ikinci veri kümesine uygulanmaması durumunda BS3 haricindeki tüm kestirimcilerin F-skor değerlerinin birbirine yakın olduğu görülmüştür. BS3'ün ciddi bir şekilde performans düşüşü sergilediği gözlenmiştir. İlk veri kümesine göre daha çok

örnek içeren ikinci veri kümesi için BS3 dışındaki tüm kestirimcilerin performanslarının genel olarak arttığı gözlenmiştir. En iyi performans iki ayrı kestirimciden üç farklı durum için alınmıştır. Bunlar CSEG için hücre başına düşen ortalama örnek sayısının 20 olduğu, MMEG için ortalama örnek sayısının 25 ve 40 olduğu durumlardır. MMEG, MMEF'ye nazaran 6 durum için (ortalama örnek sayısı {10, 20, 32, 40, 50, 100} iken) daha başarılı sonuçlar vermiştir. CSEG de 5 durum için CSEF'den daha başarılı sonuçlar sergilemiştir. Dolayısıyla örnek sayısının artması ile EG tekniğinin EF'den daha başarılı olduğu gözlenmiştir. BS2 de MM ve CS'ye yakın F-skor sonuçları vermiştir. Yine kestirimciler için örnek sayısına göre performans değişiminde derin zigzaglar gözlenmemiş, varsayılan durum olan $\sqrt{N}$ 'in sonucu en iyi sonuçlara yakın çıkmıştır. Sonuç olarak bu veri kümesi için de $\sqrt{N}$ 'in hücre sayısı için makul bir tercih olduğu söylenebilir.

İkinci veri kümesine CD işleminin uygulanması durumu için de kestirimcilerin ortalama örnek sayısına göre değerlendirmeleri yapılmıştır. Bunun sonuçları da Şekil 6.27 ve Çizelge 6.15'te verilmiştir. CS ve MM kestirimcilerinin CD işleminden pek etkilenmediği, CD'nin kullanılmadığı durumlardaki F-skor sonuçlarına yakın performanslar sergiledikleri gözlenmiştir. BS2 ve özellikle de BS3 kestirimcileri CD'nin kullanılmasından etkilenmişlerdir. BS2'nin CD kullanımı ile tüm ortalama örnek sayılarına göre performansının bir miktar arttığı gözlenmiştir. BS3'teki artışın ise ciddi ve önemli bir oranda olduğu görülmüştür.

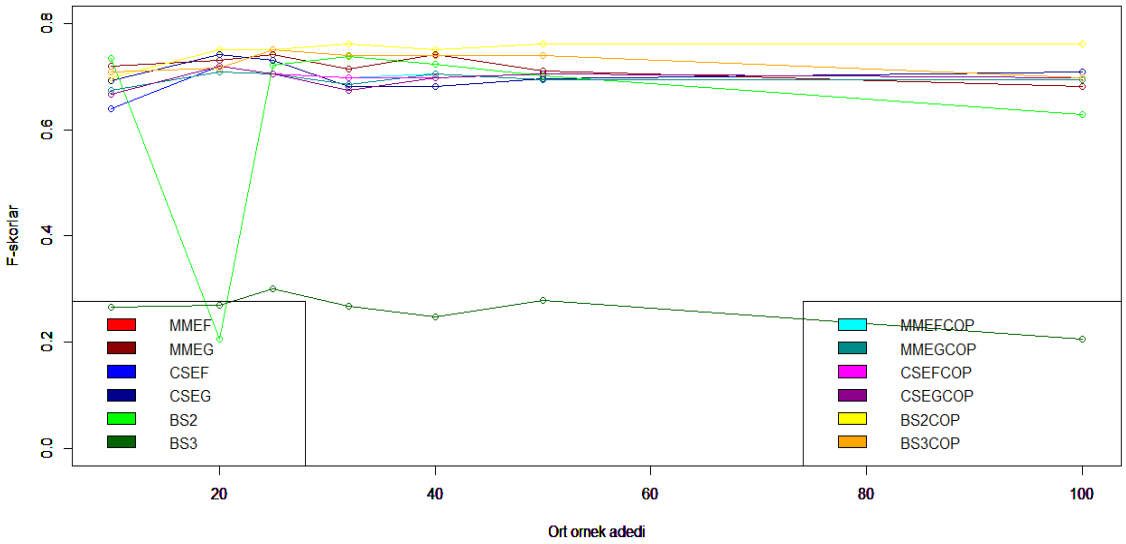


Şekil 6. 27 İkinci veri kümesine CD uygulanması durumunda her bir hücredeki ortalama örnek adedine göre kestirimcilerin performans değerlendirmesi

Çizelge 6. 15 İkinci veri kümesine CD uygulanması durumunda her bir hücredeki ortalama örnek adedine göre kestirimcilerin performans değerlendirmesi

Ort Örnek #	Ölçüt	MMEF	MMEG	CSEF	CSEG	BS2	BS3
10	**F-skor**	0.6743	0.6743	0.6667	0.6667	0.6971	0.7086
	Hassasiyet	0.8310	0.8310	0.8286	0.8286	0.8592	0.8732
20	**F-skor**	0.7086	0.7086	0.7200	0.7200	0.7500	0.7159
	Hassasiyet	0.8732	0.8732	0.8873	0.8873	0.9167	0.8750
25	**F-skor**	0.7046	0.7046	0.7046	0.7046	0.7500	0.7500
	Hassasiyet	0.8611	0.8611	0.8611	0.8611	0.9167	0.9167
32	**F-skor**	0.6971	0.6857	0.6971	0.6743	0.7614	0.7386
	Hassasiyet	0.8592	0.8451	0.8592	0.8310	0.9306	0.9028
40	**F-skor**	0.7046	0.7046	0.6971	0.6971	0.7500	0.7386
	Hassasiyet	0.8611	0.8611	0.8592	0.8592	0.9167	0.9028
50	**F-skor**	0.6932	0.6932	0.7046	0.7046	0.7614	0.7386
	Hassasiyet	0.8472	0.8472	0.8611	0.8611	0.9306	0.9028
100	**F-skor**	0.6932	0.6932	0.6971	0.6971	0.7614	0.6971
	Hassasiyet	0.8472	0.8472	0.8750	0.8750	0.9306	0.8592

Kestirimcilerin ortalama örnek sayısına göre performans değişimini ikinci veri kümesine CD ön işleminin uygulanma ve uygulanmama durumları için birlikte gösteren grafik Şekil 6.28'de verilmiştir.



Şekil 6. 28 İkinci veri kümesine CD uygulanması/uygulanmaması durumları için her bir hücredeki ortalama örnek sayısına göre kestirimci performanslarının değerlendirilmesi

İkinci veri kümesine CD ön işleminin uygulanması durumunda BS2'nin en iyi sonuçları verdiği gözlenmiştir. Aynı durum için BS3, BS2'yi takip etmektedir. MM ve CS'nin performansları da birbirine yakındır. CD kullanımı ile MMEF ve MMEG kestirimcileri bir durum haricinde (ortalama örnek sayısı 32 iken) beklendiği gibi aynı sonuçları vermiş; CSEF ve CSEG de yine aynı durum haricinde beklendiği üzere aynı performansı sergilemiştir. BS2 için en iyi sonuçlar hücre başına düşen ortalama örnek sayısı 32, 50 veya 100 iken elde edilmiştir. BS3 için ise en iyi sonuçlar ortalama örnek sayısı 25 iken elde edilmiştir. MM ve CS için ise en iyi sonuçlar ortalama örnek sayısı 20 iken elde edilmiştir. Bu gözlem, ilk veri kümesine CD uygulanması durumunda elde edilen sonuçla aynıdır. Ayrıca yine bu değerler de varsayılan durum olan ve hücre sayısının $\sqrt{N}$ olarak alındığı durumun performans değerlerine oldukça yakındır. Bu durum yine birinci sentetik veri kümesi için de aynı şekilde gözlenmiştir. Bunlara ilaveten, Şekil 6.27 kestirimci performansları için derin zigzaglar içermemekte; ortalama örnek sayıları boyunca yumuşak geçişler sergilemektedir. Dolayısıyla ilk veri kümesi için olduğu gibi bu veri kümesi için de ortalama eleman sayısını ve hücre sayısını $\sqrt{N}$ olarak almak makul bir tercihtir.

Sonuç olarak incelenen her iki veri kümesi için de, CD ön işleminin kullanıma ve kullanılmama durumları için de hücre sayısının $\sqrt{N}$ olarak alınmasının doğru bir tercih olduğu gözlenmiştir.

BÖLÜM 7

SONUÇ VE ÖNERİLER

Bu tez çalışmasında GAÇ uygulamalarında kullanılan ve kullanılması olası olan etkileşim skoru kestirimcileri detaylı bir şekilde incelenmiştir. Bu bağlamda, literatürde bulunan 27 farklı kestirimciden 12 tanesi kullanılmak üzere seçilmiştir. Seçilen 12 kestirimciden bir kısmı türevleri ile ele alınmış, böylece gerçeklenen ve karşılaştırılan kestirimcilerin sayısı 15'e tamamlanmıştır. İncelenen 27 kestirimci çeşitli bakış açılarına göre sınıflandırılmış; seçilen kestirimcilerin bir kısmı için en uygun parametre değerlerinin ne olacağı incelenmiştir. Bilgimiz dahilinde literatürde etkileşim kestirimcilerinin bu kadar detaylı incelendiği başka bir çalışma bulunmamaktadır.

Etkileşim skoru kestirimcileri üç farklı GAÇ algoritması ve *E.coli* bakterisine ait olan dört farklı gen ifade veri kümesi kullanılarak karşılaştırılmışlardır. Böylece kestirimcilerin değişen koşullardaki çıkarım performansları değerlendirilmiş ve performanslarına dair genelleme yapılabileceği gösterilmiştir. Ayrıca veri kümelerindeki örnek sayısının artması ile kestirimcilerin genel olarak çıkarım performanslarının da arttığı gözlenmiştir. Değerlendirme aşamasında kullanılan dört veri kümesinden üçü sentetik, biri gerçek biyolojik veri kümesidir. Sentetik veri kümeleri hem F-skor hem de Hassasiyet ölçütlerine göre değerlendirilmiş; her iki ölçütün birbirleriyle uyumlu sonuçlar sergilediği gösterilmiştir. Gerçek biyolojik veri kümelerinin doğrulama ağları henüz tam bilinemediği ve eksikleri olduğu için F-skor değerleri düşük çıkmaktadır ve bu veri kümeleri için Hassasiyet ölçütüne göre değerlendirme yapmanın daha makul ve sağlıklı olduğu gözlenmiştir. Halihazırda sentetik veri kümeleri için F-skor ve Hassasiyet ölçütüne göre yapılan değerlendirmelerin de uyumlu olduğu görüldüğü için bu çalışmada da gerçek biyolojik veri kümeleri Hassasiyet ölçütüne göre

değerlendirilmiştir. Yapılan değerlendirmeler sonucunda tüm veri kümeleri için en başarılı kestirimcilerin BS, ÇYK, PTG ve STG yöntemleri olduğu gözlenmiştir. Performans ölçütlerinin yanı sıra çalışma süreleri de değerlendirmeye alındığında, ÇYK'nın çalışma süresinin diğer yöntemlere nazaran ciddi oranda daha büyük olduğu gözlenmiş; BS, PTG ve STG yöntemlerinin daha çok tercih edilebilir olduğu kanısına varılmıştır. Buna ilaveten GAÇ algoritmaları arasında RelNet diğerlerinden farklı olarak dolaylı bağlantıları ağdan eleyemediği için çıkarım performansı ARACNE ve C3NET'in performanslarından daha düşüktür. ARACNE ve C3NET'in performansları birbirine yakındır, F-skora göre ARACNE'nin genel olarak C3NET'ten biraz daha iyi olduğu; Hassasiyet ölçütüne göre de C3NET'in daha iyi olduğu gözlenmiştir. Ancak bu çalışmanın temel amacının GAÇ algoritmalarını değil; etkileşim kestirimcilerini değerlendirmek olduğu unutulmamalıdır.

Etkileşim skoru kestirimcileri, veri kümesine uygulanan bir ön işlem olan CD'nin kullanılma ve kullanılmama durumlarına göre de ayrı ayrı değerlendirilmiş ve bununla ilgili ayrıntılı deneysel sonuçlar verilmiştir. CD ön işleminin veri kümelerine uygulanması durumunda kestirimcilerin çıkarım performanslarının arttığı gözlenmiştir. Bilgimiz dahilinde literatürde CD ön işleminin kestirimcilerin performansları üzerindeki etkilerinin incelendiği bir çalışma bulunmamaktadır. Bu tez çalışması bu yönüyle de bir ilktir.

Bunlara ilaveten, bu çalışmada ayrıklaştırmaya ihtiyaç duyan yöntemler (MM ve CS) için ayrıklaştırma tekniklerinin çıkarım performansına etkileri ve ayrıklaştırma esnasında kullanılacak en uygun hücre sayısının tespiti de yapılmıştır. CD ön işlemi, veri kümesindeki örneklerin orijinal değerleri yerine sıra numaralarını kullandığı için CD'nin kullanılması durumunda EF ve EG ayrıklaştırma tekniklerinin sonucunun aynı veya yakın çıkması beklenmektedir. Çünkü sıra numaralarının her iki tekniğe göre de ayrıklaştırmasının sonucunda hücrelerin dağılımı birbiriyle aynı veya benzer olur. Bu nedenle, bu iki ayrıklaştırma tekniğini değerlendirirken CD ön işleminin kullanılmadığı durumlar incelenmelidir. CD'nin kullanılmadığı durumlarda EF'nin EG tekniğinden daha başarılı çıkarım performansı sergilediği gözlenmiştir. Ayrıca, ayrıklaştırma işlemi için kullanılacak en uygun hücre sayısının, örnek sayısının karekökü değerinde olduğu da

gösterilmiştir. Bilgimiz dahilinde bu değerin en uygun ne olacağını gösteren bir çalışma daha önce yapılmamıştır. Bu tez çalışmasının literatüre bu yönüyle de katkısı olmuştur.

Bu tez çalışması kapsamında seçilen kestirimciler ve türevleri gerçeklenerek bir R yazılım paketi oluşturulmuştur. PKD, SKK, PTG, STG ve PPCN yöntemleri gerçeklenirken R'ın hazır fonksiyonlarından yararlanıldığı için bu yöntemlerin çalışma süreleri R'da oldukça kısadır. Ancak kullanıcı tanımlı fonksiyonlar ile gerçeklenen diğer kestirimcilerin R'daki çalışma süreleri oldukça uzun çıkmıştır. Özellikle, veri kümesinin boyutu arttıkça çalışma sürelerinin de oldukça arttığı gözlenmiştir. R ortamında çalışma süresi uzun çıkan kestirimciler C++'da hem seri hem de paralel kodlama mantığı ile gerçeklenmiştir. C++'da gerçeklenen bu kodlar bir kütüphanede toplanmış ve R ortamından çağrılabilecek hale getirilmişlerdir. Bu yöntemlerin paralel kodları 4 çekirdekli bir bilgisayarın 3 çekirdeği kullanılarak test edilmiştir. Paralel kodlamada çekirdekler arasında veri iletişimi, verinin ana çekirdekte yeniden toparlanması, vb. işlemleri için de belli bir süre geçtiği ve belli bir yükü olduğu için, paralelleştirme işlemi her yöntem için aynı şekilde sonuçlanmamıştır. 3 çekirdekli bilgisayarda her veri kümesi için HHG, ÇYK ve KEYK yöntemlerinin paralel kodlarının, seri kodlarına nazaran 2 kat hızlı çalıştığı gözlenmiş, çekirdek sayısının daha fazla olduğu bilgisayarlarda bu yöntemler için daha fazla kara geçileceği düşünülmüştür. Bunun dışında BS, CS ve MM yöntemleri için küçük veri kümelerinde henüz paralel kodlama ile kara geçilemediği, veri kümesi büyüdükçe kara yavaş yavaş geçilmeye başlandığı gözlenmiştir. Daha büyük veri kümelerinde veya daha çok çekirdek içeren bilgisayarlarda bu yöntemler için de paralel kodlama ile süre bazında kara geçileceği düşünülmektedir.

Kestirimcilerin gerçeklendiği R kodları ile seri ve paralel C++ kodları bir R yazılım paketinde birleştirilmiş ve bu alanda çalışan araştırmacıların kullanımına sunulması hedeflenmiştir. Bilgimiz dahilinde bu kadar çok sayıda kestirimcinin birlikte gerçeklendiği ve sunulduğu bir R yazılım paketi bulunmamaktadır. Ayrıca paralel kodlama mantığı ile çalışan bu kadar çok sayıda kestirimcinin bir arada olduğu başka bir çalışma da bulunmamaktadır. Bu tez çalışması bu yönleriyle de literatüre katkıda bulunmaktadır. Sonuç olarak, bu tez çalışmasında sunulan deneyler ve çalışma sonucunda yayınlanan R yazılım paketi ile araştırmacıların GAÇ uygulamaları için rehber niteliğinde olan bu çalışmadan faydalanmaları hedeflenmiştir.

Sonraki çalışmalarda kestirimcilerin daha farklı GAÇ algoritmaları ile de çıkarım performanslarının değerlendirilmesi planlanmaktadır. Ayrıca daha büyük gerçek biyolojik veri kümeleri ile de deneylerin yapılması planlanmaktadır. Gerçek biyolojik veri kümeleri için çoğunlukla doğrulama ağları bulunmamakta ya da doğrulama ağları eksikler içermektedir. Ganet [105] isimli bir R yazılım paketi ile doğrulama ağı bulunmayan ve literatürde sık kullanılan gerçek biyolojik veri kümelerinin genel başarım analizi yapılabilmektedir. İlerdeki çalışmalarda, bu "ganet" paketi ve gerçek kanser veri kümeleri kullanılarak da kestirimcilerin değerlendirilmesi planlanmaktadır.

KAYNAKLAR

[1] Margolin, A.A., Nemenman, I, Basso, K., Wiggins, C., Stolovitzky, G., Favera, R.D. ve Califano, A., (2006). "ARACNE: An Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context", BMC Bioinformatics, 7(Suppl 1):S7.

[2] Butte, A.J. ve Kohane, L.S., (2000). "Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements", Pacific Symposium on Biocomputing, 4-9 Jan 2000, Hawaii, 5:418-429.

[3] Altay, G. ve Emmert-Streib, F., (2010). "Inferring the conservative causal core of gene regulatory networks", BMC Systems Biology, 4(132).

[4] Mattiussi, V., Tumminello, M., Iori, G. ve Mantegna, R.N., (2011). "Comparing correlation matrix estimators via Kullback–Leibler divergence", available at Social Science Research Network (SSRN), pp. 77-104.

[5] Rogers L.C.G. ve Zhou F., (2008). "Estimating correlation from high, low, opening and closing prices", Annals of Applied Probability, 18(2):813-823.

[6] Lindskog, F., (2000). "Linear Correlation Estimation". RiskLab research papers, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.71.6545, 22.10.2013.

[7] Neemuchwala, H. ve Hero, A.O., (2005). "Image registration in high dimensional feature space", Proc. of SPIE Conference on Electronic Imaging, January 2005, San Jose.

[8] Hero, A., Ma, B., Michel., O. ve Gorman, J., (2002). "Applications of entropic spanning graphs", IEEE Signal Processing Magazine, 19(5):85-95.

[9] Póczos, B., Kirshner, S. ve Szepesvári, C., (2010). "REGO: Rank-based Estimation of Renyi Information using Euclidean Graph Optimization", Journal of Machine Learning Research - Proceedings. Track 9:605-612.

[10] Brunelli, R. ve Messelodi, S., (1995). "Robust Estimation of Correlation with Applications to Computer Vision", Pattern Recognition, 28:833-841.

[11] Shan, C., Gong, S. ve Mcowan, P.W., (2005). "Conditional mutual information based boosting for facial expression Recognition". In British Machine Vision Conference (BMVC), September 2005, Oxford, U.K.

[12] Lee, W.C.Y., (1969). “An Extended Correlation Function of Two Random Variables Applied to Mobile Radio Transmission”, Bell Sys. Tech. Journal, 48:3423-3440.

[13] Barceló-Lladó, J.E., Pérez, A.M. ve Seco-Granados, G., (2012). “Enhanced Correlation Estimators for Distributed Source Coding in Large Wireless Sensor Networks”, IEEE Sensors Journal, 12(9):2799-2806.

[14] Kerscher, M., Szapudi, I. ve Szalay, A.S., (2000). “A Comparison of Estimators for the Two-Point Correlation Function”, The Astrophysical Journal, 535:L13-L16.

[15] Habib, E., Krajewski, W.F. ve Ciach, G.J., (2001). “Estimation of Rainfall Interstation Correlation”, Journal of Hydrometeorology, 2(6):21-629.

[16] Moon, Y.I., Rajagopalan, B. ve Lall, U., (1995). “Estimation of Mutual Information Using Kernel Density Estimators”, Physical Review E, 52(3):2318-2321.

[17] Olsen, C., Meyer, P.E. ve Bontempi, G., (2009). “On the Impact of Entropy Estimation on Transcriptional Regulatory Network Inference Based on Mutual Information”, EURASIP Journal on Bioinformatics and Systems Biology, 2009(308959).

[18] Simoes, R.M. ve Emmert-Streib, F., (2011). “Influence of Statistical Estimators of Mutual Information and Data Heterogeneity on the Inference of Gene Regulatory Networks”, PLoS ONE, 6(12).

[19] Daub, C.O., Steuer, R., Selbig, J. ve Kloska, S., (2004). “Estimating mutual information using B-spline functions - an improved similarity measure for analysing gene expression data”, BMC Bioinformatics, 5(118).

[20] Altay, G., Asim, M., Markowetz, F. ve Neal, D.E., (2011). “Differential C3NET reveals disease networks of direct physical interactions”, BMC Bioinformatics, 12(296).

[21] Altay, G. ve Emmert-Streib, F., (2010). “Revealing differences in gene network inference algorithms on the network-level by ensemble methods”, Bioinformatics, 26(14):1738-1744.

[22] Walters-Williams, J. ve Li, Y., (2009) “Estimation of Mutual Information: A Survey”, Lecture Notes in Computer Science Rough Sets and Knowledge Technology, 5589:389-396,.

[23] Hausser, J. ve Strimmer, K., (2009). “Entropy Inference and the James-Stein Estimator, with Application to Nonlinear Gene Association Networks”, Journal of Machine Learning Research, 10:1469-1484.

[24] Reshef, D.N., Reshef, Y.A., Finucane, H.K., Grossman, S.R., McVean, G., Turnbaugh, P.J., Lander, E.S., Mitzenmacher, M. ve Sabeti, P.C., (2011). “Detecting Novel Associations in Large Data Sets”, Science, 334(6062):1518-1524.

[25] Faith, J.J., Hayete, B., Thaden, J.T., Mogno, I., Wierzbowski, J., Cottarel, G., Kasif, S., Collins, J.J. ve Gardner, T.S., (2007). "Large-scale mapping and validation of Escherichia coli transcriptional regulation from a compendium of expression profiles", PLoS Biol, 5:e8.

[26] Meyer, P.E., Lafitte, F. ve Bontempi, G., (2008). "minet: A R/Bioconductor Package for Inferring Large Transcriptional Networks Using Mutual Information", BMC Bioinformatics, 9:461.

[27] Van den Bulcke, T., Van Leemput, K., Naudts, B., van Remortel, P., Ma, H., Verschoren, A., De Moor, B. ve Marchal, K., (2006). "SynTReN: a generator of synthetic gene expression data for design and analysis of structure learning algorithms", BMC Bioinformatics, 7(43).

[28] Paninski, L., (2003). "Estimation of Entropy and Mutual Information", Neural Computation, 15:1191-1253.

[29] Kurt, Z., Aydın, N. ve Altay, G., (2013). "Impacts of the Different Spline Orders on the B-spline Association Estimator", 13th IEEE International Conference on BioInformatics and BioEngineering (BIBE), 10-13 November 2013, Chania, Greece.

[30] Kraskov, A., Stögbauer, H. ve Grassberger, P., (2004). "Estimating mutual information", Phys. Rev. E., 83(1).

[31] Numata, J., Ebenhöh, O. ve Knapp, E.-W., (2008). "Measuring Correlations in Metabolomic Networks with Mutual Information", Genome Inform, 20:112-122.

[32] Papana, A. ve Kugiumtzis, D., (2008). "Evaluation of mutual information estimators on nonlinear dynamic systems.", Nonlinear Phenomena in Complex Systems, 11(2):225-232.

[33] de la Fuente, A., Bing, N., Hoeschele, I. ve Mendes, P., (2004). "Discovery of meaningful associations in genomic data using partial correlation coefficients.", Bioinformatics, 20:3565-3574.

[34] Çakır, T., Hendriks, M.M.W.B., Westerhuis, J.A. ve Smilde, A.K., (2009). "Metabolic network discovery through reverse engineering of metabolome data", Metabolomics, 5:318–329.

[35] [24] üzerine eleştiriler: http://comments.sciencemag.org/content/10.1126/science.1205438, 22.10.2013.

[36] Szekely, G. ve Rizzo, M., (2009). "Brownian distance covariance", The Annals of Applied Statistics, 3(4):1236–1265.

[37] Heller, R., Heller, Y. ve Gorfine, M.A., (2012). "A consistent multivariate test of association based on ranks of distances", Technical Report, June 1.

[38] Küffner, R., Petri, T., Tavakkolkhah, P., Windhager, L. ve Zimmer, R., (2012). "Inferring Gene Regulatory Networks by ANOVA", Bioinformatics, 28(10):1376-1382.

[39] DREAM project: http://www.the-dream-project.org/, 14.12.2012.

[40] Marbach, D., Costello, J.C., Küffner, R., Vega, N.M., Prill, R.J., Camacho, D.M., Allison, K.R., The DREAM5 Consortium, Kellis, M., Collins, J.J. ve Stolovitzky, G., (2012). "Wisdom of crowds for robust gene network inference", Nature Methods, 9:796-804.

[41] Faith, J.J., Driscoll, M.E., Fusaro, V.A., Cosgrove, E.J., Hayete, B., Juhn, F.S., Schneider, S.J. ve Gardner, T.S., (2008). "Many Microbe Microarrays Database: uniformly normalized Affymetrix compendia with structured experimental metadata", Nucleic Acids Res, 36(Database issue):D866–D870.

[42] Van Hulle, M.M., (2005). "Edgeworth approximation of multivariate differential entropy", Neural Computation, 17(9):1903-1910.

[43] Suzuki, T., Sugiyama, M., Sese, J. ve Kanamori, T., (2008). "Approximating Mutual Information by Maximum Likelihood Density Ratio Estimation", The Journal of Machine Learning Research (JMLR): Workshop and Conference Proceedings, 4:5-20.

[44] Suzuki, T., Sugiyama, M., Kanamori, T. ve Sese, J., (2009). "Mutual information estimation reveals global associations between stimuli and biological processes", BMC Bioinformatics, 10 (Suppl 1):S52.

[45] Altay, G., (2012). "Empirically determining the sample size for large-scale gene network inference algorithms", IET Systems Biology, 6(2):35-63.

[46] Scutari, M., (2010). "Learning Bayesian Networks with the bnlearn R Package", Journal of Statistical Software, 35(3):1-22.

[47] Lèbre, S., (2009). "Inferring dynamic genetic networks with low order independencies", Statistical Applications in Genetics and Molecular Biology, 8(1):1-39.

[48] Heckerman, D., (1995). "A tutorial on learning with Bayesian networks", Microsoft Research. Technical Report MSR-TR-95-06. Redmond, Washington.

[49] Andricioaei, I. ve Karplus, M., (2001). "On the calculation of entropy from covariance matrices of the atomic fluctuations", J. Chem. Phys. 115, 6289.

[50] Martyusheva, L.M. ve Seleznev, V.D., (2006). "Maximum entropy production principle in physics, chemistry and biology", Physics Reports, 426(1):1–45.

[51] Krug, R.R., Hunter, W.G. ve Grieger, R.A., (1976). "Enthalpy-entropy compensation. 2. Separation of the chemical from the statistical effect", J. Phys. Chem., 80(21):2341–2351.

[52] Malarvizhi, D. ve Kuppusamy, K., (2012). "A New Entropy Encoding Algorithm For Image Compression Using DCT", International Journal of Engineering Trends and Technology, 3(3).

[53] Berger, A.L., Della Pietra, S.A. ve Della Pietra, V.J., (1996). "A Maximum Entropy approach to Natural Language Processing", Journal of Computational Linguistics, 22:39-71.

[54] Guleryuz, O.G. ve da Cunha, A.L., (2006). “Image Compression with a Geometrical Entropy Coder”, 2006 IEEE International Conference on Image Processing, 8-11 Oct. 2006, Atlanta, GA, Pages: 1161 - 1164.

[55] Bandyopadhyay, O., Chanda, B. ve Bhattacharya, B.B., (2011). “Entropy-Based Automatic Segmentation of Bones in Digital X-ray Images”, Pattern Recognition and Machine Intelligence, Lecture Notes in Computer Science, 6744:122-129.

[56] Soper, H.E., Young, A.W., Cave, B.M., Lee, A. ve Pearson, K., (1917). “On the distribution of the correlation coefficient in small samples. Appendix II to the papers of "Student" and R. A. Fisher”, A co-operative study, Biometrika, 11:328-413.

[57] Stigler, S.M., (1989). “Francis Galton's Account of the Invention of Correlation”, Statistical Science, 4(2): 73-79.

[58] Cukierski, W.J., Nandy, K., Gudla, P., Meaburn, K.J., Misteli, T., Foran, D.J. vd Lockett, S.J., (2012). “Ranked Retrieval of Segmented Nuclei for Objective Assessment of Cancer Gene Repositioning”, BMC Bioinformatics, 13:232.

[59] Zhang, G. ve Su Z., (2012). “Inferences from structural comparison: flexibility, secondary structure wobble and sequence alignment optimization”, BMC Bioinformatics, 13(Suppl 15):S12.

[60] Ray, A., Lindahl, E. ve Wallner, B., (2012). “Improved model quality assessment using ProQ2”, BMC Bioinformatics, 13:224.

[61] Malovini, A., Barbarini, N., Bellazzi, R. ve De Michelis, F., (2012). “Hierarchical Naive Bayes for genetic association studies”, BMC Bioinformatics, 13(Suppl 14):S6.

[62] Qiu, P. ve Zhang, L., (2012). “Identification of markers associated with global changes in DNA methylation regulation in cancers”, BMC Bioinformatics, 13(Suppl 13):S7.

[63] Casadei, D., (2012). “Estimating the selection efficiency”, Journal of Instrumentatio, 7(8): 8021.

[64] Veitch, J., Mandel, I., Aylott, B., Farr, B., Raymond, V., Rodriguez, C., van der Sluys, M., Kalogera, V. ve Vecchio, A., (2012). “Estimating parameters of coalescing compact binaries with proposed advanced detector networks”, Physical Review D, 85(10).

[65] Mahanta, P., Ahmed, H.A., Bhattacharyya, D.K. ve Kalita, J.K., (2012). “An effective method for network module extraction from microarray data”, BMC Bioinformatics, 13(Suppl 13):S4.

[66] Lin, H.C., Goldstein, S., Mendelowitz, L., Zhou, S. Wetze, J., Schwartz, D.C. ve Pop, M., (2012). “AGORA: Assembly Guided by Optical Restriction Alignment”, BMC Bioinformatics, 13:189.

[67] Wu, C., Zhu, J. Ve Zhang, X., (2012). “Integrating gene expression and protein-protein interaction network to prioritize cancer-associated genes”, BMC Bioinformatics, 13:182.

[68] Gonnet, G.H., (2012). “Surprising results on phylogenetic tree building methods based on molecular sequences”, BMC Bioinformatics, 13:148.

[69] Collingridge, P.W. ve Kelly, S., (2012). “MergeAlign: improving multiple sequence alignment performance by dynamic reconstruction of consensus multiple sequence alignments”, BMC Bioinformatics, 13:117.

[70] Ayadi, W., Elloumi, M., ve Hao, J.-K., (2012). “Pattern-driven neighborhood search for biclustering of microarray data”, BMC Bioinformatics, 13(Suppl 7):S11.

[71] Schäfer, J. ve Strimmer, K., (2005). “An empirical Bayes approach to inferring large-scale gene association networks”, Bioinformatics, 21: 754–764.

[72] Vu, V.Q., Yu, B. ve Kass, R.E., (2007). “Coverage-adjusted entropy estimation”, Statistics in Medicine, 26(21):4039-4060.

[73] Horvath, S., (2011). Weighted Network Analysis: Applications in Genomics and Systems Biology, 1st Edition, Springer, May 4, 2011, XXII, 414 p. 54 illus., 46 in color., Hardcover ISBN: 978-1-4419-8818-8.

[74] Meyer, P.E., Kontos, K. ve Bontemp, G., (2007). “Biological Network Inference Using Redundancy Analysis”, Proceedings of the 1st international conference on Bioinformatics research and development (BIRD'07) Lecture Notes in Computer Science; 12-14 March 2007; Berlin, Germany; 4414: 16-27.

[75] Federer, A., (2011). Estimating networks using mutual information. Master Thesis, Swiss Federal Institute of Technology Zurich Department of Mathematics.

[76] Chao, A. ve Shen, T.-J., (2003). “Nonparametric estimation of Shannon’s index of diversity when there are unseen species”, Environ. Ecol. Stat, 10:429–443.

[77] Jost, L., (2007). “Partitioning Diversity Into Independent Alpha And Beta Components”, Ecology, 88(10):2427-2439.

[78] De Filippo, C., Cavalieri, D., Di Paola, M., Ramazzotti, M., Poullet, J.B., Massart, S., Collini, S., Pieraccini, G. ve Lionetti, P., (2010). “Impact of diet in shaping gut microbiota revealed by a comparative study in children from Europe and rural Africa”, Proceedings of the National Academy of Sciences of the United States of America (PNAS), 107(33): 14691-14696.

[79] Mao, C.X. ve Colwell, R.K., (2005). “Estimation of Species Richness: Mixture Models, the Role of Rare Species, and Inferential Challenges”, Ecology, 86(5): 1143-1153.

[80] Kéry, M. ve Schmid, H., (2006). “Estimating species richness: calibrating a large avian monitoring programme”, Journal of Applied Ecology, 43(1):101-110.

[81] Victor, J.D., (2006). “Approaches to Information-Theoretic Analysis of Neural Activity”, Biological Theory, 1(3): 302-316.

[82] B-spline tanımı: http://en.wikipedia.org/wiki/Non-uniform_rational_B-spline , 22.10.2013.

[83] Steuer, R., Kurths, J., Daub, C.O., Weise, J. ve Selbig, J., (2002). "The mutual information: detecting and evaluating dependencies between variables", Bioinformatics, S231-S240.

[84] Luan, Y. ve Li, H., (2003). "Clustering of time-course gene expression data using a mixed-effects model with B-splines", Bioinformatics, 19(4): 474-482.

[85] Li, H., Sun, Y. ve Zhan, M., (2007). "Analysis of Gene Coexpression by B-Spline Based CoD Estimation", EURASIP Journal on Bioinformatics and Systems Biology, 2007:49478.

[86] Chang, D.T.-H., Ou, Y.-Y., Hung, H.-G., Yang, M.-H., Chen, C.-Y. ve Oyang Y.-J., (2008). "Prediction of Protein Secondary Structures with a Novel Kernel Density Estimator", BMC Research Notes, 1:51.

[87] Taylor, C.C., Mardia, K.V., Di Marzio, M. ve Panzerab, A., (2012). "Validating protein structure using kernel density estimates", Journal of Applied Statistics, 39(11):2379-2388.

[88] Chang, D.T.-H., Wang, C.-C. ve Chen, J.-W., (2008). "Using a kernel density estimation based classifier to predict species-specific microRNA precursors", BMC Bioinformatics, 9(Suppl 12):S2.

[89] Murakami, Y. ve Mizuguchi, K., (2010). "Applying the Naïve Bayes classifier with kernel density estimation to the prediction of protein–protein interaction sites", Bioinformatics, 26(15): 1841-1848.

[90] Liao, J.G., Lin, Y., Selvanayagam, Z.E. ve Shih, W.J., (2004). "A mixture model for estimating the local false discovery rate in DNA microarray analysis", Bioinformatics, 20(16): 2694-2701.

[91] Singh, H., Misra, N., Hnizdo, V., Fedorowicz, A. ve Demchuk, E., (2003). "Nearest neighbor estimates of entropy", American J of Math and Manag Sciences, 23: 301-321.

[92] Hnizdo, V., Darian, E., Fedorowicz, A., Demchuk, E., Li, S. ve Singh, H., (2007). "Nearest-Neighbor Nonparametric Method for Estimating the Configurational Entropy of Complex Molecules", J Comput Chem, 28(3): 655-668.

[93] Kozachenko, L.F. ve Leonenko, N.N., (1987). "Sample estimates of entropy of a random vector", Problems of Information Transmission, 23:95-101.

[94] Mnatsakanov, R.M., Misra, N., Li, S. ve Harner, E.J., (2008). "K_n-nearest neighbor estimators of entropy", Mathematical Methods of Statistics, 17(3):261-277.

[95] Rahvar, A.R.A. ve Ardakani, M., (2011). "Boundary effect correction in k-nearest-neighbor estimation", Phys. Rev. E, 83(5): 051121-1 - 051121-8.

[96] Pereda, E., Quiroga, R.Q. ve Bhattacharya, J., (2005). "Nonlinear multivariate analysis of neurophysiological signals", Progress in Neurobiology, 77(1-2):1-37.

[97] Stögbauer, H., Kraskov, A., Astakhov, S.A. ve Grassberger, P., (2004). "Least-dependent-component analysis based on mutual information", Phys. Rev. E, 70(6):066123-1 - 066123-17.

[98] Rossi, F., Lendasse, A., François, D., Wertz, V., Verleysen, M., (2006). "Mutual information for the selection of relevant variables in spectrometric nonlinear modeling", Chemometrics and Intelligent Laboratory Systems, 80(2):215-226.

[99] Sugiyama, M., Kanamori, T., Suzuki, T., Hido, S., Sese, J., Takeuchi, I. ve Wang, L., (2009). "A Density-ratio Framework for Statistical Data Processing", IPSJ Transactions on Computer Vision and Applications, 1:183-208.

[100] Hido, S., Tsuboi, Y., Kashima, H., Sugiyama, M. ve Kanamori, T., (2011). "Statistical outlier detection using direct density ratio estimation", Knowledge and Information Systems, 26(2): 309-336.

[101] Sugiyama, M. ve Suzuki, T., (2011). "Least-Squares Independence Test", IEICE Transactions on Information and Systems, E94-D(6): 1333-1336.

[102] ARACNE yazılımı: http://wiki.c2b2.columbia.edu/califanolab/index.php/Software/ARACNE, 22.10.2013.

[103] R-project: http://www.r-project.org/, 14.11.2013

[104] Gama-Castro, S., Jiménez-Jacinto, V., Peralta-Gil, M., Santos-Zavaleta, A., Peñaloza-Spinola, M.I., Contreras-Moreira, B., Segura-Salazar, J., Muñiz-Rascado, L., Martínez-Flores, I., Salgado, H., Bonavides-Martínez, C., Abreu-Goodger, C., Rodríguez-Penagos, C., Miranda-Ríos, J., Morett, E., Merino, E., Huerta, A.M., Treviño-Quintanilla, L., Collado-Vides, J., (2008). "RegulonDB (version 6.0): gene regulation model of Escherichia coli K-12 beyond transcription, active (experimental) annotated promoters and Textpresso navigation", Nucl Acids Res, 36(suppl 1):D120-124.

[105] Altay, G., Altay, N. ve Neal, D., (2013). "Global assessment of network inference algorithms based on available literature of gene/protein interactions", Turkish Journal of Biology, 37:547-555.

EK-A

ÇEKİRDEK YOĞUNLUK KESTİRİMCİSİNDE BANT GENİŞLİĞİ PARAMETRESİNİN İNCELENMESİ

Bu tez çalışmasında, Çekirdek Yoğunluk Kestirimcisinde (ÇYK) kullanılan bant genişliği parametresinin en uygun değeri de incelenmiştir. Çekirdek fonksiyonu olarak Gauss fonksiyonu seçildiğinde bant genişliği parametresi olan h, Gauss fonksiyonun standart sapma parametresine karşılık gelmektedir. Bu parameter aynı zamanda yumuşatma (smoothing) parametresi olarak da bilinir ve tahmin edilecek yoğunluğun yumuşaklık miktarını belirler [A1]. [A1]'de veri kümesinin küçük bir parçası ayrılarak h parametresinin en uygun değerini belirlemek için kullanılmıştır. [A1]'deki çalışmada ARACNE yöntemini önerenler, en uygun h değerinin tespitini [A2]'deki teknik raporlarında sunmuşlardır. Burada [A3]'ten faydalanılarak parametre optimizasyonu yapıldığı görülmüştür. [A3]'te parametrenin en uygun değeri veri kümesindeki örneklere göre adaptif olarak belirlenmeye çalışılırken soncul olasılık (posterior probability) maksimize edilmeye çalışılmıştır. Bayes karar teorisine göre soncul olasılık:

$$p\left(h\middle|\overline{X},\overline{Y}\right)\approx p\left(\overline{X},\overline{Y}\middle|h\right)\times p(h) \qquad \text{(A.1)}$$

ile ifade edilir.

$p(h)$ öncül olasılığı (prior probability) ise:

$$p(h)=\frac{1}{\pi\left(1+h^2\right)} \qquad \text{(A.2)}$$

bağıntısından elde edilir.

$h=0$ olduğu noktada $p\left(\overline{X},\overline{Y}\middle|h\right)$ olabilirliği (likelihood) en büyük değerine ulaşır:

$$p(\bar{X},\bar{Y}|h) \approx \prod_{i=1}^{M} f_h(x_i, y_i) \quad \text{(A.3)}$$

Çapraz doğrulama yaklaşımı ile Bağıntı (A.3)'teki $f_h(x,y)$ ifadesi yerine $f_{h,i}(x,y)$ ifadesi kullanılır. Böylece $K_h(\cdot)$: ölçeklenmiş çekirdek fonksiyonu olmak üzere, aşağıdaki ifade elde edilir [A2]:

$$f_{h,i}(x,y) = \frac{1}{M-1} \sum_{\substack{j=1 \\ j \neq i}} K_h(x - x_j, y - y_j) \quad \text{(A.4)}$$

Gözlem verisinin standart sapması ve örnek sayısı (N) kullanılarak h parametresi hesaplanabilir [A3, A4]:

$$h = \left(\frac{1}{2} tr(\hat{\mathbf{C}}) \right)^{1/2} \times N^{-\alpha/2} \quad \text{(A.5)}$$

Buradaki $\hat{\mathbf{C}}$ gözlem very kümesinin tahmin edilen kovaryans matrisidir ve α parametresi, 0<α<0.5 koşulunu sağlamalıdır. [A1]'deki çalışmanın gerçeklenmiş kodu incelendiğinde Bağıntı (A.5)'in biraz değiştirilerek kullanıldığı görülmüştür. σ parametresi veri kümesinin standart sapması ve $\alpha = 1/3$ seçilmiştir.

$$h = \sigma \times \left(\frac{6N}{4} \right)^{-\alpha/2} \quad \text{(A.6)}$$

[A1]'de Bağıntı (A.6)'dan yola çıkılarak belli örnek sayıları için belli h değerleri elde edilerek, en uygun h değeri tahmini için dışdeğer kestirimi (extrapolation) yapılmıştır. Örnek sayısı 100'den başlamış 360'a kadar 20'şer artırılmıştır. Deneyi yapılan her bir örnek sayısı için veri kümesinden 3'er rastgele alt küme seçilmiş, bu alt kümeler için elde edilen h değerlerinin ortalaması alınmıştır. h'ın örnek sayısına göre modellenmesi esnasında Bağıntı (A.7)'de verildiği gibi bir model olduğu varsayılmıştır [A2]:

$$\hat{h} = \alpha \times (N)^{\beta} \quad \text{(A.7)}$$

Doğrusal regrasyon ile sonuç modeli Bağıntı (A.8)'deki gibi elde edilmiştir:

$$h = 0.525 \times (M)^{-0.24} \quad \text{(A.8)}$$

Bu tez çalışmasında da Bağıntı (A.8)'deki model kullanılmıştır.

Kaynaklar - Ek A

[A1] Margolin, A.A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Favera, R.D., ve Califano A., (2006). "ARACNE: An Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context", BMC Bioinformatics, 7(Suppl 1):S7.

[A2] Margolin, A.A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Favera, R.D. ve Califano A., (2006). "Technical Report: Parameter Estimation for the ARACNE Algorithm", Tech Rep nprot.; 106-S2.

[A3] Duin R.P.W., (1976). "On the choice of smoothing parameter for Parzen estimators of probability density functions", IEEE Trans. Comput, C-25:1175-1179.

[A4] Koontz W.L.G. ve Fukunaga K., (1972). "Asymptotic analysis of a non-parametric clustering technique", IEEE Trans. Comput, C-21:967-974.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Zeyneb KURT

Doğum Tarihi ve Yeri : 10.10.1983 – Antakya

Yabancı Dili : İngilizce

E-posta : zeyneb@ce.yildiz.edu.tr

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Y. Lisans	Bilgisayar Mühendisliği	Yıldız Teknik Üni.	2007
Lisans	Bilgisayar Mühendisliği	Yıldız Teknik Üni.	2005
Lise	Isparta Gazi Lisesi	Sayısal	2000-2001
	Isparta S.D. Fen Lisesi	Sayısal	1998-2000

İŞ TECRÜBESİ

Yıl	Firma/Kurum	Görevi
2005-…	Yıldız Teknik Üni., Bilgisayar Müh.	Araştırma Görevlisi

YAYINLARI

Makale

1. **(DOKTORA TEZ'inden)** Zeyneb Kurt, Nizamettin Aydın, Gökmen Altay, "Association Estimators for the Inference of Gene Networks", IEEE Journal of Biomedical and Health Informatics, **under review**.

2. **(DOKTORA TEZ'inden)** Zeyneb Kurt, Nizamettin Aydın, Gökmen Altay, "A Comprehensive Comparison of Association Estimators for Gene Network Inference Algorithms", Oxford Bioinformatics, **under review**.

3. Gökmen Altay, Zeyneb Kurt, Matthias Dehmer, and Frank Emmert Streib, "Netmes: Assessing gene network inference algorithms by ensemble network-based measures", Evolutionary Bioinformatics, **Accepted**.

Uluslar arası Bildiri

1. (DOKTORA TEZ'inden) Zeyneb Kurt, Nizamettin Aydın, Gökmen Altay, "Impacts of the Different Spline Orders on the B-spline Association Estimator", 13th IEEE International Conference on BioInformatics and BioEngineering (BIBE) 2013, 10-13 November, Chania, Greece.

2. (DOKTORA TEZ'inden) Zeyneb Kurt, Nizamettin Aydın, Gökmen Altay, "Influence of the Copula Transform on the Association Estimators for Gene Network Inference", International Conference on Applied Informatics for Health and Life Sciences (AIHLS) 2013, 9-11 September, Istanbul, Turkey.

3. Zeyneb Kurt, Sırma Yavuz, "Improvement of the Measurement Update Step of EKF-SLAM", IEEE 16th International Conference on Intelligent Engineering Systems INES 2012, Lizbon, Portekiz.

4. Zeyneb Kurt, Sırma Yavuz, "A Comparison of EKF, UKF, FastSLAM2.0, and UKF-based FastSLAM Algorithms", IEEE 16th International Conference on Intelligent Engineering Systems INES 2012, Lizbon, Portekiz.

5. Mutlu Sakarkaya, Fahrettin Yanbol, Zeyneb Kurt, "Comparison of Several Classification Algorithms for Gender Recognition from Face Images", IEEE 16th International Conference on Intelligent Engineering Systems INES 2012, Lizbon.

6. Zeyneb Kurt, Oğuzhan Yavuz, "Searching Optimal Sigma Parameter in Radial Basis Kernel Support Vector Machine for Classification of HIV Sub-Type Viruses", IEEE ICETE-

SIGMAP 2010, Atina, Yunanistan.

7. Zeyneb Kurt, H. Irem Turkmen, M. Elif Karsligil, "Linear Discriminant Analysis in Ottoman Alphabet Character Recognition", ECC 2007 (Lecture Notes in Electrical Engineering, 2009), Atina, Yunanistan.

8. Irem Turkmen, Zeyneb Kurt, M. Elif Karsligil, "Global Feature Based Female Facial Beauty Decision System", EUSIPCO 2007, Poznan, Polonya.

Ulusal Bildiri

1. Engin Semih Basmacı, Ulaş Kaymakçıoğlu, Zeyneb Kurt, "Erkek ve Kadın Yüzünü Ayırt Etmede Özellik Çıkarımı ve Özellik Seçimi Yöntemlerinin Kullanılması", SIU 2011, Antalya, Türkiye.

2. S. Burak Böke, Banu Diri, Zeyneb Kurt, "Renkli Resimlerin Kayıplı Sıkıştırılmasında Farklı Renk Uzaylarının Etkisi", SIU 2010, Diyarbakır, Türkiye.

3. Sırma Yavuz, Zeyneb Kurt, M. Serdar Biçer, "Genişletilmiş Kalman Filtresi Yöntemine Dayalı Eş Zamanlı Konum Belirleme ve Haritalama", SIU 2009, Antalya, Türkiye.

4. Zeyneb Kurt, Ethem Alpaydın, "Makine Öğrenmesine Dayalı Çeşitli Spam Filtresi Yöntemlerinin Kıyaslanması", ASYU 2008, Isparta, Türkiye.

5. Zeyneb Kurt, M. Elif Karslıgil, "Öznitelik Seçimi Yöntemlerinin Kıyaslanması", ASYU 2008, Isparta, Türkiye.

6. Zeyneb Kurt, Sırma Yavuz, "Sıralı Monte Carlo Yöntemine Dayalı Eş Zamanlı Konum Belirleme ve Haritalama Algoritması", SİU 2008, Didim, Türkiye.

7. Zeyneb Kurt, H. İrem Türkmen, M. Elif Karslıgil, "DAA Yöntemi ile Osmanlıca Karakter Tanıma", SİU 2007, Eskişehir, Türkiye.

8. Zeyneb Kurt, Aslı Uyar, H. İrem Türkmen, "Türk İşaret Diline Yönelik Yüz İfadesi ve Kafa Hareketi Tanıma Sistemi", SİU 2007, Eskişehir, Türkiye.

9. Oya Aran, İsmail Arı, Amaç Güvensan, Hakan Haberdar, Zeyneb Kurt, İrem Türkmen, Aslı Uyar, Lale Akarun, "Türk İşaret Dili Yüz İfadesi ve Baş Hareketi Veritabanı", SİU 2007, Eskişehir, Türkiye.

10. Sırma Yavuz, M. Fatih Amasyalı, Muhammet Balcılar, Gökhan Bilgin, Tarkan Dinç, Zeyneb Kurt, "Eş Zamanlı Konum Belirleme ve Harita Oluşturma Amaçlı Otonom Bir Robot", ELECO 2006, Bursa, Türkiye.

11. İrem Türkmen, Zeyneb Kurt, M.Elif Karslıgil, “Destek Vektör Makinesi Yöntemi ile Yüz Güzelliği Kararı”, SIU 2006, Antalya, Türkiye.