דוח פרוייקט סיכום קורס ניתוח רשתות ביולגיות

חיים סודאי – 305016875

ענר וולפנפלד – 301785911

דולב דנינו – 205777568

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 מציאת המודולים ברשתות זה היא מציאת קומפלקסים חלבוניים. קומפלקס הינו קבוצה של שני חלבונים או יותר שביחד מהווים מכונה מולקולרית. את בעיה זו ניתן למדל ע"י מציאת תת-גרף צפוף מתוך רשת של אינטראקציות חלבון-חלבון. למרות שבעולם האמיתי חלבונים יכולים להשתתף במספר קומפלקסיים וכן קיימים חלבונים שלא משתתפים באף קומפלקס לרוב נשתמש בשיטות קליסטור שמחלקות את החלבונים לתתי קבוצות.

בעבודה זו נשווה בין מספר שיטות קימות למציאת קומפלקסים וכמו כן, נציע שיטה חדשה המבוססת על קליסטור נקודות במרחבים מטריים. לשם כך נמיר את רשת האינטרציות אל ווקטורים ב בעזרת כלים של למידה ועליהם נריץ אלגורמי קליסטור.

נשווה בין השיטות הבאות:

1. MCL
2. Clusterone
3. Our method - Protein Embedding

# תיאור האלגוריתמים

## אלגוריתם קליסטור מרקובי (MCL)

באלגוריתם זה נתייחס לגרף כעל רשת מרקובית כאשר הסתברויות המעברים הן אחידות מבין השכנים של כל חלבון (או לפי הגודל היחסי של משקלי הקשתות). נוכל להגדיר קלאסטר כקבוצת חלבונים כך שהילוך מקרי מאחד מהם מסתיים בתוך הקבוצה הסתברות גבוה, לכן נוכל לחלק את הגרף לקלאסטרים על ידי הדמייה של הילוך מקרי.

כדי למצוא זאת באופן יעיל נחזור על שני הפעולות הבאות על מטריצת המעברים עד להתכנסות:

1. Expansion- העאלת המטריצה בחזקת כאשר . פעולה זו מדמה הילוך מקרי של צעדים
2. Inflation- העאלת כל איבר בחזקה ונרמול העמודות. פעולה זו מגדילה בכל עמודה את הערכים הגבוהים על חשבון הערכים הנמוכים ולמעשה "מחזקת" את הקלאסטרים הקיימים.

כאשר האלגוריתם מתכנס נקבל מטריצת מעברים כאשר הצמתים מחולקים לקבוצות (קלאסטרים) כך שמכל צומת ניתן להגיע רק לצמתים אחרים באותו הקלאסטר ונוכל להתחייס לכל קלאסטר כזה כעל קומפלקס.

## Clustering with Overlapping Neighborhood Expansion (ClusterONE)

הינו אלגוריתם קלסטרינג לגרפים אשר מתמודד גם עם גרפים ממושקלים ומייצר קלסטרים בעלי חפיפה. עובדה זו שימושיות מאוד בזיהוי קומפלקסים של רשתי אינטרקציות חלבון-חלבון שכן חלבונים יכולים להשתתף ביותר מקומפלקס אחד.

אלגוריתם ה- ClusterONE מורכב משלושה שלבים עיקריים:

1. מתחילים מקודקוד גרעין ובעזרת כלל חמדן מוסיפים או מורידים קודקודים על מנת למצוא קבוצה הדוקה ביותר. תהליך זה חוזר על עצמו ממספר רב של קודקודים ותיתכן חפיפה בין הקבוצות. למרות שחלק מהחפיפות בעלות משמעות ביולוגית, קבוצות שחופפות ברובן צפויות להתאחד.
2. איחוד קבוצות בעלות חפיפה מעל לרף מסויים שמוגדר במאמר.
3. אנו מסירים רכיבים מועמדים בעלי לכל היותר שלושה חלובונים או שהצפיפות שלהם מתחת לרף המוגדר במאמר.

השיטה יכולה לזהות לא רק חפיפה חלקית אלא גם מקרים בהם רכיב מוכל כולו ברכיב אחר.

## Protein Embedding

השיטה המוצעת בעבודה זו היא לנצל אלגוריתמים ידועים לקליסטור. הקלט לאלגוריתמים אלו צריך להיות נקודות בתוך מרחב מטרי (כלומר שקיימת נורמה- פונקציה שמודדת מרחקים בין האובייקטים\קבוצות של אובייקטים). בבעית מציאת המודולים הקלט הוא גרף אינטראקציות ללא מטריקה טבעית מלבד אורך המסלול הקצר ביותר. מטריקה זו מאבדת מידע רב על הקישוריות בגרף ולטעמנו אינה מספיקה. לכן, על מנת שנוכל לנצל בצורה מיטבית את מלוא המידע שנתון ברשת הטמענו את הרשת ב בעזרת כלים של למידה עמוקה.

התנסינו בלמידה בעזרת רשת סיאמית אך זמן הריצה היה ארוך מידיי (ריבועי במספר החלבונים) ולכן השתמשנו בnode2vec. לאחר ההטמעה השתמשנו בk-means כדי לקלסטר את החלבונים המוטמעים.

### Node2vec

[שיטה](https://arxiv.org/pdf/1607.00653.pdf) זו משמשת להטמעה של רשתות במרחב באופן שמשמר תכונות של הרשת המקורית. נרצה שהמרחק של הצמתים המוטמעים קטן אם הצמתים קרובים ברשת. מכיוון שזה לא פיזבלי לאמן על הסביבה של כל צומת נגריל הילוך מקרי באורך קבוע מכל צומת (כאשר את הסתברויות המעברים קובעים בעזרת היפר פרמטרים – פרמטרים אלו קובעים את אופי ההליכה ולכן אילו אספקטים של הרשת נלמד). ניתן לראות באיור כיצד הסתברויות המעברים מחושבות מהיפר הפרמטרים

A picture containing watch, clock

Description automatically generated

כדי ללמוד את ההטמעה- לכל צומת מאתחלים ווקטור רנדומלי ב, מתחייסים לכל הילוך מקרי מצומת כלשהו כדגימה מהסביבה שלו (בדומה למשפטים ב word2vec) ונמזער את המרחק בין הצומת לסביבה שלו ובסוף התהליך נפלוט את הווקטורים לכל הצמתים.

Text

Description automatically generated

פונקציה זו אינה חשיבה, אך בעזרת מספר הנחות מפשטות נקבל את הפונקציה הבאה:

A picture containing text, watch, clock

Description automatically generated

### k-means

אלגוריתם מבוסס טיפוסים לקליסטור של נקודות.

האלגוריתם מחלק את הקבוצה ל-K קלסטרים זרים, כל קלסטר מיוצג ע"י "הטיפוס" שלו- כאשר שייכות של ווקטור לקלאסטר נקבע ע"פ הטיפוס הקרוב ביותר.

כדי למצוא את המרכזים מאותחלים הטיפוסים באופן רנדומי ואז באופן איטרטיבי עד להתכנסות:

* משייכים כל דגימה לטיפוס הקרוב ביותר אליה
* מעדכנים את הטיפוסים להיות המרכז של כל הדגימות המשויכות אליו.

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

אלגוריתם ה- K-means שקול לבחירה של הטיפוסים כך שנמזער את הפונקציה הבאה:

A picture containing text

Description automatically generated

# תהליך העבודה

על מנת להשוות בין השיטות השתמשנו במאגר המידע GO. כאשר סיננו outliers ע"י לקיחת 3-core של הרשת.

תהליך העבודה כלל הגדרת ציון לpartition על מנת שנוכל להשוות בין האלגוריתמים, מימוש של האלגוריתמים (כאשר לclusterone השתמשנו במימוש קיים) וכיוון של היפר-הפרמטרים כדי להשיג תוצאות מיטביות.

## הגדרת הציון

נשווה בין שלושת השיטות ע"י חישוב של העשרה פונקציונאלית בהתבסס על התפלגות היפר גאומטרית. נחשב לכל קלאסטר את הציון כאשר לקלאסטרים עם פחות משלושה חלבונים ניתן ציון 0. ע"מ לקבל ציון לכל החלוקה נייחס לכל חלבון את הציון של הקלאסטר שלו ונמצע את הציונים.

## מימוש האלגוריתמים

ניתן למצוא את המימושים שלנו [בגיטהאב](https://github.com/haimsaw/YeastEmbedding)

כאשר המימושים לאגוריתמים נמצאים בקבצים עם השמות המתאימים (בנוסף למספר קבצי עזר)

את השיטה שלנו הרצנו על גבי [google collab](https://colab.research.google.com/drive/1bWiEjjkkv4leswz_wqLL2Ryxqg3AUn2U?usp=sharing) בעזרת GPU

## כיוון היפר-פרמטרים

בשיטת ה Protein Embeddingישנם מספר רב של היפר-פרמטרים שנדרש לכוון בנינו תשתית שמאפשר להריץ מספר טסטים ולבחון את ההשפעה של הפרמטרים על הציון.

הרצנו ניסויים כאשר מכל צומת הגרלנו 10 מסלולים באורך 20.

הפרמטר הראשון שנדרש לכיוון הוא מספר מחזורי הלימוד. בגרף הבא ניתן לראות את הloss של ההטמעה עבור כל epoch:

Histogram

Description automatically generated

ניתן לראות כי לאחר 20 איטרציות הגרף מתיישר – כלומר השינויים בהטמעה הם זניחים ומספיק להריץ 20 איטרציות.

לאחר מכן בחנו את השפעה של (מימד ההטמעה והסתברויות ההילוך המקרי) על ההעשרה הפונקציונאלית:

A picture containing square

Description automatically generatedSquare

Description automatically generated with medium confidenceLogo

Description automatically generated with medium confidence

ניתן לראות כי למספר המימדים יש השפעה על הציון וכן יש נטייה לp נמוך וq גבוה (מה שתואם את האינטואיציה – נרצה לחקור את הסביבה של הצומת ולכן לא נרצה לבקר באותה הצומת פעמיים, כמו כן נרצה שההילוך המקרה יידמה bfs וישמר את המבנה ההומופילי של הרשת).

נשים לי כי הציונים המתקבלים הם גבוהים מאוד (מתאימים ל p-val של 4.24e-18). בחינה יותר מדוקדקת של הקלאסטרם מגלה כי מרבית הקלאסטרים הם בגודל קטן מ3 (184) וישנם קלאסטרים ספורים שמכילים את רוב הרשת (שני הקלאסטרים הכי גדולים מכירים 4513 חלבונים). נתונים אלו הם סממן לoverfitting – כאשר מספר המימדים גבוה לאלגוריתם הקליסטור יש יותר מרחב פעולה ומקבלים overfit. כדי להימנע מתופעה זו הגבלנו את מספר המימדים ל128.

וקיבלנו

## תוצאות

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Alg name** | **num of clusters** | **Score** | **Mean Cluster Size** | **max cluster size** | **min cluster size** |
| MCL | 404 | 10.7 | 12.7 | 386 | 5 |
| ClusterONE | 366 | 1.08 | 1.9 | 48 | 3 |
| Protein Embedding (2048) | 198 | 44.8 | 24.3 | 3448 | 1 |
| Protein Embedding (??) |  |  |  |  |  |

## שאלות פתוחות

השאלה הפתוחה העיקרית שעולה מעבודה זו היא כיצד ניתן להימנע מoverfit. ע"י שימוש בכח מחשוב חזק יותר ניתן היה לדגום יותר הילוכים מקריים מכל צומת וכן הילוכים מקריים ארוכים יותר שיספקו יותר מידע על מבנה הרשת.

כמו כן, מדד ההעשרה הפונקציונאלית ייצר הטייה עבור יצירת קלאסטרים מועשרים ללא התייחסות לקישוריות שלהם ברשת המקורית. שימוש במדדים אחרים לעמידה של איכות הקלאסטרים (לדוגמה Clustering Coefficient) יכולה לנטרל את ההטייה הזו ולאפשר לנו לכוון את הפרמטרים בצורה מיטבית.

לסיום, למרות שתנסינו באלגוריתמי קלאסטרינג נוספים (Affinity Propagation, DBSCAN) מכיוון שההטמעה הייתה מוטה הם הניבו תוצאות לא טובות. פתרון של בעיית ה overfitting עשוי לאפשר תוצאות טובות יותר.