SOC Summer School 2017

A logical introduction to computational biology

Wong Limsoon

National University of Singapore

About Limsoon

Position

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**

About Part I

I will describe some problem-solving principles that are common to multiple types of problems, even in different disciplines (I will illustrate using different areas in computer science, medicine, biology, and biotechnology)

These principles are simple logical ways to exploit fundamental properties of each problem domain, highlighting the value of both logical thought and domain knowledge, and bringing out the sometimes creative way of applying the former to the latter in the context of each problem being solved

Part I

Some universal scientific problem-solving paradigms based on logical analysis and exploitation of invariants

NUS
National University of Singapore

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/Chemistry is no more about Petri dish & test tube than Computer Science is about programming

Three types of reasoning

- **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

Triumph of logic

- **Logical reasoning on invariant**
  - Deduction
    - **Bet on the beans**
    - **De-noise PPI networks**
  - Abduction
    - **Identify homologs**
  - Induction
    - **Infer key mutations**
    - **Diagnose pediatric leukemias**
    - **Identify homologs**
- **Fixing violation of invariant**
  - **Make computers more secure**
  - **Improve database design**
  - **Infer key mutations**
- **Guilt by association of invariant**
  - **Predict protein function**
  - **Diagnose pediatric leukemias**

In the following examples you will see the intertwining of logic and invariants in scientific problem solving

Deduction

WHAT IS AN INVARIANT

- **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?

Bet on the last green 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 odd, …**
- **Start with y=2n+1**
- **y=2n+1 → y=2n-1**
- **y=2n+1 → y=2n+1**
- **y=2n+1 → y=2n+1**
- **y remains odd**

⇒ **Last bean must be green!**

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!**

Bet on color of the last bean … and 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**

$\Rightarrow$ **Parity of green beans is invariant**

$\Rightarrow$ **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
- 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!

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

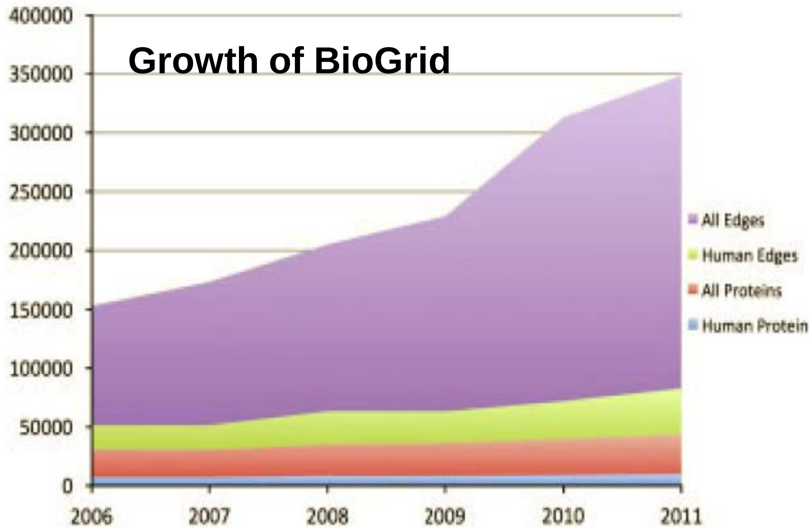
Deduction

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**

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

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?

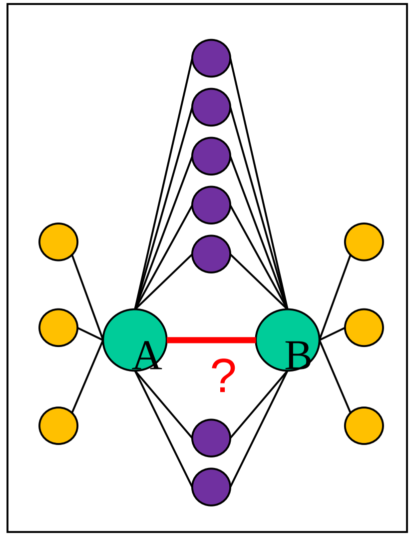
Liu et al. Complex discovery from weighted PPI networks. *Bioinformatics*, 25(15):1891-1897, 2009

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?

The triumph of logic

Two proteins should be in same place to interact



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

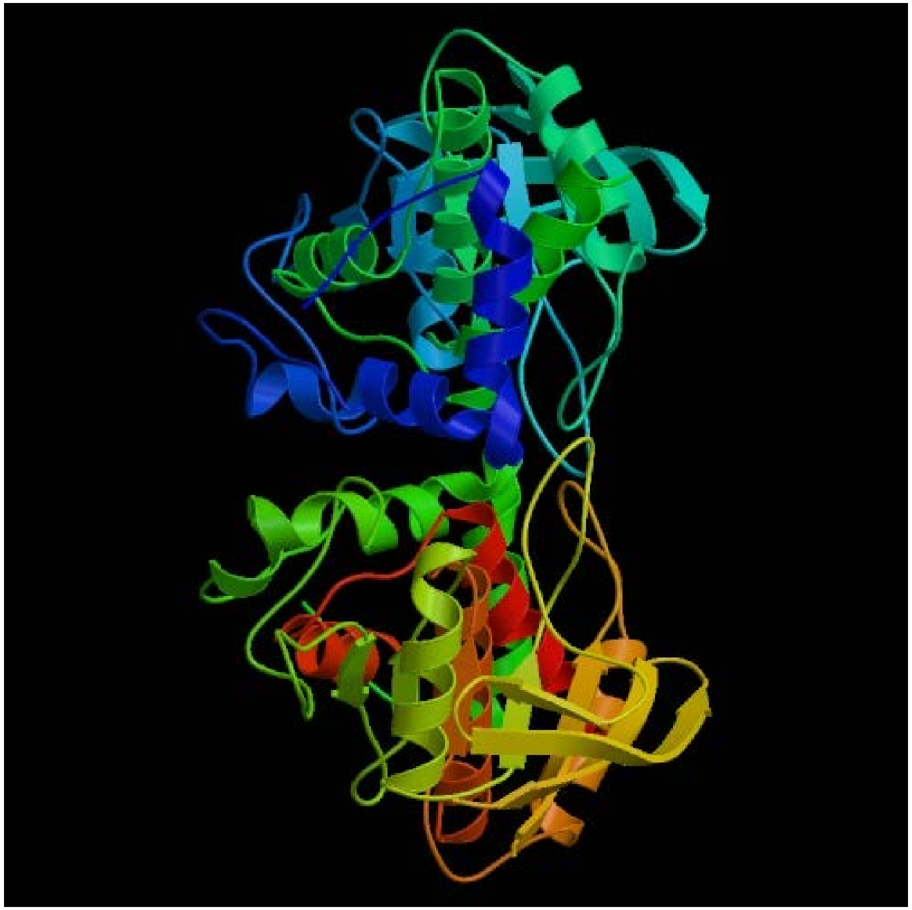
Impact:
PPI networks can be cleansed based purely on tological info, w/o needing location etc info on proteins

Induction / deduction

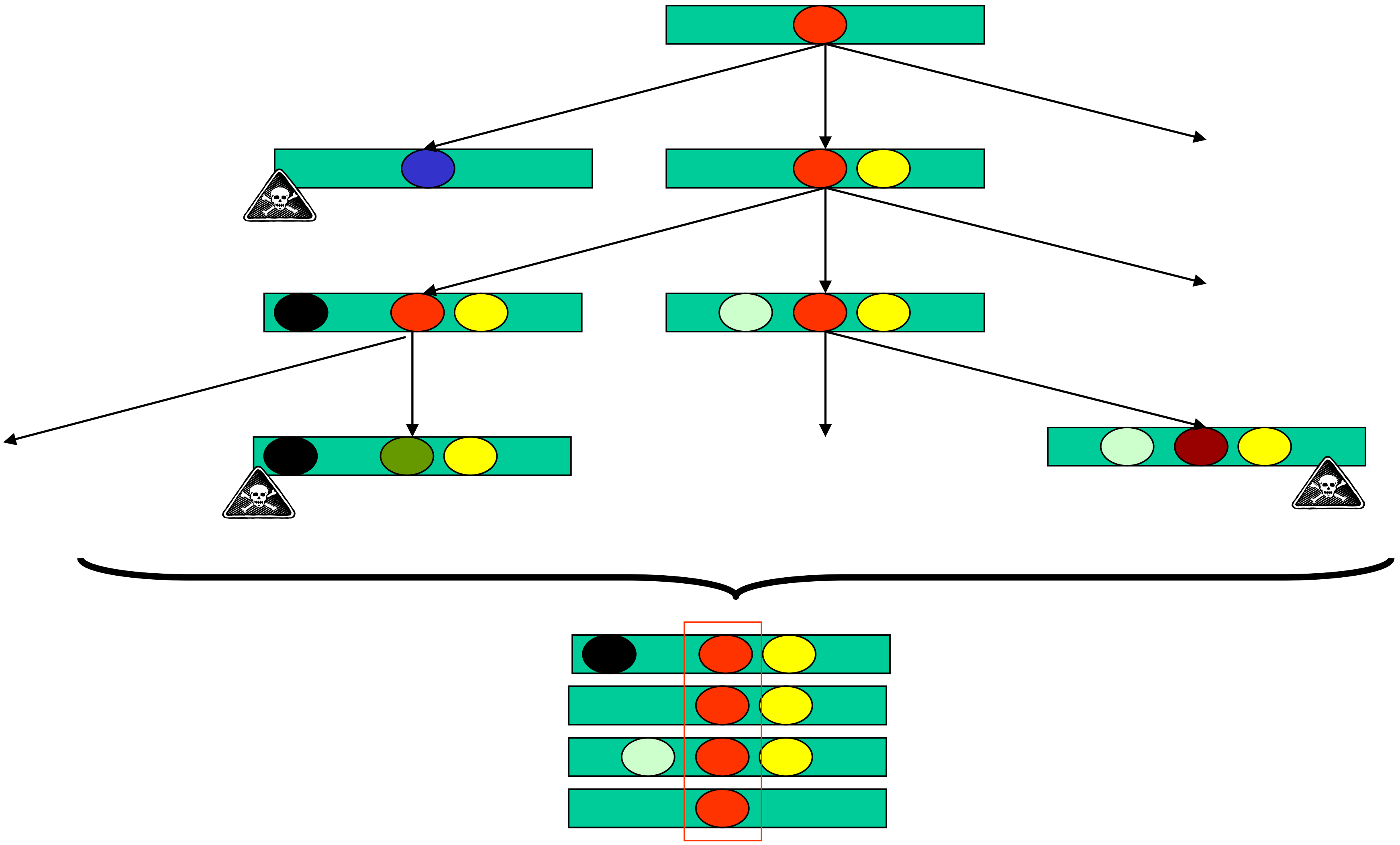
IDENTIFYING HOMOLOGOUS PROTEINS

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

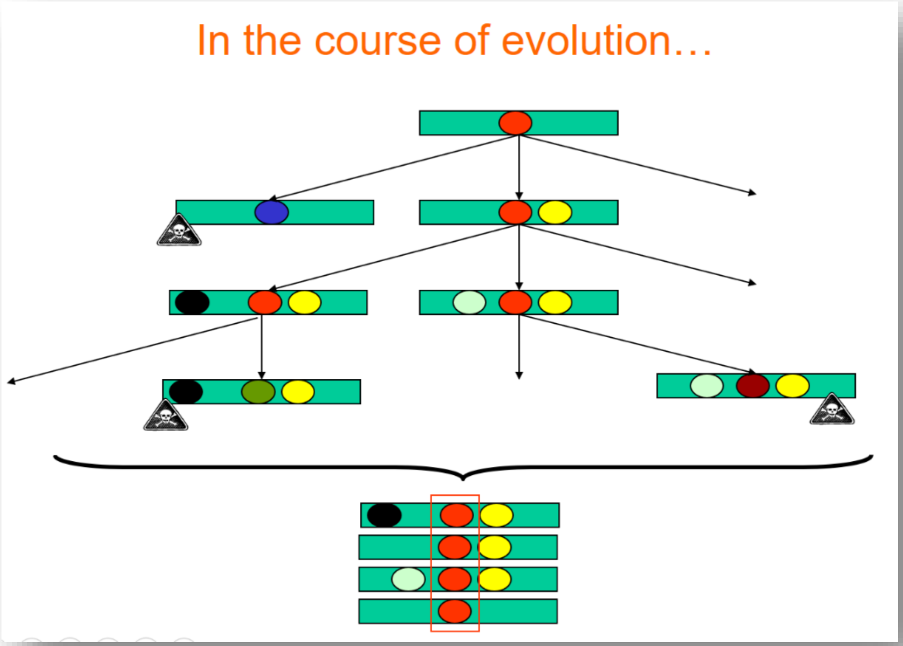
Let a = AFPHQHRVP
Let b = PQVYNIMKE

Suppose each generation differs from the previous by 1 residue

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

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

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?

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
```

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……

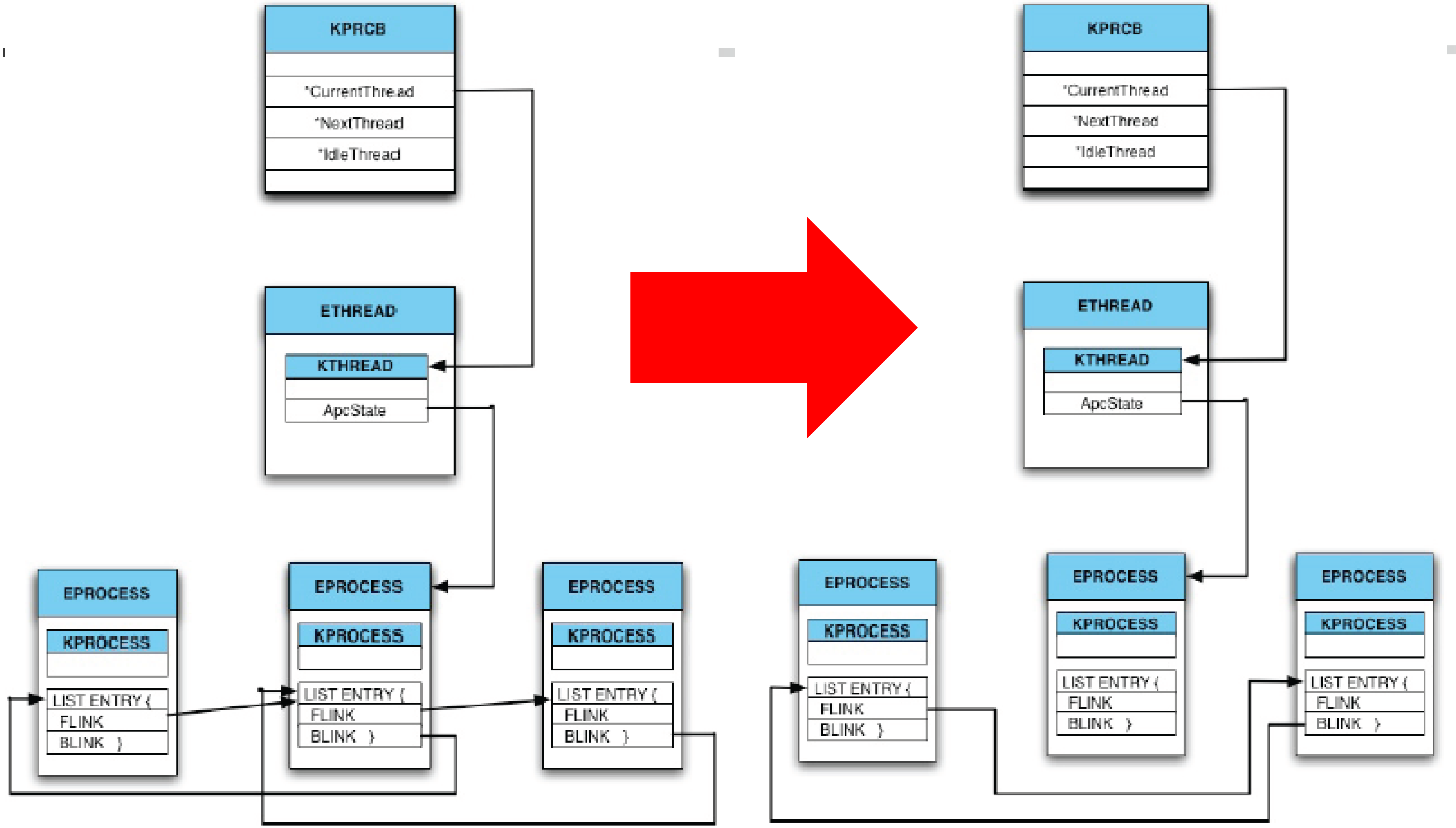
Credit: Bill Arbaugh

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

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

- **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!

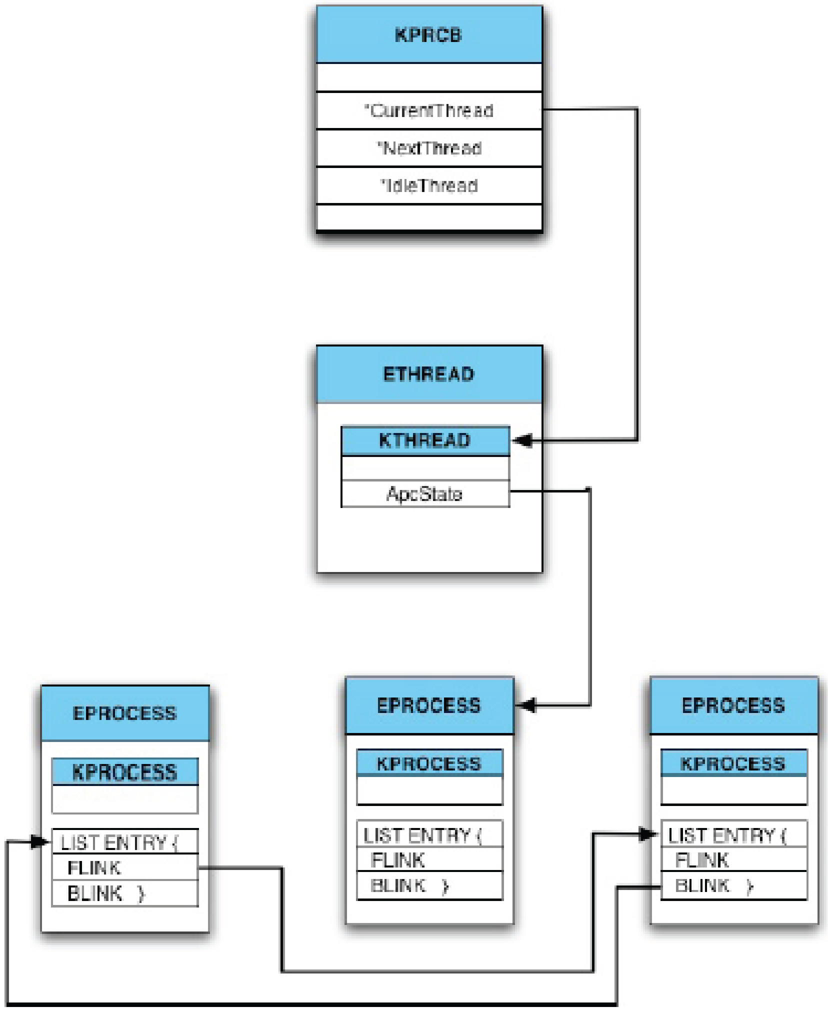
Credit: Bill Arbaugh

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!**

Impact

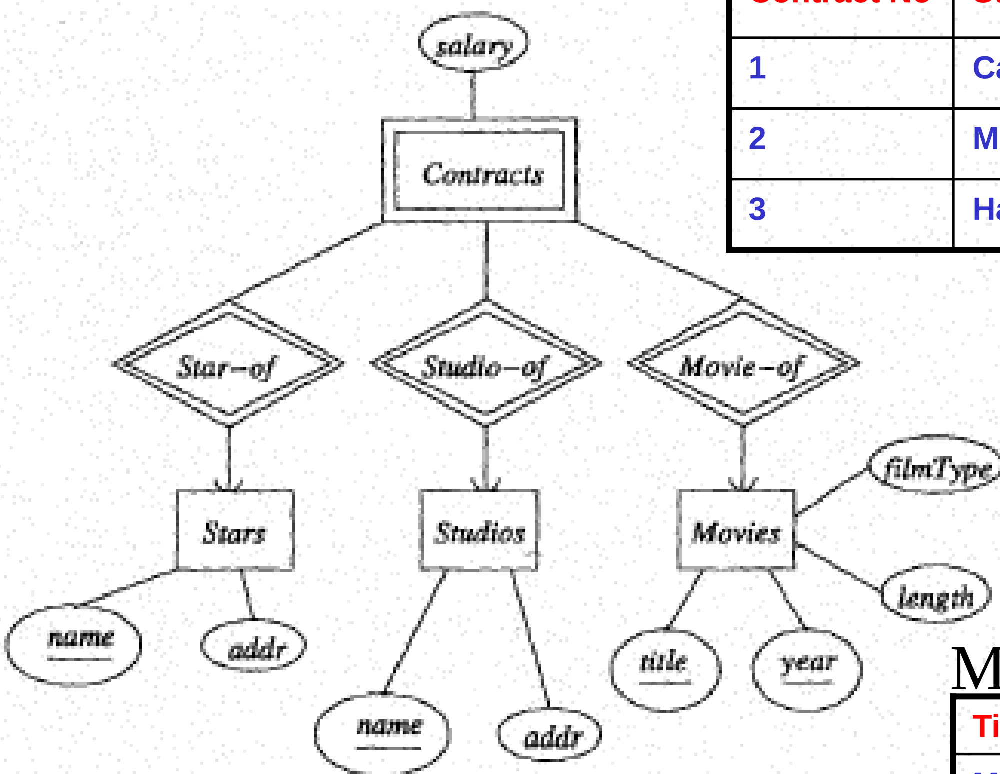
- **2008: Komoku (kHIVE) acquired by Microsoft**
- **2009: Put into MS Security Essentials (~4m hosts)**
- **2010: Put into Windows Update (~500m hosts)**

"There is no other field out there where you can get right out of university and define substantial aspects of a product that is going to go out and over 100 million people are going to use it". ---Bill Gate

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

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 ($A_1, \ldots, A_n \rightarrow B_1, \ldots, B_m$)**
  - If two rows of a table R agree on attributes $A_1, \ldots, A_n$, then they must also agree on attributes $B_1, \ldots, B_m$

  $\Rightarrow$ Values of B's depend on values of A's
- **FD ($A_1, \ldots, A_n \rightarrow B_1, \ldots, B_m$) 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 $\{A_1, \ldots, A_n\}$ 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 ($A_1, \ldots, A_n \rightarrow B_1, \ldots, B_m$) to decompose $R(A_1, \ldots, A_n, B_1, \ldots, B_m, C_1, \ldots, C_h)$ into 2 tables**
  - $R_1(A_1, \ldots, A_n, B_1, \ldots, B_m)$
  - $R_2(A_1, \ldots, A_n, C_1, \ldots, C_h)$

No update anomaly

No redundant info

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

No deletion anomaly

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

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

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

Impact

ORACLE CORPORATION

Q3 FISCAL 2010 FINANCIAL RESULTS
CONDENSED CONSOLIDATED STATEMENTS OF OPERATIONS
($ in millions, except per share data)

	Three Months Ended February 28,				% Increase (Decrease) in US $
	2010	% of Revenues	2009	% of Revenues	
REVENUES					
New software licenses	$ 1,718	27%	$ 1,516	28%	13%
Software license updates and product support	3,297	51%	2,917	53%	13%
Software Revenues	5,015	78%	4,433	81%	13%
Hardware systems products	273	4%	-	0%	*
Hardware systems support	185	3%	-	0%	*
Hardware Systems Revenues	458	7%	-	0%	*
Services	931	15%	1,020	19%	(9%)
Total Revenues	6,404	100%	5,453	100%	17%

Induction / fixing violated invariants

INFERRING KEY MUTATIONS: WHY SOME PTP IS INEFFICIENT

Protein tyrosine phosphatase

NUS National University of Singapore

Sequence from a typical PTP

```
>gi|00000|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

```
>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?

Key mutation site: PTP D1 vs D2

```
                    ?   !  ?         ?                 ?             ? ??
gi|00000|P  D2  QFHFHGWPEVGIPSDGKGMISIIAAVQKQQQQ-SGNHPITVHCSAGAGRTGTFCALSTVL
gi|126467|      QFHFTSWPDFGVPFTPIGMLKFLKKVKACNP--QYAGAIVVHCSAGVGRTGTFVVIDAML
gi|2499753      QFHFTGWPDHGVPYHATGLLSFIRRVKLSNP--PSAGPIVVHCSAGAGRTGCYIVIDIML
gi|462550|      QYHYTQWPDMGVPEYALPVLTFVRRSSAARM--PETGPVLVHCSAGVGRTGTYIVIDSML
gi|2499751      QFHFTSWPDHGVPDTTDLLINFRYLVRDYMKQSPPESPILVHCSAGVGRTGTFIAIDRLI
gi|1709906  D1  QFQFTAWPDHGVPEHPTPFLAFLRRVKTCNP--PDAGPMVVHCSAGVGRTGCFIVIDAML
gi|126471|      QLHFTSWPDFGVPFTPIGMLKFLKKVKTLNP--VHAGPIVVHCSAGVGRTGTFIVIDAMM
gi|548626|      QFHFTGWPDHGVPYHATGLLSFIRRVKLSNP--PSAGPIVVHCSAGAGRTGCYIVIDIML
gi|131570|      QFHFTGWPDHGVPYHATGLLGFVRQVKSKSP--PNAGPLVVHCSAGAGRTGCFIVIDIML
gi|2144715      QFHFTSWPDHGVPDTTDLLINFRYLVRDYMKQSPPESPILVHCSAGVGRTGTFIAIDRLI
                * ..  **. *.*      .                  . ****** ****..  .  ..
```

- **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.

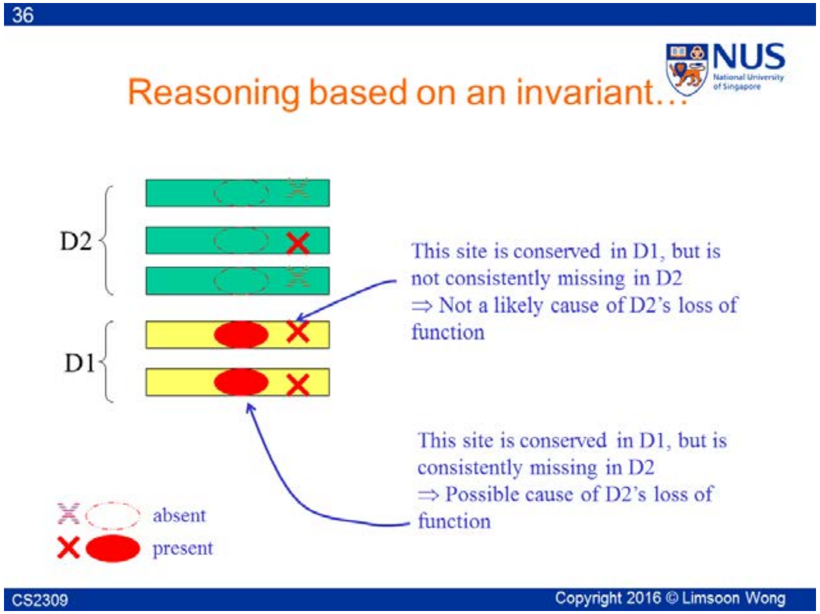
Confirmation by mutagenesis expt

- **Wet expts confirmed 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!

Time for Exercise #7

Is this abductive, deductive, or inductive reasoning?



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**

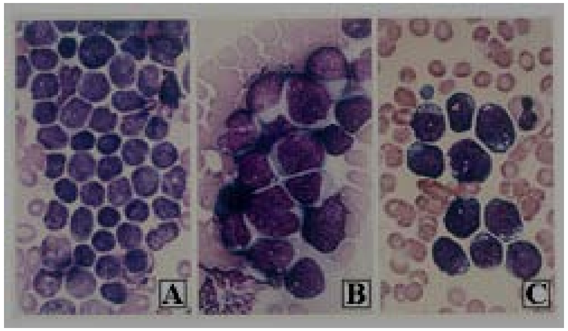
- **Abduction: Site X is critical for PTP efficiency**

Bioengineering more efficient PTPs

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

- **What have we just seen?**

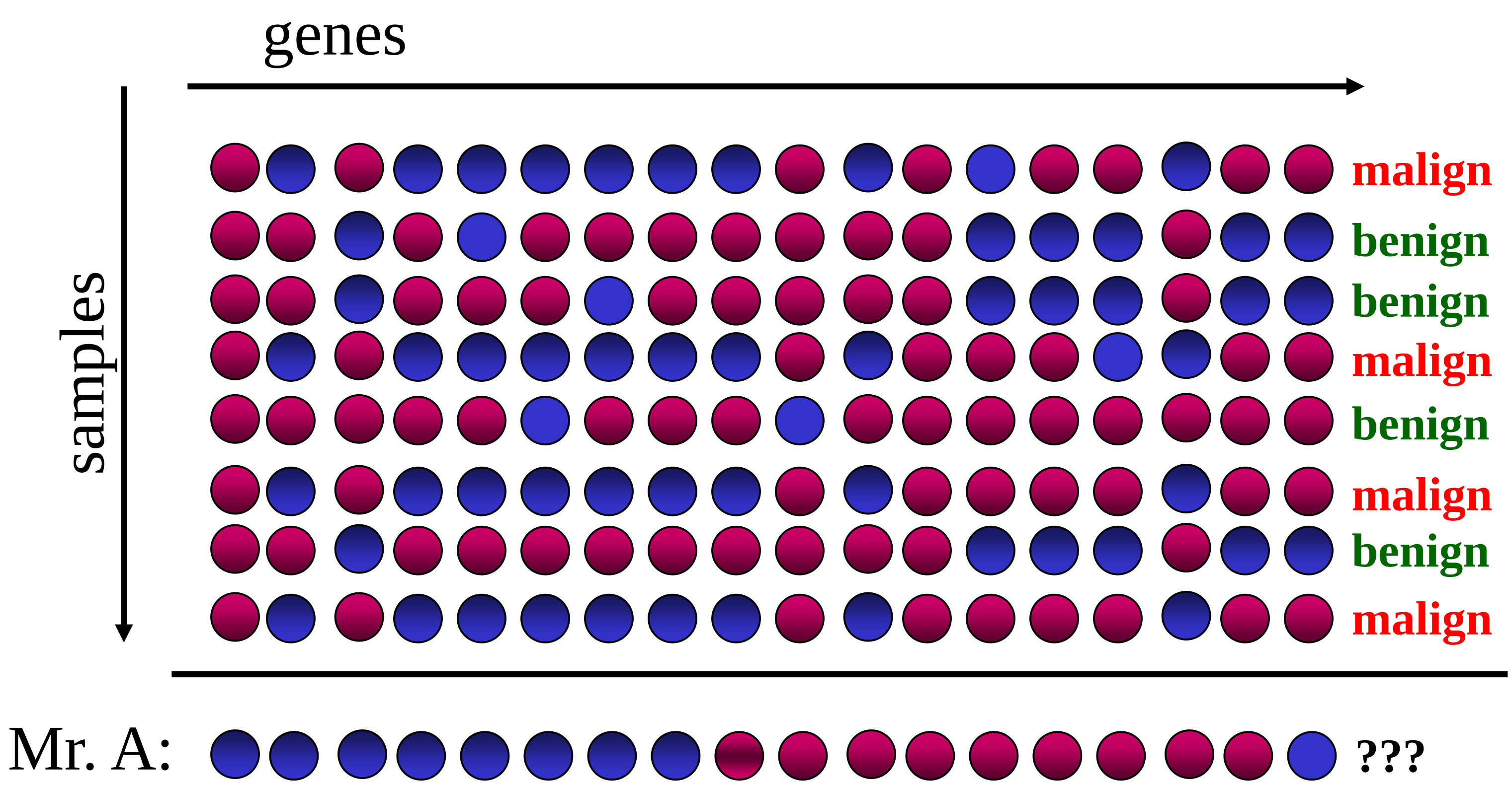
- **Create a more efficient PTP by fixing a violated invariant!**



Induction

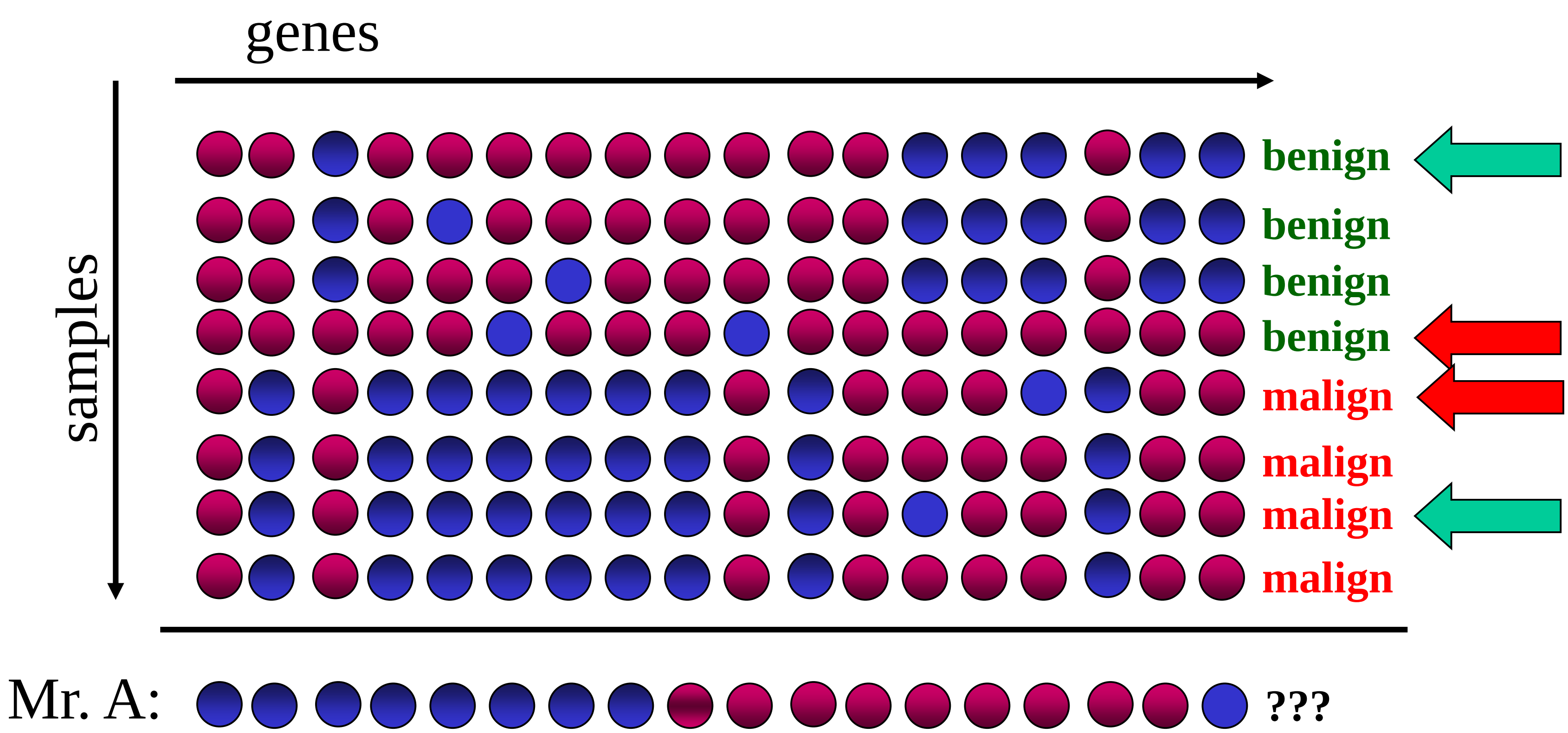
DIAGNOSING PEDIATRIC LEUKEMIAS

Some Patient Samples



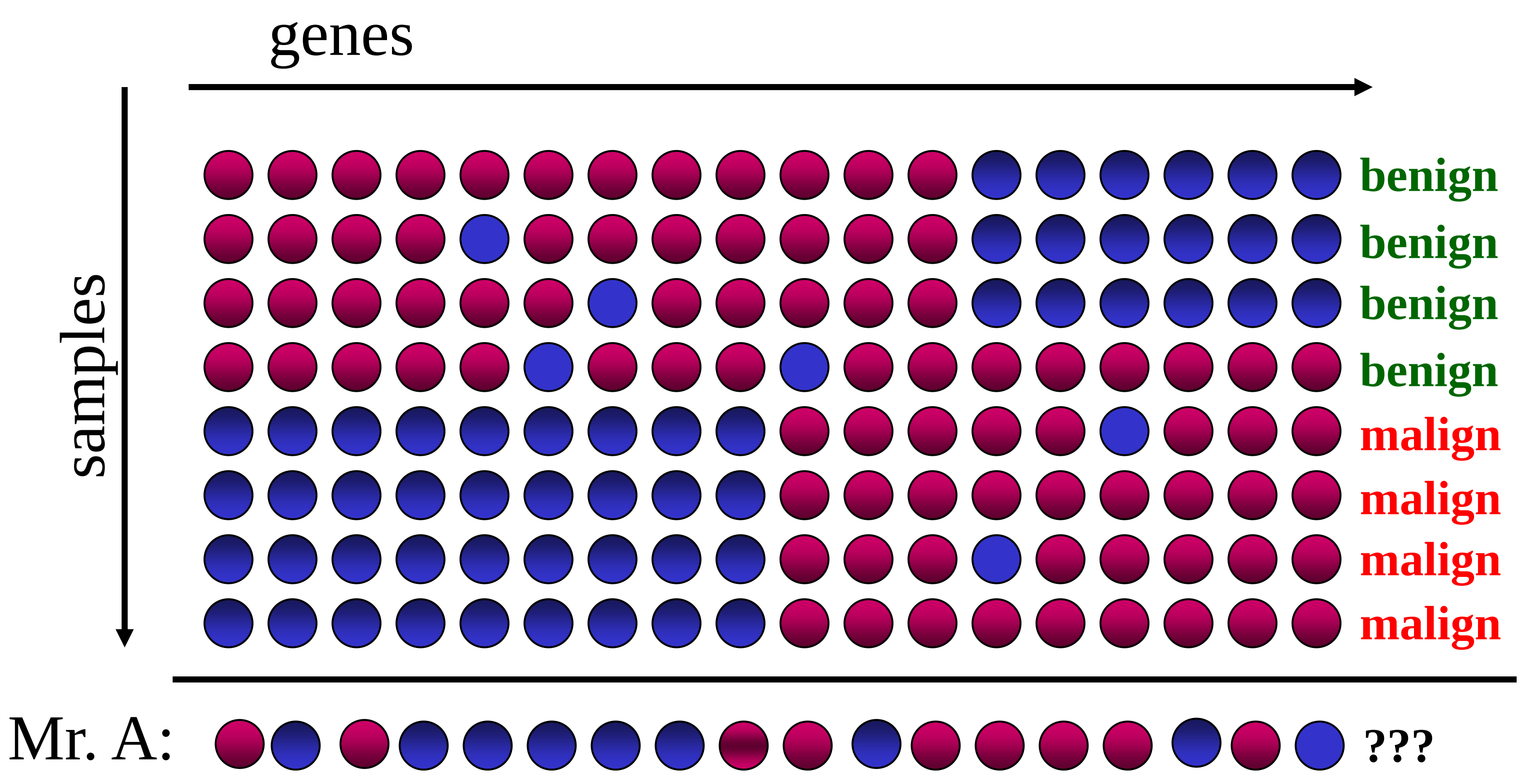
- **Does Mr. A have cancer?**

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**

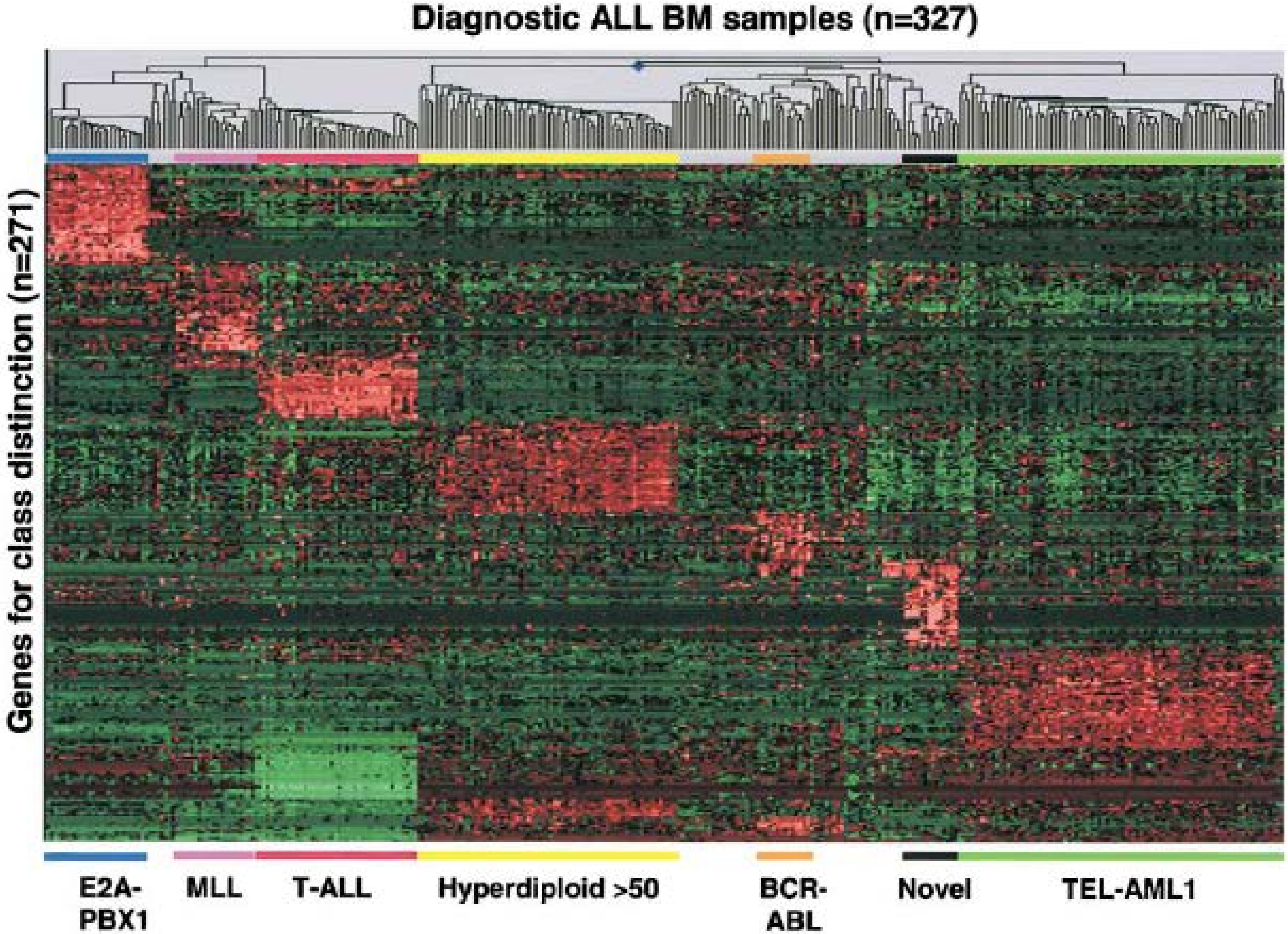
The triumph of logic

- **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 subtypes

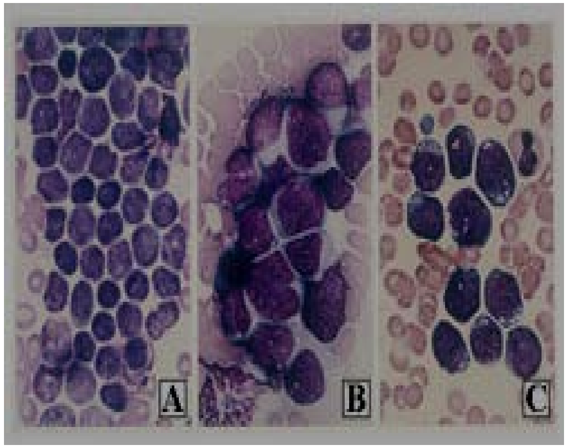


- **What have we just seen?**

- **Guilt by association of invariants**

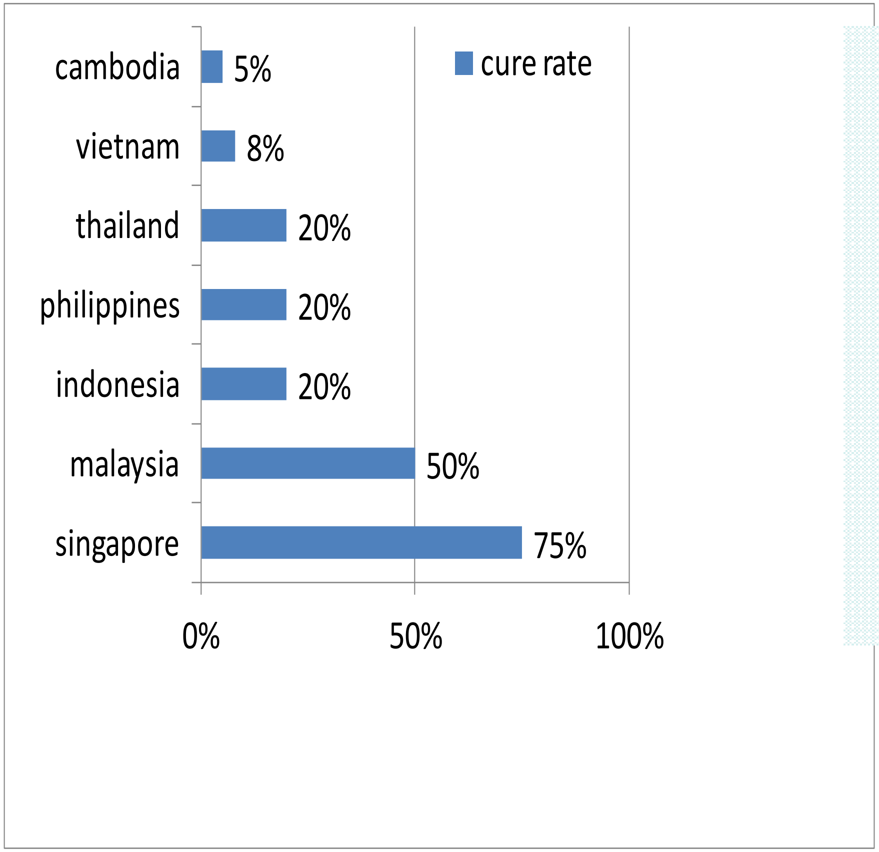
Childhood ALL

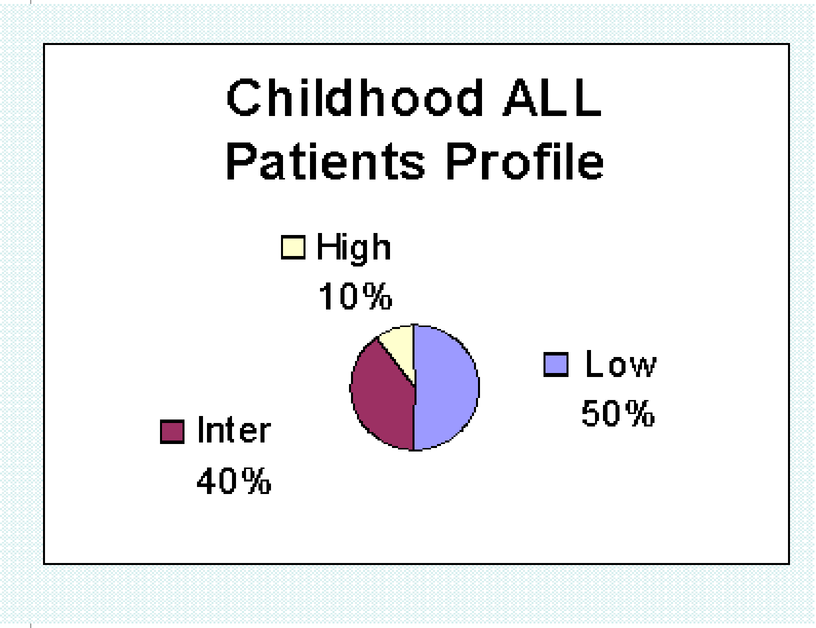
- **6 Major subtypes: T-ALL, E2A-PBX, TEL-AML, BCR-ABL, MLL genome rearrangements, Hyperdiploid>50**
- **Diff subtypes respond differently to same Tx**
- **Over-intensive Tx**
  - Development of secondary cancers
  - Reduction of IQ
- **Under-intensiveTx**
  - Relapse: suffer deterioration after a period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



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

The situation in ASEAN, circa 2000





Exploit invariant gene expr profiles

- Low-intensity treatment applied to 50% of patients
- Intermediate-intensity treatment to 40% of patients
- High-intensity treatment to 10% of patients

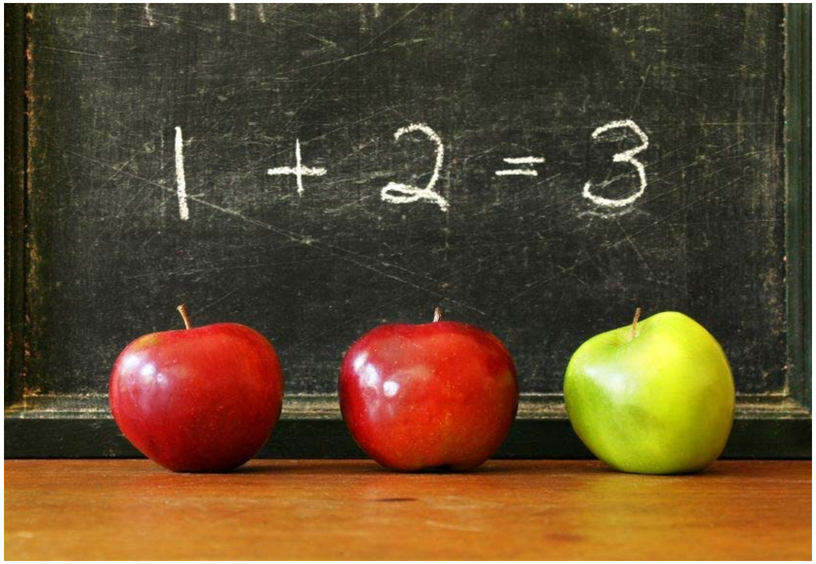
⇒ Reduced side effects
⇒ Reduced relapse
⇒ 75-80% cure rates

- 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**
- **Paradigms of problem solving**
  - Problem solving by logical reasoning on invariants
  - Problem solving by rectifying/monitoring violation of invariants
  - Guilt by association of invariants

Computer Science is no more about programming than Biology/Chemistry is about Petri dish & test tube