# דוח פרוייקט סיכום קורס ניתוח רשתות ביולגיות

לכתוב את השמות שלנו

# בהנחיית: פרופ' רודד שרן.

# מבוא

ניתוח רשתות ביולוגיות הוא תחום מתפתח שחוקר תהליכים ביולוגיים מורכבים בעזרת ייצוגם כרשתות של אינטראקציות בין מולקולות. ניתוח הרשתות המתקבלות מציב אתגרים המשלבים תורת הגרפים ולמידה חישובית עם יישומים חשובים בביולוגיה ורפואה. הקורס סוקר אלגוריתמים ושיטות ניתוח עדכניות בתחום כמו גם את יישומיהם למידע ביולוגי בקנה מידה גדול.

אחת מהבעיות שנחקרו בתחום זה היא מציאת קומפלקסים חלבוניים- קבוצה של שניים חלבונים או יותר שביחד מהווים מכונה מולקולרית. את בעיה זו ניתן למדל ע"י מציאת תת-גרף צפוף (קליסטור) מתוך רשת של אינטראקציות חלבון-חלבון.

בעבודה זו נשווה בין מספר שיטות קימות למציאת קומפלקסים. כמו כן נציע שיטה חדשה המבוססת על קליסטור במרחבים מטריים לשם כך נמיר את רשת האינטרציות אל ווקטורים ב בעזרת כלים של למידה ועליהם נבצע את תהליך התיוג.

נשווה בין השיטות הבאות:

1. MCL
2. Clusterone
3. Node2Vec and K-Means

כאשר השיטה השלישית מבוססת על מרחק מטרי.

# Clustering

"קלסטרינג" הינו טכניקה בלמידה חישובית אשר כולל קיבוץ נקודות מידע. בהינתן קבוצה של נקודות מידע, אנחנו יכול להשתמש באלגוריתם קלסטרינג על מנת לתייג כל נקודת מידע נתונה לקבוצה ספציפית המתאימה לה.

בתאוריה, נקודות אשר נמצאות באותה קבוצה צריכות לחלוק תכונות משותפות ואייפיונים דומים, בעוד שנקודות מקבוצות שונות צריכות להיות שונות מאוד בתכונותיהן ובאיפיוניהן.

שיטת הקלסטרינג משויכת ללמידה לא מפוקחת, והינה טכניקה נפוצה מאוד בקרב זיהוי ניתוח מידע סטיסטי בהמון שדות מחקר.

ולכן בתחום הביולוגיה החישובית יש צורך ממשי לזיהוי תכונות דומות של גנים וחלבונים, ומציאת אינטרקציות. כמות המידע אדירה אשר נחשפים אליה הינה גדולה מאוד וברובה המוחלט ללא תיוג.

מכך אנחנו מגיעים לאלגוריתמים של למידה לא מפוקחת ובפרט לקלסטרים.

עם מעט המידע הקיים לנו, קךסטרינג יכול להעניק כח גדול בתיוג המידע החסר.

# תיאור האלגוריתמים

## Markov Cluster Algorithm (MCL)

זהו אלגוריתם ללמידה לא מפוקחת על גרפים (רשתות) והינו מהיר ויציב. האלגוריתם מבוסס על הדמיה של תהליך סטוכסטי על הגרף.

אלגוריתם ה- MCL מוצא מבנה של קלסטרים בגרף בעזרת מנגנון bootstrapping.

התהליך באופן דטרמיניסטי מחשב את ההסתברויות להליכה רנדומית בגרף. הדמיית ההליכה נעשית בעזרת שני פעולות, expansion ו- inflation.

1. Expansion- פעולה אשר מעלה את המטריצה בחזקת כאשר
2. Inflation- פעולה זו גורמת למטריצה לחזור להיות סטוכסטית ע"י חילוק כל איבר בסכום

העמודה וכך זה יוצר בעצם ווקטור התפלגות.

Chart

Description automatically generated

נוסחא המתארת את התהליך הינה:

## Method 2: Clusterone

ClusterONE (Clustering with Overlapping Neighborhood Expansion)

הינו אלגוריתם קלסטרינג לגרפים, אשר מתמודד גם עם גרפים ממושקלים ומייצר קלסטרים בעלי חפיפה ללא קשיים.

שני תכונות אלו שימושיות מאוד בזיהוי קומפלקסים של רשתי אינטרקציות חלבון-חלבון.

ציטוט מן המאמר המתאר את שלבי האלגוריתם:

" Our algorithm consists of three major steps (Online Methods). First, starting from a single seed vertex, a greedy procedure adds or removes vertices to find groups with high cohesiveness. The growth process is repeated from different seeds to form multiple, possibly overlapping groups. Although some overlaps are likely to have biological importance, groups overlapping to a very high extent in comparison to their sizes should likely be merged. In the second step, we quantify the extent of overlap between each pair of groups and merge those for which the overlap score[4](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3543700/#R4) is above a specified threshold. In the third step, we discard complex candidates that contain less than three proteins or whose density is below a given threshold. Note that our method potentially can be used to recognize not only partial overlaps but also cases in which a complex is completely contained in another complex."

## Method 3 : K-Means

אלגוריתם ה-Kmeans מנסה להפריד דגימות ל-n קבוצות בעלות שונות זהה בעזרת פונקציית מיזעור סכום ריבועי המתואם לכל קלסטר שתפורט בהמשך.

האלגוריתם דורש שיספקו לו את מספר הקלסטרים "הרצויים". הוא מותאם היטב לעבודה עם כמות אדירה של מידע, ובשל כך יש בו שימוש נרחב.

האלגוריתם מחלק את הקבוצה ל-K קלסטרים זרים, כל קלסטר "מיוצג" ע"י הממוצע שלו - .

נציין שקודקוד מקור זה אינו בהכרח חלק מהמדגם, אבל הוא שוכן איתו באותו המרחב.

אלגוריתם ה- Kmeans בוחר את קודקוד המקור כך שהוא ממזער את הפונקציה הבאה:

A picture containing text

Description automatically generated

**within-cluster sum-of-squares criterion**:

חסרונות השיטה:

* פונקצית המזעור שאנו משתמשים בה מניחה שהמידע מסודר בצורה קמורה, כאשר זה לא תמיד נכון ולכן מגיב בצורה בנינות לצורות שאינן תואמות הנחה זו.
* ידוע שהנורמה האוקדלידית לא מועילה כאשר אנחנו משתמשים בפונקציה זו על מימדים רבים, נהיית מנופחת מידי.

**Our Algorithm**

אנו רוצים לחקור את האפשרות שאם נפזר במרחב את הגרף בעזרת כלים של רשתות עמוקות, ונפעיל על התוצאה אלגוריתם קלסטרינג מבוסס מרחק גיאומטרי נוכל להסיק יותר מידע מאשר אלגוריתמיי קלטרינג שפועלים על הגרפים.

לשם כך ניסינו לבנות רשת סיאמית אשר מטילה את הגרף במרחב אך לא הצלחנו להתכנס.

לכן השתמשנו ברשת ידועה וקיימת Node2Vec אשר הופכת כל קודקוד בגרף לווקטור במרחב.

על תוצאת הרשת הפעלנו את האלגוריתם K-means אשר פורט קודם לכן.

בהמשך לכך נרצה לטפל בחסרונות השיטה ולכן על מנת לפתור את הסוגיה הראשונה נבצע הרמה למס' גדול של מימדים אך לא מידי כך שלא יתעורר החיסרון השני של שיטה זאת.

# Research Process

השתמשנו במאגר המידע של פונקציות החלבונים : goa\_human.gaf, ובגרף האינטרקציות huri\_symbol.tsv.

עליהם הרצנו את האלגוריתם MCL אשר ממשנו ואת הקוד שלו ניתן למצוא בריפוזיטורי הבא: <https://github.com/haimsaw/YeastEmbedding>

בנוסף, הרצנו על מאגר זה את אלגוריתם ה-ClusterONE אשר קיבלנו כאפילקציה ג'אווית, אשר ניתן למצוא אותה באינטרנט לשימוש חופשי.

קיבלנו את התוצאות הבאות:

עבור ה- MCL:

cluster num of clusters : 404

cluster partitionScore : 10.718850527266893

cluster meanSize : 12.779702970297029

cluster maxSize : 386

cluster minSize: 5

עבור ClusterONE:

cluster num of clusters : 366

cluster partitionScore : 1.080335547273504

cluster meanSize : 3.8934426229508197

cluster maxSize : 48

cluster minSize: 3

במקביל ובאופן בלתי תלוי, מימשנו את האלגוריתם שלנו.

את האלגוריתם שלנו הרצנו עם היפר פרמטרים משתנים אשר ביצענו בעזרתם חקירה של איך המידע מסודר ואיך למנוע בעיות כמו overfit וכדומה.

בדוגמה הראשונה ניתן לראות ש-

ולכן אנחנו מסיקים כי XYZ

לאחר שלמדנו ש XYZ מתקיים אנחנו רוצים לשפר את היכולת הבאה ומכך ניתן לראות בדוגמה השניה שיפור בזה

ובנוסף ניתן לראות תופעה של Overfit עקב עודף נקודות בסיס לאלגוריתם ה-Kmeans.

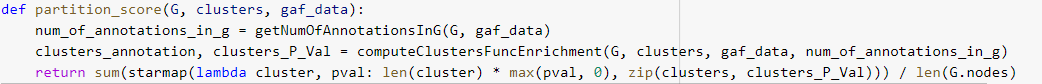
לכן לאחר צמצום אנחנו מקבלים שיפור נוסף:

כעת על מנת להקטין במקביל את הכמות שכל קלסטר מקבל וכן להגדיל ככל הניתן את המינימום איברים בכל קלסטר אנו משנים את אורך ההילוך / ***משהו שחיים צריך להגיד***

ואנו מקבלים את התוצאה הבאה:

ולכן סהכ אנחנו מקבלים שהתוצאה הטובה ביותר שקיבלנו עבור k-mean הינה:

# Result Analysis

את התוצאות אנחנו בוחנים לפי הערך P-Value שהוא מחושב בעזרת הפונקציה הבאה:

מן הנוסחה ניתן להסיק שככל שה- P-Value גדול יותר ככה היכולת לקלסטר טובה יותר.

אותה, ואת הפונקציות הפנימיות שיש בהן צורך, ניתן למצוא בריפוזיטורי שלנו תחת הסעיף : Cluster utils.

ולכן לאחר התבוננות בתוצאות אנחנו מקבלים כי האלגוריתם הבסיסי נותן את הקלסטרים הטובים ביותר והאלגוריתם הגיאומטרי נופל.

האמונה ההתחלתית שלנו אשר מתבוססת על אינטואיציה מתמטית אשר טוענת שחלבונים באים מהעולם שלנו ובו מתקיימת מטריקה, אזי על החלבונים מתקיימת מטריקה ולכן שווה לבדוק פיזור גיאומטרי לא השיגה את התוצאות שרצינו.

שני שאלות שלאחר עבודה זאת נשארו פתוחות וניתן להעמיק ולחקור איזו מטריקה מושרת עדיפה ואם זה מלמד אותנו משהו נוסף על הפיזור של החלבונים בגוף היצור החי? והאם שימוש בכלים כבדים יותר למציאת קלסטרים גיאומטרים יכולים להניב תוצאה טובה יותר?