Schueller Lab

Lab Journal

Carlos Vigil Vásquez

12de junio de2020

Índice

Enero de 2020 **2**

Mayo del 2020 **4**

Junio del 2020 **7**

Enero de 2020

2020.01.20

Reunión

- Averiguar cambio de capa de similitud de blancos por similitud de pockets.
- Cambiar de identidad de secuencia por Srel de los pockets.
- Cambiar el calculo de red densa a *sparsest-subgraph* utilizando Dijkstra como algoritmo predictivo.
- Cuantificar rendimiento de las predicciones con diferentes porcentajes de eliminación de información

ToDo

☒ Generar método de predicción basado en *shortest path* (algoritmo de Dijkstra) sin generar el subgrafo utilizado para la reimplementación de *nearest-neighbour*.
☒ Añadir al *script* para construir la red un paso donde se elimina el X % de las uniones para cada capa (esto lo dejaría dentro de la función de la construcción de las capas) ***Revisar 2020.01.21***

2020.01.21

Plan del día

☒ Implementar filtrado de uniones al crear capas de la red de consulta.
☒ Crear versión que corre con *shortest paths* en vez de *nearest neighbour* como algoritmo predictivo.
☒ Implementar función de puntuación de DASPFind.

Avances del día

- Para la implementación del filtrado de uniones se proponen 2 opciones diferentes en el *script* que construye la red de consulta:
  - Una basada en "porcentaje de información", donde frente a un porcentaje declarado se eliminarán ese porcantaje de uniones con mayor distancia. Es decir, si declaro un valor de 10 %, se eliminaran las uniones con el 10 % más alto de distancia.
  - Una basada en un "estado de la red", donde se eliminaran todas las uniones con las distancias más altas hasta llegar un punto donde si se elimina 1 unión más, la red pasa a tener más de 1 componente conectado.
- Implementación según "estado de la red": Se obtiene el *minimum spanning tree* de la capa, a partir del cual se obtiene la unión con mayor distancia. Se eliminan todas las uniones que tengan una distancia mayor a este valor.
- Implementación según "porcentaje de información": Se obtiene las distancias de todas las uniones de la capa, se ordenan de menor a mayor y se calcula el valor que se encuentra en el X % de los datos. Se eliminan todas las uniones que tengan una distancia mayor a este valor.

2020.01.23

Plan del día

☒ Implementar lo que quedo en el tintero de días anteriores.
☐ ~~Buscar que estadigrafos de redes calcular para empezar a incorporar información relacional al método predictivo.~~

Avances del día

- Se implementa método predictivo basado en el camino más corto entre 2 nodos, utilizando la función `nx.bidirectional_dijkstra(source, target, weight)` por temas de eficiencia en tiempo.
- Se implementa función de puntuación inspirada en DASPfind, donde la puntuación de una predicción está dada por:

$$score = \left(\prod P_w\right)^{2{,}26 \times len(P)}$$

2020.01.27

Avances del día

Se evaluan las predicciones que se dejaron corriendo el fin de semana, las cuales tenian por objetivo cuantificar el método predictivo con diferentes niveles de información disponible en la red. Para esto, se eliminan diferentes cantidades de uniones para cada capa de la red de consulta multicapa, bajo el principio de percentiles de información (Figuras 1 y 2). La eliminación de estas uniones se genera de la siguiente manera:

- Se genera un histograma con 10 grupos (desde 0.0 a 0.9 con incrementos de 0.1) de la distancia de las uniones para una de las capas de la red de consulta.
- Se selecciona un valor de distancia que conprenda el X % de la información a filtrar.
- Se eliminan todas las uniones que sean mayor al valor de distancia establecido en el paso anterior.

Además se genero una red multicapa con la menor densidad posible por capa (denotado como 'Sparse' en las Figuras 1 y 2), es decir, cada capa de información tiene las uniones necesarias para que de eliminarse aquella de mayor distancia la capa pasa a tener más de 1 componente conectado o más de una "isla" de nodos.

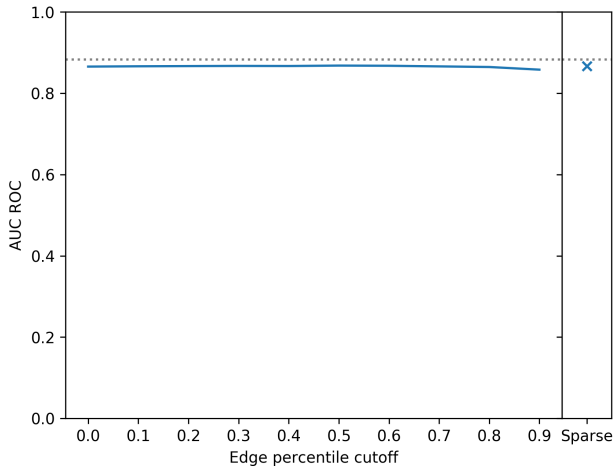

Figura 1: Área bajo la curva ROC para diferentes cantidades de información. La linea punteada gris corresponde al valor de ROC-AUC obtenido para el método predictivo basado en *Nearest-Neighbour*

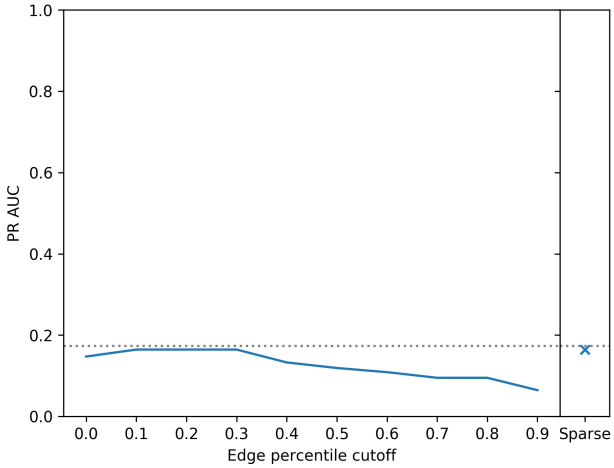

Figura 2: Área bajo la curva PR para diferentes cantidades de información. La línea punteada gris corresponde al valor de PR-AVG obtenido para el método predictivo basado en *Nearest-Neighbour*

Mayo del 2020

2020.05.14

¿Qué hay que hacer?

- ☒ Buscar los papers para los métodos con los que se compara DASPfind.
- ☐ ~~Resumir papers~~
- ☐ ~~Buscar cuales de estos papers tienen *source code* para poder correrlos localmente.~~
- ☒ Crear *workflow* en `tmux` para generar una sesión para trabajar en las cosas del laboratorio.

¿Qué se hizo?

- DASPfind se compara con los siguientes métodos basados en redes:
  - NRWRH
  - DT-Hybrid (Alaimo, 2013)
  - HGBI (Wang, 2013)
- Se crea un archivo en la Wiki para almacenar los resumenes de la bibliografia utilizada para el proyecto de redes (Bibliografía)
- Existen varios papers más del tema, que parecieran ser más "importantes". Sería bueno generar una lista con todos estos papers e ir leyendo de a poco estos para entender bien el estado del arte de estos métodos.
- Nuevo *workflow* para el laboratorio: se crea un pequeño *script* llamado `tmux_lab`, el cual permite crear una sesion de `tmux` con 2 ventanas esenciales: (i) Vim con el *Lab Journal* abierto y (ii) una ventana con 2 paneles para manejar `git`.

2020.05.16

¿Qué hay que hacer?

- ☐ ~~Lectura de papers.~~
- ☒ Calcular un par de estadisticas en el set de datos KEGG Nuclear Receptor

¿Qué se hizo?

- Quiero revisar esta idea de *guilt-by-association* para ver si es posible maximizar este efecto para obtener mejores predicciones. Para esto voy a hacer un pequeño algoritmo que va eliminando las uniones más débiles para luego calcular los valores de *sparsity* y la razón entre el promedio peso de las uniones que tienen nodos que unen el mismo blanco y aquellos que unen blancos diferentes. Referirse a Wang et al., 2013 para más información, en la sección 2.2 (*Intra-similarity analysis*) hacen referencia a este efecto.
- Se creó un script que calcula la densidad de la red, junto con la diferencia del promedio de la similitud entre nodos que comparten blanco y aquellos que no comparten blanco. Podría ser interesante calcular comunidades con estos *cutoffs* de similitud entre ligandos para determinar si existe un efecto de eliminar las uniones más debiles de la red.

2020.05.18

¿Qué hay que hacer?

☒ Analizar 20 entradas de output coronavirus.
☐ ~~Buscar papers de métodos basados en redes.~~

¿Qué se hizo?

- Se dercargó el archivo de salida para predicciones de fármacos de coronavirus enviado al canal de *slack*, se creo una planilla de Excel y se ordeno por Tc de forma decreciente. Del análisis de las entradas de output de las predicciones para el proyecto de coronavirus obtuvimos los siguientes casos interesantes:

Cuadro 1: Candidatos interesantes para SARS-CoV-2 obtenidos a partir de las predicciones generadas por A. Schueller

Query ligand	*Hit ligand*	Tanimoto	Comentarios
CHEMBL940572	D2D_ENB	0.7924	*Query*: Inhibidor de proteasa viral (*Rhinovirus*) *Hit*: Inhibidor de ampli espectro para SARS-CoV
CHEMBL318132	D2D_ENB	0.7924	*Query*: Inhibidor de proteasa viral (*Rhinovirus*) *Hit*: Inhibidor de ampli espectro para SARS-CoV
CHEMBL261723	OP9_WR1	0.8938	*Query*: Inhibidor Cruzipaina (antibacterial) Estructura similar a los ligandos que ocupan todos los bolsillos de SARS-CoV-2
CHEMBL2381930	OP9_WR1	0.8919	*Query*: Inhibidor de Cathepsinas (proteasa)
CHEMBL2381921	OP9_WR1	0.8919	*Query*: Inhibidor de Cathepsinas (proteasa)
CHEMBL2381928	OP9_WR1	0.8919	*Query*: Inhibidor de Cathepsinas (proteasa)
CHEMBL3143630	OP9_WR1	0.8919	*Query*: Inhibidor de tripsina (serin-proteasa)

- La idea sería hacer el análisis de farmacóforos para estas predicciones y después hacer los *dockings* moleculares para ver si los compuestos predichos tienen una pose similar a los ya conocidos para SARS-CoV-2.

2020.05.19

¿Qué hay que hacer?

☒ Presentación de avance
☒ Leer papers impresos hasta el momento

¿Qué se hizo?

- Se hace la presentación de avance para el dia de hoy. Tuve "problemas" haciendo las imagenes para los compuestos, quizas seria interesante hacer un pequeño script que me dibuje las moléculas y marque las

diferencias que tienen entre si de existir un *flag* en el input.

- De la lectura + *skimming* de los papers que he descargado hasta el momento, todos correponderían a métodos basados en la distribución de recursos desde el nodo correspondiente al ligando hacia el nodo del blanco protéico al cual se quiere generar una predicción.
  - Zhou propone un método para redistribuir estos recursos en una red bipartita, generando un método de recomandación basado en cuando del recurso le llega al objeto a recomendar. Funciona muy bien en cuanto al *accuracy* de las predicciones, pero las recomendaciones no son novedosas. Ante esto propone, ProbS y HeatS, los cuales juntos son capaces de generar predicciones precisas y novedosas.
  - Cheng aplica NBI en un contexto de interacciones proteína-ligando, teniendo un mejor rendimiento que utilizando algoritmos basados en la similitud estructural de ligandos y/o proteínas. Puede ser por esta razón que mi método basado en *guilt-by-association* por medio de los caminos más cortos no rinda tan bien, quizas deberiamos considerar el contexto de las otras interacciones a la hora de hacer las predicciones. Como obtenemos un peso para el nodo, podriamos buscar un camino donde la distribución de los recursos es alta y donde el largo del camino más corto es minima, así teniendo un *scoring* en conjunto capaz de generar predicciones con una precision y *recall* muy alto. No generará predicciones novedosas, pero sí las que tendrían lógica bajo el supuesto de *guilt-by-association*.
  - Alaimo extiende esta propuesta de Zhou, normalizando las recomendaciones con la información de similitud estructural, así mejorando el rendimiento de este algoritmo en un contexto de predicciones para interacción proteína-ligando.

¿Qué debería hacer a partir de esto?

- Implementar algoritmo predictivo basado en caminos más cortos que utiliza la redistribución de recursos planteada por Zhou (partamos por lo más sencillo), para así penalizar o bonificar predicciones que pasan por caminos de diferente importancia.
  - Podemos usar diferentes valores o métricas para los nodos como recurso inicial.
- Buscar métodos predictivos de interacciones proteína-ligando basados en esta idea de usar caminos como herramienta para encontrar nuevas interacciones.

Junio del 2020

2020.06.08

¿Qué hay que hacer?

- ☒ Implementar NBI en Python
- ☒ Implementar NBI en Julia
- ☒ Presentación de avance BIO296F

¿Qué se hizo?

- Se implemento NBI en Julia usando matrices de adyacencia rectangulares, es decir, para matrices de adyacencia con forma MxN, donde M corresponden a los ligandos y N a los blancos.
  - Por el momento, se probó para una matriz de adyacencia pequeña, la cual tiene de respaldo que se calculo NBI a mano.
- Se implementó NBI en Python, también utilizando la matriz de adyacencia de la red, pero no se ha probado con un set de datos conocido como en el caso de Julia. Faltaría crear, utilizando el *pipeline* para crear redes utilizado para las predicciones proteína-ligando para corroborar que está bien implementado. Posterior a eso, habria que revisar si se puede optimizar la función, ya que por el momento calcula toda la matriz W (NxN) en vez de solo calcular las predicciones deseadas.
- Se hizo una figura para explicar el método predictivo de NBI (Figura 3), desglozando los distintos pasos del método y como se obtiene la puntuación de la predicción.

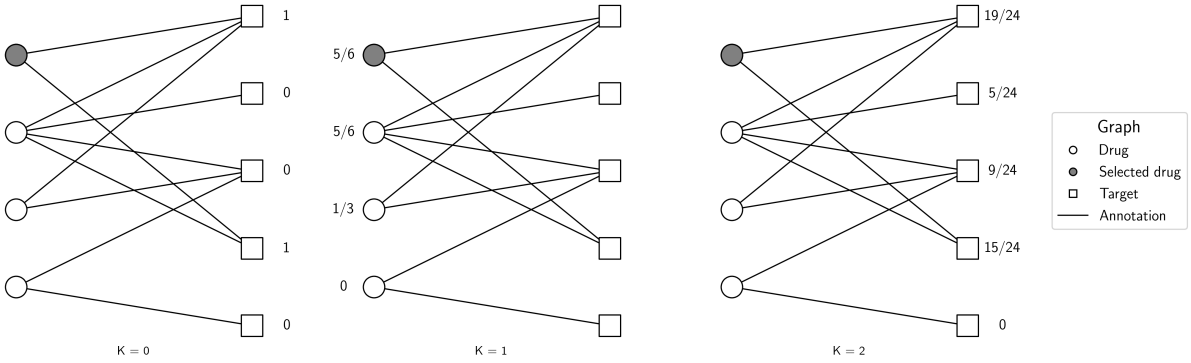

Figura 3: Esquema resumen de NBI

2020.06.09

¿Qué hay que hacer?

- ☐ Probar implementación de NBI en Python
- [ ]