מבוא לגנומיקה חישובית ומערכתית

שבוע #4

- אתגר הקורס

האתגר

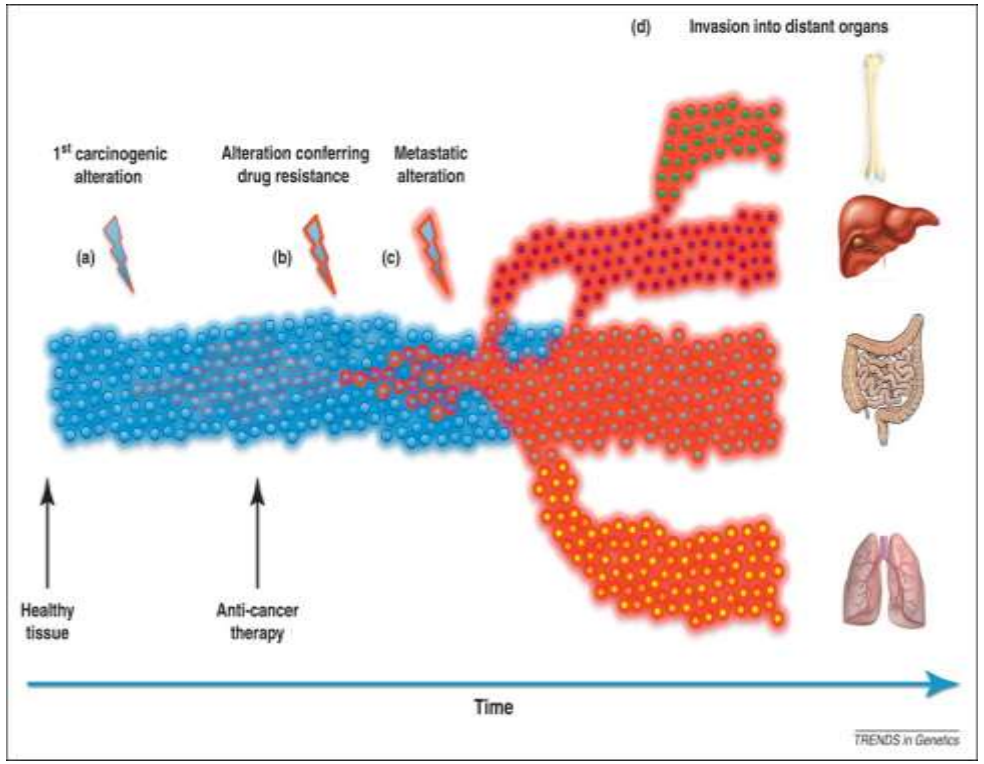
א. ליצור מסווג בין שני סוגי סרטן (LUSC ו-HNSC) המשתמש במידע גנומי (מוטציות) המגיע מ-100 גנים הקשורים לסרטן.

ב. ליצור מסווג בין שני סוגי סרטן המשתמש במידע גנומי **ורמות מתילציה** מאותם 100 גנים הקשורים לסרטן.

משימות!

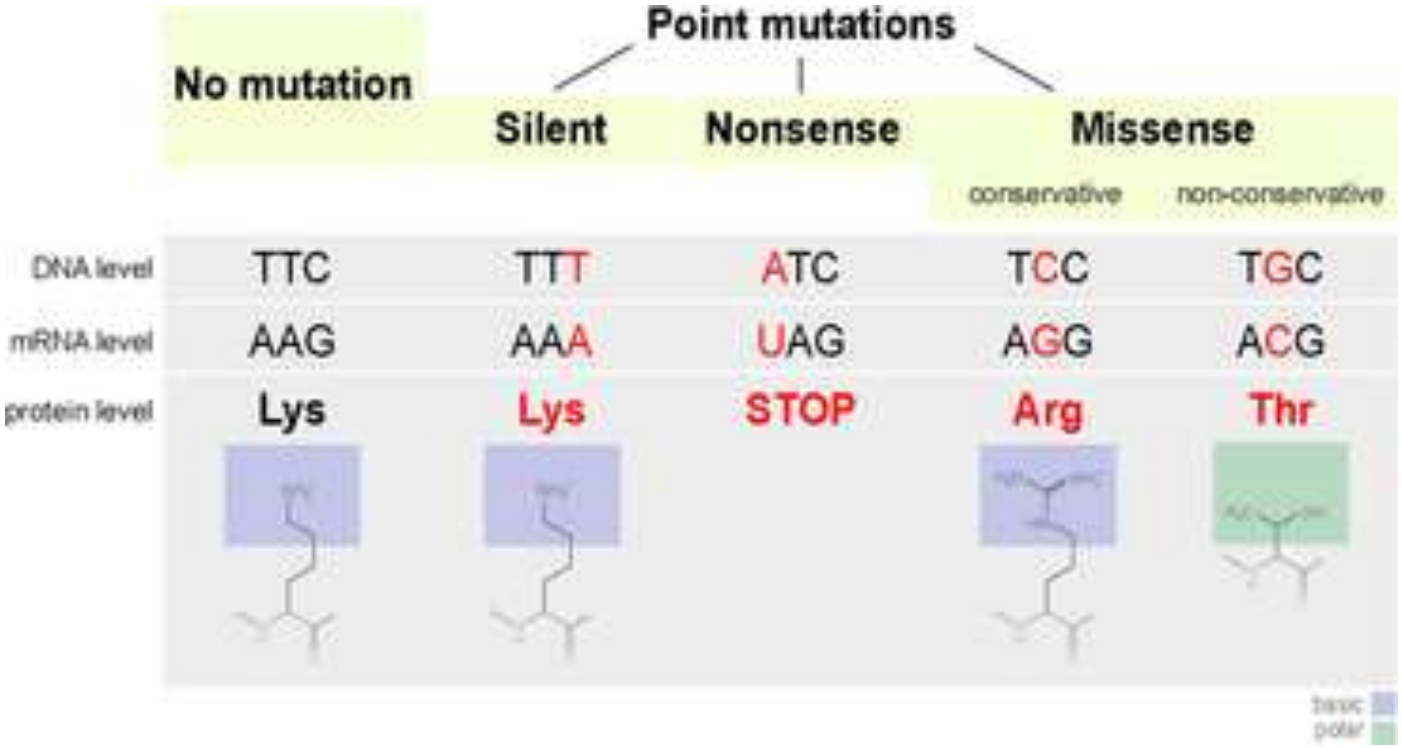
- **ללמוד על הרעיון של feature selection**
- **ללמוד על overfitting**
- **לקבל תחושה של לעבוד עם "דאטא גנומי גדול"**
- תחושה ראשונית של עבודה עם מוטציות באיזורים שונים בגן

מבוא לאבולוציה של סרטן

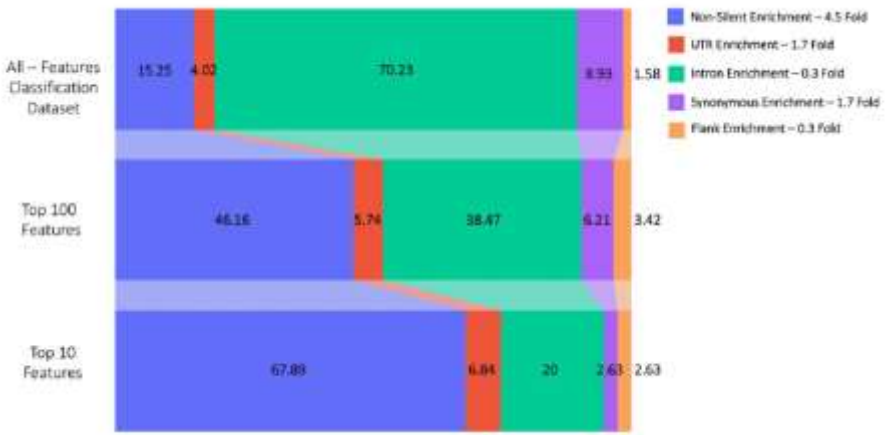


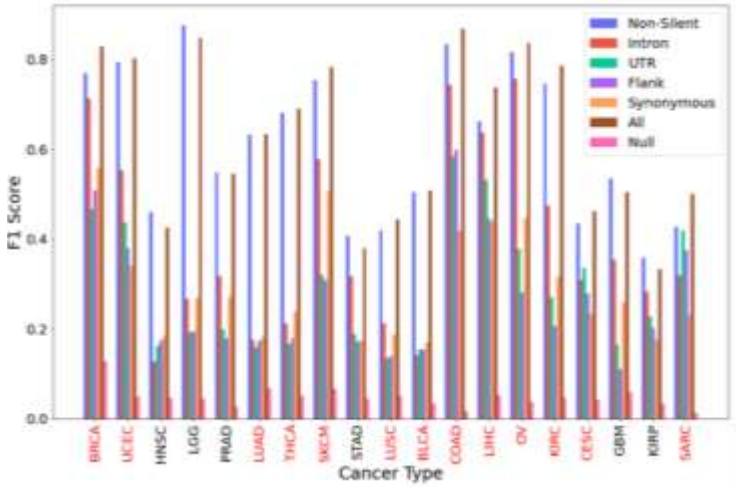
Oncogene – overexpression leads to cancer
Tumor Suppressor Gene – loss of function leads to cancer

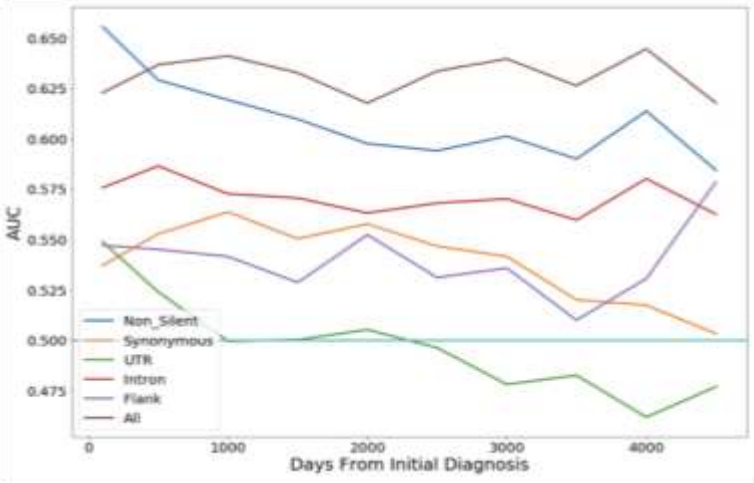
סוגים שונים של מוטציות



מוטציות שקטות פרדקטיביות!



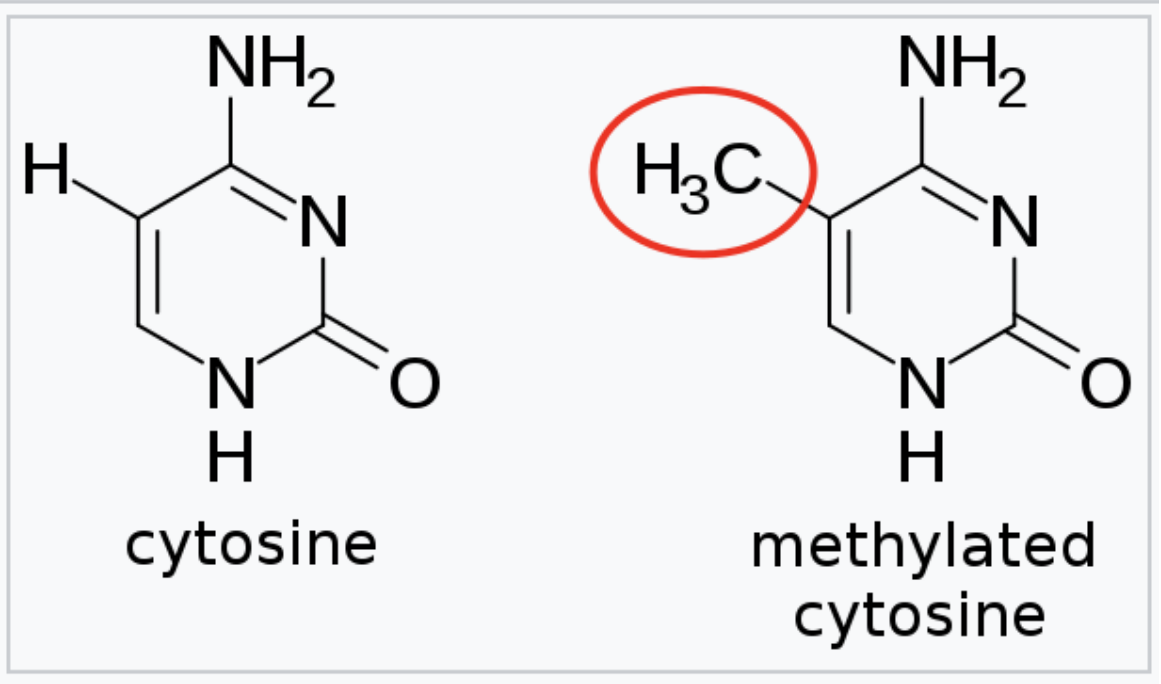




מוטציות - הדאטא

case_id	Label	Gene_name	Chromosome	Start_Position	End_Position	Mut_Strand	Variant_Classification	Reference_Allele	Tumor_Seq_Allele1	Tumor_Seq_Allele2
13f63218-	2	TP53	chr17	7673728	7673728	+	Nonsense_Mutation	C	C	A
13f63218-	2	BRCA2	chr13	32339828	32339828	+	Missense_Mutation	G	G	C
13f63218-	2	APC	chr5	112838253	112838253	+	Missense_Mutation	A	A	T
13f63218-	2	ATM	chr11	108268379	108268379	+	Intron	C	C	G
13f63218-	2	ATM	chr11	108353791	108353791	+	Missense_Mutation	C	C	G

מתילציה

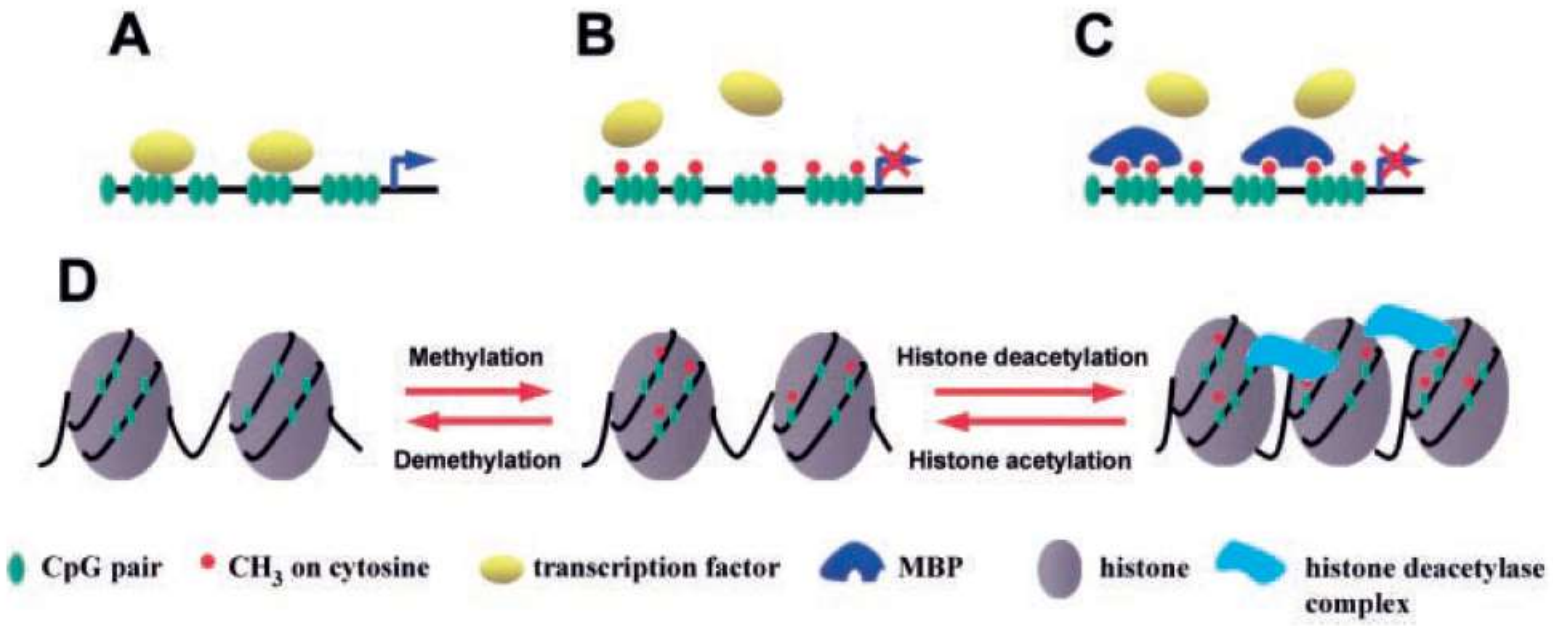


CpG Islands

לדוגמא:

…GCGCGCGTTGCGCGCGCGGCC…

מתילציה מדכאת שעתוק



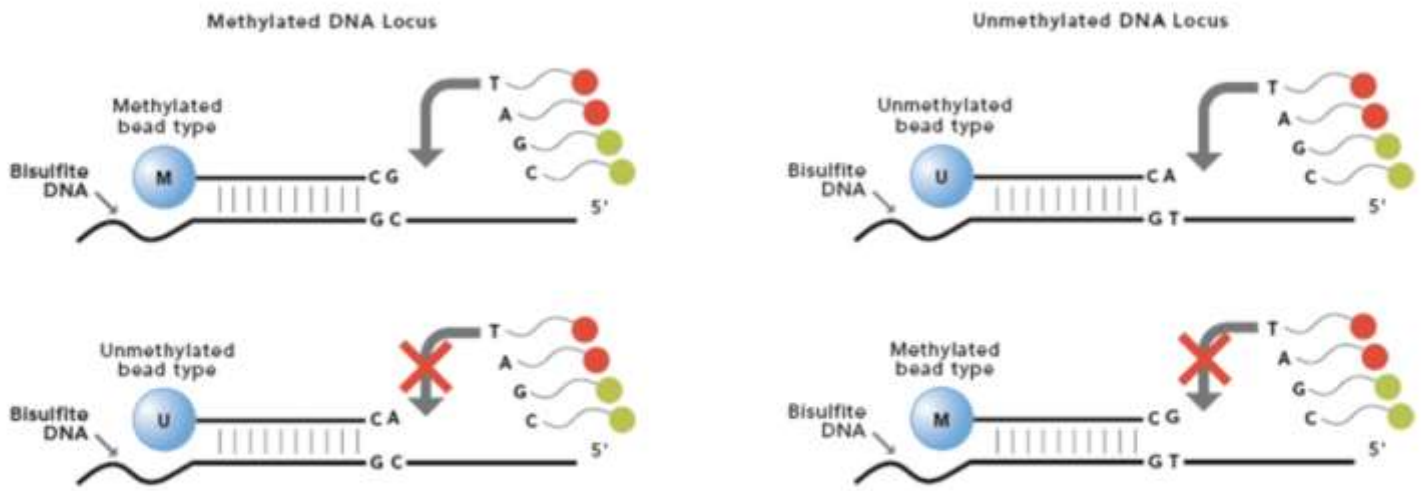
מתילציה - הדאטא

- טיפול עם ביסולפט הופך C (ציטוזין) *ללא מתליציה* ל-T (טימין), בעוד C *עם מתילציה* נשאר אותו דבר
- פרובים של DNA מוכוונים לאתרים ספציפים ב-CpG islands, ונקבל יותר פעמים C (מאשר T) אם היה האתר לרוב עבר מתילציה ויותר פעמים T אם הוא היה לרוב בלי מתילציה.
- M זה מדד לכמות הפעמים שהיה מתילציה, U זה מדד לכמות הפעמים שלא היה מתליציה:

$$\beta := \frac{\mathrm{M}}{\mathrm{M}+\mathrm{U}} \in [0,1]$$

מתילציה - הדאטא

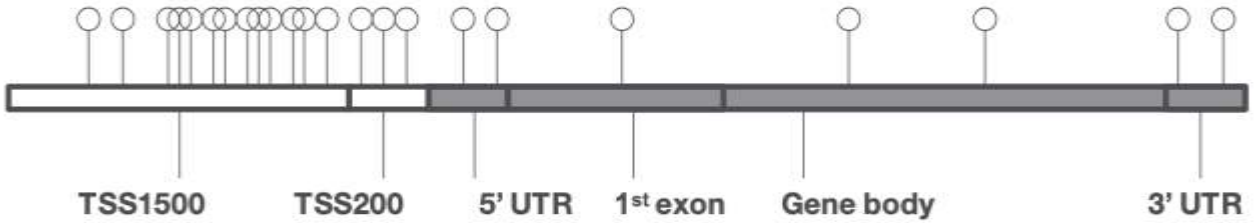
Infinium Methylation Assay



The Infinium Methylation Assay uses two different bead types to detect CpG methylation. The U bead type matches the unmethylated CpG site; the M bead type matches the methylated site. In the top figure, the unmethylated CpG target site matches with the U probe, enabling single-base extension and detection. It has a single-base mismatch to the M probe, which inhibits extension. If the CpG locus of interest is methylated (bottom figure), the reverse occurs.

מתילציה - הדאטא

Figure 2: HumanMethylation450 BeadChip Provides Coverage Throughout Gene Regions



Feature Type	Genes Mapped	Percent Genes Covered	Number of Loci on Array
NM_TSS200	14895	0.79	2.56
NM_TS1500	17820	0.94	3.41
NM_5'UTR	13865	0.78	3.34
NM_1stExon	15127	0.80	1.62
NM_3'UTR	13042	0.72	1.02
NM_GeneBody	17071	0.97	8.97
NR_TSS200	1967	0.65	1.84
NR_TSS1500	2672	0.88	2.92
NR_GeneBody	2345	0.77	5.34

מתילציה - הדאטא

case_id	Label	probeID	beta_val	CpG_chrm	CpG_beg	CpG_end	probe_stra	matching_genes
01420c4e-6013-4d3f-87bd-a6cd9e3f8e87	2	cg00008446	0.29792	chr1	156845981	156845983	-	NTRK1
01420c4e-6013-4d3f-87bd-a6cd9e3f8e87	2	cg00016156	0.05066	chr5	171388178	171388180	+	NPM1
01420c4e-6013-4d3f-87bd-a6cd9e3f8e87	2	cg00022858	0.08044	chr4	54230448	54230450	-	PDGFRA
01420c4e-6013-4d3f-87bd-a6cd9e3f8e87	2	cg00031759	0.073481	chr13	32315396	32315398	+	BRCA2

פיצ'רים מוכנים

- כמות המוטציות בכל גן (לכל חולה) עבור 100 הגנים שנבחרו

אתם תצטרכו לייצר פיצ'רים נוספים! עבור מטלת המטילציה תצטרכו ליצור את כל הפיצ'רים

פרטים על המטלה

- המטלה להגשה עד ה-10.6 (10% מהציון הסופי)
- יש מסמך מפורט במודל מה נדרש להגשה
- אפשר לעשות לבד או בזוגות\שלשות
- ציון 100 עבור דו"ח משכנע... רוצים לראות ניסיון אמיתי!
- הפרדיקציות של הקבוצות יושוו לתיוג האמיתי ואנחנו נחשב את השגיאה שלכם על הטסט
- בונוס למנצחים בתחרות!
- המנצחים בתחרות כנראה יסבירו על השיטות שלהם בכיתה

שימו לב!

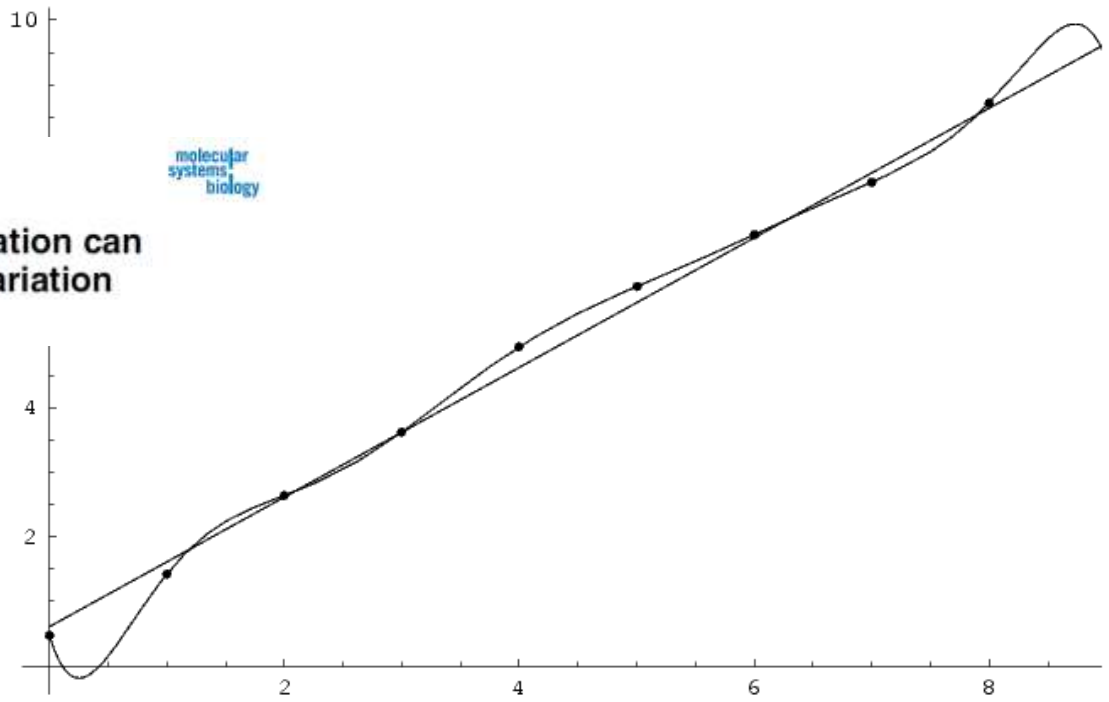
- Overfitting
- הדאטא רועש\biased
- אתם יכולים (וצריכים) לחשוב על פיצ'רים משלכם
- אתם יכולים להשתמש במודלים סטטיסטיים מסובכים, אבל מהניסיון שלנו קשה לנצח מודל פשוט + ידע בביולוגיה + ידע בסטטיסטיקה\overfitting

Overfitting

Molecular Systems Biology 6: Article number 400; doi:10.1038/msb.2010.59
Citation: Molecular Systems Biology 6:400

www.molecularsystemsbiology.com

Sequence signatures and mRNA concentration can explain two-thirds of protein abundance variation in a human cell line



Noisy (roughly linear) data is fitted to both linear and polynomial functions. Although the polynomial function passes through each data point, and the linear function through few, the linear version is a better fit. If the regression curves were used to extrapolate the data, the overfit would do worse.

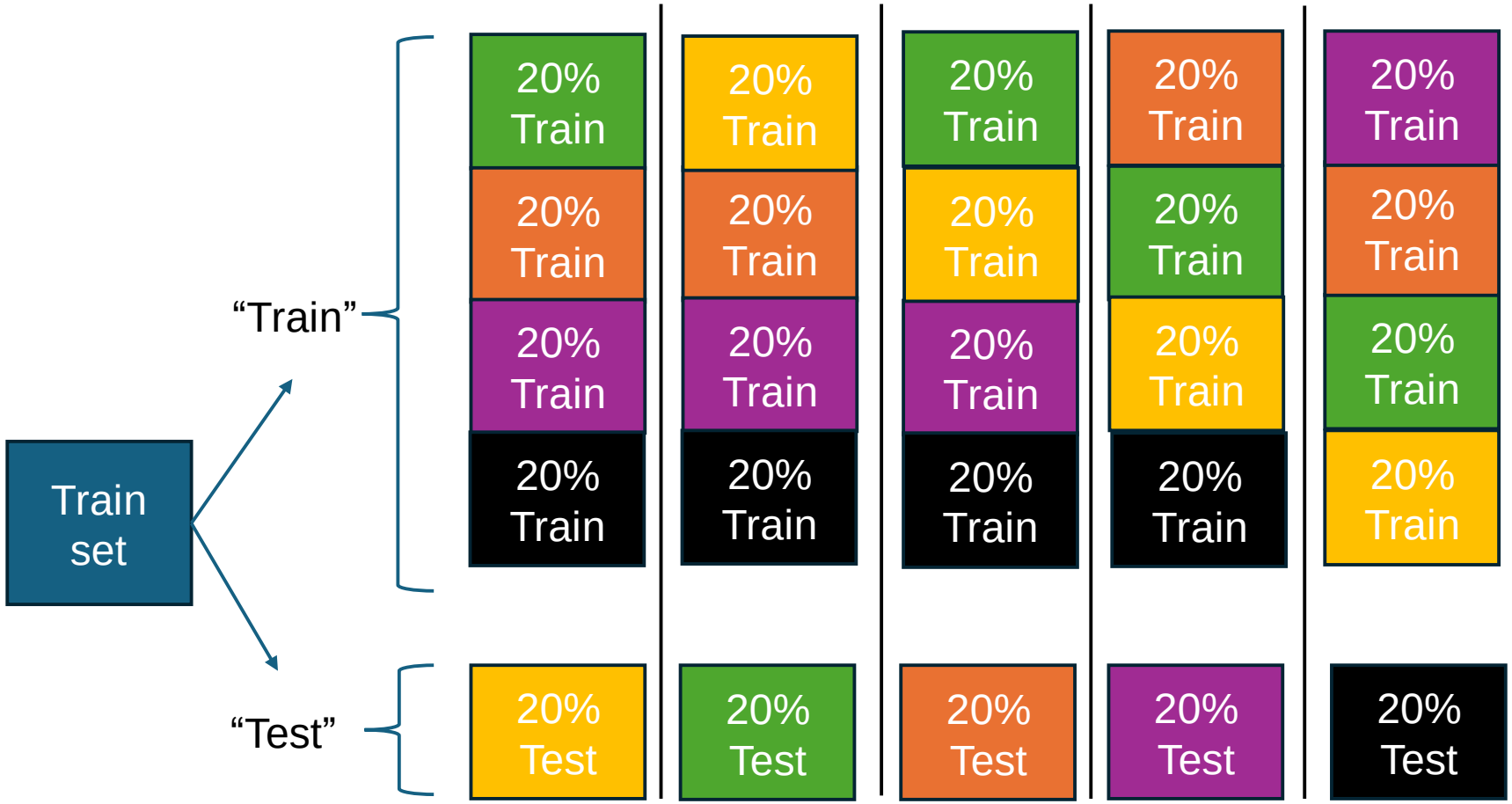
Any idea how to deal with it ?

איך להתמודד עם Overfitting

- Cross validation – evaluation via train/test..
- Regularization -- penalizing models with too many features
- Pruning – a phase of increasing the modeling after “growing” it
- Minimum description length – optimize the log-likelihood + the model complexity
- Adjusted correlation.

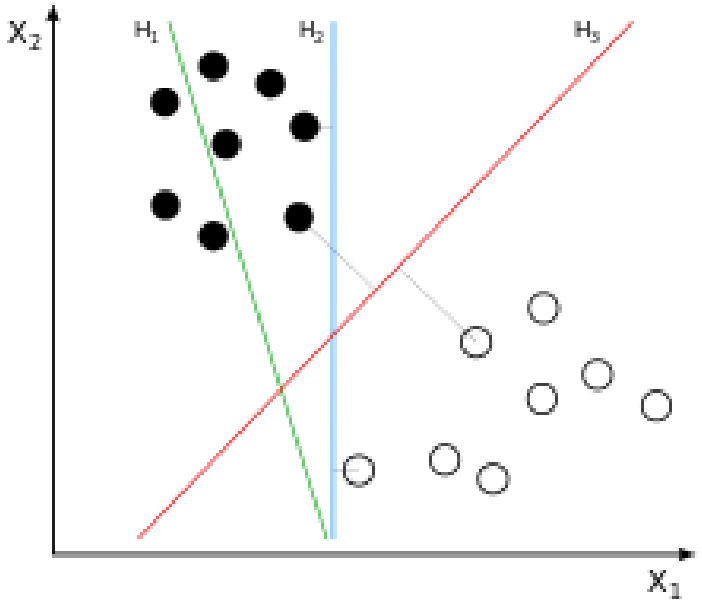
The approaches are strongly related

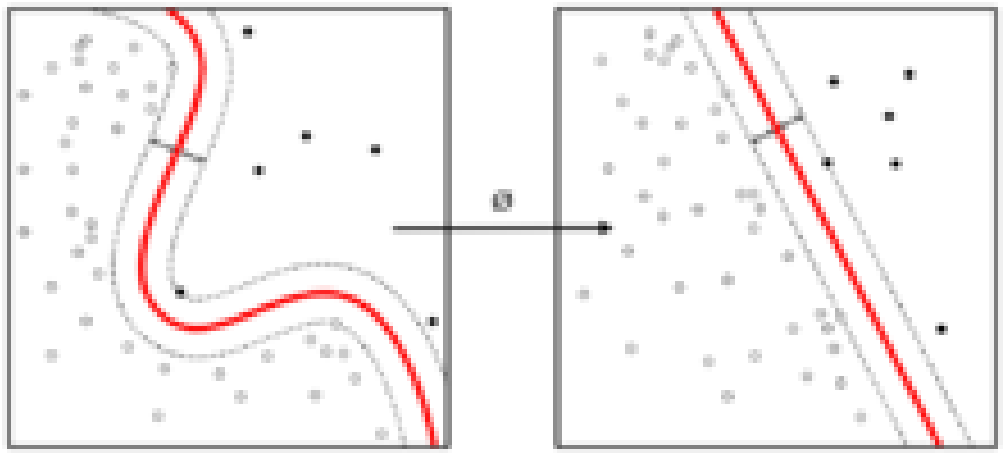
Cross validation



- הרעיון זה לשערך את השיפור דרך הטסט ולא דרך ה-train

Support Vector Machine – SVM





בהמשך...

הצעה ראשונית לשלבים במשימה

1) לייצר הרבה פיצ'רים

2) להשתמש בדאטא הידוע (סט האימון) כדי לעשות בחירת פיצ'רים ולמצוא את משקולות המודל

3) להשתמש במודל על הדאטא הלא ידוע (סט הטסט) כדי לחזות את סוג הסרטן

דוגמא

- known:

סוג סרטן

G1 ATG AAA GAG TCT CGC CCC TAG CCA .. C1

G2 ATT AAA GAG TTT CGC CCC TAG CCA .. C1

G3 ATG AAA GAG TTT CGC CCC TAG CCA .. C2

G4 ATG AAA GAG TTT CGC CCC TAG CCA .. C1

G5 ATT AAA GAG TCT CGC CCC TAG CCA .. C2

G6 ATG AAA GAG TCT CGC CCC TAG CCA .. C2

G7 ATG AAA GAG TTT CGC CCC TAG CCA .. C1

G8 ATG AAA GAG TTT CGC CCC TAG CCA .. C2

- Unknown:

G9 ATG AAA GAG TCT CGC CCC TAG CCA ..

G10 ATT AAA GAG TTT CGC CCC TAG CCA ..

1) לייצר פיצ'רים

Mutation G->T in position 3

Mutation T->C in position 11

- known:

G1 ATG AAA GAG TCT CGC CCC TAG CCA ..	0	1	…
G2 ATT AAA GAG TTT CGC CCC TAG CCA ..	1	0	
G3 ATG AAA GAG TTT CGC CCC TAG CCA ..	0	0	
G4 ATG AAA GAG TTT CGC CCC TAG CCA ..	0	0	
G5 ATT AAA GAG TCT CGC CCC TAG CCA ..	1	1	
G6 ATG AAA GAG TCT CGC CCC TAG CCA ..	0	1	
G7 ATG AAA GAG TTT CGC CCC TAG CCA ..	0	0	
G8 ATG AAA GAG TTT CGC CCC TAG CCA ..	0	0	

Many additional features

- Unknown:

G9 ATG AAA GAG TCT CGC CCC TAG CCA ..

G10 ATT AAA GAG TTT CGC CCC TAG CCA ..

Feature Selection (2

- Divide the known set to 3 subsets:

סוג סרטן

Known Train (40%):

G1 ATG AAA GAG TCT CGC CCC TAG CCA .. C1

G2 ATT AAA GAG TTT CGC CCC TAG CCA .. C1

G3 ATG AAA GAG TTT CGC CCC TAG CCA .. C2

Known Validation (40%):

G4 ATG AAA GAG TTT CGC CCC TAG CCA .. C1

G5 ATT AAA GAG TCT CGC CCC TAG CCA .. C2

G6 ATG AAA GAG TCT CGC CCC TAG CCA .. C2

Known Test (~20%):

G7 ATG AAA GAG TTT CGC CCC TAG CCA .. C1

G8 ATG AAA GAG TTT CGC CCC TAG CCA .. C2

מצאו את הפיצ'ר הראשון הכי טוב בהתבסס על הסט "ולידציה" (שבחרתם מתוך הסט אימון)

בדקו את כל הפיצ'רים לחוד:

- אמנו SVM עם פיצ'ר בודד (בהמשך נסביר יותר לעומק מה זה)
- בדקו את אותו SVM על סט ה-ולידציה (מתוך סט ה-Known) ע"י חישוב השגיאה

* בחרו את הפיצ'ר עם הביצועים הכי טובים על סט ה"ולידציה". למשל:

		סוג סרטן	חיזוי סוג סרטן
G4	ATG AAA GAG TTT CGC CCC TAG CCA ..	C1	C1
G5	ATT AAA GAG TCT CGC CCC TAG CCA ..	C2	C1
G6	ATG AAA GAG TCT CGC CCC TAG CCA ..	C2	C2

שגיאה = 33% = E1

מצאו את הפיצ׳ר **השני** הכי טוב בהתבסס על הסט "ולידציה" (שבחרתם מתוך הסט אימון)

נניח כי הפיצ'ר הראשון הכי טוב היה Mutation G->T in position 3 (שנגדיר בתור "פיצ'ר 1") שנותן את השגיאה הכי נמוכה (E1)

עכשיו נחפש את הפיצ'ר הבא. עבור כל פיצ'ר שהוא לא פיצ'ר 1:

- נאמן SVM על סט האימון (כמו מקודם) שהפיצ'רים שלו הם פיצ'ר 1 והפיצ'ר הנבדק
- נחשב את השגיאה של אותו SVM על סט ה-"ולידציה" (מתוך סט הKnown)

מצאו את הפיצ׳ר **השני** הכי טוב בהתבסס על הסט "ולידציה" (שבחרתם מתוך הסט אימון)

נניח שהפיצ'ר השני הכי טוב היה Mutation T->C in position 11 ("פיצ'ר 2") שנותן את השגיאה הכי הקטנה על סט ה-"ולידציה" (מסט הKnown) והערך שלה הוא E2

רק אם E2<E1 נמשיך את החיפוש אחר פיצ'ר שלישי. אחרת נעבור לשלב האחרון ב-Feature Selection.

מצאו את הפיצ׳ר השלישי הכי טוב בהתבסס על הסט "ולידציה" (שבחרתם מתוך הסט אימון)

בואו נניח שאכן E2 קטן מ-E1. אז נחפש את הפיצ'ר השלישי בדומה לפעם הקודמת:

עבור כל פיצ'ר שהוא לא פיצ'ר 1\2:

- נאמן SVM על סט האימון (כמו מקודם) שהפיצ'רים שלו הם פיצ'ר 1, פיצ'ר 2 והפיצ'ר הנבדק
- נחשב את השגיאה של אותו SVM על סט ה-"ולידציה" (מתוך סט הknown)

מצאו את הפיצ'ר השלישי הכי טוב בהתבסס על הסט "ולידציה" (שבחרתם מתוך הסט אימון)

נניח הפיצ'ר השלישי הכי טוב הוא Mutation T->G in position 1000, וש-הSVM שמשתמש בו (ובפיצ'ר 1 ו-2) נותן את השגיאה E3.

רק אם E3<E2 נמשיך את החיפוש אחר פיצ'ר רביעי. אם E3≥E2 אז נעצור את החיפוש ונעבור לשלב האחרון ב-Feature Selection.

שלב אחרון – Feature Selection

נניח כי עצרנו אחרי שניסינו לחפש פיצ'ר שלישי ויצא לנו שעבור הפיצ'ר השלישי הכי טוב מתקיים כי E3≥E2, כלומר הSVM עם פיצ'רים 1,2,3 נותן שגיאה גדולה יותר מאשר ה-SVM עם פיצ'רים 1,2.

אז ניקח את הSVM עם אותם 2 פיצ'רים ונחשב את השגיאה שלו על הסט ה-"טסט" מתוך הסט הknown.

Known Test

G9 C1’

G10 C2’

שלב אחרון – Feature Selection

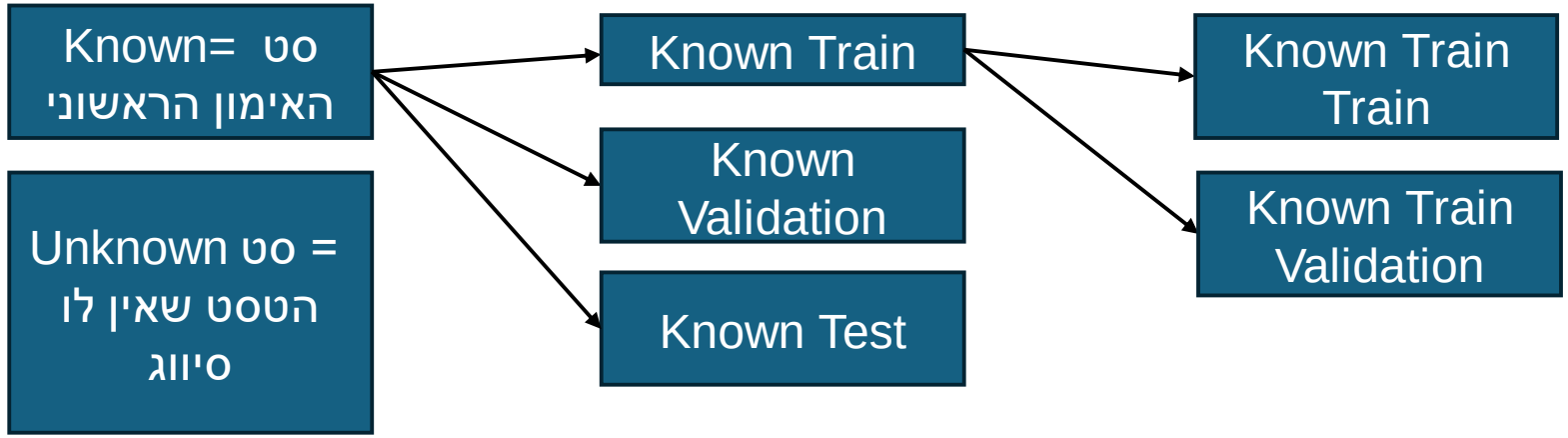
אם השגיאה על סט ה"טסט" הknown יצאה קטנה (או דומה) לשגיאה שקיבלנו על סט ה"ולידציה" מהסט הknown והיא מספיק טובה בשבילנו – ניקח את המודל הזה ואיתו נחזה את הסט Unknown.

אם השגיאה לא טובה על סט ה"טסט" מסט הknown, נסו להשתמש בשיטה אחרת לבחירת פיצ'רים שאולי תוביל לפחות פיצ'רים והיעזרו בשיטות שהוזכרו כדי להתמודד עם overfitting (תקראו עליהן).

*יכול להיות פשוט שהסיווג הוא קשה מדי עם הדאטא הנוכחי... (בצ'אלנג' הנוכחי זה לא ככה ☺)

ניואנס נוסף – Feature Selection

אם למודל שלנו יש היפר-פרמטרים שאנחנו רוצים לבחור כדאי לפצל את הknown train גם לknown train train וknown train validation ואז לאמן על הknown train train ולבדוק את השגיאה המתקבלת עם אפשרויות שונות של ההיפר פרמטרים על ה known train validation (ואז לבחור את ההיפרפרמטרים הכי טובים). לפעמים לא אפשרי כי אין מספיק דאטא.



Cross Validation + Feature Selection

דרך טובה להימנע מ-overfitting זה לעשות קרוס ולידציה (פיצול למשל ל-5 Known ו-Unknown) ואז בכל פיצול לעשות על הKnown את ה Feature Selection.

המודל הסופי יכול להיות:

א. הצבעה משותפת של כל המודלים מהקרוס ולידציה

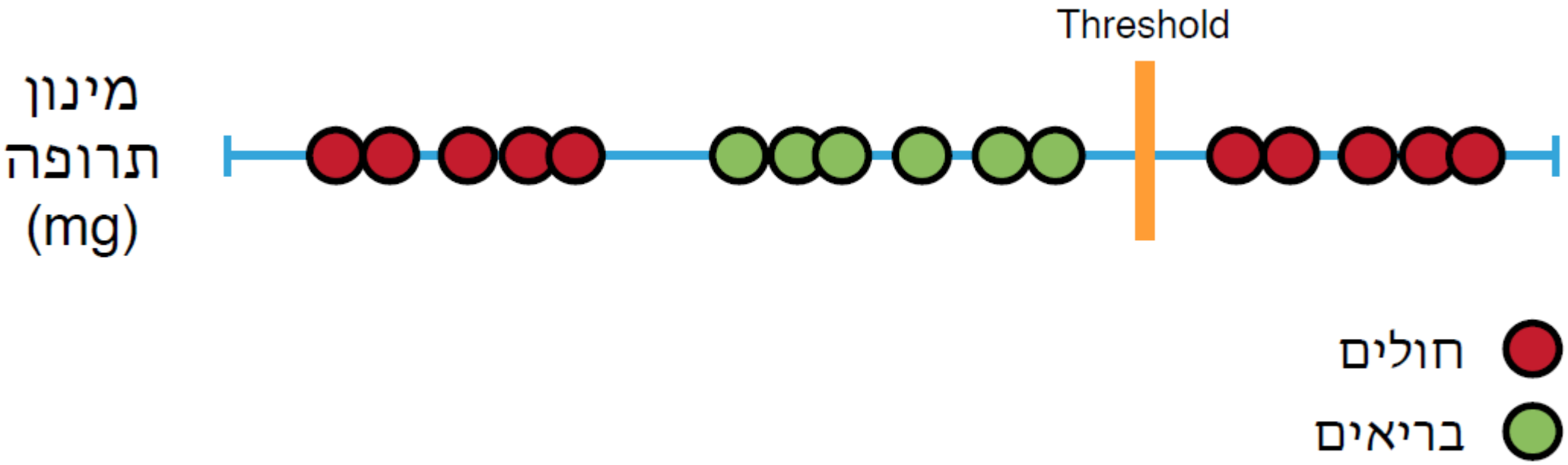
ב. המודל הכי טוב מבין כל המודלים מהקרוס ולידציה

ג. מודל שאומן על כל הדאטא עם הפיצ'רים+היפר פרמטרים הכי נפוצים בכל מודל מהקרוס-ולידציה (זה עלול להיות בעייתי... למה?)

Support Vector Machine – SVM

- אלגוריתם של Machine Learning מסוג Supervised
- משמש לקלסיפיקציה (סיווג או לפעמים נקרא מיון)
- ניתן לסווג איתו דאטא שלא ניתן להפרדה באופן לינארי

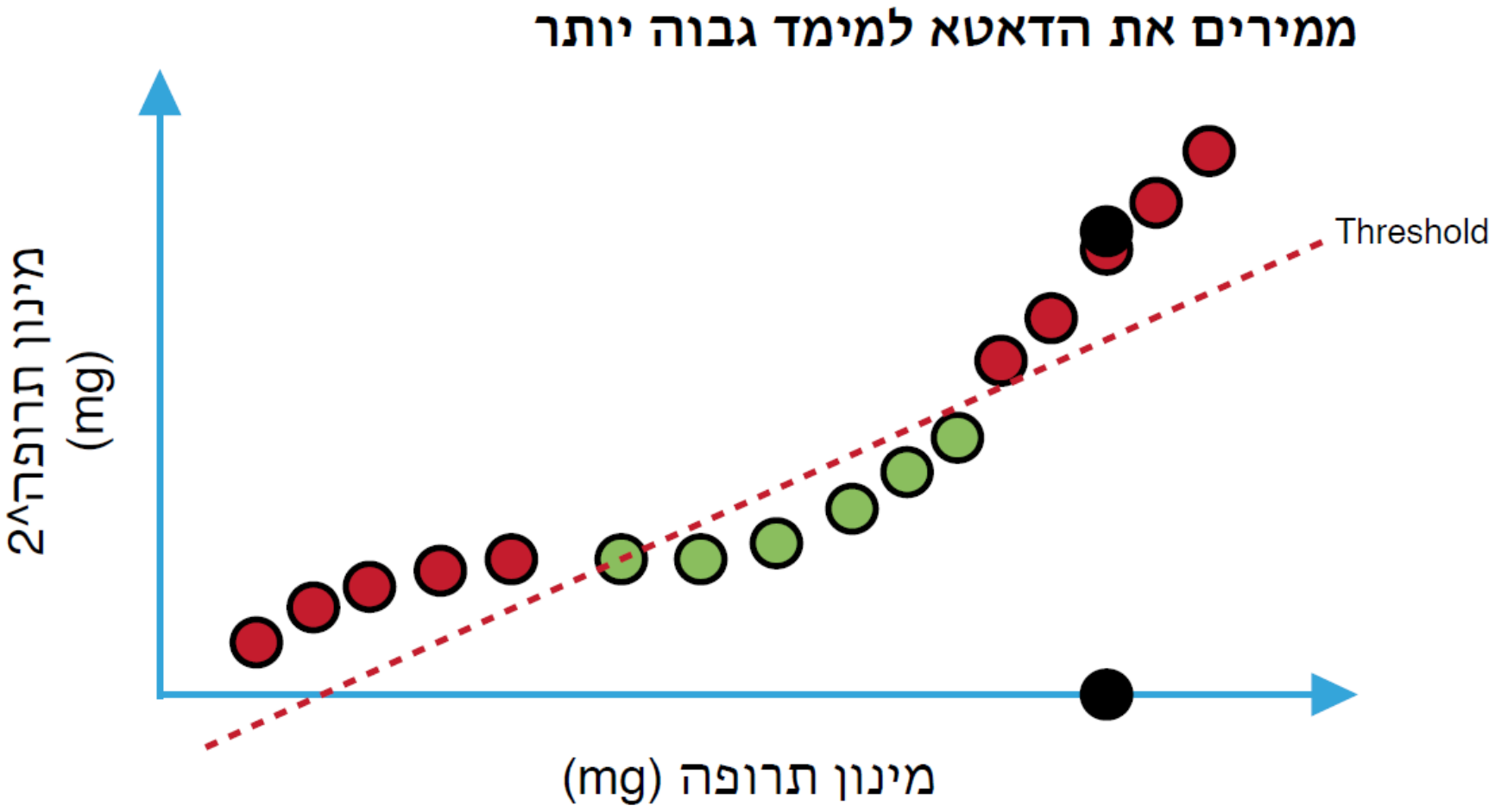
Support Vector Machine – SVM



האם ניתן למצוא סף טוב יותר?

כן!!

Support Vector Machine – SVM



Support Vector Machine – SVM

Kernel functions

רדיאלי

פולינום

קרוס-ולידציה

(כדי לבחור את ההיפר-פרמטר הכי טוב)

האתגר – הצעה לתהליך בסיכום

1. Feature Generation - יצרו פיצ'רים חדשים (תחשבו על רעיונות, תקראו מאמרים, באינטרנט)
2. Cross Validation - פצלו את סט האימון שנתון לכם ל-K פיצולים (תנסו כמה אופציות ותמשיכו עם הצעד הבא כל פעם) של "Known"(ה-"Unknown" תמיד אותו דבר)
3. Feature Selection – עשו את Feature Selection כלשהו על סט הKnown (למשל כפי שתואר בהרצאה)
4. השתמשו במודל סופי המשתמש בפיצ'רים (והיפר-פרמטרים) שמצאתם כדי לסווג את הסט TEST (שאין לכם את הסיווג שלו). המודל יכול להיות למשל הצבעה המשותפת של המודלים שנבנו בקרוס ולידציה

Feature Generation

Case id	Gene name	chromosome	Start position	End position	Reference Allele	Tumor Seq Allele	Variant classification
Yossi	IDH1	1	300	300	G	A	5'UTR
Yossi	BRCA1	3	350	351	AT	--	Frame shift deletion
Hanna	IDH1	1	303	303	T	C	5'UTR
Leah	TP53	7	715	715	C	G	Intron
Leah	IDH1	1	300	300	G	A	5'UTR

Feature Generation

פיצ׳ר אפשרי: האם יש לפציינטים את המוטציה הספציפית או לא (צריך גם להגדיר מה זו מוטציה ספציפית בעינינו? אותו מיקום? בדיוק אותו שינוי באותו מיקום? כאן בדוגמא- בדיוק אותו שינוי של נוקלאוטידים באותו המיקום)

Chr7 – 715:715 – C to G	Chr1 – 303:303 – T to C	Chr3 – 350:351 – -- AT to	Chr1 – 300:300 – G to A	Patient/ Feature
0	0	1	1	Yossi
0	1	0	0	Hanna
1	0	0	1	Leah

חלוקת הדאטא (פיצול יחיד)

```
df_known_temp, df_known_test = train_test_split(
    df_known,
    test_size=0.2,
    stratify=df_known['Label'],
    random_state=0)
df_known_train, df_known_validation = train_test_split(
    df_known_temp,
    test_size=0.5,
    stratify=df_known_temp['Label'],
    random_state=0)
print(f'Known Train {len(df_known_train)}:',Counter(df_known_train.Label))
print(f'Known Validation {len(df_known_validation)}:',Counter(df_known_validation.Label))
print(f'Known Test {len(df_known_test)}:',Counter(df_known_test.Label))
```

```
Known Train 257: Counter({2.0: 130, 1.0: 127})
Known Validation 258: Counter({2.0: 131, 1.0: 127})
Known Test 129: Counter({2.0: 65, 1.0: 64})
```

Cross Validation

```
from sklearn.model_selection import train_test_split
def split_data(df, rand_i):
    # Function that splits data into known train,validation,test
    ### Input:
    # df - Dataframe of known data
    # rand_i - random state setter
    ### Outout:
    # df_known_train, df_known_validation, df_known_test - Dataframes of known train, validation, test
    df_known_temp, df_known_test = train_test_split(df_known,
        test_size=0.2,
        stratify=df_known['Label'],
        random_state=rand_i)
    df_known_train, df_known_validation = train_test_split(df_known_temp,
        test_size=0.5,
        stratify=df_known_temp['Label'],
        random_state=rand_i)
    return df_known_train, df_known_validation, df_known_test
```

Cross Validation

```
k_splits = 5
for i in range(k_splits):
    df_known_train, df_known_validation, df_known_test = split_data(df, i)
    #Feature Selection
```

Feature Selection

```
from sklearn.svm import SVC
feats_chosen = ['Mutations_in_ABL1', 'Mutations_in_AKT1']
X_train = df_known_train[feats_chosen]
y_train = df_known_train['Label']
X_val = df_known_validation[feats_chosen]
y_val = df_known_validation['Label']
# Initialize and train the SVM model
svm = SVC(kernel='poly', C=1, degree= 2, random_state=42)
svm.fit(X_train, y_train)
# Predict on the validation set
y_pred = svm.predict(X_val)
error = sum(y_pred!=y_val)/len(y_val)
```

להריץ כל פעם עם פיצ'ר אחר ולהוסיף פיצ'רים שמורידים את השגיאה

בחירת היפר-פרמטרים - Feature Selection

Parameter	Description	Applies to Kernel(s)
`C`	Regularization parameter (controls margin vs. misclassification trade-off)	All
`kernel`	Type of kernel function: `'linear'`, `'poly'`, `'rbf'`, `'sigmoid'`, `'precomputed'`	All
`degree`	Degree of the polynomial kernel function	`'poly'`

*רשימה חלקית עבור SVM

ניתן להשתמש למשל בGridSeachCV או
בRandomizedSearchCV או לכתוב קוד שבודק את זה ישירות

Cross Validation + Feature Selection

- ניתן להשתמש בקוד מהאינטרנט שעושה חלק מהדברים או הכל ביחד, ואולי ישתמש בשיטות אחרות לבחירת הפיצ'רים

- **חשוב מאוד שתבינו מה הקוד עושה ולמה!**

Final model

```
X_train = df_train[best_feats]
y_train = df_train['Label']
best_hps = {'kernel':'poly','degree':2, 'C':0.001}
final_svm = SVC(kernel=best_hps['kernel'], C=best_hps['C'], degree= best_hps['degree'], random_state=42)
final_svm.fit(X_train, y_train)
y_pred = final_svm.predict(df_test[best_feats])
df_test.loc[:,'predict_label'] = y_pred
df_submit = df_test[['case_id','predict_label']]
df_submit.head()
```

	case_id	predict_label
0	03b57fee-55c7-4873-b7ef-e29abd98863a	2.0
1	0425cb44-66af-4e25-afcb-cacccc2f1179	2.0
2	044579ef-16a6-4f00-b951-edc423d8a14f	2.0
3	05d4d9c7-fb6f-439a-bb97-f0e2737a7773	2.0
4	0858c8b7-e2eb-4461-b65e-9d476029ad8d	2.0

שימו לב כי אפשר היה לבחור לאמן כמה מודלים (קרוס ולידציה) ולבחור את האופציה שקיבלה הכי הרבה קולות מאותם מודלים