TUBİTAK 1001 ARAŞTIRMA PROJESİ
112E137: BIONETALIGN: Biyokimyasal Ağlarda Global Hizalamalar

BIONETALIGN: Biyokimyasal Ağlarda Global Hizalamalar

DOÇ. DR. CESİM ERTEN

ARALIK 2014
İSTANBUL

İçindekiler

Bölüm 1

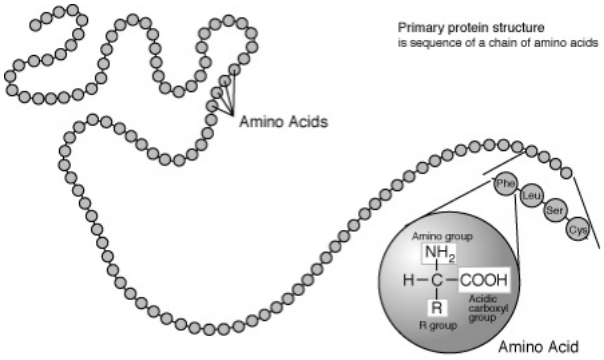
Giriş

Moleküler biyoloji ile ilgili deneysel tekniklerde yakın yıllardaki gelişmelerden dolayı, biyolojik ağlar alanında gerçekleştirilen çalışmaların önemi gün geçtikçe artmaktadır. Uzmanlaşan deneysel teknikler, niceliği üstel olarak artan güvenilirliği yüksek veri üretimine, bu da alanda oldukça faydalı yeni çıkarım ve hipotezler oluşturulmasına veya var olanlar hakkında olumlayıcı/olumsuzlayıcı kanıtlar sunulmasına yol açar. Protein çalışmaları bağlamında hem deneysel veriler, hem de son yıllarda geliştirilen hesapsal yöntemler sayesinde yoğun çalışılan türlere ait büyük biyolojik ağlar elde edilmeye başlanmıştır. Bu biyolojik ağlardan protein-protein etkileşim ağları (PPE), proteinlerarası etkileşimleri bütünsel olarak bir türün tüm proteinleri üstünde modellemeyi amaçlarken, metabolik yolaklar, daha ayrıntılı olarak belli bir fonksiyona yoğunlaşmış ve onu gerçekleştiren reaksiyonları ve reaksiyonlarda rol alan enzimler, girdi/çıktı bileşenleri arasındaki ilişkilerin tamamını modeller.

Biyolojik ağların analizi, son on yılda yoğun ilgi duyulan bir sistem biyolojisi ve biyoenformatik çalışma alanıdır. Analiz problemlerinden en önemlilerinden biri biyolojik ağların hizalanma problemidir. Hizalama basit bir anlatımla, değişik türlere ait verili biyolojik ağların içeriğindeki karşılık gelen yapıtaşlarının (PPE ağlarında karşılık gelen proteinler, protein kompleksleri, veya metabolik yolaklar için karşılık gelen reaksiyonlar vs.) çıkarsanmasına karşılık gelir. Biyolojik ağ hizalama problemi hücreiçi işleyişi anlamamız, fonksiyonu bilinmeyen proteinlerin fonksiyon çıkarımı, bilinenler için doğrulama ve türlerarası evrimsel ilişkileri keşfetmemiz açısından oldukça önemlidir. Güzel bir konuya giriş ve tetkik makalesi olarak [75]'e bakılabilir. Protein etkileşimleri çalışmalarından fonksiyonel ortoloji çıkarsanması bu bağlamda ele alınması gereken önemli bir örnektir. Her bir proteinin yerine getirdiği fonksiyonlar,

bağımsız ve izole olarak gerçekleştirilmiş işlevlerden ziyade, diğer partner proteinlerle etkileşimlerle oluşturduğu kompleks formasyonlar sayesindedir. Fonksiyonu bilinmeyen proteinlerin işlevlerinin açığa çıkarılmasında veya fonksiyonu tahmin edilenler için de bu tahminlerin doğrulanmasında protein-protein etkileşimlerinin analizi ve ağ hizalaması önemli rol oynar. Bir X proteini ile ilgili fonksiyon verisi biliniyorsa, bu veri X'in hizalandığı bir başka türe ait protein/proteinlere fonksiyon aktarımı şeklinde kullanılıp, onların da benzer fonksiyona sahip olduğu öne sürülebilir. Alanın doğası gereği çok iyi çalışılmış türler yanında görece az çalışılmış türler de oluğundan bu tarz aktarımların gerçekleştirilmesi birçok durumda gereklidir de; güvenilir sonuçlar elde edilmiş bir türün her bir proteininin, aynı konularda görece az çalışılmış veya çalışma imkanının olmadığı bir başka türdeki ortologu (hem dizisel benzerlik gösteren hem de fonksiyonel benzerliği olduğu varsayılan diğer bir türde karşılık gelen protein) bulunarak, ortolog protein için orjinal protein üzerinden hem fonksiyon aktarımı yapılabilir hem de ortolog proteinin kendi türündeki etkileşimleri orjinal proteinin kendi etkileşimlerinden çıkarsanabilir. Konuyla ilgili daha genel bilgiler edinmek için bakınız [40].

Proje genel çerçevesi içinde biyolojik ağların global hizalanması bağlamında değişik problem versiyonu tanımları yaptık ve herbir versiyon için uygun algoritmalar tasarlayıp başarım sınamalarını gerçekleştirdik. Bölüm 2'de global bire-bir ağ hizalaması, Bölüm 3'de global bire-çoklu ağ hizalaması, Bölüm 4'de kısıtlı global ağ hizalamalarının çizge-teorik incelemelerini, Bölüm 5'de global çoklu ağ hizalamalarını ve son olarak Bölüm 6'da eş zamanlı etkileşim çıkarımı ve global ağ hizalama konularını ele aldık. Bölümler 2, 5 ve 6'da önerilen problem versiyonları ve algoritmalar spesifik olarak yönsüz çizgelerden oluşan PPE ağları üzerinde uygulanmışken Bölüm 3'dekiler yönlü çizgelerle modellenen metabolik yolaklarda sınanmışlardır. Ancak bütün bölümlerde önerilen model ve yönetemlerin kolaylıkla hem yönlü hem de yönsüz çizgeler için, dolayısıyla o çizgelerle modellenen bütün biyolojik ağlar için kolaylıkla uyarlanabileceğini belirtmek gerekir. Proje çerçevesinde her bölüm çalışmalarının herbiri alanda önemli dergi ve konferanslarda yayınlanmış ve sunulmuştur. Bölüm 2, Bölüm 3 ve Bölüm 5 kapsamındaki çalışmaların herbiri *Bioinformatics* dergisinde ayrı ayrı üç yayın şeklinde basılmıştır [2, 5, 6]. Yakın zamanda yayınlanmış olmalarına rağmen bu makalelere şimdiye kadar 26 atıf verilmiş, tasarlanan algoritmalar şimdiden anılan problem versiyonları için mihenk taşı olarak kullanılmaya başlanmıştır [21]. Bölüm 4 kapsamındaki çizge-teorik çalışmalar *Discrete Applied Mathematics* dergisine gönderilmiş ve değerlendirme aşamasındadır. Yine Bölüm 6 kapsamındaki çalışmalar yayın haline getirilip *Bioinformatics* dergisine yollanmış ve revizyon sonrası kabul durumundadır. Bölüm 2 çalışmaları biyoenformatiğin önemli konferanslardan ISMB'de (ISMB/ECCB'13) tam bildiri olarak kabul edilmiş ve sunulmuştur.



Şekil 1.1: Proteinlerin temel yapısı, aminoasit zincir dizilimidir [1].

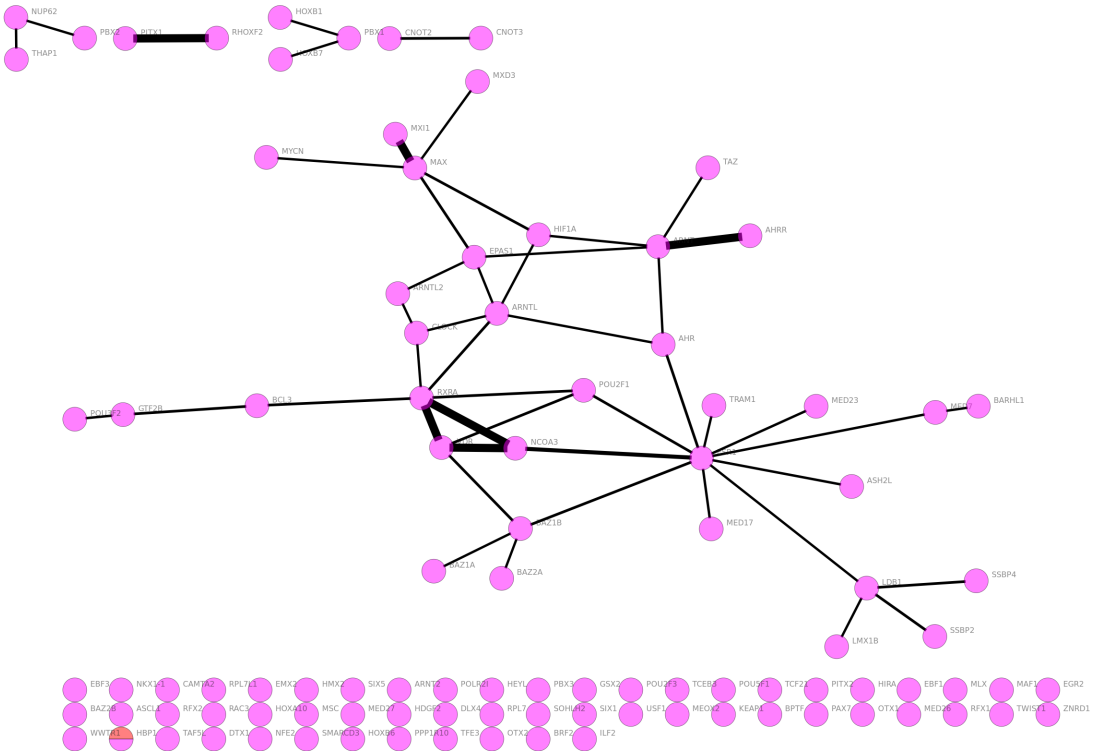
Takip eden bölümlerde biyolojik ağ hizalama bağlamında tanımlanan problem versiyonları ve her versiyon için gerçekleştirilen çalışmaların ayrıntılarına geçmeden önce, aşağıdaki altbölümlerde basit bir özet olarak biyoenformatik yapılar olarak PPE ağları, metabolik yolaklar ve gen ontolojisi hakkında genel bilgiler sunulacak ve biyolojik ağ hizalama problemine genel bir giriş yapılacaktır.

1.1 Biyoenformatik Ağ Yapıları

Biyoenformatik, hesaplama yöntemlerinin çeşitli biyolojik kaynaklardan alınan verilerin analiz edilmesi ve yorumlanması için kullanıldığı alandır. Biyoenformatik, biyoloji ve hesaplamalı bilimlerin buluştuğu disiplinlerarası bir alandır. Moleküler Biyolojide deneylerden veri yığınları üretilmektedir. Ancak bu büyük miktarlardaki veri hem gürültülüdür, hem de eksik kayıtlara sahiptir. Bu verilerle ilgili bir diğer sorun ise salt gözle yorumlamanın oldukça zor olmasıdır. Bu noktada, biyoenformatik bize gürültüyü giderme, verileri eksik bilgilere rağmen yorumlama ve yorum yapabilmek için büyük resmi görselleştirme imkanı tanımaktadır. Biyoenformatik özetle, bilgisayar bilimsel ve istatistiksel yöntemlerin moleküler biyoloji sorunları üzerine uygulanmasıdır. Bu sorunlara protein etkileşimleri, etkileşim tahminleri, iki tür arasında etkileşim ağ hizalamaları, gen ifadesi, ilaç keşfi, protein yapı hizalaması ve tahmini, dizilim hizalaması, gen bulma örnek olarak verilebilir. Aşağıdaki altbölümlerde proje çerçevesinde uygulama alanı olarak seçilen biyoenfirmatik yapılardan protein-protein etkileşim ağları ile metabolik yolaklar hakkında genel bilgiler sunulacaktır.

1.1.1 Protein-Protein Etkileşim Ağları

Protein, aminoasit dizilimlerinden oluşan büyük organik bir bileşiktir (bakınız Şekil 1.1). Amino asitlerin zincirdeki sırası proteini ve proteinin işlevini tanımlar. Çok sayıda protein bir araya gelerek veya kararlı bir bileşik oluşturarak hücre içinde ve dışında temel işlemleri gerçekleştirebilirler. Transkripsiyon, translasyon, bağlanma,



Şekil 1.2: Homo Sapiens PPE ağından küçük bir kesit.

inhibisyon, katalizasyon bu işlemlere bazı örneklerdir. Proteinler tek başına hareket etmekten ziyade diğer proteinlerle etkileşerek birlikte çeşitli biyolojik etkinliklerde bulunurlar. Protein-Protein Etkileşim (PPE) ağları (örnek için bakınız Şekil 1.2) bu etkileşimlerin çizge temsilleridir. İki boyutlu jel elektroforu (two-dimensional gel electrophoresis), çekim kromatografisi (affinity chromatography), maya iki-hibrit kalburlaması (yeast two-hybrid screening) gibi yüksek üretimli teknikleri de içeren geniş yelpazeli deneylerle verili bir tür için etkileşimler keşfedilir ve veriler proteinlerin ağ düğümlerini, etkileşimlerinse düğümler arasındaki ayrıtları simgelediği, onbinlerce düğüm ve ayrıttan oluşan büyük ağlar oluşturulur. Bu ağlarda ortalama derecenin (bir proteinin etkileştiği ortalama protein sayısı) altı ile sekiz arasında olduğu ve ağların genelde *ıskala-serbest* (scale-free) özellik gösterdiği gözlemlenmiştir [40]. Bu genel özelliklerin dışında her ağın kendine has özellikleri ve diğer biyolojik verilerle entegrasyonu gereken yapıları vardır. PPE Ağ verisinin ve bu temsili gösterimin bir eksikliği gerçekleşen etkileşimlerin koşulları ve zamanları hakkında bilgi vermemesidir. Yani bir protein 10 proteinle etkileşiyor görülebilir, bunların hepsi aynı anda da oluyor olabilir, farklı zamanlarda, farklı kombinasyonlarda da gerçekleşebilir. Bu gösterim ise sadece tüm muhtemel ikili etkileşimleri sunmaktadır ve birden fazla etkileşimin aynı anda gerçekleştiğini bize söyleyememektedir.

Proteinlerin çeşitli işlevleri vardır. Bu proteinleri üreten genlerin ürettikleri pro-

teinlerin işlevlerine göre sınıflandırılması gerekmektedir. Bu amaca yönelik çeşitli sınıflandırma sistemleri geliştirilmiştir [13, 19, 84]. Ancak her birinin kendine özgü sınıflandırma sistematiği vardır. Gene Ontology Consortium [8] tüm bu veritabanlarıyla 1998 yılında ve çok sayıda diğer veritabanıyla ilerleyen yıllarda işbirliği yaptı ve *Gene Ontology Database* adlı veritabanını üretti. Bu veritabanında bulunan *Gene Association* (Gen İlişki Verisi), hangi genin hangi işlevde protein ürettiği bilgisini vermektedir. Yine bu veritabanında bulunan *Gene Ontology (GO) Tree* ise sınıfların hiyerarşi bilgisini sunmaktadır. Zira ana kategoriler olduğu gibi bu kategorilerin altında alt kategoriler, onların da altında alt kategoriler mevcuttur. GO ağacında kategoriler çocuklarıyla listelenmiştir. Üç adet üst-seviye kategori mevcuttur: *biological process, cellular component* ve *molecular function*. Bu üç üst-seviye kategorinin altında yüksek-seviye kategoriler ve bunların altında çok sayıda alt-kategoriler mevcuttur. Bu yapı bir ağacı andırmaktadır. Ancak bir alt-kategori birden fazla kategorinin altında yer alabilir. Bu sebeple GO ağacında tekrarlar mevcut olup gerçekte yapı ağaçtan ziyade yönlü çevrimsiz çizgedir (directed acyclic graph). GO Ağacından başka, GO konsorsiyumu gen-kategori eşleşmeleri hakkında bilgi vermektedir. GO Annotation verisinde her gen, ilişkili olduğu kategoriyle birlikte bir satırda belirtilmektedir. Bir gen birden çok kategoride olabileceği gibi bir kategoride birden çok gen olabilir. Bu sebeple genler-kategoriler arasında çoka-çoklu ilişkiler vardır.

1.1.2 Metabolik Yolaklar

Metabolik ağ, metabolitler ve onların arasındaki biyokimyasal reaksiyonlardan oluşan ağlardır. Metabolitler glükoz ve aminoasit gibi küçük moleküllere karşılık gelebildiği gibi polisakkarit ve glikan gibi büyük molekülleri de ifade edebilir. Metabolitler arası reaksiyonlar genellikle proteinlerle, yani enzimlerle, katalize olur. Hücredeki sadece birkaç reaksiyon anlıktır, yani enzimatik değildir. Biyokimya alanında önemli kavramlardan *metabolik yolak*, spesifik bir metabolik fonksiyon, örneğin glikoliz, penisilin biyosentezi gibi bir fonksiyon için gerçekleşen ardıl biyokimyasal reaksiyonlar serisidir. Metabolik yolak metabolik ağın küçük bir parçası gibi düşünülebilir [46]. Tam bir metabolik ağın hücredeki materyal akışının olası bütün modlarını göstermesi gerekir. Dolayısıyla hücrenin bütün metabolik potansiyelini ve ve kapasitesini gösterir. Başka bir ifadeyle, metabolik ağ, fonksiyonel hücrenin materyal işleme merkezidir. Hücre çevreden substratları alıp özümsemek, ATP şeklinde enerji yaratmak ve büyüme ve yaşaması için gerekli materyalleri sentezlemek için bu ağlara dayanır. Biyokimyadaki temel önemi ve birçok uygulamanın doğrudan hücresel metabolizma üzerine inşası metabolik ağ ve yolakların derinlemesine araştırılmasını gerekli kılmıştır.

Çeşitli organizmaların metabolik yolaklarını sunan aralarında KEGG [47] ve BioCyc [17]'in de bulunduğu birçok çevrimiçi veritabanı vardır. Türlerarası metabo-

lik yolakların karşılaştırmalı analizleri, evrim, türleşme, filogeni yeniyapım [41, 64], ilaç hedef keşfi [38] gibi birçok önemli biyoloji, biyokimya problemlerine ışık tutar. Böylesi karşılaştırmalı analizler sadece türlerarası yolaklara özgü de değildir; analizler kanser tipli yolaklar ile sağlıklı hücre yolakları arasında da karşılaştırmalı olarak gerçekleştirilebilir ve böylesi analizler kansere spesifik metabolik özellikleri daha iyi anlamamıza yardımcı olurlar [4].

1.2 Biyolojik Ağ Hizalama

Basit ve genel ifadelerle biyolojik ağ hizalama, verili iki ya da daha çok biyolojik ağın (düğümlerin ağdaki temel yapıları, örneğin protein, metabolit vs., ayrıtların ise bu yapılar arasındaki yönlü ya da yönsüz ilişkileri temsil ettiği), düğümlerini ya da altağlarını hizalamaya (her ağdan farklı sayıda düğüm olabilecek şekilde öbeklemeye) karşılık gelir. Hizalanan düğümler ya da altağların benzer fonksiyonlu olmaları temel amaçtır. Örneğin PPE ağları karşılaştırmalı analizinin bir parçası olarak hizalamanın temel motivasyonlarından biri fonksiyonel ortolojidir; başarılı bir ağ hizalama sonucu türler boyunca aynı ya da benzer fonksiyona sahip proteinlere karar vermede bir temel oluşturabilir. Böylesi bir hizalama bigisi ayrıca türler arasında ortak ortolog yolakların çıkarsanmasında [48], ya da farklı türlerin evrimsel dinamiklerinin yeniyapımında kullanılabilir [53]. Ağ hizalaması bir model olarak kulanılmaya başlanmadan önce, PPE ağlarında ortolog protein gruplarının, diğer ağlarda ortolog yapıtaşlarının bulunması amacıyla önerilen yaygın yöntemler, çoğunlukla sadece dizisel benzerlik tipindeki evrimsel ilişkilerden oluşturulmuş verilerle sağlanmıştır. HomoloGene ve Inparanoid [70] bunlara birkaç örnektir. Ağ hizalama algoritmaları öte yandan, dizisel benzerlik bilgilerinin yanında etkleşim verilerini de hizalama oluşturma amaçlı entegre ederler. Fonksiyonel ortolog yapıların arasındaki etkileşimlerin türler boyunca korunması gerektiği varsayımına dayalı olarak, bu tarz bir entegrasyon genellikle hem hizalanan yapıların dizisel/yapısal benzerliklerini hem de hizalanmış yapı çiftleri arasındaki korunmuş etkileşim sayılarını yüksek tutmayı hedefler.

Liteartürde ağ hizalama problemine benzer başka problemler arasında etkileşim ağlarında ve yolaklarda sorgulama da sayılabilir [10, 24, 68, 79]. Genel biyolojik ağ hizalama çerçevesinde ise iki ana problem versiyonu literatürde çalışılmıştır. Bunlardan *lokal ağ hizalamada* amaç verili girdi ağlardan, hem topoloji hem de dizisel anlamda yakından eşleşen altağların çıkarsanmasıdır. Bu versiyon için literatürde önerilen yaklaşımların arasında PathBLAST [49], NetworkBLAST [76], MaWISh [51], Graemlin [30], ve de çizge eşleme-ve-ayırma algoritması [65] sayılabilir. Tipik olarak, tek bir ağdan olası olarak örtüşebilen pekçok altağ lokal hizalamaların çıktısı olarak verilir. Global ağ hizalama versiyonu ise verili girdi ağları bütünsel olarak hizalamayı

hedefler. Global hizalamalar da kendi içlerinde problem kısıtlarına göre ayrılırlar. Bazı global hizalamalar bir çift ağı hizalamaya odaklıyken [2, 5, 9, 53], bazıları herhangi sayıda ağı girdi olarak alabilir [57, 72, 80]. Global hizalamalar arasında bir diğer ayrım da eşleme tiplerine göredir. Bire-bir hizalamalarda verili bir ağdan her yapıtaşı diğer ağdan ya tek bir proteinle eşleştirilir ya da eşleşmemiş bırakılır [5, 20, 80]. Bire-çoklu ağ hizalamalarında verili bir ağdan her yapıtaşı diğer ağdan bir yapıtaşları altkümesiyle eşleştirilir [2, 9]. Son olarak da, global çoklu ağ hizalamalarında amaç, her öbeğin her ağdan herhangi sayıda yapıtaşından oluştuğu bir öbekler kümesini bulmaktır [31, 57, 72]. Bu durumda sunulan öbekler hizalamalara karşılık gelir ve herbir öbekte sunulan yapıtaşlarının fonksiyonel olarak benzer olmaları beklenir.

Bölüm 2

Global Bire-Bir Ağ Hizalamaları[1]

Global bire-bir ağ hizalamalarında temel amaç verili bir çift ağın bütünsel olarak ele alınıp ağ çiftinin düğümleri arasında birebir eşleşmeler oluşturmaktır. Söz konusu PPE ağları ise çıktı eşleştirmeden gelen protein çiftlerinin fonksiyonel olarak ortolog olması eşleştirmenin başarılı olduğunu gösterir. IsoRank [80] ile başlayarak az çok aynı formel tanımlara yoğunlaşan pek çok global bire-bir PPE ağ hizalama algoritması önerilmiştir. IsoRank lokal komşuluk hizalamalarının özdeğer formülasyonuna dayalıdır. PATH ve GA ise çifte stokastik matris kümeleri üzerinde tanımlı optimizasyon formülasyonlarının uygun esnekleştirilmeleri tabanlı yaklaşımlardır [89]. PISwap, lokal optimuma erişene kadar tekrarlı eşdeğişimleri tabanlı obur buluşsalları kullanır [20]. Öte yandan GRAAL [52] ve varyantları yöntemler MI-GRAAL [53], C-GRAAL [62], H-GRAAL [63] graphlet derece imzaları, dereceler, bölütlenme katsayıları ve BLAST E-değere dayalı dizi benzerliklerinden bir ya da daha fazlasına yönelik optimizasyon formülasyonlarına dayalı obur buluşsallar kullanır.

Ağ hizalamasında temel bir zorluk uygun bütün optimizasyon problemlerinin hesapsal zorluğudur. Bu problem onbinlerce düğüm içeren PPE ağlarının hizalanması söz konusu olduğunda daha da ağırlaşır. Bu durumda uygun bir global ağ hizalama algortimasından beklenen ölçeklenebilir olmasıdır; önerilen yöntemin hesapsal zaman gereksinimi performans ağ büyüklüğü artınca aşırı düşüş göstermemelidir. Aynı zamanda uygun optimizasyon formülasyonarının en iyi değerlerine yaklaşık skorlar üreten hizalamalar da doğal bir beklentidir. Ancak halihazırda var olan yöntemler ya ölçeklenebilirlik pahasına agresif bir şekilde skor optimizasyonuna yönelmekte ya da tersi şekilde skor performansından ödün vererek daha iyi hesapsal zaman gereksin-

[1] Bu bölümde işlenen konular [5]'de yayınlanmıştır. Ayrıntılar için [5]'e bakınız.

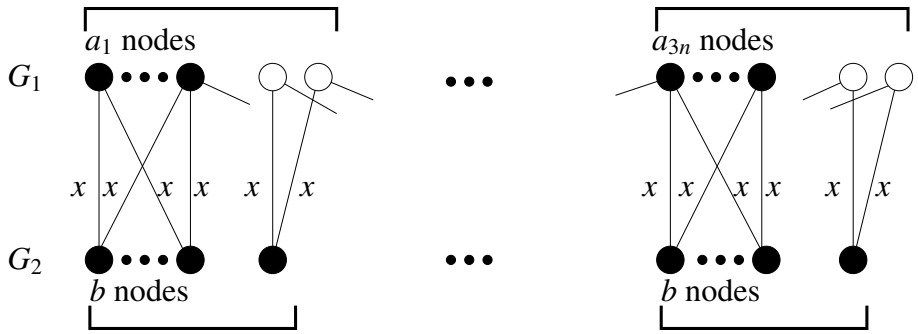
imi sunmaya odaklıdırlar. Temel olarak kaba-ayarlı eşleşme skoru tahmini ve ince-ayarlı çelişki çözümü oluşturulan, iki aşamalı özgün SPINAL algoritması bahis konusu zorluğun üstesinden gelmek için önerilmiştir. Her iki aşama da *komşuluk çift-katmanlı çizge* ve *katılımcılar* kümesi oluşturma primitiflerine dayalıdır. Önerilen algoritmanın belkemiğini bu primitiflerin tekrarlı lokal iyileştirmeler içerisinde kullanmak oluşturur. SPINAL algoritması ölçeklenebilirlik açısından rakip algoritmalardan çok daha hızlı çalışmakta ve aynı zamanda deneysel enstantanelerin hemen hepsinde onlara oranla daha doğru hizalama sonuçları üretmektedir.

Takip eden altbölümlerde önce global bire-bir PPE ağ hizalama problemi için formel bir optimizasyon problemi formülasyonu sunulacaktır. Ardından söz konusu optimizasyon probleminin hesapsal karmaşıklığı ele alınıp, en basit problem enstantanesinde bile NP-zor olduğu gösterilecektir. Üçüncü altbölümde SPINAL algoritması ayrıntılı olarak sunulacak, dördüncü altbölümde ise global bire-bir ağ hizalama literatüründe popüler algoritmalarla performans karşılaştırmaları irdelenecektir. Son olarak global bire-bir ağ hizalamalarının fonksiyon transferi ve çıkarımı amaçlı nasıl kullanılabileceği tartışılacaktır.

2.1 Problem Tanımı

Global bire-bir hizalama için literatürün bilinen hizalama algoritmalarından IsoRank, GRAAL, H-GRAAL, MI-GRAAL, GA, PATH ve PISwap algoritmalarının kullandığı problem tanımı aynıdır. $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$ verili PPE ağlarına karşılık gelen yönsüz çizgeler olsun. Burada, $1 \leq i \leq 2$ için, V_i düğümler (proteinler) kümesini, E_i ise ayrıtlar (proteinler arasi etkileşimler) kümesini ifade eder. Düğüm kümesi V_1 ve V_2'deki düğüm çiftlerinden oluşan bir hizalama çizgesi, A_{12}=(V_{12}, E_{12}) tanımlanabilir. V_{12}'deki her bir düğüm $\prec u_i, v_j \succ$ çiftine karşılık gelir öyle ki $u_i \in V_1$ ve $v_j \in V_2$. Bire-bir hizalamayı ifade etmek adına, her düğüm çifti $\prec u_i, v_j \succ \in V_{12}$ ve $\prec u'_i, v'_j \succ \in V_{12}$ için $u_i \neq u'_i$ ve $v_j \neq v'_j$ şartı sağlanmalıdır. Ayrıt kümesi E_{12} öyle tanımlanır ki, her korunmuş etkileşim, hizalama çizgesinde bir ayrıta karşılık gelsin. Yani, $\prec u_i, v_j \succ \in V_{12}$ ve $\prec u'_i, v'_j \succ \in V_{12}$ için $(\prec u_i, v_j \succ, \prec u'_i, v'_j \succ) \in E_{12}$ ancak ve ancak $(u_i, u'_i) \in E_1$ ve $(v_j, v'_j) \in E_2$.

Hizalama çizgesi tanımı açık olarak verilmese de PPE ağ hizalama çalışmalarının çoğunda ortak amaç hizalama sonucunda olabildiğince büyük bir E_{12} kümesi elde etmek ve de V_{12} içindeki eşleştirilmiş protein çiftlerinin olabildiğince yüksek dizi benzerliğine sahip olmasıdır [20, 53, 80, 89]. Formel olarak *global bire-bir PPE ağ hizalama* problemi aşağıda tanımı sunulan *GNAS* skorunu maksimize eden hizalama ağı A_{12}=(V_{12}, E_{12})'yi bulmaya karşılık gelir:



Şekil 2.1: Global bire-bir hizalama probleminin NP-zorluk ispatında kullanılan araç (*NP-hardness gadget*).

$$GNAS(A_{12}) = \alpha \times |E_{12}| + (1-\alpha) \times \sum_{\forall \prec u_i, v_j \succ} seq(u_i, v_j) \tag{2.1}$$

Topolojik benzerliğin hizalama skoruna katkısı formüldeki ilk terimle sağlanır, dizisel benzerliğin katkısı ikinci terimle ifade edilir. Sabit $\alpha \in [0,1]$ topolojik benzerlik ve dizisel benzerliğin önemlerini göreceli değiştirmek için kullanılan bir dengeleme parametresidir. İkinci terimde yer alan toplam, hizalama düğüm kümesi V_{12}'de yer alan bütün düğümler üzerinde tanımlıdır. Her $seq(u_i, v_j)$, u_i ve v_j düğümlerine karşılık gelen proteinlerin dizisel benzerliklerinin ölçüsü olarak protein çiftinin BLAST bit skorudur.

2.2 Problemin Hesapsal Kompleksitesi

$\alpha = 1$ özel durumu için bu problem, maksimum ortak ayrıt altçizge (Maximum Common Edge Subgraph, MCES) bulma problemine karşılık gelir. MCES problemi basit tanımlamayla verili iki çizgenin en fazla ayrıt içeren ortak altçizgesini (indükte altçizge zorunluğu olmadan) bulmaktır. MCES problemi NP-zordur [36]. Bu, tanımı verilen global ağ hizalama probleminin de hesapsal zorluğunu gerektirir. Bu sonuç, bazı açılardan faydalı olmakla birlikte, problemin hesapsal doğasını tam olarak anlamamıza yardımcı olmaz. Temel neden bu sonucun $\alpha = 1$ özel durum varsayımına dayanmasıdır. Global bire-bir ağ hizalama problemi doğası gereği bazı durumlarda çelişmesi olası iki farklı özelliğin eşzamanlı bir optimizasyonu problemidir; bahis konusu özel durum bu eşzamanlı optimizasyon doğasını dıştalamaktadır. Proje kapsamında elde ettiğimiz temel somut bulgulardan biri, problemin bu en genel doğasında hesapsal karmaşıklığını irdeleyen aşağıdaki teoremdir.

Teorem 2.2.1. *Global bire-bir hizalama problemi bir çift patika için NP-zordur.*

İspat. G_1 çizgesi $u_1, \ldots, u_{nb+6n}$ düğümlerinin aynı sıralı bir patikası, G_2 çizgesi de $v_1, ..., v_{nb+n}$ düğümlerinin aynı sıralı patikasıdır. Dizisel benzerlik fonksiyonu *seq*'i

bir matris olarak tanımlarız. Matrisin r ninci satırı birinci patikanın r ninci düğümü u_r'ye karşılık gelir. Benzer şekilde matrisin c ninci kolonu ikinci patikanın c ninci düğümü v_c'ye karşılık gelir. Bir pozitif tamsayı x, öyle ki $x > (nb+n-1)\alpha/(1-\alpha)$ olsun. Sayı dizisinde yapay ilk değer $a_0 = 0$ olarak tanımlansın. $1 \leq k_1 \leq 3n$, öyle ki $0 \leq t \leq k_1 - 1$ için dizideki a_t'ler toplamı $r - 2(k_1 - 1)$'den küçük ve de $0 \leq t \leq k_1$ için dizideki a_t'ler toplamı aynı değerden büyük olacak şekilde bir k_1 tamsayısı varsa birinci patika düğümü u_r'yi bir *bölüm düğümü* olarak adlandıralım. Dizisel benzerliği ifade eden matriste herbir bölüm düğümüne karşılık gelen satırda $k_2 b + k_2$ kolonunda, $1 \leq k_2 \leq n$ olmak üzere, değer 0'a diğer bütün kolonlardaki değerler x'e eşitlenir. Bölüm düğümü olmayan tüm düğümlere karşılık gelen satırlarsa tam tersi değer ataması ile oluşturulur; bakınız Figür 2.1. Figürde birinci patikanın bölüm düğümleri siyahla, diğer düğümleri ise beyazla gösterilmiştir. Birinci ve ikinci patika düğümleri arasında çizilen ayrıtlarda karşılık gelen matris değerleri ifade edilmiştir. Ayrıtı gösterilmeyen satır/sütun çifti için değer 0dır. Burda ispatın dayandığı temel fikri formel ayrıntılara inmeden özetleyeceğiz. Tam ispat, bu sonucun da yer aldığı dergi makalemizde bulunabilir; bakınız Ek1. Özet olarak, ispatın bir yönü için, eğer 3-PARTITION girdisinin tanıma uygun bir bölümlenmesi mümkünse, işaretli herbir bölüm düğümü ve bir ekstra düğüm G2'de yer alan bir $(b+1)$ lik grupla hizalanabilir. İspatın diğer yönü için de, öncelikle şu gözlem ifade edilebilir: Maksimum hizalama skoru sunacak herhangi bir hizalama, ikinci patikadan herbir düğümü mutlaka seq değeri x olan bir birinci patika düğümü ile eşleştirmelidir. Problem bu noktadan itibaren verili kısıta uyarak maksimum sayıda ortak ayrıt üretecek hizalamayı oluşturmaya dönüşür. Bu da ancak herbir $(b+1)$ lik G_2 grubundaki b tane düğümün G_1'den bütün bölüm düğümleriyle eşleşmesi ile mümkün olur, ki bu da 3-PARTITION probleminin çözümüne denk gelir. □

2.3 SPINAL Global Bire-Bir Ağ Hizalama Algoritması

Problemin NP-zorluğu, özellikle de en kolay durum olan patika hizalamasında bile hesapsal olarak zor olması, global optimumu sağlayan çözümlerdense lokal buluşsal yaklaşımların geliştirilmesine yol açmıştır. Global bire-bir ağ hizalama algoritmalarının birçoğu iki aşamalı olarak düşünülebilir. Öncelikle kaba-ayarlı eşleşme skoru tahmin aşamasında her çift $u_i \in V_1, v_j \in V_2$ için bir *tahmini güvenirlik skoru* aranır. Bu (u_i, v_j) eşleştirmesinin Denklem 2.1'de ifade edilen global skorunu maksimize eden hizalamada yer aldığına dair güvenirliği ifade eder. Bu aşamayı, genelde burda elde edilen skorları kullanarak elde edilen bir ilk hizalamayı rafine eden ikinci bir ince-ayarlı rafine aşaması takip eder. SPINAL algoritması da bu genel çerçevede diğer yaklaşımlarla ortaklaşarak yine bu iki aşamalı çerçeveyi takip eder. Ancak SPINAL'de özgün olan

kaba-ayarlı aşamada güvenirlik skor matrisi tanımı ve oluşumu ile ince-ayarlı aşamada rafine işleminin nasıl tanımlandığı ve gerçeklendiğidir. Öncelikle her iki aşamada da kullandığımız komşuluk çift-katmanlı çizgesi ($\mathcal{NBG}$, *neighborhood bipartite graph*) tanımı ve bu çizgenin maksimum ağırlıklı eşleştirmesini tanımlamamız gerekir. S her çift düğüm $u_i \in V_1, v_j \in V_2$'yi reel bir sayıya atayan bir fonksiyon olsun. u_i'nin G_1'deki komşuları $N(u_i)$ ile v_j'nin G_2'deki komşuları $N(v_j)$ ile gösterilsin. S üstünden bir düğüm çifti $\prec u_i, v_j \succ$'nin komşuluk çift-katmanlı çizgesi $\mathcal{NBG}(\{\prec u_i, v_j \succ\}, S)$ katmanlar $N(u_i)$ ve $N(v_j)$ üstünde tanımlı ayrıt-ağırlıklı bir tam çizgedir (complete bipartite graph). $\mathcal{NBG}$'de bir ayrıt (x_i, y_j)'nin ağırlığı $S(x_i, y_j)$'dir. Benzer şekilde tek bir çift yerine bir çiftler kümesinin $\mathcal{NBG}$'si de, kümeyi oluşturan çiftlerin $\mathcal{NBG}$'lerinin birleşimi olarak tanımlanır.

SPINAL algoritması Algoritma 1'de sunulmuştur. Algoritma temel olarak iki aşamadan oluşur: Kaba-ayarlı eşleşme skoru tahmin aşaması ve de ince-ayarlı çelişki çözümü ve hizalama. 3-14 arası satırlar ilk aşamayı ifade ederken kodun geriye kalanı ikinci aşamaya aittir.

2.3.1 Kaba-ayarlı Tahmini Skor Oluşturma

$P(u_i, v_j)$, u_i, V_1'den bir düğüm, v_j, V_2'den bir düğüm olmak üzere, u_i ile v_j'nin tahmini eşleşme skoru olsun. *Katılımcılar* kümesi, yani $\mathcal{NBG}(\{\prec u_i, v_j \succ\}, S)$'nin maksimum ağırlıklı eşleşmesi, C ile gösterilsin. $\mathcal{NBG}$'deki bütün ayrıtlardan sadece katılımcılar $P(u_i, v_j)$ skoruna katkıda bulunacaklardır. Böylece $P(u_i, v_j)$ şöyle tanımlanır:

$$\alpha \times \frac{\sum_{(x_i,y_j)\in C} \frac{P(x_i,y_j)}{deg_{G_1}(x_i) \times deg_{G_2}(y_j)}}{\sqrt{|C|}} + (1-\alpha) \times seq(u_i, v_j) \tag{2.2}$$

Verilen denklemde $deg_{G_1}(x_i), deg_{G_2}(y_j)$ sırasıyla x_i ve y_j'nin G_1 ve G_2'deki derecelerini gösterirken, $seq(u_i, v_j)$ ise u_i ve v_j'ye karşılık gelen proteinlerin BLAST bit skorlarını gösterir. SPINAL algoritmasında tahmini skorlar matrisi P'yi oluşturmak için, enerji minimizasyonunda yaygın kullanılan [42] basit bir tekrarlı bayır yöntemi kullanılır. Her çiftin skorunu $1-$boyutlu koordinat gibi ele alırsak, her tekrarda herbir nokta için komşuluk noktalarının (bu durumda M içinde yer alan komşuluk çiftleri) uyguladığı "çekme" kuvveti ile yeni bir koordinat belirlenir. Tekrarlar, sistem lokal bir minimum enerji seviyesine eriştiğinde, yani herbir çiftin skorunun bir önceki tekrardaki skorla aynı olduğu durumda, sonlanır; bakınız Algoritma 1'de $7-14$ satırları. Benzer iteratif yöntemlerde olduğu gibi hem lokal minimuma erişim için gerekli tekrar sayısının azlığı bakımından hem de daha kaliteli sonuçlar elde etmek için, iyi bir başlangıç konfigürasyonu ile tekrarları başlatmak önemlidir. Derece farkları ($0-1$ arasında normalize edilmiş şekliyle) ve dizisel benzerliğin, α bazlı konveks kombi-

Algorithm 1 SPINAL Global bire-bir ağ hizalama algoritması

Input: $G_1 = (V_1, E_1), G_2 = (V_2, E_2), seq, \alpha$
Output: Node set V_{12} of the global alignment network A_{12}
// Coarse-grained
for all $u_i \in V_1, v_j \in V_2$ **do**
    $P(u_i, v_j) = \alpha \times DegDiff(u_i, v_j) + (1-\alpha) \times seq(u_i, v_j)$
end for
repeat
    $P' = P$
    for all $u_i \in V_1, v_j \in V_2$ **do**
        construct $\mathcal{NBG}(\{\prec u_i, v_j \succ\}, P')$
        construct contributors set C of $\mathcal{NBG}$
        compute $P(u_i, v_j)$ as in Equation 2.2
    end for
until enough iterations
// Fine-grained
$SP =$ List of $\prec u_i, v_j \succ$ sorted w.r.t P, for $u_i \in V_1, v_j \in V_2$
repeat
    // Find new connected component in A_{12}
    pop unaligned $\prec u_i, v_j \succ$ from SP, insert into V_{12}
    repeat
        construct $\mathcal{NBG}(V_{12}, P)$
        construct contributors set C of $\mathcal{NBG}$
        swap improvements for each $\mathcal{NBG}$ edge not in C
        insert $\prec x_i, y_j \succ$ into V_{12}, for each $(x_i, y_j) \in C$
    until no contributors
until no unaligned pair in SP

nasyonu iyi bir alternatiftir. İki ağın hizalanması söz konusu olduğunda, gereken tekrar sayısı k, ağlardaki düğüm maksimum dereceleri de Δ_1, Δ_2 olmak üzere, SPINAL algoritmasının bu aşamasının zaman kompleksitesi $O(k|V_1||V_2|\Delta_1\Delta_2 \log \Delta_1\Delta_2)$ olur.

2.3.2 İnce-ayarlı Çelişki Çözümü ve Hizalama

SPINAL'in ince-ayarlı çelişki çözümü ve hizalama ile ilgili ikinci aşamasının temel fikri 'tohumla-ve-genişlet'e (seed-and-extend) dayalıdır. Bu aşama, her tekrarında hizalama çizgesi A_{12}'nin bağlı bir parçasını (connected component) bulmaya odaklı tekrarlı bir buluşsaldan oluşur; bakınız Algoritma 1 kodu, 16-26 satırlar. Her tekrar, bağlı parçanın oluşumuna, elemanları henüz eşleşmemiş ve önceki aşamada üretilen P skor matrisinde en yüksek tahmini skora sahip çifti bağlı parçada hizalayarak başlar. Yine her bağlı parçanın kendisi de tekrarlı bir oluşturma süreci boyunca büyütülerek oluşturulur. Bağlı parça nerdeyse sığ öncelikli arama (breadth first search) tarzında büyür. Herbir tekrarda, parçanın yeni katmanı (o andaki parçadan tek ayrıt erişimli)

oluşturulur. PL_1, G_1'in önceki tekrarda elde edilen katmanı olsun. Üretilen yeni katman L_1, PL_1'de eşleşmemiş düğümlerden ve de PL_1 komşuluğunda yer alan eşleşmemiş düğümlerden oluşur. Benzer tanımlar PL_2 ve L_2 için de geçerlidir. Katman kümeleri L_1 ve L_2 olan bir çift-katmanlı çizge $\mathcal{NBG}$ oluşturulur. Biri L_1'de biri L_2'de yer alan bir düğüm çifti arasına, eğer çift şimdiye kadar oluşmuş hizalamalarla bir korunmuş ayrıt üretiyorsa, $\mathcal{NBG}$'de bir ayrıt eklenir. Böyelikle $\mathcal{NBG}$'deki her ayrıt, oluşturulacak sığ öncelikli aramada hizalama çizgesi parçasına eklenecek yeni katmanda yer alabilecek olası bir düğüme karşılık gelir. Her $\mathcal{NBG}$ ayrıtının ağırlığı yine tahmini skor matrisi P'den elde edilir. $\mathcal{NBG}$ üzerinde yapılan maksimum ağırlıklı eşleştirme halihazırdaki parçaya eklenecek yeni katman için adaylardan oluşur. Son olarak aday kümesindeki her eşleşme, kümede yer almayan çelişen eşleşmelerle (en az bir düğüm kesişen) *GNAS* skoruna katkı bağlamında kazanç karşılaştırması yapılır. Bu durumda çelişen eşleşmeler daha fazla kazanç sağlarsa aday kümesindeki söz konusu eşleşme ile yer değiştirirler. Ayrıntılı bir analize girmeden, büyük ağ $|V_1|$ olmak üzere, bu aşamanın zaman kompleksitesinin $O(|V_1||V_2|\Delta_1\Delta_2 + |V_1|^2 \log |V_1|)$ olduğunu belirtelim. Ayrtınıtılı hesapsal zaman analizi için bakınız Ek1.

2.4 Karşılaştırmalı Deneysel Sonuçlar

SPINAL gerçekleştirimi C++ ve LEDA yazılım kitaplığı kullanılarak yapılmıştır. Açık kaynak kod, deneyler için faydalı Python skriptleri ve deneysel sonuçların tamamına http://code.google.com/p/spinal/ adresinden erişilebilir. Burda özet olarak sunacağımız deneysel sonuçların ayrıntıları Ek1'deki makalede bulunabilir.

Deneyler, dört tür çiftine ait PPE ve BLAST benzerlik skorları verileri kullanılarak gerçekleştirildi. Bunlar Saccharomyces cerevisiae, Drosophila melanogaster, Caenorhabditis elegans ve Homo sapiens'ten oluşur. Bütün veriler IsoBase (Park et al., 2011) veritabanından elde edilmiştir. Bu veriler aynı zamanda IsoRank ve IsoRankN gibi hizalama yöntemlerinde de kullanılmıştır. PPE ağ büyüklükleri sırasıyla şöyledir: S.cerevisiae ağında 5499 protein ve 31261 etkileşim, D.melanogaster ağında 7518 protein ve 25635 etkileşim, C.elegans ağında 2805 protein ve 4495 etkileşim ve son olarak H.sapiens ağında da 9633 protein ve 34327 etkileşim. SPINAL algoritmasını deneylerle PPE ağ hizalama literatüründe en iyilerden olarak bilinen iki algoritmayla karşılaştırdık: IsoRank ve MI-GRAAL.

Birinci tip deneyler, formel problem tanımında optimizasyon amacı olarak belirlenen GNAS skoru bağlamında elde ettiğimiz sonuçları kapsar. Her üç algoritmanın herbir tür çifti verisi için, değişik parametre ayarlarında bu fonksiyondan elde ettikleri skorlar Tablo 2.1'de sunulmuştur. MI-GRAAL algoritmasının çeşitli versiyonları söz konusudur. Bunlardan MI-GRAAL'in sunulduğu orijinal makalede en iyi olduğu be-

Tablo 2.1: GNAS değerlendirmeleri. *ce C. Elegans*'a, *dm D. Melanogaster*'e, *hs H. Sapiens*'e ve *sc S. Cerevisiae*'ye karşılık gelir. Her tür çifti için ilk sıra $|E_{12}|$'yi listelerken, ikinci sıra karşılık gelen algoritmanın hizalamasının $GNAS(A_{12})$ skorunu verir.

Veri kümesi	SPINAL			IsoRank			MI-GRAAL
	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.7$	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.7$	*(Alignment3)*
ce-dm	2343	2300	2258	335	325	328	**2390**
	717.99	1159.93	1586.87	125.22	179.70	239.49	1673.00
ce-hs	2370	2437	**2512**	299	290	293	2396
	728.26	1229.95	1764.93	116.54	163.76	215.81	1677.23
ce-sc	2326	2323	**2398**	410	385	339	2290
	709.12	1168.95	1683.13	155.14	214.65	250.52	1603.00
dm-hs	6189	6282	**6344**	823	830	829	X
	1883.22	3160.48	4451.60	334.53	475.82	615.04	X
dm-sc	5203	5311	**5360**	840	837	763	4990
	1579.06	2668.65	3759.07	312.41	461.22	559.30	3493.06
hs-sc	5703	5651	**5798**	786	817	761	X
	1731.81	2839.00	4066.22	292.00	489.21	556.05	X

lirtilen 'Alignment3' versiyonu kullanılmıştır. SPINAL ve IsoRank algoritmalarının her ikisinde de GNAS tanımında yer alan α için 0.3, 0.5 ve 0.7 değerlerinde parametre ayarı yapılmıştır. Herbir tür çiftine karşılık gelen her satırda iki değer yer alır. Bunlardan üstteki bulunan hizalamadan gelen korunmuş ayrıt sayısını, alttaki ise *GNAS* skorunu belirtir. Her deney satırı için en yüksek korunmuş ayrıt sayısı koyu renkle işaretlenmiştir. Şimdiye kadar literatürde korunmuş ayrıt bağlamında en iyi algoritma MI-GRAAL olarak bilinmekteydi. Tablodan da görüleceği üzere SPINAL algoritması hemen her durumda çok daha yüksek ayrıt koruması sağlarken, aynı zamanda en iyi *GNAS* skorlarını da elde etmeyi başarmaktadır.

İkinci tip deneylerde amaç, algoritmaların sunduğu ağ hizalamalarının biyolojik olarak ne kadar anlamlı olduğunu sınamaktır. Buna yönelik Gen Ontoloji tutarlılığı (Gene Ontology Consistency, GOC) adında bir skor tanımladık. GO konsorsiyumunun hazırladığı ve genleri ontolojik sınıflandırma amaçlı GO ağacında çeşitli GO terimleri ve bunlarla anote edilmiş genlerin listeleri vardır. Verili bir hizalamanın GOC skoru bu terimler ve anotasyonlar ışığında hesaplanır. Hizalanmış herbir ¡u,v¿ çifti için ortaklaştıkları GO terimleri sayısının, her ikisinin toplam GO anotasyonu sayısı ile normalize edilmiş değerlerinin toplamı bize GOC skorunu verir. Tablo 2.2'de her algoritmanın sunduğu hizalamaladan elde edilen GOC skoru sunulmuştur. MI-GRAAL algoritmasının sonuçları her durumda diğer iki algoritmayla karşılaştırıldığında görece kötü olduğundan bu deneyde sadece SPINAL ve IsoRank sonuçlarını karşılaştırdık. Burda belirtilmesi gereken önemli bir nokta, GO anotasyonlarının pek çok durumda sadece dizisel benzerlik verileri ışığında oluşturulduğudur; birçok durumda anotasyon verisi çıkaracak başka bilgi kaynağı henüz söz konusu değildir. Dolayısıyla hizalama probleminin tanımlı haliyle hem dizisel benzerliği hem de ortak ayrıt sayısını

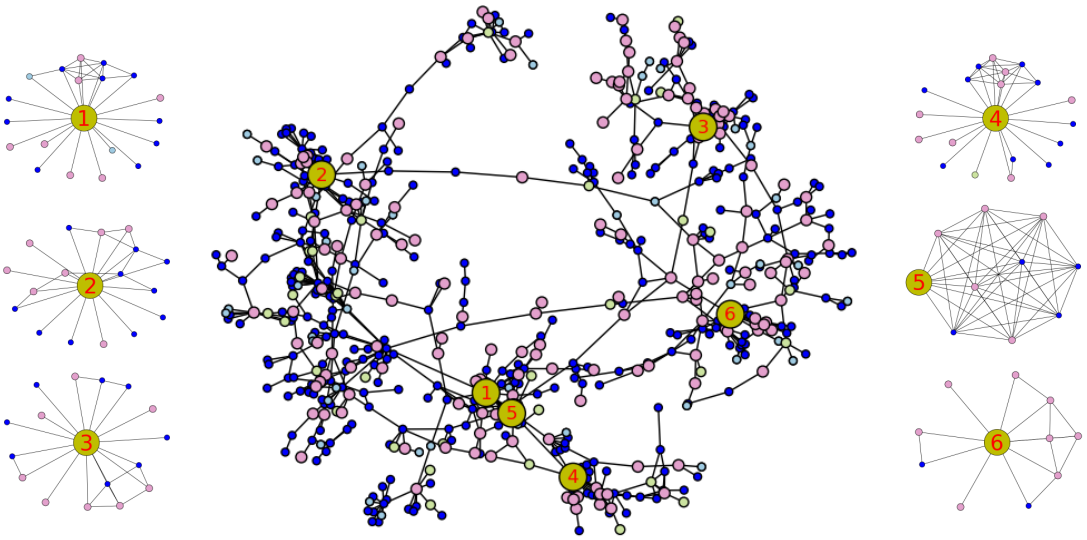
Tablo 2.2: GOC değerlendirmeleri. Satırlarda gösterilen herbir algortimanın ürettiği A_{12} ağı için soldaki kolonlar $GOC(A_{12})$ skorlarını verirken sağdakiler aynı hizalamanın $|E_{12}|$ değerlerini gösterir.

		GOC Skorları			Korunmuş Etkileşimler		
Veri kümesi	Algoritma	α=0.3	α=0.5	α=0.7	α=0.3	α=0.5	α=0.7
ce-dm	SPINAL$_I$	235.28	**231.87**	**225.99**	**575**	**611**	**655**
	IsoRank$_{HSP}$	**236.48**	229.49	222.18	484	499	468
ce-hs	SPINAL$_I$	100.83	**100.31**	**99.45**	**518**	**535**	**605**
	IsoRank$_{HSP}$	**102.18**	98.75	98.39	447	448	439
ce-sc	SPINAL$_I$	**148.53**	**149.51**	**148.75**	**810**	**815**	**809**
	IsoRank$_{HSP}$	145.89	144.92	142.59	612	596	607
dm-hs	SPINAL$_I$	**317.35**	**310.33**	**318.02**	**1546**	**1636**	**1747**
	IsoRank$_{HSP}$	304.73	299.13	289.56	1089	1107	1127
dm-sc	SPINAL$_I$	**392.41**	**389.28**	**385.42**	**1645**	**1647**	**1681**
	IsoRank$_{HSP}$	384.95	381.66	375.54	1275	1232	1188
hs-sc	SPINAL$_I$	**341.15**	**342.07**	**340.08**	**2209**	**2226**	**2262**
	IsoRank$_{HSP}$	320.44	319.13	315.33	1692	1698	1664

maksimize etmeyi amaçlayan doğasına uygun bir karşılaştırma yapabilmek için bu deneyde her iki algoritmanın da tam olarak aynı dizisel benzerlik toplamı sundukları hizalama sonuçlarını karşılaştırdık. Dolayısıyla adil bir karşılaştırma sağlamak için, tabloda yazılı parametre değerleri IsoRank algoritmasında kullanılan parametrelerken, SPINAL için belirtilen parametrede IsoRank'in sunduğu hizalamayla aynı dizisel benzerlik toplamı veren parametre ayarı kullanılmıştır. Onsekiz hizalama verisinin sadece ikisinde IsoRank SPINAL'e oranla daha iyi GOC skorları üretirken geriye kalan verinin tamamında SPINAL daha iyi sonuçlar üretmiştir. Tabloda ayrıca her durum için algoritmaların sonuç hizalamalarından elde edilen korunmuş ayrıt sayıları da gösterilmiştir. Korunmuş ayrıt sayıları bağlamında verilerin tamamında SPINAL, IsoRank'den daha iyi sonuçlar üretmiştir.

Hem korunmuş ayrıt sayısı ve optimizasyon hedefi olan GNAS skorları açısından, hem de sunulan biyolojik anlam kriterleri açısından SPINAL'in literatürde şimdiye kadar en iyi sonuçları verdikleri ifade edilen iki algoritmaya olan üstünlüğünün dışında bir diğer avantajı da gerektirdiği hesap zamanlarıdır. Algoritmaların kullandıkları CPU zamanı ile ilgili bir fikir vermesi açısından en büyük ve en yoğun ağ çifti olan H.sapiens–S.cerevisiae veri çifti uygundur. Bahsedilen ağ çifti için SPINAL, IsoRank ve MI-GRAAL algoritmalarının kullandıkları CPU zamanları sırasıyla 49, 116 ve 305 dakikadır. Bu deney Intel Core i5 2.27 GHz işlemcili 4 GB hafızalı 64-bit bir makinede gerçekleştirilmiştir.

Burda özeti sunulan deneysel sonuçların ayrıntılarına ve de 'ortalama normalize entropi', 'kapsama', spesifisite' gibi kavramlar bağlamında gerçekleştirilen diğer deneysel karşılaştırma ve tartışmalara Ek1'deki tam makaleden erişilebilir.



Şekil 2.2: H.Sapiens-S.Cerevisiae hizalama ağının en büyük bağlı parçası.

2.5 Global Bire-Bir Ağ Hizalamaları ile Fonksiyon Çıkarımı

Anote edilmemiş proteinlerin fonksiyon çıkarımı veya biyolojik yolay çıkarımı amaçlı olarak tek bir tür bağlamında PPE ağları yaygın olarak çalışılmıştır; konuyla ilgili ayrıntılıl bir tarama makalesi için bakınız [77]. Benzer bilgi çıkarımı amaçlı olarak bir başka yol da dizi benzerliği olan proteinlerin analizidir [58]. Hizalama ağları her iki tür bilgiyi de içerdiğinden onların analizinin de bu çıkarımlarda faydalı olması doğal bir beklentidir. Bu bağlamda literatürde önerilmiş ağ hizalama çalışmalarının bazıları *anotasyon transferi* yoluyla protein fonksiyon çıkarımını önermişlerdir [53, 80]. Anotasyon transferi hizalamada eşleşmiş ve anote edilmemiş bir proteine eşlemedeki diğer proteinin, anote edilmişse, fonksiyonunun aktarımına karşılık gelir. Ancak ayrıntılı bir analiz, bu tarz otomatlaştırlmıştransferlerin tek başlarına doğrudan fonksiyon çıkarım için yeterli olmadıklarını gösterir. Global ağ hizalamasından çıkarsanan sonuçları ancak ağ analiz teknikleriyle entegre edilirse daha güvenilir çıkarımlar elde edilir [75].

Konunun ayrıntılı metodolojik irdelenmesi kapsam dışında olsa da, böylesi bir entegrasyonun temellerini oluşturmak amaçlı *H.Sapiens-S.Cerevisiae* hizalama ağı irdelenecektir. Bu amaçla Tablo 2.2'de $\alpha = 0.3$ kolonunda sunulan SPINAL$_I$ hizalamasını analiz edeceğiz. Söz konusu hizalama ağında 5298 düğüm (eşleşmiş protein çifti) ve 2209 ayrıt (her iki ağda da bulunan korunmuş ayrıt) vardır. Hizalama ağının en büyük bağlı parçası (largest connected component) Şekil 2.2'de gösterilmiştir. Bu parçada 569 düğüm ve 757 ayrıt vardır. Hizalama ağının GO örtüşmeli indükte altçizgesi, yani en az bir GO anotasyonunun örtüştüğü düğümlerden oluşan altçizgede 1781 düğüm ve 433 ayrıt söz konusudur. Bu indükte altçizgenin her düğümünün Tablo 2.2'deki skora katkı sağladığını belirtelim. Şekildeki çizgede küçük düğümlerden en büyüklere

doğru, mavi düğümler eşleşmedeki proteinlerde hiç anotasyon örtüşmesi olmayanları, açık mavi düğümler tek bir GO anotasyonu örtüşmesi olanları, yeşiller iki örtüşmesi olanları ve pembeler en az üç örtüşmesi olanları gösterir. Numara etiketleri, karşılık gelen eşleşmeler, domine kategoriler sırasıyla şöyledir: 1- TBP|YER148W GO:0006355 (regulation of transcription, DNA-dependent), 2- RAN|YLR293C GO:0006810 (transport), 3- LOC392454|YBR088C GO:0003677 (DNA binding), 4- POLR2A|YDL140C GO:0006351 (transcription, DNA-dependent), 5- TAF7|YPL011C GO:0051123 (RNA polymerase II transciptional preinitiation complex assembly), 6- MCM2|YBL023C GO:0006260 (DNA replication).

Lokal PPE ağ yapılarının analiziyle *anahtar regülatör proteinlerin* çizge-teorik yaklaşımlarla belirlenmesi daha önceden önerilmiştir [33]. Ancak bu yaklaşım tek bir PPE ağı için önerilmiştir. Bu yaklaşımı genişleterek benzer bir fikirle, bir çift PPE ağıyla ilgili bilgi içeren hizalama ağında anahtar protein çiftlerinin belirlenmesi için her düğümün komşuluğunun indükte ettiği altçizgeyi çıkarırız. her anahtar çift anotasyon transferi için bir potansiyel olarak ele alınır. Şekil 2.2'de dış çeperde bulunan çizgeler potanisyel regülatör göbekleri ve onların komşuluklarının indükte altçizgeleri gösterilmiştir. Bu amaçla her hizalanmış çift $\prec u_i, v_j \succ$ için bir *domine anotasyon*, $dom(\prec u_i, v_j \succ)$ ve *domine sayısı*, $dc(\prec u_i, v_j \succ)$ hesaplanır. S_{u_i,v_j}, $\prec u_i, v_j \succ \cup \{\prec x_i, y_j \succ : (\prec x_i, y_j \succ, \prec u_i, v_j \succ) \in E_{12}\}$ üstünden indükte altçizgeyi gösteriyor olsun. Her GO anotasyonunun S_{u_i,v_j} düğümlerinde kaç defa geçtiği bulunur. Her düğümde yer alan protein çiftinden herhangi biriyle eşleşmiş bir anotasyonun bu sayıya katkıda bulunduğunu belirtmek gerekir. En yüksek sayı $dc(\prec u_i, v_j \succ)$ ve buna karşılık gelen anotasyon da $dom(\prec u_i, v_j \succ)$ ile ifade edilir. Temel amaç regülatör göbeklerini (regulating hub) çıkarsamak olduğundan bu işlemde *Hücresel Parça (Cellular Component)*'dan gelen GO anotasyonları kullanılmaz. Azalan önem sırasında bir göbekler listesi elde etmek için öncelikle, iki istisna hariç, tüm düğümler dc değerlerine göre sıralanır. Bunlardan ilkine göre eğer $\prec u_i', v_j' \succ \in S_{u_i,v_j}$ ve $dc(\prec u_i', v_j' \succ) < dc(\prec u_i, v_j \succ)$ ise $\prec u_i', v_j' \succ$ listeye eklenmez. İkinci olarak da eğer $dom(\prec u_i', v_j' \succ) = dom(\prec u_i, v_j \succ)$ and $dc(\prec u_i', v_j' \succ) < dc(\prec u_i, v_j \succ)$ ise $\prec u_i', v_j' \succ$ listeye eklenmez.

Takip eden analiz için sıralılistedeki ilk 10 düğümden beş ya da daha fazla komşusu üç ya da daha fazla GO anotasyon örtüşmesine sahip olanları ele aldık. Bu özelliğe sahip Şekil 2.2'de çeper çizgelerin merkezinde gösterilen altı düğüm söz konusudur. Bunların dominasyon sayıları ise aynı sırayla 17, 15, 14, 13, 10, ve 10'dur. Belirlenmiş kimi göbek eşleşmelerinin kendilerinin oldukça yüksek GO anotasyon örtüşmesine sahip olduklarını belirtmekte fayda vardır; TBP|YER148W eşleşmesinde 5, RAN|YLR293C eşleşmesinde 10, POLR2A|YDL140C eşleşmesinde 9 ve son olarak MCM2|YBL023C eşleşmesinde 14 örtüşme söz konusudur. Eşleşmede yer alan her proteinin domine anotasyonla aynı ya da 'benzer' (GO yönlü çevrimsiz çizgesinde, *di-*

rected acyclic graph DAG, çok uzak olmayan ortak atadan gelen) bir anotasyona sahip olması beklenir.

GO anotasyonu transferi için temel kural şu şekilde belirlenir: *Anote edilmemiş bir protein için anotasyon transferi, hizalamadaki eşi ve kendi PPE'sindeki 'yeterli' sayıda komşusu domine anotasyonla anote edilmişse gerçekleştirilir.* Bu kural ışığında, TBP|YER148W eşleşmesindeki her proteinin zaten domine anotasyonla anote edildiği görülür. RAN|YLR293C'deki proteinler ise domine anotasyon olan GO:0006810 ile anote edilmemekle beraber her ikisi de benzer bir kategori, GO:0006886 (intracellular protein transport) ile anote edilmişlerdir. Dolayısıyla her iki durumda da anotasyona transferine gerek yoktur. LOC392454|YBR088C eşleşmesinde LOC392454 hiç anotasyon içermemekteyken YBR088C eşleşmenin domine anotasyonu GO:0003677 (DNA binding) ile anotedir. Bu durumda kural gereği komşuluğa bakılır. LOC392454 proteininin H.Sapiens PPE ağında 81 protein vardır. Bunlardan 44 tanesi hiç anote edilmemişken 14 tanesi DNA binding ile anote edilmemiştir. Geriye kalanlardan 12 komşu tam olarak DNA binding ile 11 tanesi ise benzer kategorilerle (nucleic acid binding, chromatin binding, double-stranded DNA binding, damaged DNA binding) anote edilmiştir. Bu LOC392454|YBR088C eşleşmesinin doğru bir şekilde regülatör göbek olarak belirlendiğini ve LOC392454 proteini için anotasyon transferinin uygun olduğunu, yani proteinin domine anotasyon GO:0003677 (DNA binding) ile anote edilebileceğini gösterir. POLR2A|YDL140C eşleşmesinde, YDL140C, GO:0006351 (transcription, DNA-dependent) ile anotedir. POLR2Aaynı kategori ile anote değilse de benzer bir kategori, GO:0006355 (regulation of transcription, DNA-dependent) ile anote olduğundan herhangi bir transfere gerek yoktur. TAF7|YPL011C eşleşmesinde YPL011C tam da domine anotasyon ile anotedir. Hiç anotasyonu bulunmayan TAF7'ye anotasyon transferi otomat transfer yöntemleriyle mümkünken, dikkatli bir analiz ile TAF7'nin kendi PPE ağındaki 20 komşusundan sadece birinin GO:0051123 anotasyonuna sahip olduğu görülebilir. 12 tanesi benzer kategori ile anote değilken geriye kalanlar hiç anote değildir. Belirlediğimiz transfer kuralları gereği anotasyon transferi gerçekleşmez. Bu aynı zamanda literatür sonuçlarıyla da uyumludur [33]. Zira TAF7|YPL011C, [33]'de *tek parçalı göbek (single-component hub)* olarak anılan bir göbektir ve regülatör göbek olarak alınamaz; bakınız Şekil 2.2'de 5 nolu düğüm merkezli çeper altçizgesi. Son olarak, MCM2|YBL023C eşleşmesindeki proteinler domine anotasyon ile anote olduğundan herhangi bir işlem gerektirmezler.

Bölüm 3

Global Bire-Çoklu Ağ Hizalamaları[1]

Bu bölümde genel global ağ hizalama pobleminin bire-çoklu versiyonu tanımlanıp oluşan çerçevenin metabolik ağların hizalamalarına uyarlamaları tartışılacaktır. Metabolik yolaklar, önceki bölümde uygulama konusu olan PPE ağlarından farklı olarak yönlü çizgelerle (directed graph) temsil edilir. Ancak önerilen bire-çoklu hizalama çerçevesi ve buna yönelik geliştirilen ve bu bölümde tartışılacak hizalama algoritması CAMP-Ways kolaylıkla PPE ağlarına da uygulanabilir. Elde edilen metabolik yolak hizalama sonuçları, filogenetik ağaçların yeniden oluşturulmasında, ilaç tasarımında ve hücresel metabolizmaların anlaşılmasında oldukça faydalıdır. Farklı organizmalarda yapılan metabolik ağ hizalama uygulamaları sonucunda biyolojik türlerin evrimleri hakkında bilgi edinilebilir. Yapılacak eşleştirmeler sadece farklı organizmalara değil aynı zamanda aynı türden ancak farklı biyokimyasal özellikler taşıyan yolaklar üzerinde de uygulanabilir; sağlıklı ve kanserli hücreler üzerinde gerçekleştirilen metabolik yolak hizalamaları ile bu tür hastalıkların işleyişi hakkında önemli çıkarımlar yapılabilir.

Spesifik olarak metabolik yolakların hizalanmasına yönelik literatürde pekçok yöntem vardır. Bunlardan biri enzim hiyerarşileri ve enzim numara benzerliklerine dayalı hizalama yöntemidir [82]. Patika eşleştirme ve belli başlı yolakların çizge eşleştirme ile sorgulanması Yang ve Sze tarafından önerilmiştir [88]. Reaksiyonlar arası bağlılık şartı aranmaksızın yolaklarda reaksiyon kümelerinin karşılaştırılması bir başka hizalama problem versiyonu olarak sunulmuştur [22]. Heymans ve Singh ise bir enzim çizgesi yaratıp, bir çift yolağın enzimleri arasında bire-bir hizalama için maksimum ağırlıklı çift katmanlı eşleşme yöntemini geliştirmişlerdir [41]. Benzer enzim çizgesi oluşturma fikri bir diğer hizalama algortimasında da kullanılmıştır [68].

[1] Bu bölümde işlenen konular [2]'de yayınlanmıştır. Ayrıntılar için [2]'ye bakınız.

Burda farklı olarak ağaç yapısı varsayılarak, verili çift yolak üzerinde bir altağaç homomorfizm algoritması geliştirilmiştir. Tamsayı ikinci derece programlama tabanlı bir yöntem de önerilen yaklaşımlardan bir başkasıdır [90]. Üzerinde durduğumuz metabolik yolak hizalama problemi ise, Ay ve çalışma arkadaşları'nın kullanmış olduğu *reaksiyon temelli yolakların* bire-çoklu hizalanmasına dayanmaktadır [9]. Her ne kadar genel bire-çoklu hizalama modeli aynı olsa da çalışmamız bu noktadan itibaren [9]'den ayrılır. Bu bölümde sunulan çalışmanın özgünlükleri üç başlıkta özetlenebilir. İlk olarak bire-çoklu hizalamalara yönelik özgün bir *kısıtlı hizalama çerçevesi* önerilmiştir. Bu çerçeve, biyolojik ağ hizalama probleminde ilk defa uyarlanmıştır. İkinci olarak belirtilen çerçevede hizalama probleminin en basit versiyonunun bile hesapsal olarak zor olduğunu gösterilmiştir. Son olarak kısıtlı ağ hizalama çerçevesinde bire-çoklu hizalamalara yönelik özgün bir algortima geliştirilmiştir.

Takip eden altbölümlerde önce global bire-çoklu ağ hizalama problemine formel bir tanım getirelecek ardından problemde uyarlanacak kısıtlı ağ hizalama çerçevesi tanıtılacaktır. Problemin hesapsal karmaşıklığı tartışılacak ve problemin kolay durumlarda bile NP-zor olduğu gösterilecektir. Sonraki altbölümlerde bire-çoklu hizalamalar için geliştirdiğimiz CAMPWays algoritması ayrıntılandırılacak ve CAMPWays'in alternatif bir bire-çoklu hizalama yöntemi ile karşılaştırmalı başarımı sunulacaktır.

3.1 Problem Tanımı

Verili bir metabolik yolak P için, reaksiyon temelli gösterimi $G_P = (V_P, E_P)$ olarak ifade etmekteyiz. G_P yönlü bir çizge olmak üzere, çizgedeki her düğüm, bir reaksiyon r_i'ye karşılık gelmekte olup, $u_{r_i} \in V_P$ olarak ifade edilmektedir. Eğer r_i reaksiyonunun bir çıktı bileşeni, r_j reaksiyonunun bir girdi bileşeni ise, yönlü bir (u_{r_i}, u_{r_j}) ayrıtı eklenir. Eğer r_i tersinir bir reaksiyon ise, ayrıtın var olma koşulu, r_i'nin bir girdi bileşeninin, r_j'nin bir girdi bileşeni olma durumuna göre genişletilebilir ve aynı durum r_j açısından da geçerlidir. Bu durumda, eğer her iki reaksiyon da tersinir ise, bir ayrıtın var olabilmesi için dört durum meydana gelir.

Verili iki yolak gösterimi G_P, G'_P için, bire-çoklu eşleşme kısıtlaması olacak şekilde, izin verilecek eşleşmelerin tiplerinin açıklanması gerekmektedir. R_x, V_P'nin bir alt kümesini ifade etsin, öyle ki R_x içindeki düğümlerin alt çizgesi, temel yönsüz çizge içinde bağlı olsun. $\mathcal{R}_k$, k sayısına eşit ya da ondan daha küçük ve sıfırdan büyük boyuttaki tüm alt kümelerin kümesini göstersin. $\mathcal{R}'_k$ de G'_P için benzer kümeyi ifade etsin. G_P, G'_P arasında legal hizalama $\mathcal{A}$, $R_x \in \mathcal{R}_k$, $R'_x \in \mathcal{R}'_k$ olacak şekilde (R_x, R'_x) eşleşme kümesidir ve aşağıdaki koşulların sağlanması gereklidir:

1. $(R_x, R'_x) \in \mathcal{A}$ için, $|R_x| = 1$ veya $|R'_x| = 1$.
2. $(R_x, R'_x) \in \mathcal{A}$ ve $(R_y, R'_y) \in \mathcal{A}$ için, $R_x \cap R_y = \emptyset$ ve $R'_x \cap R'_y = \emptyset$.

İlk koşul, hizalama içindeki tüm eşleşmelerin bire-çoklu şekilde olacağı anlamına gelirken, ikinci koşul ise bir reaksiyonun sadece bir eşleşme içinde yer alabileceğini ifade eder. Bir hizalamanın kalitesi genelde birbirine zıt olan ölçümlerle tanımlanır: Homolojik benzerlik ve topolojik benzerlik. İlki, bir hizalama içindeki tüm eşleşmelerin homolojik skorlarının toplamı olarak ifade edilir. Verili bir (R_x, R'_x) eşleşmesinin homolojik benzerlik skoru, girdi bileşenlerinin, çıktı bileşenlerinin ve enzimlerinin benzerlikleri açısından tanımlanabilir. Bu şekildeki benzerlik skorları, incelenen moleküllerin (enzimler ya da girdi/çıktı bileşenleri) dizisel benzerlik analizlerinin sonucunda belirlenir. Bu çalışma için, Ay ve çalışma arkadaşları tarafından üretilmiş olan homolojik benzerlik skorları kullanılmaktadır [9]. Bir (R_x, R'_x) eşleşmesi için, öncelikle R_x ve R'_x in reaksiyon alt kümeleri ile ilgili olan enzimlerin birleşimleri E_x, E'_x sırasıyla üretilir. E_x, E'_x arasındaki enzimatik homolojik benzerlik, bir parçasında E_x içindeki enzimler, diğer parçasında E'_x içindeki enzimler olacak şekilde, bir ikili çizge üretilerek hesaplanır. E_x ve E'_x den her enzim çiftinin arasındaki benzerlik skoru, çift katmanlı çizgede (bipartite graph), o enzimler arasına eklenen bir ayrıta ağırlık olarak atanır. E_x, E_y arasındaki homolojik skor, üretilmiş ikili çizgede maksimum ağırlıklı çift katmanlı eşleme (maximum weight bipartite matching) yöntemi kullanılarak hesaplanır. Aynı eşleme yöntemi, girdi bileşenlerinin birleşimi I_x, I'_x ve çıktı bileşenlerinin birleşimi O_x, O'_x için de uygulanır. R_x, R'_x arasındaki homolojik benzerlik skoru, enzimler, girdi bileşenleri ve çıktı bileşenlerinin ayrı ayrı hesaplanan skorlarının konveks kombinasyonu sonucunda elde edilir. Diğer yandan, topolojik benzerlik, hizalama içinde verilmiş olan eşleşme kümesi ile ilgili olan ağ topolojisinin korunumunun ölçümü ile elde edilir. Verili bir çift eşleşme $(R_x, R'_x) \in \mathcal{A}$ ve $(R_y, R'_y) \in \mathcal{A}$ için, eğer R_x içindeki bir reaksiyondan, R_y içindeki bir reaksiyona bir ayrıt var ve R'_x içindeki bir reaksiyondan, R'_y içindeki bir reaksiyona bir ayrıt var ise, ya da bu durumun tam tersi söz konusu ise, korunmuş bir ayrıt mevcuttur, denir. Topolojik benzerlik, hizalama içinde eşleşmiş olan çiftler tarafından meydana getirilmiş korunmuş ayrıt sayısı ile doğru orantılı olan bir skor vasıtasıyla ifade edilir. Bu iki tipteki benzerlik skorları, bir kere çözümlendikten sonra, ağ hizalama problemi, bu iki skorun konveks kombinasyonunun maksimizasyonu olarak belirlenir.

3.2 Kısıtlı Hizalama Çerçevesi

Öncelikle, elde edilmiş bire-çoklu yolak hizalamaları modeli dahilinde, kısıtlı hizalama yapımız için, homolojik ve topolojik benzerliğin eş zamanlı optimizasyonu gibi bir problemden ziyade, homolojik benzerliğin üzerinde kısıtlamalar tanımlayıp, tek hedefin topolojik benzerliğin maksimizasyonu olduğu formel bir tanım sunmaktayız.

Verili bir yolak gösterimi $G_P = (V_P, E_P)$ için, ${G_p}^k$, G_p'nin k nıncı uzantısını

ifade etsin ve yönlü, ayrıt-ağırlıklı bir çizge olsun. ${G_p}^k$ içindeki her düğüm u_{R_x}, bir reaksiyon alt kümesi $R_x \in \mathcal{R}_k$'ye karşılık gelmektedir. ${G_p}^k$ içinde, $r_i \in R_x$ ve $r_j \in R_y$ olacak şekilde, eğer u_{r_i}'den u_{r_j}'ye G_p'de yönlü bir ayrıt var ise, yönlü (u_{R_x}, u_{R_y}) ayrıtı mevcuttur, denir. $w(u_{R_x}, u_{R_y})$, bu ayrıtların toplam sayısını ifade etsin. ${G'_p}^k$ de benzer şekilde ifade edilebilir. ${G_p}^k$ içindeki u_{R_x} düğümünün kısıtlarının kümesi $Cons(u_{R_x})$ olarak ifade edilir ve u_{R_x}'in ${G'_p}^k$ içinde eşleşebileceği düğümlerin alt kümesi olarak tanımlanır. Bu tanım, benzer şekilde ${G'_p}^k$ için de verilebilir. Bu tanımın simetriktir, yani $u_{R'_y} \in Cons(u_{R_x})$ ancak ve ancak $u_{R_x} \in Cons(u_{R'_y})$. k_1 ve k_2 sabit doğal sayılar olsun. Her $u_{R_x} \in {G_p}^k$ için $|Cons(u_{R_x})| \leq k_1$ ve de her $u_{R'_y} \in {G'_p}^k$ için $|Cons(u_{R'_y})| \leq k_2$ varsayalım. Tüm kısıtlamalar, bir parçası ${G_p}^k$'nin düğümlerinden, diğer parçası ${G'_p}^k$'nin düğümlerinden oluşan bir ikili *benzerlik çizgesi* ile gösterilebilir ve her kısıt bu ikili çizgede bir ayrıt ile ifade edebilir. Kısıtlı hizalama probleminin amacı, kısıtların bir alt kümesi içinden (verdiğimiz tanıma göre ikili çizgedeki ayrıtların bir altkümesi) uygun bir hizalama sağlayan ve *korunmuş ayrıt* sayısını maksimize eden bir ayrıt altkümesinin bulunmasıdır. Önerilen kısıtlı hizalama modelinin, hem yönsüz etkileşim ağları için hem de bire-bir hizalamalar için de uyarlanabileceği dikkate alınmalıdır.

Biyolojik ağ hizalamaları literatüründe Zaslavskiy ve arkadaşları protein-protein etkileşim (PPE) ağlarının global bire-bir hizalamasına uygun olan bir kısıtlı ağ hizalama tanımı geliştirmiştir [89]. Bizim önerdiğimiz model, bu sunulmuş tanıma göre daha geneldir ve Zaslavskiy ve çalışma arkadaşlarının modelini tam olarak kapsarken, onların sunduğu tanım ile ifade edilemeyecek örnekler için de çalışır. Tanımladığımız gösterimi kullanarak, her bir ağdan verilmiş u_{R_x}, u_{R_y} için, eğer $Cons(u_{R_x}) \cap Cons(u_{R_y}) \neq \emptyset$ ise, onların modeli $Cons(u_{R_x}) = Cons(u_{R_y})$ olmasını zorunlu kılar. Bu durumda, *Cons* tanımının yüksek homolojik benzerliği yansıttığı göz önünde bulundurulur, ancak bu kısıtlı bir durumdur. Çünkü yanlış üretilmiş, uzun, homolojik yönden benzer, düğüm dizileri meydana gelebilir ya da homolojik yönden benzer çiftler tamamen gözden kaçırılabilir.

3.3 Problemin Hesapsal Kompleksitesi

Fertin ve arkadaşlarının [29], $MAX(\mu_G, \mu_H)$ adıyla sundukları problem için buldukları hesapsal kompleksite sonucunu konumuz bire-çoklu hizalamaların kısıtlı çerçevesine uyarlarsak aşağıdaki hesapsal zorluk teoremini elde ederiz:

Teorem 3.3.1. *$k = k_1 = 1$ ve $k_2 = 2$ olduğu durumda kısıtlı ağ hizalama problemi APX-zordur.*

Bu teorem tanımlanan global ağ hizalama probleminin en basit durumda bile optimum hesaplanmasının ötesinde, sabit bir yaklaşımının bile zor olduğunu gösterir. Problemi daha derinlemesine anlamamız açısından bu hesapsal zorluğun hangi noktada

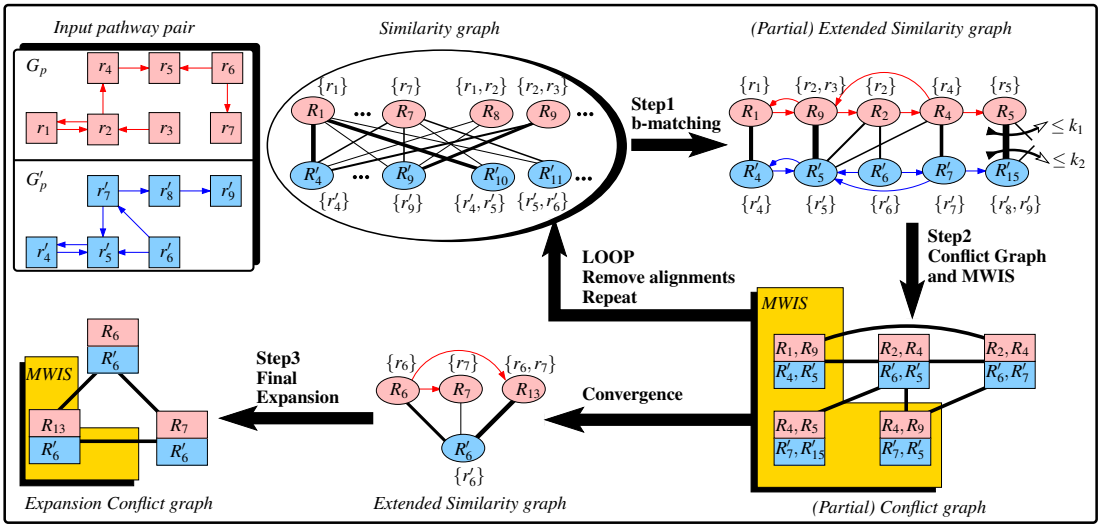
çözündüğüne bakmakta yarar vardır. Bir sonraki sonucumuz, ağlardan birinin yapısal bazı özellikler taşıdığı durumda problemin hesapsal açıdan kolaylaştığını gösterir:

Teorem 3.3.2. *$k = k1 = 1$ ve k_2 herhangi bir pozitif tamsayı olmak üzere, kısıtlı ağ hizalama problemi eğer ağlardan biri çevrim içermezse polinom zamanda çözülebilir.*

İspat. Öncelikle benzerlik çizgesi ve girdi ağlardan faydalanarak bir *koruma çizgesi* (conservation graph) oluştururuz. Bu çizgede her düğüm $c_{x'}$, herbir $u_{R'_x} \in G'_p$ düğümü için $u_{R'_x} \cup Cons(u_{R'_x})$'ye karşılık gelir. $c_{x'}, c_{y'}$ koruma çizgesinin iki düğümü olsun. Eğer, $c_{x'}$ den $c_{y'}$ ye indükte olmuş korunmuş bir ayrıt varsa, yani bazı $u_{R_w} \in Cons(u_{R'_x})$ ve $u_{R_z} \in Cons(u_{R'_y})$ için G'_p de yönlü ayrıt $(u_{R'_x}, u_{R'_y})$ ve de G_p de yönlü ayrıt (u_{R_w}, u_{R_z}) varsa, koruma çizgesinde $(c_{x'}, c_{y'})$ yönlü ayrıtı vardır. Bu yapıtaşıyla koruma çizgesi düğüm kümesi büyüklüğününün $|V'_p|$ ve ayrıt kümesi büyüklüğünün $O(|E'_p|)$ olduğunu belirtmek gerekir. Dolayısıyla koruma çizgesi boyutu problem boyutuna göre polinomiyaldir. Eğer koruma çizgesinde çervrim (cycle) varsa hem G_p ve G'_pnin her ikisinde de yönlü çizge vardır; koruma çizgesinde bir $c_{x1'}, \ldots c_{xt'}, c_{x1'}$ çevrimi ancak bazı $u_{R_{x1}} \in Cons(u_{R'_{x1}}), \ldots, u_{R_{xt}} \in Cons(u_{R'_{xt}})$ için $u_{R'_{x1}}, \ldots, u_{R'_{xt}}, u_{R'_{x1}} \in G'_p$ ve aynı zamanda $u_{R_{x1}}, \ldots, u_{R_{xt}}, u_{R_{x1}} \in G_p$ ise vardır. Verili ağlar G_p ve G'_p den en az birini çevrimsiz varsaydığımızdan o halde koruma çizgesinin de çevrimsiz olması gereklidir. $\mathcal{T}$ çevrimsiz koruma çizgesinin topolojik sıralaması (topological ordering) olsun. Düğümleri $\mathcal{T}$'deki sıraya göre gezinen ve her düğümde yer alan k_2 eşleşmenin herbirinin skorunu (korunmuş ayrıt sayısını) kendine yönelen komşuların (incoming neighbors) skorlarından hesaplayan bir dinamik programlama çözümü açıktır ki $\mathcal{T}$'nin son düğümün kendisi için, yani dış-derecesi (out-degree) 0 olan düğüm için optimum eşleşmeyi oluşturur. Bu durumda koruma çizgesinin geriye kalan düğümleri için optimum geriizleme (backtracking) ile düğümleri $\mathcal{T}$'nin tersi sırayla gezinerek oluşturulabilir. Hem ileri hem geri, her iki gezinmede de her düğümde harcanan zaman polinomiyal olduğundan algoritmanın bütünü polinomiyal zaman gerektirir. □

3.4 CAMPWays Global Bire-Çoklu Ağ Hizalama Algoritması

Önceki altbölümde Teorem 3.3.2'de sunulan çözüm polinom zamanlı da olsa, önerilen kısıtlar problemin oldukça özel bir durumuna karşılık gelir. Öte yandan Teorem 3.3.1 problemin pratikte faydalı enstantanelerinde hesapsal olarak zor olduğunu gösterir. Bu nedenle tüm durumlarda optimum sonucu sağlayamasa da, genel olarak yüksek kalitede hizalamalar sağlayan buluşsal bir algoritma tasarladık. ${G_p}^k$, ${G'_p}^k$, k_1, k_2 sabitleri, her düğüm $u_{R_x} \in {G_p}^k$ ve $u_{R'_y} \in {G'_p}^k$ için $(u_{R_x}, u_{R'_y})$ çiftinin homolojik benzerliğini verili varsayarsak, algoritmamız temel olarak 3 aşamadan oluşur. Bu ana adımlar, örnek bir yolak çifti için Şekil 3.1'de gösterilmiştir.



Şekil 3.1: $k = 2$ için örnek bir girdi üzerinde CAMPways algoritmasının temel adımları.

3.4.1 İkili Benzerlik Çizgesini Oluşturma

Bu adım, ${G_p}^k$ içindeki her u_{R_x} düğümü için, $|Cons(u_{R_x})| \leq k_1$ koşulu ve ${G'_p}^k$ içindeki her $u_{R'_y}$ düğümü için, $|Cons(u_{R'_y})| \leq k_2$ koşulu sağlanacak şekilde $Cons(u_{R_x}), Cons(u_{R'_y})$ kümelerinin bulunmasını içerir. Bir parçasında ${G_p}^k$ düğümlerinin ve diğer parçasında ${G'_p}^k$ düğümlerinin kümesi olan ve bu düğümler arasındaki ayrıtların, düğüm çiftleri arasındaki homolojik benzerliği ifade eden ağırlıklara sahip olduğu, ayrıt-ağırlıklı bir ikili çizgede, k_1, k_2 kısıtlamalarını sağlayacak ayrıt altkümesinin bulunması ve bulunan altküme için ayrıt ağırlıklarının maksimize edilmesi amaçlanır. Problem böylece, Edmonds'un çalışmalarına öncülük ettiği b-eşleşme (derece kısıtlı alt çizge problemi) problemine dönüşür [26]. Bu problem için, polinom zamanlı sonuç veren, uygun modifikasyonlu ağ akışı (network flow) algoritması [35] ve inanç yayılımı (belief propogation) yöntemleri [11] önerilmiştir. Hesapsal zaman gereksinimleri dolayısıyla biz bu adım için, basit, obur bir algoritma kullanmaktayız. Kullandığımız algoritma, her seferinde k_1, k_2 derece kısıtlarını bozmayacak şekilde, en ağır ayrıtı seçerek, onu sonuç olarak sunulacak ayrıt altkümesine ekler. Sonuç kümesine daha fazla eklenecek ayrıt kalmadığında, algoritma durur ve bu seçilen ayrıtlardan oluşan bir ikili benzerlik çizgesi, S, elde edilir.

3.4.2 Çelişki Çizgesi Üretimi ve Çelişki Çözümü

İkili benzerlik çizgesi, S, ${G_p}^k$ ve ${G'_p}^k$ nin yönlü ayrıtları ile genişletilir: Eğer ${G_p}^k$ içinde bir (u_{R_x}, u_{R_y}) ayrıtı var ise, S çizgesine yönlü (u_{R_x}, u_{R_y}) ayrıtı eklenir. Benzer ekleme işlemi ${G'_p}^k$ için de yapılır. Daha sonra, yönsüz, düğüm ağırlıklı bir *çelişki çizgesi*, C oluşturulur; C çizgesinin düğümleri, genişletilmiş çizge S içinde korunmuş ayrıt

sağlayacak 4 düğümden oluşan kümelere karşılık gelir. Açık ifadeyle, ancak ve ancak aşağıdaki koşulları sağlanırsa çelişki çizgesi içinde dörtlü $\prec u_{R_x}, u_{R_y}, u_{R'_x}, u_{R'_y} \succ$'ye karşılık gelen bir düğüm eklenir:

1. $R_x \cap R_y = \emptyset$ ve $R'_x \cap R'_y = \emptyset$.
2. $(u_{R_x}, u_{R_y}) \in {G_p}^k$, $(u_{R'_x}, u_{R'_y}) \in {G'_p}^k$ ya da $(u_{R_y}, u_{R_x}) \in {G_p}^k$, $(u_{R'_y}, u_{R'_x}) \in {G'_p}^k$.
3. $\{u_{R_x}, u_{R'_x}\}, \{u_{R_y}, u_{R'_y}\} \in S$.

Böylesi dört düğümün indükte ettiği çizgenin yönsüz versiyonu bir $4-$çevrime karşılık geldiğinden bu dörtlü için c_4 notasyonu kullanalım. Koşul *ii*'nin sadece bir tarafını karşılayan $c_4$lere 1 ağırlığı, her iki tarafını karşılayanlara 2 ağırlığı atanır. Çelişki çizgesindeki her c_4 düğümünün en azından bir korunmuş ayrıt üreten bir çift reaksiyon altkümesi elşeşmelerine karşılık geldiği açıktır. Dahası düğümün ağırlığı eşleşme çiftinin ürettiği korunmuş ayrıt sayısını verir. Şekil 3.1'de gösterilen çelişki çizgesi, figürde kısmi gösterilmiş genişletilmiş benzerlik çizgesinden türetilmiş tam çelişki çizgesidir. Dörtlü $\prec u_{R_9}, u_{R_2}, u_{R'_5}, u_{R'_6} \succ$ yapısal olarak c_4'e benzese de, yani koşullar *ii* ve *iii* geçerli olsa da, koşul *i* sağlanmadığından çelişki çizgesinde bir düğüme karşılık gelmez. Ağırlıklarla ilgili olarak da, $\prec u_{R_1}, u_{R_9}, u_{R'_4}, u_{R'_5} \succ$'in ağırlığının 2, geriye kalanların ağırlığının 1 olduğunu belirtmek gerekir.

C_1=$\prec u_{R_x}, u_{R_y}, u_{R'_x}, u_{R'_y} \succ$, C_2=$\prec u_{R_w}, u_{R_z}, u_{R'_w}, u_{R'_z} \succ$, $S_1 \in \{R_x, R_y\}, S_2 \in \{R_w, R_z\}$ ve $S'_1 \in \{R'_x, R'_y\}, S'_2 \in \{R'_w, R'_z\}$ olsun. Bir c_4 C_i için, $\mathcal{M}_{C_i}(u)$, C_i'nin diğer ağdaki komşusu u'yu göstersin. Aşağıdaki koşullardan en az bir tanesi sağlandığında, çelişki çizgesinde her iki c_4 düğümü arasına bir ayrıt eklenir:

1. $\exists S_1, S_2$ öyle ki, $S_1 \neq S_2$ ve $S_1 \cap S_2 \neq \emptyset$.
2. $\exists S'_1, S'_2$ öyle ki, $S'_1 \neq S'_2$ ve $S'_1 \cap S'_2 \neq \emptyset$.
3. $\exists S_1, S_2$ öyle ki, $S_1 = S_2$ ve $\mathcal{M}_{C_1}(S_1) \neq \mathcal{M}_{C_2}(S_2)$.
4. $\exists S'_1, S'_2$ ötle ki, $S'_1 = S'_2$ ve $\mathcal{M}_{C_1}(S'_1) \neq \mathcal{M}_{C_2}(S'_2)$.

Bu model, bir çift c_4 arasındaki bir ayrıtın, sadece, $c_4$ler tarafından gösterilen korunmuş ayrıt çiftinin, herhangi bir uygun hizalamada birlikte var olmadığı durumlarda olabileceğini ifade eder. Örnek olarak, Şekil 3.1'de verilen çelişki çizgesi için, $\prec u_{R_1}, u_{R_9}, u_{R'_4}, u_{R'_5} \succ$ ve $\prec u_{R_2}, u_{R_4}, u_{R'_6}, u_{R'_7} \succ$ c_4 leri arasındaki ayrıt koşul *i* dolayısıyladır; R_9 ve R_2 reaksiyon kümeleri bir reaksiyonu paylaşmaktadır. Bu yüzden, hiçbir uygun hizalama, bu korunmuş ayrıtların ikisini de içeremez. Öte yandan, $\prec u_{R_4}, u_{R_5}, u_{R'_7}, u_{R'_{15}} \succ$ ve $\prec u_{R_2}, u_{R_4}, u_{R'_6}, u_{R'_5} \succ$ c_4 leri arasındaki ayrıt koşul *iii* dolayısıyladır. Her iki c_4'te de, R_4 reaksiyonu, farklı reaksiyon altkümeleri ile eşleşmiştir ki bu durum, hiçbir legal

hizalama için söz konusu olamaz. Çelişki çizgesi ile ilgili verilen tanımlar şu sonuca yol açar:

Theorem 3.4.1. *C çelişki çizgesinin maksimum ağırlıklı bağımsız kümesi (maximum weight independent set, MWIS), kısıtlı hizalama problemi için optimum çözümü verir.*

Ancak, kısıtlı hizalama probleminin uygulamaları içinde, çelişki çizgesi modelimizin daha kullanışlı olabilmesi için bazı modifikasyonların yapılması gerekmektedir. İlk olarak, çelişki çizgesindeki her düğüm, kesinlikle ikili değere, bu durumda 1 ya da 2, sahip olmak zorunda değildir. Bu yüzden, çelişki çizgesi düğümlerinin ağırlıkları için uygun genellemeler yaparak, iki alternatif ağırlıklandırma modeli sunmaktayız. Benzerlik çizgesi S'de bir ayrıt e için, $w_S(e)$, e ayrıtının ağırlığını ifade etsin. $C_1 = \prec u_{R_x}, u_{R_y}, u_{R'_x}, u_{R'_y} \succ$ için ilk ağırlık modeli $\mathcal{W}_1$, $\alpha \times H(C_1) + (1-\alpha) \times I(C_1)$ ataması yapar, öyle ki:

$$H(C_1) = \frac{1}{2} \times (w_S(u_{R_x}, u_{R'_x}) + w_S(u_{R_y}, u_{R'_y}))$$

$$I(C_1) = \frac{1}{2(k^2+1)} \times \sum_{\substack{i,j \in \{u_{R_x}, u_{R_y}\}, i \neq j \\ i',j' \in \{u_{R'_x}, u_{R'_y}\}, i' \neq j'}} w(i,j) + w(i',j')$$

$I(C_1)$'in hesaplanması için, herhangi bir c_4 içindeki R_x, R_y ve R'_x, R'_y arasındaki toplam yönlü ayrıt sayısı, herhangi bir $c_4$teki olası maksimum yönlü ayrıt ${G_p}^k, {G'_p}^k$ sayısı ile normalize edilir. α, korunmuş etkileşimler ve homoloji benzerliği ağırlığı arasındaki dengeleyici parametredir. İkinci ağırlıklandırma modeli, korunmuş ayrıtları kontrol etmez; en az bir tane korunmuş ayrıt olduğu sürece, ayrıt koruma modeli aynı şekilde devam eder. Diğer yandan, organizmalar arasındaki evrimsel uzaklığa bağlı olarak, verimli bire-pekçok hizalamaları, bire-birkaç hizalamalardan ayırt etmek daha mantıklı olabilir. Bu yüzden, $\mathcal{W}_2$ olarak gösterdiğimiz ikinci model, ek olarak $\alpha_1 + \alpha_2 + \ldots + \alpha_k = 1$ olacak şekilde girdi parametreleri $\alpha_1, \alpha_2 \ldots \alpha_k$'yı alır. Her bir α_i, tüm hizalama içindeki *bire-i* lik eşleşmelere verilen önemi yansıtır. Genellik kaybı olmadan, $|R_x| >= |R'_x|$ ve $|R_y| >= |R'_y|$ olsun. Bu durumda, $C_1 = \prec u_{R_x}, u_{R_y}, u_{R'_x}, u_{R'_y} \succ$'nin ağırlığı $\alpha_{|R_x|} \times |R_x| + \alpha_{|R_y|} \times |R_y|$ olarak ifade edilir.

İkinci sorun, çelişki çözümüyle ilgilidir, yani çelişki çizgesi üzerinde MWIS'in uygulanması gerekmektedir ve bu problem genel olarak NP-tamdır [36]. Bu problem için birçok obur buluşsal Sakai ve arkadaşları tarafından önerilmiştir [73]. Biz bu yöntemlerin hepsinin gerçekleştirimini sağladık ve kapsamlı bir şekilde performanslarını sınadık. GWMIN2 yöntemi, $N_C^+(u)$, u ile u'nun C'deki komşuluğu olmak üzere, tekrarlı olarak C çelişki çizgesi içinden $\mathcal{W}(u) / \sum_{v \in N_C^+(u)} \mathcal{W}(v)$ değerini maksimize eden u düğümünü seçer. Sınamalarımızda bu yöntem diğerlerine göre daha iyi

sonuçlar vermiştir. Üstelik bu yöntem, V_C, C'nin düğüm kümesi olmak üzere, sonuç bağımsız kümenin ağırlığının en az $\sum_{u \in V_C} [\mathcal{W}(u)^2 / \sum_{v \in N_C^+(u)} \mathcal{W}(v)]$ olacağını teorik olarak garanti eder. Bu nedenlerde, MWIS çözümü için algoritmamızda GWMIN2 yöntemini kullanmayı tercih ettik.

Son olarak, algoritmanın birinci adımı sonucunda üretilen eşleşmelerin sınırlı sayıda olduğunu göz önünde bulundurarak, hizalamayı genişletmek için, 1. ve 2. adımlar çalıştırıldıktan sonra${G_p}^k$, ${G'_p}^k$'den eşleşmiş düğümlerin hepsini silip, 1. ve 2. adımı tekrar uygularız. İşlemlerin tekrarı, üretilen çelişki çizgesi C'nin boş olmasına kadar devam eder. Örneğin, Şekil 3.1'deki yolak hizalamasında, döngü sadece bir defa gerçekleşir; geriye kalan benzerlik çizgesi R_6, R_7, R_{13} ve R'_6 reaksiyon kümesini içerir ve bu da boş bir çelişki çizgesi üretir.

3.4.3 Son Hizalama Genişletmesi

İlk iki adımı içeren tekrarlı işlem, kısıtlı hizalama yapısının, korunmuş etkileşimleri maksimize etme hedefi nedeniyle $c_4$lere dayanan eşleşmeler üretir. Bu işlemler, daha fazla korunmuş etkileşim bulunamayacak duruma gelene kadar devam eder. Ancak, yine de hala yüksek homolojik benzerliğe sahip olabilecek eşleşmeler var olabilir ve bunların da hizalamaya dahil edilmesi gerekir. Böyle bir genişletmeyi uygulamak için, ilk olarak ${G_p}^k$, ${G'_p}^k$ den eşleşmiş olan tüm düğümler silinir ve tüm homolojik benzerlikleri ifade eden ayrıtlar sıfırlanır. Bu işlem sonucu elde edilen benzerlik çizgesi S'ye dayanarak yeni bir çelişki çizgesi üretilir ve bu çizge, *genişletilmiş çelişki çizgesi* olarak isimlendirilir. Bu genişletilmiş çelişki çizgesi içindeki her düğüm bir ikili grup $\prec u_{R_x}, u_{R'_x} \succ$'e karşılık gelir, öyle ki bu gruptaki düğümler arasında, S çizgesinde ayrıtlar var olsun. Aynı yolaktan gelen ve bu düğümlerle ilişkili olan reaksiyon altkümelerinin kesişimi boş küme değil ise, Şekil 3.1'de gösterildiği gibi çelişki çizgesinde bu düğümler arasına bir ayrıt eklenir. Bu noktada, ikinci adımda oluşturulan çelişki çizgesi ile bu adımda oluşturulan genişletilmiş çelişki çizgesinin tanımlarının farklı olduğu dikkate alınmalıdır. Son olarak, genişletilmiş çelişki çizgesine GWMIN2 algoritması uygulanır ve buradan gelen sonuçlar da esas hizalama sonuçlarına eklenir.

3.5 Karşılaştırmalı Deneysel Sonuçlar

CAMPways, C++ ve LEDA kütüphanesi [61] kullanılarak geliştirilmiştir. Kaynak kod, test etme ve değerlendirme için kullanışlı betikler, tüm veri ve sonuçlar, tamamlayıcı materyalin bir parçası olarak, http://code.google.com/p/campways/ adresinden erişilebilir durumdadır. Deneyler, KEGG verileri üstünde gerçekleştirilmişti [47]. Bu altbölümde, SubMAP [9] ile CAMPWays algoritmalarının karşılaştırmalı performans değerlendirmelerini sunacağız. SubMAP'in seçilmiş olmasının amacı, bu aracın da

konumuz olan global bire-çoklu ağ hizalamaları elde etmeyi amaçlamış olmasıdır.

KEGG veritabanı, Glycerolipid metabolizması, Tryptophan metabolizması gibi detaylı metabolizma kategorilerinin yolaklarını sağlamasına rağmen, bu yolakların doğrudan ağ hizalama çalışmasında kullanılması, ortaya yeterli bilgi çıkarmamaktadır. Bunun en önemli nedeni, hizalama kalitesinin tarafsız değerlendirme temeli olacak altın standart eksikliğidir. Daha az önemli olmasına rağmen, küçük yolak boyutları da bir başka problemi teşkil eder. Hizalama yönteminin davranışını sınamak, bu ölçekte güvenilir sonuçlar üretmeyebilir. Bu iki sorunu da giderecek mekanizma, KEGG veritabanındaki, aynı detaylı metabolizma kategorilerinde bulunan tüm yolakları birleştirmek ve böylece daha genel metabolizma kategorileri elde etmektir. Bunun için, listelenmiş, yüksek düzeyli ilk 11 kategori için, her biri içinde bulunan belirli yolakları birleştirilerek, daha geniş metabolik ağlar elde ettik. Böylece toplamda 11 tane geniş metabolik ağ elde ettik: 1.1 Karbonhidrat metabolizması, 1.2 Enerji metabolizması, 1.3 Yağ metabolizması, 1.4 Nükleotid metabolizması, 1.5 Amino asit metabolizması, 1.6 Diğer amino asitlerin metabolizması, 1.7 Glikan biyosentezi ve metabolizması, 1.8 Kofaktörlerin ve vitaminlerin metabolizması, 1.9 Terpenoid vePolyketide metabolizması , 1.10 Diğer ikincil metabolitlerin biyosentezi, 1.11 Xenobioticlerin biyolojik yıkımı ve metabolizmaları. Her geniş metabolik ağ içinde, sayısı 2 ile 15 arasında değişen yolaklar bulunur. Bu bölümde açıklanan deneysel hesaplamaların tamamı, bu geniş metabolik ağların, farklı türlerden alınan çiftlerine aittir.

Aşağıda sunulan iki alt bölüm, CAMPways ve SUBMAP sonuçlarının doğruluk açısından karşılaştırılmalarını içerir. Bu amaçla iki doğruluk tanımı kullanılmıştır. İlki, çıktı hizalamalarının tersine mühendislik başarılarına dayanırken, ikincisi ise, KEGG'de bulunan fonksiyonel grup çevrimi kategorileri kullanılarak, biyokimyasal açıdan önemli sonuçların alınıp alınamadığı ve bunların tutarlılığı açısından kullanılır. Değerlendirmeler CAMPways ve SubMAP için gözlemlenmiş olan çalışma hızlarının karşılaştırılmasıyla son bulur.

3.5.1 Metabolik Yolaklarda Tersine Mühendislik Sınamaları

Üzerinde çalıştığımız metabolik ağlar, ayrıntılı metabolizma kategorileri üzerindeki küçük yolakların sınama amaçlı birleştirilmeleriyle oluşturulmuştur. Bu durumda, doğruluk ölçümü, sonuç olarak verilmiş çıktı hizalamasının tersine mühendislik kapasitesidir; bir hizalamadaki eşleşmiş reaksiyonlar, orijinal KEGG yolaklarında, aynı yolağa ait ise, bu durumda yüksek kaliteli hizalamanın olduğu varsayılır. Böylece, ayrıntılı metabolizma kategorilerindeki yolaklar, bizim için altın standarda karşılık gelirler. Burada, KEGG veritabanındaki tüm yolakların, hiçbir eksik veri içermeksizin ve yanlış yolak ilişkileri olmaksızın, tamamen doğru olduğu kabul edilmiştir. X, X' 2 ayrı türü ve $G_X, G_X{}'$ de bu türlerin, yukarıdaki bölümde açıklanmış olan metabolik

Tablo 3.1: Aynı üst alem verilerinde tersine mühendislik deneyleri.

TR	Kapsam			Doğru Eşleşmeler			Oran		
	S	C1	C2	S	C1	C2	S	C1	C2
437	-	*435*	*435*	-	*211*	*213*	-	*0.99*	*0.98*
458	-	416	416	-	166	171	-	0.82	0.83
62	*62*	*62*	*62*	*29*	*31*	*31*	*0.96*	*1*	*1*
116	105	110	110	45	51	51	0.93	0.94	0.94
745	-	*726*	*726*	-	*361*	*361*	-	*0.99*	*0.99*
264	244	254	254	96	105	103	0.82	0.82	0.83
320	-	*320*	*320*	-	*159*	*159*	-	*0.99*	*0.99*
296	280	262	262	110	128	128	0.90	0.98	0.98
496	*491*	*481*	*481*	*221*	*239*	*239*	*0.96*	*0.99*	*0.99*
369	352	340	339	122	143	143	0.79	0.86	0.86
134	*128*	*130*	*130*	*59*	*64*	*64*	*0.96*	*0.98*	*0.98*
108	102	97	97	37	39	39	0.78	0.82	0.82
168	*148*	*168*	*168*	*73*	*76*	*76*	*1*	*0.90*	*0.90*
73	69	64	64	31	31	31	0.96	0.96	0.96
307	-	*306*	*307*	-	*150*	*151*	-	*0.98*	*0.98*
334	325	324	326	129	143	144	0.87	0.89	0.90
31	*28*	*28*	*28*	*12*	*14*	*14*	*1*	*1*	*1*
51	43	43	44	15	17	17	0.78	0.80	0.77
35	*34*	*34*	*34*	*16*	*17*	*17*	*1*	*1*	*1*
23	21	20	20	8	9	9	0.8	0.9	0.9
207	*201*	*200*	*200*	*87*	*100*	*100*	*0.92*	*1*	*1*
175	153	134	134	53	60	60	0.81	0.89	0.89

ağlarından biri, 1.m, olsun. R_x' in X'e ait bir reaksiyon altkümesi ve R'_x'in de X' e ait bir reaksiyon altkümesi olduğu durumda, $\prec u_{R_x}, u_{R'_x} \succ$, $G_X, G_X{}'$'in hizalanmasından elde edilmiş bir eşleşme olsun. Genelleme kaybolmaksızın, $R_x = \{r_x\}$, yani bire-çok eşleşme içindeki bu altküme, sadece bir tane reaksiyon içeriyor olsun. $P_1 \ldots P_x$, X türünün 1.m metabolizması ile ilgili olan ve r_x reaksiyonunu içeren yolaklar olsun. Bu durumda, bir eşleşmenin doğru sayılabilmesi için, R'_x altkümesindeki her reaksiyon, $P_i{}'$, X' türünün 1.m metabolizmasındaki bir yolağı gösterecek şekilde, $P_1{}', \ldots P_x{}'$ yolaklarının en az birinde olmak zorunda iken, aynı durum X türünün P_i reaksiyonu için de geçerlidir. Deneysel değerlendirmelerimizi ikiye ayırırız: aynı üst aleme ait türler arasındaki hizalamalar ve farklı üst alemlere ait olan türler arasındaki hizalamalar. Ökaryot üst aleminden temsili olarak Homo sapiens (hsa) ve Mus musculus (mmu) türleri, bakterilerden de Escherichia coli (eco) ve Agrobacterium tumefaciens (atc) türleri seçilmiştir. Parametrelerden k değeri 3'e sabitlenmiştir, yani, bir ağdaki her reaksiyonun, diğer ağdan en fazla 3 reaksiyon ile eşleşebileceği garantilenmiştir. CAMPways hizalamaları için k_1 ve k_2 değerleri de 3 olarak seçilmiştir.

Aynı Üst Alem Hizalamaları

hsa-mmu ve atc-eco türlerine ait 11 yüksek-düzeyli metabolik kategori için elde edilmiş sonuçlar Tablo 3.1'de gösterilmektedir. Tablodaki her çoklu-satır, 1.1 den başlayarak 1.11 . metabolizmaya kadar olan ağlar için, ilk satırı hsa-mmu için, ikinci satırı atc-eco için olmak üzere, bu tür çiftleri üzerinde yapılan deney sonuçlarını içermektedir. Tablodaki TR kolonu, o ağ çiftinde bulunan toplam reaksiyon sayısını belirtmektedir. *Kapsam* kolonu, hizalama içindeki eşleşmelerde bulunan toplam reaksiyon sayısını göstermektedir. *Doğru eşleşmeler* kolonu, hizalamadaki doğru eşleşme sayısını ifade ederken, *oran* kolonu da o hizalamada elde edilen doğru eşleşmelerin, tüm eşleşmelere oranını göstermektedir. Her altkolon içinde, eşleşme skorlarının elde edildiği algoritmanın ismi belirtilmektedir. S ile gösterilmiş olan kolon SubMAP uygulanılarak elde edilen sonuçlar için, C1 ile gösterilmiş olan kolon CAMPways algoritmasının $\mathcal{W}_1^{'}$ skorlaması ve α=0.3 olduğu durumda elde edilmiş sonuçlar için kullanılmıştır. Bu skorlama yapısı için, diğer α değerleri de neredeyse aynı sonuçları sunmaktadır. C2 ile gösterilmiş olan kolon, CAMPways algoritmasının $\mathcal{W}_2^{'}$ skorlaması ve α_1=0.4, α_2=0.5 , α_3=0.1 durumundaki sonuçlar için kullanılmaktadır. Her iki algoritmanın da kapsama düzeyi birbirine yakın olup, bazı örnekler için SubMAP daha iyi sonuçlar sağlarken, çoğu zaman CAMPways algoritmasının iki versiyonu da daha iyi kapsama göstermektedir. Doğru eşleştirme sayısı bakımından, CAMPways sonuçları kesinlikle SubMAP sonuçlarından üstündür. Atc-eco türünün 1.11 metabolizması için hizalanmasında, SubMAP, CAMPwaysden daha çok kapsama sağlamasına rağmen (153'e 134), CAMPways yine de SubMAP'den daha doğru sonuçlar sunmaktadır. (60'a 53). Bu durum, bazı durumlarda SubMAP'in daha fazla kapsama sağlamasına rağmen, sağladığı birçok eşleşmede, reaksiyonların aynı yolağı paylaşmadığını göstermektedir. Tablo 1'deki, 22 örnek için, 5 örnek de, SubMAP in fazla bellek tüketim problemi olması gerekçesi ile, SubMAP sonuçları alınamamış ve bu alanlar boş bırakılmıştır. 16 örnek için, CAMPways, daha fazla doğru eşleşme sunarken, sadece 1 örnekte hem SubMAP hem de CAMPways aynı sayıda doğru eşleşme sağlamıştır. Ayrıca bulunmuş oranlar da, CAMPways algoritmasının SubMAP'den daha üstün olduğunu doğrulamaktadır. Burada dikkat edilmesi gereken nokta, oranın doğru eşleşme sayısı ile tüm çıktı eşleşmelerini oranlamış olmasıdır; oran doğru eşleşme sayısı ile kapsanan reaksiyon sayısını oranlamamaktadır. Böylece, doğru eşleşmeler kapsamında bir ölçüm olarak kullanılmaktadır.

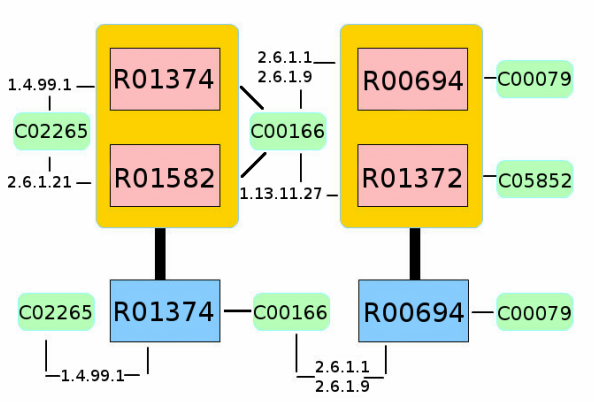
Farklı Üst Alem Hizalamaları

İlgilendiğimiz türlerin her çifti için, her biri farklı üst alemden seçilmiş olmak üzere, aynı testler uygulanmış ve her metabolizma için 4 tane ikili hizalama elde edilmiştir.

Tablo 3.2: Aynı üst alemler için biyokimyasal önem deneyleri.

Düzey 1		Düzey 2		Düzey 3		Düzey 4		Düzey 5	
S	C	S	C	S	C	S	C	S	C
-	*193*	-	*193*	-	*193*	-	*192*	-	*192*
-	154	-	154	-	151	-	144	-	138
23	*23*	*22*	*23*	*22*	*23*	*21*	*23*	*21*	*22*
32	41	32	41	32	39	32	39	32	39
323	*343*	*323*	*343*	*323*	*343*	*318*	*340*	*316*	*338*
97	105	97	105	97	104	93	103	92	102
-	*103*	-	*103*	-	*101*	-	*101*	-	*101*
66	84	66	84	64	80	64	80	63	80
209	*229*	*209*	*229*	*208*	*229*	*205*	*227*	*205*	*227*
117	143	110	139	104	132	97	130	93	127
53	*57*	*53*	*57*	*52*	*57*	*52*	*57*	*52*	*56*
37	35	37	35	34	33	33	33	33	32
5	*6*	*5*	*6*	*5*	*6*	*5*	*6*	*5*	*6*
20	21	20	21	20	21	20	21	19	21
-	*123*	-	*123*	-	*123*	-	*123*	-	*123*
96	115	94	114	93	111	93	110	90	109
9	*13*	*9*	*13*	*9*	*13*	*9*	*13*	*9*	*13*
16	17	16	16	16	16	15	15	14	15
14	*16*	*14*	*16*	*13*	*16*	*13*	*16*	*13*	*16*
7	9	7	9	7	9	6	8	6	8
79	*97*	*78*	*97*	*76*	*97*	*76*	*97*	*76*	*97*
44	59	44	58	42	55	42	55	42	54

Bu testler sonucunda, 2 önemli sonuç bulunmuştur. İlki, Tablo 3.1'deki sonuçlar ile karşılaştırıldığında, doğru eşleşmeler ve doğruluk oranları azalmasıdır. Bunun sebebi, türler arasındaki ayrımın artması, dolayısıyla global hizalamanın, daha az benzer eşleşmeler üretmesi ve çıkan eşleşmelerdeki reaksiyonların farklı yolaklara ait olmasıdır. İkincisi, algoritmaların hizalama kaliteleri karşılaştırıldığında, durum, aynı üst alemler üzerinde yapılan testlerden elde sonuçlar gibidir; birçok durumda, CAMPways daha fazla doğru eşleşme ve daha iyi doğruluk oranları sağlamaktadır. 44 örnek durum için, SubMap 20 örnek de sonuç üretememiştir. 7 örnek için, her iki algoritma da aynı sayıda doğru eşleşme sağlamıştır. 16 örnek için ise, CAMPways hizalamaları, daha fazla doğru eşleşme sunarken, sadece 1 örnek de, SubMAP in vermiş olduğu doğru eşleşme sayısı daha fazladır. Farklı üst aleme ait ayrıntılı sonuçların yer aldığı tam tablo için bakınız Ek 1. Ayrıca $i = 1,2,3$ değerleri, ve $\mathcal{W}_2$ skorlamasının $\alpha_1, \alpha_2, \alpha_3$ seçenekleri için farklı değerler denenerek, doğruluk değerlerinin ve *bire-i*li eşleşmelerinin nasıl değişiklik gösterdiği test edilmiştir. Bu sonuçlara ait ayrıntılı açıklamalar yine Ek 2'de bulunabilir.



Şekil 3.2: Amino asit metabolizması ağından CAMPways hizalama örneği.

3.5.2 Hizalamaların Biyokimyasal Önemi

Biyokimyasal önemi açısından, her iki algoritmanın hizalama kalitesini karşılaştırmak için, KEGG'in RCLASS veritabanın bir parçası olan fonksiyonel grup çevrim (FGC) hiyerarşi verileri kullanılmıştır [47]. Veritabanındaki reaksiyonlar, fonksiyonel grup kategorilerine göre hiyerarşik olarak sınıflandırılmıştır. Aynı fonksiyonel grup, parçası olduğu molekülün boyutuna bakılmaksızın, benzer ya da aynı kimyasal reaksiyonlara maruz kalır [59]. Böylece, farklı türlerden bir yolak çiftinin hizalaması, eğer eşleşmiş reaksiyon altkümeleri aynı FGC kategorisi altında sınıflandırılmış ise, hizalama biyokimyasal açıdan doğrulanmış sayılır. KEGG hiyeraşisinde 5 düzey mevcuttur; ilk kök düzeyde 8 tane yüksek düzeyli FGC sınıfı mevcuttur: karbon-ilişkili, hidrojen-ilişkili, izomerizasyon-ilişkili, nitrojen-ilişkili, fosfor-ilişkili sülfür-ilişkili ve halojen-ilişkili. Doğruluk tanımı, bir önceki bölümdekine benzerdir: Eğer, eşleşme içindeki tüm reaksiyonları içeren, en az bir tane, FGC sınıflandırılmasında sabit bir i. düzeyde olan bir kategori varsa, bu eşleşme doğru olarak kabul edilir. Başlangıç düzeyinden başlayarak, hiyerarşinin ilk 5 düzeyi için, CAMPways ve SubMAP algoritmalarının sonuçları karşılaştırılmış ve doğruluk değerlerine bakılmıştır.

Önceki bölümde olduğu gibi, iki tip ölçüm yapılmıştır: Aynı üst alem hizalamaları ve farklı üst alem hizalamaları. Bu ölçümlerin sonuçları Tablo 3.2'dedir. Kullanılmış ağ çitleri, satır ve çoklu-satırların anlamları, önceki bölümde belirtildiği gibidir. S ile ifade edilmiş alt kolonlar, SubMAP hizalamalarının sonuçlarını belirtirken, C ile belirtilmiş alt kolonlar ise, CAMPways algoritmasının $\mathcal{W}_1$ skorlamasının kullanıldığı versiyonunu göstermektedir. $\mathcal{W}_2$ versiyonu kullanıldığında da çok benzer sonuçlar elde edildiği için, bu versiyon tabloya eklenmemiştir. Esas kolon isimleri, eşleşmenin doğruluk tanımı için kullanılan kategorileri sağlayan FGC hiyerarşisinin 5 düzeyini göstermektedir. Bu kolonlardaki her tablo değeri, doğru eşleşmelerin sayısını vermektedir. Sonuçlar incelendiğinde, CAMPways'in SubMAP'den daha iyi performans sergilediği gözlemlenebilir. Aynı üst aleme ait türler için, üzerinde çalışılan ağ

çiftleri, FGC hiyerarşisinde kök-düzey 1'in soyut sınıflandırılmalarından, daha derinlere inildikçe daha az soyut olan sınıflara gidiliyorken, doğru eşleşme sayısının dikkate değecek kadar azalmadığı gözlemlenmiştir. Ayrıca 1.7 glikan biyosentezi ve metabolizması için, hsa-mmu çifti için ortalama 80 eşleşme bulunmasına rağmen, her iki algoritmada çok az sayıda doğru eşleşme göstermiştir. Doğru eşleşmelerin, toplam eşleşme sayısına oranı %6 dır. Bu durum, Tablo 3.1'de sunulmuş olan tersine mühendislik sonuçlarında aynı çiftin %90 olan doğruluk oranına zıtlık göstermektedir. Bunun temel nedeni, ağdaki reaksiyonların çoğunun FGC sınıflandırılmasında yer almamasıdır. Farklı üst alem çerçevesinde elde edilen sonuçlar ile ilgili olarak, Tablo 2'deki sonuçlara benzer oldukları, tüm hiyerarşi düzeylerindeki tüm ağ örnekleri için, CAMPways algoritmasının çoğu durumda, SUBmap'ten daha iyi olduğu söylenebilir. Tek istisna hsa-atc türü için 1.10 metabolizmasıdır; ancak bu örnekte elde edilmiş doğruluk değerleri, bir önem gösteremeyecek kadar azdır. Farklı üst alem verilerinde gerçekleştirilen deneylerin ayrıntılı sonuçları Ek 2'de bulunabilir.

Fonksiyonel grup çevrim hiyerarşisine dayanan söz konusu analizler, KEGG içinde sunulmuş olan RPAIR verisi açısından, atc-eco çiftlerinin üzerinde, her iki algoritmanın da uygulanması ile elde edilen sonuçların sunduğu bir örnek gösterilerek genişletilebilir. Bir reaktant çifti, bir özgenil çifti ve bir enzimatik reaksiyonlar boyunca kimyasal altyapıyı koruyan bir ürün olarak tanımlanabilir. Aslında, RCLASS veritabanı sınıflandırması, reaktant çiftleri ile ilgili olarak bilgi sağlayabilir. RCLASS sınıflandırılması, kimyasal altyapı ya da moleküler hizalamaya dayanan modeller kullanılarak yapılırken, RPAIR, reaktant çiftlerinin manuel elde edilmesi ve biyokimyasal bilgilerin dahilinde yapılan moleküler hizalamalar ile üretilir. CAMPways algoritması tarafından sağlanmış olan örnek eşleşme, Şekil 3.2'de gösterilmiştir. Şekilde üst kısımdaki reaksiyonlar atc ağına , alt kısımdakiler ise eco ağına aittir. Eşleşmiş reaksiyonlar (reaksiyon alt kümeleri) dikey ayrıtlar kullanılarak belirtilmiştir. Enzimler, EC numaraları kullanılarak gösterilirken, bileşenler küçük dikdörtgenler içinde ifade edilmiştir. atc türüne ait R01374 [D-phenylalanine: acceptor oxidoreductase (deaminating)] ve R01582 (D-phenylalanine:2-oxoglutarate aminotransferase) reaksiyonları, eco türüne ait R01374 reaksiyonu ile eşleşmişken, yine atc türüne ait olan R00694 (L-phenylalanine: 2-oxoglutarate aminotransferase) ve R01372[phenylpyruvate: oxygen oxidoreductase (hydroxylating,decarboxylating)] reaksiyonları, eco türüne ait olan R00694 reaksiyonu ile eşleşmiştir. R01374 ve R01582 reaksiyonlarının çıktı bileşeni olan C00166 (phenylpyruvate), R00694 ve R01372 reaksiyonların girdi bileşenidir. Sonuç olarak, atc yolağında, R01374 ve R01582 reaksiyon altkümesine karşılık gelen düğümden, R00694 ve R01372 reaksiyon altkümesine karşılık gelen düğüme yönlü bir ayrıt vardır. Benzer şekilde, eco yolağında, R01374 reaksiyonuna karşılık gelen düğümden, R00694 reaksiyonuna karşılık gelen düğüme yönlü bir ayrıt bulun-

maktadır. Bu durum, sunulan eşleşme içinde korunmuş ayrıtların olduğunu gösterir. Sınıflandırma ile ilgili olarak, R01374 ve R01582 reaksiyonlarının FGC kategorileri içinde tüm 5 düzeyinin aynı olması, bu eşleşmenin RCLASS bakımından da geçerli ve doğru olduğunu onaylar. Her iki reaksiyon da RCLASS özelliğini gösteren en uzak düzeyde birlikte kategorilenmiş olup, RCLASS numaraları RC0006 dır. RPATH verileri incelendiğinde, daha güvenilir bir onay sağlanmış olur: Her iki reaksiyon da RP00289 RPATH'ine aittir. Bunun aksine, SubMAP algoritması sonucunda atc türüne ait olan R01582 ve R01373 [prephenate hydro-lyase (decarboxylating;phenylpyruvate-forming)] reaksiyonları, eco türüne ait olan R01373 reaksiyonu ile eşleşmiştir. R01373 ve R01582 reaksiyonları FGC kategorilerinin 2. düzeyinden itibaren farklılaşmaya başlamış olup, farklı RCLASS girdilerine aitlerdir. Üstelik, RPAIR veritabanında, bu iki reaksiyon arasında dikkate alınacak bir ilişkiye rastlanamamıştır.

3.5.3 Çalışma Hızı ve Bellek Gereksinimleri

G_p, G'_p içindeki her düğümün derecesinin, bir sabit ile sınırlandırıldığı düşünülürse, CAMPways algoritmasının çalışma zamanı, $|V_p|$'nin, genellemeyi bozmaksızın $|V'_p|$ den daha geniş olduğu düşünülerek, $O(|V_p|^2 \log^2 |V_p|)$ dir. Çalışma zamanı analizi ile ilgili ayrıntılı çalışmalara, Ek 2'den bakılabilir. Buna karşılık olarak, SubMAP algoritması için belirgin bir çalışma hızı zamanı analizi sunulmamıştır. Bu bölümde üzerinde durulan tüm deneysel sonuçlar, 24 GB bellek sağlayan, Intel(R) Xeon(R) CPU 2.67GHz özelliklerine sahip makine üzerinde elde edilmiştir. Test edilen tüm ağlar için gerekli olan CPU zamanları Tablo 3.3'de belirtilmiştir. İlk 3 satır, aynı üst alem çerçevesinden elde edilmiş sonuçları gösterirken, diğer satırlarda farklı üst alem çerçevesinden elde edilmiş sonuçlar belirtilmiştir. Kolon isimleri, Tablo 3.2 de ifade edildiği gibidir. Tabloda 0.1'den küçük bütün zamanlar 0.1'e yuvarlanmıştır. SubMAP algoritmasının önemli bir kısıtlaması, aşırı bellek tüketimidir; bazı ağ çiftleri için,SubMAP kodu sonuna kadar çalıştırılamamaktadır. Örneğin, hsa-mmu türünün 1.1 karbonhidrat metabolizması için hizalaması CAMPways ile 3 dakikadan daha kısa sürerken, SubMAP 2 saat çalıştıktan sonra, tüm belleği tüketip sonlanır.

Aynı üst alem sınamalarında 17 örneğin 15inde, CAMPways SubMAP'den daha hızlı çalışmaktadır. Farklı üst alem çerçevesinde, CAMPways, 28 örneğin 14ünde SubMAP'ten daha iyi çalışma hızı sağlamaktadır. SubMAP'in daha iyi sonuç verdiği örneklerde, her iki algoritma da oldukça yakın zamanlarda tamamlanırken, CAMPways algoritmasının hızlı çalıştığı örneklerde, CAMPways ve SubMAP'in çalışma zamanları arasındaki fark fazladır. Aynı ve farklı üst alem çerçevelerinde, algoritmaların hesap zamanı verimlilikleri arasındaki fark oldukça ilginçtir. Aynı üst alem çerçevesinde, tür çiftlerinin metabolik ağları evrimsel olarak birbirine yakındır. Bu yüzden, hizalanmış ağlarda oldukça fazla sayıda korunmuş ayrıt olduğu gözlemlenir.

Tablo 3.3: CPU zamanları saniye cinsinden S ve C altkolonlarında gösterilmiştir.

S	C	S	C	S	C	S	C	S	C	S	C
3.0	0.3	62.8	2.3	454.2	13.4	1620	15.7	975.3	39.9	121.4	25.2
48.1	1.4	18.0	0.9	0.3	2.9	0.5	0.3	1788.8	25.2	0.1	0.1
0.2	0.1	0.1	0.1	0.1	0.1	3.3	1.0	0.7	5.4		
33.2	2.8	6.6	0.8	6.5	0.7	34.7	2.7	40.5	1.7	21.5	1.2
20.7	1.1	42.0	1.4	0.4	10.3	0.3	6.6	0.4	6.1	0.4	10.2
0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
0.1	0.1	0.1	0.1	1.3	11.0	1.9	20.0	1.8	13.2	1.3	9.6

Ağ çiftlerindeki reaksiyonların çoğu, CAMPways algoritmasının ana döngüsünde hizalanır, bu yüzden de üretilen çelişki çizgesi, fazla korunmuş ayrıt olması sebebiyle oldukça büyüktür. Bu durumda, problemin doğası gereği, hem homolojik hem de topolojik optimizasyon eş zamanlı olarak düşünülür. Tür çiftleri, evrimsel açıdan uzak olduğu durumda, iki algoritmanın da ana kısmında bulunan korunmuş ayrıt sayısı doğal olarak az olmaktadır ve bu durumda her iki algoritmanın temel işlevi sadece yüksek homolojik benzerlik veren hizalamalar üretmeye indirgenir.

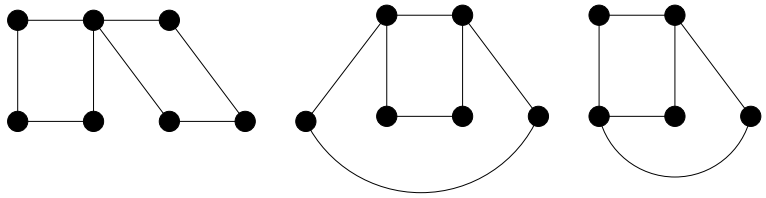
Bölüm 4

Eşleme Kısıtlı Global Ağ Hizalamaları[1]

Önceki bölümde kısıtlı hizalama çerçevesi tanıtılmış ve bu genel çerçeve kullanılarak global bire-çoklu hizalamaya yönelik bir algoritma geliştirilmişti. Bu bölümde kısıtlı hizalamanın özel durumları ile ilgili çizge-teorik incelemeler yapılacak ve bulunan teorik sonuçlar sunulacaktır. Kısıtlı hizalamada verili bir ağ çiftinden her proteinin diğer ağdan yalnızca belli sayıda proteinle eşleşme olasılığı vardır. Bu tarz eşleşme kısıtları ile problem boyutu küçültülerek hizalama probleminin hesapsal komplek-sitesinin azaltılması sağlanır. Eşleşme olasılığı olan çiftlerin verili olduğu varsayılır; bu veri bir çeşit benzerlik tanımı, genelde de dizisel benzerlik, kullanılarak oluşturulur. Formel olarak, $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$ bir çift yönsüz çizge ve S, (V_1, V_2) kümeleri katmanları olmak üzere, bir çift katmanlı çizge (bipartite graph) olsun. S çizgesinde $i = 1, 2$ için V_i katmanındaki her düğümün derecesi maksimum m_i olsun. Legal bir A hizalaması S'de bir eşlemeye (matching) karşılık gelir; yani her düğüm diğer ağdan sadece bir düğüm ile eşleşebilir. u_1u_2, v_1v_2, A'dan bir çift ayrıt olsun, öyle ki $u_1, v_1 \in V_1$ ve $u_2, v_2 \in V_2$. Eğer $u_1v_1 \in E_1$ ve $u_2v_2 \in E_2$ ise bu çift ayrıt bir *korunmuş ayrıt* üretiyor denir. Kısıtlı hizalamada amaç maksimum sayıda korunmuş ayrıt üreten legal hizalama bulmaktır. Bu bölümde sunulan bütün sonuçlar, sabit parametre kolay işlenirlik (fixed parameter tractability) sonucu hariç, $m_2 = 1$ durumu, yani herbir G_2 düğümünün sadece bir G_1 düğümü ile eşleşme olasılığı olduğu durum, ile ilgilidir. Sonuçlar önceki bölümde tanımlanan çelişki çizgelerinin çizge-teorik özelliklerinin ayrıntılı incelenip çıkarsanmasından bulunur.

Literatürde pekçok ilgili çalışma yapılmıştır. Goldman ve arkadaşları kontak

[1] Bu bölümde işlenen sonuçlar *Discrete Applied Mathematics* dergisinde yayın amaçlı değerlendirme aşamasındadır. Ayrıntılar için Ek 3'e bakınız.



Şekil 4.1: İki c_4'ün $\mathcal{C}_U$'da olası bütün çelişme konfigürasyonları.

haritası örtüşmesi (contact map overlap) problemini ele almışlardır. Amaç yine yine korunmuş ayrıt sayısını maksimize etmek olsa da kısıtlı hizalamadan farklı olarak S çizgesi gibi kısıt bilgisinin olduğu bir yapı söz konusu değildir. Onun yerine G_1, G_2 düğümleri için doğrusal bir sıranın verili olduğu varsayılır ve çıktı hizalamada bu sıranın korunması beklenir. Amacın bütün ayrıtların korunması olduğu ortolojiler ile eşleşme problemi de bir başka ilintili problemdir [27]. Problemin $m_1 = 1, m_2 = 3$ ve çizgelerin çift katmanlı olma durumunda bile NP-zor olduğu gösterilmiştir. Son olarak, Fertin ve arkadaşları $MAX(\mu_G, \mu_H)$ olarak isimlendirdikleri burda tanımladığımız kısıtlı hizalama probleminin aynısını $m_2 = 1$ durumu için çalışmışlardır [29]. Problemin APX-zor olduğunu göstermiş ve birçok yaklaşım algoritması (approximation algorithm) ve sabit parametre kolay işlenirlik sonuçları sunmuşlardır. Onlar da çelişki çizgesi tanımlayıp sabit parametre kolay işlenirlik sonucunu bu çizgenin derecesi ile ilgili kolay bir argümana dayandırsalar da, bu çizgelerle ilgili başka herhangi yapısal özellik sunmamışlardır.

Takip eden altbölümlerde kısıtlı hizalama probleminin çeşitli özel durumlarında çelişki çizgelerinin birçok yapısal özelliğini sunacağız. Bu özellikler ilgili maksimum bağımsız küme (maximum independent set) çözümlerini kullanmayı elverişli kılıp [29]'den daha güçlü algoritmik sonuçlar elde etmemizi sağlar. Spesifik olarak, $\Delta(G)$, G'nin derecesi (çizge derecesi çizgedeki düğümlerin maksimum derecesidir) olmak üzere, [29] problemin çift $\Delta(G_1)$ için $2\lceil 3\Delta(G_1)/5 \rceil$ yaklaşımlı tek $\Delta(G_1)$ için $2\lceil (3\Delta(G_1)+2)/5 \rceil$ göstermişlerdir. Biz ise $\Delta_{min} = min(\Delta(G_1), \Delta(G_2))$ olmak üzere, problemin $\Delta_{min} + 1$ yaklaşımlı olduğunu göstereceğiz. Benzer şekilde onlar probleme, G_1 sınırlı derece çizge (bounded degree graph) olmak üzere sabit parametre kolay işlenirlik çözümü sunarken, biz çizge dereceleri için herhangi bir kısıt olmaksızın aynı çözümü sunmaktayız. Hatta m_2 üstündeki kısıtı kaldırarak, en genel haliyle herhangi positif tamsayı m_1, m_2 için problemin derece kısıtlı çizgeler için sabit parametre kolay işlenir olduğunu göstereceğiz.

4.1 Çelişki Çizgesi Oluşturma

Takip eden kısım için $c_4(x) = abcd$ ile gösterilen bir 4-çevrim (4-cycle) x, $ab \in E_1$, $cd \in E_2$ ve $ad, bc \in S$ olmak üzere $a-b-c-d-a$'dan oluşan bir çevrimdir. İki c_4 için eğer S ayrıtları herhangi bir legal hizalamada birlikte yer alamıyorsa, yani en az bir çift ayrıtları bir S eşleşmesinde yer alamıyorsa, *çelişiyorlar* denir. Verili $\prec G_1, G_2, S \succ$ için çelişki çizgesi $\mathcal{C}$, her c_4'e karşılık bir düğüm ve her çelişen c_4 çifti arasına da bir ayrıt eklenerek oluşturulur. $\mathcal{C}_U$ çelişki çizgesinin altında yatan çizge, yani çelişki çizgesinde herhangi bir c_4 de yer almayan bütün düğüm ve ayrıtlar hariç, G_1, G_2 ve S'nin birleşimi olsun. $\mathcal{C}_U$'da çelişen herhangi c_4 çiftinin Şekil 4.1'de gösterilen üç konfigürasyondan birinde olması gerektiğini görmek kolaydır. Şekilde her konfigürasyon için üstteki düğümler V_1 düğümleri iken alttakiler V_2 düğümüdürler. Çelişki kategorileri soldan sağa doğru *Tip1, Tip2, Tip3* olarak adnlandırılır. Bu tipler için c_4 çiftleri sırasıyla bir, iki ve üç düğüm paylaşırlar.

Çelişki çizgesinin bu oluşum tanımıyla kısıtlı hizalama problemi açık bir şekilde maksimum bağımsız küme problemine dönüşür. $\mathcal{M}$, çelişki çizgesi $\mathcal{C}$'nin bir bağımsız kümesi olsun. $\mathcal{M}$ düğümlerine karşılık gelen $c_4$lerin S ayrıtları kümesi, ancak ve ancak $\mathcal{M}$, $\mathcal{C}$'nin maksimum bağımsız kümesi ise kısıtlı hizalamanın optimum çözümünü verir. Bu dönüşüm sayesinde çelişki çizgesinin faydalı yapısal özellikleri çıkarsanırsa ilgili bağımsız küme çözümleri hizalama problemine çözüm getirebilecektir.

4.2 Özel Durum $m_1 = 2$ İçin Kısıtlı Hizalamalar

Önce $m_1 = 2$ durumu için çelişki çizgesi özellikleri çıkarsayacak sonra bu özellikleri kullanarak polinom zamanlı bir yaklaşım algoritmasdı sunacağız. $c_4(v) = abcd$'ye göre Tip1 ve Tip3 çelişkileri ikiye ayıralım. $X = 1, 3$ için, TipXa, çelişki a düğümünde olan durumları, TipXb ise çelişki b düğümünde olan durumları gösteriyor olsun. Spesifik olarak $f \neq d$ için çelişen bir c_4 $af \in S$ içeriyorsa birinci tip, $bf \in S$ içeriyorsa ikinci tip sayılır.

Gerçek 4.2.1. *Herhangi $a, b \in V(G_1)$'nin G_2'de ortak komşusu yoktur.*

Gerçek 4.2.2. *Düğüm v, $c_4(v) = abcd$'yi gösteriyor olsun. Düğüm v'nin $\mathcal{C}$'de Tip2 çelişkisinde en fazla bir komşusu vardır.*

İspat. $c_4(x) = abef$, v'nin Tip2 çelişkisinde ilk komşusu olsun. $c_4(y) = abrs$'yi de ikinci bir Tip2 çelişkili komşu varsayalım. Tip2 tanımından ve Gerçek 4.2.1'den c, d, e, f'nin dört farklı düğüm olması gerekir. r, $\{c, d, e, f\}$ içinde olmalıdır, yoksa a ya da b için m_1 kısıtı sağlanamaz. Öte yandan Gerçek 4.2.1 ve r'deki m_1 kısıtı gereği r, $\{c, d, e, f\}$ içinde olamaz. □

Gerçek 4.2.3. *Düğüm v, $c_4(v) = abcd$'yi gösteriyor olsun. Düğüm v'nin C'de Tip3a ve Tip3b çelişkili en fazla birer komşusu vardır.*

İspat. $c_4(x) = abce$, v'nin Tip3a çelişkisinde ilk komşusu olsun ve $c_4(y) = abrf$'nin de Tip3a çelişkili v'nin bir başka komşusu olduğunu varsayalım. Eğer $f \notin \{c,d,e\}$ ise, a için m_1 kısıtı sağlanamaz. Öte yandan $f \in \{c,d,e\}$ ise, $\{c,d,e\}$'den biri hem a hem b'ye komşudur, ki bu da Gerçek 4.2.1 ile mümkün değildir. Aynı argüman Tip3b için de geçerlidir. □

Gerçek 4.2.4. *Düğüm v, $c_4(v) = abcd$'yi gösteriyor olsun. Düğüm v ile Tip2, Tip3a ve Tip3b çelişkili $c_4$ler varsa, tektirler ve v ile birlikte C'de, $t = 0,1,2,3$ bu tiplerde çelişen $c_4$lerin sayısı olmak üzere, bir K_{1+t} oluştururlar.*

Propozisyon 4.2.5. *üğüm v, $c_4(v) = abcd$'yi gösteriyor olsun. $m_1 = 2$ ve $m_2 = 1$ olsun.*

1. *v ile Tip1a çelişkili herhangi bir c_4'ün C'de kendisinin de v ile Tip1a çelişkisi olan en fazla bir komşusu vardır. Aynısı Tip1b çelişkileri için de doğrudur.*
2. *v ile Tip1a çelişkili herhangi bir c_4'ün C'de kendisinin de v ile Tip1b çelişkisi olan en fazla bir komşusu vardır. Aynısı Tip1b çelişkileri için de doğrudur.*
3. *v ile Tip1 çelişkili herhangi bir c_4'ün v ile Tip2 çelişkili bir komşusu yoktur.*
4. *$l \geq 3$ için, sadece v ile Tip1 çelişkili $c_4$lerden oluşan tam çizge K_l yoktur.*
5. *$k \geq 4$ için, sadece v ile Tip1 çelişkili $c_4$lerden oluşan indükte patika P_k yoktur.*

İspat. Bakınız Ek 3. □

Gerçek 4.2.6. *Düğüm v, $c_4(v) = abcd$'yi gösteriyor olsun. Düğüm v ile Tip3a çelişkili c_4, v ile Tip1b çelişkili her c_4 ile komşudur ve düğüm v ile Tip3a çelişkili c_4'ün v ile Tip1a çelişkili $c_4$ler ile çelişkisi yoktur. Aynısı Tip3b çelişkiler için de doğrudur.*

İspat. $c_4(x) = apef$ Tip1a çelişkili bir c_4 olsun. Propozisyon 4.2.5 ile, Tip1a çelişkili c_4 ayrıt af'yi kullanmalı ve Tip3a düğümü de af'yi içermeli. Bu da $c_4$lerin çelişmediği anlamına gelir. Her Tip1b c_4'ün içerdiği ayrıt bg olsun. Gerçek 4.2.1 ile, f ve g farklıdırlar ve böylece Tip3a olan c_4 Tip1b olan herhangi c_4 ile çelişir. □

Teorem 4.2.7. *$m_1 = 2$ ve $m_2 = 1$ için, tekerlek çizge $W_k, k \geq 5$ çelişki çizgesinin indükte bir altçizgesi değildir.*

İspat. Düğüm v, $c_4(v) = abcd$'yi gösteriyor olsun. İndükte çevrimi (cycle) C_k olan ve C_k'nin tüm düğümlerine komşu merkez düğümü v olan bir tekerlek W_k'yi alalım. $k \leq 4$ olduğunu ispatlarız. Gerçek 4.2.4 sayesinde Tip3a, Tip3b ve Tip2 düğümleri

aynı zamanda var olamazlar çünkü diğer türlü bir K_4 oluştururlar ve $W_k, k \geq 5$ içinde K_4 yoktur. Dahası Tip2 düğümü yoktur, çünkü Tip3a ve Tip3b'nin her ikisini gerektirir (v ile hep birlikte bir K_4 oluştururlar). Gerçek 4.2.6 sayesinde Tip3a ve Tip3b olan her iki düğümün C_k oluşturması gerekir ve Tip1a ile Tip1b'den en fazla bir düğüm seçebiliriz. C_k'yi, sadece Tip1 düğümleriyle oluşturmak istersek, Propozisyon 4.2.5 sayesinde, böylesi en fazla dört düğüm olabilir. Böylece en fazla $k \leq 4$ için W_k oluşturulabilir. □

Fan·çizgeF_n'nin n düğümlü patika P_n ve de onun bütün düğümlerine bağlı bir v düğümünden oluştuğunu hatırlatalım.

Sonuç 4.2.8. *$m_1 = 2$ ve $m_2 = 1$ için, $F_t, t \geq 8$, C'nin indükte bir altçizgesi değildir.*

İspat. Düğüm v'nin komşularının oluşturduğu indükte patika $P_t, t \geq 4$'ü düşünelim. Propozisyon 4.2.5 ile Tip1 düğümlerden en fazla P_3 seçebiliriz. P_t'de hem Tip3a hem Tip3b seçersek, en fazla bir Tip1a ve bir Tip1b düğümü olabilir. Tip2 düğüm kullanmak bütün Tip1 düğümlerini eler. Propozisyon 4.2.5 ile Tip3 düğümün iki P_3'ü Tip1 düğümlerle birleştirdiği bir P_7 olası en büyüktür. □

$\Delta(C)$ çelişki çizgesi C'nin derecesini ve α da çizgenin bağımsızlık numarasını (maksimum bağımsız kümesinin büyüklüğü) gösteriyor olsun. Aşağıdaki sonuç ile boyutu $\Delta(C)$'ye bağlı bir bağımsız küme ve dolayısıyla bağımsız küme boyutunda korunmuş ayrıtı olan bir global hizalama oluşturulabilir.

Sonuç 4.2.9. *$m_1 = 2$ ve $m_2 = 1$ için, $(\Delta(C) - 2)/2 \leq \alpha(C)$.*

İspat. Düğüm v be C'nin en büyük dereceye sahip düğümü olsun. Düğümün Tip1 komşularını ve varsa Tip2 komşularını düşünelim. Propozisyon 4.2.5 ile Tip1 komşular P_1, P_2, P_3, C_4 şeklinde olabilir. Her P_1, herbir P_2'den bir düğüm, herbir P_3'ten iki düğüm, her C_4'ten iki düğüm ve de Tip2 komşular bağımsız kümeye katılır. □

4.3 Herhangi Sabit m_1 İçin Kısıtlı Hizalamalar

Daha genel olan m_1'in herhangi pozitif sayı olması durumu için de yapısal özellikler sunarak önceki altbölümün sonuçlarını genişleteceğiz. Önceki altbölümdeki gibi bu özellikler kullanılarak kısıtlı hizalamaya yönelik uygun polinom zamanlı algoritmalar vereceğiz.

4.3.1 Çelişki Çizgesinde Tekerlek Altçizgeler

Gerçek 4.3.1. *Çelişen c_4 çiftleri bir G_1 düğümünü paylaşırlar.*

Gerçek 4.3.2. *G_1'den iki düğüm paylaşan bir c_4 çifti çelişkilidir.*

Lemma 4.3.3. *C'nin indükte bir P_4'ünün $c_4$lerinin tümü bir G_1 düğümü paylaşamaz.*

İspat. C'nin indükte bir P_4'ü $x-w-y-z$ ve $c_4(x) = abcd$ olsun. Düğüm $a \in G_1$ bütün $c_4$lerde ortak olduğunu varsayalım. $c_4$ler $c_4(y)$ ve $c_4(z)$'nin her ikisi de ayrıt ad'yi içermelidir, diğer türlü herbiri $c_4(x)$ ile çelişirdi. Benzer şekilde $c_4(w)$ de ad'yi içermelidir, diğer türlü $c_4(w)$ ve $c_4(z)$ çelişirlerdi. O halde tüm $c_4$ler ayrıt ad'yi içerir ve böylece herhangi bir çift arasındaki çelişki ancak Tip3 olabilir ki bu da bütün $c_4$lerin b'yi içermesini gerektirir. Gerçek 4.3.2 ile bu $c_4(x)$ ve $c_4(y)$ arasında bir çelişkiye neden olur ki bu da indükte P_4'de mümkün değildir. □

Bu ara sonuçlarla, Teorem 4.2.7'ye analog aşağıdaki teoreme erişiriz:

Teorem 4.3.4. *$m_2 = 1$ için, $W_k, k \geq 7$, C'nin indükte bir altçizgesi değildir.*

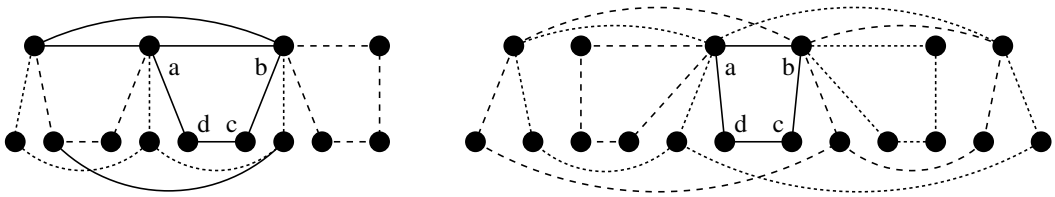
İspat. İndukte bir W_k varlığını varsayalım ve $c_4(v) = abcd$ de merkezi düğüm olsun. $x_1 - x_2 \ldots x_k - x_1$, çelişki çizgesinde tekerlek W_k'nin indükte $C_k$si olsun. Gerçek 4.3.1 ile her $x_i, 1 \leq i \leq k$'nin a veya b'den en az birini karşılık gelen $c_4$ünde içermesi gerekir. Lemma 4.3.3 ile hepsinin a'yı içermesi de imkansızdır, hepsinin b'yi içermesi de. Böylece birinde a diğerinde b olan bir çift çelişen c_4 olmalı ki karşılık gelen düğümleri C_k'de komşu olsun. İlki $c(x_1) = aklm$, diğeri $c(x_k) = bpqr$ olsun.

İspat $c_4$ler x_3, x_4, x_5'in olası konfigürasyonlarına dayalıdır. $3 \leq j \leq 5$ için, eğer x_j, a'yı içerirse ayrıt am'yi, eğer b'yi içerirse ayrıt br'yi içermelidir. Bunun nedeni her x_j'in a ya da b'yi içermesi ve x_1 veya x_k ile çelişmemesi gerektiğidir. Dahası $3 \leq j \leq 5$ için, x_j a ve b'yi birlikte içeremez. İçerseydi $abrm$'ye eşit olurdu ve $abrm$ ile çelişen her c_4'ün ayrıt am', $m' \neq m$'yi ya da ayrıt br', $r' \neq r$'yi içermesi gerekirdi. Ancak böylesi durumda $3 \leq j \leq 5$ ve $j' \neq j$ için, x_j, hiçbir $x_{j'}$ ile çelişemezdi. Yukarda anılan özelliklerden çıkan bir başka gerçek de $c_4$ler x_3, x_4, x_5'in hepsinin a, b hariç bir $x \in V_1$ düğümünü paylaşmaları gerektiğidir. Bunun nedeni a, b'nin hiçbir c_4'te birlikte yer alamaması ve de x_4'ün x_3 ve x_5 ile çelişmesidir. Aşağıda $c_4$ler x_3, x_4, x_5'in olası üç konfigürasyonunu inceleyeceğiz. Herbirinde $c_4$ler arasında bir çelişki ortaya çıkar, ki böylesi bir çelişki mümkün değildir.

Durum-1: x_3 ve x_5'in aynı ayrıt am veya br'yi paylaştıklarını varsayalım. Üç c_4 de a veya b'ye ek x'i de paylaştıklarından Gerçek 4.3.2 ile x_3 ve x_5 çelişirler.

Durum-2: x_4 ve x_5'in x_3'te olmayan am veya br'yi paylaştıklarını varsayalım. Genellik kaybolmaksızın x_3 ayrıt am'yi ve x_4, x_5 ayrıt br'yi içersin. Bu durumda eğer x_6, a'yı içerirse ayrıt amyi, eğer b'yi içerirse ayrıt br'yi içermelidir. Bunun nedeni x_6'nın a ya da b'yi içermesi ve de x_3, x_4 ile çelişmemesi gerektiğidir. x_6, x_5 ile çeliştiğinden bu, x_6'nın $c_4$ler x_3, x_4, x_5'in tümünce paylaşılan aynı düğüm x'i de içermesini gerektirir. Ama bu durumda Gerçek 4.3.2 ile x_6, x_3 veya x_4 ile çelişir.

Durum-3: Son olarak x_3, x_4'ün x_5'de olmayan am veya br'yi paylaştıklarını varsayalım. İspat Durum-2'dekine benzer. Genellik kaybolmaksızın x_5 ayrıt br'yi



Şekil 4.2: $m_1 = 3$ için W_5 (sol) ve W_6 (sağ) içeren örnek $C_U$lar.

ve x_3, x_4 ayrıt am'yi içersin. Bu durumda eğer x_2, a'yı içerirse ayrıt amyi, eğer b'yi içerirse ayrıt br'yi içermelidir. Bunun nedeni x_2'nin a ya da b'yi içermesi ve de x_4, x_5 ile çelişmemesi gerektiğidir. x_2, x_3 ile çeliştiğinden bu, x_2'nin $c_4$ler x_3, x_4, x_5'in tümünce paylaşılan aynı düğüm x'i de içermesini gerektirir. Ancak böylesi bir durumda da Gerçek 4.3.2 ile x_2, x_4 veya x_5 ile çelişir. □

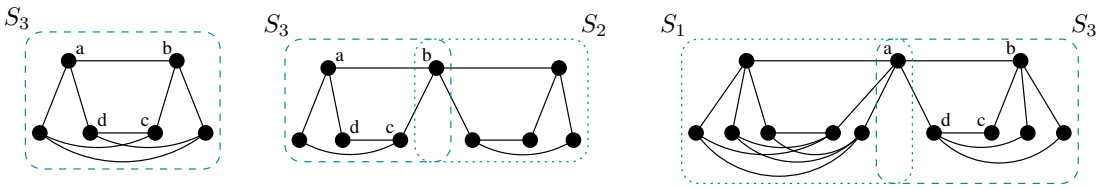
Önceki altbölümde Teorem 4.2.7'nin $m_1 = 2$ için, $W_k, k \geq 5$'in çelişki çizgesinde yer alamayacağını belirttiğini not edelim. Üstteki teorem herhangi m_1 için aynısının $W_k, k \geq 7$ için doğru olduğunu söylese de, W_5 ve W_6, $m_1 > 2$ için bir çelişki çizgesinde hala olasıdır; bakınız Şekil 4.2. Şekilde tekerleklerin merkezini $abcd$ ile gösterilen c_4 oluşturur. Dolayısıyla indükte tekerlek altçizgeler baklımından sonuçlarımız hiç açıklık bırakmaz.

4.3.2 Çelişki Çizgesinde Klik Altçizgeler

Bu altbölümde herhangi m_1 için çelişki çizgelerindeki klik altçizgelerini ilgilendiren sonuçları sunacağız. C'de bir klik K_t olduğunu varsayalım ve bu klikten bir düğüme karşılık gelen c_4 ise $c_4(x) = abcd$ olsun. K_t'deki tüm $c_4$leri referans $c_4(x)$'e göre üç ayrık referans kümesine ayırırız. S_1, S_2 sırasıyla $c_4(x)$ ile Tip1a ve Tip1b çelişkileri olan tüm $c_4$lerden oluşsun. S_3 de $c_4(x)$ ile Tip2 ve Tip3 çelişkileri olan tüm $c_4$ler ile $c_4(x)$'in kendisinden oluşsun.

Lemma 4.3.5. *Farklı referans kümelerinden bir çift c_4, bir S ayrıtı paylaşmaz.*

İspat. c_4 çifti aynı kliğin parçaları olduğundan, G_1'den en az bir düğüm paylaşarak çelişmeleri gerekir. İki duruma bakarız. İlk durum için $c_4$lerden birinin S_1 veya S_2'de diğerinin S_3'te olduğunu varsayalım. Genellik kaybolmaksızın ilk c_4'ün S_1'de olup, $x \neq b$ olmak üzere G_1'den x, a düğümlerini içerdiğini varsayalım. S_3'ten diğer c_4, G_1'den a, b'nin her ikisini içerdiğinden, bu çift c_4 sadece düğüm a'yı paylaşabilir ve bu da aralarında Tip1 çelişkiye yol açar. İkinci durum için $c_4$lerden birinin S_1'de ötekinin S_2'de ye raldığını varsayalım. Bu durumda referans $c_4(x) = abcd$ ile ilkinin Tip1a çelişkisi, diğerinin Tip1b çelişkisi olması gerekir. S_1 ve S_2'den $c_4$ler $a \neq b$ olduğundan G_1'den sadece bir düğüm paylaşabilirler ve bu da çift arasında bir Tip1 çelişkisine yol



Şekil 4.3: $K_{m_1^2}$ içeren örnek $\mathcal{C}_U$lar.

açar. Yani her iki durumda da çiftin birbiriyle Tip1 çelişkisi vardır. Tip1 çelişkisi olan c_4 çiftinin S ayrıtı paylaşmadığı gerçeği ispatı tamamlar. □

Teorem 4.3.6. *$m_2 = 1$ için, $\mathcal{C}$'de en büyük kliğin boyutu m_1^2dir.*

İspat. İki duruma bakarız:

Durum-1: Önce S_1, S_2'den en az birinin boş olduğu duruma bakalım. Genellik kaybolmaksızın S_1 boş varsayalım. S_3 $c_4$lerilerinde b'ye dokunan S ayrıtlarının sayısı p olsun. Biri a'ya biri b'ye dokunan her çift S ayrıtı en fazla bir c_4 oluşturduğundan S_3'teki $c_4$lerin sayısı en fazla $m_1 p$dir. Lemma 4.3.5 ile S_3'teki $c_4$ler S_2'deki $c_4$ler ile bir S ayrıtı paylaşamaz. Buna göre S_2'de bulunan $c_4$lerde b'ye dokunan S ayrıtlarının sayısı en fazla $m_1 - p$dir. Böylesi bır ayrıt bc' olsun ve $S_{bc'}$, bc'ı paylaşan S_2'de bulunan $c_4$lerin kümesi olsun. $S_{bc'}$'dan herhangi bir çift c_4, S'den bir ayrıt paylaştıklarından aralarında Tip3 çelişkisi içinde olmaları gerekir. Bu da b'ye ek olarak bir G_1 düğümü daha paylaşmalarını gerektirir. Buna göre $|S_{bc'}| \leq m_1$ ve bu da toplamda en fazla $(m_1 - p)m_1$ tane c_4'ün S_2'de olmasını getirir. S_2, S_3'ten $c_4$lerin oluşturduğu klikte en fazla m_1^2 düğüm olur.

Durum-2: S_1, S_2'nin her ikisinin boş olduğu duruma bakalım. $S_1 \cup S_2$'deki bütün $c_4$lerin bir G_1 düğümü e'yi paylaşmaları gerekir, öyle ki $e \neq a, e \neq b$. Bunun nedeni biri S_1'den diğeri S_2'den herhangi bir çift c_4'ün sadece Tip1 çelişkisine sahip olabilecekleri ve Tip1 çelişkisinde paylaşılan düğümün a ya da b olamayacağıdır. S_3 $c_4$lerinde a ve b'ye dokunan S ayrıtlarının sayısı sırasıyla p, q olsun.

S_3'deki $c_4$lerin sayısı en fazla pqdur. Lemma 4.3.5 ile S_1 $c_4$lerinden a'ya dokunan S ayrıtlarının sayısı en fazla $m_1 - p$ ve S_2 $c_4$lerinden b'ye dokunan S ayrıtlarının sayısı en fazla $m_1 - q$'dur. S_1 $c_4$lerinden e'ye dokunan S ayrıtlarının sayısı r olsun. Yine Lemma 4.3.5 S_2 $c_4$lerinden e'ye dokunan S ayrıtlarının sayısı en fazla $m_1 - r$'dir. Buna göre S_1, S_2'deki $c_4$lerin sayısı sırasıyla $(m_1 - p)r$ ve $(m_1 - q)(m_1 - r)$'dir. Dolayısıyla $1 < p, q, r < m_1$ olmak üzere, tüm üç referans kümesindeki $c_4$lerden oluşan kliğin boyutu en fazla $pq + (m_1 - p)r + (m_1 - q)(m_1 - r)$dir. Genellik kaybolmaksızın $p \leq q$ durumunda $pq + (m_1 - p)r + (m_1 - q)(m_1 - r) \leq pq + (m_1 - p)m_1 < m_1^2$. □

Herhangi pozitif tamsayı m_1 için $K_{m_1^2}$'nin çelişki çizgesi $\mathcal{C}$'de mümkün olduğunu belirtelim. Gerçekten de üstteki ispatın *Durum-1*'i böylesi çelişki çizgesinin nasıl

yaratılacağını da içerir; bakınız Şekil 4.3. Şekilde referans $c_4(x) = abcd$dir. İlk iki $m_1 = 2$ için örnek çizge gösterirken sonuncu $m_1 = 3$ durumu içindir.

Ramsey teorisine dayalı klasik maksimum bağımsız küme sonuçlarından biri Boppana and Halldórsson'a aittir [14]. Verili bir n düğümlü, m ayrıtlı yönsüz çizgede, $|I||C| \geq \frac{1}{4}\log^2 n$ olmak üzere bir I bağımsız kümesi ve C kliğinin $O(n+m)$ zamanda bulunabileceği gösterilmiştir. Bunu Teorem 4.3.6 ile birleştirirsek:

Sonuç 4.3.7. *Verili bir kısıtlı hizalama enstantanesi $\prec G_1, G_2, S\succ$ ve $m_2 = 1$ için, V_C çelişki çizgesi C'nin düğüm kümesi olsun. En az $\frac{\log^2 |V_C|}{4m_1^2}$ korunmuş ayrıt içeren bir hizalama polinom zamanda bulunabilir.*

V_C'nin boyutunun $m_2 = 1$ durumunda $|E_2|$ ile sınırlandığını belirtelim. Sabit r için, maksimum bağımsız küme probleminin K_r içermeyen çizgeler sınıfında çıktı boyutu ile parametrize edilmiş olarak, sabit parametre kolay işlenir (fixed parameter tractable) olduğu gösterilmiştir [69, 23]. Bu sonucu Teorem 4.3.6 ile birleştirerek aşağıdaki sonuca varırız:

Sonuç 4.3.8. *m_1 herhangi pozitif tamsayı, $m_2 = 1$ durumunda kısıtlı hizalama problemi sabit parametre kolay işlenirdir.*

Fertin ve arkadaşlarının sunduğu benzer sonuç sadece sınırlı dereceli çizgeler için geçerli olduğundan, üstte sunduğumuz sonuçtan daha kısıtlıdır [29].

4.3.3 Çelişki Çizgesinde Pençe Altçizgeler

Bu altbölümde bazı pençe (claw) altçizgeleri içermeyen çelişki çizgelerini karakterize edeceğiz. Verili bir yönsüz çizgede bir $d-pençe, talon$ olarak adlandırılan d bağımsız düğümden ve bunların tümüne bağlı bir *merkez* düğümden oluşan indükte altçizgesidir. Δ_1, Δ_2 sırasıyla G_1 ve G_2'nin dereceleri olmak üzere $\Delta_{min} = min(\Delta_1, \Delta_2)$ olsun.

Teorem 4.3.9. *$m_2 = 1$ için, $(2\Delta_{min}+2)$-pençe C'nin indükte altçizgesi değildir.*

İspat. Pençenin merkez düğümüne karşılık gelen c_4 *abcd* olsun. Bununla Tip2 veya Tip3 çelişkisine sahip bir talon *abkl* olsun. Merkez düğüm c_4'ü ile Tip2 veya Tip3 çelişkisine sahip herhangi başka talon da a, b'yi paylaşacağından Gerçek 4.3.2 ile *abkl* ile çelişir, ki bu mümkün değildir. Dolayısıyla *abcd* ile Tip2 ve Tip3 çelişkili talonların sayısı en fazla 1dir. Tip1 çelişkili talonların sayısını bulmak için önce Tip1a çelişkilerine bakalım. Merkez *abcd* ile Tip1a çelişkisinde bir talon *apqr* olsun. Merkez *abcd* ile Tip1a çelişkili herhangi bir talon *ar* ayrıtını paylaşmalıdır, çünkü diğer türlü *apqr* ile çelişir. Düğüm *a*'ya dokunan herhangi G_1 ayrıtı sadece bir c_4'e dahil olabilir, çünkü diğer türlü Gerçek 4.3.2 ile talonlara karşılık gelen bir çift c_4 çelişir. Ayrıca $m_2 = 1$ olduğundan her G_2 ayrıtı tek bir c_4'e dahil olabilir. Dolayısıyla Tip1a çelişkili

talonların sayısı Δ_{min} ile sınırlıdır. Aynısı Tip1b çelişkileri için de geçerli olduğundan bağımsız en fazla $(2\Delta_{min}+1)$ vardır. □

Yukardaki teoremi $d-$pençesiz çizgelerde maksimum bağımsız küme için bir $d/2$ yaklaşımı bulunabileceğini belirten sonuç [12] ile birleştirdiğimizde polinom zamanlı bir yaklaşım elde ederiz:

Sonuç 4.3.10. *$m_2 = 1$ için, kısıtlı hizalama problemine $(\Delta_{min}+1)$ oranlı polinom zamanlı bir yaklaşım algoritması vardır.*

Bulduğumuz bu yaklaşım oranının Fertin ve arkadaşlarının oranından daha iyi olduğunu belirtelim. Onların oran çift $\Delta(G_1)$ için $2\lceil 3\Delta(G_1)/5\rceil$ iken tek $\Delta(G_1)$ için $2\lceil (3\Delta(G_1)+2)/5\rceil$dir [29]. Sonuçları $\Delta(G_1)+1$ oranını ancak lineer arborisite tahmini doğru ise yakalar.

4.4 Herhangi Sabit m_1 ve m_2 İçin Kısıtlı Hizalamalar

Son olarak çelişki çizgesi derecesi ile girdi çizge dereceleri ve m_1, m_2 arasında bir bağıntı kurarız. Bu son durum, m_1 ve m_2 herhangi pozitif tamsayı olabileceğinden öncekilere göre daha geneldir. Bu ayrıca, [29]'de sunulan sonucu da, orda sadece $m_2 = 1$ düşünüldüğünden, genelleştirir

Lemma 4.4.1. *$\mathcal{C}$'nin derecesi $2\Delta_1 m_1^2 + 2\Delta_2 m_2^2 - 2\Delta_1 m_1 - 2\Delta_2 m_2 - m_1^2 - m_2^2 + 2m_1 + 2m_2 - 2$ ile sınırlıdır.*

İspat. Çelişki çizgesi $\mathcal{C}$'de bir düğüme karşılık gelen bir c_4, $c_4(x) = abcd$ olsun. $S = S_1 \cup S_2$, $c_4(x)$'in $\mathcal{C}$'deki komşularının kümesi olsun. İlk küme S_1, $c_4(x)$ ile çelişen ve ad veya bc'yi içeren düğümlerden, S_2 ise $c_4(x)$ ile çelişen geriye kalan düğümlerden oluşur. S_1'den bir c_4, $c_4(x)$ ile ayrıt $ad(bc)$'yi paylaşırsa, $c_4(x)$ ile çelişebilmek için $b(a)$ veya $c(d)$2yi de içermelidir. Her durumda, b ve c (a ve d)'ye dokunan uygun benzerlik ayrıtları ($c_4(x)$ ile çelişki yaratabilecek S ayrıtları) sayısı $m_1 + m_2 - 2$ ile sınırlıdır. Buna göre $2m_1 + 2m_2 - 4$, $|S_1|$ için bir üstsınırdır. İkinci küme S_2 için öncelikle bir çift benzerlik ayrıtının tek bir c_4 yaratabileceğini belirtelim. Buna göre, ab'den farklı her G_1 ayrıtı, S_2'de en fazla $m_1^2 - m_1$ farklı c_4'ün parçası olabilir ve cd'den farklı her G_2 ayrıtı, S_2'de en fazla $m_2^2 - m_2$ farklı c_4'ün parçası olabilir. Düğüm a veya b'ye dokunan, ab'den farklı G_1 ayrıtlarının sayısı en fazla $2\Delta_1 - 2$ ve de düğüm c veya d'ye dokunan, cd'den farklı G_2 ayrıtlarının sayısı en fazla $2\Delta_2 - 2$ olduğundan ab veya cd içermeyen S_2 $c_4$lerinin sayısı $(2\Delta_1 - 2)(m_1^2 - m_1) + (2\Delta_2 - 2)(m_2^2 - m_2)$ ile sınırlıdır. Ayrıtlar ab ve cd'nin kendileri S_2'de sırasıyla en fazla $(m_1 - 1)^2$ ve $(m_2 - 1)^2$ farklı c_4'ün parçası olabilirler. Dolayısıyla herhangi bir c_4'ün derecesi $2\Delta_1 m_1^2 + 2\Delta_2 m_2^2 - 2\Delta_1 m_1 - 2\Delta_2 m_2 - m_1^2 - m_2^2 + 2m_1 + 2m_2 - 2$ ile sınırlıdır. □

Maksimum bağımsız kümeye polinom zamanlı ve $\Omega(\Delta\log(\Delta)/\log\log\Delta)$ oranını garantileyen bir yaklaşım algoritması vardır [39]. Burda Δ girdi çizgenin derecesini ifade eder. Yukardaki lemma ile bu sonucu birleştirirsek, kısıtlı hizalamanın en genel durumundan, yani m_1, m_2'nin herhangi pozitif tamsayı olma durumunda, bir yaklaşım algoritması elde ederiz:

Sonuç 4.4.2. *m_1, m_2 herhangi pozitif tamsayı için, kısıtlı hizalama problemi yaklaşım $\Omega((\Delta_1+\Delta_2)\log(\Delta_1+\Delta_2)/\log\log(\Delta_1+\Delta_2))$ olmak üzere polinom zamanlı bir yaklaşım algortimasına sahiptir.*

Sınırlı arama teknikleriyle (bounded search techniques) [25] $O(n(\Delta(G)+1)^k)$ zamanda boyutu k olan bir bağımsız kümenin G çizgesinde bulunduğu ya da böyle bir kümenin olmadığı sonucunun verildiği bilinmektedir. Fertin ve arkadaşları bu sonucu kısıtlı hizalamanın $m_2 = 1$ durumunda sınırlı derece çizgeler için sabit parametre kolay işlenir olduğunu göstermek için kullanırlar [29]. Aynı sonucu biz yukarda sunduğumuz lemma ile birlikte kullanırsak onların sonucunu aşağıdaki şekilde daha da geliştirebiliriz:

Sonuç 4.4.3. *G_1 ve G_2 sınırlı derece çizgelerse, m_1, m_2 herhangi pozitif tamsayı, k çıktı korunmuş ayrıt sayısı ve $D = O(\Delta_1+\Delta_2)$ için, kısıtlı hizalama problemi, parametre k olmak üzere, sabit parametre kolay işlenirdir ve $O(min(|E_1|,|E_2|)(D+1)^k)$ zamanda çözülebilir.*

Bölüm 5

Global Çoklu Ağ Hizalamaları[1]

Şimdiye kadarki hizalama problem tanımları bir ağ çiftine yönelikti. Bu bölümdeki hizalama çalışmalarında ise problem iki yönlü genelleştirilmiştir. Öncelikle, ikiden fazla ağ söz konusu olabilir. Ayrıca ağ hizalaması, 'çoklu' eşleştirmelere yöneliktir. Yani, herbir eşleştirme herhangi bir ağdan herhangi bir sayıda proteini içerebilen bir öbeğe karşılık gelecektir. Sonuçta oluşturulan her öbekteki protein grubunun fonksiyonel olarak ortolog olması beklenir. Bu en genel versiyonun öncekilere göre, hizalama probleminin biyolojik anlamı düşünüldüğünde, daha uygun olduğu görülebilir. Evrimsel moleküler biyolojik açıdan, üstünde çalışılan türlerin birbirine evrimsel uzaklıkları gayet farklı olabilir, ki bu da farklı türlerde aynı fonksiyon işlevi gören farklı sayıda proteinin olması gerçeğini beraberinde getirir.

Literatürde son yıllarda bu en genel ve dolayısıyla hesap karmaşıklığı en yüksek problem de çalışılmıştır [31, 57, 72]. Ancak bunlardan hiçbiri formel tanımlı optimizasyon amaçları açık belirtilmiş problem tanımı sunmamışlardır. Biz öncelikle global çoklu hizalama problemine formel bir tanım getireceğiz. Daha sonra problemi iki ana altprobleme böldüğümüz genel bir çerçeve geliştireceğiz. Bu altproblemler sırasıyla *omurga çıkarımı* ve *omurga birleştirme*dir. Basit bir ifadeyle her omurga, her ağdan en fazla bir protein içeren, yakın ilişkili merkezi protein gruplarına karşılık gelir. Tüö omurgalar belirlendikten sonra ikinci aşamada bir ortolog protein öbeğinde olma şansı yüksek omurgalar birbirleriyle birleştirilir. BEAMS isimli her iki aşama için de uygun buluşsallar içeren algoritmamızı sunacağız.

Takip eden altbölümlerde önce global çoklu hizalama problemi için formel bir tanım sunacağız. Ardından problemin hesapsal karmaşıklığını tartıaşacağız. Prob-

[1]Bu bölümde işlenen konular [6]'da yayınlanmıştır. Ayrıntılar için [6]'ya bakınız.

leme yönelik geliştirdiğimiz özgün BEAMS algoritmasını ayrıntılı işleyip, son olarak BEAMS'in hizalama performans kalitesini literatürde global çoklu hizalama problemi için önerilmiş alternatif algoritmalar IsoRankN ve SMETANA ile karşılaştıracağız.

5.1 Problem Tanımı

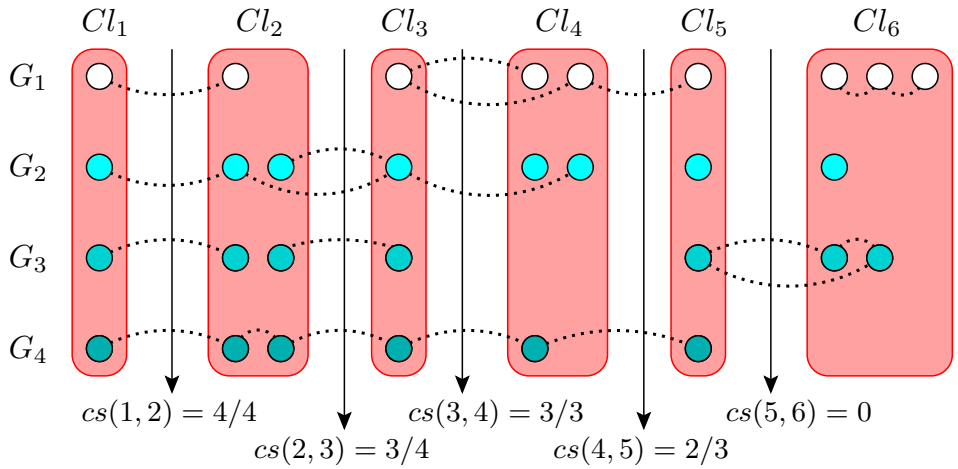
Farklı türlere ait verilmiş olan k adet protein ağı ve farklı ağlar arası proteinlerin dizilim benzerlik puanları için (genellikle BLAST bit skorları), çoklu hizalama problemi her biri başka bir fonksiyonel görevi temsil eden kesişmeyen protein kümeleri bulmayı amaçlar. Ağların hizalanması ile oluşan kümeler en az iki ağdan protein içermeli, bir protein sadece bir kümede yer alabilmeli ve mümkün olabildiğince çok protein çeşitli kümelere atanmış olmalıdır. Bu problem için daha önceden çeşitli algoritmalar geliştirilmiş, fakat hiçbir zaman problem için formel bir kombinatoryel tanım yapılmamıştır. Sunacağımız tanım global bire-bir ağ hizalama probleminin, problem doğasına uygun bir şekilde doğal bir genişletilmesinden oluşur.

Bu kısımda k adet protein ağının global çoklu (many-to-many) hizalanması problemi bir optimizasyon problemi olarak tanımlanmış ve optimizasyon hedefi şu şekilde belirlenmiştir. Verili $G_1(V_1,E_1)$, $G_2(V_2,E_2)$, ..., $G_k(V_k,E_k)$ ağları için, G_i, i'nci ağı, V_i düğümleri yani proteinleri, E_i ayrıtlrı yani etkileşimleri betimlesin; S $k-$parçalı kenar ağırlıklı tam çizgesinin (complete k-partite graph) i'nci bölümünü V_i oluştursun ve her bir $e(u,v)$ kenarının ağırlık değeri, $w(u,v)$, de u ile v arasındaki BLAST bit skoru tarafından belirlensin. S_β, S ile aynı düğüm kümesinde tanımlı S'nin bir altçizgesi olsun. S_β, aslında sadece yüksek dizisel benzerlik sağlayan proteinler arasında ayrıtların tutulduğu benzerlik çizgesi S'nin filtrelenmiş bir versiyonudur. Belirli bir S_β için, global çoklu hizalama aşağıda tanımlı AS skorunu maksimize eden ve kesişmeyen *öbeklerden* oluşan $\mathcal{CL} = \{Cl_1, Cl_2, \ldots, Cl_m\}$ *maksimal* süperkümesini bulmayı amaçlar:

$$AS(\mathcal{CL}) = \alpha \times CIQ(\mathcal{CL}) + (1-\alpha) \times \frac{\sum_{\forall Cl_i \in \mathcal{CL}} ICQ(Cl_i)}{|\mathcal{CL}|} \tag{5.1}$$

Burada kullanılan $\alpha \in [0,1]$ topolojik benzerlik ve dizisel benzerliğin göreceli önemlerini değiştirmek için kullanılan bir dengeleme parametresidir. İfadede kullanılan herhangi bir Cl_i öbeği, S_β çizgesinin $1 < c \leq k$ olmak üzere $c-$parçalı bir tam alt çizgesini betimler ve $\mathcal{CL}$ süperkümesi S_β üzerinden yeni bir hizalama yaratılamadığı durumunda maksimal hale gelir. AS skorunu maksimize etmenin otomatik olarak çıktı öbeklerin maksimalliğini garantilemediğini belirtelim.

Denklemdeki $CIQ(\mathcal{CL})$ hizalama sonucunun kenarlar (etkileşimler) üzerindeki koruma kalitesini gösterir ve *öbekler etkileşim skoru* olarak adlandırılır. İki farklı Cl_m, Cl_n öbeği için, düğümleri bu iki ayrı öbek içinde olan ve farklı ağlardan gelebilen protein etkileşimlerinin kümesi E_{Cl_m,Cl_n} olsun; bu kümedeki her bir (u,v) ayrıtının



Şekil 5.1: Conservation scores on a sample alignment covering all notable cases. Rectangular groups represent the clusters of the alignment. The dotted edges represent the protein-protein interactions. Proteins of each PPI network are drawn at separate horizontal layers. The *CIQ* score for this alignment is $(4 \times 4/4 + 4 \times 3/4 + 4 \times 3/3 + 2 \times 2/3 + 0)/16 = 0.771$. Note that since no other PPI edges exist between any other pair of clusters, only the indicated *cs* scores contribute to *CIQ*.

korunma skoru $cs(u,v)$, E_{Cl_m,Cl_n} kümesinin temsil ettiği ağ sayısı $s'_{m,n}$'in, Cl_m, Cl_n öbeklerinin her ikisinde de temsil edilen ağ sayısı olan $s_{m,n}$'ye bölümüdür. Fakat $s'_{m,n}$'in bire eşit olduğu durumlarda bu kenar hiç korunmamış olarak kabul edilir ve korunma skoru 0 olarak alınır. Bu durumda $CIQ(\mathcal{CL})$ aşağıdaki denklemle ifade edilir:

$$CIQ(\mathcal{CL}) = \frac{\sum_{\forall Cl_m,Cl_n} |E_{Cl_m,Cl_n}| \times cs(m,n)}{\sum_{\forall Cl_m,Cl_n} |E_{Cl_m,Cl_n}|} \tag{5.2}$$

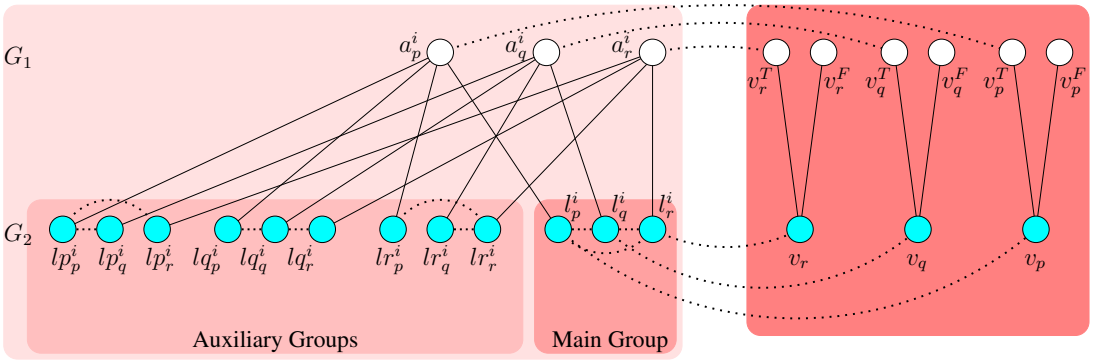
Denklem 5.1'deki $ICQ(Cl_i)$, Cl_i öbeğine ait proteinlerin sıra dizilim benzerlikleri üzerinden bu kümenin iç benzerlik kalitesini gösterir. $E(Cl_i)$, Cl_i düğümlerine dokunan S_β ayrıtlarının kümesi ve $w_{max}(u)$ da S_β'da u'ya dokunan ayrıtların en yüksek ağırlığı olsun. $ICQ(Cl_i)$ şöyle tanımlanır:

$$ICQ(Cl_i) = \frac{\sum_{\forall (u,v) \in E(Cl_i)} \sqrt{\frac{w(u,v)^2}{w_{max}(u) \times w_{max}(v)}}}{|E(Cl_i)|} \tag{5.3}$$

5.2 Problemin Hesapsal Karmaşıklığı

Problemin hesapsal kompleksitesi bağlamında elde edilen sonuç aşağıdaki teoremle ifade edilir:

Teorem 5.2.1. *Global çoklu ağ hizalama problemi, sıfırdan farklı tüm mümkün* α *değerleri için, iki ağın hizalandığı ve tüm benzerlik skorlarının aynı olduğu kısıtlı bir durumda bile NP-zordur.*



Şekil 5.2: Global çoklu hizalamanın NP-zorluk ispatında kullanılan araç.

İspat. Sunacağımız ispat Monotone1in3SAT probleminden bir indirgemeye dayanır ve bu problem 3SAT probleminin kısıtlanmış bir versiyonudur. 3SAT problemi verili bir 3'lü önermeler süperkümesi $((p_x, p_y, p_z); (p'_y, p'_k, p_l); \ldots)$ ve bu üçlü önermelerde kullanılan farklı değişkenler (önermeler) $(p_x, p'_x, p_y, p'_y, \ldots)$ için, bu kümelerin herbirinde en az bir doğru olacak şekilde tüm değişkenlere geçerli bir atama yapılıp yapılamayacağı problemidir. Monotone 1in3SAT problemindeki kısıt ise verilmiş olan kümedeki her bir üçlüde, üçlüyü oluşturan önermelerinin hiçbirinin zıttının alınmamış olması ve her üçlüde sadece bir önermenin doğru olarak atanması gerekliliğidir. İlk olarak, verili bir Monotone 1in3SAT girdisine karşılık global çoklu hizalama problemine girdi G_1, G_2 etkileşim çizgeleri ve S_β fonksiyonel benzerlik çizgelerinin nasıl oluşturuldukları gösterilecektir.

Formel indirgemede kullanılan değişken ve üçlü araçları (gadget) Şekil 5.2'de özet olarak görülebilir. Özet şekilde bir üçlü önerme grubu c_i'ye karşılık gelen üçlü aracı ve etkileşimde bulunduğu değişken araçları tasvir edilmiştir. Buradaki i üçlünün etiketi p, q ve r de üçlü içindeki değişkenleri betimlemektedir. Not etmeliyiz ki diğer üçlüler için oluşturulan üçlü araçları da p, q ve r için örnek gösterilen aynı değişken araçları ile etkileşimde bulunabilirler. Şekilde açıklık sağlamak için belirtilmemiş olsa da üçlü aracındaki her bir "auxiliary" grup içerisindeki lp düğümü diğer auxiliary gruplar içerisindeki lq ve lr düğümleriyle, lq düğümü lp ve lr düğümleriyle, lr düğümü de lp ve lr düğümleriyle etkileşmektedir. S_β 'de hangi benzerliklerin olacağıda figürde açıkça gösterilmiş ve tüm benzerliklere 1 atanmaktadır. Burada yapılan indirgeme ile herhangi bir Monotone-1in3SAT girdisine geçerli atama yapılmasının ancak ve ancak buna karşılık oluşturulan G_1 ve G_2 ağlarının S_β benzerlik çizgesine göre çoklu hizalanması sonucunda hizalamanın AS skorunun maksimum yani 1 olması halinde mümkün olduğu gösterilecektir. İlk olarak geçerli atama yapılabilen bir Monotone-1in3SAT girdisi için böyle bir hizalamanın mümkünlüğünü ispatlayalım. Geçerli girdi için doğru olarak atanan p önermelerine karşılık değişken araçlarında V_p ile V_p^T, yanlış olarak atanan r önermeleri için de V_r ile V_r^F'nin hizalanması seçilir. Herhangi bir üçlü

$c_i(p,q,r)$ için ise mesela p doğru atanmış sayalım. Bu durumda a_p^i ile l_p^i, a_q^i ile lp_q^i, a_r^i ile de lp_r^i hizalaması seçilir. Böyle bir durumda G_1'in tüm düğümleri hizalandığı için hizalama kümesi maksimal hale gelir ve uygun bir hizalama yapılmış olur. Bu yöntemle hizalamalar yapıldığı sürece şekilden de açıkça görüleceği üzere tüm öbekler arası etkileşimler korunmuş olur. Üçlü aracı içinde hiç korunması gereken etkileşim olmayacak sadece doğru olarak atanan p'nin hizalaması ile aradaki etkileşim (a_p^i ve V_p^T) ve (l_p^i ve V_p) etkileşimleri ile karşılıklı olarak korunacaktır. S_β çizgesindeki tüm benzerlik değerleri de aynı olduğu için bu durumda nihai AS skoru 1 olur. Diğer yönden kanıt için ise AS skoru 1 olan G_1, G_2 hizalamasından Monotone-1in3SAT girdisine geçerli atama yapılabileceği kanıtlanacaktır. Herhangi bir atamanın maksimal olması için üçlü aracı içindeki tüm G1 düğümleri hiç ortak benzerlik taşımadıklarından herbiri farklı bir öbek oluşturacaktır. Aynı durum değişken araçlarındaki G_2 düğümleri içinde söylenebilir. Ayrıca AS skorunun 1 olması demek tüm öbekler arası etkileşimin korunması anlamına gelmektedir. AS skoru 1 olan maksimal bir hizalamada şu açık olmalıdırki her bir üçlü aracı içerisinden bir adet ana grup ve bu ana gruba bağlı olarak 2 adet auxiliary grup hizalaması yapılması gerekir çünkü diğer durumlarda üçlü aracı içerisindeki öbeklerde korunmamış etkileşimler kalacaktır. Monotone-1in3SAT ataması bu noktada başlar. Herhangi bir üçlü için $c_i = (p,q,r)$ eğer p ana gruptan hizalanmışsa ona doğru ve diğerlerine yanlış atanır. Bu atamanın yapılabileceği de şöyle kanıtlanır. a_p^i ve l_p^i beraber öbek oluşturduğu için l_p^i ve V_p arasındaki etkileşim sadece V_p'nin V_p^T ile öbek oluşturmasıyla sağlanır. Bu öbeğin var olması ve tüm etkileşimlerin korunuyor olması da p'yi içeren diğer bütün üçlü araçlarında p'nin ana grupla öbek oluşturmasını mecbur kılar. Bu durum her bir üçlüye bir doğru atanmasını sağlar. Ayrıca bahsettiğimiz ilk üçlüde q ve r'nin p'nin ana gruptan öbeklenmesi ve etkileşimlerin korunması için oluşturdukları (a_q^i, lp_q^i), (a_r^i, lp_r^i) öbeklerin korunmamış etkileşime yol açmamaları için değişken araçlarında (V_q, V_q^F), (V_r, V_r^F) öbeklemesini mecbur kılar. Bu da q ve r'nin hiçbir üçlü içerisinde ana gruptan öbeklenmemesini sağlar. Buradan görüldüğü gibi bu indirgemeyle global çoklu hizalama probleminin aynı Monotone-1in3SAT problemi gibi NP-zor olduğu kanıtlanmıştır. □

5.3 BEAMS Global Çoklu Ağ Hizalama Algoritması

Ele almış olduğumuz hizalama probleminin NP-zor olması, problemin çözümü için akıllı buluşsal algoritmaların tasarımını gerekli kılmaktadır. Tasarlamış olduğumuz BEAMS buluşsal algoritmasının temeli diğer birçok hizalama algoritmasında da var olan tohum bulma ve büyütme konseptine dayanmaktadır. Fakat BEAMS algoritmasınca yapılmış olan tohumların bulunuş ve büyütülüş tanımlamaları literatürden yenilikçi farklılıklar göstererek algoritmanın daha yüksek performansta çalışmasını

sağlamaktadır. Algoritma tanımına geçmeden önce ilk olarak herhangi bir Cl_i öbeği ile ilgili şu gözlemden bahsetmek gerekir. Aslında her biri birer $c-$parçalı tam çizge olan herhangi bir Cl_i öbeği, en az n tane c'den küçük boyutlu tam çizgeye ayrılabilir ve bu n, Cl_i içersindeki parçalardan en büyüğünün düğüm sayısına eşittir. Buradan yola çıkarak, hizalama problemi iki altprobleme bölünebilmektedir. Birincil problem Cl_i'leri oluşturacak olan kesişmeyen tam çizgelerin (omurgaların) AS skorunu maksimize eden minimal kümesini bulmayı amaçlarken, ikincil problem ise varolan omurgalar kümesi üzerinde AS skorunu maksimize edecek olan minimal birleşmeler kümesini bulmayı amaçlar. Buradaki birleşme geçerli bir Cl_i öbeği oluşturabilen tam çizge kümesi anlamına gelmektedir. Her iki problemdeki kümelerin minimalliği ise, kümeler içerisindeki herhangi iki elemanın birleşerek geçerli bir küme elemanı yaratamaması anlamına gelir. Buradaki birincil problem, aslında global çoklu hizalamalarda birebir hizalama problemine de denk düşer. Oluşturulan algoritma temelde şu üç ana parçadan oluşur: S_β'nın oluşturulması, omurgaların çıkarsanması ve omurgaların birleştirilmesi. Çoka-çok global çoklu hizalama probleminin kendisi NP-zor olduğu gibi, onu ayırmış olduğumuz iki alt problemin de NP-zor olduğunu kanıtlarız:

Teorem 5.3.1. $\alpha \neq 0$*'nun bütün değerleri için, omurga çıkarsama problemi sadece bir çift ağ ve bütün* S_β *ayrık ağırlıklarının eşit olması durumunda bile NP-tamdır.*

İspat. Bakınız Ek4. □

Teorem 5.3.2. $\alpha \neq 0$*'nun bütün değerleri için, omurga çıkarsama problemi sadece bir çift ağ, bütün omurgalar 2-klik ve bütün* S_β *ayrık ağırlıklarının eşit olması durumunda bile NP-tamdır.*

İspat. Bakınız Ek4. □

5.3.1 S_β'nın Oluşturulması

Girdi olarak alınan ağların büyüklükleri nedeniyle, hizalama probleminin çözümü için oluşturulan S benzerlik çizgesi çok büyük bir çizgeye denk gelmektedir. Elimizde bulunan bu gereksiz büyüklükteki çizge hem tasarlanacak olan algoritmaların daha çok işlem yapmasına sebep olacak hem de ulaşılmak istenen fonksiyonel ortolog gruplarda hatalara yol açacaktır. Farklı ağlardan gelen iki protein arasındaki çok ufak benzerlik değerleri fonksiyonel benzerlik açısından bir önem taşımadığından bunların filtrelenmesi gerekmektedir. Ele alınan ağlar arasındaki evrimsel mesafeler birbirlerinden farklı olduğundan, filtreleme işlemini bir bağıl filtre yardımıyla gerçekleştiririz. Kullanıcı tarafından belirlenmiş olan bir β değeri için, filterelenmiş S_β çizgesi, $w(u,v) < \beta \times max(u,v)$ olan tüm (u,v) ayrıtlarının çizgeden silinmesiyle oluşturulur. Burada $max(u,v)$, u',v' sırasıyla u ve v'nin bulunduğu ağlardan birer düğüm şartıyla, $w(u,v')$ ve $w(u',v)$ olası değerlerinden en büyük olanını ifade eder.

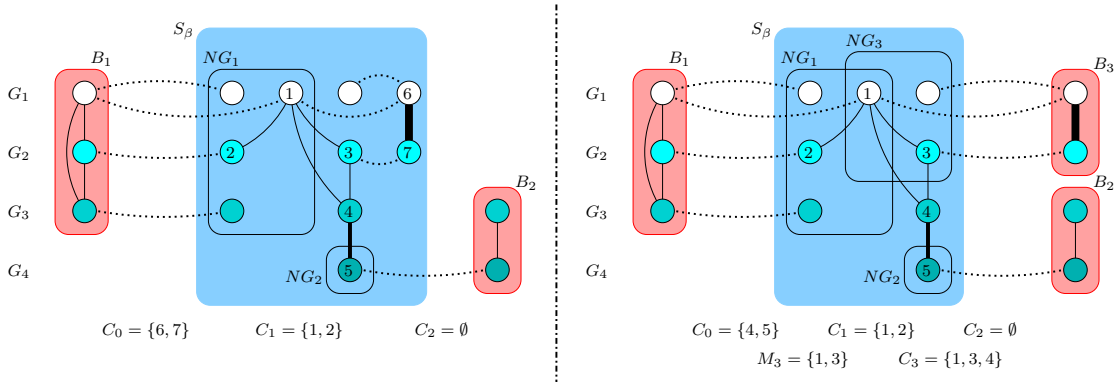
Algorithm 2 *OMURGA_ÇIKARSANMASI*

Input: $S_\beta, G_1, G_2, \ldots, G_k, \alpha$
Output: Set of backbones $B = \{B_1, B_2, \ldots, B_n\}$
$B = \emptyset; C = \emptyset$
`//Initial candidate`
$C_0 = MEWC(S_\beta); C = C \cup \{C_0\}$
repeat
    $B_{new} = Select\_Cand(C, B); B = B \cup \{B_{new}\}$
    Remove B_{new} from S_β
    `//Generate new candidate`
    $C_{new} = Generate\_Cand(S_\beta, B_{new}); C = C \cup \{C_{new}\}$
    `//Update each candidate in C`
    for all $C_i \in C$ **do**
        if $C_i \cap B_{new} \neq \emptyset$ **then**
            if $i == 0$ **then**
                $C_0 = MEWC(S_\beta)$
            else
                $C_i = Generate\_Cand(S_\beta, B_i)$
            end if
        end if
    end for
until S_β contains only isolated nodes
`//Each isolated node is a backbone itself`
for all nodes $u \in S_\beta$ **do**
    $B_{new} = \{u\}; B = B \cup \{B_{new}\}$
end for

5.3.2 Omurgaların Çıkarsanması

Omurga oluşturma problemi NP-zor olduğundan ötürü problemin çözümü için polinom zamanda çalışan iteratif bir buluşsal algoritma tasarlanmıştır. Tasarlanan algoritma için kod Algoritma 2'de sunulmuştur. Algoritmamız temelde üç konsepte dayanmaktadır: Alt çizge olarak en ağır tam çizgeyi bulmak (MEWC), komşuluk çizgesi üzerinden aday üretme ve AS skoru maksimizasyonu için buluşsal aday seçilimi. Buradaki MEWC problemi, ağırlıklı bir çizge içerisindeki, kenar ağırlıkları toplamı en yüksek olan tam çizge (clique) alt çizgesini bulma problemidir.

Algoritma başlangıçta boş bir omurga kümesi ve tek elemanlı bir aday omurga kümesi ile başlar ve bu tek aday S_β'nın MEWC'i bulunarak oluşturulur. İterasyon j'ye gelindiğinde, algoritma 4 ana adımdan geçer. O andaki j adet aday içerisinden en iyi aday buluşsal olarak seçilir ve omurga kabul edilir, bu aday S_β'dan silinir ve omurga kümesine eklenir, bu omurganın komşuluk çizgesi üzerinden yeni bir aday oluşturulur ve son olarak da bütün aday omurgalar güncellenir. Teker teker inceleyecek olursak, adaylar içerisinden en iyi seçilimi, şu ana kadar bulunmuş olan omur-



Şekil 5.3: BEAMS algoritmasında omurga oluşturma.

galar ile oluşturacakları hizalamanın *AS* skoru üzerinden yapılır ve en yüksek olası skora ulaşan en iyi kabul edilir. En iyi aday bize o turda bir omurga verdiği için daha sonra yaşanacak kesişmelerin engellenmesi için çizgeden silinir. Üçüncü adım olan aday yaratma adımı ise, adayların varolan omurgaların S_β'da kalmış olan gerçek ağ komşuları üzerinden yapılır, çünkü bu sayede tüm adayların en azından bir omurgayla arasında bulunan tüm ayrıtların korunması sağlanır. Tüm adayların her iterasyonda güncellenmesinin sebebi ise adayların birbirleriyle örtüşebilmesi ve de bu yüzden bazı adayların seçilmedikleri iterasyonlarda bile elemanlarını kısmen kaybetmesidir. Böyle durumlarda adaylar bağlı oldukları omurgalar üzerinden tekrar hesaplanıp güncellenir. Şekil 5.3'de bir örnek üzerinde üçüncü iterasyonun öncesi ve sonrası için S_β, komşuluk çizgeleri, omurgalar ve adaylar görülebilmektedir. Şekilde noktalı ayrıtlar protein etkileşimlerini gösterir. Her ağ ayrı bir yatay sırada gösterilmiştir. Farklı sıralar arası ayrıtlar S_β ayrıtlarıdır. Soldaki şekil için, omurgalar B_1, B_2 ile birlikte düşünüldüğünde C_0'ın *AS* skorunun, C_1'in aynı skorundan daha yüksek olduğunu varsayarsak, C_0 yeni yaratılmış omurga olur. Sağdaki şekil için, önce B_3, S_β'dan çıkarılır. Yeni aday C_3'ü yaratmak için B_3'ün komşuluk çizgesi NG_3 ve NG_3'ün MEWC'i M_3 oluşturulur. M_3'ün S_β'daki G-MEWC'i yeni aday C_3 olur. Son olarak, B_3 ile düğüm paylaşan tek aday olan C_0 yenilenir. S_β'nın MEWC'inin ayrıt $(4, 5)$'den olştuğunu varsayarsak, bu yenilenmiş C_0 olur.

5.3.3 En Yüksek Ayrıt Ağırlıklı Alt Tam Çizgenin Bulunması

MEWC isimli bu problemin çözümü için BEAMS algoritması bünyesinde, yeni bir dal-sınır (branch-and-bound) algoritması başarı ile tasarlanmış ve BEAMS algoritmasına entegre edilmiştir. Bu algoritmanın detayları için bakınız Ek4.

5.3.4 Omurgaların Birleştirilmesi

Bu NP-zor problem için tasarlanmış olan buluşsal iteratif algoritma BEAMS algoritmasının ikinci temel kısmını oluşturur. Verili bir B omurga kümesi kopyalanarak MB birleştirmeler kümesi oluşturulur ve bu kümenin her elemanı geçerli bir omurga birleşmesini simgeler. Buradaki geçerlilik birleşme içindeki omurgaların c-parçalı bir tam çizgeyi oluşturabilmesi anlamına gelmektedir. Bu MB listesi her iterasyonda omurga oluşturmada kullanılan aday yaratma ve seçme yöntemiyle güncellenir ve döngü yeni aday yaratılamayana kadar devam eder. Her iterasyonda MB içerisindeki birleşmelerden, birbirlerine katılarak yeni bir birleşme oluşturabilen tüm çiftler aday olarak kabul edilir ve bu çift katılımı sonucunda MB'nin ulaşacağı AS skoru en yüksek olan çift en iyi aday olarak seçilir. Bu çift, tek birleşme olacak şekilde MB güncellenir ve diğer iterasyona geçilir. Algoritmanın kodlama ve daha ayrıntılı iç detayları için bakınız Ek4.

5.4 Karşılaştırmalı Deneysel Sonuçlar

BEAMS algoritması LEDA kütüphanesinin de [61] yardımıyla C++ ile kodlanmış ve literatürde bulunan iki adet algoritma ile karşılaştırılmıştır: SMETANA ve IsoRankN. Bu algoritmalar da çoka-çok global çoklu hizalama problemine çözüm olarak sunulmuşlar ve kendilerinden önceki diğer tüm algoritmalardan biyolojik olarak daha tutarlı sonuçlar elde etmişlerdir [57, 71]. IsoRankN ve BEAMS algoritmaları problem tanımlarında benzerlikler gösterdiğinden her ikisi için de birçok farklı α değerlerindeki sonuçlar karşılaştırılmış ve bu sonuçlar için BEAMS algoritması içindeki β değeri 0.4 olarak sabitlenmiştir. SMETANA tarafından kullanılan kullanıcı parametreleri ise ilgili makalesinde kullanılmakta olan değerlerce belirlenmiştir.

BEAMS algoritması için yapılan karşılaştırmalı testlerde hem gerçek hem de sentetik protein etkileşim ağları kullanılmıştır. Burada ise özellikle gerçek protein etkileşim ağları üzerinde elde edilen sonuçlar sunulmuştur. Girdi olarak kullanılacak olan gerçek ağlar ve protein benzerlik BLAST skorları IsoBase [67] veritabanından alınmış ve PPE ağları kullanılacak olan 5 tür C. Elegans, D. Melanogaster, H. Sapiens, M. Musculus ve S. Cerevisiae olarak seçilmiştir. Ek karşılaştırmalar için kullanılan sentetik ağlar ise NAPAbench veritabanından elde edilmiştir [71]; bakınız Ek1. Bir sonraki bölümde elde edilen sonuçların ilk olarak nicel analizi ve daha sonra nitel karşılaştırmaları sunulmaktadır.

5.4.1 Çıktı Öbeklerinin Nicel Analizi

Tablo 6.2'de algoritmalarca elde edilen öbeklerin genel analizlerinin yanında, detaylı analizler için öbeklerin içerdikleri ağ sayısına göre gruplanmış analizleri de verilmek-

Tablo 5.1: Çıktı Öbeklerin Nicel Analizi

	BEAMS			IsoRankN			SMETANA
	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.7$	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.7$	
$c = 2$	**7251**	7242	7245	0	0	0	6104
	20540	20419	20392	0	0	0	14956
$c = 3$	3259	3277	3277	**4717**	4708	4699	2808
	12089	12259	12204	**15891**	15827	15807	10941
$c = 4$	3281	3283	**3291**	3058	3036	3040	3180
	16254	16311	16450	14651	14540	14550	**18189**
$c = 5$	2090	2081	2074	2099	2104	2083	**2412**
	13117	13012	12940	12834	12868	12697	**19158**
Toplam Kapsam	15881	15883	**15887**	9874	9848	9822	14504
	62000	62001	61986	43376	43235	43054	**63244**
Etkileşimler	7060	7425	7407	5978	6024	5766	**13498**
	114889	114323	114306	109364	108374	106642	**122450**
	6.15%	6.49%	6.48%	5.47%	5.56%	5.41%	**11.02%**
AS	**0.5261**	0.3860	0.2455	0.3970	0.2932	0.1882	0.4766

tedir. Bu analiz değerleri için ilk beş satırdaki değerlerin üstte olanı üretilen öbek sayısını, alttaki değerler ise öbeklerdeki toplam protein sayısını verir. İlk dört çoklu satır için c incelenen öbeklerdeki ağ sayısı olmak üzere, $c = 2,3,4,5$ durumları için sonuçları verir. Burada görüldüğü üzere BEAMS ve SMETANA algoritmalarının kapsamları birbirlerine yakın olmakla beraber, IsoRankN algoritmasından oldukça üstün seviyelerdedir. Kapsamın yüksek olması algoritmaların daha fazla veri açıklayabilme yetilerini göstermektedir. *Etkileşimler* çoklu satırındaki ilk satır algoritma tarafından korunan ayrıt sayısını, ikinci satır öbekler arası toplam ayrıt sayısını ve son satır ise aralarındaki oranı vermektedir. SMETANA algoritması her ne kadar BEAMS algoritmasından üstün görünse de korunan ayrıtların özellikle tutarlı öbekler arasında olması daha önemlidir ve bu özellik bir sonraki bölümde ele alınmıştır. Son satırdaki *AS* değeri ise hizalamanın toplam *AS* skorunu göstermektedir ve bu maksimizasyon hedefine en çok yaklaşan algoritmanın ise BEAMS olduğu görülmektedir.

5.4.2 Biyolojik Tutarlılık Analizleri

Diğer birçok hizalama algoritmasının tutarlılık analizlerinde kullanıldığı gibi biz de burada hiyerarşik gen ontoloji veritabanından (GO) yararlanmaktayız. Burada elde edilen öbekler için, birçok proteinin faklı seviyelerde olmak üzere belirlenmiş olan GO etiketleri üzerinden sonuçların biyolojik olarak uyumlu olup olmadıkları test edilmiştir. Tüm testler öncesinde standartlaşmayı sağlamak için proteinlerin sadece beşinci seviyeye karşılık gelen etiketleri kullanılmış ve bunun altındaki tüm etiketler yok sayılmıştır. Bir öbek, içerisinde en az 2 adet GO anotasyonuna sahip protein içerdiği takdirde *anote edilmiş* sayılmış ve de öbek içerisindeki tüm anotasyonlu proteinler ortak bir GO anotasyonuna sahip oldukları zaman öbek *tutarlı* sayılmıştır. Tablo 5.2'de BEAMS,

Tablo 5.2: Biolojik tutarlılık analizleri.

	BEAMS			IsoRankN			SMETANA
	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.7$	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.7$	
$c = 2$	**2150**	2147	2132	0	0	0	1593
	1997	**1997**	1985	0	0	0	1489
	92.9%	93.0%	93.1%	-	-	-	**93.5%**
$c = 3$	1791	1792	1784	2523	2524	2524	1497
	1478	1479	1466	1926	1938	1943	1179
	82.5%	**82.5%**	82.2%	76.3%	76.8%	77.0%	78.8%
$c = 4$	2497	2499	**2517**	2275	2253	2255	2208
	1843	1840	**1853**	1616	1608	1601	1436
	73.8%	73.6%	73.6%	71.0%	71.4%	71.0%	65.0%
$c = 5$	1971	1961	1954	1958	1963	1943	**2233**
	1375	**1384**	1371	1309	1305	1298	1346
	69.8%	**70.6%**	70.2%	66.9%	66.5%	66.8%	60.3%
Toplam	**8409**	8399	8387	6756	6740	6722	7531
	6693	**6700**	6675	4851	4851	4842	5450
	79.59	**79.77**	79.59	71.8	71.97	72.03	72.37
Duyarlılık	0.3780	**0.3791**	0.3783	0.3203	0.3199	0.3198	0.3606
DD	22231	**22304**	22218	16350	16334	16301	20227
	71.1%	**71.4%**	71.1%	67.2%	67.3%	67.3%	64.1%
GD_1	11397	11425	11406	3382	3310	3350	-
GD_2	6979	**7056**	6949	-	-	-	5325
MNE	**1.2881**	1.2902	1.2899	1.4685	1.4672	1.4672	1.3943
NGOC	0.3093	0.3086	**0.3096**	0.2413	0.2424	0.2422	0.2471
COI	3331	**3590**	3491	2374	2359	2335	2694

IsoRankN ve SMETANA algoritmalarının üretmiş olduğu öbeklerin biyolojik tutarlılık analizleri ilk 5 satırda sunulmuştur. Bu satırlarda, en üstteki değer anote edilmiş öbek sayısını, orta değer tutarlı öbek sayısını, son değer ise tutarlılık oranını vermektedir. Bu oarnın bazı çalışmalartda 'spesifisite' olarak anıldığını belirtelim. Oluşturulan tüm öbekler göze alındığında BEAMS algoritmasının açık bir şekilde daha çok tutarlı öbek oluşturduğu görülmekte ve BEAMS öbeklerinin IsoRankN ve SMETANA öbeklerinden daha spesifik olduğu açıkça görülmektedir.

Elde edilen öbeklerin ne kadar duyarlı oldukları ise Flannick ve arkadaşları'nın belirttiği tanım göz önüne alınarak hesaplanmıştır [31]. Buna göre verili bir GO kategorisi için, onun *en yakın öbeği*, o kategori ile anote olmuş en fazla protein içeren öbek olsun. Bir hizalamanın *duyarlılığı* o halde, en yakın öbekte de olan bir GO kategorisi ile anote olmuş hizalanmış düğümlerin oranının bütün GO kategorileri üstünden ortalamasıdır. BEAMS algoritması en yüksek duyarlılığa sahip öbekleri oluşturmuştur. Diğer bir duyarlılık değeri olan *doğru düğümler* (DD) ise tutarlı öbeklerde hizalanmış toplam protein sayısını gösterir ve bu performans parametresi bakımından da yine aynı şekilde BEAMS hizalamaları en yüksek değerleri vermiştir. Bu duyarlılık değerlerine ek, çalışmamızda başka bir duyarlılık değeri tanımlanarak *göreceli duyarlılıklar* da hesaplanmıştır. Göreceli duyarlılık (GD) bir algoritma tarafından tutarlı öbeklenen

fakat diğeri tarafından tutarsızca öbeklenen protein sayısını vermektedir. GD_1 satırında BEAMS kolonu altındaki değer aynı α kullanıldığında BEAMS tarafından tutarlı bir öbeğe fakat IsoRankN ile tutarsız bir öbeğe atanan proteinlerin sayısını verirken IsoRankN kolonu altındaki değer tam tersini verir. GD_2 ise benzer şekilde BEAMS ile SMETANA göreceli değerlerini verir. Değerler incelendiğinde BEAMS algoritmasının göreceli olarak da diğerlerinden üstün olduğu görülmektedir.

Ortalama entropi değeri (MNE) öbeklerin genel biyolojik tutarlılık kontrolü için tanımlanmış diğer birçok çalışmada da kullanılan bir karşılaştırma değeridir [57, 71]. NGOC ise yine aynı şekilde bu proje kapsamında gerçekleştirilmiş olan SPINAL algoritmasınca belirlenmiş olan GOC biyolojik tutarlılık değerinin çoklu hizalamalar için genelleştirilmiş halidir. Her iki genel tutarlılık değerleri için BEAMS algoritmasının IsoRankN ve SMETANA'dan oldukça üstün olduğu kolayca anlaşılmaktadır.

Tüm bu öbek değerlendirmelerinin yanında bir de oluşturulan öbekler arasında kalan etkileşimler incelenmiştir. Tüm algoritmalar arasından BEAMS algoritmasının en yüksek korunmuş tutarlı etkileşim sayısına (COI) ulaştığı görülmektedir. Bu analiz sayesinde, bir önceki nicel değerlendirmeler içerisinde SMETANA'nın verdiği yüksek korunmuş etkileşim sayısının aslında çok da anlamlı olmadığı görülmüştür. Ayrıntılı karşılaştırmalı değerlendirmelere Ek4'den baıklabilir.

Bölüm 6

Eşzamanlı Ağ Çıkarımı ve Global Ağ Hizalamaları[1]

PPE ağlarının çıkarımı, yani bir türün bilinen proteinleri arasındaki etkileşimlerin bulunması biyoenformatik alanında önemli problemlerden biridir. Deneysel etkileşim çıkarımı genellikle zaman alıcı ve pahalıdır. Bu yüzden geniş bir hesapsal teknikler yelpazesinden pekçok hesapsal yöntem bu amaçlı önerilmiştir [60, 37, 3]. Etkileşim çıkarımı için kullanılan bilgi tipine göre hesapsal yöntemler farklılaşır. Bunlar arasında genom içeriği, yapı ya da dizi bilgisi kullananlar vardır; faydalı tarama makaleleri için bakınız [85, 78, 81, 87]. Çıkarsanan ağlarla ilgili önemli bir problem, çıkarsanan etkileşimlerde yanlış pozitif ve yanlış negatifler bağlamında hatalı sonuçlar olmasıdır. Deneysel teknikler düşünüldüğünde bu daha çok yüksek dereceli veri gürültüsünden kaynaklanırken, hesapsal teknikler, hem kullanılan buluşsalların tanımlı problemleri optimal çözmemelerinden, hem de kullanılan gürültülü veriden dolayı sorunlar yaşarlar. Bu nedenle literatürde sadece ağ topolojisine bağlı pekçok *ağ yeniyapım* algoritması önerilmiştir [56, 16]. Bu problemde verili bir etkileşim ağında düğüm komşuluklarına bakılarak etkileşim ekleme çıkarmalarıyla ağ daha güvenilir hale getirip yeni bir ağ çıkarsanmaya çalışılır.

Aslında ağ çıkarım ve ağ hizalama problemleri doğaları gereği içiçe geçmiştir. Hizalama yöntemleri çıkarsanmış etkileşim ağlarını kullanarak yüksek etkileşim korunumlu ve dizisel benzerlikler veren çıktı hizalamalar üretir. Öte yandan hizalamalar sayesinde elde edilen ortoloji bilgisi ise yeni etkileşim tahmini veya var olan etkileşim yanlışlaması için kullanılabilir [55, 45, 66]. Bu yüzden ortaya çıkan bu yumurta-tavuk benzeri ilişkiyi çözümlemek amacıyla problemleri bağımsız olarak ele almak yerine

[1]Bu bölümde işlenen sonuçların *Bioinformatics* dergisinde yayını revizyon sonrası kabul aşamasındadır. Ayrıntılar için Ek 5'e bakınız.

Algorithm 3 *Eşzamanlı Çıkarım ve Hizalama Çerçevesi*

Input: $G_1(V_1,E_1), G_2(V_2,E_2), BL(V_1,V_2), k$
Output: Updated G_1, G_2, Alignment A
$A = Alignment(G_1, G_2, BL(V_1,V_2))$
for iteration = 1 to k **do**
    $T_1 = Topological\_Similarity(G_1)$
    $T_2 = Topological\_Similarity(G_2)$
    $\prec G_1, G_2, A \succ = SiPAN(G_1, G_2, A, T_1, T_2, BL(V_1,V_2))$
end for

her iki problemi eşzamanlı ele alan yeni bir problem tanımı, *eşzamanlı ağ çıkarımı ve ağ hizalaması* (simultanesou prediction and alignment of networks, SiPAN) öneririz. bilgimiz dahilinde bu, ağ yeniyapımı ve hizalamasını eşzamanlı ele alan bir model sunan ilk çalışmadır.

Takip eden altbölümlerde önce bir eşzamanlı ağ çıkarım ve hizalama çerçevesi tanımını yapacağız. Ardından, belirlenen çerçevede bu eşzamanlı probleme yönelik geliştirdiğimiz SiPAN algoritmasını işleyeceğiz. Son olarak hem ağ yeniyapımı hem de hizalama bağlamlarında literatürde önerilmiş alternatif yöntemlerle karşılaştırmalı başarım sınamalarını sunacağız.

6.1 Eşzamanlı Ağ Çıkarım ve Hizalama Çerçevesi

$G_1(V_1,E_1)$ ve $G_2(V_2,E_2)$ iki farklı türe ait PPE ağlarını temsil eden yönsüz çizgeler olsun. V_1, V_2 düğüm kümelerini, gösterirken E_1, E_2 de ayrıt kümelerini simgeler. Genel çerçeve kod olarak Algoritma 3'de tanımlanmıştır. İlk girdi etkileşim ağları G_1, G_2 ile başlarız. Algoritma önce girdi ağların bir hizalaması A'yı oluşturur. Sonra tekrarlı olarak topolojik benzerlik matrisleri T_1, T_2 oluşturulup, A, T_1, T_2 kullanılarak bizim önerdiğimiz eşzamanlı çıkarım ve hizamala algoritması SiPAN ile ağlar G_1, G_2 ve hizalama A yenilenir. Burda hizalama A, $u \in V_1, u' \in V_2$ olmak üzere bire-bir (u,u') eşleşmeler kümesidir. Hizalama elde etmek için Bölüm 1'de sunulan SPINAL global bire-bir ağ hizalama algoritmasını kullanırız. Yakın zamanlı bir tarama makalesi çıktı hizalamaların biyolojik anlamı açısından SPINAL'in en iyi performans gösteren ağ hizalayıcıları arasında olduğunu göstermiştir [21]. Genel SiPAN çerçevesi içinde kullanılan topolojik benzerlik bulma algoritması olarak da *Direnişli Rastgele Yürüme* (Random Walk with Resistance, RWS) algoritması kullaılmıştır [56]. Bu da topoloji tabanlı pekçok ağ tahmin/doğrulama algoritmasından biridir [83, 32, 54, 28]. RWS'in alternatiflerine oranla fonksiyonel ilişki bağlamında, gen ontoloji anotasyonları, gen ifade verisi, protein kompleksleri ve türlerarası korunum gibi çoklu bilgi kaynağı kullanan biyolojik önem parametreleri düşünüldüğünde daha doğru sonuçlar ürettiği bilinmektedir [56]. Verili bir çift PPE ağı için tüm topoloji tabanle ağ doğrulama algo-

Algorithm 4 *SiPAN Algoritması*

Input: $G_1(V_1,E_1), G_2(V_2,E_2), A, T_1, T_2, BL(V_1,V_2)$
Output: Updated $\prec G_1, G_2, A \succ$
for $x = 1,2$ **do**
    Construct *candidate* set C_x
    Sort C_x with respect to scores in T_x
end for
Compute *breakpoints* p_1, p_2
$Resolve\_Indels(C_1, C_2, p_1, p_2)$
// Update networks and the alignment
for $x = 1,2$ **do**
    Commit the best $\beta_x \times |D_x^{p_x}|$ deletions in $D_x^{p_x}$
    Commit the best $\beta_x \times |I_x^{p_x}|$ insertions in $I_x^{p_x}$
end for
$A = Alignment(G_1, G_2, BL(V_1,V_2))$

ritmaları bir topolojik benzerlik matrisi oluştururlar. Algoritma 3'deki T_1, RWS'den alınan $|V_1| \times |V_1|$ boyutlu reel değerlerden oluşan bir matristir. Matriste her (u,v) noktasındaki değer, u,v arasındaki etkileşimin varlığına dair hesaplanmış güvene karşılık gelir. T_2 de benzer şekilde G_2'nin düğüm kümesi V_2 üstünde tanımlıdır. SiPAN genel çerçevesinin iki temel adımı olan topolojik benzerlik matrisi oluşturma ve hizalama k kullanıcı tanımlı bir parametre olmak üzere, k kere tekrarlanır.

6.2 SiPAN Eşzamanlı Ağ Çıkarım ve Hizalama Algoritması

Önerilen yaklaşımın esas özgünlüğü SiPAN'ın o anki ağ topolojileri, verilen ağ hizalamaları ve de topolojik benzerlik skoru matrislerini kullanarak ağları nasıl *yeniyapıma* uğrattığıdır. Ana SiPAN algoritması Algorithmlar 4 ve 5'de sunulmuştur. Algoritmanın önemli adımlarının işleyişi Şekil 6.1'de bulunabilir.

6.2.1 Kaydadeğer Korunmamış Ayrıtlar

Verili bir eşleme $(u,u'),(v,v') \in A$ için, eğer $(u,v) \in E_1, (u',v') \notin E_2$ veya $(u,v) \notin E_1, (u',v') \in E_2$ eşleme *korunmamış ayrıt* üretiyor denir. Korunmamış birinci türden bir ayrıtı *çözümlemek* için ya E_1'den bir ayrıt silebilir ya da E_2'ye eksik olan bir ayrıtı ekleyebiliriz. İkinci tür korunmamış ayrıt da benzer şekillerde çözümlenebilir. Algoritmanın genel hedefi hizalama A kaynaklı tüm *kaydadeğer* korunmamış ayrıtları topolojik benzerlik skorları matrislerini dikkate alarak çözümlemektir.

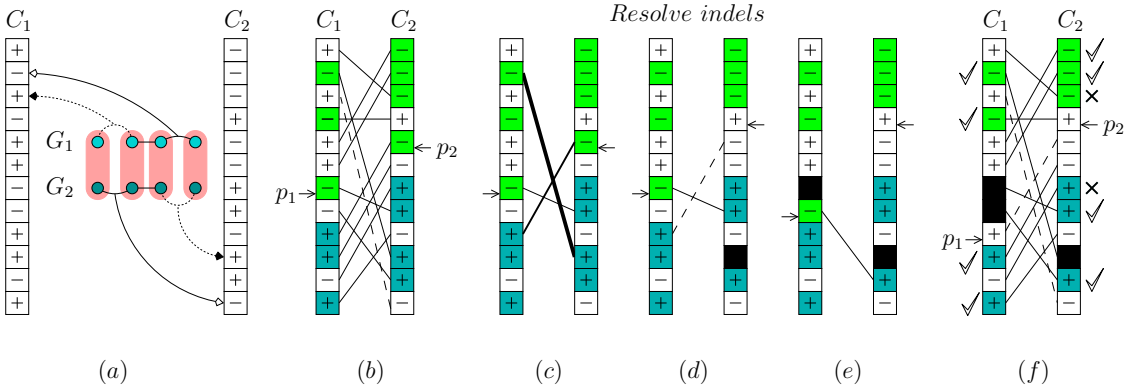
Önce *aday* kümeleri C_1, C_2 oluşturarak başlarız. herbir kümede o andaki hizalama A'ya göre aynı ağdan korunmamış bir ayrıta yol açan düğüm çiftlerini içerir. Spesifik olarak $x,y = 1,2$ ve $x,y = 2,1$ için C_x'i $\{(u,v) | (u,v) \notin E_x \wedge (A(u),A(v)) \in E_y\}$

Algorithm 5 *İndel_Çözümlemeleri*

Input: C_1, C_2, p_1, p_2
Output: Updated $\prec C_1, C_2, p_1, p_2 \succ$
Construct priority queue Q of all *indels*
while Q not empty **do**
    Remove $\prec (u,v),(u',v') \succ$ from Q
    if $\prec (u,v),(u',v') \succ$ not an *indel* **then**
        continue
    end if
    if $w(u,v) < w(u',v')$ **then**
        Remove (u',v') from C_2
        Recompute $p_2, I_2^{p_2}, D_2^{p_2}$
    else
        Remove (u,v) from C_1
        Recompute $p_1, I_1^{p_1}, D_1^{p_1}$
    end if
    Insert new *indels*, if any, to Q
end while

ve $\{(u,v) | (u,v) \in E_x \wedge (A(u),A(v)) \notin E_y\}$'nin birleşimi olarak tanımlarız. Birleşimdeki ilk küme G_x' olası etkileşim eklemelerini (insertion) gösterirken ikincisi G_x'ten olası etkileşim silmelerine karşılık gelir. Aslında bir ağın aday kümesi, diğer ağda hiçbir değişiklik yapılmazsa, korunmamış ayrıtları çözümlemek için ağda gerçekleştirilecek tüm işlemlerin kümesidir. Ancak iki ağda da yanlış pozitif/negatifler olabileceğinden bütün yenilemeleri tek bir ağ üzerinde gerçekleştirmek hatalı ağ yeniyapımlarına yol açar. Dolayısıyla esas zorluk bütünsel bir uygun yenilemeler kümesini her iki ağı eşzamanlı ele alarak oluşturmaktır. Baist ifadelerle amaç, C_1, C_2'nin her ikisinden, seçilen eklemelerin seçilen silmelerden topolojik benzerlik skorları bağlamında daha yüksek güvene sahip olduğu, bir işlemler altkümesi seçmektir.

Bu doğrultuda önce aday listelerini topolojik benzerlik skorlarına göre artan sırada sıralarız; bakınız Şekil 6.1-a. Sıralanmış her aday listesi için bir *kesim noktası* (breakpoint) buluruz. Kesim noktası her ağda olası ekleme/çıkarmaları öyle bir noktadan ayırır ki, gerçekleştirilen tüm silmelerin endeksleri kesim noktasından küçük eşit, tüm eklemelerin endeksleri de ondan büyük olsun. Bir endeks i için D_x^i, i'den küçük eşit endeksli bütün silmeler listesi, I_x^i de i'den büyük endeksli eklemeler listesi olsun. Verili her ağ için yanlış negatiflerin yanlış pozitiflere oranı farklı olabileceğinden kesim noktaları p_1, p_2'yi kullanıcı tanımlı parametreler α_1, α_2 ile belirleriz. $x = 1,2$ için p_x, $\alpha_x \times |D_x^{p_x}| = |I_x^{p_x}|$ eşitliğini sağlayan endeks olsun. Korunmamış ayrıt $(u,v) \in E_1, (u',v') \notin E_2$ veya $(u,v) \notin E_1, (u',v') \in E_2$ için eğer $(u,v) \in D_1^{p_1} \cup I_1^{p_1}$ veya $(u',v') \in D_2^{p_2} \cup I_2^{p_2}$ ise *kaydadeğer* denir. Bir başka ifadeyle kesim noktaları aday kümeler içinde istenmeyen ekleme/çıkarmaların sınırlarını belirlerler; en-



Şekil 6.1: SiPAN'ın temel adımları. *(a)* Kısmi G_1, G_2, o anki hizalama ve C_1, C_2. Hizalı düğümler elips içindedir. C_1, C_2 endeksleri yukardan aşağı doğru artan sıradadır. Eksiler silme, artılar ekleme işlemini gösterir. C_1, C_2'deki işlemler topolojik benzerliğe göre sıralanmıştır. *(b)* $\alpha_1 = \alpha_2 = 1$ için p_1, p_2. bu durumda $|D_1^{p_1}| = |I_1^{p_1}| = 3$, $|D_2^{p_2}| = |I_2^{p_2}| = 4$. $D_1^{p_1}, D_2^{p_2}$ açık yeşil $I_1^{p_1}, I_2^{p_2}$ koyu mavi ile renkidir.Korunmamış ayrıtlar C_1, C_2 arasındaki çizgilerle gösterilir. Aralık çizgili ile gösterilen dışında tüm korunmamış ayrıtlar kaydadeğerdir. *(c)* Indeller çizgilerle gösterilmiştir. Çizgi kalınlığı indel önceliğini gösterir. En kalın çizgili indein ağırlığı $\frac{2}{12} \times \frac{3}{12} = \frac{1}{2}$,ki bu da en öncelikli olandır. *(d)* İşlenen indel için silmenin ağırlığı eklemeden küçüktür; $\frac{1}{6}$2a karşılık $\frac{1}{4}$. Ekleme $C_2$2den çıkarılır ve p_2 yukarı kayar ve bu da, aralıklı çizgili, var olan bir indeli yok eder. *(e)* Kalan tek indel silme işlemi çıkarılarak çözümlenir, p_1 aşağı kayar. Bu çizgiyle gösterilen yeni bir indele yol açar. *(f)* Yeni indel silme işleminin çıkarılmasıyla çözümlenir, p_1 aşağı kayar. Başka indel kalmaz.Kesikli çizgiyle gösterilen korunmamış ayrıtlar kaydadeğer değildir. $\beta_1 = 1, \beta_2 = \frac{2}{3}$ için tik atılmış bütün işlemler gerçekleştirilir.

deksten aşağıdaki eklemeler (kesim noktasındaki topolojik benzerlik skorundan daha düşük skorları olanlar) ve yukardaki silmeler (kesim noktasındaki topolojik benzerlik skorundan daha yüksek skorları olanlar) kaydadeğer değillerdir ve hiçbir zaman gerçekleştirilmeyeceklerdir; bakınız Şekil 6.1-b.

6.2.2 Indel Çözümlemeleri

Basitçe kesim noktaları ile belirlenen tüm ekleme/silmeleri gerçekleştirmenin tüm kaydadeğer korunmamış ayrıtları çözümlemeyeceğini belirtmek gerekir. $(u,u'),(v,v') \in A$ ve de korunmamış ayrıt $(u,v) \in E_1, (u',v') \notin E_2$ için eğer $(u,v) \in D_1^{p_1}$ ve $(u',v') \in I_2^{p_2}$ ise E_1'den (u,v)'yi silmek ve E_2'ye (u',v')'i eklemek basitçe korunmamışlığın yönünü tersine çevirir. Benzer şekilde eğer $(u,v) \in I_1^{p_1}$ ve $(u',v') \in D_2^{p_2}$, he riki işlemi birden gerçekleştirisek korunmamışlık hala devam eder. Böyle düğüm ikili düğüm çiftleri $\prec(u,v),(u',v')\succ$'e *indel* deriz. Amaç her indeli tanıladığı ekleme ya da silmelerden birini seçerek çözümlemektir.

İndelleri çözümlerken çözümlemelerin genel kalitesini etkileyen bir etmen indellerin hangi sırayla işlendiğidir. bunun nedeni, indeller çözümlendikçe kesim nokta-

larının değişmesi ve bunun da potansiyel olarak var olan bazı indelleri yok etmesi ya da önceden var olmayan yeni indelleri ortaya çıkarmasıdır. Önce yüksek *önemde* indelleri çözümlenmelidir ki, gelecekte yapılacak indel çözümlemeleri onları yok etmesin. Bu amaçla halihazırda olan tüm indelleri içeren bir *öncelik kuyruğu* (priority queue) yapısı oluştururuz. Değişken in, (u,v)'nin sıralı aday listesi C_1'deki endeksini, in', (u',v')'nin sıralı aday listesi C_2'deki endeksini gösteriyor olsun. Her indel $\prec(u,v),(u',v')\succ$'ye $w(u,v)\times w(u',v')$'lik önemi belirtecek şekilde bir ağırlık atarız. Burda $(u,v)\in D_1^{p_1}$ ve $(u',v')\in I_2^{p_2}$ ise, $w(u,v)=\frac{in+1}{|C_1|}$ ve $w(u',v')=\frac{|C_2|-in'}{|C_2|}$. Diğer yandan eğer $(u,v)\in I_1^{p_1}$ ve $(u',v')\in D_2^{p_2}$ ise $w(u,v)=\frac{|C_1|-in}{|C_1|}$ ve $w(u',v')=\frac{in'+1}{|C_2|}$; bakınız Şekil 6.1-c. Bu tanımlarla silmeye karşılık gelen bir çiftin ağırlığı topolojik benzerlik skoruyla orantılı iken, ekleme çiftinin ağırlığı ters orantılıdır. Basit ifadeyle, düşük indel ağırlığı düşük topolojik benzerlik skorlu bir etkileşimi silmeye ve yüksek topolojik benzerlik skorlu eksik bir etkileşimi de eklemeye, yani yüksek önemli bir indele karşılık gelir. Tekrarlı olarak Q'dan en küçük ağırlıklı indeli çıkarırız, gerekli veri yapılarını yenilereyer işleriz ve bunlara Q boş olana kadar devam ederiz. Verili ağırlık tanımlarıyla, işlem ekleme de silme de olsa, ağırlık küçük oldukça o işlemi yapmaya olan güvenin daha yüksek olduğunu belirtelim. Dolayısıyla bir indeli işleme öncelikle yüksek ağırlıklı işlemi aday listesi C_x'den çıkarmayı gerektirir; işlem hiç gerçekleştirilmeyecektir. Bu işlem kesim noktası p_x'in yerini değiştirir. Dolayısıyla $\alpha_x\times|D_x^{p_x}|=|I_x^{p_x}|$ formülünü kullanarak p_x ve kümeler $D_x^{p_x}$, $I_x^{p_x}$'i yeniden hesaplarız. $D_x^{p_x}$,$I_x^{p_x}$'i değiştirmek var olan bir indelin yok olmasına neden olabilir; bakınız Şekil 6.1-d. Bu yüzden aslında her tekrarın başında Algoritma 5'nin 6. satırında eldeki indelin hala indel olup olmadığına bakarız. Bu $(u,v)\in D_1^{p_1}\cup I_1^{p_1}$ ve $(u',v')\in D_2^{p_2}\cup I_2^{p_2}$'nin sağlanıp sağlanmadığına bakılarak yapılır. İndel çıkarılmasına ek olarak, $D_x^{p_x}$,$I_x^{p_x}$'deki değişiklikler aynı zamanda önceden Q'da olmayan yeni indeller de yaratabilir, ki bunlar da basitçe Q'ya eklenirler; bakınız Şekil 6.1-e.

6.2.3 Ağları ve Hizalamayı Yenileme

Bütün indeller çözümlenince, kaydadeğer korunmamış ayrıtları çözümlemek için, $x=1,2$ için $D_x^{p_x}$,$I_x^{p_x}$'de kalan bütün işlemler eşzamanlı gerçekleştirilebilir. Ancak ortaya çıkan ekleme silmelerden endeksleri p_x kesim noktasına yakın işlemlerin birbirine çok yakın topolojik benzerlik skorları olabilir, ki bu da silmeler düşünüldüğünde yüksek, eklemeler düşünüldüğünde ise düşük görülebilir. Bu yüzden gözü kapalı bir şekilde tüm işlemleri gerçekleştirmek yerine $x=1,2$ için kullanıcı tanımlı β_x parametrelerini kullanırız. Gerçekleştirilen silmeler $D_x^{p_x}$2deki bütün silmeler arasından en iyi $\beta_x\times|D_x^{p_x}|$ tanesi, yani en küçük $\beta_x\times|D_x^{p_x}|$ endeksliler, eklemeler ise $I_x^{p_x}$'deki en yüksek $\beta_x\times|I_x^{p_x}|$ endekslilerdir; bakınız Şekil 6.1-f. Bununla SiPAN'ın ağ yeniyapım aşaması tamamlanır. Sonra yenilenmiş ağlar tekrar hizalanır ve yeni hizalama SiPAN'ın takip

eden tekrarına girdi olur.

6.3 Karşılaştırmalı Deneysel Sonuçlar

SiPAN gerçekleştirimi C++'da LEDA kütüphanesi [61] kullanılarak yapıldı. Açık kaynak kod, derlenmiş program ve veriler http://webprs.khas.edu.tr/~cesim/SiPAN.tar.gz adresinden edinilebilir.

Değerlendirmelerimizi dört yoğun çalışılmış tür üzerinde gerçekleştirdik: C. Elegans, D. Melanogaster, H. Sapiens ve S. Cerevisiae. Girdi olarak SiPAN bunlardan bir ağ çiftine ve ilgili bütün proteinlerin çiftlerinin dizisel benzerlik skorlarına ihtiyaç duyar. Bütün bu veriler PPE ağ hizalama çalışmalarının pekçoğunun da kullandığı IsoBase veritabanından çekilmiştir [67]. Burdaki PPE ağları, aralarında DIP [74], BIOGRID [15], HPRD [50], MINT [18] ve IntAct [7]'in de bulunduğu pekçok veritabanından verilerin birleştirilmesiyle oluşturulmuştur. Dizisel benzerlik skorları Ensembl [44]'dan alınan protein dizilerinin BLAST bit değerleridir.

Genel SiPAN çerçevesi eşzamanlı olarak hem verili ağları yeniyapıma uğratıp hem hizaladığından her iki probleme yönelik performans değerlendirmelerini sunarız. İlk problem için karşılaştırma yöntemi RWS [56] iken ikinci için SPINAL [5] kullanılır. Literatürde bu problemler için başarımı yüksek olarak bilinen algoritmalar olmalarının yanında, bu algoritmalarla karşılaştırma yapmak SiPAN genel çerçevesinin getirdiği genel iyileştirmeyi de görmemizi sağlar; zira SiPAN her iki algoritmadan da değişik adımlarında faydalanmaktadır. Bütün sınamalar için SiPAN parametresi k, 5 seçilmiştir, yani genel SiPAN çerçevesi beş tekrarlı çalışır. Parametreler β_1, β_2'nin her ikisi de 0.5dir.

6.3.1 Değerlendirme Metrikleri

Performans değerlendirmesi için kullanılan kriterler iki veritabanına dayalıdır: Gene Ontology (GO) veritabanı [8] ve STRING veritabanı [34]. GO veritabanındaki GO naotasyonlarını [80, 57, 5]'daki gibi standartlaştırmak için protein anotasyonlarını 5. düzey ile kısıtlarız. Bir *öbek* ağ yeniyapım değerlendirmelerinde bir çift etkileşen proteini, ağ hizalama değerlendirmelerinde ise hizalanmış bir çift proteini simgeliyor olsun. Bir öbek, içindeki iki protein de belli GO anotasyonuna sahipse *anote* olarak anılır. Anote bir öbeğe her iki protein de en az bir ortak standart GO anotasyonuna sahipse *tutarlı* denir.

GO tabanlı değerlendirmeler için beş metrik kullanırız. *Duyarlılık* tutarlı öbek sayısına karşılık gelirken *spesifiklik* tutarlı öbeklerin anote öbeklere oranını belirtir. Üç yeni metrik daha tanımlarız. *GO dağılım duyarlığı* ortalama bir standard GO kategorii tarafından tutarlı kılınan proteinlerin sayısını ifade ederken, *GO dağılım spesifikliği*

Tablo 6.1: Hizalama kalitesi değerlendirmeleri.

Networks	Algorithm	Sen.	Spe.	GO Sen.	GO Spe.	GOC
ce-dm	SPINAL	1062	0.663	**307.2**	**0.183**	413.0
	SiPAN	**1070**	**0.670**	304.8	0.181	**417.9**
ce-hs	SPINAL	731	0.688	243.8	**0.118**	230.6
	SiPAN	**732**	**0.698**	**244.8**	0.117	**232.8**
ce-sc	SPINAL	**854**	**0.507**	**221.4**	0.135	**340.8**
	SiPAN	853	**0.507**	220.8	**0.136**	340.2
dm-hs	SPINAL	**1921**	**0.684**	**377.0**	**0.120**	606.8
	SiPAN	1920	**0.684**	375.5	0.119	**609.5**
dm-sc	SPINAL	1914	0.537	328.0	**0.128**	697.8
	SiPAN	**1925**	**0.539**	**328.3**	0.127	**704.4**
hs-sc	SPINAL	1406	0.555	**271.8**	**0.086**	454.2
	SiPAN	**1418**	**0.570**	270.9	**0.086**	**458.3**

bunu bir GO teriminin verdiği anotasyon sayısı ile normalize eder. Son olarak, anote bir öbek x için $GO_{int}(x)$ ve $GO_{uni}(x)$, sırasıyla x'deki proteinşerin GO naotasyonlarının kesişimi ve birleşimini gösteriyor olsun. *GO tutarlılık* (GO consistency, GOC) skoru bütün anote öbekler üstünden $|GO_{int}|/|GO_{uni}|$ toplamına karşılık gelir.

STRING tabanlı değerlendirmelerde *neighborhood, fusion, coexpression, experimental, database, textmining* metriklerinden faydalnırız. Tüm bu metrikler [86]'de tanımlanmıştır. Verili bir protein etkileşimi için STRING veritabanı her metrik için etkileşime bir skor verir. Bir bütün ağ için biz, bütün etkileşimlerin STRING'den gelen skorlarının ortalamsını alırız. STRING tabanlı değerlendirmelerin sadece ağ yeniyapım değerlendirmelerine uygun olduğunu belirtelim.

6.3.2 Ağ Hizalama Kalitesi

SiPAN'ın hizalama başarımı için, SPINAL'in orjinal ağlar üzerindeki hizalamalarını, SiPAN'ın yeniyapım sonrası oluşturduğu aynı boyutlu ağ hizalamaları ile karşılaştırırız. Sonuçlar Tablo 6.1'de bulunabilir. Toplam 30 enstantanenin 11'inde SPINAL sonuçları daha iyiyken 16 tanesinde SiPAN daha iyi sonuçlar verir ve 3 durum için de eşitlik vardır. SiPAN'ın başarım kalitesi (C. Elegans, D. Melanogaster), (D. Melanogaster, S. Cerevisiae)ve (H. Sapiens, S. Cerevisiae) çiftlerinde daha belirgindir. Bu hizalamların herbirinde GO dağılım duyalığı ve DO dağılım spesifisitesi skorları çok yakın olsa da, doğrudan GO-tabanlı metrikler olan duyarlık, spesifisite ve GO tutarlılık metriklerinde SiPAN çok daha iyi sonuçlar üretmektedir. Kalan iki çift (C. Elegans, H. Sapiens) ve (C. Elegans, S. Cerevisiae) içinse hizalama kalitesi skorları arasındaki fark yok sayılabilecek kadar azdır.

6.3.3 Ağ Yeniyapım Kalitesi

İki çeşit değerlendirme sunarız. Birinde ağ boyutu korunurken diğerinde ağ boyutu artar. Özellikle ikincisi, PPE ağlarının oldukça yüksek yanlış negatif oranları olduğundan (genelde yanlış pozitif oranlarından çok daha yüksek) [43] oldukça önemlidir.

Korunmuş Ağ Boyutları

Ağ boyutlarını korumak için SiPAN'da α_1, α_2'yi 1 seçeriz. Sonuçlar Tablo 6.2'de sunulmuştur. İlk kolonu X_{org} işaretli her satırda X'e ait orjinal ağın değerlendirmeleri, X_{rws} ile işaretlenende X'e RWS uygulandıktan sonra elde edilen ağın değerlendirmeleri vardır. İlk kolonu X_Y ile işaretli her satırda SiPAN X, Y çifti üstünde çalıştırıldıktan sonra elde edilen X ağının değerlendirmeleri vardır. Toplam 52 durumdan 10'unda RWS ağları, 12'sinde orjinal ağlar, 26'sında SiPAN ağları daha iyi skor üretmiştir. SiPAN ağları ve orjinal ağlar 4 durumda eşit skorludur.

Ağ boyutu koruma durumunda, önerilmiş ağ yeniyapımının eğer değerlendirme sonuçları orjinal ağa azçok yakın sonuçlar üretiyorsa başarılı sayıldığını belirtelim. RWS ağlarını orjinallerle karşılaştırırsak, 17'sinde RWS ağları daha iyiyken 35 tanede orjinaller daha iyidir. Diğer taraftan SiPAN ağları çoğu durumda orjinallerden daha iyidir. 14 durumda orjinaller daha iyiyken, 34 durumda SiPAN ağları daha iyidir. Bütün türler için RWS ağlarının duyarlık ve spesifisite skorlarının hem orjinallerden hem de SiPAN ağlarından daha kötü olması dikkat çekicidir.

Ağ Boyutlarının Büyümesi

PPE ağlarının yüksek yanlış negatif oranlarından dolayı ağ yeniyapımında temel hedef ağ kalitesini azçok koruyarak ağ yoğunluğunu arttırmaktır. Ağ boyutları büyüdüğünde elde edilen sonuçlar Tablo 6.3'de sunulmuştur. SiPAN için bu tarz ağ büyümesi α parametresi ile olur. Sunulan sonuçlar için ağ yoğunluklarına ters orantılı α değerleri kullandık; değerlendirilen bir ağ $G_x(V_x, E_x)$ için karşılık gelen α, $\frac{|V_x|^2}{|E_x| \times 250}$ olarak seçildi. Öte yandan RWS açık bir ağ yeniyapımı oluşturmak yerine (u, v) noktasındaki değerin u, v etkileşimine olan güveni ifade eden matrisler üretir. Yeni d boyutlu bir ağ bundan sonra matriste en yüksek skorlu d değere karşılık gelen etkileşimlerden oluşan ağ olarak alınır. Tabloda X_Y ile işaretli her satırda SiPAN'ın X, Y çifti üstünde çalıştıktan sonra ürettiği X ağının sonuçlarını verir. X'_Y, SiPAN'ın X, Y üstünde çalışıp X'te ürettiği kadar ayrıtı olan RWS tarafından üretilmiş X ağının sonuçlarını verir. $|E|$ kolonu ağların yeni boyutlarını verir. Aynı α parametresi kullanılmasına rağmen, verili bir ağ X için, yeniyapıma uğramış ağ X_Y, Y yoğunlaştıkça daha da fazla büyüdüğünü belirtelim. Bu SiPAN'ın bir özelliğidir; Y ağı yoğunlaştıkça, X, Y hizalamasında X'den daha da fazla ayrıt korunmamış olur, ki bu da korunmamış ayrıt çözümlemesi için daha

fazla ayrıtın olası hale gelmesini sağlar. Bu istenen bir özelliktir, zira bu Y'deki bilgi arttıkça, Y'den X'e daha fazla bilgi aktarımı anlamına gelir.

C. Elegans ağı için yeniyapım ağların kalitesi SiPAN için RWS'den bütün metriklerde daha iyidir. GO dağılım duyarlık ve GO dağılım spesifisite metrikleri istisnadır. Aslında tüm türler için RWS, SiPAN'dan bu iki metrik bağlamında daha iyi sonuçlar üretir. Ancak, geriye kalan GO tabanlı metriklerden farklı olarak, bu iki metrik yeniyapım ağın doğruluğuyla ilgili doğrudan ölçüm metrikleri değillerdir. Bunlar aslında tersi yönden değerlendirmeler için uygun metriklerdir; yeniyapım ağlar doğru varsayılırsa, tanımlı GO kategorilerinin ne kadar uygun olduğunu ölçerler. Yine de geçmiş ağ hizalama çalışmaları ile tutarlı olmak adına bu iki metrik sonuçlarını da ağ yeniyapımı için dolaylı metrikler olarak sunarız. C. Elegans ağında SiPAN, RWS'ye göre daha iyi duyarlık ve spesisifite sunarken, algoritmaların GOC skorları benzerdir. STRING bazlı değerlendirmelere gelince, nerdeyse tüm metrikler için SiPAN daha iyi sonuçlar verir. Benzer argümanlar D. Melanogaster için de geçerlidir; SiPAN yeniyapım ağları duyarlık, spesifisite ve GOC skorları için RWS'den daha iyidir. STRING metriklerinde de hemen bütün durumlarda SiPAN daha iyi sonuçlar üretir. H. Sapiens'de SiPAN'ın üstünlüğü diğer türlerde olduğu kadar açık değildir. Tüm üç durumda SiPAN spesifisite skorları daha iyiyken, üçün ikisinde RWS'nin duyalık skorları daha yüksektir. RWS'nin GOC skorları da her durumda daha yüksektir. STRING tabanlı metriklerdeyse başarım oranları yarı yarıyadır. Yine de *birleşik skor*a karşılık gelen combined score metriğinde SiPAN skorları üç dürümğn üçünde de RWS skorlarından yüksektir. SiPAN'ın üstünlüğü en iyi S. Cerevisiae ağında görülür. Her üç durumda da ve hemen her metrik için SiPAN skorları RWS'ye oranla oldukça yüksektir. Son olarak şu gözlemi de belirtmekte fayda vardır. SiPAN yeniyapım ağları orjinal ağlara oranlara çok daha yoğundur; bazı durumlarda nerdeyse üç kat daha yoğun. Böyle de olsa, ağ boyutu normalizasyonu kullanan metriklere (spesifisite ve STRING'in sunduğu bütün metrikler) bile baksak SiPAN skorlarının orjinallere oldukça yakın olduğunu görebiliriz. Bu da SiPAN'ın ağ yeniyapımında temel hedef olan, ağ kalitesinde olabildiğince küçük kayıplarla ağı büyütmede elde ettiği başarıya bir başka işarettir.

	Gene Ontology					*STRING*							
	Duy.	Spe.	GO Duy.	GO Spe.	GOC	NGH	GF	COC	COE	EXP	DAT	TM	CS
ce_{org}	**978**	0.379	32.6	0.032	219.1	0.457	**0.240**	0.536	33.87	**391.3**	19.5	40.5	**414.2**
ce_{rws}	780	0.373	**38.4**	**0.040**	228.2	0.567	0.233	0.598	27.54	200.8	17.1	33.1	228.0
ce_{dm}	967	0.388	35.4	0.035	**234.4**	0.457	**0.240**	0.547	33.84	363.3	20.0	41.5	387.5
ce_{hs}	950	**0.389**	36.4	0.036	229.5	**0.580**	0.239	**0.638**	33.60	353.1	**21.5**	41.6	377.6
ce_{sc}	958	0.383	34.1	0.034	228.3	0.457	**0.240**	0.607	**35.27**	357.4	**21.5**	**42.6**	383.0
dm_{org}	4077	0.309	58.4	0.029	794.9	0.267	**0.055**	0.295	19.77	162.4	23.2	43.5	189.2
dm_{rws}	2919	0.227	**65.5**	**0.033**	693.2	0.299	0.051	0.482	16.12	98.6	20.6	31.0	125.8
dm_{ce}	4099	0.310	59.0	0.030	802.0	0.267	0.054	0.303	19.82	162.0	23.7	43.8	189.0
dm_{hs}	4392	**0.328**	62.3	0.031	910.6	0.349	0.054	0.480	22.19	**167.1**	**32.2**	**50.0**	**198.6**
dm_{sc}	**4434**	0.327	60.5	0.030	**911.3**	**0.358**	**0.055**	**0.529**	**23.13**	165.2	29.1	47.4	195.2
hs_{org}	12417	0.673	121.9	0.038	2126.7	0.677	**0.096**	0.743	23.54	**514.9**	131.9	**196.8**	**586.9**
hs_{rws}	10598	0.583	**177.0**	**0.056**	**2478.9**	0.869	0.090	**1.159**	24.93	176.0	109.9	115.3	277.4
hs_{ce}	12260	0.669	122.5	0.038	2118.6	0.686	**0.096**	0.773	23.93	503.8	131.5	194.0	576.6
hs_{dm}	12388	0.671	123.0	0.038	2178.6	0.676	**0.096**	0.797	24.25	485.5	131.3	191.1	559.7
hs_{sc}	**12576**	**0.686**	136.7	0.042	2350.4	**1.793**	**0.096**	1.064	**36.77**	471.2	**147.5**	195.3	559.6
sc_{org}	**32229**	0.433	**71.2**	0.039	8556.1	**3.230**	**0.157**	0.580	66.37	**515.8**	65.3	**125.8**	**542.2**
sc_{rws}	14305	0.231	69.3	**0.077**	4502.5	2.998	0.022	0.249	**69.48**	82.6	11.5	42.2	133.9
sc_{ce}	32132	0.432	70.6	0.038	**8574.2**	3.182	**0.157**	**0.582**	66.18	494.3	65.2	123.8	521.6
sc_{dm}	30791	0.425	67.0	0.036	8339.5	3.082	0.140	0.538	65.01	438.4	63.4	116.4	466.1
sc_{hs}	30671	**0.434**	69.1	0.038	8405.8	3.063	0.127	0.573	66.01	412.1	**65.5**	116.6	443.2

Tablo 6.2: Korunmuş ağ büyüklükleri sınamaları.

Tablo 6.3: Ağ boyutları büyüdüğünde ağ yeniyapım başarımları.

	$\|E\|$	*Gene Ontology*					*STRING*							
		Sen.	Spe.	GO Sen.	GO Spe.	GOC	NGH	GF	COC	COE	EXP	DAT	TM	CS
ce'_{dm}	6530	1030	0.356	**34.7**	**0.035**	**282.7**	**0.496**	0.175	**0.449**	24.87	170.6	14.4	29.1	196.3
ce_{dm}	6530	**1146**	**0.373**	31.6	0.031	259.6	0.338	**0.177**	0.425	**29.31**	**292.0**	**16.3**	**32.2**	**314.2**
ce'_{hs}	7405	1131	0.347	**33.5**	**0.033**	**305.0**	0.463	0.154	0.411	24.91	158.9	13.6	27.5	185.1
ce_{hs}	7405	**1244**	**0.382**	30.9	0.031	297.7	**0.489**	**0.157**	**0.563**	**29.49**	**262.8**	**18.7**	**32.7**	**286.3**
ce'_{sc}	11247	1536	0.324	**29.2**	**0.029**	412.6	0.395	0.103	0.300	22.18	123.9	10.8	21.4	147.6
ce_{sc}	11247	**1769**	**0.359**	24.7	0.024	**425.5**	**2.655**	**0.168**	**0.548**	**36.97**	**184.8**	**28.1**	**27.0**	**210.9**
dm'_{ce}	25644	2996	0.226	**64.7**	**0.032**	711.6	**0.300**	0.050	**0.476**	16.20	97.3	20.3	30.7	124.4
dm_{ce}	25644	**4232**	**0.312**	58.4	0.029	**822.1**	0.263	**0.053**	0.310	**19.66**	**158.8**	**23.4**	**43.4**	**185.9**
dm'_{hs}	29155	3409	0.226	**60.2**	**0.030**	795.1	0.264	0.044	0.428	15.95	91.2	19.2	29.5	118.0
dm_{hs}	29155	**5383**	**0.344**	58.5	0.029	**1082.0**	**0.331**	**0.047**	**0.431**	**22.91**	**154.9**	**34.1**	**49.6**	**188.7**
dm'_{sc}	30866	3673	0.230	**59.0**	**0.029**	847.8	0.261	0.042	0.404	15.85	88.8	18.9	29.2	115.7
dm_{sc}	30866	**6171**	**0.349**	53.6	0.027	**1250.5**	**0.732**	**0.046**	**0.518**	**29.1**	**149.0**	**32.9**	**46.8**	**183.3**
hs'_{ce}	60151	11618	0.581	**170.7**	**0.054**	**2682.2**	**0.838**	**0.090**	**1.103**	**24.12**	168.9	106.0	111.7	268.7
hs_{ce}	60151	**13281**	**0.662**	115.5	0.036	2236.5	0.634	0.087	0.719	22.25	**469.1**	**121.7**	**181.2**	**538.7**
hs'_{dm}	75361	14393	0.576	**158.2**	**0.049**	**3210.3**	**0.794**	**0.081**	**0.946**	**22.4**	152.9	97.2	103.3	248.5
hs_{dm}	75361	**15424**	**0.618**	100.9	0.031	2550.6	0.527	0.070	0.596	18.54	**374.6**	**98.9**	**148.0**	**434.7**
hs'_{sc}	114900	**21600**	0.568	**134.8**	**0.041**	**4577.1**	0.702	**0.054**	**0.729**	19.84	125.9	**80.9**	89.4	213.7
hs_{sc}	114900	21584	**0.587**	87.7	0.027	3542.0	**1.309**	**0.054**	0.643	**28.47**	**274.2**	77.8	**112.7**	**335.1**
sc'_{ce}	85652	14898	0.232	**69.2**	**0.075**	4659.4	2.971	0.021	0.243	**70.27**	81.6	11.5	42.2	134.4
sc_{ce}	85652	**32928**	**0.427**	68.8	0.037	**8712.0**	**3.110**	**0.152**	**0.558**	64.85	**483.0**	**63.1**	**120.7**	**510.7**
sc'_{dm}	91277	15980	0.233	**68.5**	**0.072**	4939.0	2.882	0.020	0.240	**71.44**	80.0	11.5	42.0	135.2
sc_{dm}	91277	**32824**	**0.410**	64.1	0.035	**8713.0**	**2.950**	**0.151**	**0.523**	60.35	**427.0**	**58.9**	**111.2**	**453.7**
sc'_{hs}	97520	17218	0.234	**69.6**	**0.071**	5256.6	2.761	0.026	0.239	**72.23**	78.5	11.6	42.0	135.9
sc_{hs}	97520	**33869**	**0.409**	63.6	0.034	**9043.0**	**2.763**	**0.130**	**0.517**	57.56	**379.3**	**57.3**	**105.3**	**408.6**

Bölüm 7

Sonuç

Biyolojik ağların global hizalanması problemi çerçevesinde gerçekleştirdiğimiz bilimsel çalışmaların sonuçlarını sunduk. Hizalama problemi bağlamında çeşitli versiyonları formel tanımlayıp ugun algoritma tasarımı gerçekleştirdik.

Üzerinde durulan ilk hizalama versiyonu, global bire-bir ağ hizalama problemydi. Bölüm 2'de ayrıntıları işlenen bu problem versiyonu için geliştirdiğimiz özgün SPINAL algoritmasını sunduk ve SPINAL hizalamalarını PPE ağlarının hizalanmasına uyguladık. Yapılan karşılaştırmalı başarım sınamaları, SPINAL'in alternatiflere oranla hesapsal olarak daha verimli olduğunu ve ürettiği hizalama sonuçlarının da biyolojik anlam çerçevesinde tanımlı metrikler bağlamında daha yüksek performanslı olduğunu gösterdi. Yapılan çalışmaların yayınlanması kısa süre önce olmasına rağmen şimdiden global bire-bir ağ hizalama çalışmalarında bir mihenk taşı olmuş durumdadır [21]. Bölüm 2'de ele alınan bir diğer altproblem de, ortaya çıkan global ağ hizalamalarının proteinler için fonkiyon çıkarımında nasıl kullanılabileceğiydi. Bu bağlamda doğrudan anotasyon transferleri yerine hizalama ağında regülatör öbek çıkarsanmasına dayalı bir transferin daha başarılı fonksiyon çıkarımında bulunabileceğini gösterdik.

Bölüm 3'de global bire-çoklu ağ hizalama problem versiyonunu tanımladıktan sonra probleme yönelik geliştirdiğimiz CAMPWays algoritmasını sunduk. Bu bölümde ağ hizalamayı yönlü metablik yolaklar üzerinde gerçekleştirdik. Yine alternatif bir bire-çoklu ağ hizalama algoritması ile üretilen sonuç hizalamaların anlamlılığını sınadık ve CAMPWays'in çoğu durumda daha başarılı sonuçlar ürettiğini gördük.

Takip eden bölümde global ağ hizalamalarına kısıtlı ağ hizalama çerçevesini önerdik. Kısıtlı hizalamalar bağlamında yoğun çizge-teorik sonuçlar üzerinde durduk. Elde edilen bu sonuçlar hizalama problemine uygun maksimum bağımsız küme

çözümlerinin uyarlanabilir olmasını sağladı ve böylece kısıtlı ağ hizalamasına yönelik çeşitli yaklaşım algoritmaları (approximation algorithms) ve sabit parametre kolay işlenirlik (fixed parameter tractability) çözümleri elde edebildik.

Bölüm 5'de global ağ hizalamanın en genel hali olan çoklu hizalamaları konu edindik. Burda girdi olarak herhangi sayıda biyolojik ağ varsayılır. Çıktı olarak da hedef çoklu hizalamalar, yani her öbekte her ağdan herhangi sayıda ağ yapıtaşının bulunduğu öbekler kümesi, üretmektir. Bu problem versiyonuna yönelik de BEAMS algoritmasını geliştirdik ve alternatif çoklu hizalama algoritmalarından daha iyi başarım performanslı olduğunu simülasyon deneyleriyle veritabanlarından çekilen gerçek verilerle gösterdik. Çoklu ağ hizalamaları da yine PPE ağlarına uygulandı.

Son olarak, Bölüm 6'da içiçe geçmiş iki problem olan ağ çıkarımı ve ağ hizalama problemlerini eşzamanlı ele alan bir çerçeve önerdik. Problemlere eşzamanlı çözüm getiren SiPAN algoritmasını tasarladık. SiPAN algoritmasını her probleme ayrık yaklaşan alternatif ağ çıkarım ve ağ hizalama algoritmaları ile karşılaştırdık ve daha yüksek başarımlı sonuçlar gözlemledik.

Sonuç olarak gerçekleştirilen çalışmalarda bulunan sonuçlar global ağ hizalama ve onunla ilintili ağ çıkarımı gibi konularda önemli açıklıkları kapatmıştır. Dolayısıyla gelecekte yapılan çalışmalarda tek başına ağ hizalamalarına odaklanmaktansa, global ağ hizalamalarının, gen ifadesi, protein kompleksleri, filogenetik yapılar gibi diğer biyoenformatik yapılarla nasıl ilişkilendikleri, onların çıkarsanması, doğrulanması ve yeniden yapımları gibi problemlerde nasıl kullanılabilecekleri, ya da o yapılarla entegre bir şekilde fonksiyon tahmini gibi önemli biyolojik sorunları nasıl çözebilecekleri konularına yoğunlaşılacaktır.

Referanslar

[1] http://biotech.matcmadison.edu/resources/proteins/labManual/images/220_04_113.png.

[2] Gamze Abaka, Turker Biyikoglu, and Cesim Erten. Campways: constrained alignment framework for the comparative analysis of a pair of metabolic pathways. *Bioinformatics*, 29(13):i145–i153, 2013.

[3] R Aebersold and M Mann. Mass spectrometry-based proteomics. *Nature*, 422(6928):198–207, 2003.

[4] Rasmus Agren, Sergio Bordel, Adil Mardinoglu, Natapol Pornputtapong, Intawat Nookaew, and Jens Nielsen. Reconstruction of genome-scale active metabolic networks for 69 human cell types and 16 cancer types using init. *PLoS Computational Biology*, 8(5), 2012.

[5] A. E. Aladağ and C. Erten. Spinal: Scalable protein interaction network alignment. *Bioinformatics*, 29(7):917–924, 2013.

[6] Ferhat Alkan and Cesim Erten. Beams: backbone extraction and merge strategy for the global many-to-many alignment of multiple ppi networks. *Bioinformatics*, 30(4):531–539, 2014.

[7] B. Aranda, P. Achuthan, Y. Alam-Faruque, I. Armean, A. Bridge, C. Derow, and et al. The intact molecular interaction database in 2010. *Nucleic Acids Research*, 38(Database-Issue):525–531, 2010.

[8] Michael Ashburner, Catherine A. Ball, Judith A. Blake, and et al. Gene Ontology: tool for the unification of biology. *Nat Genet*, 25(1):25–29, 2000.

[9] Ferhat Ay, Manolis Kellis, and Tamer Kahveci. Submap: aligning metabolic pathways with subnetwork mappings. *Journal of Computational Biology*, 18(13):219–235, 2011.

[10] Eric Banks, Elena Nabieva, Ryan Peterson, and Mona Singh. NetGrep: fast network schema searches in interactomes. *Genome Biology*, 9(9):R138+, 2008.

[11] Mohsen Bayati, Christian Borgs, Jennifer T. Chayes, and Riccardo Zecchina. Belief propagation for weighted b-matchings on arbitrary graphs and its relation to linear programs with integer solutions. *SIAM J. Discrete Math.*, 25(2):989–1011, 2011.

[12] Piotr Berman. A d/2 approximation for maximum weight independent set in d-claw free graphs. In *Proceedings of the 7th Scandinavian Workshop on Algorithm Theory*, SWAT '00, pages 214–219, 2000.

[13] Judith A. Blake, Joel E. Richardson, Carol J. Bult, James A. Kadin, and Janan T. Eppig. The mouse genome database (mgd): the model organism database for the laboratory mouse. *Nucleic Acids Research*, 30(1):113–115, 2002.

[14] Ravi Boppana and Magnús M. Halldórsson. Approximating maximum independent sets by excluding subgraphs. *BIT*, 32(2):180–196, May 1992.

[15] B.J. Breitkreutz, C. Stark, T. Reguly, L. Boucher, A. Breitkreutz, M. Livstone, R. Oughtred, D.H. Lackner, J. Bahler, V. Wood, and et al. The biogrid interaction database: 2008 update. *Nucleic Acids Research*, 36(Database-Issue):637–640, 2008.

[16] Carlo Vittorio Cannistraci, Gregorio Alanis-Lobato, and Timothy Ravasi. Minimum curvilinearity to enhance topological prediction of protein interactions by network embedding. *Bioinformatics*, 29(13):i199–i209, 2013.

[17] R. Caspi, H. Foerster, C.A. Fulcher, P. Kaipa, M. Krummenacker, M. Latendresse, S. Paley, S.Y. Rhee, A.G. Shearer, C. Tissier, T.C. Walk, P. Zhang, and P.D. Karp. The metacyc database of metabolic pathways and enzymes and the biocyc collection of pathway/genome databases. *Nucleic Acids Res*, 36(Database issue):D623–31, 2008.

[18] A. Ceol, A. Chatr Aryamontri, L. Licata, D. Peluso, L. Briganti, L. Perfetto, L. Castagnoli, and G. Cesareni. Mint, the molecular interaction database: 2009 update. *Nucleic Acids Research*, 38(Database-Issue):532–539, 2010.

[19] J. M. Cherry, C. Adler, C. Ball, S. A. Chervitz, S. S. Dwight, E. T. Hester, Y. Jia, G. Juvik, T. Roe, M. Schroeder, S. Weng, and D. Botstein. SGD: Saccharomyces Genome Database. *Nucleic acids research*, 26(1):73–79, January 1998.

[20] Leonid Chindelevitch, Chung-Shou Liao, and Bonnie Berger. Local optimization for global alignment of protein interaction networks. In *Pacific Symposium on Biocomputing*, pages 123–132, 2010.

[21] Connor Clark and Jugal Kalita. A comparison of algorithms for the pairwise alignment of biological networks. *Bioinformatics*, 30(16):2351–2359, 2014.

[22] Jose C. Clemente, Kenji Satou, and Gabriel Valiente. Phylogenetic reconstruction from non-genomic data. *Bioinformatics*, 23(2):e110–e115, January 2007.

[23] Konrad Dabrowski, Vadim V. Lozin, Haiko Müller, and Dieter Rautenbach. Parameterized complexity of the weighted independent set problem beyond graphs of bounded clique number. *J. Discrete Algorithms*, 14:207–213, 2012.

[24] Banu Dost, Tomer Shlomi, Nitin Gupta, Eytan Ruppin, Vineet Bafna, and Roded Sharan. Qnet: A tool for querying protein interaction networks. *Journal of Computational Biology*, 15(7):913–925, 2008.

[25] Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Springer-Verlag, 1999.

[26] Jack Edmonds. Maximum matching and a polyhedron with 0, 1-vertices. *Journal of Research of the National Bureau of Standards B*, 69:125–130, 1965.

[27] Isabelle Fagnot, Gaëlle Lelandais, and Stéphane Vialette. Bounded list injective homomorphism for comparative analysis of protein-protein interaction graphs. *J. of Discrete Algorithms*, 6(2):178–191, June 2008.

[28] Yi Fang, William Benjamin, Mengtian Sun, and Karthik Ramani. Global geometric affinity for revealing high fidelity protein interaction network. *PLoS ONE*, 6(5):e19349, 2013.

[29] Guillaume Fertin, Romeo Rizzi, and Stéphane Vialette. Finding occurrences of protein complexes in protein–protein interaction graphs. *Journal of Discrete Algorithms*, 7(1):90 – 101, 2009.

[30] Jason Flannick, Antal Novak, Balaji S. Srinivasan, Harley H. McAdams, and Serafim Batzoglou. Graemlin: general and robust alignment of multiple large interaction networks. *Genome Research*, 16(9):1169–1181, 2006.

[31] Jason Flannick, Antal F. Novak, Chuong B. Do, Balaji S. Srinivasan, and Serafim Batzoglou. Automatic parameter learning for multiple local network alignment. *Journal of Computational Biology*, 16(8):1001–1022, 2009.

[32] Francois Fouss, Alain Pirotte, Jean-Michel Renders, and Marco Saerens. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Trans. on Knowl. and Data Eng.*, 19(3):355–369, March 2007.

[33] Andrew D. Fox, Benjamin J. Hescott, Anselm C. Blumer, and Donna K. Slonim. Connectedness of PPI network neighborhoods identifies regulatory hub proteins. *Bioinformatics*, 27(8):1135–1142, 2011.

[34] Andrea Franceschini, Damian Szklarczyk, Sune Frankild, Michael Kuhn, Milan Simonovic, Alexander Roth, Jianyi Lin, Pablo Minguez, Peer Bork, Christian von Mering, and Lars Juhl Jensen. STRING v9.1: protein-protein interaction networks, with increased coverage and integration. *Nucleic Acids Research*, 41(Database-Issue):808–815, 2013.

[35] Harold N. Gabow. Scaling algorithms for network problems. In *Proceedings of the 24th Annual Symposium on Foundations of Computer Science*, SFCS '83, pages 248–258, Washington, DC, USA, 1983. IEEE Computer Society.

[36] M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[37] C. S. Goh and F. E. Cohen. Co-evolutionary analysis reveals insights into protein-protein interactions. *Journal of molecular biology*, 324(1):177–192, 2002.

[38] R. Guimerà, M. Sales-Pardo, and L.A.N. Amaral. A network-based method for target selection in metabolic networks. *Bioinformatics*, 23(13):1616–1622, July 2007.

[39] M. M. Halldórsson. Approximations of weighted independent set and hereditary subset problems. *Journal of Graph Algorithms and Applications*, 4(1):1–16, 2000.

[40] V. Helms. *Principles of Computational Cell Biology*. Wiley-VCH, 2008.

[41] M. Heymans and A. Singh. Deriving phylogenetic trees from the similarity analysis of metabolic pathways. *Bioinformatics*, 19:138–146, 2003.

[42] H.D. Höltje, G. Folkers, and T. Beier. *Molecular modeling: basic principles and applications*. Methods and principles in medicinal chemistry. VCH, 1997.

[43] Hailiang Huang, Bruno Jedynak, and Joel S. Bader. Where have all the interactions gone? estimating the coverage of two-hybrid protein interaction maps. *PLoS Computational Biology*, 3(11), 2007.

[44] T.J. Hubbard, B.L. Aken, S. Ayling, B. Ballester, K. Beal, E. Bragin, S. Brent, Y. Chen, P. Clapham, L. Clarke, and et al. Ensembl 2009. *Nucleic Acids Research*, 37(Database-Issue):690–697, 2009.

[45] Jose MG Izarzugaza, David Juan, Carles Pons, Florencio Pazos, and Alfonso Valencia. Enhancing the prediction of protein pairings between interacting families using orthology information. *BMC Bioinformatics*, 9(35), 2008.

[46] B. Junker and F. Schreiber. *Analysis of Biological Networks*. Wiley-Interscience, 2008.

[47] Minoru Kanehisa, Susumu Goto, Yoko Sato, Miho Furumichi, and Mao Tanabe. Kegg for integration and interpretation of large-scale molecular data sets. *Nucleic Acids Research*, 40(Database-Issue):109–114, 2012.

[48] Brian P. Kelley, Roded Sharan, Richard M. Karp, Taylor Sittler, David E. Root, Brent R. Stockwell, and Trey Ideker. Conserved pathways within bacteria and yeast as revealed by global protein network alignment. *Proceedings of the National Academy of Sciences*, 100(20):11394–11399, 2003.

[49] Brian P. Kelley, Bingbing Yuan, Fran Lewitter, Roded Sharan, Brent R. Stockwell, and Trey Ideker. Pathblast: a tool for alignment of protein interaction networks. *Nucleic Acids Research*, 32(Web-Server-Issue):83–88, 2004.

[50] T.S. Keshava Prasad, R. Goel, K. Kandasamy, S. Keerthikumar, S. Kumar, S. Mathivanan, D. Telikicherla, R. Raju, B. Shafreen, A. Venugopal, and et al. Human protein reference database-2009 update. *Nucleic Acids Research*, 37(Database-Issue):767–772, 2009.

[51] Mehmet Koyutürk, Yohan Kim, Umut Topkara, Shankar Subramaniam, Wojciech Szpankowski, and Ananth Grama. Pairwise alignment of protein interaction networks. *Journal of Computational Biology*, 13(2):182–199, 2006.

[52] Oleksii Kuchaiev, Tijana Milenković, Vesna Memišević, Wayne Hayes, and Nataša Pržulj. Topological network alignment uncovers biological function and phylogeny. *Journal of The Royal Society Interface*, 7(50):1341–1354, 2010.

[53] Oleksii Kuchaiev and Natasa Pržulj. Integrative network alignment reveals large regions of global network similarity in yeast and human. *Bioinformatics*, 27(10):1390–1396, 2011.

[54] Oleksii Kuchaiev, Marija Rasajski, Desmond J. Higham, and Natasa Przulj. Geometric de-noising of protein-protein interaction networks. *PLoS Computational Biology*, 5(8):e1000454, 2009.

[55] Sheng-An Lee, Cheng hsiung Chan, Chi-Hung Tsai, Jin-Mei Lai, Feng-Sheng Wang, Cheng-Yan Kao, and Chi-Ying F Huang. Ortholog-based protein-protein interaction prediction and its application to inter-species interactions. *BMC Bioinformatics*, 9:S11, 2008.

[56] Chengwei Lei and Jianhua Ruan. A novel link prediction algorithm for reconstructing protein-protein interaction networks by topological similarity. *Bioinformatics*, 29(3):355–364, 2013.

[57] Chung-Shou Liao, Kanghao Lu, Michael Baym, Rohit Singh, and Bonnie Berger. Isorankn: spectral methods for global alignment of multiple protein networks. *Bioinformatics*, 25(12):i253–i258, 2009.

[58] Brenton Louie, Roger Higdon, and Eugene Kolker. A statistical model of protein sequence similarity and function similarity reveals overly-specific function predictions. *PLoS One*, 4(10):e7546, 2009.

[59] J. March. *Advanced Organic Chemistry: Reactions, Mechanisms, and Structure*. Wiley, New York, 1985.

[60] Edward M. Marcotte, Matteo Pellegrini, Ho-Leung Ng, Danny W. Rice, Todd O. Yeates, and David Eisenberg. Detecting protein function and protein-protein interactions from genome sequences. *Science (New York, N.Y.)*, 285(5428):751–753, 1999.

[61] Kurt Mehlhorn and Stefan Naher. *Leda: A Platform for Combinatorial and Geometric Computing*. Cambridge University Press, 1999.

[62] V. Memišević and N. Pržulj. C-graal: Common-neighbors-based global graph alignment of biological networks. *Integr Biol (Camb)*, 4(7):734–43, 2012.

[63] Tijana Milenković, Weng Leong Ng, Wayne Hayes, and Nataša Pržulj. Optimal network alignment with graphlet degree vectors. *Cancer Inform.*, 9:121–137, 2010.

[64] Aziz Mithani, Jotun Hein, and Gail M. Preston. Comparative analysis of metabolic networks provides insight into the evolution of plant pathogenic and non-pathogenic lifestyles in Pseudomonas. *Mol Biol Evol*, 28(1):483–499, 2011.

[65] Manikandan Narayanan and Richard M. Karp. Comparing Protein Interaction Networks via a Graph Match-and-Split Algorithm. *Journal of Computational Biology*, 14(7):892–907, 2007.

[66] RA Pache and P Aloy. Increasing the precision of orthology-based complex prediction through network alignment. *PeerJ.*, 2:e413, 2014.

[67] Daniel Park, Rohit Singh, Michael Baym, Chung-Shou Liao, and Bonnie Berger. Isobase: a database of functionally related proteins across ppi networks. *Nucleic Acids Research*, 39(Database-Issue):295–300, 2011.

[68] Ron Y. Pinter, Oleg Rokhlenko, Esti Yeger Lotem, and Michal Ziv-Ukelson. Alignment of metabolic pathways. *Bioinformatics*, 21(16):3401–3408, 2005.

[69] Venkatesh Raman and Saket Saurabh. Triangles, 4-cycles and parameterized (in-)tractability. In *Proceedings of the 10th Scandinavian Conference on Algorithm Theory*, SWAT'06, pages 304–315, 2006.

[70] M. Remm, C. E. Storm, and E. L. Sonnhammer. Automatic clustering of orthologs and in-paralogs from pairwise species comparisons. *Journal of molecular biology*, 314(5):1041–1052, 2001.

[71] Sayed M. Sahraeian and Byung-Jun Yoon. A Network Synthesis Model for Generating Protein Interaction Network Families. *PLoS ONE*, 7(8):e41474+, August 2012.

[72] Sayed Mohammad Ebrahim Sahraeian and Byung-Jun Yoon. Smetana: Accurate and scalable algorithm for probabilistic alignment of large-scale biological networks. *PLoS ONE*, 8(7):e67995, 07 2013.

[73] Shuichi Sakai, Mitsunori Togasaki, and Koichi Yamazaki. A note on greedy algorithms for the maximum weighted independent set problem. *Discrete Appl. Math.*, 126(2-3):313–322, March 2003.

[74] L. Salwinski, C.S. Miller, A.J. Smith, F.K. Pettit, J.U. Bowie, and D. Eisenberg. The database of interacting proteins: 2004 update. *Nucleic Acids Research*, 32:449–451, 2004.

[75] Roded Sharan and Trey Ideker. Modeling cellular machinery through biological network comparison. *Nature Biotechnology*, 24(4):427–433, 2006.

[76] Roded Sharan, Silpa Suthram, Ryan M. Kelley, Tanja Kuhn, Scott McCuine, Peter Uetz, Taylor Sittler, Richard M. Karp, and Trey Ideker. Conserved patterns of protein interaction in multiple species. *Proceedings of the National Academy of Sciences of the United States of America*, 102(6):1974–1979, 2005.

[77] Roded Sharan, Igor Ulitsky, and Ron Shamir. Network-based prediction of protein function. *Molecular systems biology*, 3(1), 2007.

[78] TL Shi, YX Li, YD Cai, and KC Chou. Computational methods for protein-protein interaction and their application. *Curr Protein Pept Sci.*, 6:443–449, 2005.

[79] Tomer Shlomi, Daniel Segal, Eytan Ruppin, and Roded Sharan. QPath: a method for querying pathways in a protein-protein interaction network. *BMC Bioinformatics*, 7:199+, 2006.

[80] Rohit Singh, Jinbo Xu, and Bonnie Berger. Global alignment of multiple protein interaction networks. In *Pacific Symposium on Biocomputing*, pages 303–314, 2008.

[81] Lucy Skrabanek, Harpreet K. Saini, Gary D. Bader, and Anton J. Enright. Computational prediction of protein-protein interactions. *Molecular Biotechnology*, 38(1):1–17, 2008.

[82] Yukako Tohsato, Hideo Matsuda, and Akihiro Hashimoto. A multiple alignment algorithm for metabolic pathway analysis using enzyme hierarchy. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, pages 376–383. AAAI, 2000.

[83] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. Fast random walk with restart and its applications. In *Proceedings of the Sixth International Conference on Data Mining*, ICDM '06, pages 613–622, 2006.

[84] Susan Tweedie, Michael Ashburner, Kathleen Falls, Paul Leyland, Peter Mcquilton, Steven Marygold, Gillian Millburn, David Osumi-Sutherland, Andrew Schroeder, Ruth Seal, Haiyan Zhang, and The FlyBase Consortium. FlyBase: enhancing Drosophila Gene Ontology annotations. *Nucl. Acids Res.*, 37(suppl_1):D555–559, January 2009.

[85] A Valencia and F Pazos. Computational methods for the prediction of protein interactions. *Curr Opin Struct Biol.*, 3:368–373, 2002.

[86] Christian von Mering, Lars Juhl Jensen, Berend Snel, Sean D. Hooper, Markus Krupp, Mathilde Foglierini, Nelly Jouffre, Martijn A. Huynen, and Peer Bork. String: known and predicted protein-protein associations, integrated and transferred across organisms. *Nucleic Acids Research*, 33(Database-Issue):433–437, 2005.

[87] JF Xia, SL Wang, and YK Lei. Computational methods for the prediction of protein-protein interactions. *Protein Pept Lett.*, 9:1069–1078, 2010.

[88] Qingwu Yang and Sing-Hoi Sze. Path matching and graph matching in biological networks. *Journal of Computational Biology*, 14(1):56–67, 2007.

[89] Mikhail Zaslavskiy, Francis R. Bach, and Jean-Philippe Vert. Global alignment of protein-protein interaction networks by graph matching methods. *Bioinformatics*, 25(12):259–267, 2009.

[90] Li Zhenping, Shihua Zhang, Yong Wang, Xiang-Sun Zhang, and Luonan Chen. Alignment of molecular networks by integer quadratic programming. *Bioinformatics*, 23(13):1631–1639, July 2007.

TÜBİTAK
PROJE ÖZET BİLGİ FORMU

Proje Yürütücüsü:	Doç. Dr. CESİM ERTEN
Proje No:	112E137
Proje Başlığı:	Bionetalign: Biyokimyasal Ağlarda Fonksiyonel Ortoloji Çıkarımı Amaçlı Global Hizalamalar
Proje Türü:	1001 - Araştırma
Proje Süresi:	24
Araştırmacılar:	TÜRKER BIYIKOĞLU
Danışmanlar:	TÜRKAN HALİLOĞLU
Projenin Yürütüldüğü Kuruluş ve Adresi:	KADİR HAS Ü. MÜHENDİSLİK VE DOĞA BİLİMLERİ F. BİLGİSAYAR MÜHENDİSLİĞİ B.
Projenin Başlangıç ve Bitiş Tarihleri:	01/10/2012 - 01/01/2015
Onaylanan Bütçe:	146320.0
Harcanan Bütçe:	108055.03
Öz:	Biyolojik ağların analizi, son on yılda yoğun ilgi duyulan bir sistem biyolojisi ve biyoenformatik çalışma alanıdır. Analiz problemlerinden en önemlilerinden biri biyolojik ağların hizalanma problemidir. Hizalama basit bir anlatımla, değişik türlere ait verili biyolojik ağların içeriğindeki karşılık gelen yapıtaşlarının (PPE ağlarında karşılık gelen proteinler, protein kompleksleri, veya metabolik yolaklar için karşılık gelen reaksiyonlar vs.) çıkarsanmasına karşılık gelir. Biyolojik ağ hizalama problemi hücreiçi işleyişi anlamamız, fonksiyonu bilinmeyen proteinlerin fonksiyon çıkarımı, bilinenler için doğrulama ve türlerarası ortoloji ilişkilerini keşfetmemiz açısından oldukça önemlidir. Proje çerçevesinde biyolojik ağların global hizalanması bağlamında değişik problem versiyonu tanımları yaptık ve herbir versiyon için uygun algoritmalar tasarlayıp başarım sınamalarını gerçekleştirdik. Bu çerçevede global bire-bir ağ hizalaması, global bire-çoklu ağ hizalaması, kısıtlı global ağ hizalamalarının çizge-teorik incelemeleri, global çoklu ağ hizalamaları ve son olarak da eş zamanlı etkileşim çıkarımı ve global ağ hizalama konularını ele aldık.
Anahtar Kelimeler:	Biyolojik ağ analizi, protein-protein etkileşim ağları, metabolik yolaklar, ağ hizalaması
Fikri Ürün Bildirim Formu Sunuldu Mu?:	Hayır
Projeden Yapılan Yayınlar:	1- SPINAL: scalable protein interaction network alignment (Makale/Kitap/Kitapta Bölüm)2- BEAMS: backbone extraction and merge strategy for the global many-to-many alignment of multiple PPI networks (Makale - Diğer Hakemli Makale), 3- Constrained Alignments of a Pair of Graphs (Bildiri - Uluslararası Bildiri - Sözlü Sunum),