# Logic in computational biology

Wong Limsoon



#### **About Limsoon**



Kwan-Im-Thong-Hood-Cho-Temple Chair Professor, Dept of Computer Science, NUS

#### Research

database systems & theory, knowledge discovery, bioinformatics & computational biology

#### Honours

- ACM Fellow
- FEER Asian Innovation Gold Award 2003
- ICDT Test of Time Award 2014

#### 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

Biology is no more about Petri dish & test tube than Computational biology is about programming



# Three types of logical inferences

#### Deduction

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

#### Induction

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

#### Abduction

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

# Logic in computational biology



- Deduction
  - De-noise PPI networks
- Deduction/Induction
  - Identify homologs
- Induction/Abduction
  - Infer key mutations
  - Diagnose pediatric leukemias
  - Predict protein function
- Practical tactics for the above
  - Fixing violation of invariants
  - Guilt by association of invariants

Logical inference based on bio/physical invariants



# **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:

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

#### Bet on the last red bean



- 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?

- When the parity of # of green beans (y) is even, ...
- Start with y=2n
- y=2n → y=2n-2
- y=2n → y=2n
- y=2n → y=2n
- y remains even
- ⇒ Last bean must be red!



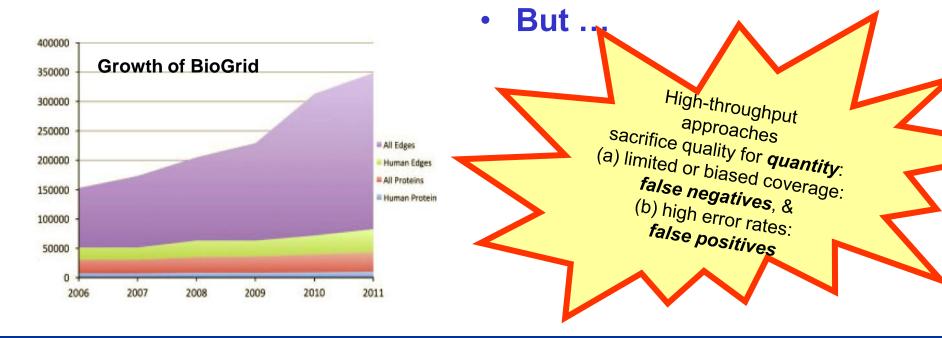
#### **Deduction**

# REMOVING NOISE FROM PPI EXPERIMENTS

# Protein-protein interaction detection National University of Singapore

Many high-throughput assays for PPIs

Generating <u>large amounts</u>
of expt data on PPIs can be
done with ease



### Noise in PPI networks



Experimental method category <sup>a</sup>	Number of interacting pairs	Co-localization <sup>b</sup> (%)	Co-cellular-role <sup>b</sup> (%)
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

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



#### 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

Good idea. But you need to do more expts → More time & more \$ has to be spent

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

- r<sub>i</sub> is reliability of expt source i,
- E<sub>u,v</sub> 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?

Good idea. But the two proteins in the PPI you are looking at may not have localization annotation

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

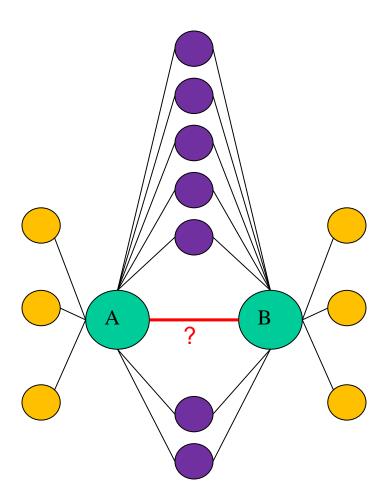


#### Time for Exercise #2

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

### 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

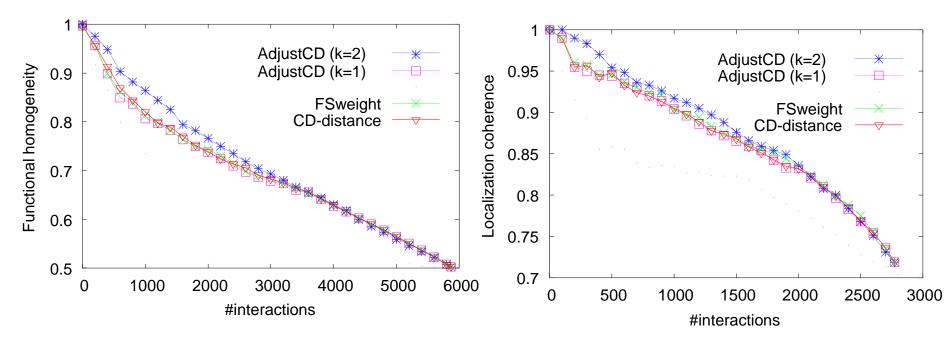
$$CD(u,v) = \frac{2 |N_u \cap N_v|}{|N_u| + |N_v|}$$

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%

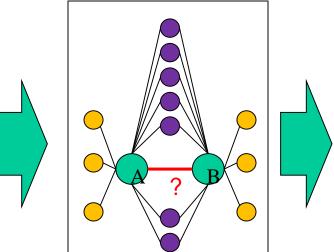


 CD-distance and its variations correlate very well with functional homogeneity and localization coherence

# 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 purely on tological info, w/o needing location etc info on proteins



**Deduction / induction** 

# IDENTIFYING HOMOLOGOUS PROTEINS

### A protein is a ...

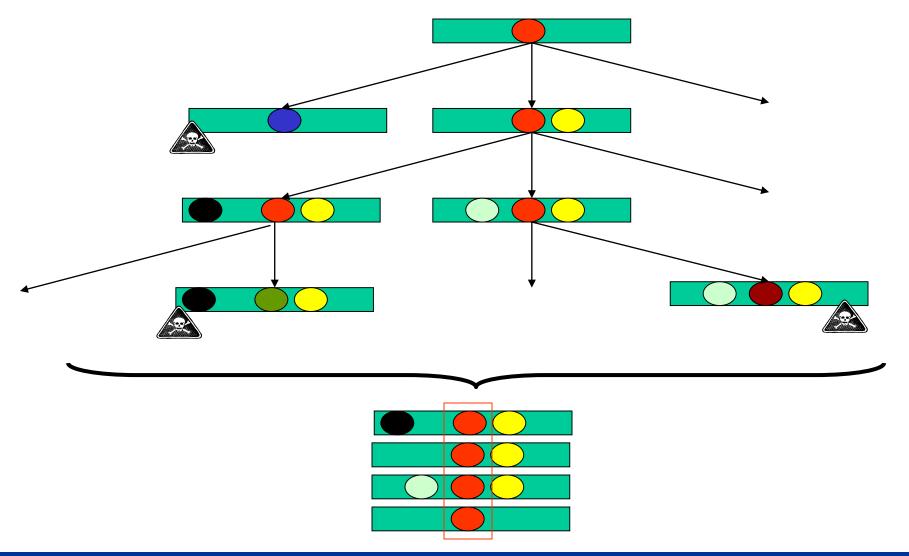


- A protein is a large complex molecule made up of one or more chains of amino acids
- Proteins perform a wide variety of activities in the cell



### 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 average difference between the 2<sup>nd</sup> generation of a

What is the average difference between the 2<sup>nd</sup> 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

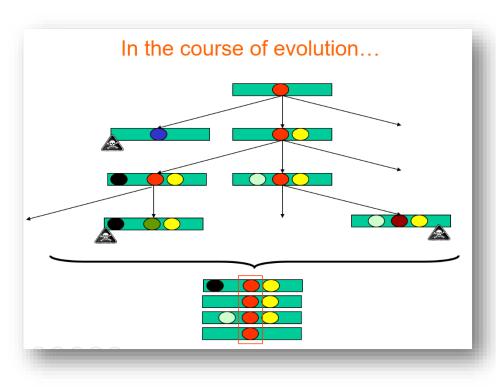
Each 2<sup>nd</sup>-gen of a differs from a by 2 residues and two 2<sup>nd</sup>-gen of a differ by at most 4 residues

a and b differ in 9 residues

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



SPSTNRKYPPLPVDKLEEEINRRMADDNKLFREEFNALPACPIQATCEAASKEENKEKNR
YVNILPYDHSRVHLTPVEGVPDSDYINASFINGYQEKNKFIAAQGPKEETVNDFWRMIWE
QNTATIVMVTNLKERKECKCAQYWPDQGCWTYGNVRVSVEDVTVLVDYTVRKFCIQQVGD
VTNRKPQRLITQFHFTSWPDFGVPFTPIGMLKFLKKVKACNPQYAGAIVVHCSAGVGRTG
TFVVIDAMLDMMHSERKVDVYGFVSRIRAQRCQMVQTDMQYVFIYQALLEHYLYGDTELE
VT

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



### Time for Exercise #4

How can we guess the function of a protein?

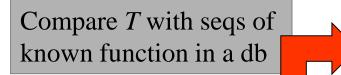
## Abductive reasoning

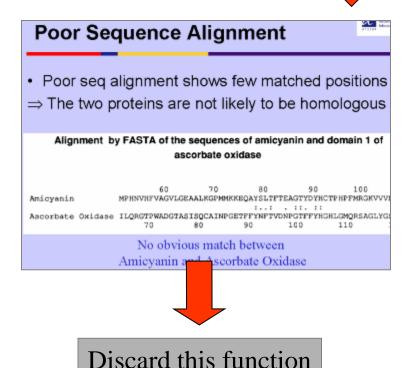


- Law: Two proteins inheriting their function from a common ancestor have very similar amino acid sequences
- Observation: Proteins X and Y are very similar in their sequence
- Abduction: Proteins X and Y are descended from the same ancestor and inherit their function from this ancestor
- ⇒ Proteins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

## Guilt by association







as a candidate

# Good Sequence Alignment

- Good alignment usually has clusters of extensive matched positions
- ⇒ The two proteins are likely to be homologous

Assign to *T* same function as homologs

Confirm with suitable wet experiments

# Earliest research in seq comparison National University of Singapore

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
```



**Violation of invariant** 

# MAKING COMPUTER SYSTEMS MORE SECURE

# COMPUTERWORLD An IDG

# 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

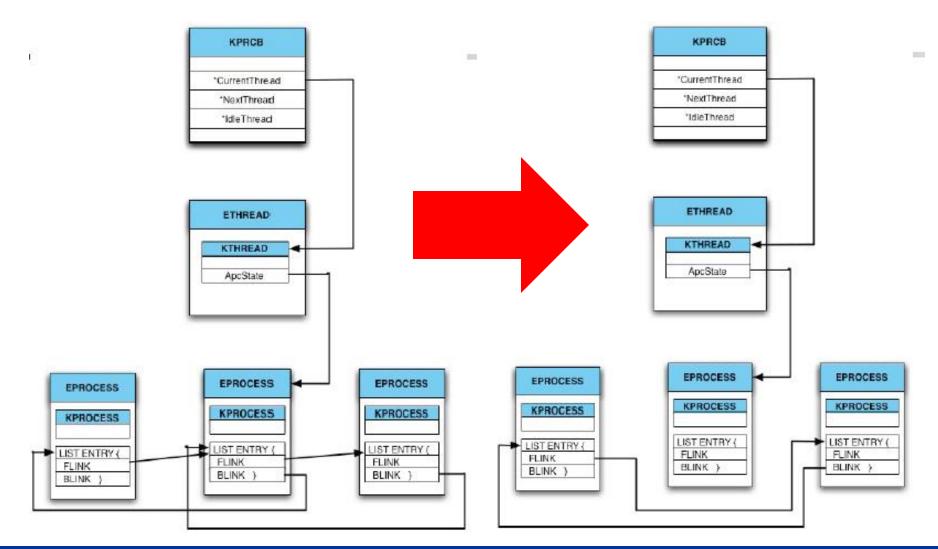
#### **Modern rootkits**

- 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 systemms 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

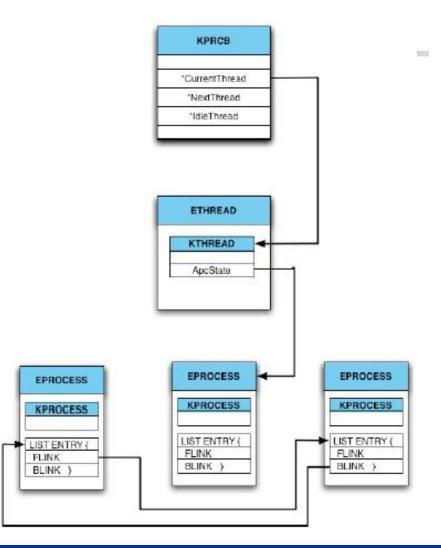
#### **DKOM Example**



 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!



**Violation of invariant** 

# IMPROVING DATABASE DESIGN

#### Relational data model







Movies

year

title

Studios

addr

Stars

addr

filmType

length

Name	Address
Carrie Fisher	Hollywood
Mark Hamill	Brentwood
Harrison Ford	<b>Beverly Hills</b>

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?

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?

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?

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 (A₁, ..., An → B₁, ..., Bm)
  - 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<sub>1</sub>, ..., A<sub>n</sub>  $\rightarrow$  B<sub>1</sub>, ..., B<sub>m</sub>) is trivial if a B<sub>i</sub> is an A<sub>j</sub>
  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<sub>1</sub>, ..., A<sub>n</sub>} 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  $(A_1, ..., A_n \rightarrow B_1, ..., B_m)$  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  $(A_1, ..., A_n \rightarrow B_1, ..., B_m)$  to decompose  $R(A_1, ..., A_n, B_1, ..., B_m, C_1, ..., C_h)$  into 2 tables

- $R_1(A_1, ..., A_n, B_1, ..., B_m)$
- $R_2(A_1, ..., A_n, C_1, ..., C_h)$

	No update anomaly No redundant info					
Т	ïtle	Year	Length	Fil 1 Type	Studio	
S	tar Wars	1997	124	Color	Fox	
M	lighty Ducks	1991	104	Color	Disney	

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



Induction / fixing violated invariants

# INFERRING KEY MUTATIONS: WHY SOME PTP IS INEFFICIENT

#### Protein tyrosine phosphatase



Sequence from a typical PTP

>gi|00000|PTPA-D2

EEEFKKLTSIKIQNDKMRTGNLPANMKKNRVLQIIPYEFNRVIIPVKRGEENTDYVNASF IDGYRQKDSYIASQGPLLHTIEDFWRMIWEWKSCSIVMLTELEERGQEKCAQYWPSDGLV SYGDITVELKKEEECESYTVRDLLVTNTRENKSRQIRQFHFHGWPEVGIPSDGKGMISII AAVQKQQQQSGNHPITVHCSAGAGRTGTFCALSTVLERVKAEGILDVFQTVKSLRLQRPH MVOTLEOYEFCYKVVOEYIDAFSDYANFK

- 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

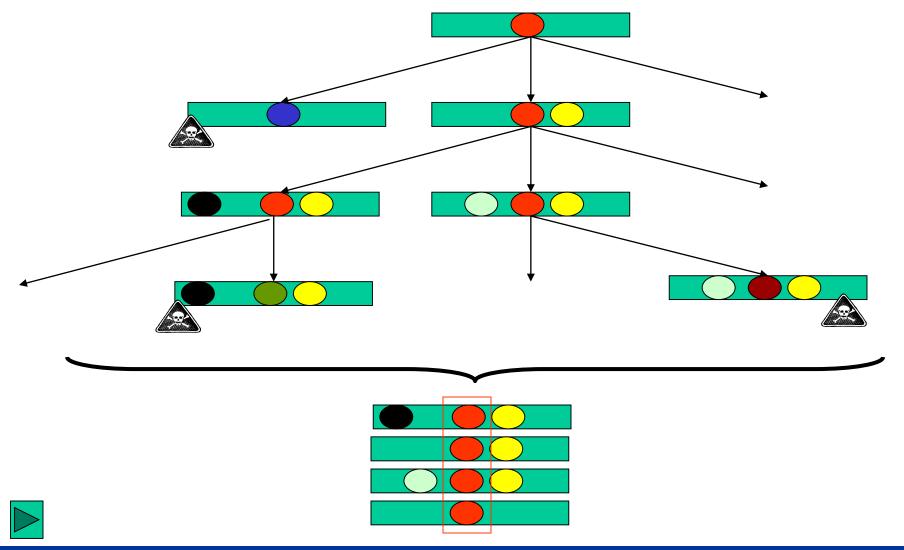
>gi|00000|PTPA-D2

EEEFKKLTSIKIQNDKMRTGNLPANMKKNRVLQIIPYEFNRVIIPVKRGEENTDYVNASF IDGYRQKDSYIASQGPLLHTIEDFWRMIWEWKSCSIVMLTELEERGQEKCAQYWPSDGLV SYGDITVELKKEEECESYTVRDLLVTNTRENKSRQIRQFHFHGWPEVGIPSDGKGMISII AAVQKQQQQSGNHPITVHCSAGAGRTGTFCALSTVLERVKAEGILDVFQTVKSLRLQRPH MVQTLEQYEFCYKVVQEYIDAFSDYANFK

Some PTPs are much less efficient than others

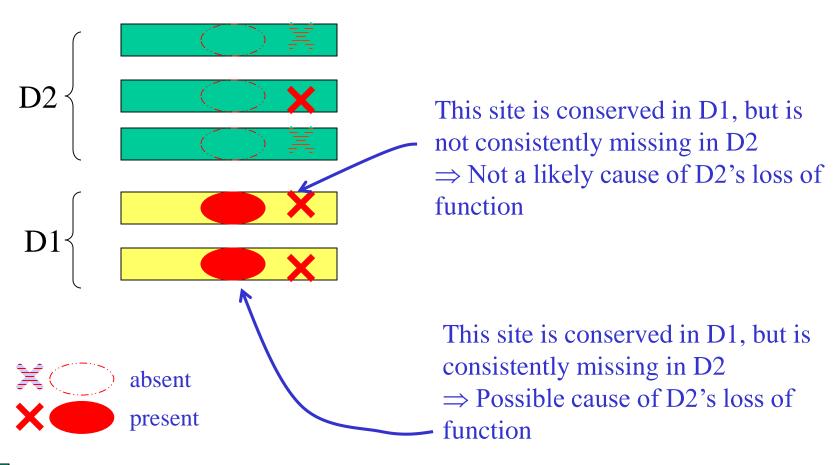
How do you figure out which mutations cause the loss of efficiency?

# Some sites are impt for PTP functional University of Singapore



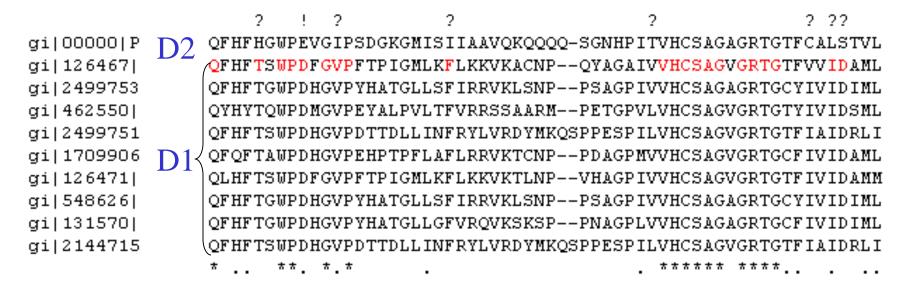
### Reasoning based on an invariant.







### Key mutation site: PTP D1 vs D2 National University of Singapore



- 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 expensional University of Singapore

- Wet experiments confirm the predictions
  - Mutate D → E in D1
    - i.e., check if D → E can cause efficiency loss
  - Mutate E  $\rightarrow$  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



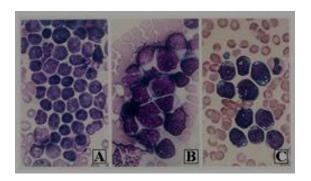
- Induction/hypothesis: A site that is critical for PTP efficiency is present in all efficient PTPs and absent in all inefficient PTPs
- Observation: A site X is present in all efficient PTPs and absent in all inefficient PTPs
- Abduction: Site X is critical for PTP efficiency

### Bioengineering more efficient PTP National University of Singapore

- Replace an inefficient PTP in the organism by an efficient version
  - Mutate  $E \rightarrow D$  in D2

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



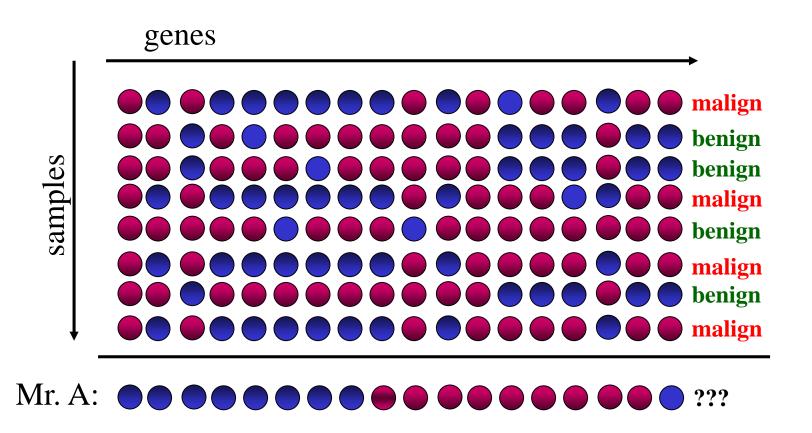


Induction

# DIAGNOSING PEDIATRIC LEUKEMIAS

#### Some Patient Samples

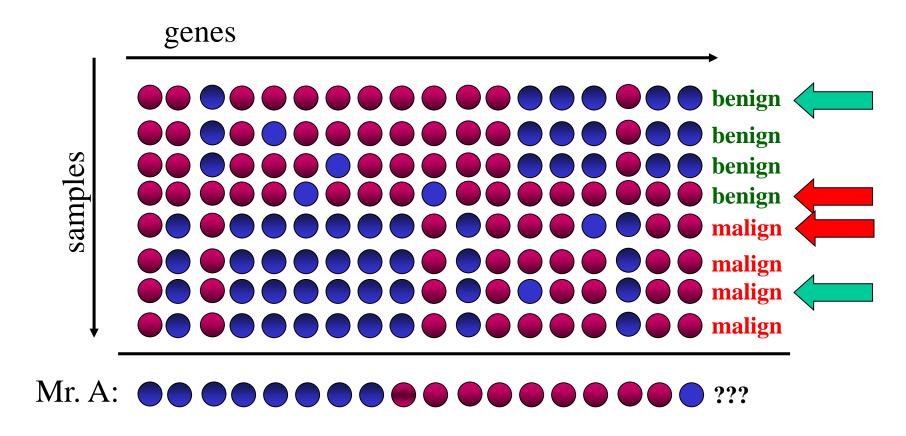




Does Mr. A have cancer?

#### Let's rearrange the rows...

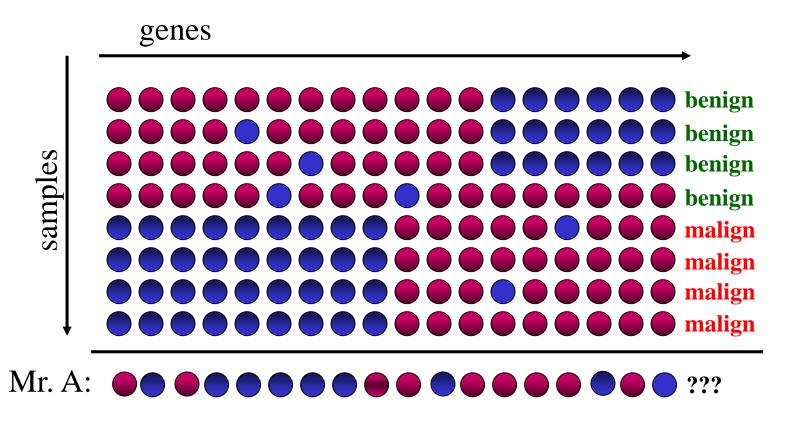




Does Mr. A have cancer?

#### and the columns too...





 Induction/hypothesis: Benign (malignant) tumour has lots of red (blue) genes on the left and blue (red) genes on the right

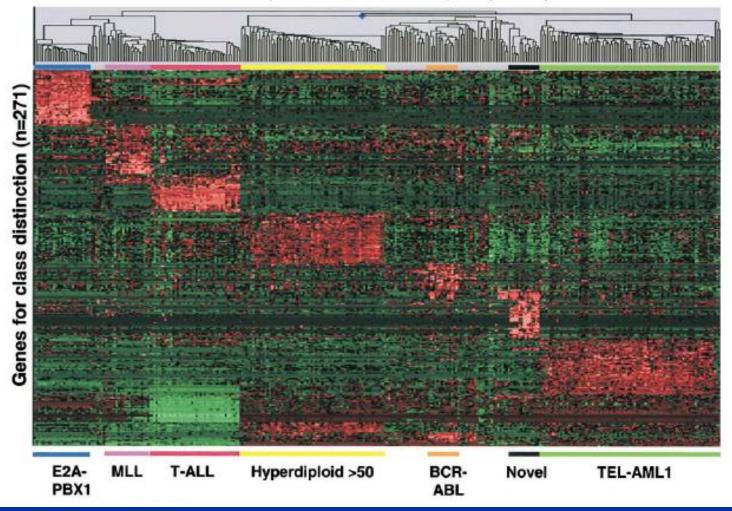
#### The triumph of logic



- Induction/hypothesis: Benign (malignant) tumour has lots of red (blue) genes on the left and blue (red) genes on the right
- Observation: Mr A's tumour has lots of blue genes on the left and red genes on the right
- Abduction: Mr A's tumour is malignant

# Invariant profile of leukemia subtype NUS National University of Singapore







What have we just seen?

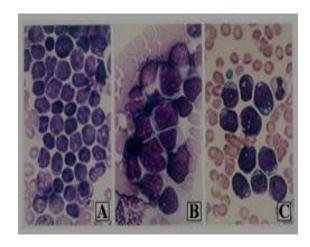
Guilt by association of invariants

#### Childhood ALL



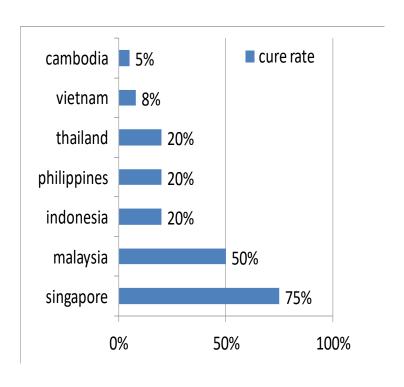
- 6 Major subtypes: T-ALL, E2A-PBX, TEL-AML, BCR-ABL, MLL genome rearrangements, Hyperdiploid>50
- Diff subtypes respond differently to same Tx
- Over-intensive Tx
  - Development of secondary cancers
  - Reduction of IQ
- Under-intensiveTx
  - Relapse: suffer deterioration after a period of improvement.

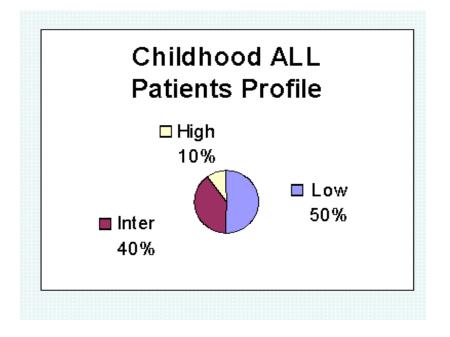
The subtypes look similar



- Conventional diagnosis
  - Immunophenotyping
  - Cytogenetics
  - Molecular diagnostics
- Unavailable in most ASEAN countries

### The situation in ASEAN, circa 200 National University of Singapore





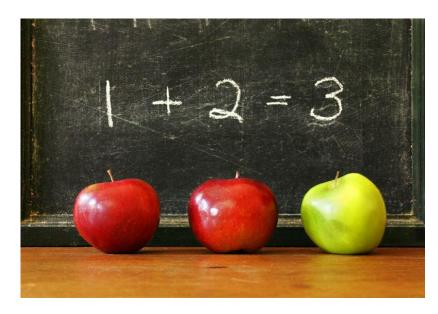
### Exploit invariant gene expr profile

- Low-intensity treatment applied to 50% of patients
- Intermediate-intensity treatment to 40% of patients
- High-intensity treatment to 10% of patients
- ⇒ Reduced side effects
- ⇒ Reduced relapse
- $\Rightarrow$  75-80% cure rates

- US\$36m (US\$36k \* 2000 \* 50%) for low intensity
- US\$48m (US\$60k \* 2000 \* 40%) for intermediate intensity
- US\$14.4m (US\$72k \* 2000 \* 10%) for high intensity
- Total US\$98.4m/yr
- ⇒ Save US\$51.6m/yr, compared to applying intermediate-intensity treatment to everyone

Yeoh et al, Cancer Cell 2002





**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

Computational biology is no more about programming than Biology is about Petri dish & test tube