Merhaba Kerem,

Öncelikle ağ oluşturma kodunla ilgili şu noktalara dikkat çekmeliyim:

• Pearson's correlation (r) hesapladıktan sonra mutlak değerini al, örn. "correlation = numpy.abs(correlation)".

Sadece pozitif ve çok correlated (1'e yakın) genleri değil (birlikte artan, birlikte azalan) negatif ve çok (-1'e yakın) correlated (biri artarken diğeri azalan) genleri de birbirine bağlamak istiyoruz.

(+/- durumunu görselleştiren https://www.spss-tutorials.com/pearson-correlation-coefficient/ 'daki scatter plot'a bakabilirsin.)

Örneğin ilk durumda bir protein diğerinin üretilmesini promote ediyor bilgisini ağa koyarken diğerinde de inhibit ediyor bilgisi katılıyor.

Oluşturduğumuz gen-gen coexpression ilişkisini temsil etmeye çalıştığımız bir ağ.

Sadece pozitif correlation ile ağ oluşturan çalışmalar da var; biz ise negatif correlation'u da bağlamak için kullanan çalışmaları izliyoruz.

• Her gen çifti için 0.6/0.7/0.8 sınırlarını geçiyorsa dosyalara yazarken if/(else if = elif)/(else if) kullanırsan örneğin 0.85 değerine sahip bir çifti 0.8'in dosyasına yazar; ama 0.6 ve 0.7 dosyalarına yazmaz.

Bu şekilde yapmak istiyorsan if/if/if şeklinde düzenlemelisin.

İlgili değişikliklerden sonra ağların özet bilgilerini içeren bir tablo hazırlamanı rica edeceğim. Tabloda:

• Satırlar hücre grupları (AP1, AP2, ...),

• sütunlarda da şu bilgiler olsun:

expression data sütun grubu (örnek yani cell id sayısı ve gen sayısı) ve

her threshold için (0.6, 0.7 ve 0.8) bir ağ sütun grubu (node sayısı ve kenar sayısı).

Sıradaki görev için kısaca bağlamımız şöyle:

Prize-collecting Steiner tree/forest (PCST, PCSF) problemini bir ağda çözmek, node'lar (burada genler) hakkındaki ön bilgiyi (burada otizm ile ilişkilendirme) ve ağın bilgisini birlikte göz önünde bulundurarak o ağ (network/graph) için minimum bir alt ağ (subnetwork/subgraph) bulabilmemizi sağlıyor.

Elimizde is 7 tane ağ var; minimum bir alt ağ bulurken diğer ağlardaki bilgiyi de kullanmak istiyoruz.

Bunu da bir ağ için alt ağ bulurken diğer ağlar için bulduğumuz alt ağları göz önünde bulundurmasını sağlayarak yapmayı deniyoruz; örneğin, alt ağların birbirine ya da belirli ağlara benzemesini teşvik ediyoruz.

Bu görev için adımları şöyle listeleyebilirim:

1. Makalemizin Preprint'deki Background'ın son paragrafında açıklanan basit örneği ve Yöntemler bölümününü gözden geçir.

( Şuradan erişebilirsin: https://www.biorxiv.org/content/early/2018/03/08/256693 )

2. ST-Steiner kodunu indir, örneği çalıştır.

( Şuradan: https://github.com/ciceklab/ST-Steiner . Makaledeki örnek ile aynı.)

Sadece Linux için rehber var şimdilik; Windows ya da OS X kullanıyorsan dilersen sanal makine kur ya da Morpheus sunucumuzda çalış - nasıl kolay olacak ise.

Olmadı Windows için de bir rehber hazırlayabilirim.

3. Şu şekilde çalışan bir yinelemeli çözücü method hazırla:

• Oluşturduğun her ağ için ayrı ayrı bir alt ağ bul.

(bunu ilgili komut satırı metnini oluşturup örneğin python'da subprocess.run(.) komutuyla çağırarak yapabilirsin)

• rastgele bir ağı seç. Diğer bulduğun alt ağları göz önünde bulundurarak bu ağ için yeni bir çözüm üret.

(yeniden çöz, yani alt ağ bulma algoritmasını "--cluster\_list\_file=..." ile diğerlerinin alt ağ dosyalarını listesini göstererek yeniden çalıştır)

(göz önünde bulundurmak derken, örnekteki ikinci spatio-temporal window'un birincinin sonucuna bakması şeklinde)

• önceki adımı N kere yinele.

Method'un parametreleri beta, lambda ve N olsun (beta ve lambda, ST-Steiner ve makalede tanımlandıkları şekli ile).

Makalede beta ve lambda için bir seçim yönteminde bahsediliyor; ancak henüz github'a bu seçiciyi koymadım.

Şimdilik birkaç beta ve lambda değeri için deneyebiliriz. Koyunca kodunu güncelleriz.

Kodunu yeniden kullanılabilirlik odaklıklı hazırlamaya çalışırsak çok daha iyi olur; projeye/github'a ekleyebiliriz sonrasında, dilersen.

Ek olarak:

• Kullanacağın node prize dosyasını ekledim.

Bu dosyada gen ve -log(TADA q-value) eşleşmesi var.

Bu değer, genin otizm ile ilişkilendirilmesini ölçen TADA istatistiksel testi q-değerinin eksi logaritması; ne kadar yüksek, bilebildiğimiz kadarı ile o gen otizm ile o kadar ilişkili demek.

• Bazı genlerin isimlerini değiştirmek gerekiyor node prize dosyası ile uyuşmaları için.

Yeniden isimlendirilecek genler için gen adı - yeni gen adı dosyasını ekte bulabilirsin.

• Edge cost'larını (1-r^2) olarak hesaplamak için "--config\_file=config/setting\_1-rsqd.ini" parametresini eklemelisin: correlation arttıkça azalan bir "edge cost" fonksiyonuna ihtiyacımız var; 1-r^2 de öyle bir fonksiyon ve bu ayar dosyası, o edge cost fonksiyonunu seçiyor.

Açık olmayan yerler varsa ya da takıldığın noktalar olursa lütfen yaz.

İyi çalışmalar dilerim,

--

Utku Norman

MS Candidate in Dept. of Computer Engineering, Bilkent University, Ankara.