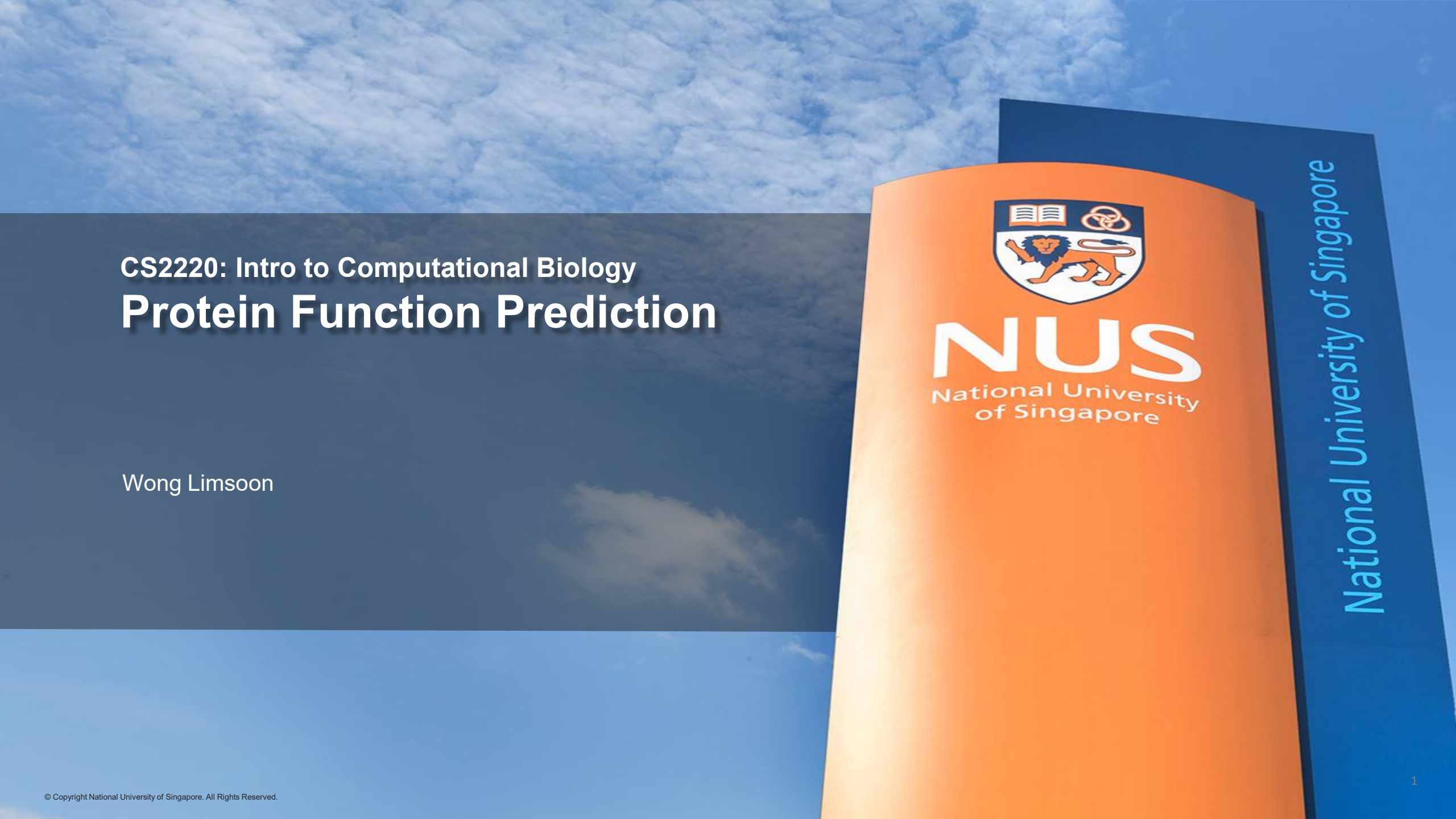


CS2220: Intro to Computational Biology

Protein Function Prediction

Wong Limsoon

Outline

Guilt by association of similarity of protein sequences

More thoughtful classifier evaluation

Guilt by association of similarity of dissimilarities

Guilt by association of similarity of interaction partners

Guilt by association of similarity of protein sequence

Protein function assignment

A protein is a large complex molecule made up of one or more chains of amino acids

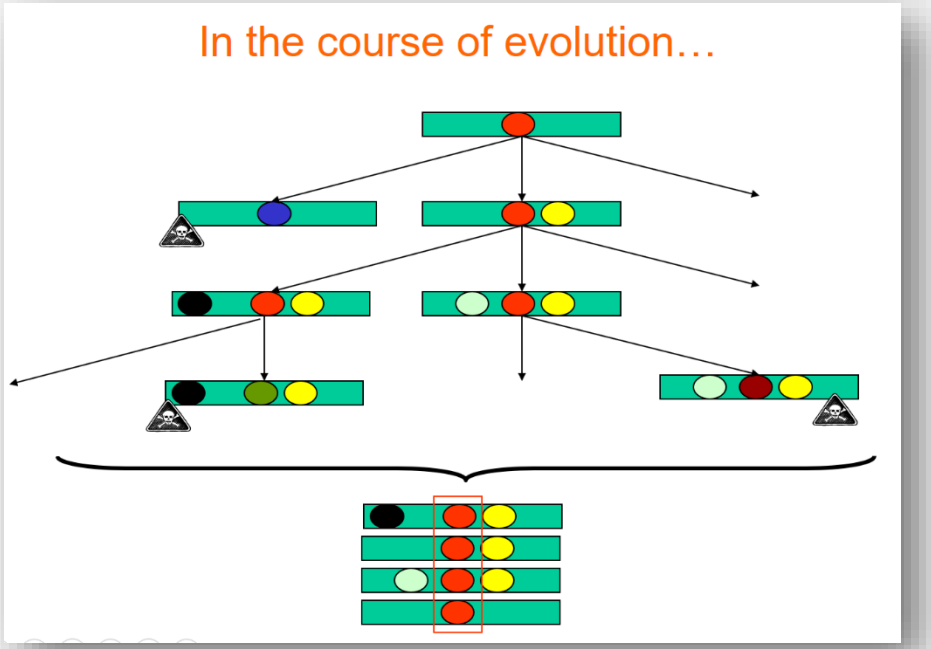
Usually, only the sequence of amino acid is known

```
SPSTNRKYPPLPVDKLEEEINRRMADDNKLFREEFNALPACPIQATCEAASKEENKEKNR
YVNILPYDHSRVHLTPVEGVPDSDYINASFINGYQEKNKFIAAQGPKEETVNDFWRMIWE
QNTATIVMVTNLKERKECKCAQYWPDQGCWTYGNVRVSVEDVTVLVDYTVRKFCIQQVGD
VTNRKPQRLITQFHFTSWPDFGVPFTPIGMLKFLKKVKACNPQYAGAIVVHCSAGVGRTG
TFVVIDAMLDMMHSERKVDVYGFVSRIRAQRCQMVQTDMQYVFIYQALLEHYLYGDTELE
VT
```

Proteins perform a wide variety of activities in the cell

How do we predict the function of a protein?

A standard postulate based on evolution



Two proteins (not) inheriting their function from a common ancestor (do not) have similar amino acid sequences

Guilt by association

Compare *T* with seqs of known function in a db

Poor Sequence Alignment

- Poor seq alignment shows few matched positions

⇒ The two proteins are not likely to be homologous

Alignment by FASTA of the sequences of amicyanin and domain 1 of ascorbate oxidase

```
                    60        70        80        90       100
Amicyanin           MPHNVHFVAGVLGEAALKGPMMKKEQAYSLPFTEAGTYDYHCTPHPFMRGKVVV
                                                :.:  . ::. ::
Ascorbate Oxidase ILQRGTPWADGTASISQCAINPGETFFYNFTVDNPGTFFYHGHLGMQRSAGLYG
                    70        80        90       100       110
```

No obvious match between
Amicyanin and Ascorbate Oxidase

Discard this function as a candidate

Good Sequence Alignment

- Good alignment usually has clusters of extensive matched positions

⇒ The two proteins are likely to be homologous

```
>gi|13476732|ref|NP_108301.1|  unknown protein [Mesorhizobium loti]
gi|14027493|dbj|BAB53762.1|  unknown protein [Mesorhizobium loti]
          Length = 105

 Score =  105 bits (262), Expect = 1e-22
 Identities = 61/106 (57%), Positives = 73/106 (68%), Gaps = 1/106 (0%)

Query: 1   MKPGRLASIALAIIFLPWAVPAHAATIEITMENLVISPTEVSAKVGDTIRWVNKDVFAHT 60
           MK G L  ++       MA PA AATIE+T++ LV SP  V AKVGDTI WVN DV AHT
Sbjct: 1   MKAGALIRLSVLAALALMAAPAAAATIEVTIDKLVFSPATVEAKVGDTIEWVNNDVVAHT 60
```

good match between
Amicyanin and unknown M. loti protein

Assign to *T* same function as homologs

Confirm with suitable wet experiments

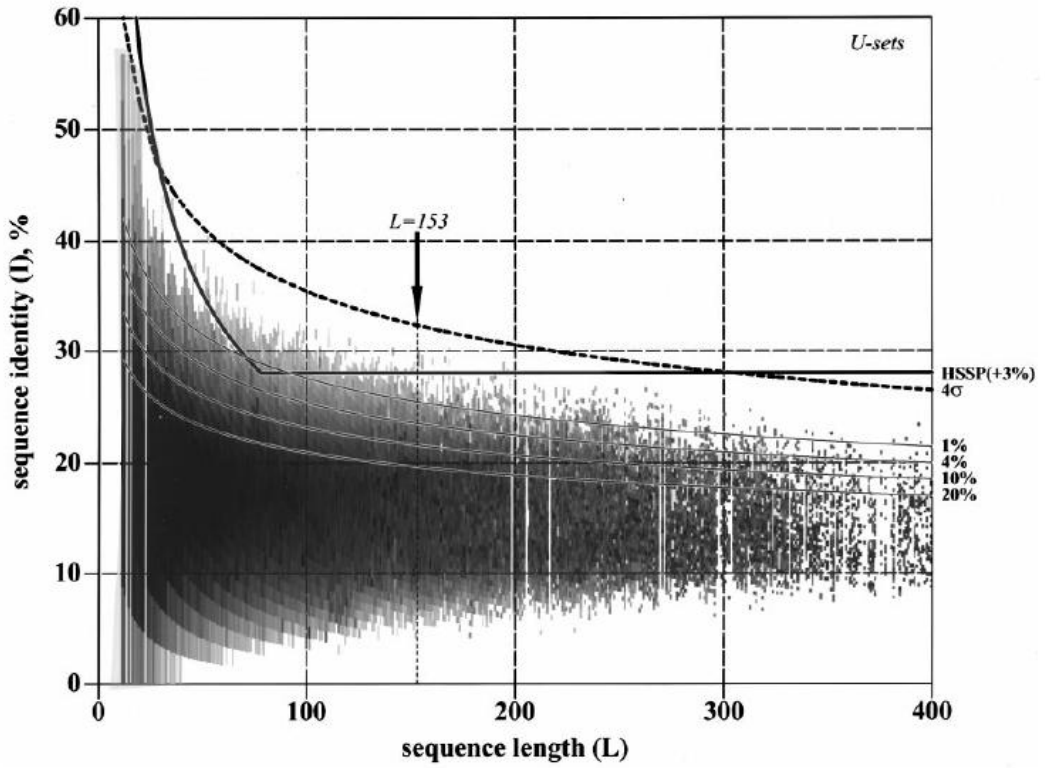
Limit of sequence similarity-based protein function assignment

So, need clever methods for the “twilight zone”

Similarity to ref proteins high ⇒ easy

Similarity is low (~30%) ⇒ error prone

Similarity is very low ⇒ hard



Abagyan RA, Batalov S. *J Mol Biol.,* 273(1):355-68, 1997

More thoughtful classifier evaluation

Many deep learning models to the rescue?

DeepFam (2018, CNN)

DeepGO (2018, CNN + DNN)

DeepPred (2019, hierarchical DNN)

DeepGoPlus (2019, CNN + DNN)

UDSMProt (2020, LSTM)

MultiPredGO (2020, multimodal DL)

TALE+ (2021, transformer)

DeepGraphGO (2021, CNN + DNN, multimodal)

DeepGoZero (2022, zero-shot learning), …

Exercise

DeepFam, deep learning for protein family prediction has reported outstanding performance

Really?

Discuss

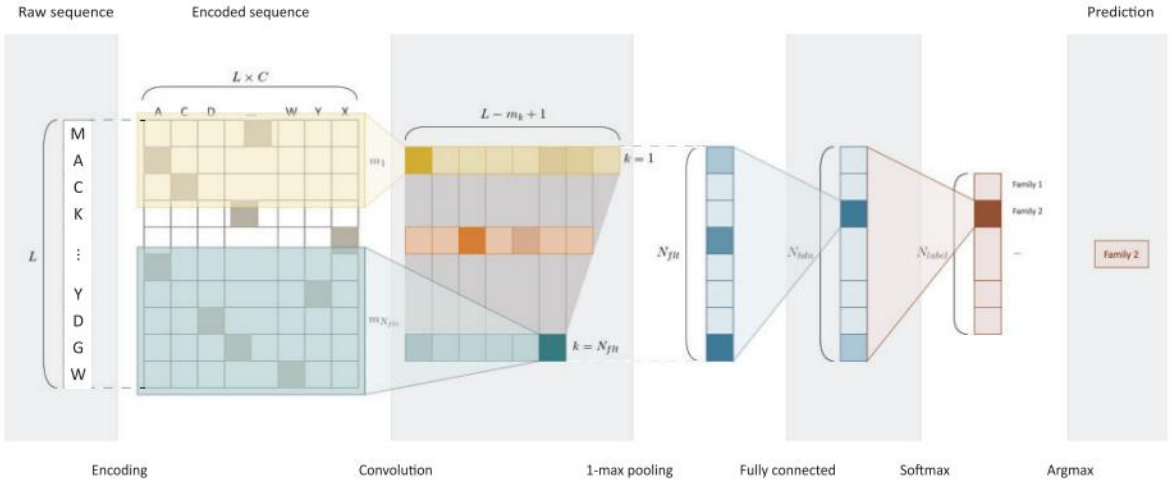


Fig. 1. The overview of DeepFam model. It is a feedforward convolution neural network whose last layer represents the probabilities of each family. convolution layer and 1-max pooling layer calculate a score (activation) of the existence of a conserved regions. The next layer is fully-connected neural network which can detect longer or complex sites. In order to infer the probability of each family, the last layer is designed as softmax layer (multinomial logistic regression), generally used for multi-class classification

Table 2. Prediction accuracy (%) comparison of COG dataset

Dataset	COG-500-1074	COG-250-1796	COG-100-2892
DeepFam	**95.40**	**94.08**	91.40
pHMM	91.75	91.78	**91.67**
3-mer LR	85.59	81.15	75.44
Protvec LR	47.34	41.76	37.05

Bold indicates the best performance for each dataset.

Seo et al., *Bioinformatics,* 34(13):i254-i262, 2018

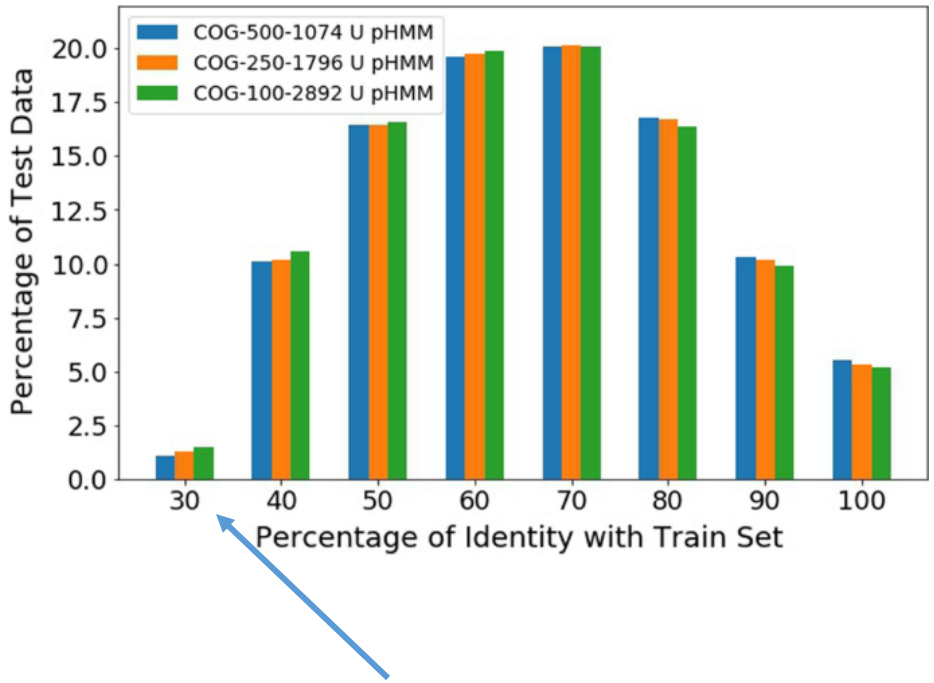
Exercise

In the context of protein function prediction

What should be considered easy test cases?

What should be considered hard test cases?

Exercise



Suppose DeepFam has poor performance on twilight zone proteins

There are very few twilight zone proteins in the test set & reference database

Therefore, we shouldn't worry too much about using DeepFam, right?

Exercise

DeepFam is a multi-class classifier, trained on n classes

Given a test instance from any of these n classes

DeepFam predicts the class to which it belongs

What kind of test instances would be considered a “surprise” to DeepFam?

How would DeepFam perform on them?

Do the test sets contain surprising questions?

How does DeepFam perform on these?

The test sets don’t have surprises (proteins from novel classes)

If you give DeepFam a protein from a novel class it was not trained on, it will always (wrongly) assign it to one of the classes it was trained on

There are thousands of function classes DeepFam cannot be trained on due to too few samples…

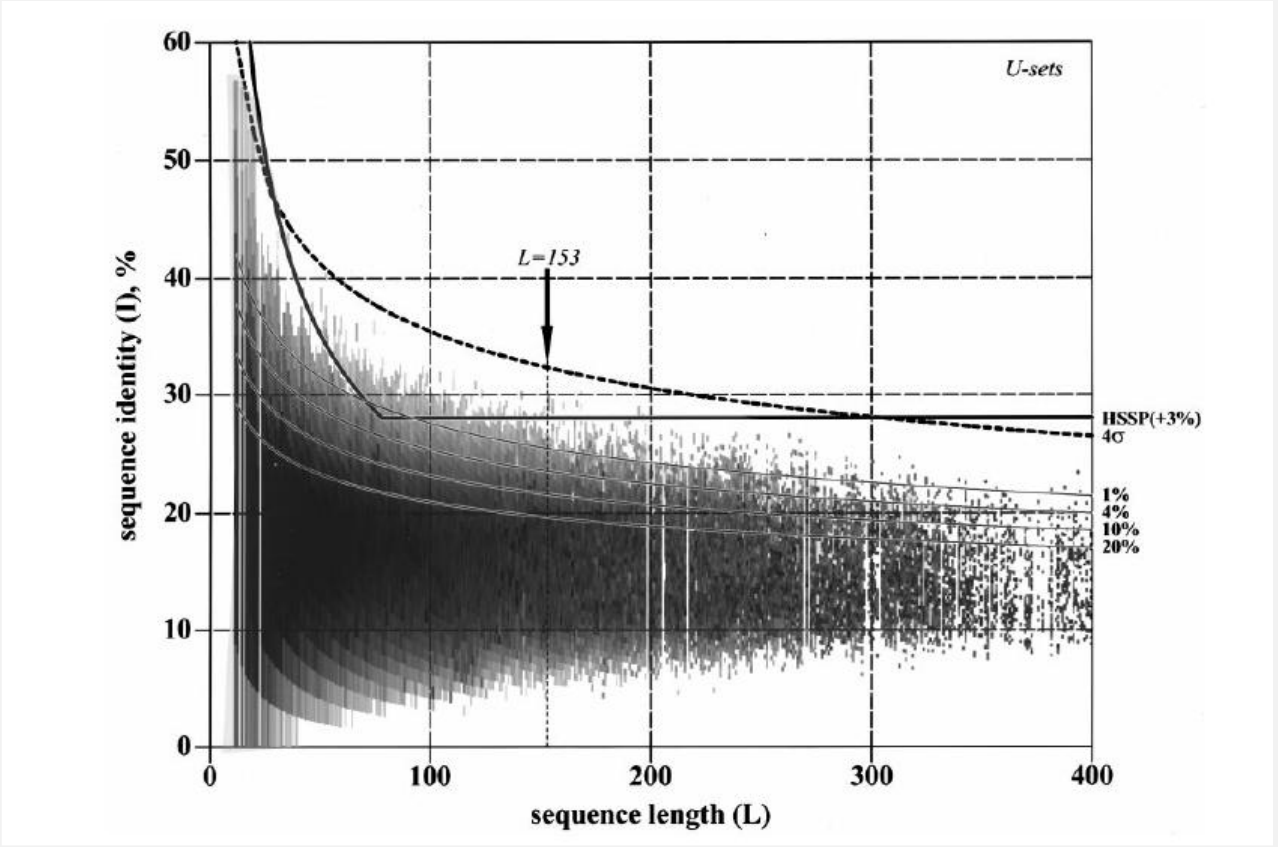
Exercise

Summarize what you have learned so far in this session

Guilt by association of similarity of dissimilarities

Inferring protein function from low-similarity reference proteins is harmful

Really?



Similarity to ref proteins high ⇒ easy

Similarity is low (~30%) ⇒ error prone

Similarity is very low ⇒ really hard

Abagyan RA, Batalov S. *J Mol Biol.,* 273(1):355-68, 1997

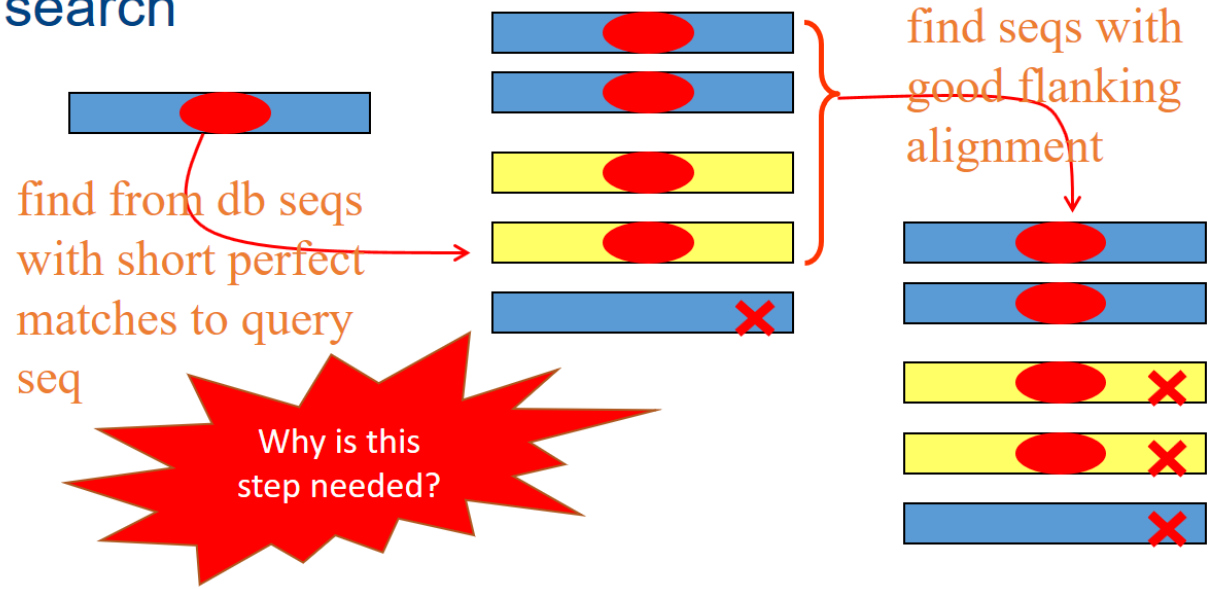
Many homology search tools are optimized to skip comparing & retrieving low-similarity proteins

A twilight-zone protein is low similarity ⇒ get nothing back!

BLAST: How it works

Altschul et al., *JMB*, 215:403-410, 1990

BLAST is one of the most popular tool for doing fast "guilt-by-association" sequence homology search



Similarities of dissimilarities

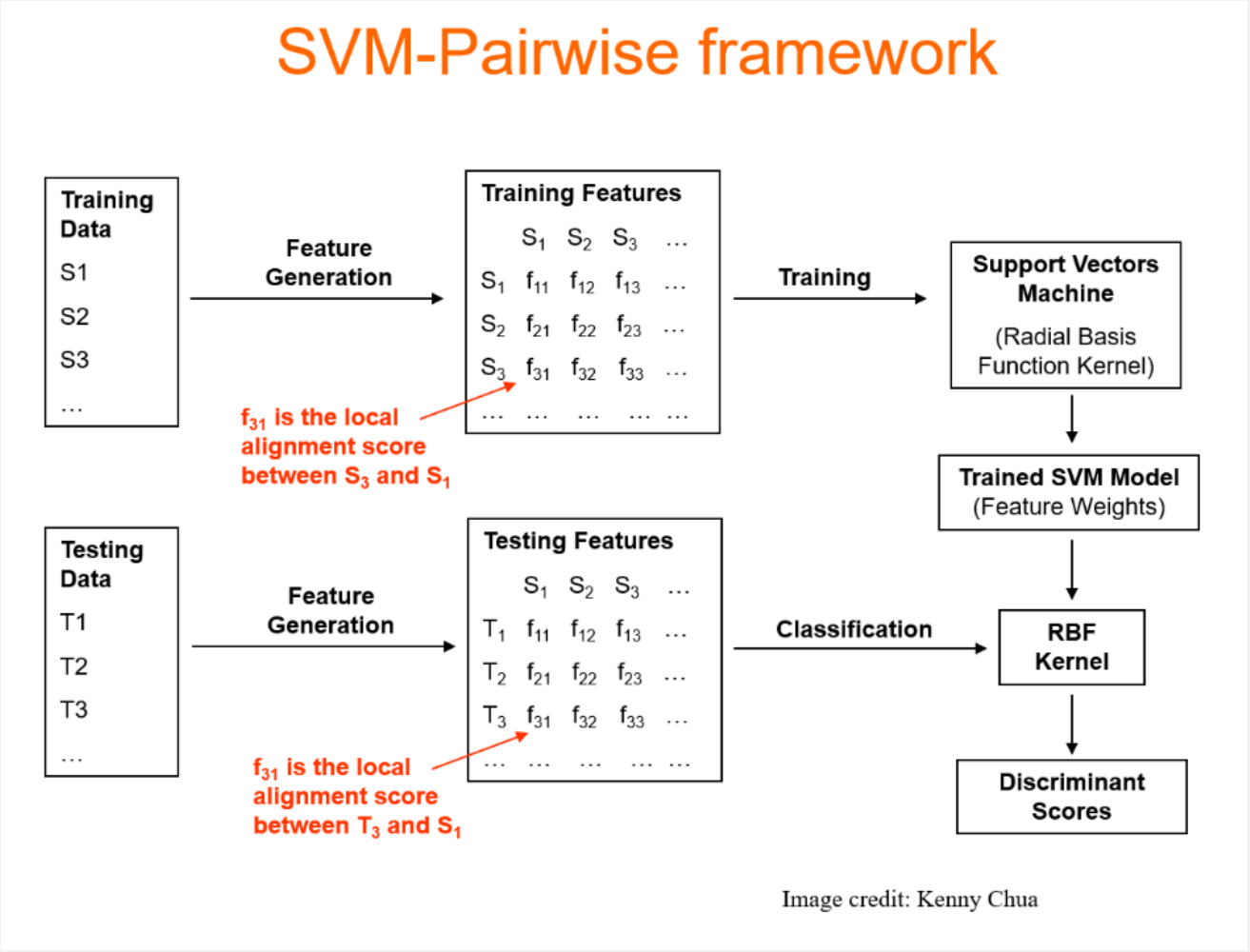
The difference between any apple to orange / banana / mango / etc. are mostly same as the difference between any other apple with that orange / banana / mango / etc.

The difference between a mysterious fruit X to orange / banana / mango / etc. are mostly same as the difference between an apple to orange / banana / mango / etc.

$\Rightarrow$The fruit X is likely an apple

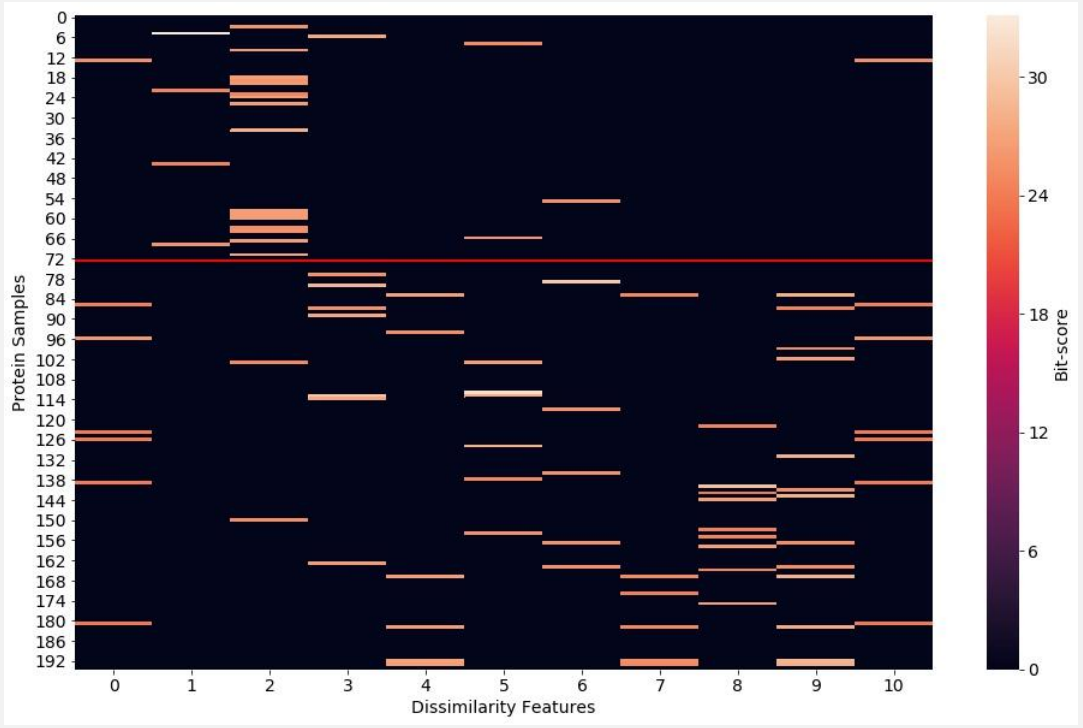
EnsembleFam uses low- / dis-similarity information discarded by other methods!

Inspired by SVM-pairwise



Neamul Kabir & Wong, *BMC Bioinformatics,* 23:90, 2022

HeatMap of dissimilarity features



Note: Bitscore of proteins in the same family is typically ~200

There is consistency in the way two proteins of the same family differ from the other families

Exercise

How are “surprises” (i.e., proteins not in any of the training classes) handled?

Can EnsembleFam / SVM-Pairwise type of approach avoid the problem that DeepFam has wrt these proteins?

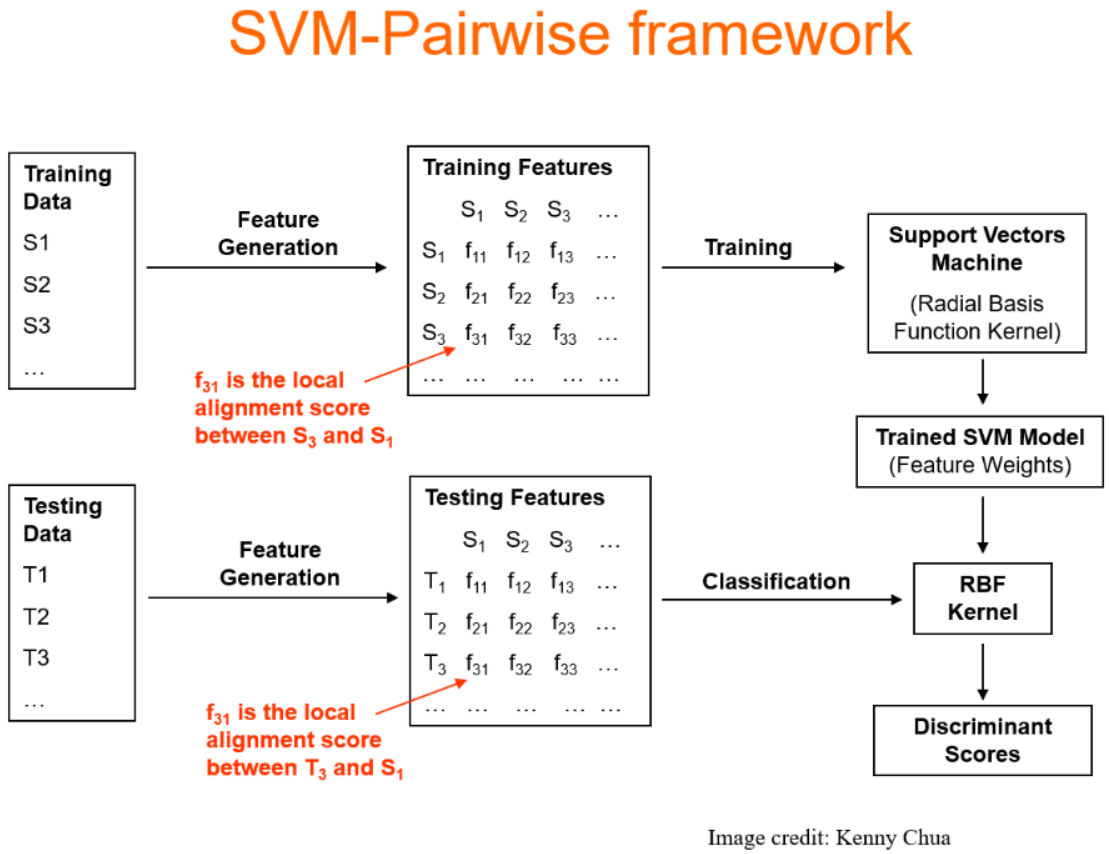


Image credit: Kenny Chua

Exercise

If there are 1,000,000 sequences in the training set, there would be 1,000,000 features

Curse of dimensionality: Huge model

Inefficient: Take lots of time to generate the feature vector of a test sequence

Discuss how this can be solved

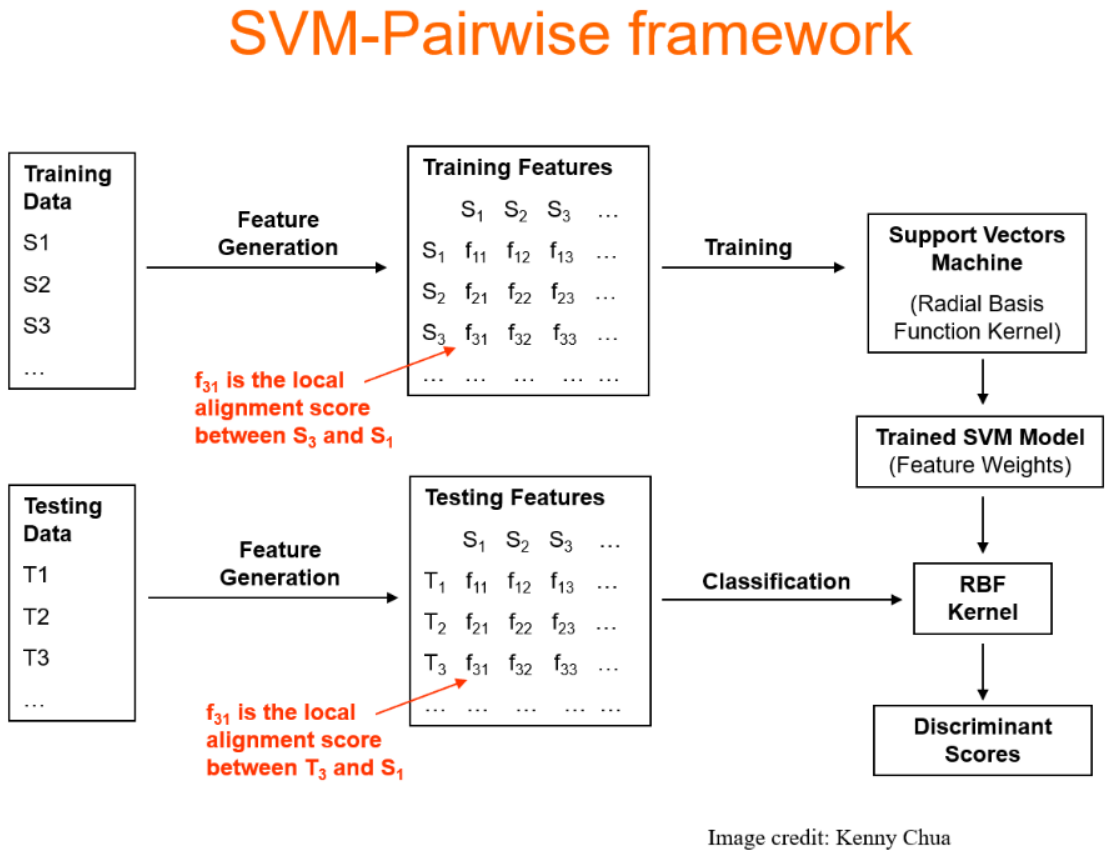
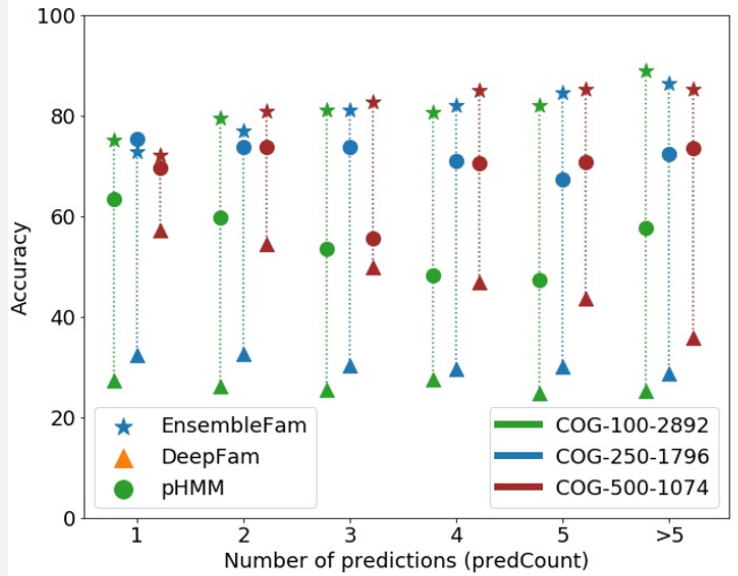


Image credit: Kenny Chua

EnsembleFam performance on the whole COG test set

Dataset	Method	predCount = 1	predCount = 2	predCount = 3	predCount = 4	predCount = 5	predCount > 5
COG-500-1074	EnsembleFam	**98.14**	**98.59**	**98.73**	**98.87**	**98.87**	**99.13**
	pHMM	96.46	97.22	97.06	96.63	95.74	95.44
	DeepFam	84.88	82.49	80.86	79.46	78.18	77.25
COG-250-1796	EnsembleFam	**97.74**	**98.41**	**98.54**	**98.81**	**98.80**	**99.17**
	pHMM	96.44	97.08	96.89	96.13	94.88	95.04
	DeepFam	72.29	71.60	71.22	71.28	71.20	70.48
COG-100-2892	EnsembleFam	**98.01**	**98.44**	**98.71**	**98.82**	**98.96**	**99.37**
	pHMM	96.59	96.75	95.83	94.36	95.70	95.84
	DeepFam	61.51	62.59	64.87	67.41	68.12	67.89

EnsembleFam performance in the twilight zone



0 < identity <=30

0 < identity <= 30							
Dataset	**Method**	**Pred Count=1**	**Pred Count=2**	**Pred Count=3**	**Pred Count=4**	**Pred Count=5**	**Pred Count > 5**
COG-500-1074	Ensemble Fam	**72.07**	**81.00**	**82.82**	**84.96**	**85.33**	**85.27**
	pHMM	69.54	73.75	55.51	70.62	70.85	73.55
	DeepFam	57.14	54.52	49.90	46.92	43.64	35.94
COG-250-1796	Ensemble Fam	72.84	**77.07**	**81.02**	**82.14**	**84.66**	**86.45**
	pHMM	**75.39**	73.82	73.84	71.02	67.44	72.43
	DeepFam	32.44	32.54	30.24	29.53	30.02	28.68
COG-100-2892	Ensemble Fam	**75.24**	**79.55**	**81.21**	**80.63**	**82.05**	**88.95**
	pHMM	63.44	59.69	53.45	48.16	47.42	57.57
	DeepFam	27.30	26.13	25.54	27.62	24.83	25.36

Contribution of dissimilarities

Method	predCount = 1	predCount = 2	predCount = 3	predCount = 4	predCount = 5	predCount > 5
Identity: $0 \leq x \leq 30$						
SVM Model 1	59.82	66.96	67.60	67.96	67.32	74.67
SVM Model 2	57.11	65.09	65.01	65.86	64.55	71.80
SVM Model 3	57.34	65.34	64.29	65.02	63.36	70.13
EnsembleFam	**72.07**	**81.00**	**82.82**	**84.96**	**85.33**	**85.27**

Base SVMs have similar performance, Not much better than e.g. DeepFam

Where does performance increment of the ensemble come from?

Dissimilarity features contribute half of the predictions by EnsembleFam

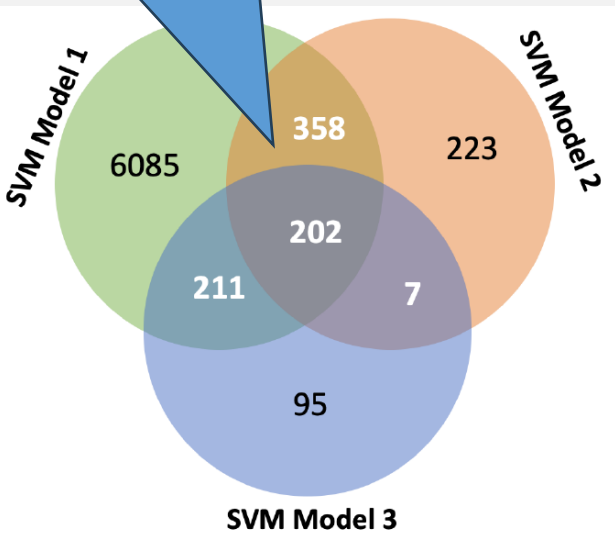


Figure 3.7: Prediction overlap of the three base classifier on the twilight zone proteins in $0 \leq$ identity ≤ 30 region. The Venn diagram is drawn based on the prediction made on twilight zone proteins of the testset of COG-500-1074 dataset. The number of predictions made by each base classifier is indicated in the figure. Numbers highlighted in white indicate overlap between at least two methods, hence predicted by EnsembleFam.

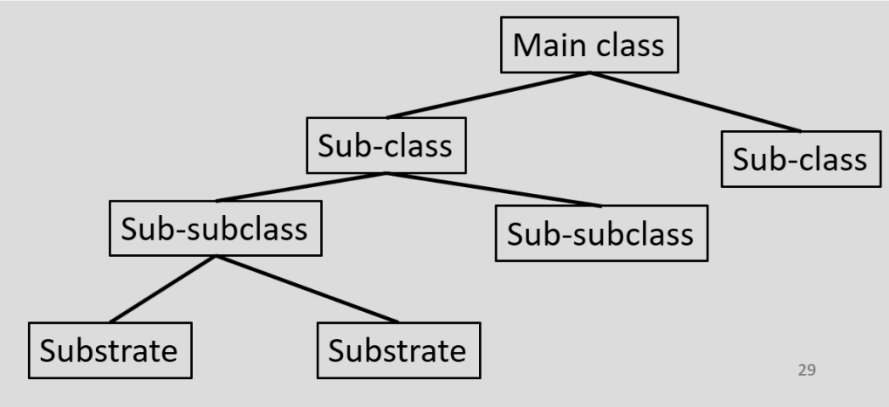
Enzyme Commission (EC) # prediction

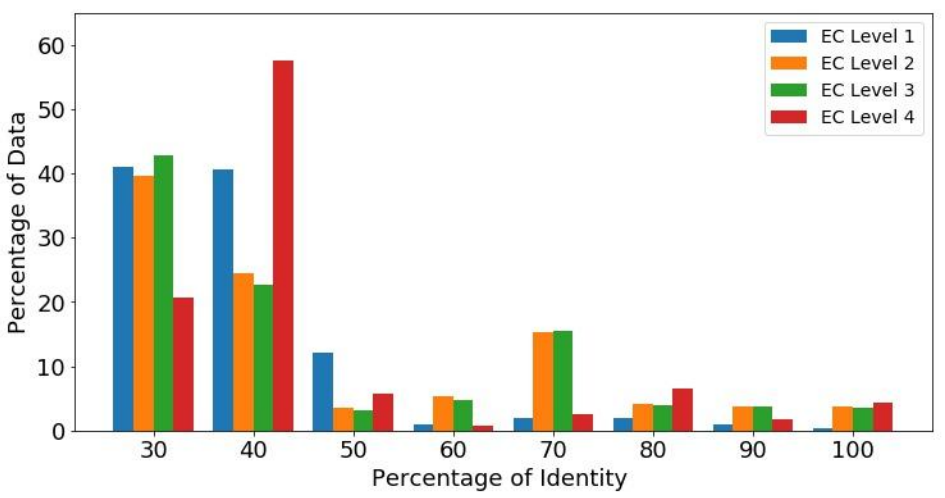
Enzymes are more heterogeneous due to hierarchical nature

This heterogeneity adds more difficulty in annotation

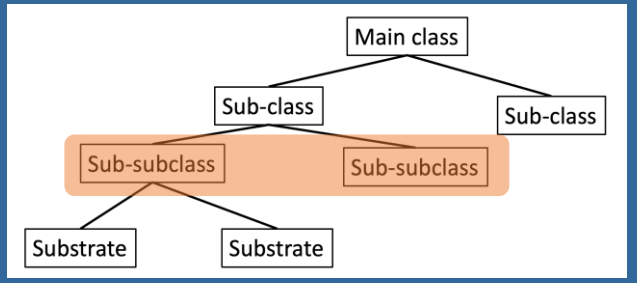
There are also twilight zone enzymes

Current methods do not provide good support for twilight zone





EC # - level 3



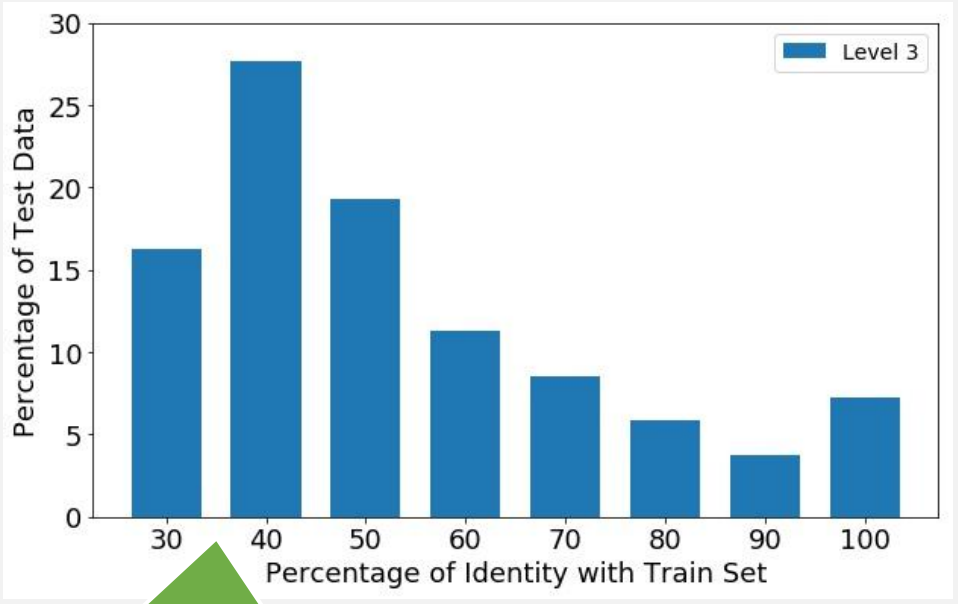
Dataset used for expt: Swiss-Prot

Total # of families: 185

Min # of annotated member: 30

Total	Training Data	Blind Test Set*
SWISS-PROT	Annotated before 2019	Annotated in 2019 or later
254,176 seq	251,817 seq	2,359 seq

*As all methods, compared in our experiment, are trained using Swiss-Prot annotations of 2018 or earlier, we used the rest, annotated in 2019 or later, as our blind test set.



45% test data lies in or near twilight zone

EnsembleFam's performance on EC-level 3 predictions in and near the twilight zone in Swiss-Prot

0 <= identity <= 30	TP	FP	Specificity	Sensitivity	Precision	F1-score	Geo Mean
e-EnsembleFam	**109**	290	98.54	**17.69**	27.32	**21.48**	**41.75**
DEEPre	82	231	98.71	13.31	26.20	17.65	36.25
DeepEC	46	29	**99.98**	7.47	**61.33**	13.31	27.33
EFICAz$^{2.5}$	81	75	99.92	13.15	51.92	20.98	36.25
CatFam	33	32	99.96	5.36	50.77	9.69	23.14
ECPred	17	**25**	**99.98**	2.76	40.48	5.17	16.61

30 < identity <= 40	TP	FP	Specificity	Sensitivity	Precision	F1-score	Geo Mean
e-EnsembleFam	**398**	121	98.89	**57.85**	76.69	**65.95**	**75.64**
DEEPre	211	196	96.64	30.67	51.84	38.54	54.44
DeepEC	77	47	99.96	11.20	62.10	18.97	33.46
EFICAz$^{2.5}$	357	78	99.92	51.89	92.07	63.58	72.01
CatFam	109	**14**	**99.99**	15.84	**88.62**	26.88	39.80
ECPred	34	27	99.98	4.94	55.73	9.08	22.22

Identifying enzymes in new genomes

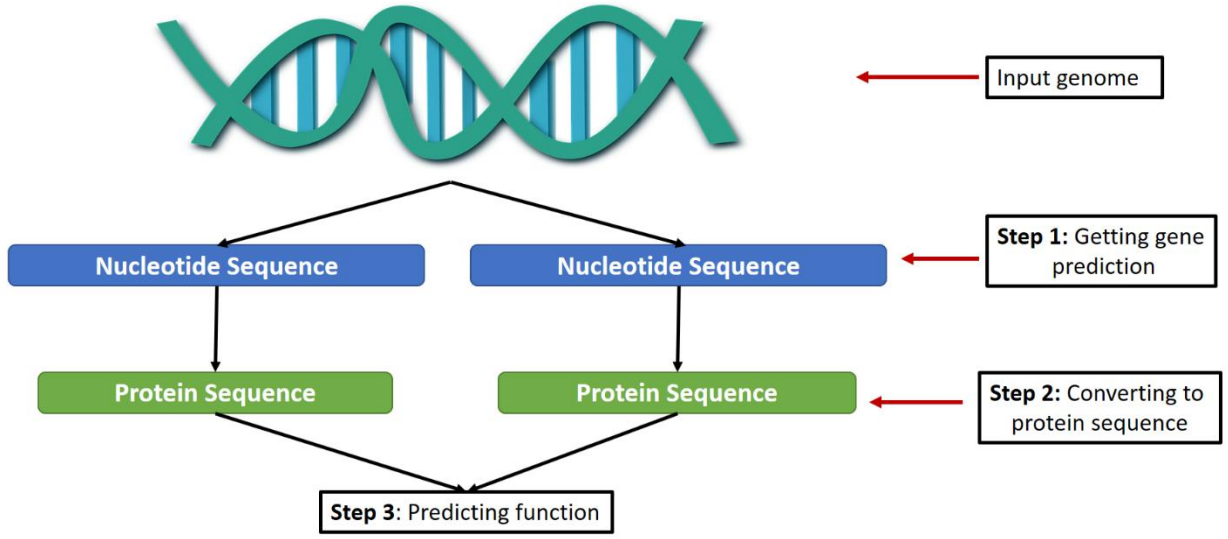
- **Input**

Genome

Enzyme Name	EC Number
PETase (PET hydrolase)	3.1.1.101
MHETase (MHET hydrolase)	3.1.1.102
Terpene (MLH)	3.1.1.83

Enzymes of interest

- **Output**
  - Enzymes of interest exist in genome or not



Enzyme hunting in new fungal genomes

EC level 3 prediction of different methods. EnsembleFam provides more predictions than competing methods

Genome 1: # predicted genes = 3302		
	Enzyme	Non-enzyme
EnsembleFam	635 (19%)	2667 (81%)
DeepEC	56 (2%)	3246 (98%)

Genome 2: # predicted genes = 3864		
	Enzyme	Non-enzyme
EnsembleFam	725 (19%)	3139 (81%)
DeepEC	54 (1%)	3810 (99%)

Genome 3: # predicted genes = 3599		
	Enzyme	Non-enzyme
EnsembleFam	684 (19%)	2915 (81%)
DeepEC	68 (2%)	3531 (98%)

About the inventor: Neamul Kabir

Neamul Kabir

PhD, NUS, 2023

Lecturer

NUS SOC DISA

Exercise

Summarize what you have learned so far in this session

Guilt by association of similarity of interaction partners

Protein-protein interaction network

Protein-protein interaction (PPI) is an interaction between two proteins that result in a biological function

PPI network (PPIN) is a graphical representation where nodes are proteins and edges are interactions

PPI datasets are produced from “two-hybrid” experiments, co-immunoprecipitation, affinity purification-mass spectrometry, etc.

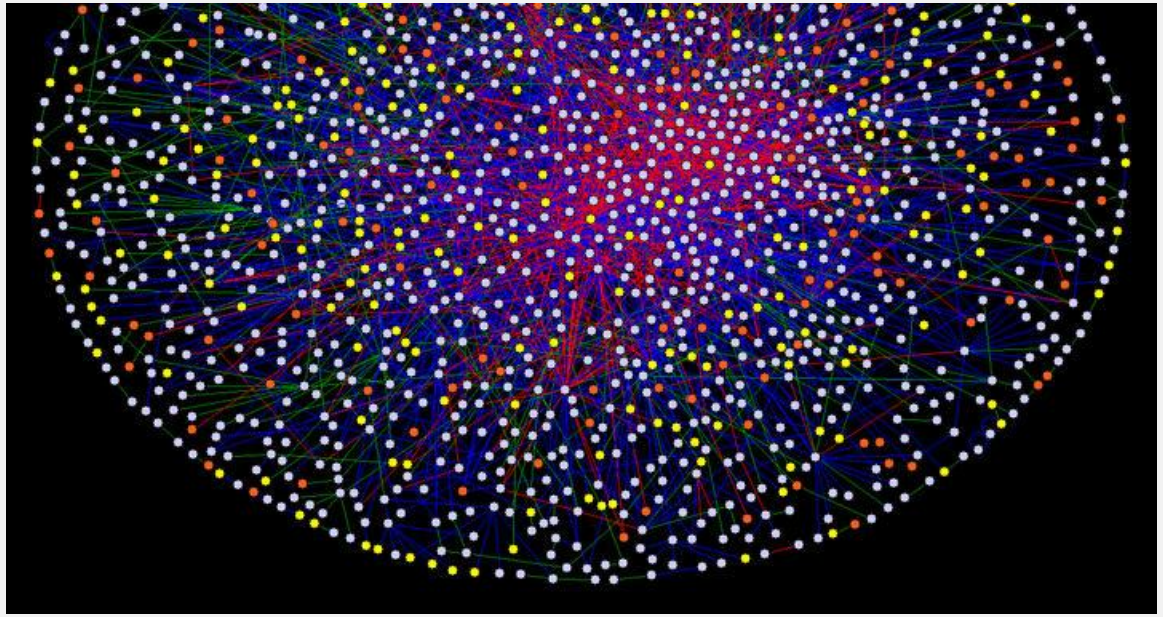


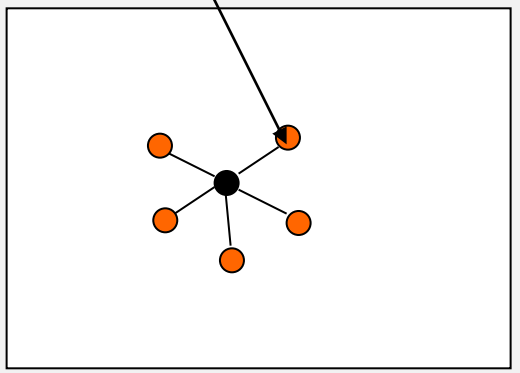
Image credit: Max Delbrueck Center

Functional association through interactions

Direct functional association

Proteins from the same pathways are likely to interact

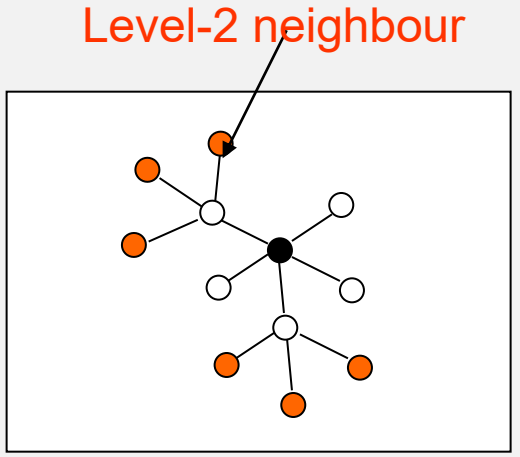
⇒ Interaction partners of a protein share functions w/ it



Indirect functional association

Proteins that have common biochemical, physical properties and/or subcellular localization are likely to bind to the same proteins

⇒ Proteins that share interaction partners with a protein share functions w/ it

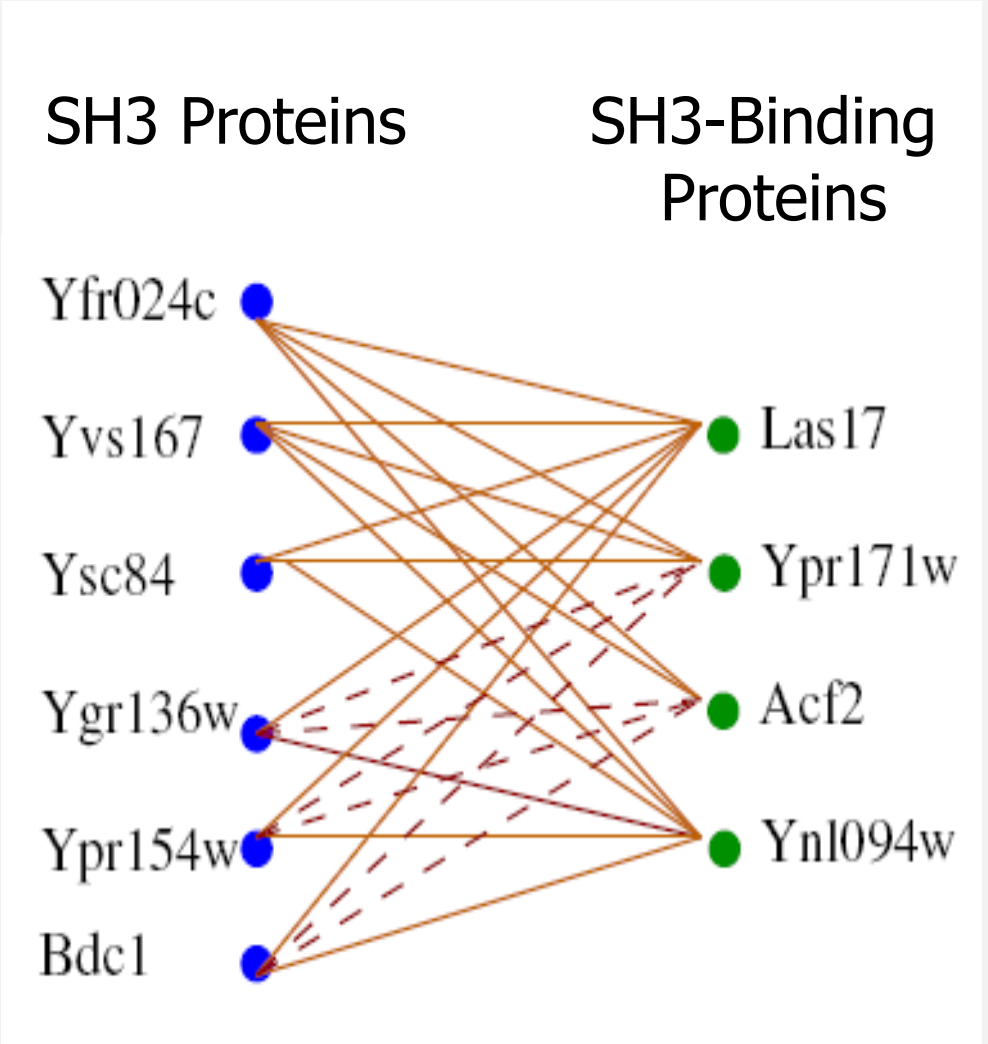


An illustrative case of indirect functional association?

Is indirect functional association plausible?

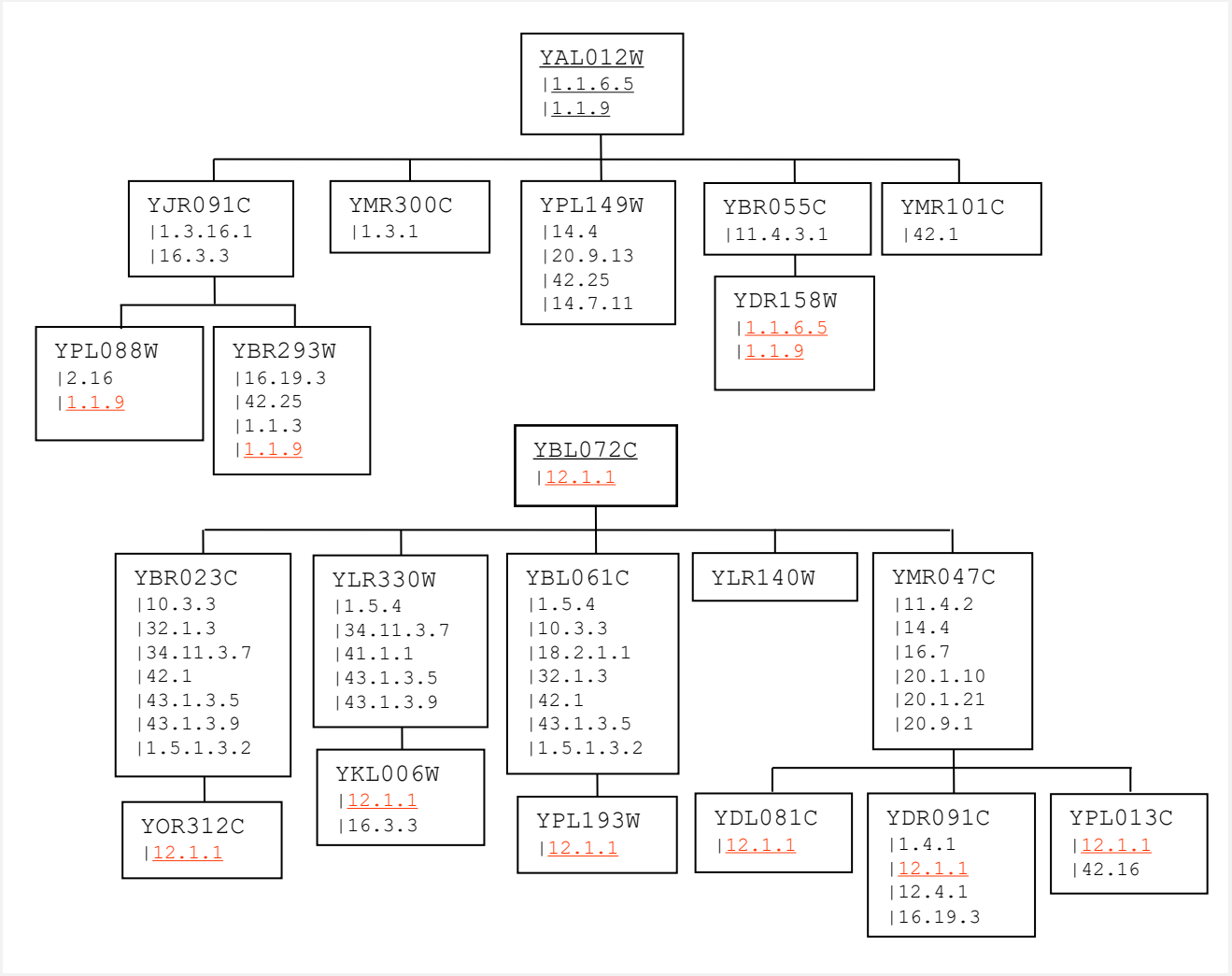
Is it found often in real interaction data?

Can it be used to improve protein function prediction from protein interaction data?



Frequency of indirect functional association

Shared Functions with	Fraction
Level-1 neighbours exclusively	0.016338
Level-2 neighbours exclusively	0.226574
Level-1 and Level-2 neighbours	0.463960
Level-1 or Level-2 neighbours	0.706872



Source: Kenny Chua

Prediction by simple neighbour counting

For a given set of “neighbours” of a protein

The “neighbour counting” method predicts the functions of each protein by counting the frequency with which its neighbours have each function

The function that is the *n*-th most frequent in a protein's neighbours is predicted as the *n*-th most probable function of the protein

The rank of each predicted function is taken as its score

Prediction power by neighbour counting

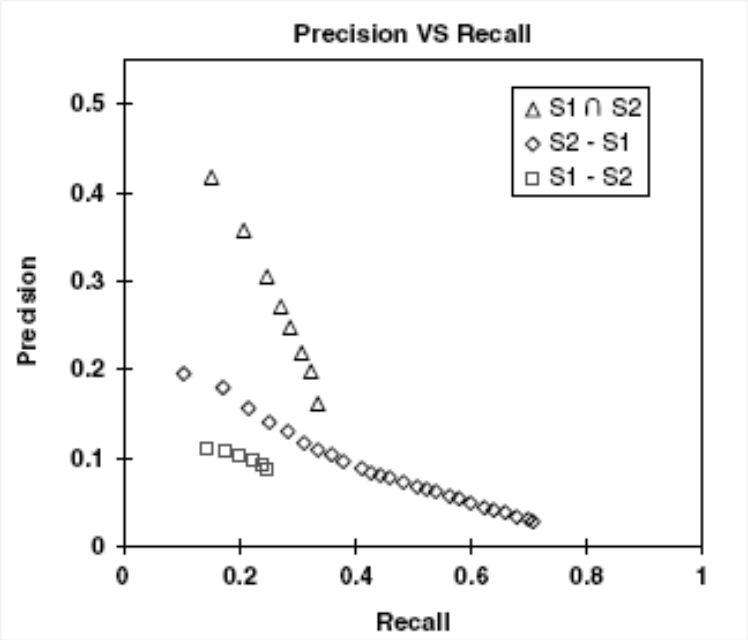
Sensitivity vs Precision analysis

$$PR = \frac{\sum_i^K k_i}{\sum_i^K m_i} \qquad SN = \frac{\sum_i^K k_i}{\sum_i^K n_i}$$

n_i is # functions of protein i

m_i is # functions predicted for protein i

k_i is # functions predicted correctly for protein i



level-2 only neighbours is good

level-1 ∩ level-2 neighbours is best

Exercise

PPIN is usually noisy

Lots of edges are artifacts

Make simple neighbour counting unreliable

How to avoid counting "noise" neighbours?

Experimental method category[a]	Number of interacting pairs	Co-localization[b] (%)	Co-cellular-role[b] (%)
All: All methods	9347	64	49
A: Small scale Y2H	1861	73	62
A0: GY2H Uetz *et al.* (published results)	956	66	45
A1: GY2H Uetz *et al.* (unpublished results)	516	53	33
A2: GY2H Ito *et al.* (core)	798	64	40
A3: GY2H Ito *et al.* (all)	3655	41	15
B: Physical methods	71	98	95
C: Genetic methods	1052	77	75
D1: Biochemical, *in vitro*	614	87	79
D2: Biochemical, chromatography	648	93	88
E1: Immunological, direct	1025	90	90
E2: Immunological, indirect	34	100	93
2M: Two different methods	2360	87	85
3M: Three different methods	1212	92	94
4M: Four different methods	570	95	93

Sprinzak et al., *JMB*, 327:919-923, 2003

Large disagreement betw methods

- **High level of noise**

⇒ **Need to clean up before making inference on PPI networks**

Exercise

To know whether two proteins are localized to the same cellular compartment, do you really need to know where two proteins are?

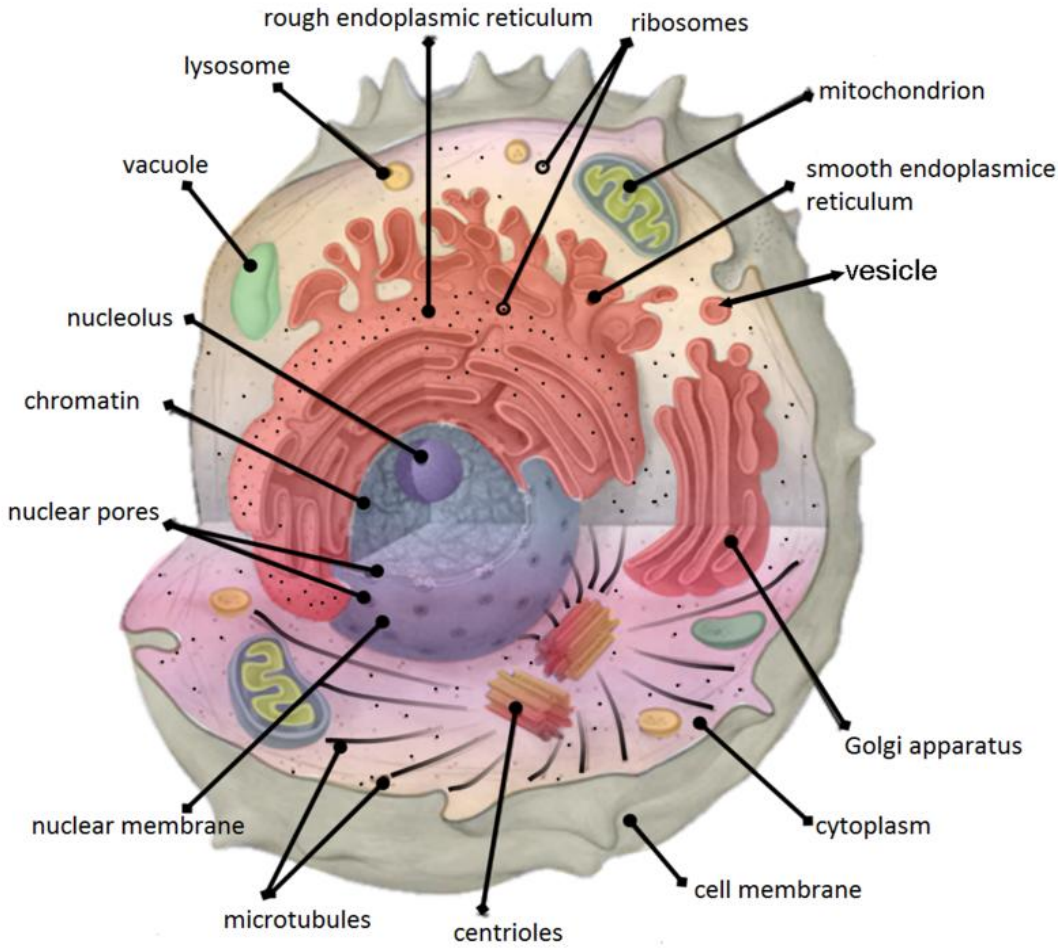


Image credit: Wikipedia

Functional similarity estimate: Czekanowski-Dice distance

Functional distance between two proteins

$$D(u,v) = \frac{|N_u \Delta N_v|}{|N_u \cup N_v| + |N_u \cap N_v|}$$

N_k is set of interacting partners of k

Δ is symmetric diff

Brun et al., *Genome Biol.*, 5:R6, 2003

Similarity can be defined as

$$S(u,v) = 1 - D(u,v) = \frac{2\,|N_u \cap N_v|}{|N_u| + |N_v|}$$

Issue:

Consider relative intersection size of the two neighbor sets, not absolute intersection size

- Case 1: $|N_u| = 1$, $|N_v| = 1$, $|N_u \cap N_v| = 1$, CD(u,v)=1
- Case 2: $|N_u| = 10$, $|N_v| = 10$, $|N_u \cap N_v| = 10$, CD(u,v)=1

Functional similarity estimate: FS-weighted measure

FS-weighted measure

$$S(u,v) = \frac{2|N_u \cap N_v|}{|N_u - N_v| + 2|N_u \cap N_v|} \times \frac{2|N_u \cap N_v|}{|N_v - N_u| + 2|N_u \cap N_v|}$$

N_k is the set of interacting partners of k

Correlation with functional similarity

Neighbours	CD-distance	FS-Weight
S_1	0.471810	0.498745
S_2	0.224705	0.298843
$S_1 \cup S_2$	0.224581	0.29629

Reliability of expt sources

Assign reliability to an interaction based on its expt sources

Reliability between u and v computed by:

$$r_{u,v} = 1 - \prod_{i \in E_{u,v}} (1 - r_i)$$

r_i is reliability of expt source i,

$E_{u,v}$ is the set of expt sources in which interaction betw u and v is observed

Source	Reliability
Affinity Chromatography	0.823077
Affinity Precipitation	0.455904
Biochemical Assay	0.666667
Dosage Lethality	0.5
Purified Complex	0.891473
Reconstituted Complex	0.5
Synthetic Lethality	0.37386
Synthetic Rescue	1
Two Hybrid	0.265407

Nabieva et al, *Bioinformatics*, 19(S1):i302-i310, 2005

Exercise

Can you think of things a biologist can do to assess the overall reliability of a PPI screening assay / source?

Source	Reliability
Affinity Chromatography	0.823077
Affinity Precipitation	0.455904
Biochemical Assay	0.666667
Dosage Lethality	0.5
Purified Complex	0.891473
Reconstituted Complex	0.5
Synthetic Lethality	0.37386
Synthetic Rescue	1
Two Hybrid	0.265407

Nabieva et al, *Bioinformatics*, 19(S1):i302-i310, 2005

Functional similarity estimate: FS-weighted measure with reliability

Take reliability into consideration when computing FS-weighted measure:

$$S_R(u,v) = \frac{2\sum_{w\in(N_u\cap N_v)} r_{u,w}r_{v,w}}{\left(\sum_{w\in N_u-N_v} r_{u,w} + \sum_{w\in(N_u\cap N_v)} r_{u,w}(1-r_{v,w})\right) + 2\sum_{w\in(N_u\cap N_v)} r_{u,w}r_{v,w}} \times \frac{2\sum_{w\in(N_u\cap N_v)} r_{u,w}r_{v,w}}{\left(\sum_{w\in N_v-N_u} r_{v,w} + \sum_{w\in(N_u\cap N_v)} r_{v,w}(1-r_{u,w})\right) + 2\sum_{w\in(N_u\cap N_v)} r_{u,w}r_{v,w}}$$

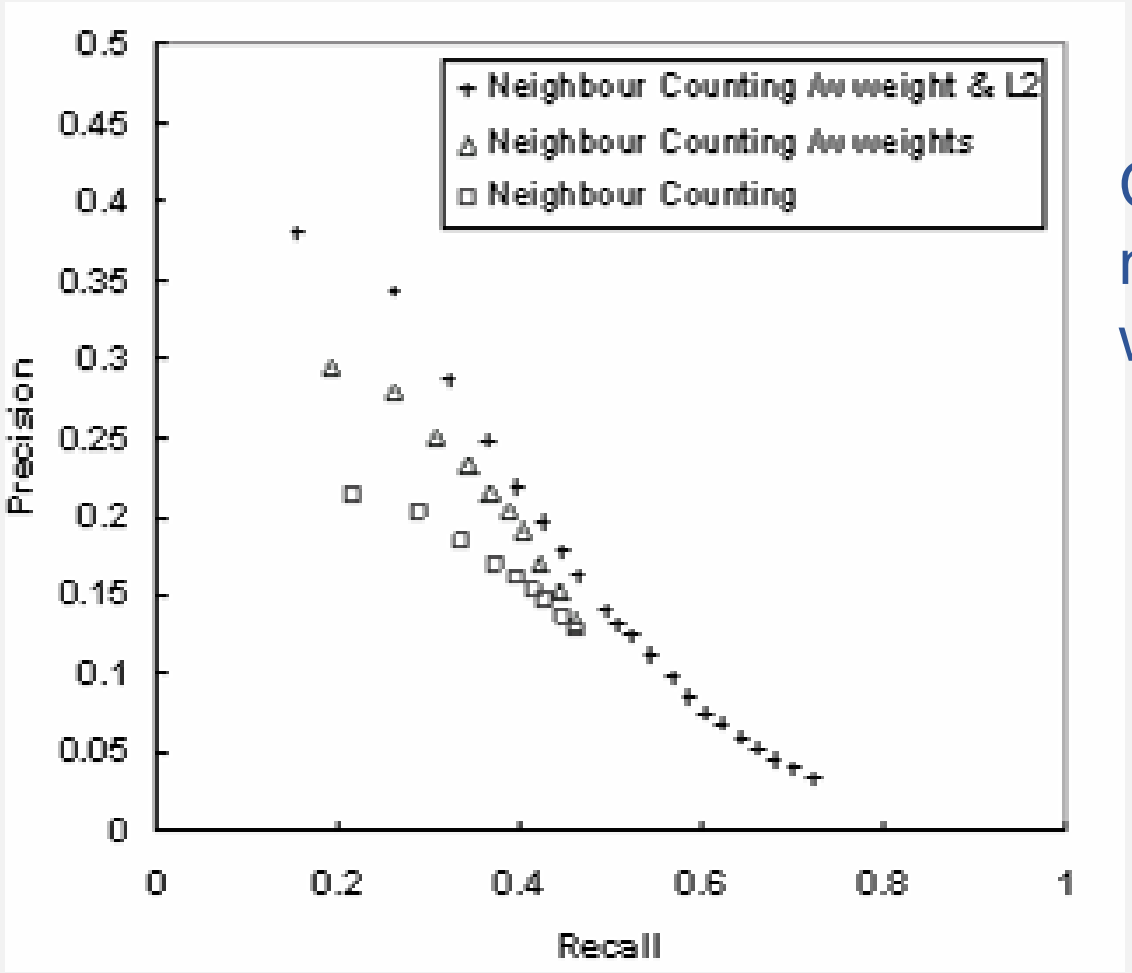
N_k is the set of interacting partners of k

$r_{u,w}$ is reliability weight of interaction betw u and v

Integrating reliability

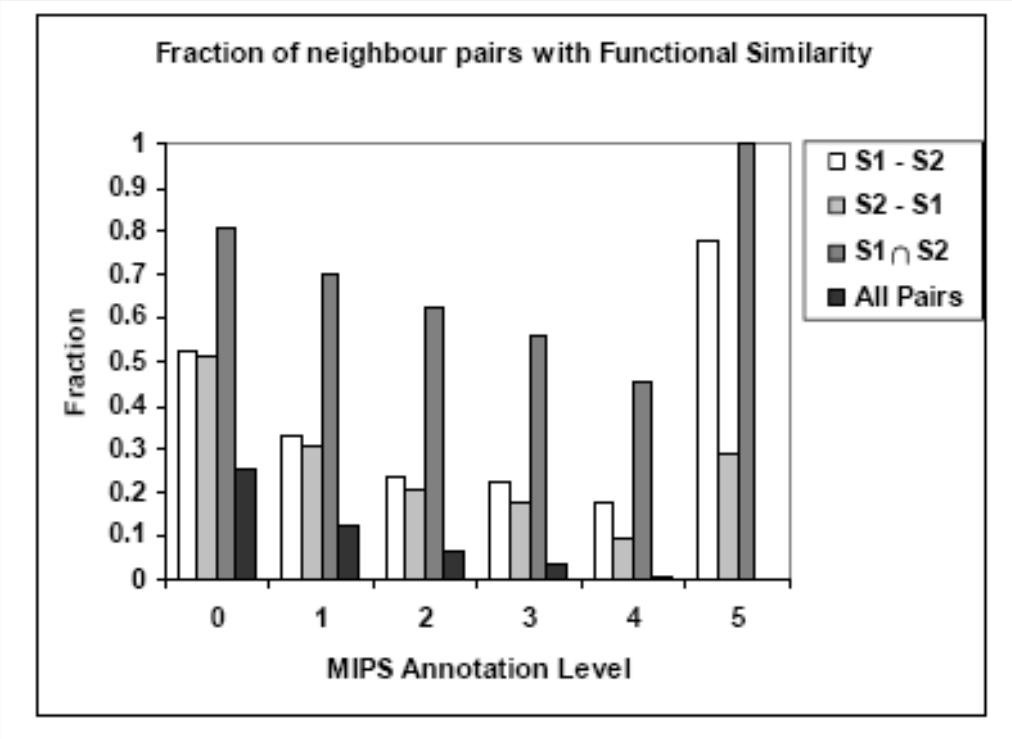
Neighbours	CD-distance	FS-Weight	FS-Weight R
S_1	0.471810	0.498745	0.532596
S_2	0.224705	0.298843	0.375317
$S_1 \cup S_2$	0.224581	0.29629	0.363025

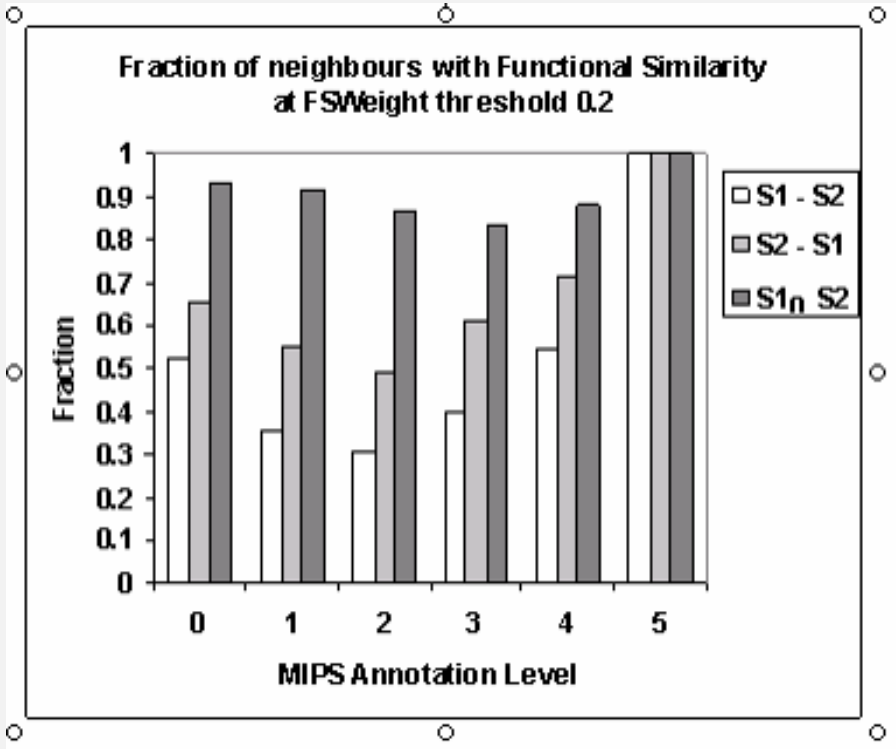
Improvement to prediction power by neighbor counting



Considering only neighbours w/ FS weight > 0.2

Improvement to over-rep of functions in neighbours





Use level-1 & level-2 neighbours for prediction

FS-weighted Average

$$f_x(u) = \frac{1}{Z}\left[\lambda r_{\text{int}} \pi_x + \sum_{v \in N_u}\left(S_{TR}(u,v)\delta(v,x) + \sum_{w \in N_v} S_{TR}(u,w)\delta(w,x)\right)\right]$$

r_{int} is fraction of all interaction pairs sharing function

λ is weight of contribution of background freq

$\delta(k, x) = 1$ if k has function x, 0 otherwise

N_k is the set of interacting partners of k

π_x is freq of function x in the dataset

Z is sum of all weights

$$Z = 1 + \sum_{v \in N_u}\left(S_{TR}(u,v) + \sum_{w \in N_v} S_{TR}(u,w)\right)$$

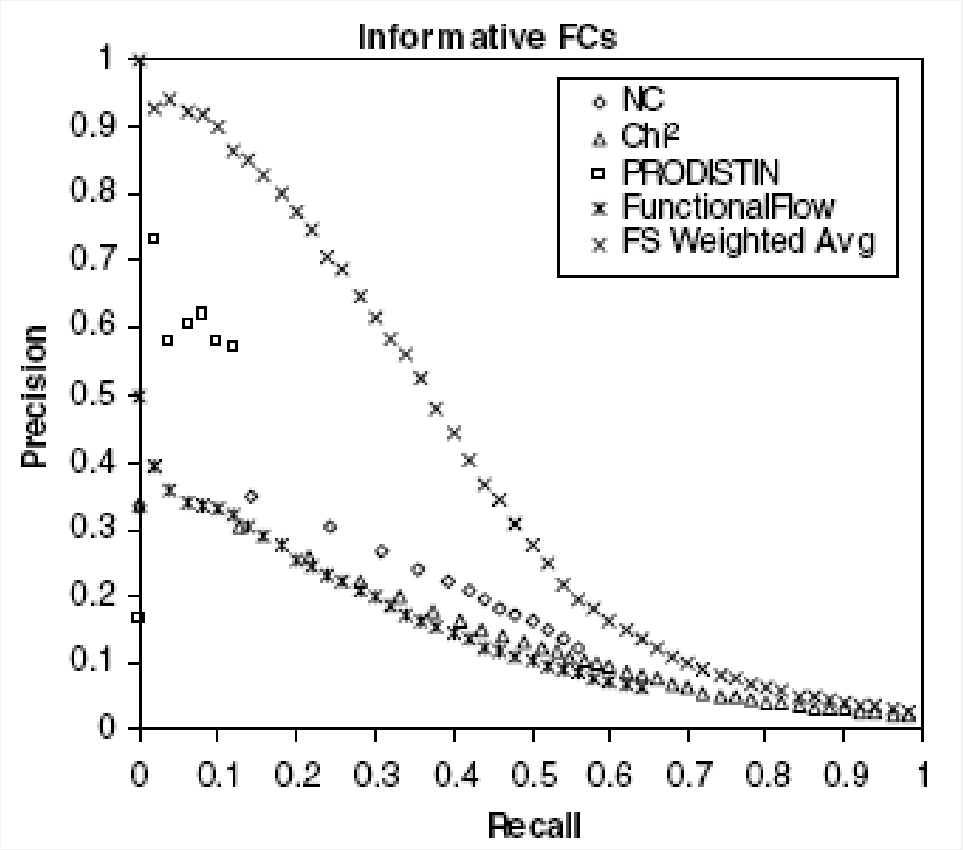
Performance of FS-weighted averaging

LOOCV comparison with:

Neighbour Counting

Chi-Square

PRODISTIN



About the inventor: Kenny Chua Hon Nian

Chua Hon Nian

PhD, NUS, 2008

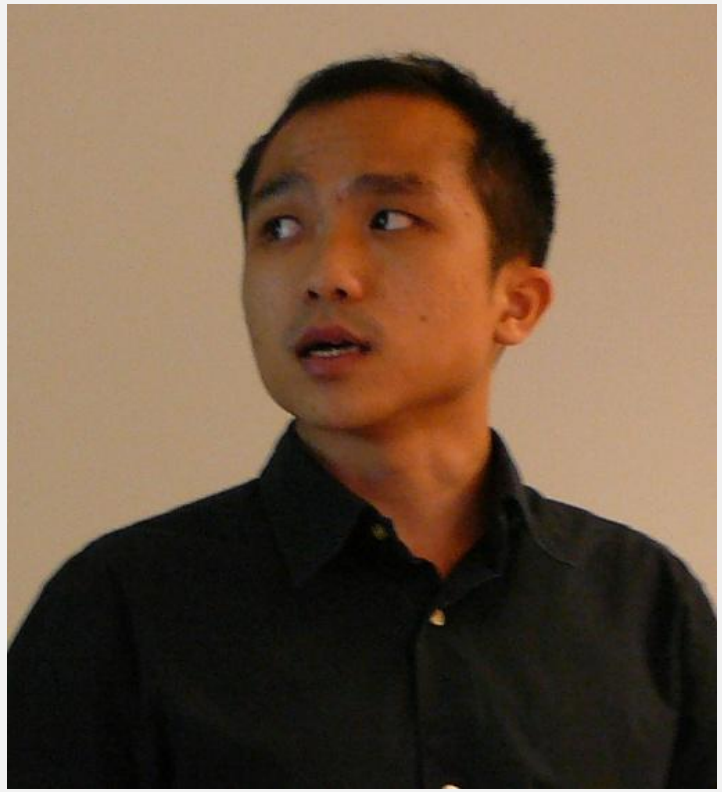
Postdoc at Harvard & Univ of Toronto

49th hottest paper in Computer Science published in 2006

Winner, DREAM2 challenge PPI subnetwork, 2007

Head of R&D at Data Robot, 2014 - 2022

Currently, CTO of FeatureByte



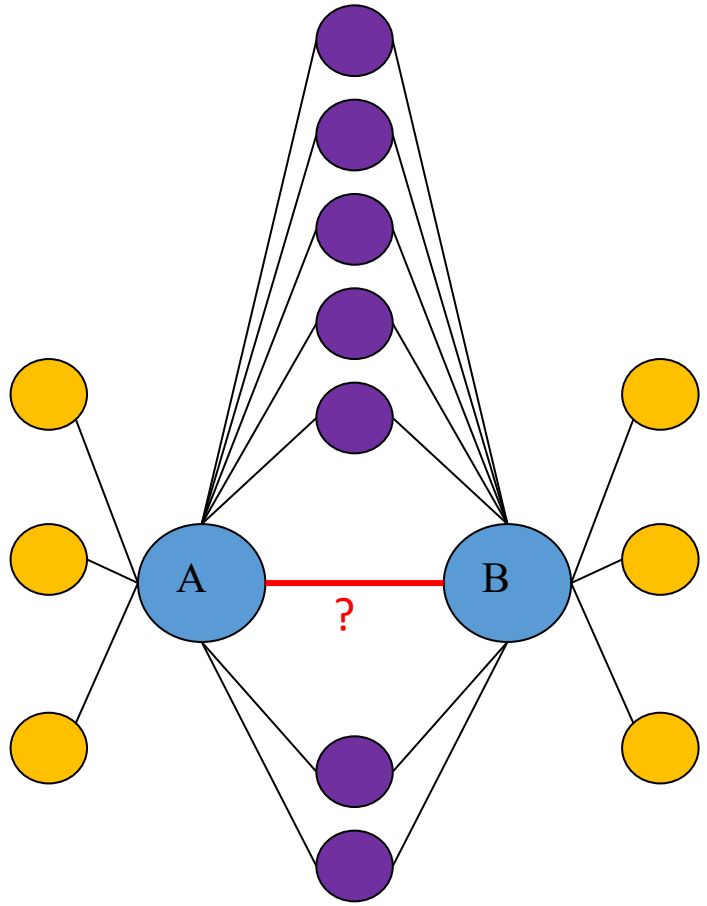
Exercise

Suggest a refinement to the concept of counting shared neighbours for assessing reliability of PPI

Hint

The PPI subnetwork shown here is only a small part of the complete PPI network

Proteins not connecting to A and B are omitted from the subnetwork



Closing remarks

“The ultimate goal of knowledge discovery is to reach a point where machine learning is no longer needed”

What do you think about this?

Good to read

Kabir & Wong, “EnsembleFam: Towards more accurate protein family prediction in the twilight zone,” *BMC Bioinformatics*, 23:90, 2022

Chua & Wong. “Increasing the reliability of protein interactomes,” *Drug Discovery Today*, 13(15/16):652-658, 2008

Chua, et al. “Exploiting indirect neighbours and topological weight to predict protein function from protein-protein interactions,” *Bioinformatics*, 22:1623-1630, 2006

Jaakkola, et al. “A discriminative framework for detecting remote homologies,” *JCB*, 7(1-2):95-114, 2000

Hawkins & Kihara. “Function prediction of uncharacterized proteins,” *JBCB*, 5(1):1-30, 2007