Computing is no more about programming than biology is about test tubes

Wong Limsoon

Golden thread of science

***Science* is characterized by**

- **Observing an invariant**
- **Proving that it is true, i.e., a law**
- **Exploiting it to solve problems logically**

Three types of logical inferences

- **Induction**
  - Socrates is a man
  - Socrates is mortal
  - ⇒All men are mortal,
    provided there is no counter example

- **Deduction**
  - All men are mortal
  - Socrates is a man
  - ⇒Socrates is mortal

- **Abduction**
  - All men are mortal
  - Socrates is mortal
  - ⇒Socrates is a man,
    provided there is no other explanation of Socrates' mortality

And two simple tactics

- **Fixing violation of invariants**
- **Guilt by association**

NUS National University of Singapore

INVARIANT & SCIENCE

- **Suppose you have a bag of x red beans and y green beans**
- **Repeat the following:**
  - Remove 2 beans
  - If both green, discard both
  - If both red, discard one, put back one
  - If one green and one red, discard red, put back green
- **If one bean is left behind, can you predict its colour?**

Shall we bet on the color of the bean that is left behind?

You can always win

- **Suppose you have a bag of x red beans and y green beans**
- **Repeat the following:**
  - Remove 2 beans
  - If both green, discard both
  - If both red, discard one, put back one
  - If one green and one red, discard red, put back green
- **If one bean is left behind, can you predict its colour?**

- **If you start w/ odd # (even #) of green beans, there will always be an odd # (even #) of green beans in the bag**

⇒ **Parity of green beans is invariant**

⇒ **Bean left behind is green iff you start with odd # of green beans**

- **What have we just seen?**
- **Problem solving by (deductive) logical reasoning on invariants**

Science is characterized by …

Observing an invariant:
Parity of green beans is invariant

Proving it:

Bet on the last red bean

• Suppose you have a bag of x red beans and y green beans	• When the parity of # of green beans (y) is even, …
• Repeat the following:	• Start with y=2n
– Remove 2 beans	
– If both green, discard both	• y=2n → y=2n-2
– If both red, discard one, put back one	• y=2n → y=2n
– If one green and one red, discard red, put back green	• y=2n → y=2n
• If one bean is left behind, can you predict its colour?	• y remains even ⇒ Last bean must be red!

Exploit it to solve problems:
Predict colour of the last bean

NUS National University of Singapore

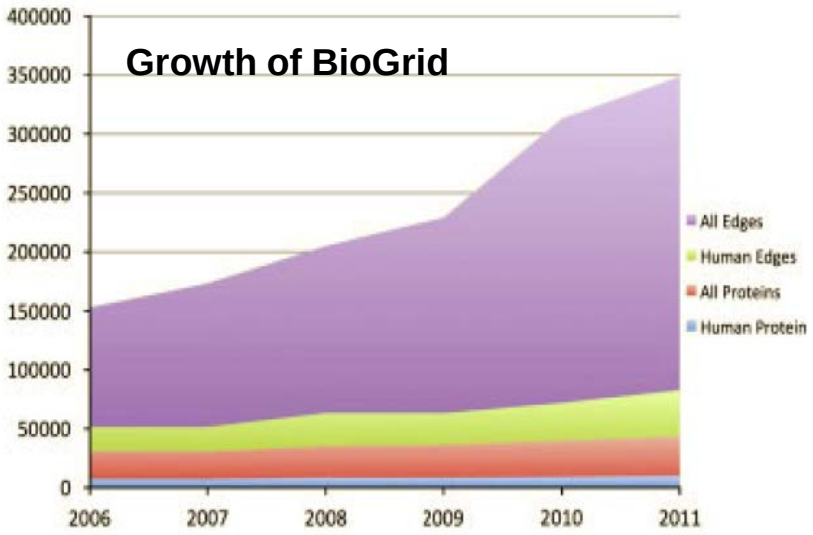
Deduction

REMOVING NOISE FROM PPI EXPERIMENTS

Protein-protein interaction detection

- **Many high-throughput assays for PPIs**

Generating *large amounts* of expt data on PPIs can be done with ease



- **But ...**

High-throughput approaches sacrifice quality for ***quantity***: (a) limited or biased coverage: ***false negatives***, & (b) high error rates: ***false positives***

Noise in PPI networks

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**

$\Rightarrow$ **Need to clean up**

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

NUS National University of Singapore

Time for Exercise #1

Can you think of things a biologist can do to remove PPIs that are likely to be noise?

De-noising PPI networks using Reproducibility

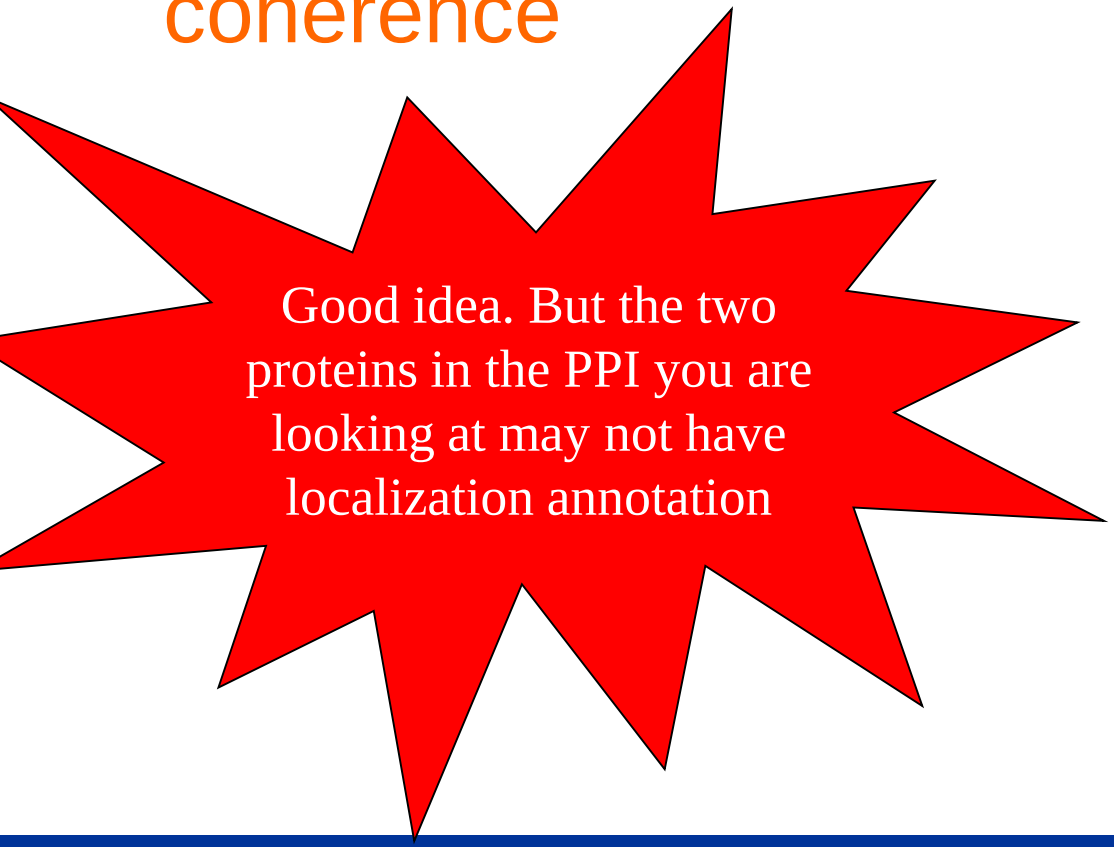
- **A PPI reported in several independent experiments is more reliable than those reported in only one experiment**

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

- 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

De-noising PPI networks using localization coherence

- **Two proteins should be in the same place to interact. Agree?**

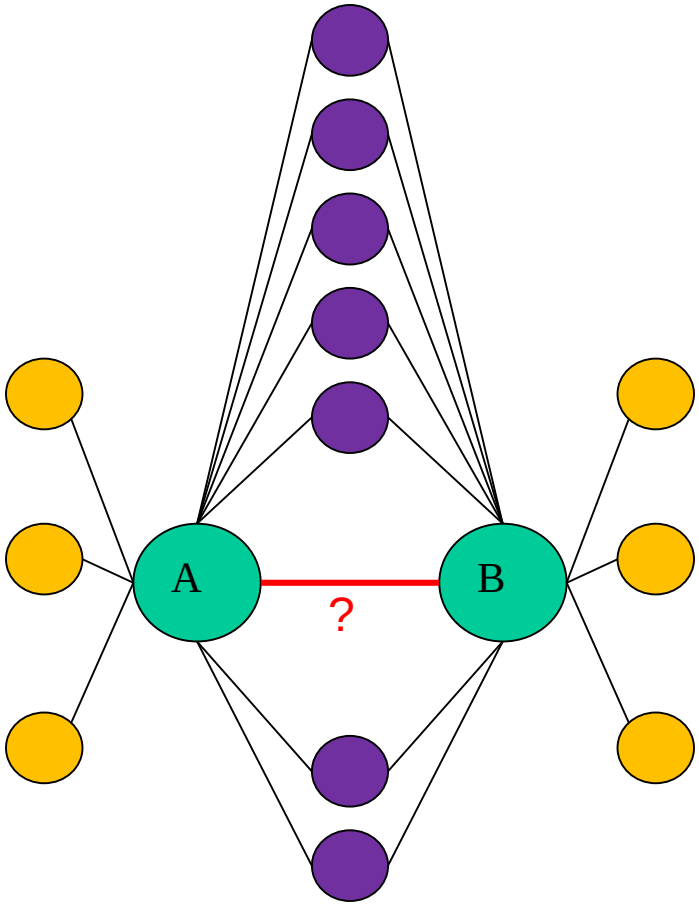


Liu et al. Complex discovery from weighted PPI networks.
Bioinformatics, 25(15):1891-1897, 2009

Time for Exercise #2

Do you really need to know where two proteins are, in order to know whether they are in the same place? If not, how?

Topology of neighbourhood of real PPIs



- **Suppose 20% of putative PPIs are noise**

⇒ **≥ 3 purple proteins are real partners of both A & B**

⇒ **A and B are likely localized to the same cellular compartment (Why?)**

⇒ **A and B are more likely PPI than not**

Brun, et al. *Genome Biology,* 5(1):R6, 2003

Czekanowski-Dice distance

Given a pair of proteins (u, v) in a PPI network

N_u = the set of neighbors of u

N_v = the set of neighbors of v

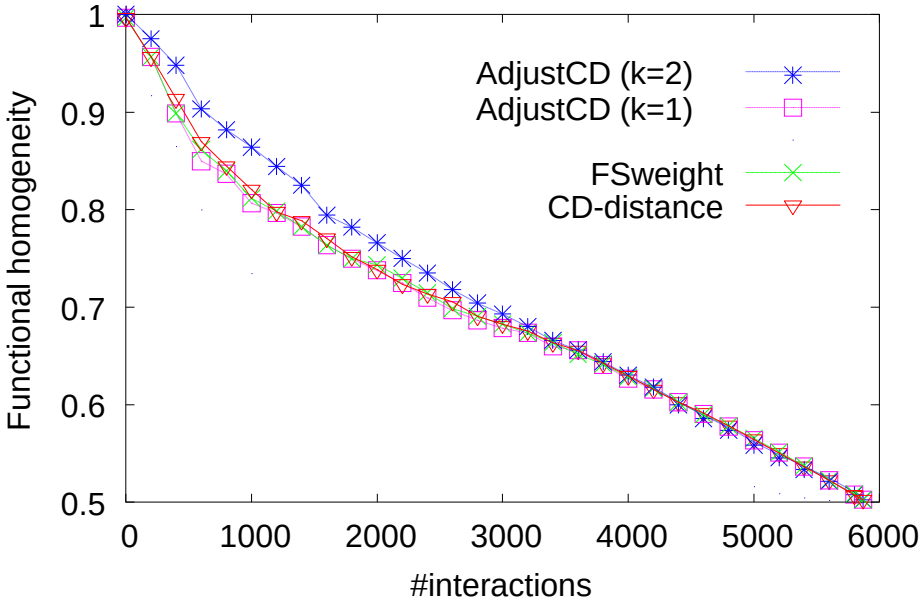
CD(u,v) = $$\frac{2 \mid N_u \cap N_v \mid}{\mid N_u \mid + \mid N_v \mid}$$

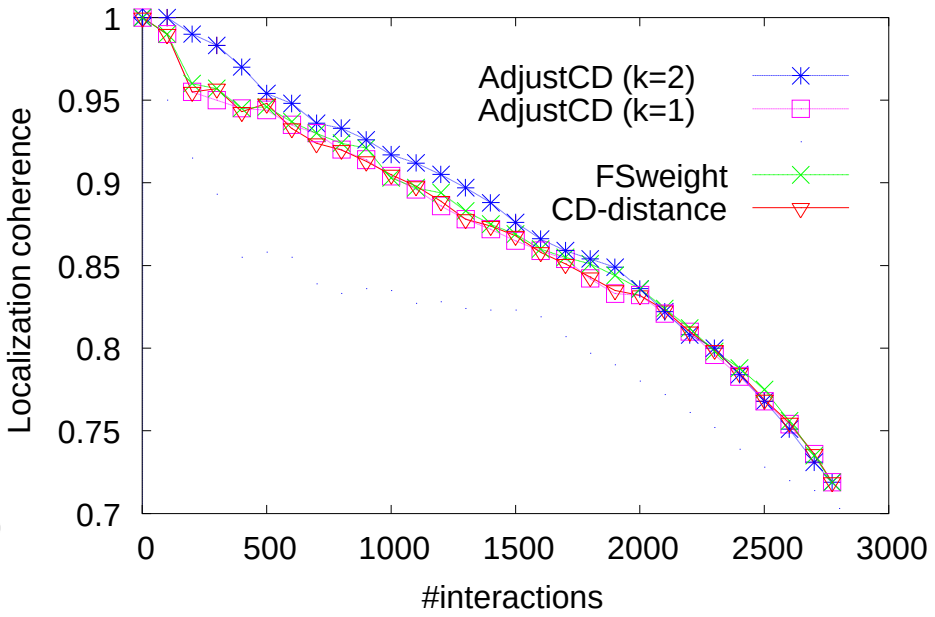
See also Liu et al. (*Bioinformatics* 2009, 25:1891-1897) for a simple modification of CD to make it more robust for biological & power law-like networks

Liu et al. Complex discovery from weighted PPI networks.
Bioinformatics, 25(15):1891-1897, 2009

Identifying false-positive PPIs

Cf. ave localization coherence of protein pairs in DIP < 5%
ave localization coherence of PPI in DIP < 55%

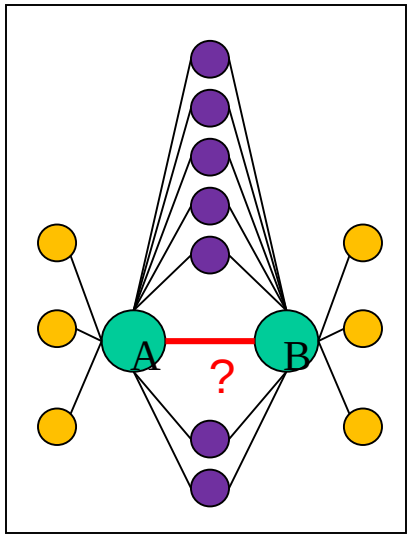




- **CD-distance and its variations correlate very well with functional homogeneity and localization coherence**

The triumph of logic

Two proteins should be in same place to interact



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

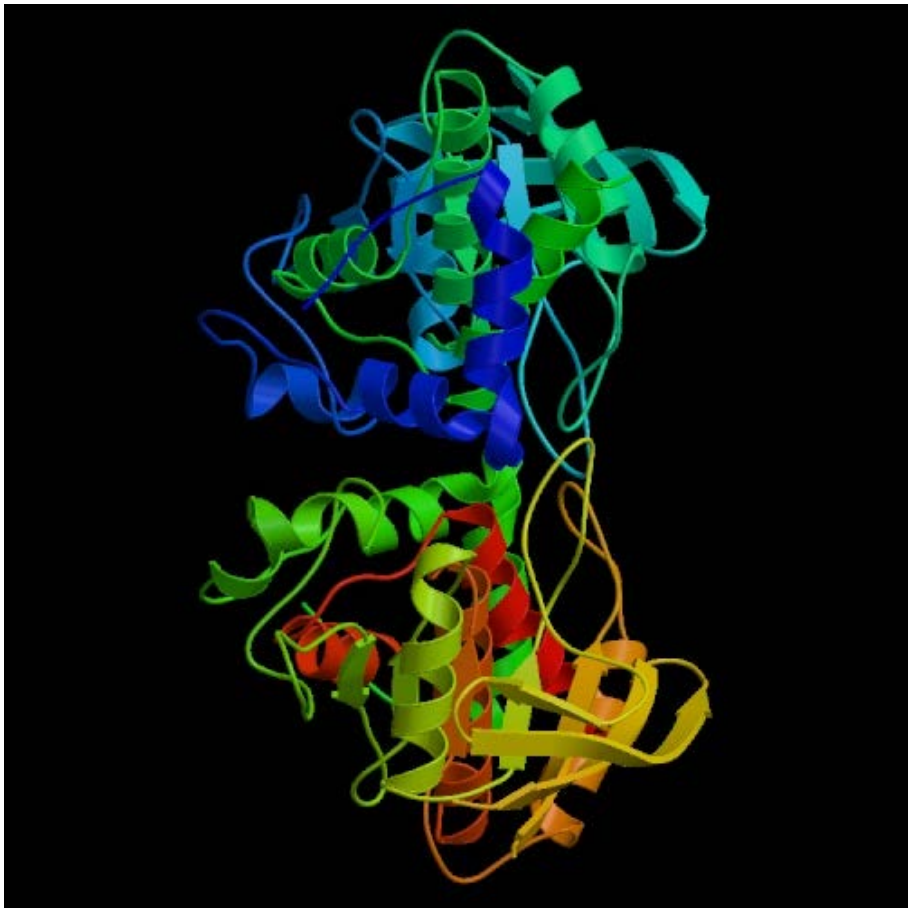
Impact:
PPI networks can be cleansed based on topological info, w/o needing location etc info on proteins

Deduction / induction

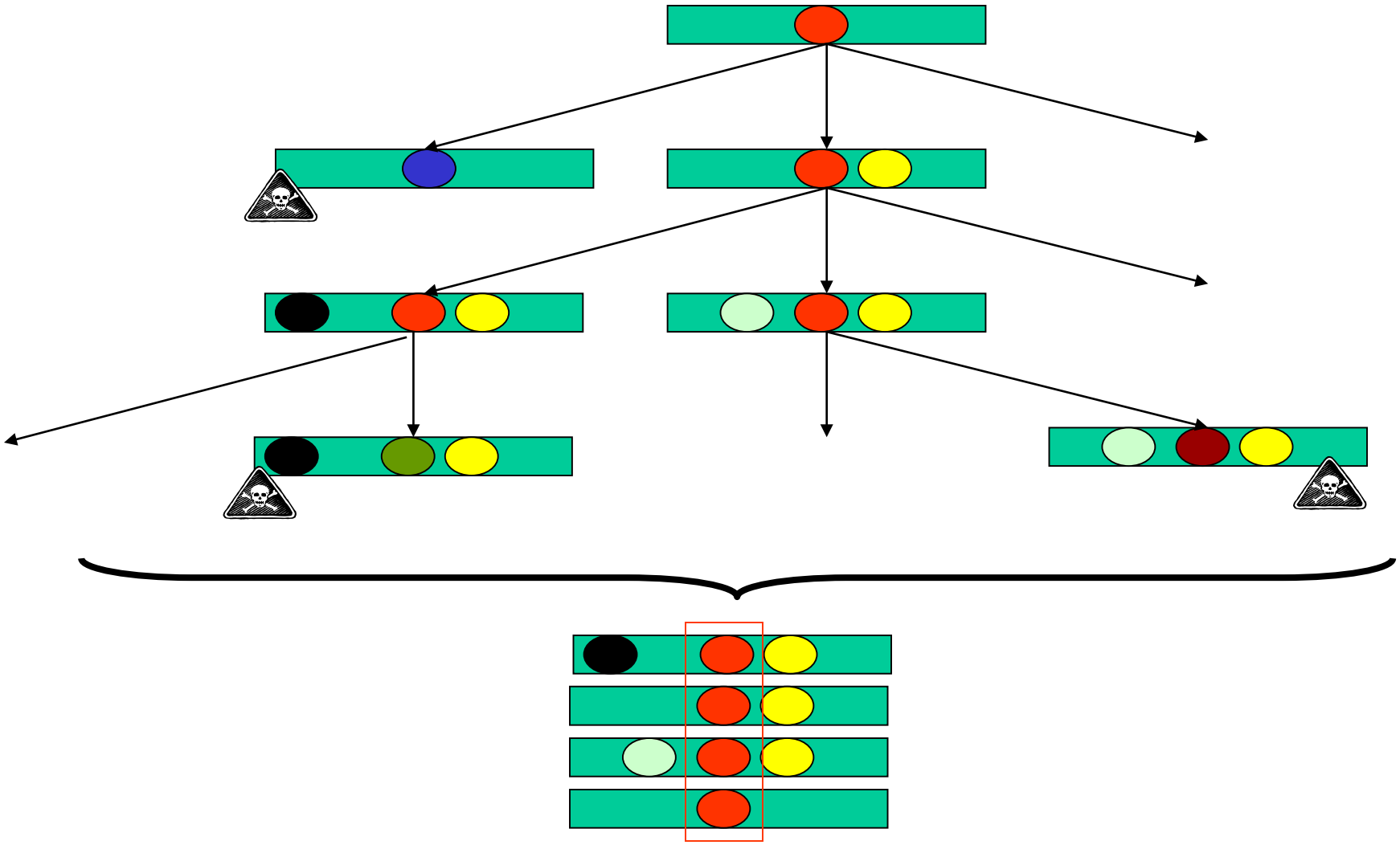
IDENTIFYING HOMOLOGOUS PROTEINS

A protein is a ...

- **A protein is a large complex molecule made up of one or more chains of amino acids**
- **Proteins perform a wide variety of activities in the cell**



In the course of evolution…



Time for Exercise #3

Let a = AFP HQH RVP
Let b = PQV YNI MKE

Suppose each generation differs from the previous by 1 residue

What is the max difference between the 2nd generation of a

What is the min difference between the 2nd generation of a and b?

In the course of evolution…

a = AFP HQH RVP

b = PQV YNI MKE

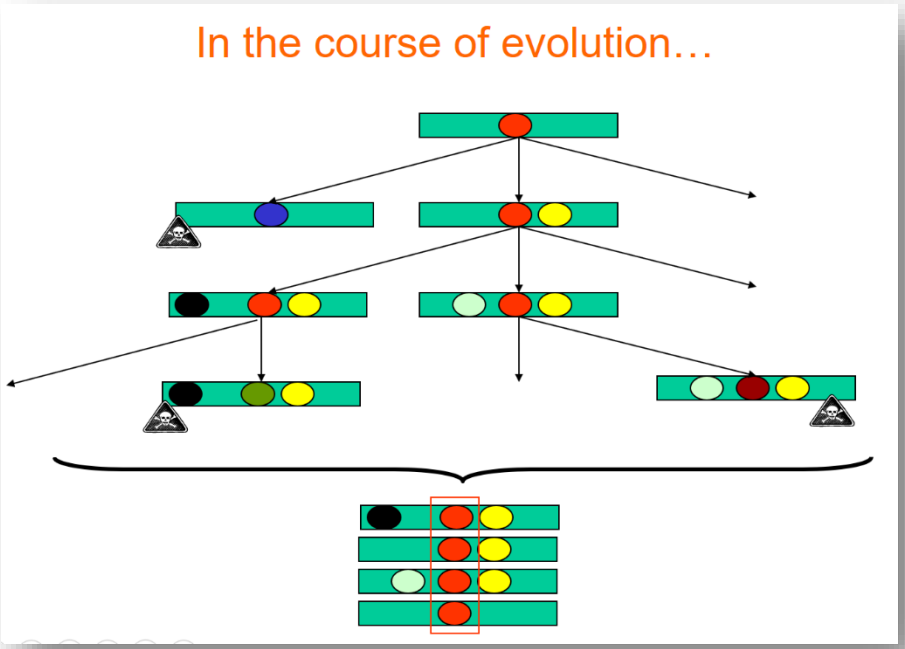
Each gen differs from its parent by 1 residue

Each 2nd-gen of a differs from a by 2 residues and two 2nd-gen of a differ by at most 4 residues

a and b differ in 9 residues

Each 2nd-gen of b differs from b by 2 residues and so differs from a by at least 7 residues; thus each 2nd-gen of b differs from each 2nd-gen of a by at least 5 residues

The triumph of logic



Two proteins inheriting their function from a common ancestor have very similar amino acid sequences

Abduction

PROTEIN FUNCTION PREDICTION

Function assignment to a protein seq

NUS National University of Singapore

**SPSTNRKYPPLPVDKLEEEINRRMADDNKLFREEFNALPACPIQATCEAASKEENKEKNR
YVNILPYDHSRVHLTPVEGVPDSDYINASFINGYQEKNKFIAAQGPKEETVNDFWRMIWE
QNTATIVMVTNLKERKECKCAQYWPDQGCWTYGNVRVSVEDVTVLVDYTVRKFCIQQVGD
VTNRKPQRLITQFHFTSWPDFGVPFTPIGMLKFLKKVKACNPQYAGAIVVHCSAGVGRTG
TFVVIDAMLDMMHSERKVDVYGFVSRIRAQRCQMVQTDMQYVFIYQALLEHYLYGDTELE
VT**

- **How do we attempt to assign a function to a new protein sequence?**

NUS National University of Singapore

Time for Exercise #4

How can we guess the function of a protein?

Abductive reasoning

- **Law: Two proteins inheriting their function from a common ancestor have very similar amino acid sequences**
- **Observation: Proteins X and Y are very similar in their sequence**
- **Abduction: Proteins X and Y are descended from the same ancestor and inherit their function from this ancestor**

$\Rightarrow$ **Proteins X and Y have a common function**

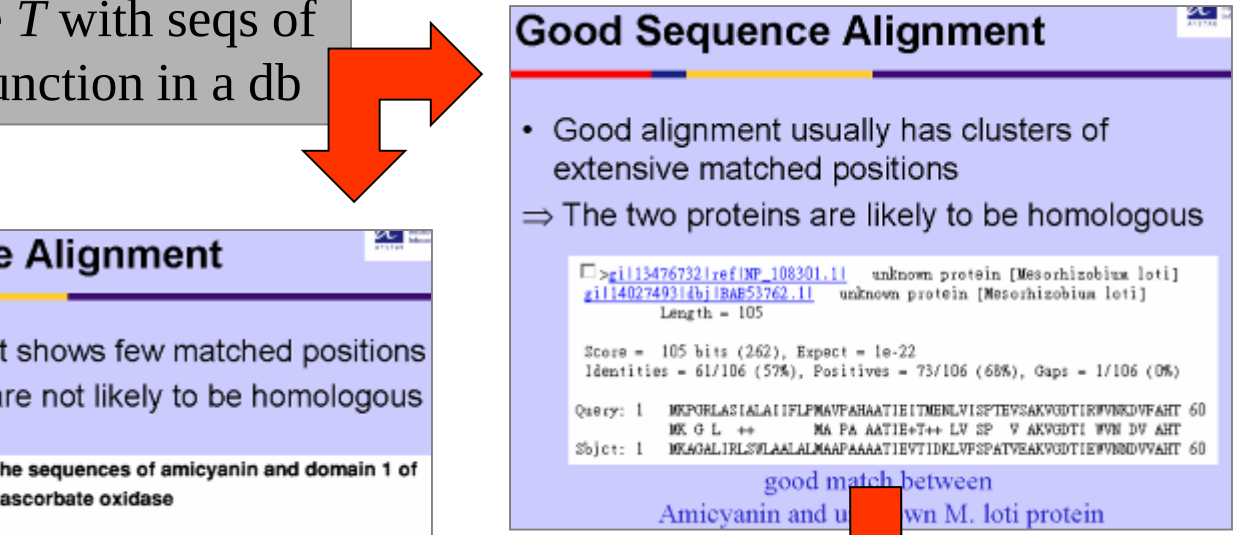
Guilt by association

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

Good Sequence Alignment

- Good alignment usually has clusters of extensive matched positions

⇒ The two proteins are likely to be homologous



good match between Amicyanin and u[illegible]wn M. loti protein

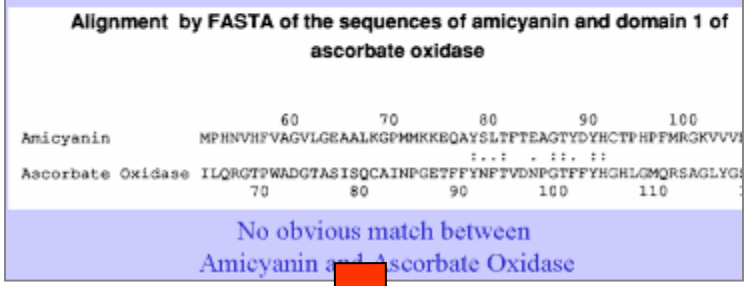
Assign to *T* same function as homologs

Confirm with suitable wet experiments

Poor Sequence Alignment

- Poor seq alignment shows few matched positions

⇒ The two proteins are not likely to be homologous



No obvious match between Amicyanin and Ascorbate Oxidase

Discard this function as a candidate

Earliest research in seq comparison

Source: Ken Sung

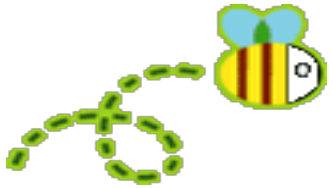
- **Doolittle et al. (*Science*, July 1983) searched for platelet-derived growth factor (PDGF) in his own DB. He found that PDGF is similar to v-sis oncogene**

```
PDGF-2   1       SLGSLTIAEPAMIAECKTREEVFCICRRL?DR??  34
p28sis  61 LARGKRSLGSLSVAEPAMIAECKTRTEVFEISRRLIDRTN 100
```

Guilt by other types of associations

- **Similarity of phylogenetic profiles**
- **Similarity of dissimilarities**
- **Similarity of subcellular co-localization & other physico-chemico properties**
- **Similarity of gene expression profiles**
- **Similarity of protein-protein interaction partners**
- **…**

Guilt by association of dissimilarities

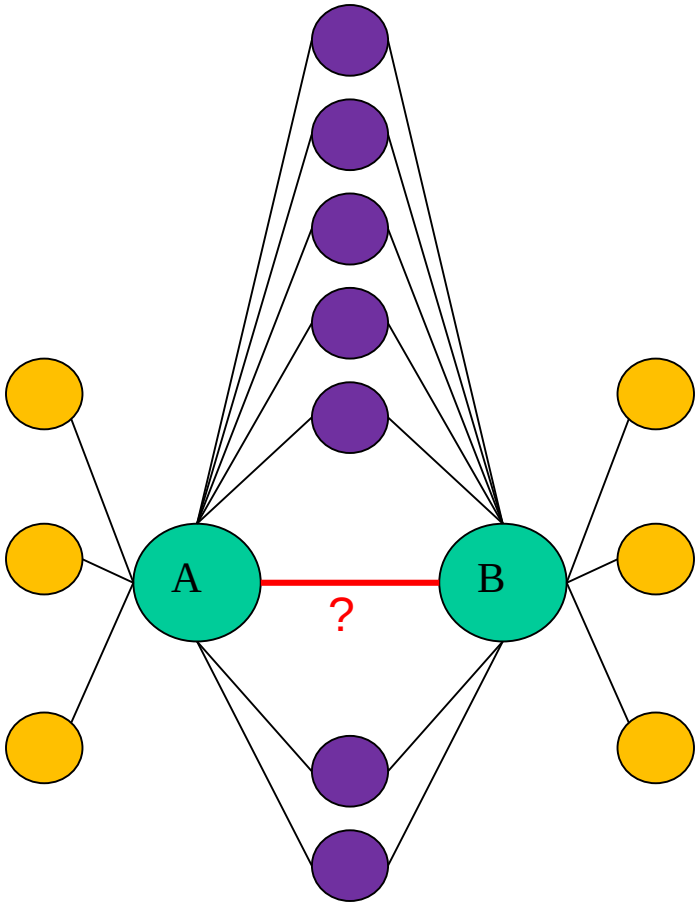


Differences of "unknown" to other fruits are same as "apple" to other fruits

"unknown" is an "apple"!

	Orange$_1$	Banana$_1$	...
Apple$_1$	Color = red vs orange Skin = smooth vs rough **Size = small vs small** **Shape = round vs round**	Color = red vs yellow **Skin = smooth vs smooth** **Size = small vs small** Shape = round vs oblong	...
Orange$_2$	**Color = orange vs orange** **Skin = rough vs rough** **Size = small vs small** **Shape = round vs round**	Color = orange vs yellow **Skin = rough vs smooth** **Size = small vs small** Shape = round vs oblong	...
Unknown$_1$	**Color = red vs orange** **Skin = smooth vs rough** Size = small vs small Shape = round vs round	**Color = red vs yellow** Skin = smooth vs smooth Size = small vs small **Shape = round vs oblong**	...
...	...	...	...

Guilt by association of neighbourhood



- **Suppose 20% of putative PPIs are noise**

⇒ **≥ 3 purple proteins are real partners of both A & B**

⇒ **A and B are likely localized to the same cellular compartment (Why?)**

⇒ **A and B are more likely PPI than not**

NUS National University of Singapore

Violation of invariant

MAKING COMPUTER SYSTEMS MORE SECURE

COMPUTERWORLD An IDG company

RSA: Microsoft on 'rootkits': Be afraid, be very afraid

Rootkits are a new generation of powerful system-monitoring programs

News Story by Paul Roberts

FEBRUARY 17, 2005 (IDG NEWS SERVICE) - Microsoft Corp. security researchers are warning about a new generation of powerful system-monitoring programs, or "rootkits," that are almost impossible to detect using current security products and could pose a serious risk to corporations and individuals..... the only reliable way to remove kernel rootkits is to completely erase an infected hard drive and reinstall the operating system from scratch.....

Credit: Bill Arbaugh

Rootkit Problem

Traditional rootkits

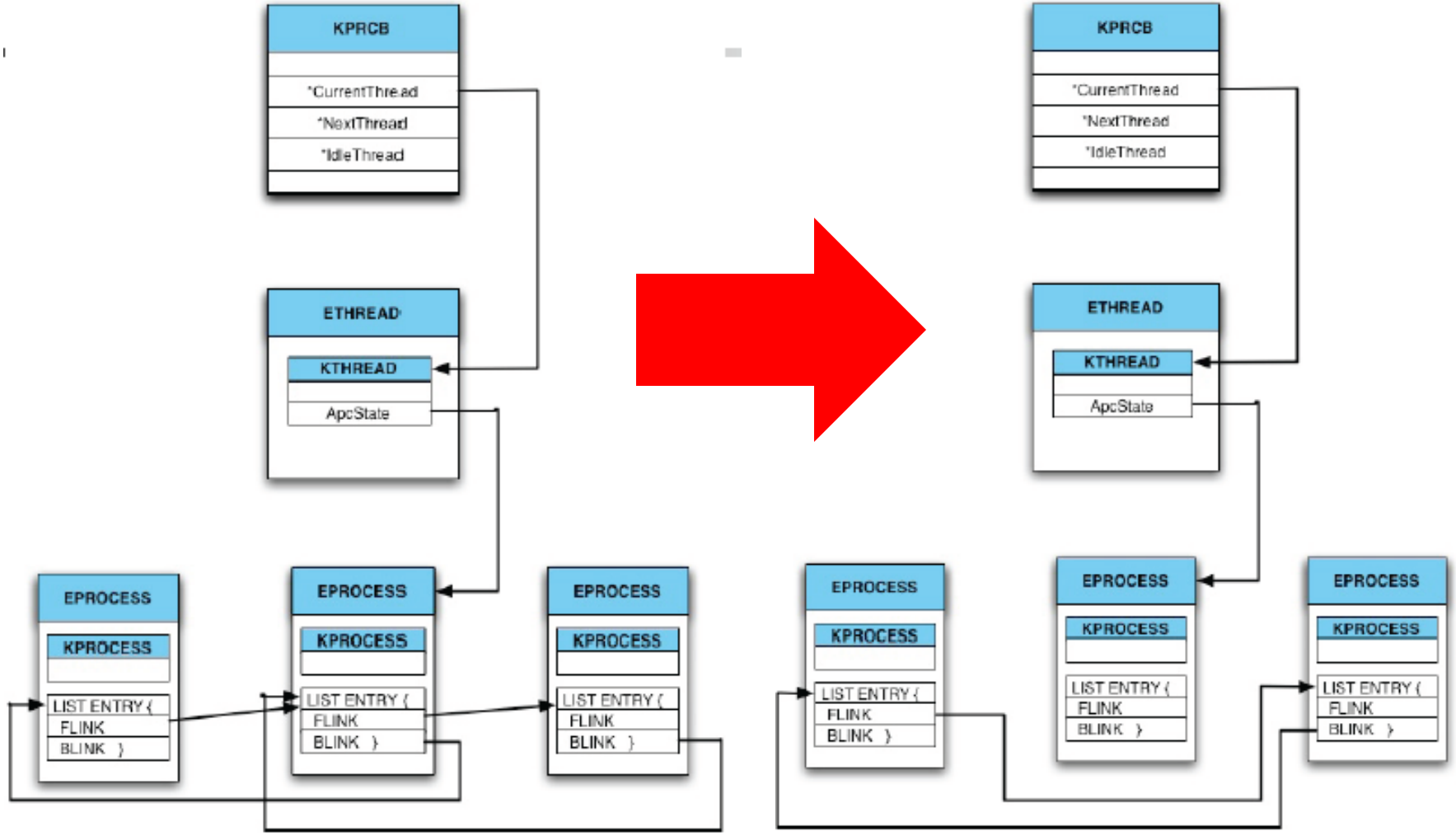
- **Modify static scalar invariants in OS**
  - kernel text
  - interrupt table
  - syscall table

Modern rootkits

- **Direct Kernel Object Manipulation (DKOM)**
- **Rather than modify scalar invariants in OS, dynamic data of kernel are modified to:**
  - Hide processes
  - Increase privilege level

Credit: Bill Arbaugh

Hiding a window process



Semantic integrity

- **Earlier integrity monitoring systems focus on the scalar / static nature of the monitored data**
  - Don't work for non-scalar / dynamic data

- **Current systems rely on semantic integrity**
  - Monitor non-invariant portions of a system via predicates that remain valid during the proper operation of the system
  - I.e., monitor invariant dynamic properties!

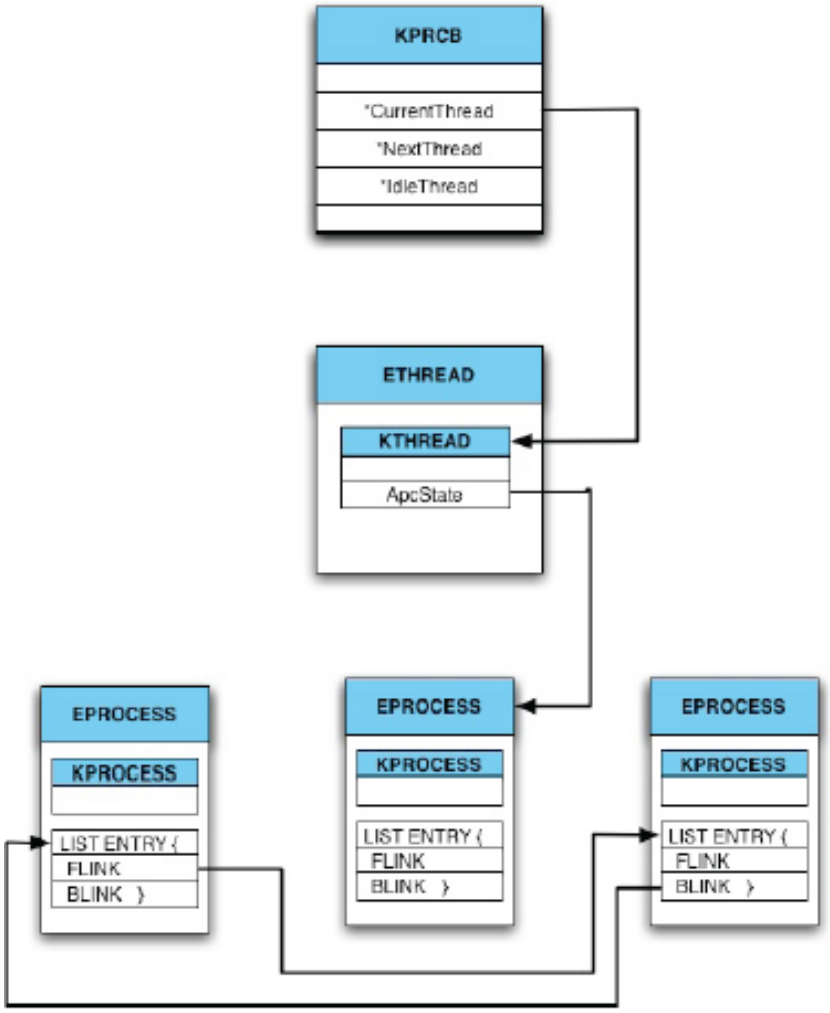
Credit: Bill Arbaugh

DKOM Example

- **Semantic integrity predicate (ie., dynamic invariant) is**

There is no thread such that its parent process is not on the process list

⇒ **kHIVE (contains 20k other predicates)**



- **What have we just seen?**

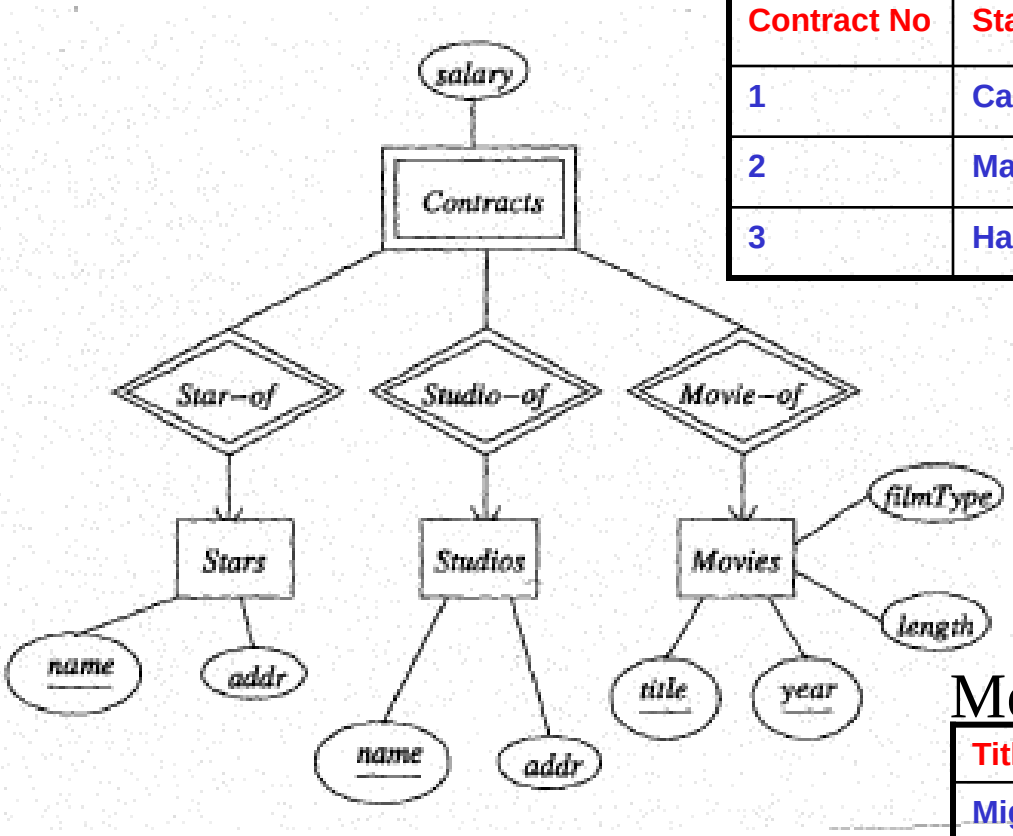
- **Maintain computer safety by checking violation of invariants!**

NUS National University of Singapore

Violation of invariant

IMPROVING DATABASE DESIGN

Relational data model



Contracts

Contract No	Star	Studio	Title	Salary
1	Carrie Fisher	Fox	Star Wars	$$$
2	Mark Hamill	Fox	Star Wars	$$$
3	Harrison Ford	Fox	Star Wars	$$$

Stars

Name	Address
Carrie Fisher	Hollywood
Mark Hamill	Brentwood
Harrison Ford	Beverly Hills

Movies

Title	Year	Length	Film Type
Mighty Ducks	1991	104	Color
Wayne's World	1992	95	Color
Star Wars	1977	124	Color

Design issues

- **How many possible alternate ways to represent movies using tables?**
- **Why this particular set of tables to represent movies?**
- **Indeed, why not use this alternative single table below to represent movies?**

Wrong Movies

Title	Year	Length	Film Type	Studio	Star
Star Wars	1997	124	Color	Fox	Carrie Fisher
Star Wars	1997	124	Color	Fox	Mark Hamill
Star Wars	1997	124	Color	Fox	Harrison Ford
Mighty Ducks	1991	104	Color	Disney	Emilio Estevez

Exercise #5

What's wrong with the "Wrong Movies" table?

Wrong Movies

Title	Year	Length	Film Type	Studio	Star
Star Wars	1997	124	Color	Fox	Carrie Fisher
Star Wars	1997	124	Color	Fox	Mark Hamill
Star Wars	1997	124	Color	Fox	Harrison Ford
Mighty Ducks	1991	104	Color	Disney	Emilio Estevez

Anomalies

- **What's wrong with the "Wrong Movies" table?**

Wrong Movies

Title	Year	Length	Film Type	Studio	Star
Star Wars	1997	124	Color	Fox	Carrie Fisher
Star Wars	1997	124	Color	Fox	Mark Hamill
Star Wars	1997	124	Color	Fox	Harrison Ford
Mighty Ducks	1991	104	Color	Disney	Emilio Estevez

- **Redundancy: Unnecessary repetition of info**
- **Update anomalies: If Star Wars is 125 min, we might carelessly update row 1 but not rows 2 & 3**
- **Deletion anomalies: If Emilio Estevez is deleted from stars of Mighty Ducks, we lose all info on that movie**

Some interesting questions

- **How to differentiate a good database design from a bad one?**

- **How to produce a good database design automatically from a bad one?**

Functional dependency

- **Functional dependency (A_1, …, A_n → B_1, …, B_m)**
  - If two rows of a table R agree on attributes A_1, …, A_n, then they must also agree on attributes B_1, …, B_m

  ⇒ Values of B's depend on values of A's
- **FD (A_1, …, A_n → B_1, …, B_m) is trivial if a B_i is an A_j**

Wrong Movies

Title	Year	Length	Film Type	Studio	Star
Star Wars	1997	124	Color	Fox	Carrie Fisher
Star Wars	1997	124	Color	Fox	Mark Hamill
Star Wars	1997	124	Color	Fox	Harrison Ford
Mighty Ducks	1991	104	Color	Disney	Emilio Estevez

- **Example: Title, Year → Length, Film Type, Studio**

Keys

- **Key is a minimal set of attributes {A_1, …, A_n} that functionally determine all other attributes of a table**
- **Superkey is a set of attributes that contains a key**

Wrong Movies

Title	Year	Length	Film Type	Studio	Star
Star Wars	1997	124	Color	Fox	Carrie Fisher
Star Wars	1997	124	Color	Fox	Mark Hamill
Star Wars	1997	124	Color	Fox	Harrison Ford
Mighty Ducks	1991	104	Color	Disney	Emilio Estevez

- **Example superkey: Any set of attributes that contains {Title, Year, Star} as a subset**

Boyce-Codd Normal Form

A relation R is in Boyce-Codd Normal Form iff whenever there is a nontrivial FD (A_1, …, A_n → B_1, …, B_m) for R, it is the case that {A_1, …, A_n} is a superkey for R

Theorem (Codd, 1972)

A database design has no anomalies due to FD iff all its relations are in Boyce-Codd Normal Form

How is BCNF violated here?

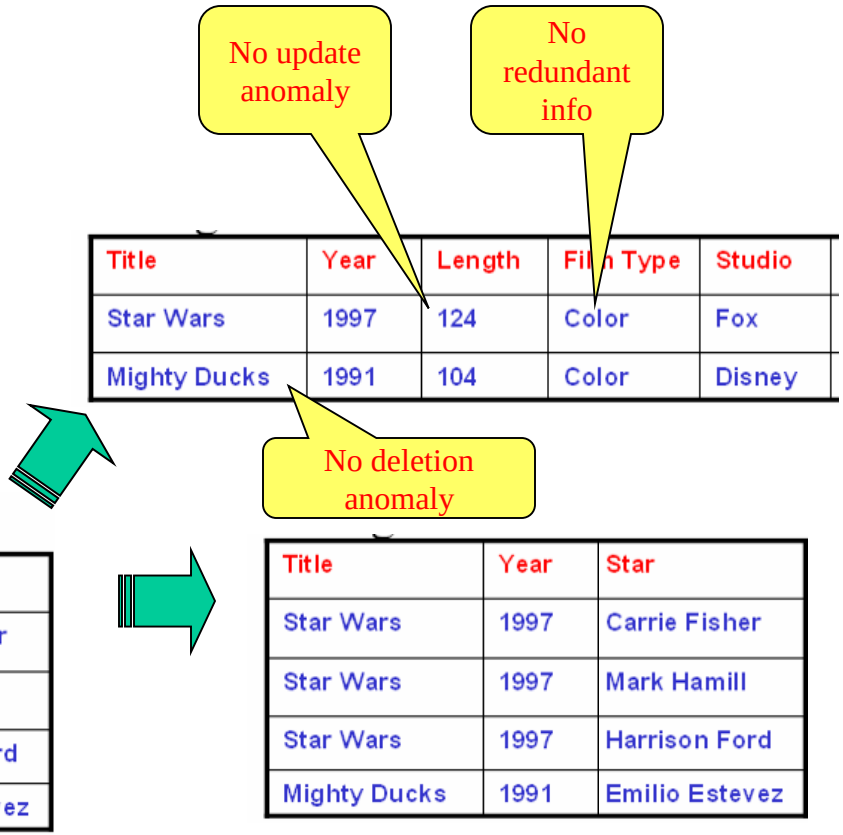
Title	Year	Length	Film Type	Studio	Star
Star Wars	1997	124	Color	Fox	Carrie Fisher
Star Wars	1997	124	Color	Fox	Mark Hamill
Star Wars	1997	124	Color	Fox	Harrison Ford
Mighty Ducks	1991	104	Color	Disney	Emilio Estevez

- **A nontrivial FD:**
  - Title, Year → Length, Film Type, Studio
  - The LHS not superset of the key {Title,Year, Star}

  ⇒ Violate BCNF!
- **Anomalies are due to FD's whose LHS is not superkey**

Towards a better design

- **Use an *offending* FD (A_1, …, A_n → B_1, …, B_m) to decompose R(A_1, …, A_n, B_1, …, B_m, C_1, …, C_h) into 2 tables**
  - $R_1(A_1, \ldots, A_n, B_1, \ldots, B_m)$
  - $R_2(A_1, \ldots, A_n, C_1, \ldots, C_h)$



Title	Year	Length	Film Type	Studio
Star Wars	1997	124	Color	Fox
Mighty Ducks	1991	104	Color	Disney

Wrong Movies

Title	Year	Length	Film Type	Studio	Star
Star Wars	1997	124	Color	Fox	Carrie Fisher
Star Wars	1997	124	Color	Fox	Mark Hamill
Star Wars	1997	124	Color	Fox	Harrison Ford
Mighty Ducks	1991	104	Color	Disney	Emilio Estevez

Title	Year	Star
Star Wars	1997	Carrie Fisher
Star Wars	1997	Mark Hamill
Star Wars	1997	Harrison Ford
Mighty Ducks	1991	Emilio Estevez

The “Invariant” Perspective

- **The invariants:**

  BCNF is an invariant of a good database design

- **The lesson learned:**

Deliver a better database design by fixing violated invariants

NUS National University of Singapore

Induction / fixing violated invariants

INFERRING KEY MUTATIONS: WHY SOME PTP IS INEFFICIENT

Protein tyrosine phosphatase

Sequence from a typical PTP

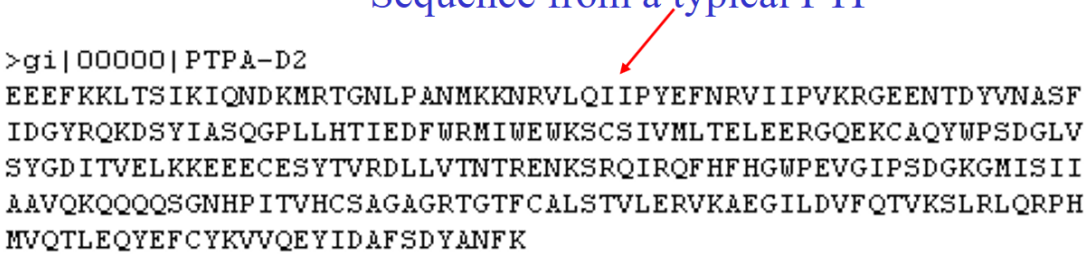
```
>gi|OOOOO|PTPA-D2
EEEFKKLTSIKIQNDKMRTGNLPANMKKNRVLQIIPYEFNRVIIPVKRGEENTDYVNASF
IDGYRQKDSYIASQGPLLHTIEDFWRMIWEWKSCSIVMLTELEERGQEKCAQYWPSDGLV
SYGDITVELKKEEECESYTVRDLLVTNTRENKSRQIRQFHFHGWPEVGIPSDGKGMISII
AAVQKQQQQSGNHPITVHCSAGAGRTGTFCALSTVLERVKAEGILDVFQTVKSLRLQRPH
MVQTLEQYEFCYKVVQEYIDAFSDYANFK
```

- **Some PTPs are much less efficient than others**
- **Why? And how do you figure out which mutations cause the loss of efficiency?**

Exercise #6

Protein tyrosine phosphatase

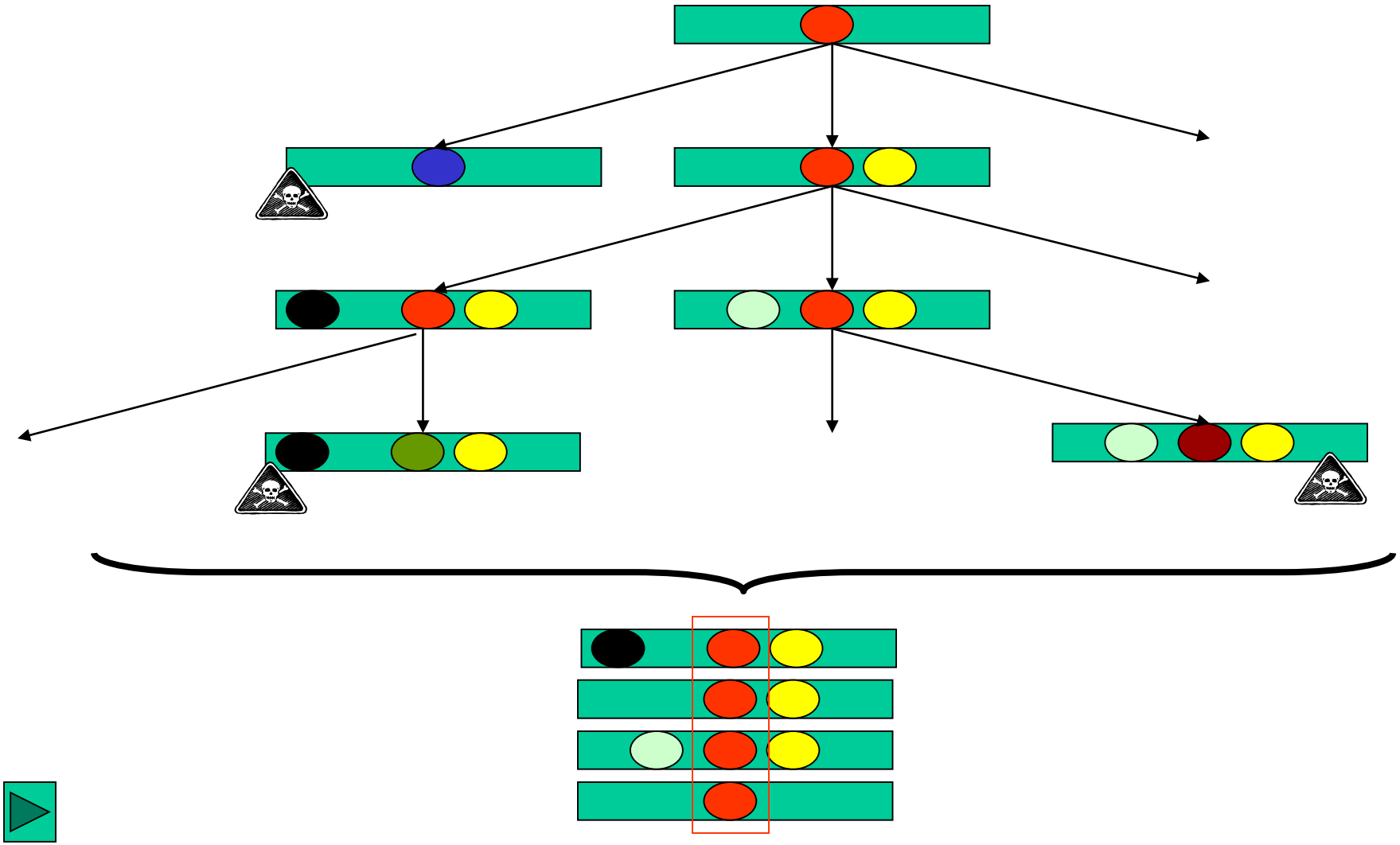
Sequence from a typical PTP



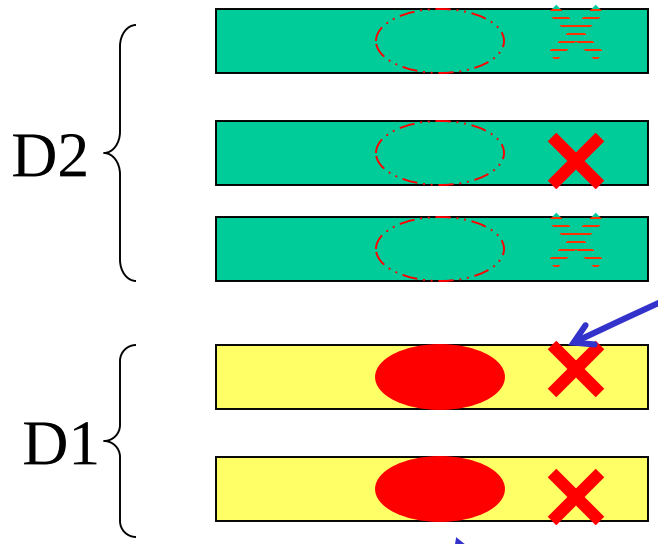
- **Some PTPs are much less efficient than others**

How do you figure out which mutations cause the loss of efficiency?

Some sites are impt for PTP function

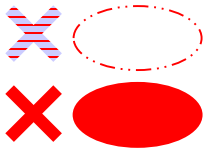


Reasoning based on an invariant…

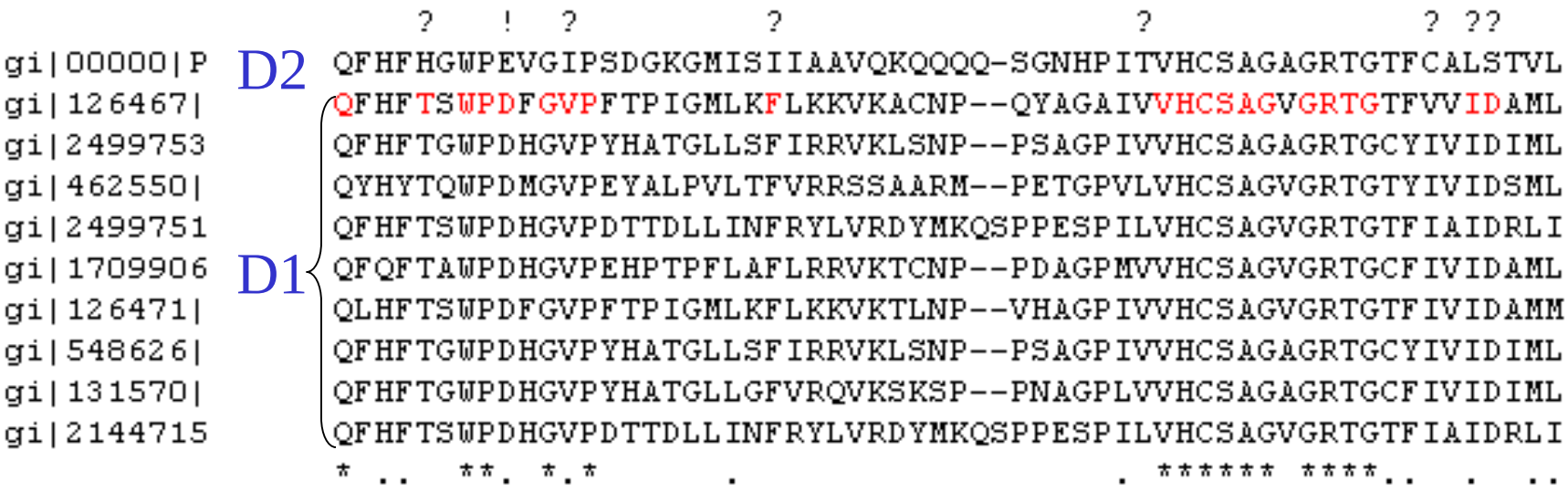


This site is conserved in D1, but is not consistently missing in D2
⇒ Not a likely cause of D2's loss of function

This site is conserved in D1, but is consistently missing in D2
⇒ Possible cause of D2's loss of function



Key mutation site: PTP D1 vs D2



- **Positions marked by “!” and “?” are likely places responsible for reduced PTP activity**
  - All PTP D1 agree on them
  - All PTP D2 disagree on them

Lim et al. Journal of Biological Chemistry 273:28986-28993,1998.

Confirmation by mutagenesis expt

- **Wet experiments confirm the predictions**
  - Mutate D → E in D1
    - **i.e., check if D → E can cause efficiency loss**
  - Mutate E → D in D2
    - **i.e., show D → E is the cause of efficiency loss**

Impact:
Hundreds of mutagenesis expts saved by simple reasoning on (violation of) invariants!

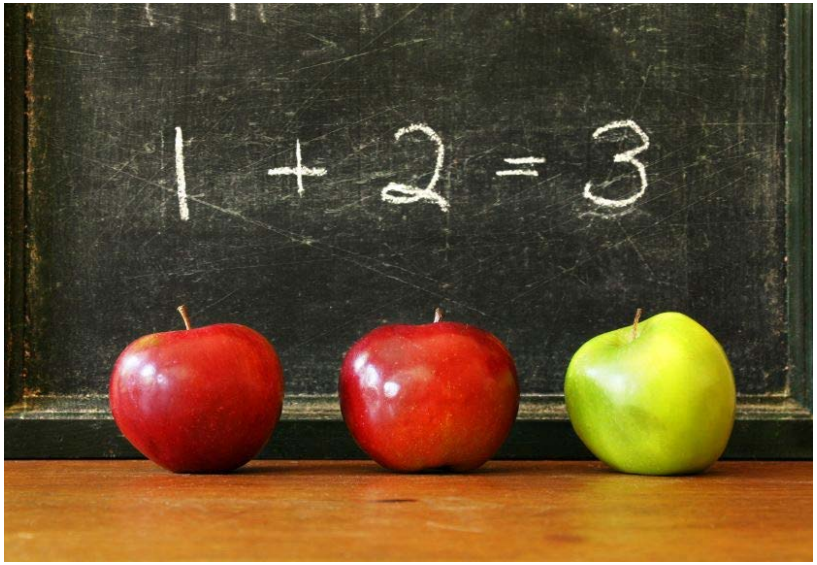
The triumph of logic

- **Induction/hypothesis: A site that is critical for PTP efficiency is present in all efficient PTPs and absent in all inefficient PTPs**
- **Observation: A site X is present in all efficient PTPs and absent in all inefficient PTPs**
- **Abduction: Site X is critical for PTP efficiency**

Bioengineering more efficient PTP

- **Replace an inefficient PTP in the organism by an efficient version**
  - Mutate E → D in D2

- **What have we just seen?**
- **Create a more efficient PTP by fixing a violated invariant!**



SUMMARY

What have we learned?

- **Three types of logical reasoning**
- **Invariant is a fundamental property of many problems**
- **Tactics of logical problem solving**
  - Problem solving by logical reasoning on invariants
  - Problem solving by rectifying/monitoring violation of invariants
  - Guilt by association of invariants