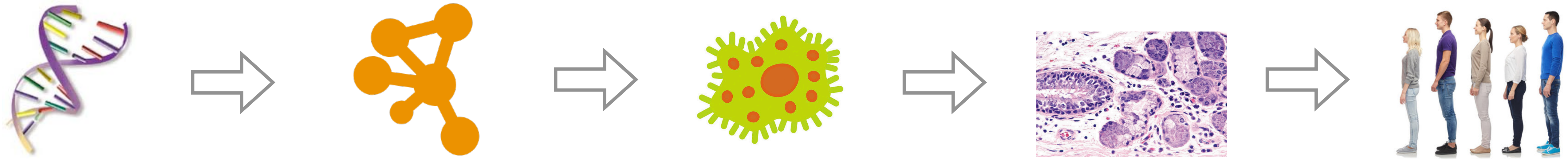
Review of CSE427

Sheng Wang

CSE427: Computational methods for biology at different scales



Gene
(1 nm)

Protein complexes (function)
(10-100nm)

Cell
(1–10 μm)

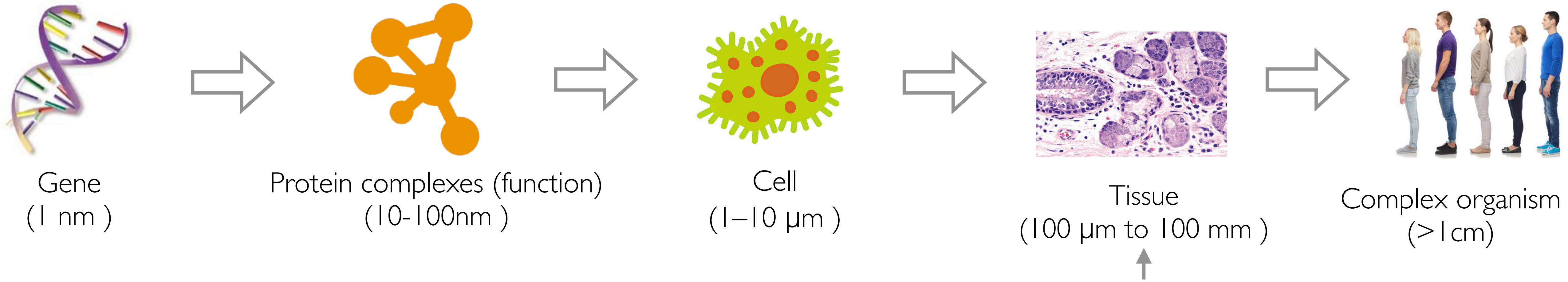
Tissue
(100 μm to 100 mm)

Complex organism
(>1cm)

A rich hierarchy of biological subsystems at multiple scales: genotypic variations in nucleotides (1 nm scale) -> proteins (1–10 nm) -> protein complexes (10–100 nm), cellular processes (100 nm) -> phenotypic behaviors of cells (1–10 μm), tissues (100 μm to 100 mm), -> complex organisms (>1 m).

source: Yu, Michael Ku, et al. "Translation of genotype to phenotype by a hierarchy of cell subsystems." *Cell systems* 2.2 (2016): 77-88.

How a computer scientist study comp bio? Understand the input and output first



Gene
(1 nm)

Protein complexes (function)
(10-100nm)

Cell
(1–10 μm)

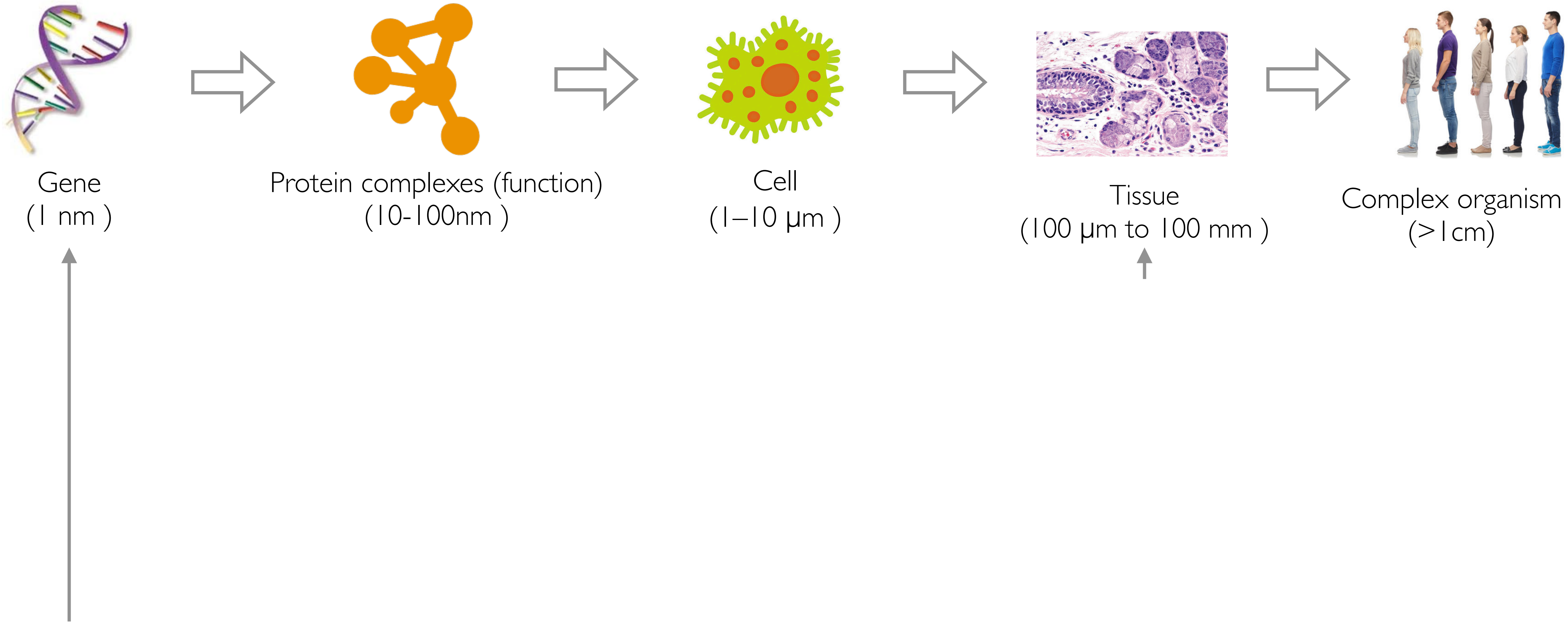
Tissue
(100 μm to 100 mm)

Complex organism
(>1cm)

Biologists: which input should I use for this problem? Gene expression? Tissue images?

Computer scientists: Given the input we have, which method should we use to solve this problem?

Data structure for each scale: protein



Gene
(1 nm)

Protein complexes (function)
(10-100nm)

Cell
(1–10 μm)

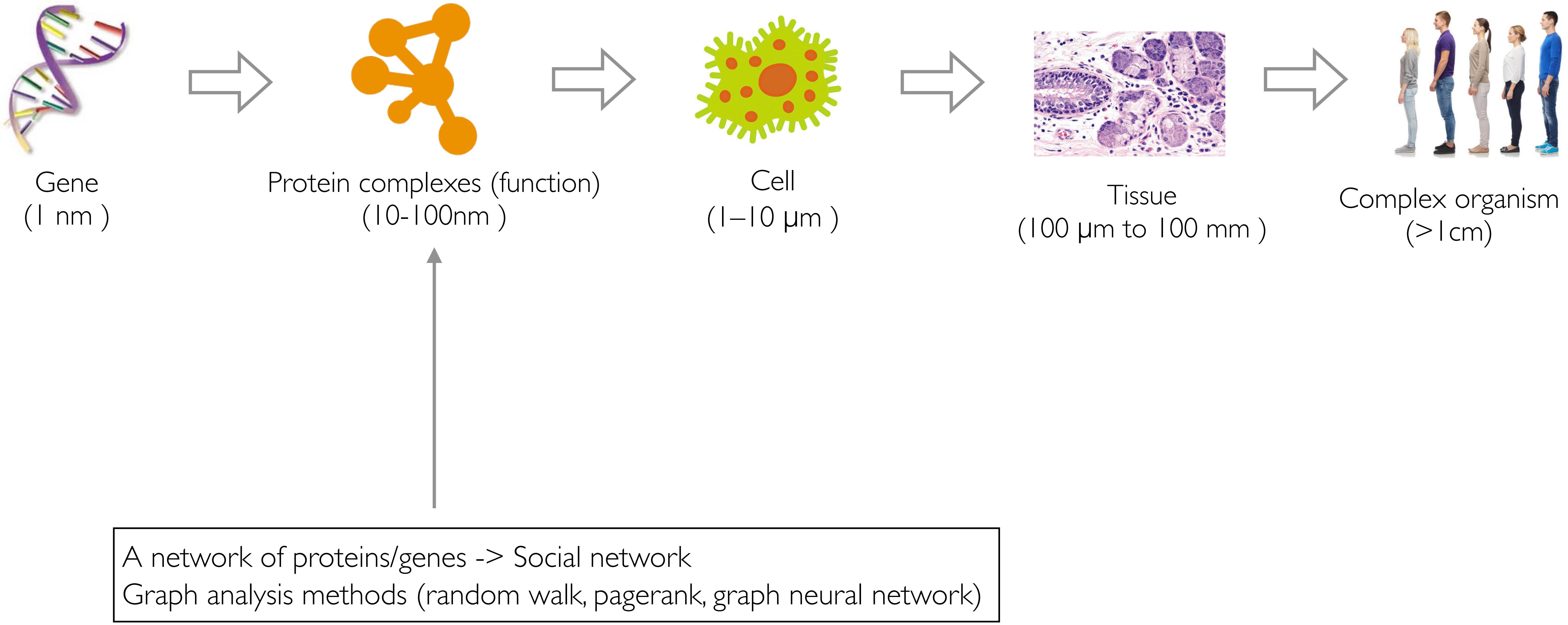
Tissue
(100 μm to 100 mm)

Complex organism
(>1cm)

A sequence of amino acids/nucleic acids -> A sequence of word/character
NLP methods (edit distance, LSTM, BERT)

Computational challenge: modeling the order in the sequence

Data structure for each scale: network



Gene
(1 nm)

Protein complexes (function)
(10-100nm)

Cell
(1–10 μm)

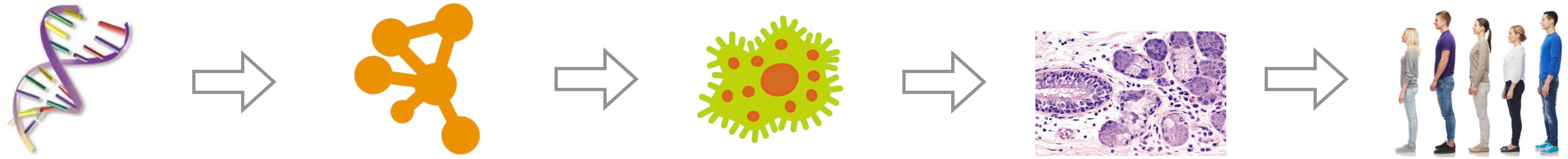
Tissue
(100 μm to 100 mm)

Complex organism
(>1cm)

A network of proteins/genes -> Social network
Graph analysis methods (random walk, pagerank, graph neural network)

Computational challenge: interaction, synergistic effect

Data structure for each scale: cell



Gene
(1 nm)

Protein complexes (function)
(10-100nm)

Cell
(1–10 µm)

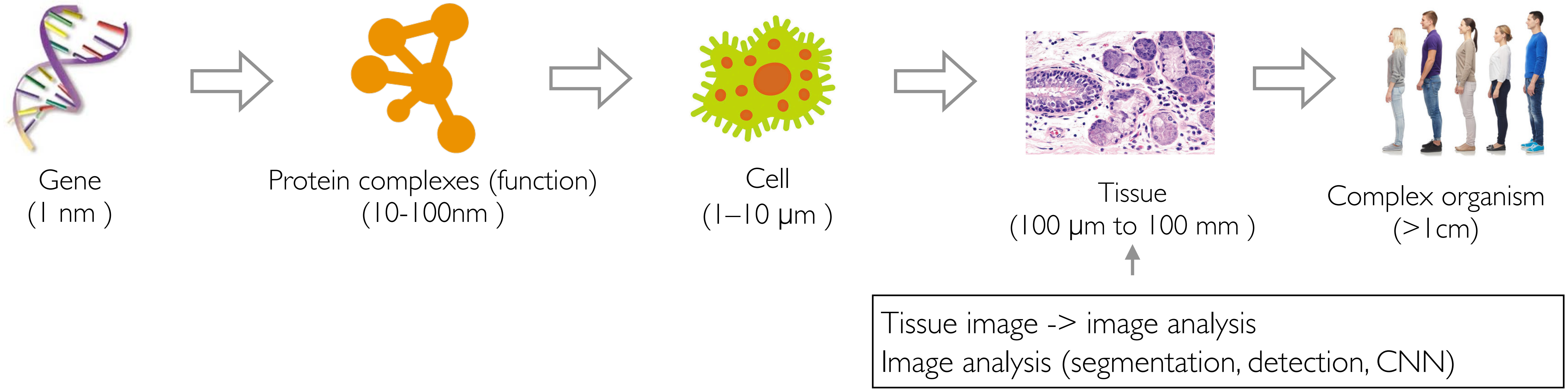
Tissue
(100 µm to 100 mm)

Complex organism
(>1cm)

A cell by gene matrix -> vector/matrix (high-dimensional, no spatial information)
Dimensionality reduction methods (PCA, t-SNE, variety of embedding methods)

High-dimensional, noisy, large-scale

Data structure for each scale: tissue



Gene
(1 nm)

Protein complexes (function)
(10-100nm)

Cell
(1–10 µm)

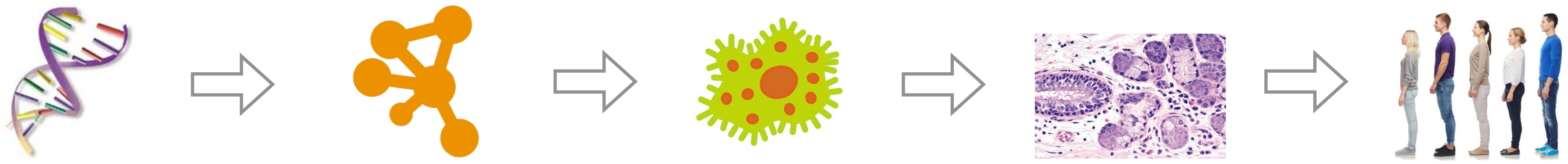
Tissue
(100 µm to 100 mm)

Complex organism
(>1cm)

Tissue image -> image analysis
Image analysis (segmentation, detection, CNN)

Image analysis, lack of high-quality annotations

Data structure for each scale: organism



Gene
(1 nm)

Protein complexes (function)
(10-100nm)

Cell
(1–10 μm)

Tissue
(100 μm to 100 mm)

Complex organism
(>1cm)

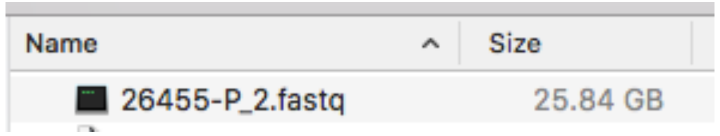
Disease mechanisms -> Multimodality
Integration of information from sequences, networks, images and matrixes

Multi-modality and heterogeneous

How did they do this?

DNA sample

Sequencing machine
~2000 dollars

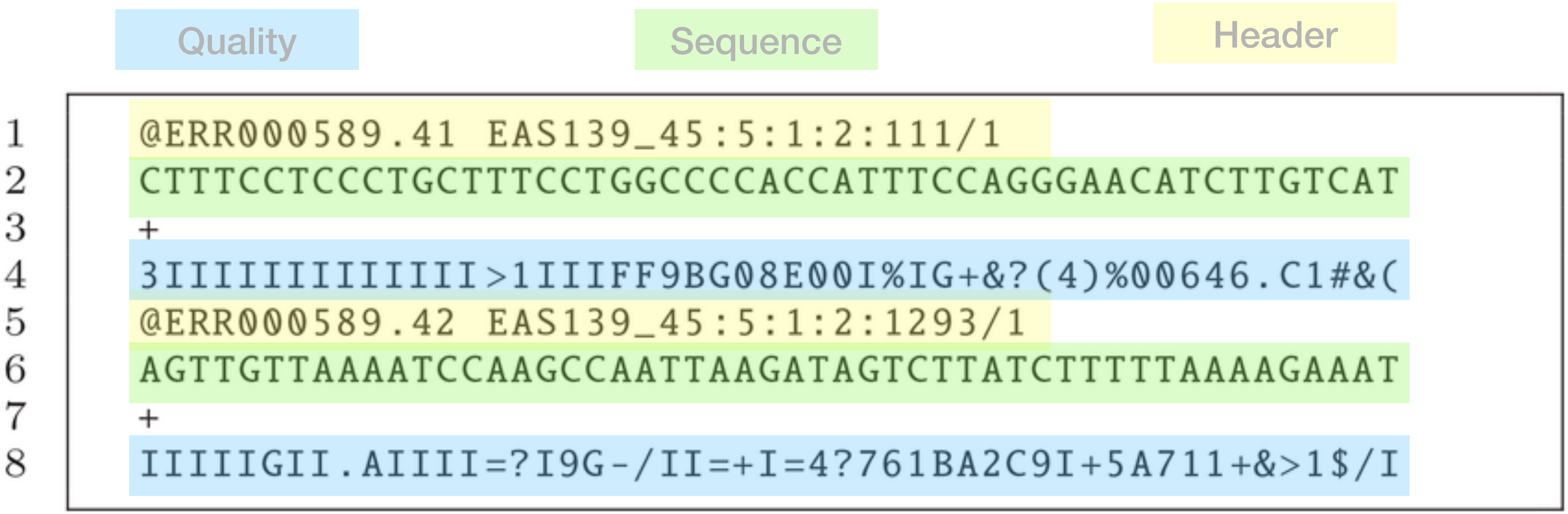


Name	Size
26455-P_2.fastq	25.84 GB

Your entire genome sequence
*.fastq file

Our job as a computer scientist: analyze *.fastq file

What does a fastq file look like?



```
@ERR000589.41 EAS139_45:5:1:2:111/1
CTTTCCTCCCTGCTTTCCTGGCCCCACCATTTCCAGGGAACATCTTGTCAT
+
3IIIIIIIIIIIII>1IIIFF9BG08E00I%IG+&?(4)%00646.C1#&(
@ERR000589.42 EAS139_45:5:1:2:1293/1
AGTTGTTAAAATCCAAGCCAATTAAGATAGTCTTATCTTTTTAAAAGAAT
+
IIIIIGII.AIIII=?I9G-/II=+I=4?761BA2C9I+5A711+&>1$/I
```

Very large! ~300000000 lines
Quality: ASCII chars

What should we do? Map each short sequence (we call it read) to the entire human genome

What does a fastq file look like?

Reference genome: "average" human genome.
Most widely used human genome GRCh38: derived from 13 thirteen anonymous volunteers

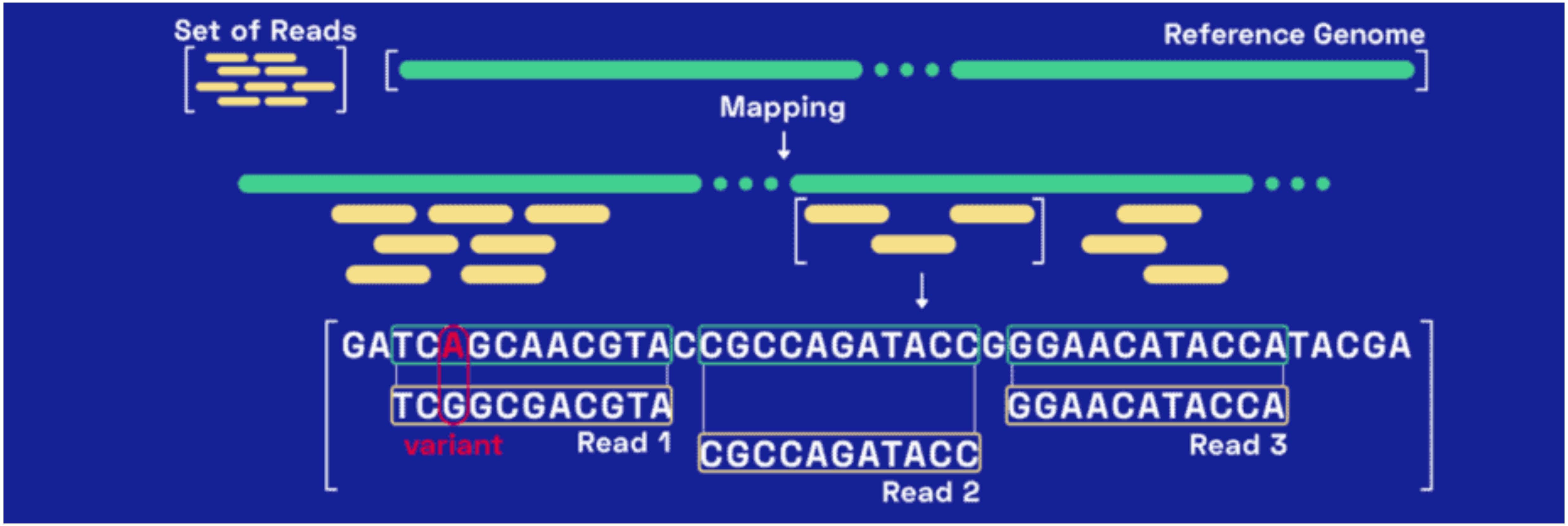
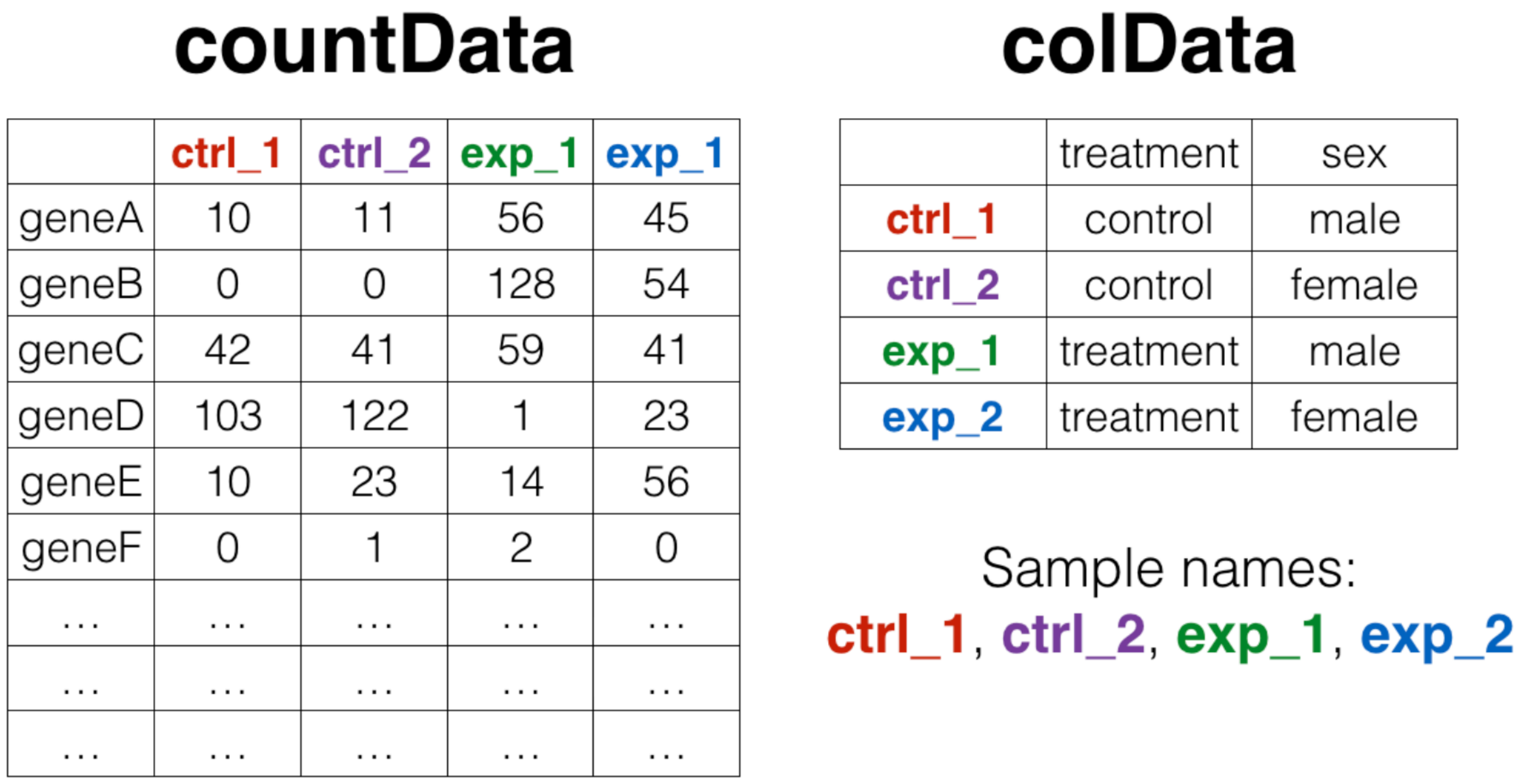


Image from Nebula Genomics

Processed data



countData

	ctrl_1	ctrl_2	exp_1	exp_1
geneA	10	11	56	45
geneB	0	0	128	54
geneC	42	41	59	41
geneD	103	122	1	23
geneE	10	23	14	56
geneF	0	1	2	0
…	…	…	…	…
…	…	…	…	…
…	…	…	…	…

colData

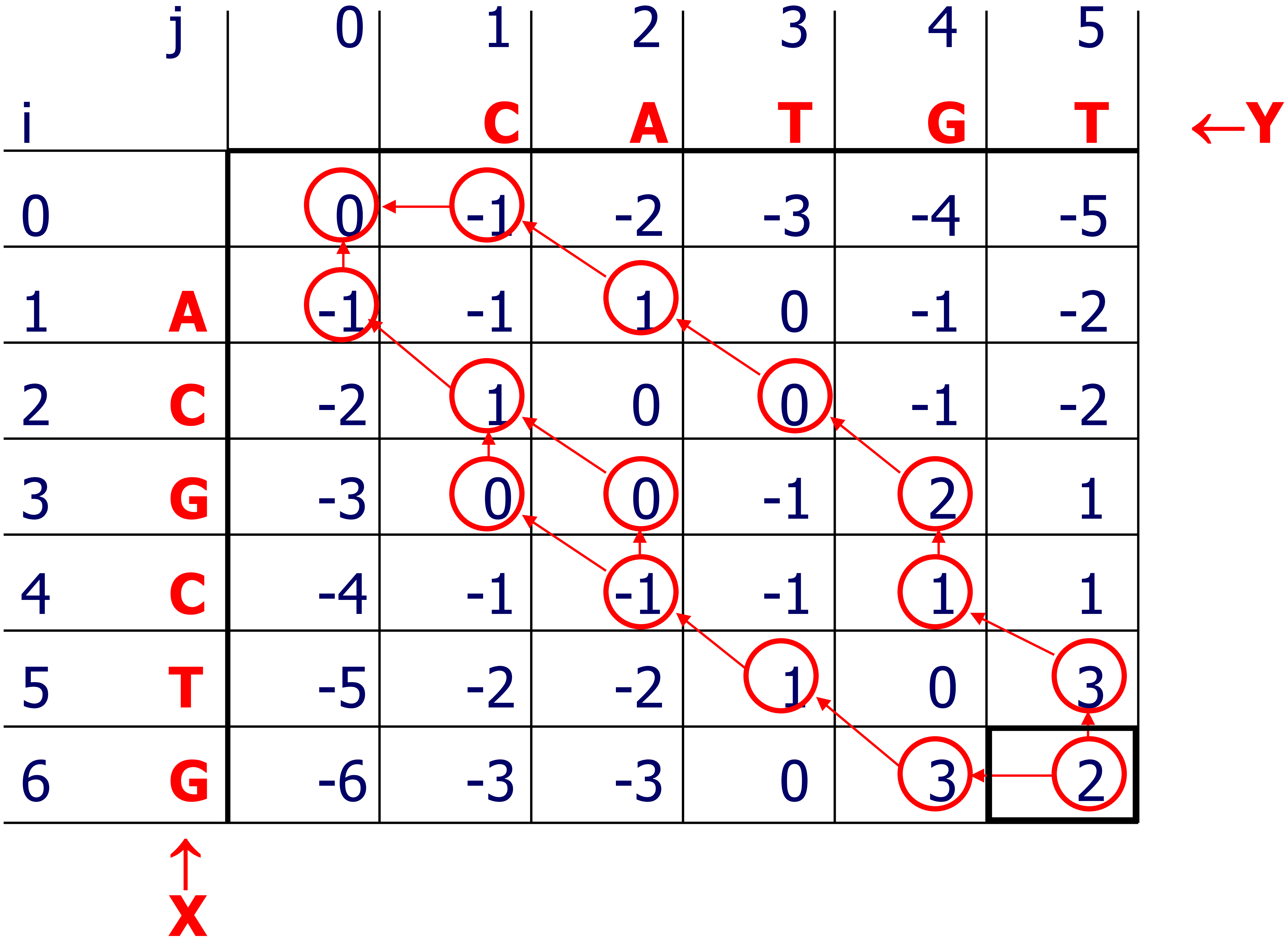
	treatment	sex
ctrl_1	control	male
ctrl_2	control	female
exp_1	treatment	male
exp_2	treatment	female

Sample names:
ctrl_1, **ctrl_2**, **exp_1**, **exp_2**

source: https://bioconnector.github.io/bims8382/r-rnaseq-airway.html

Finding alignments: trace back

Arrows = (ties for) max in F(i,j); 3 LR-to-UL paths = 3 optimal alignments

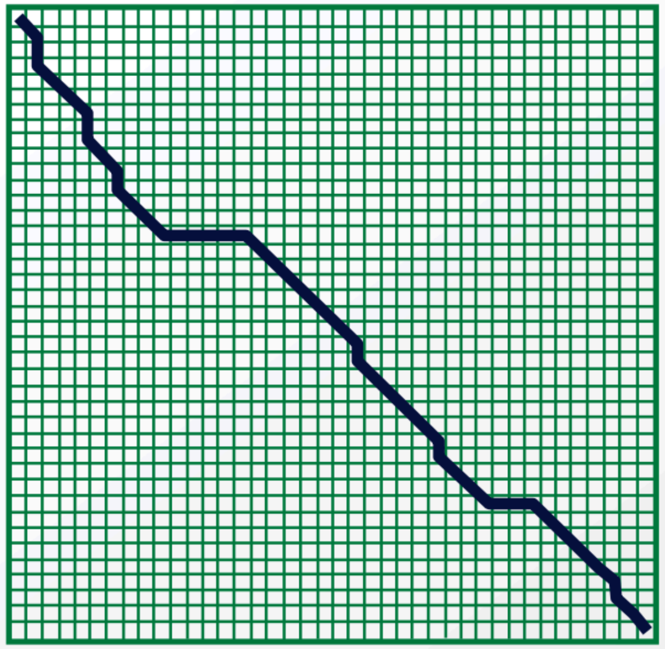


j		0	1	2	3	4	5
i			C	A	T	G	T
0		0	-1	-2	-3	-4	-5
1	A	-1	-1	1	0	-1	-2
2	C	-2	1	0	0	-1	-2
3	G	-3	0	0	-1	2	1
4	C	-4	-1	-1	-1	1	1
5	T	-5	-2	-2	1	0	3
6	G	-6	-3	-3	0	3	2

←Y

↑
X

Global Alignment vs. Local alignment



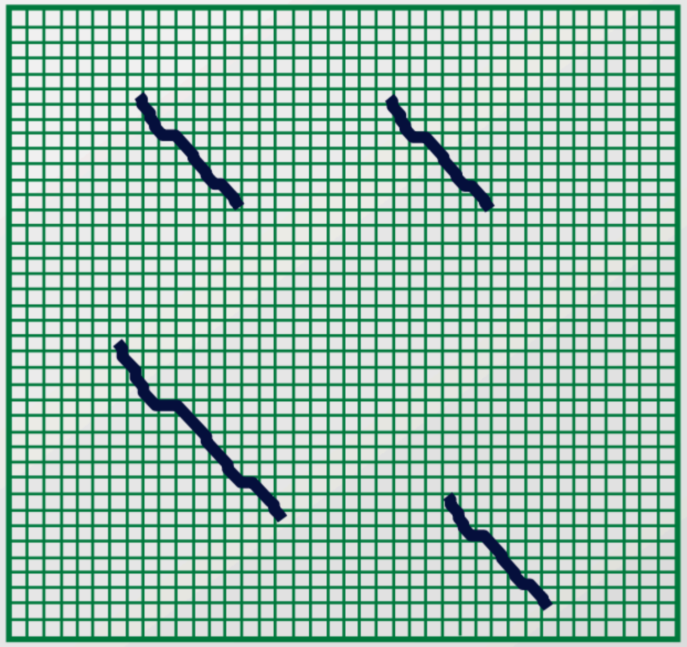
Needleman-Wunsch algorithm

Initialization: $F(0, 0) = 0$

Iteration:

$$F(i, j) = \max \begin{cases} F(i-1, j) - d \\ F(i, j-1) - d \\ F(i-1, j-1) + s(x_i, y_j) \end{cases}$$

Termination: Bottom right



Smith-Waterman algorithm

Initialization: **$F(0, j) = F(i, 0)$** $= 0$

Iteration:

$$F(i, j) = \max \begin{cases} \mathbf{0} \\ F(i-1, j) - d \\ F(i, j-1) - d \\ F(i-1, j-1) + s(x_i, y_j) \end{cases}$$

Termination: **Anywhere**

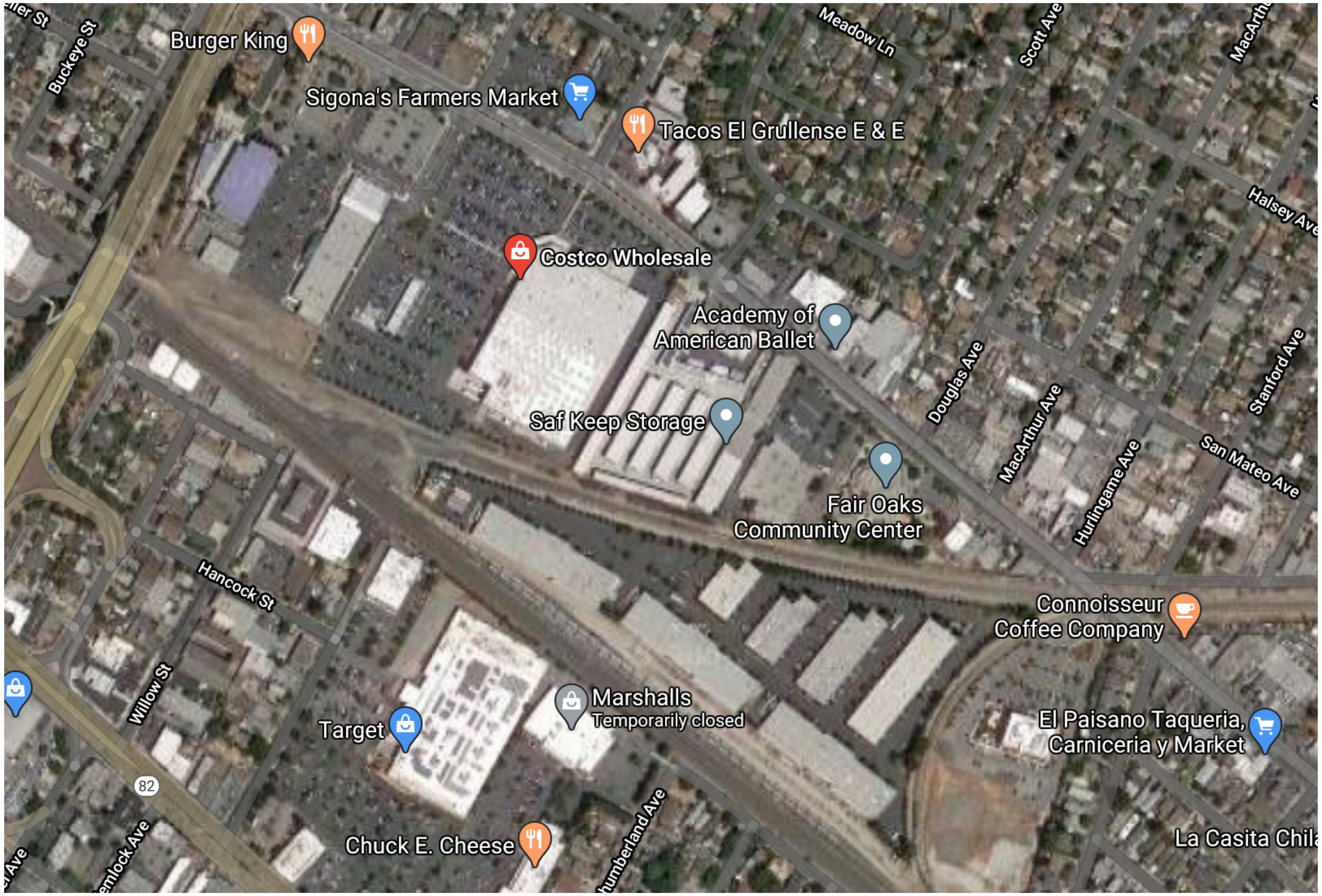
What is protein function prediction?

Human body = country

Single cell = town

Protein = brick, window, carpet, etc.

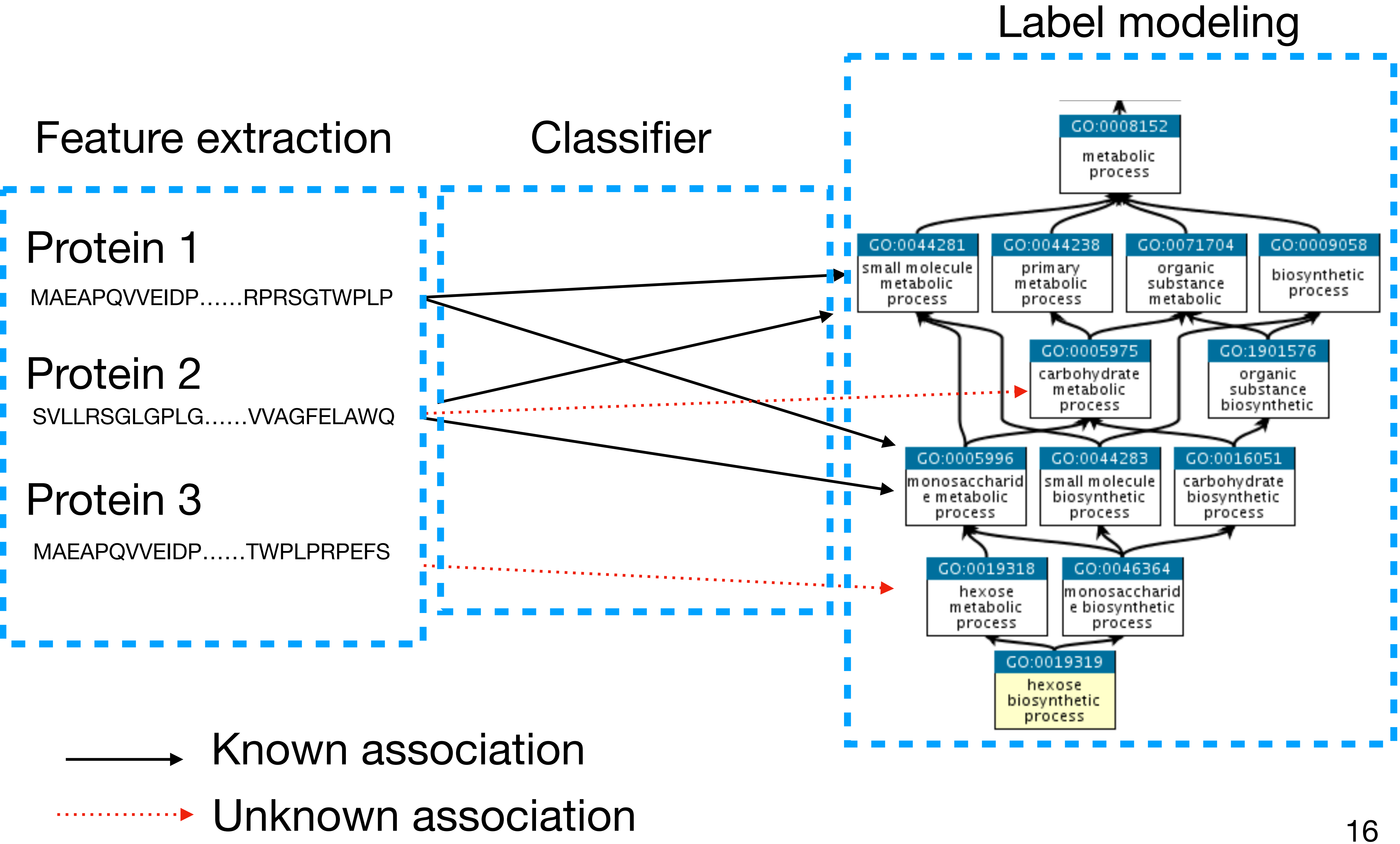
Protein function = fireproof, soundproof, etc.



Goal: classify each protein into its protein functions (multi-label)

Solution: find proteins with similar sequences

Problem setting for protein function prediction



Label modeling

Feature extraction

Classifier

Protein 1

MAEAPQVVEIDP……RPRSGTWPLP

Protein 2

SVLLRSGLGPLG……VVAGFELAWQ

Protein 3

MAEAPQVVEIDP……TWPLPRPEFS

GO:0008152 metabolic process

GO:0044281 small molecule metabolic process

GO:0044238 primary metabolic process

GO:0071704 organic substance metabolic

GO:0009058 biosynthetic process

GO:0005975 carbohydrate metabolic process

GO:1901576 organic substance biosynthetic

GO:0005996 monosaccharide metabolic process

GO:0044283 small molecule biosynthetic process

GO:0016051 carbohydrate biosynthetic process

GO:0019318 hexose metabolic process

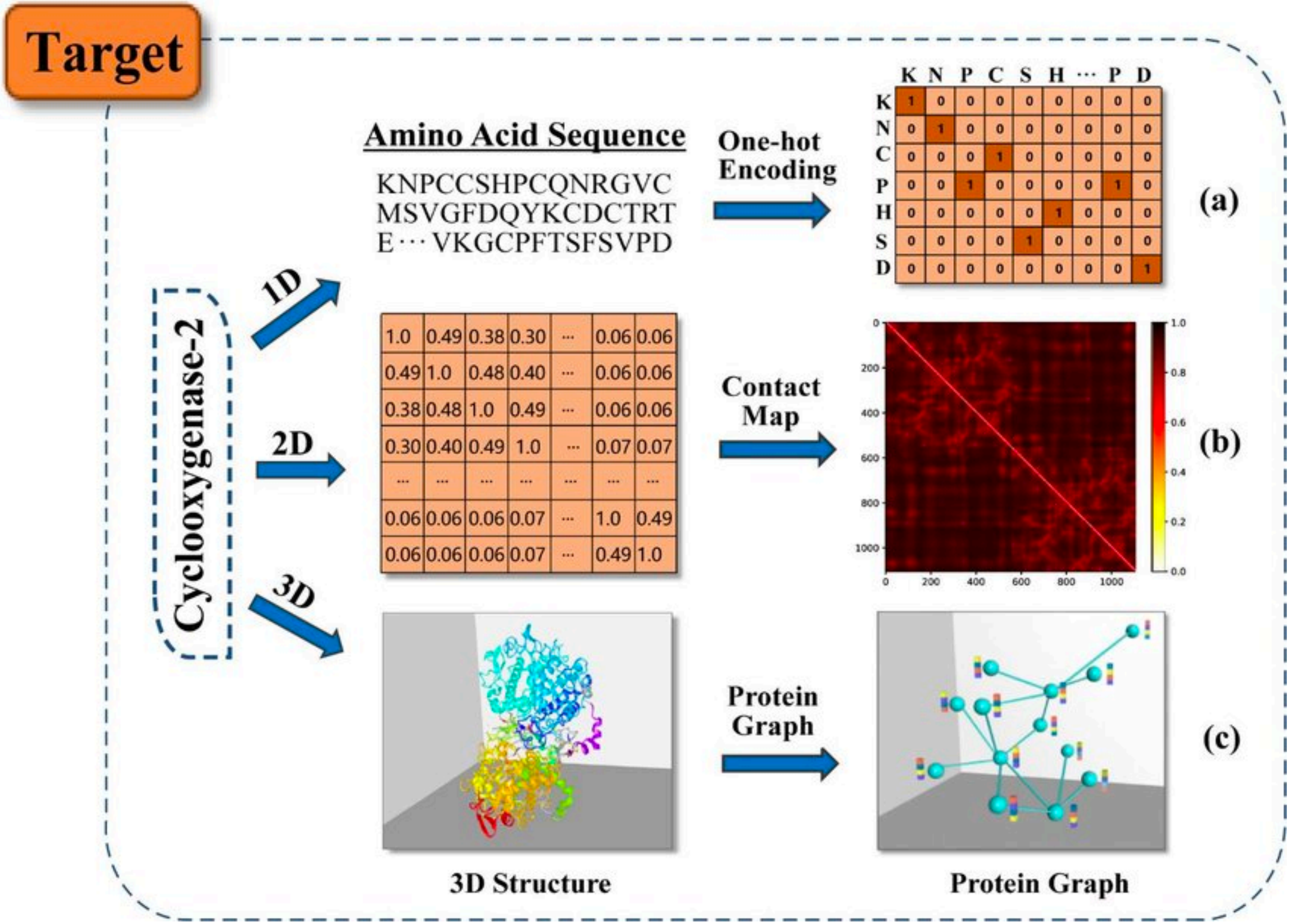
GO:0046364 monosaccharide biosynthetic process

GO:0019319 hexose biosynthetic process

Known association

Unknown association

Converting proteins to numeral features



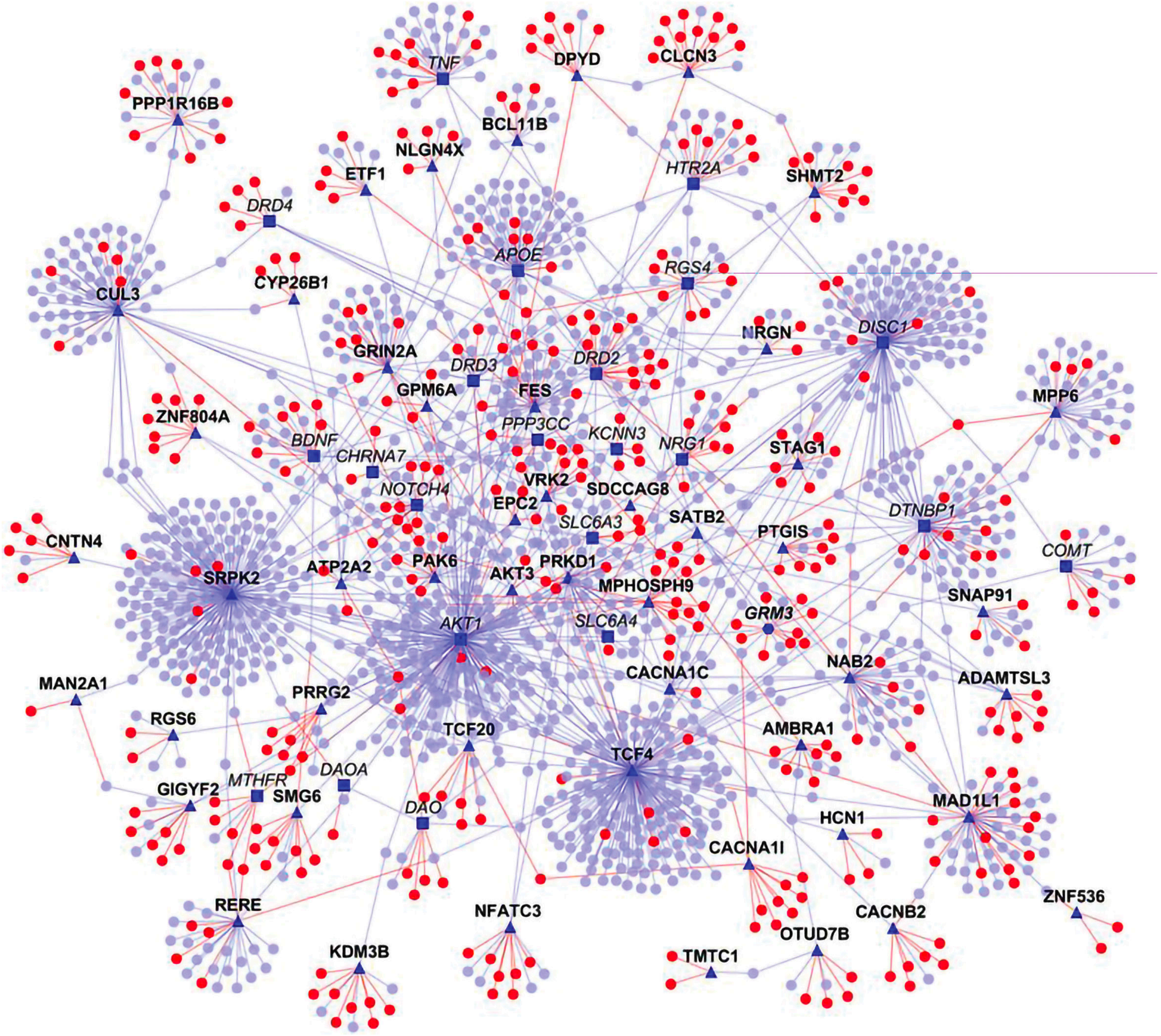
KNPCCSHPCQNRGVC
MSVGFDQYKCDCTRT
E··· VKGCPFTSFSVPD

	K	N	P	C	S	H	···	P	D
K	1	0	0	0	0	0	0	0	0
N	0	1	0	0	0	0	0	0	0
C	0	0	0	1	0	0	0	0	0
P	0	0	1	0	0	0	0	1	0
H	0	0	0	0	0	1	0	0	0
S	0	0	0	0	1	0	0	0	0
D	0	0	0	0	0	0	0	0	1

1.0	0.49	0.38	0.30	···	0.06	0.06
0.49	1.0	0.48	0.40	···	0.06	0.06
0.38	0.48	1.0	0.49	···	0.06	0.06
0.30	0.40	0.49	1.0	···	0.07	0.07
···	···	···	···	···	···	···
0.06	0.06	0.06	0.07	···	1.0	0.49
0.06	0.06	0.06	0.07	···	0.49	1.0

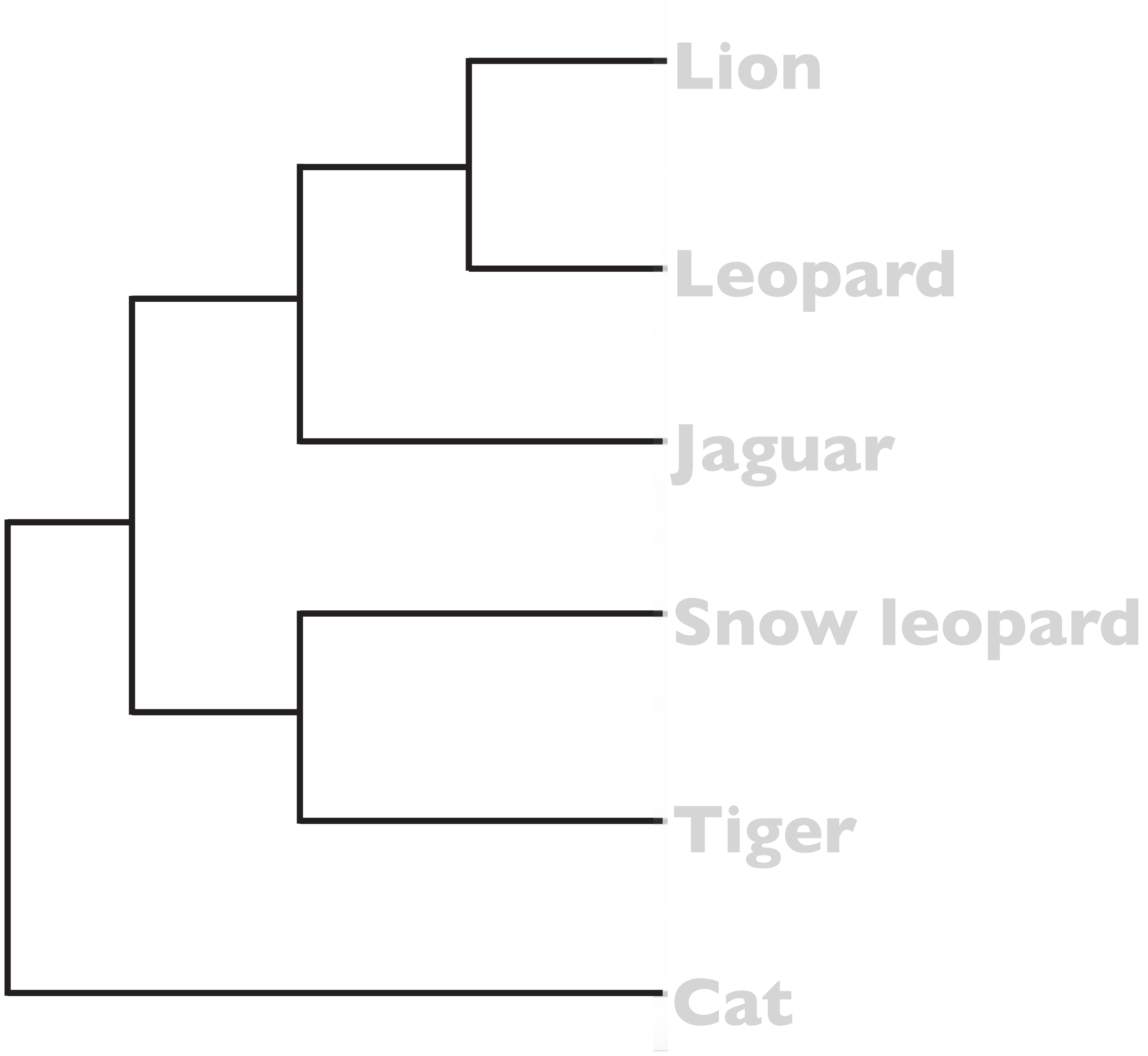
source: Deep learning for drug repurposing: methods, databases, and applications

Protein protein network



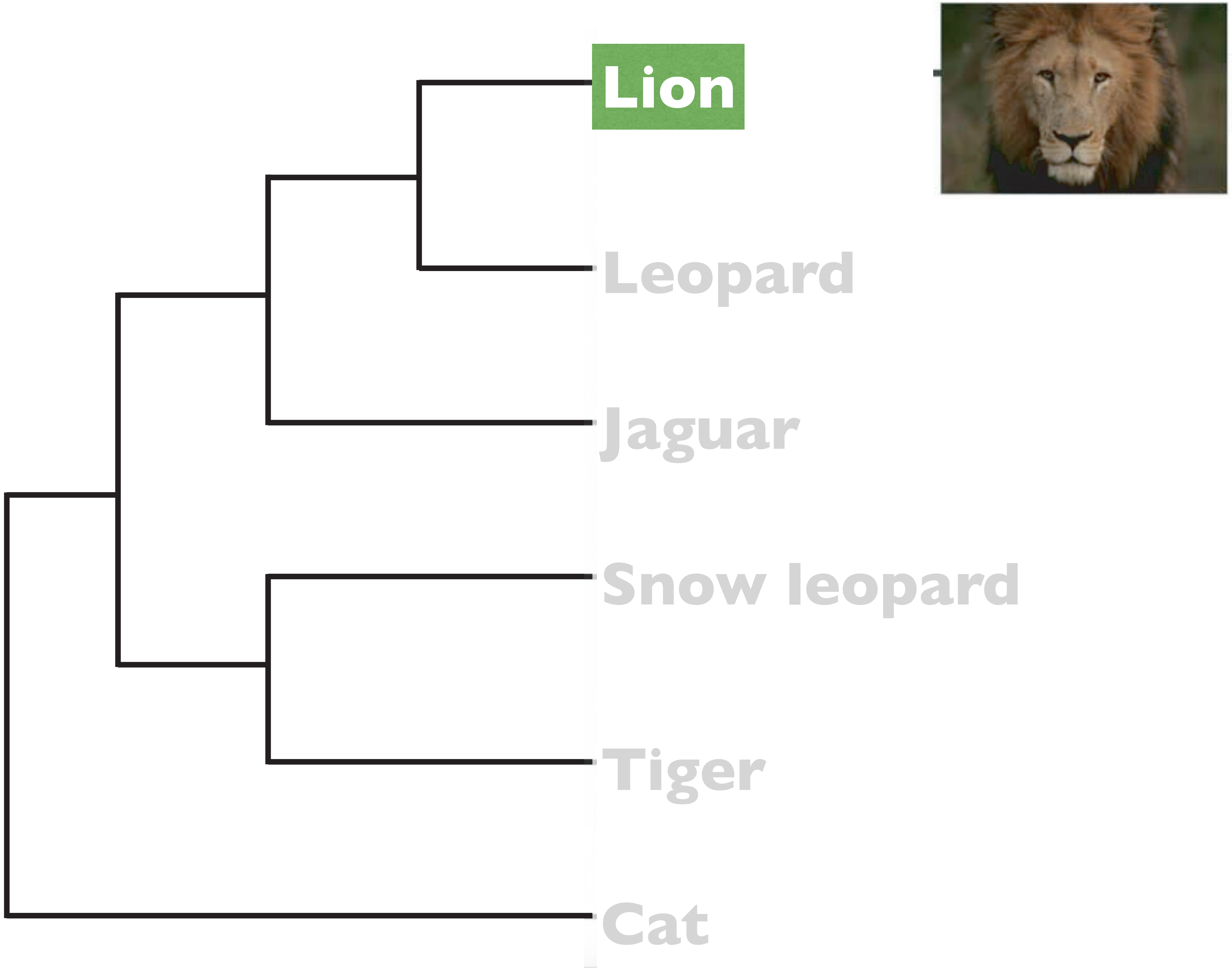
ONTOLOGICAL CLASSIFICATION OF UNSEEN ANIMALS

Ontology of great cats



ONTOLOGICAL CLASSIFICATION OF UNSEEN ANIMALS

Ontology of great cats



Lion

Leopard

Jaguar

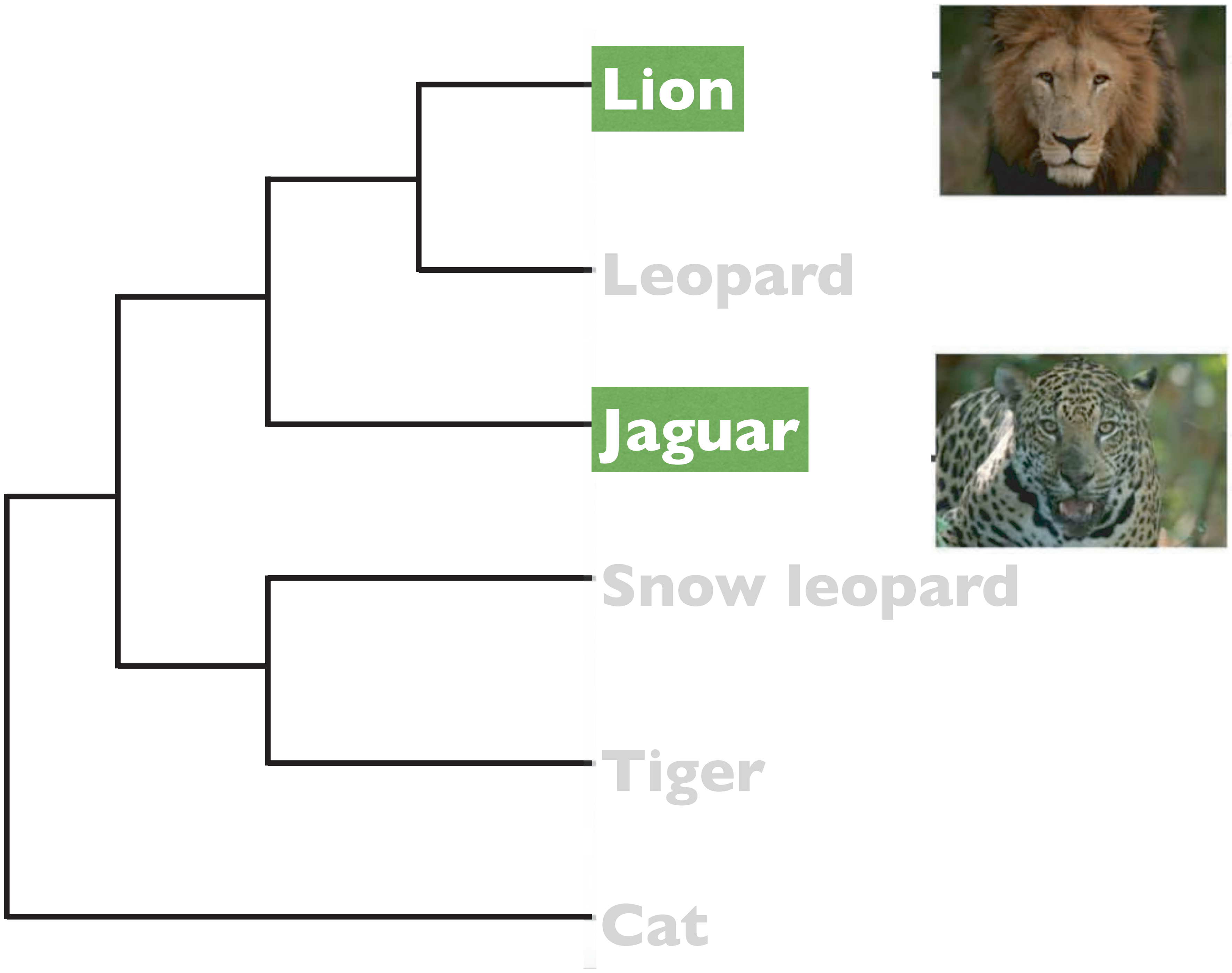
Snow leopard

Tiger

Cat

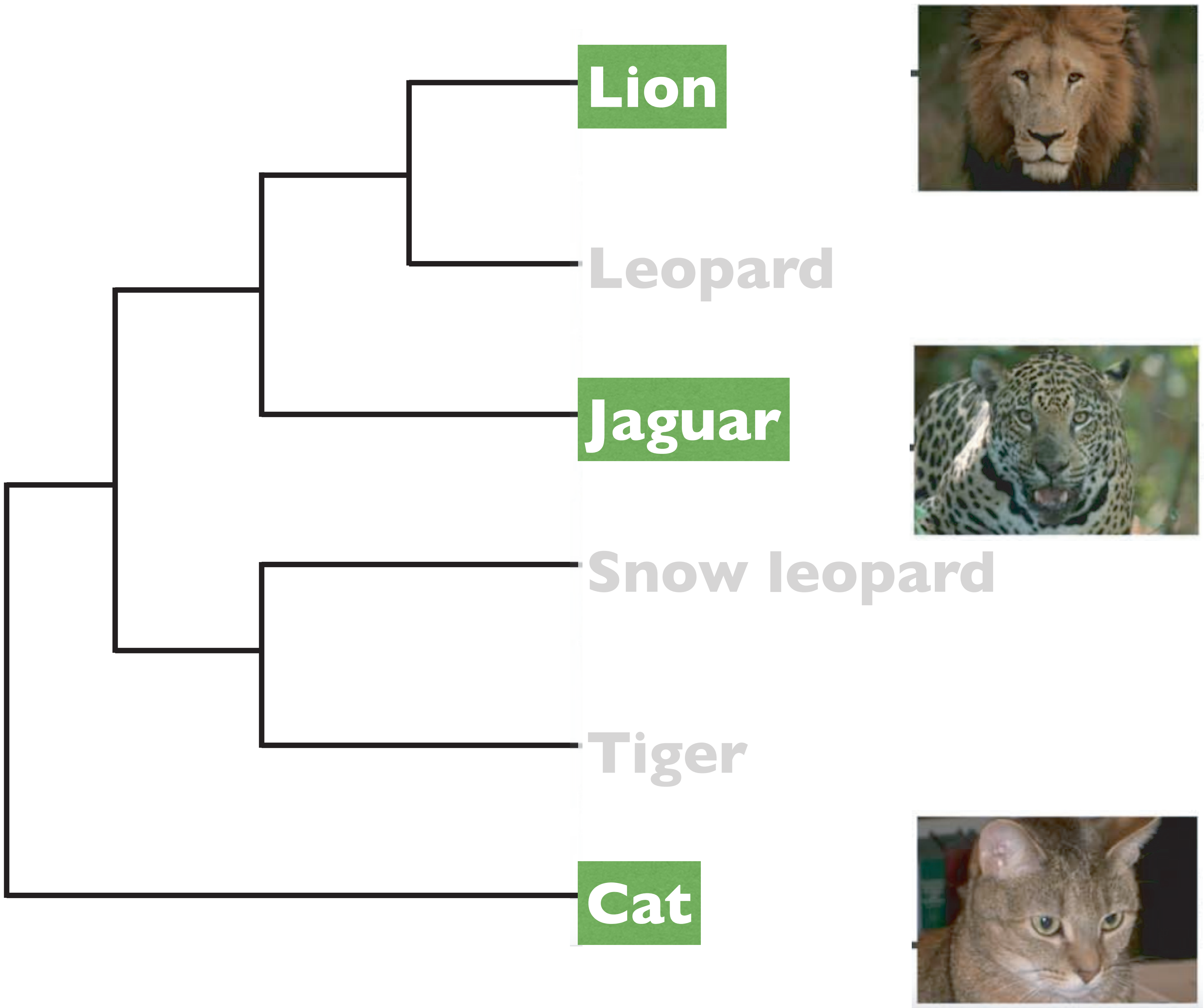
ONTOLOGICAL CLASSIFICATION OF UNSEEN ANIMALS

Ontology of great cats



ONTOLOGICAL CLASSIFICATION OF UNSEEN ANIMALS

Ontology of great cats



Lion

Leopard

Jaguar

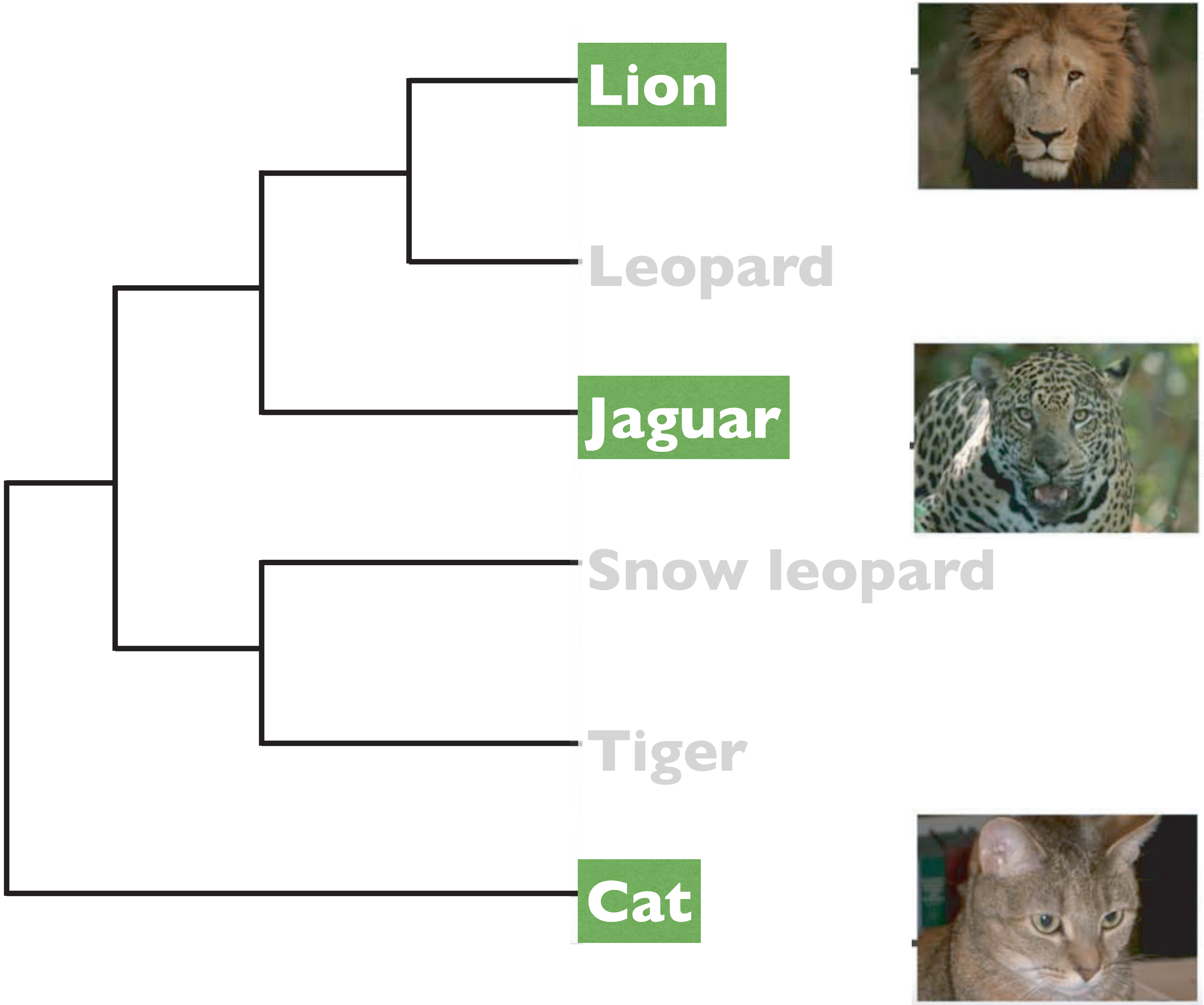
Snow leopard

Tiger

Cat

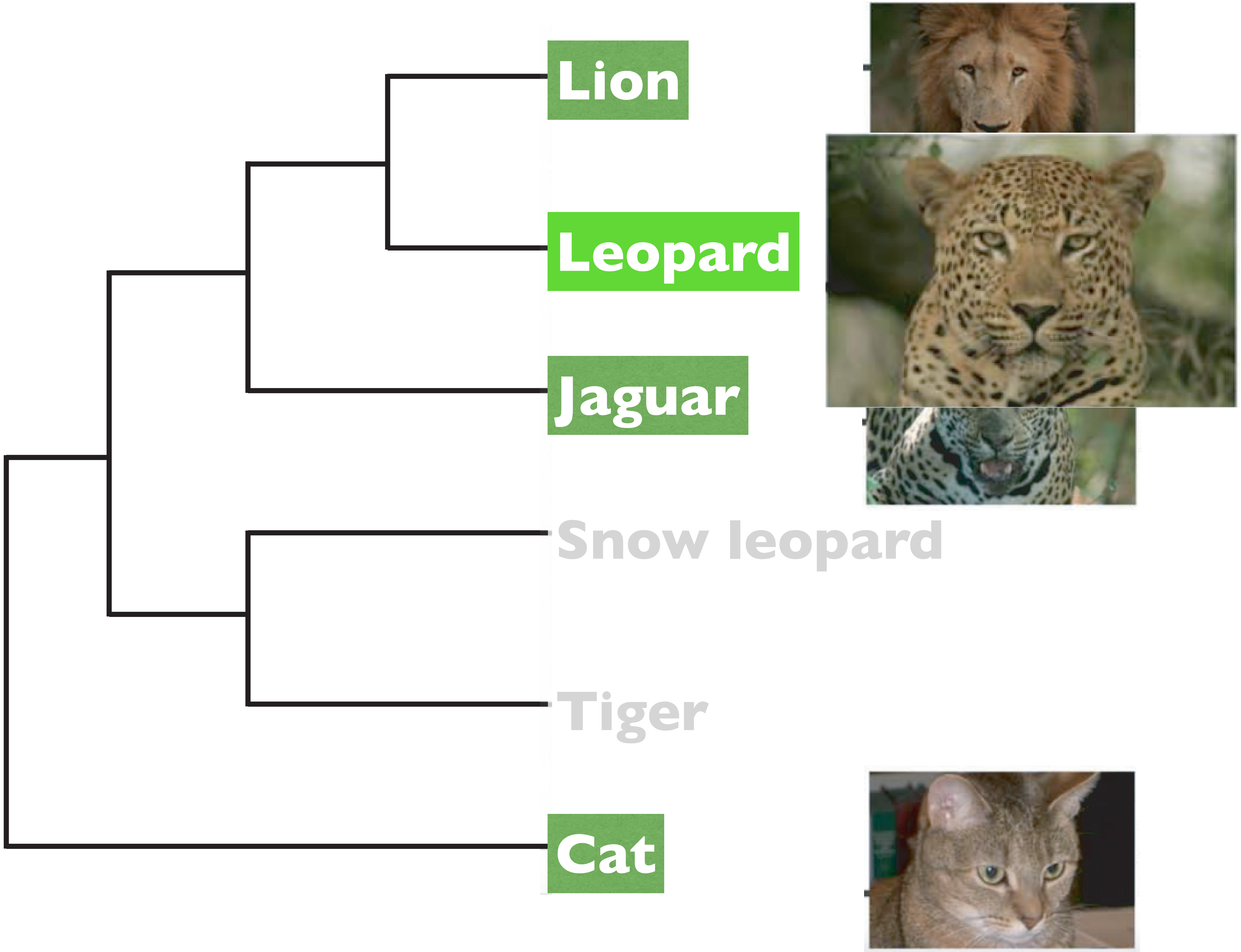
ONTOLOGICAL CLASSIFICATION OF UNSEEN ANIMALS

Ontology of great cats



ONTOLOGICAL CLASSIFICATION OF UNSEEN ANIMALS

Ontology of great cats



Lion

Leopard

Jaguar

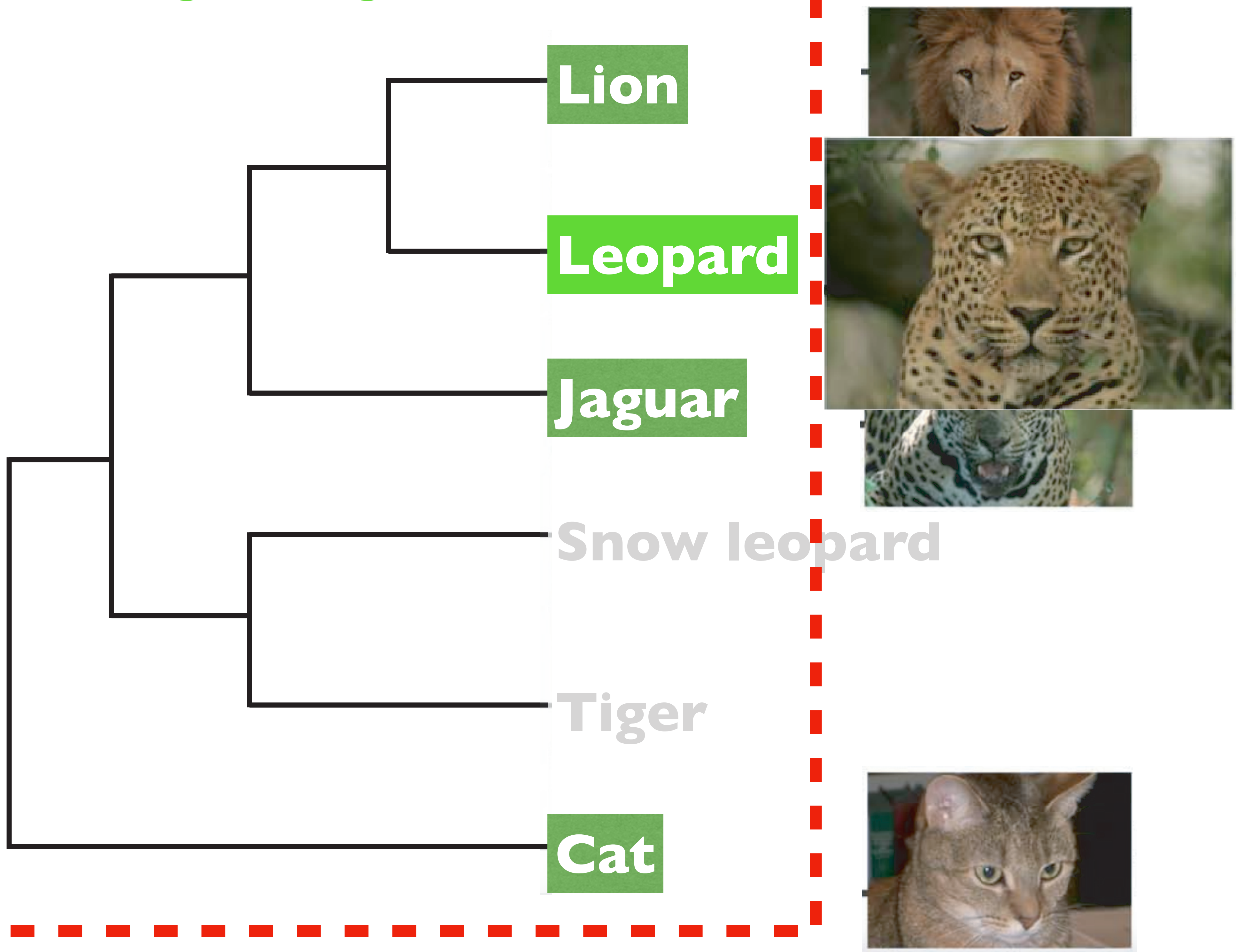
Snow leopard

Tiger

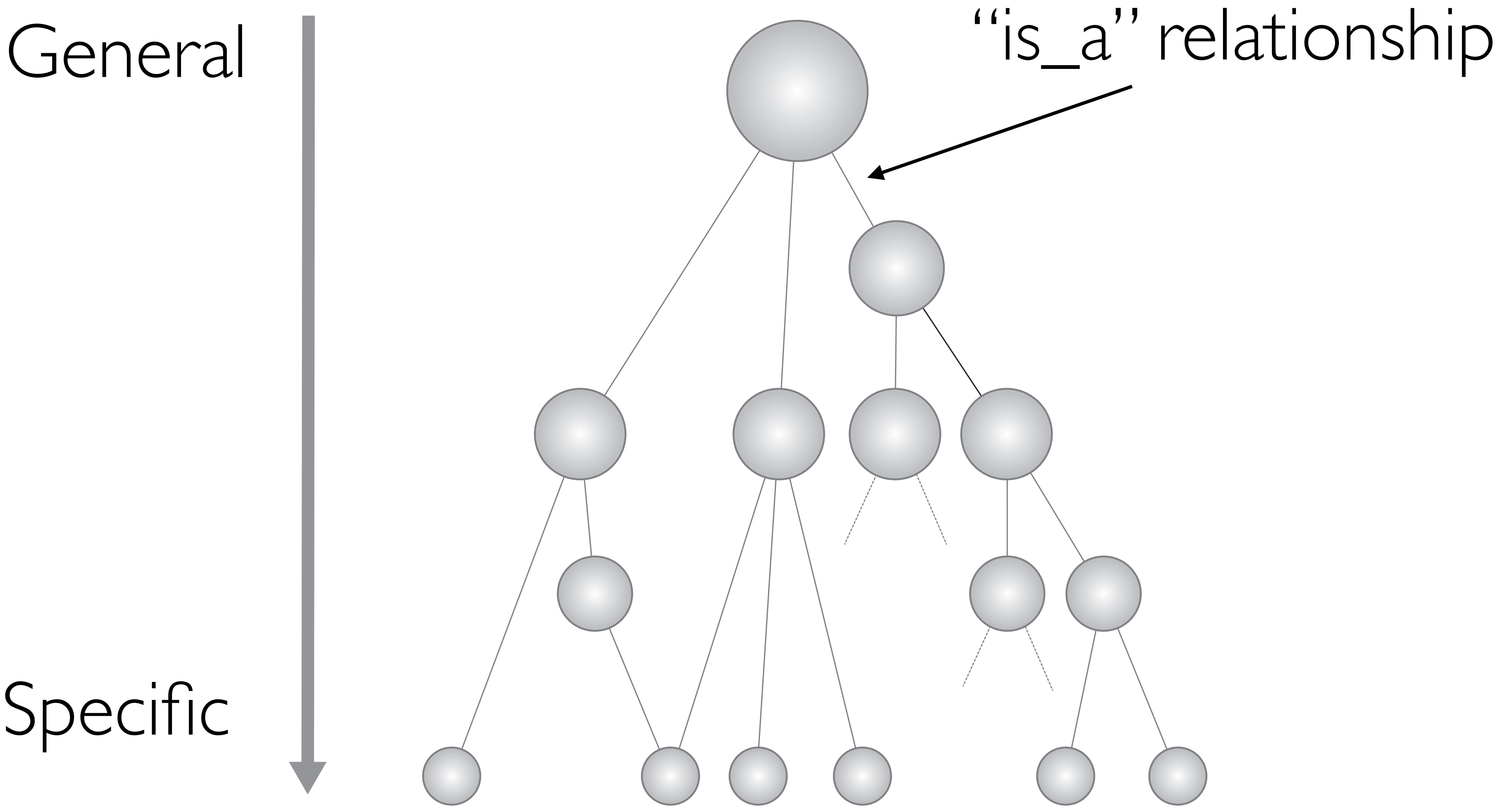
Cat

ONTOLOGICAL CLASSIFICATION OF UNSEEN ANIMALS

Ontology of great cats



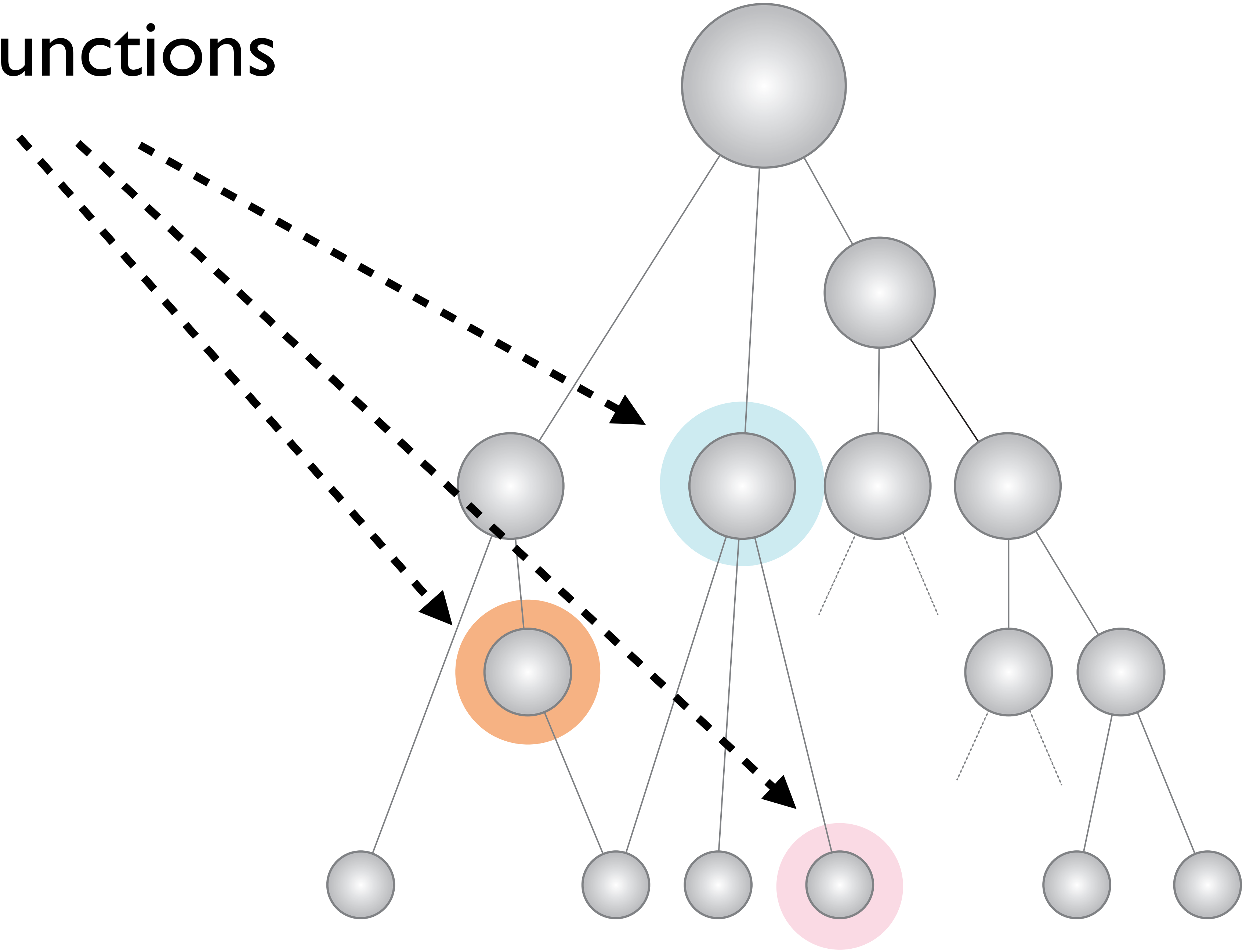
solution: use gene ontology as side information



Each node is a function. 23k functions in total.

solution: use gene ontology as side information

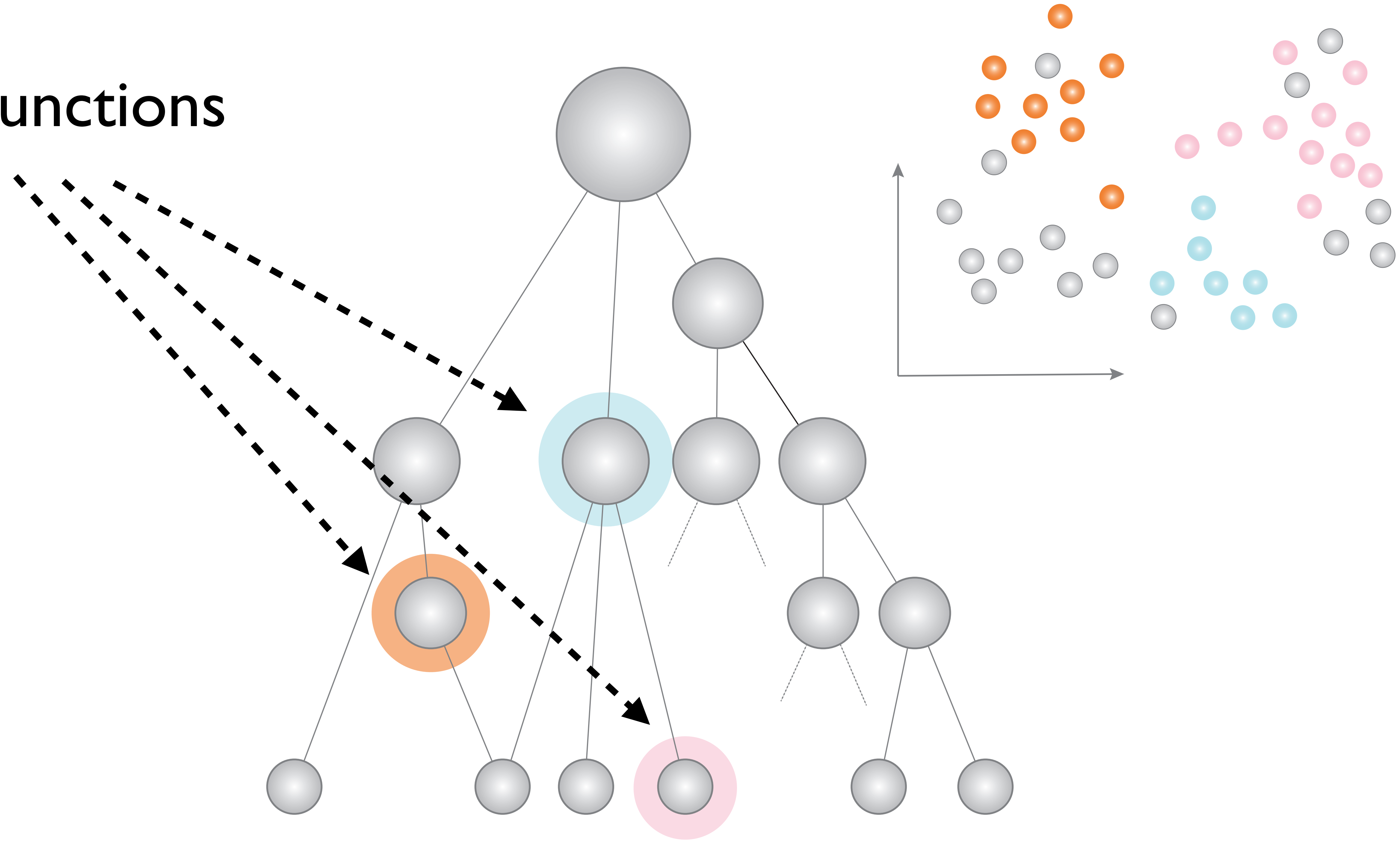
Seen functions



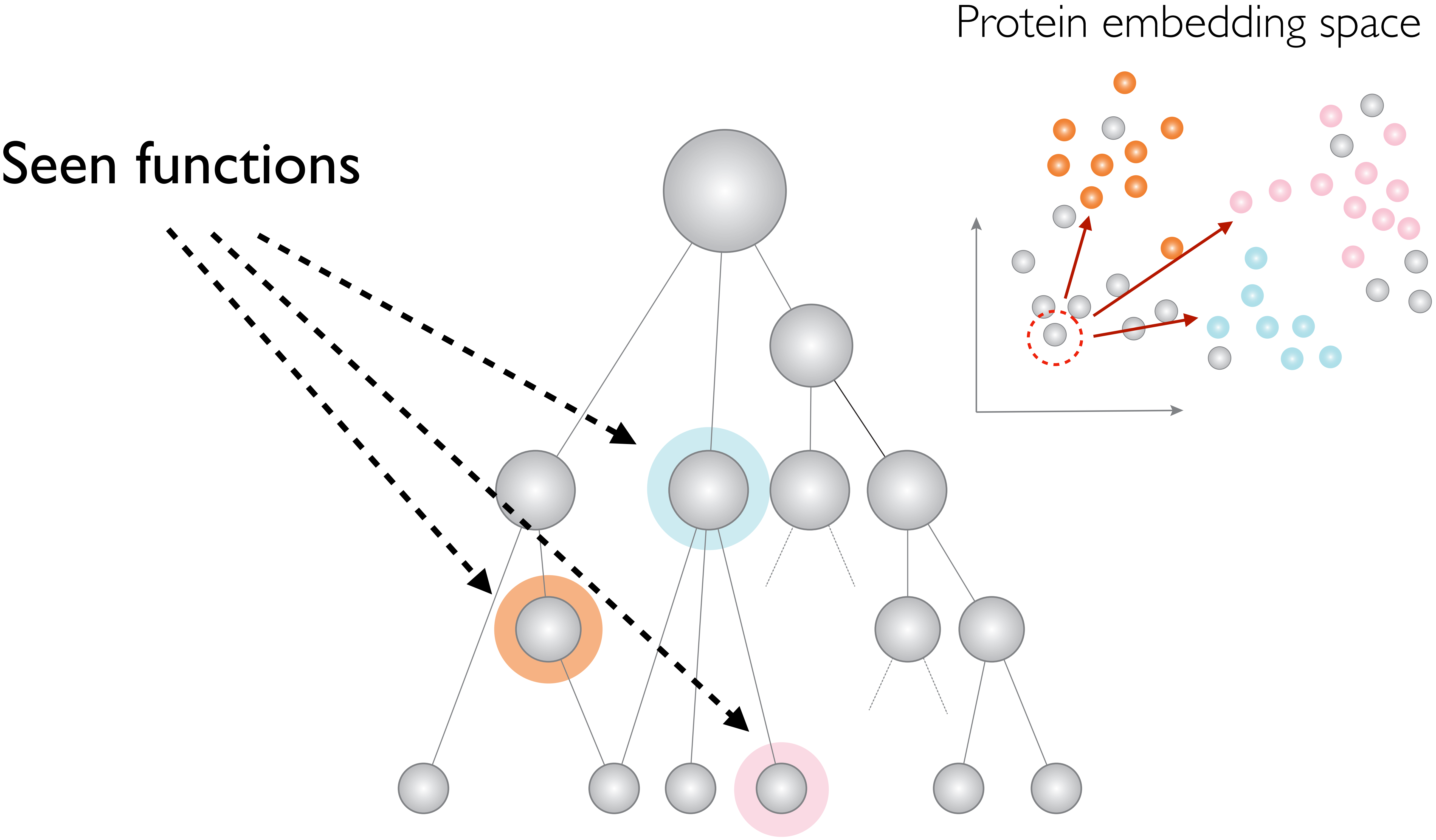
solution: use gene ontology as side information

Seen functions

Protein embedding space



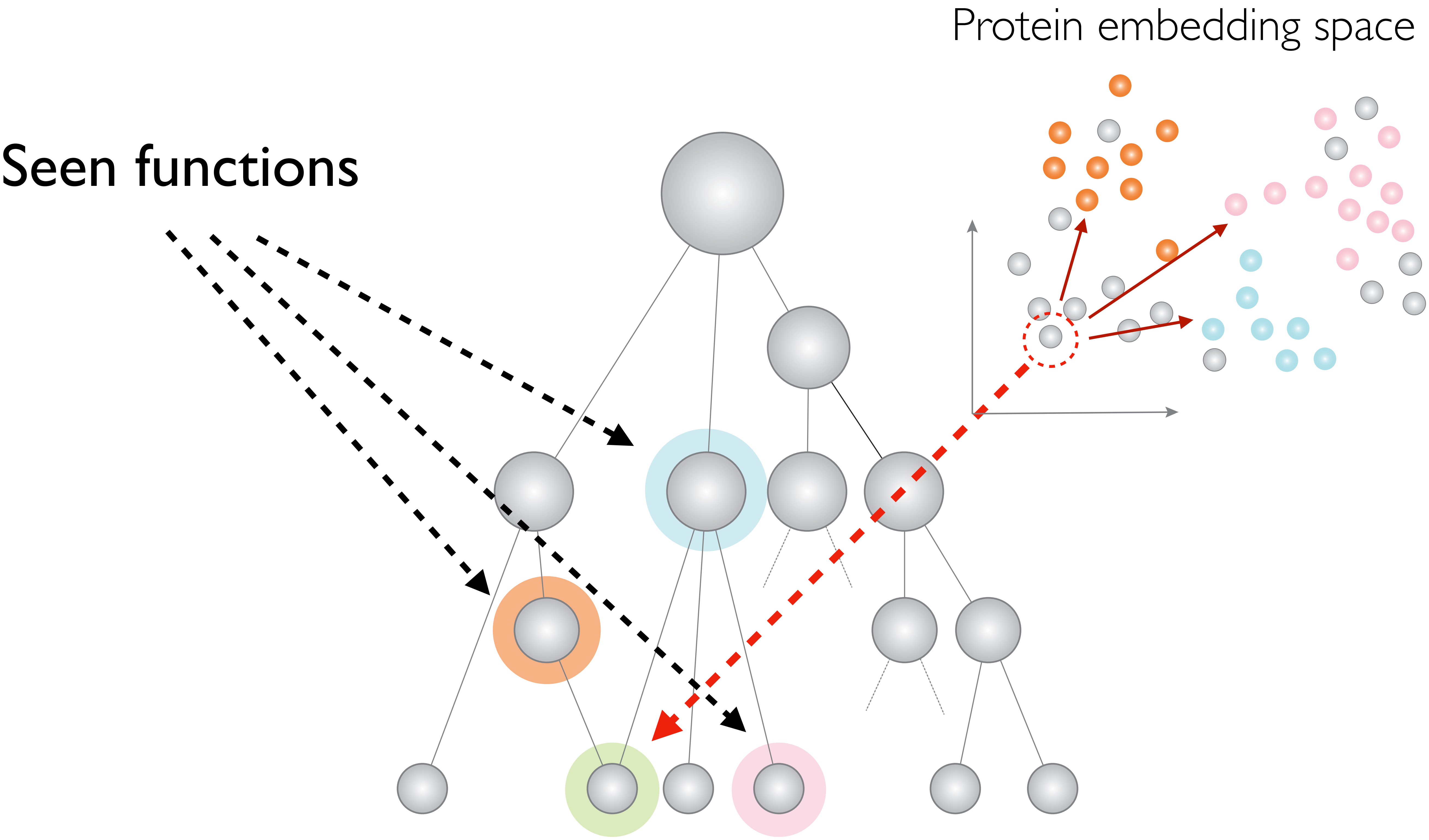
solution: use gene ontology as side information



Seen functions

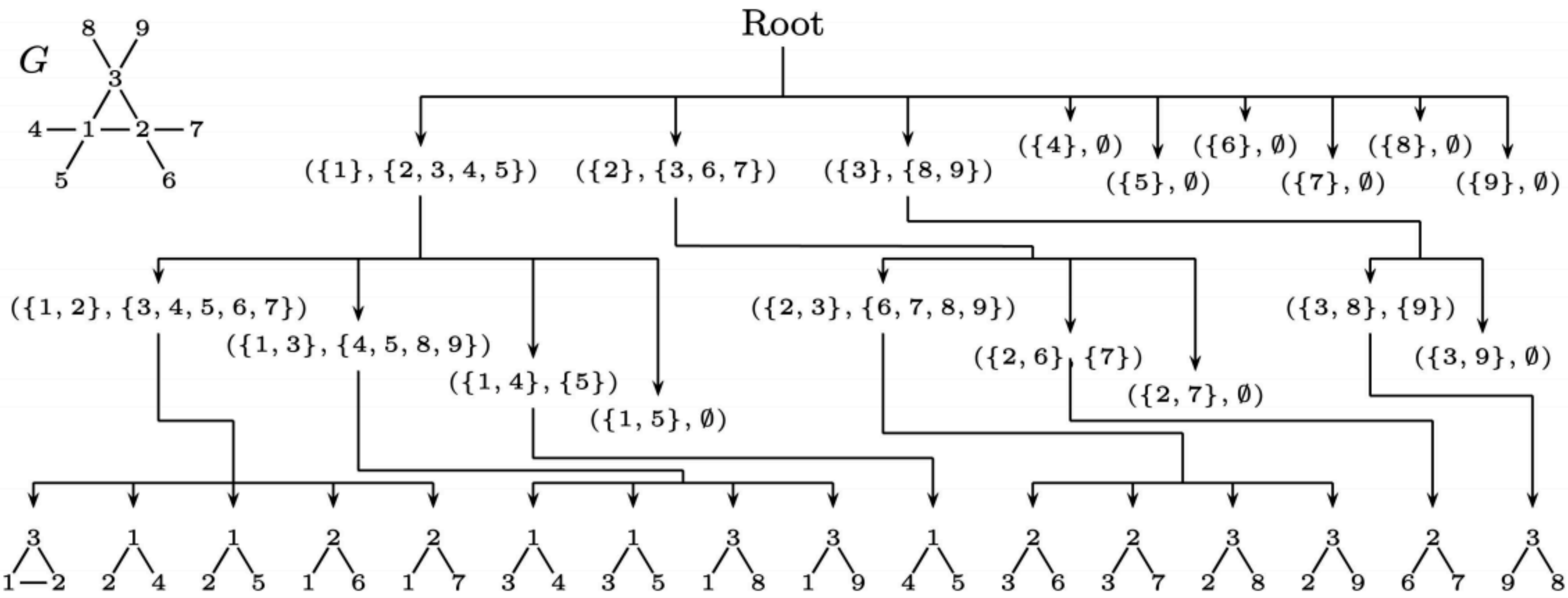
Protein embedding space

solution: use gene ontology as side information



Exact Subgraph Enumeration

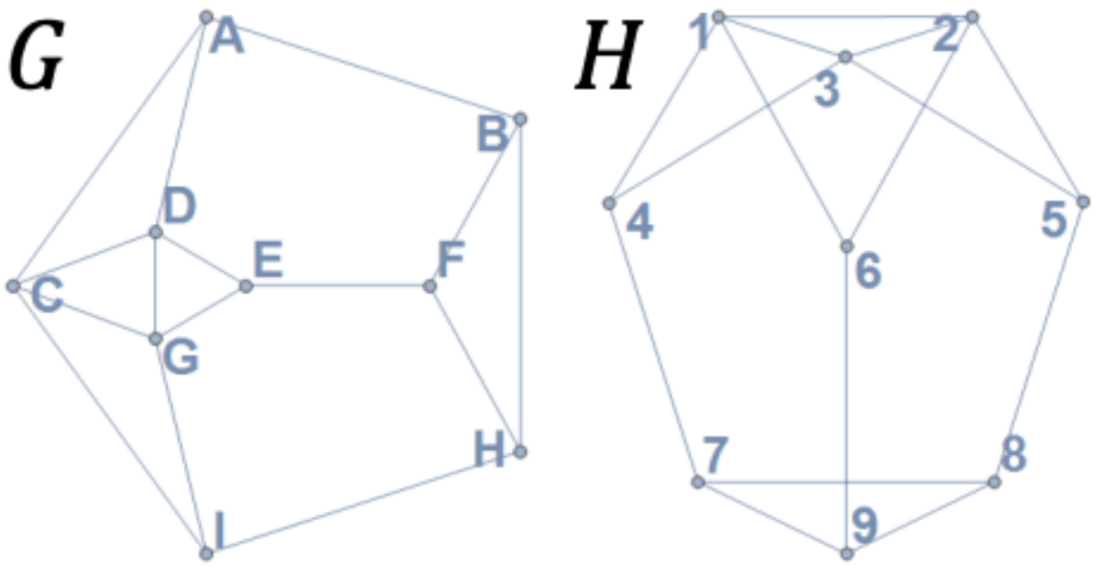
K=3



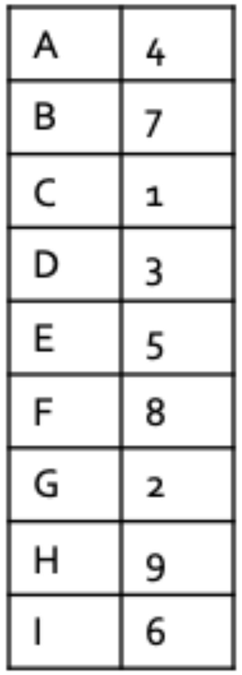
Node set (currently in the motif)
Candidate set (neighbors of node set)
Nodes in the candidate set must have larger node id than nodes in the node set to avoid duplicate computing

Graph Isomorphism

- If you ignore the node types/features and edge types/features, you will find some subgraphs are topologically equivalent.



f

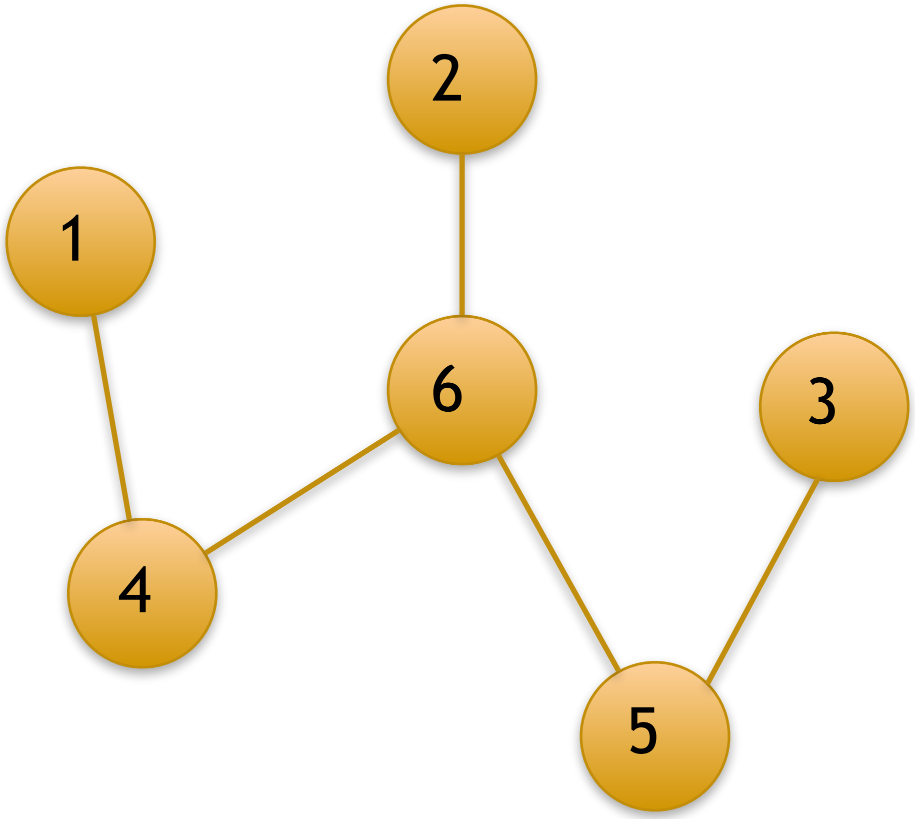


A	4
B	7
C	1
D	3
E	5
F	8
G	2
H	9
I	6

- Graphs G and H are isomorphic if there exists a bijection f: V(G) → V(H) such that: Any two nodes u and v of G are adjacent in G iff f(u) and f(v) are adjacent in H.

Graph Isomorphism Detection Algorithm

- **McKay's Canonical Graph Labeling Algorithm:** Nauty, Trace, Bliss all have their own implementations of Mckay's algorithm. [McKay 1981]
- Time complexity exp(O(n2/3))
- Intuition: First hash two graphs as two strings and then compare two strings.
- Label each node according to their degrees first. Iterate over each edge.
- Put a "1" if there is an edge between those two nodes, a "0" if not.



(1,2) (1,3) (1,4) (1,5) (1,6)	00100
(2,3) (2,4) (2,5) (2,6)	0001
(3,4) (3,5) (3,6)	010
(4,5) (4,6)	01
(5,6)	1

001000001010011

N node: N * (N-1)/2 edges
5 node: 15 edges

Graph Isomorphism Detection Algorithm

- But the order of the edge matters in this hash coding. To solve that problem we want to enumerate all the orderings.
- We first sort the all the nodes according to their degrees.
- Within each degree bin, we enumerate all the orderings.

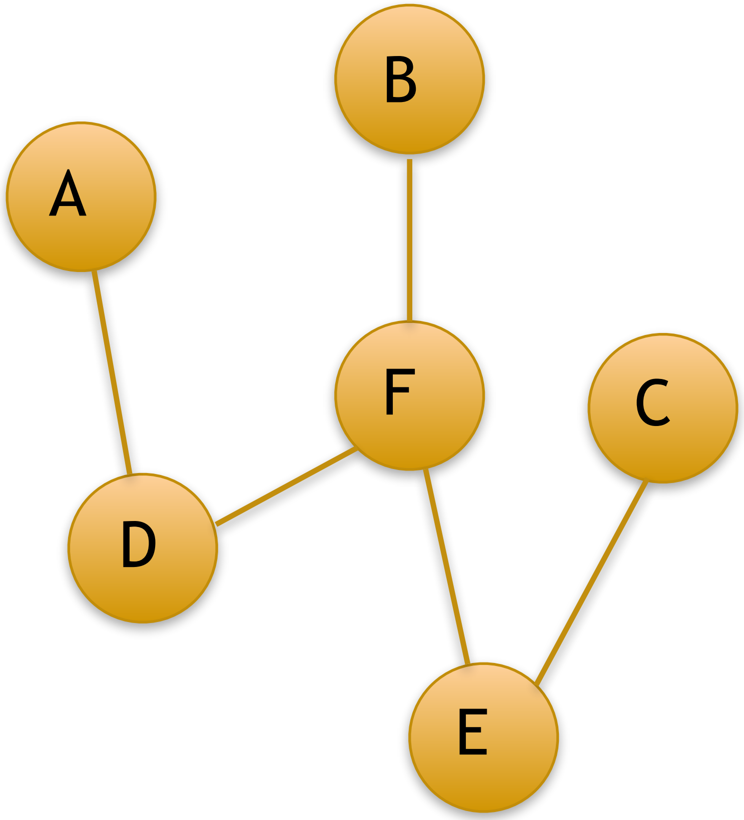
{A,B,C} | {D, E} | {F}

{A,B,C} | {D, E} | {F} {A,C,B} | {D, E} | {F} {B,A,C} | {D, E} | {F} {B,C, A} | {D, E} | {F}

{A,B,C} | {D, E} | {F} {A,B,C} | {E, D} | {F}

001000001010011 000100101000011

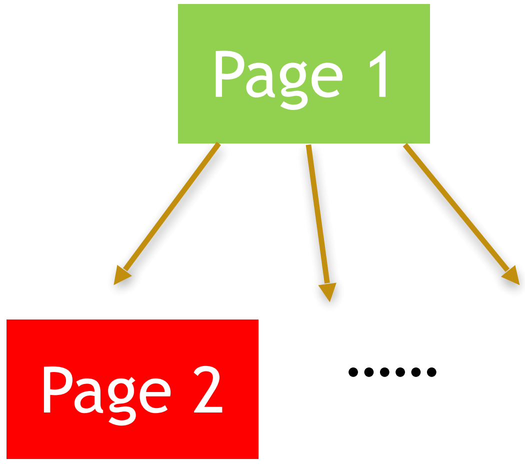
This hash code matches the previous one!



Random walk interpretation

The vector r can be reinterpreted as a probability vector to visit each website

- Imagine a random web surfer
  - At any time *k*, surfer has a probability vector r^k to visit a web page following the out-link.
  - Process repeats indefinitely



Page 1

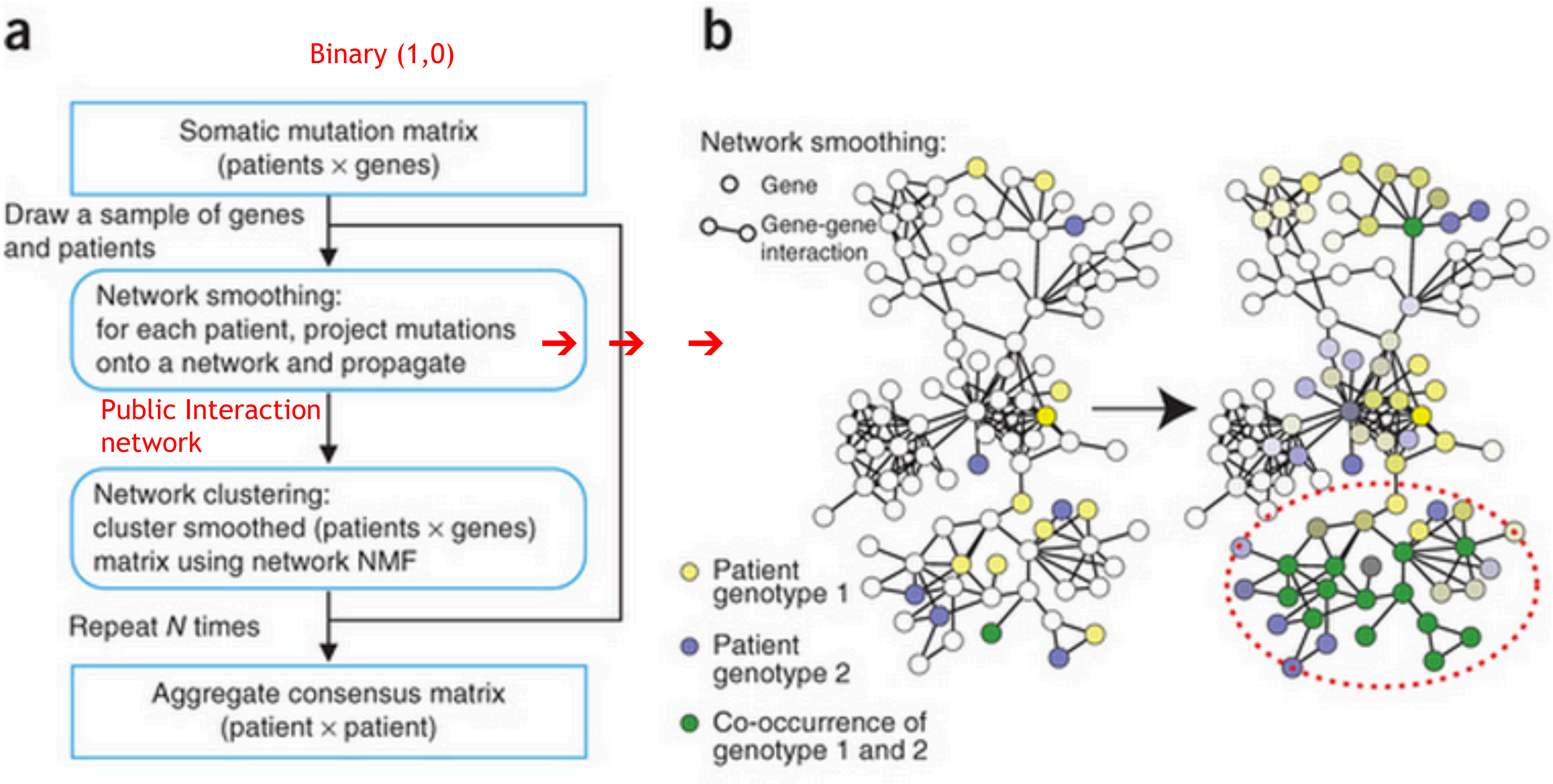
Page 2

......

$$\begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_j \\ \vdots \\ r_N \end{bmatrix} = \begin{bmatrix} 0 & , & 1/d_2 & , & \dots & , & 1/d_N \\ 1/d_1 & , & 0 & , & \dots & , & 0 \\ \vdots & & \vdots & & & & \vdots \\ 1/d_1 & , & 1/d_2 & , & \dots & , & 1/d_N \\ \vdots & & \vdots & & & & \vdots \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_j \\ \vdots \\ r_N \end{bmatrix}$$

$$r = Mr$$

Overview of network-based stratification



Binary (1,0)

Somatic mutation matrix (patients × genes)

Draw a sample of genes and patients

Network smoothing: for each patient, project mutations onto a network and propagate

Public Interaction network

Network clustering: cluster smoothed (patients × genes) matrix using network NMF

Repeat *N* times

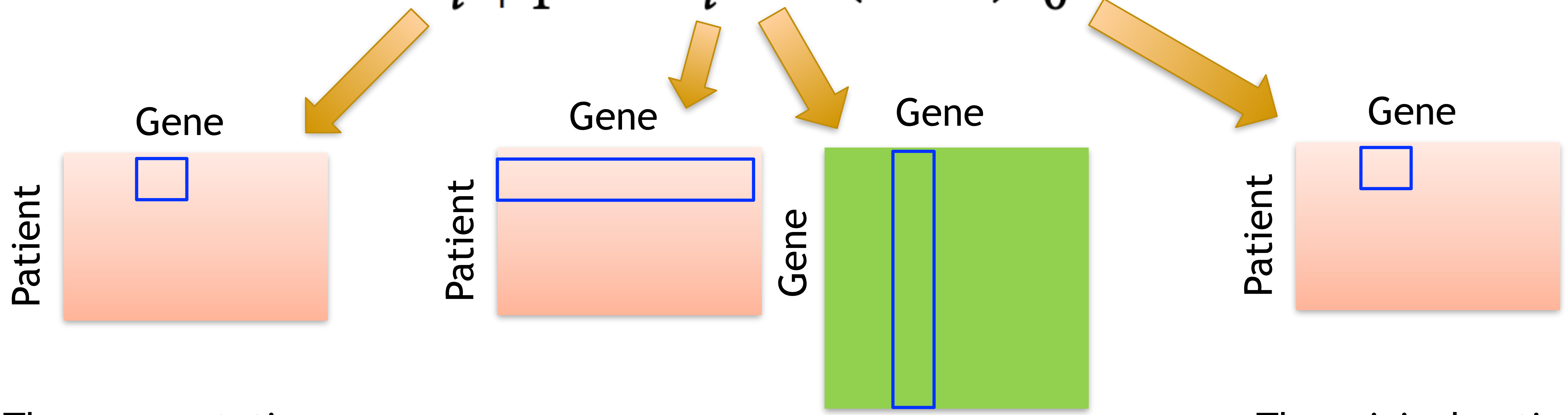
Aggregate consensus matrix (patient × patient)

Source: Network-based stratification of tumor mutations

Algorithm

Performing random walk with restart for each patient

$$F_{t+1} = \alpha F_t A + (1-\alpha) F_0$$



The new mutation status for patient 1, gene j

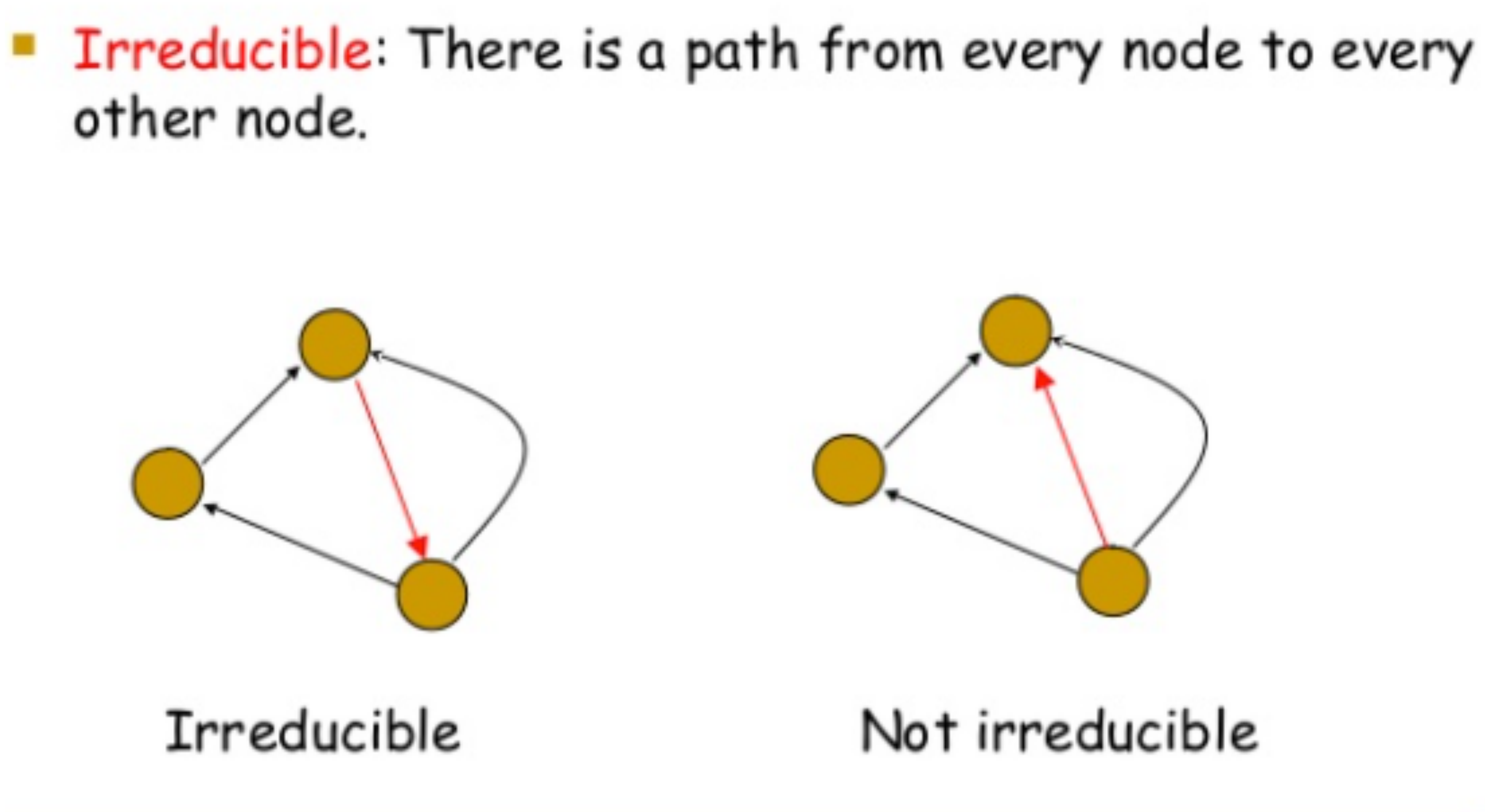
First consider the mutations for all the genes of patient 1

Then consider all the neighbors of gene j

The original patient mutation matrix

Source: Network-based stratification of tumor mutations

Random walk has stationary distribution when the graph is irreducible and aperiodic



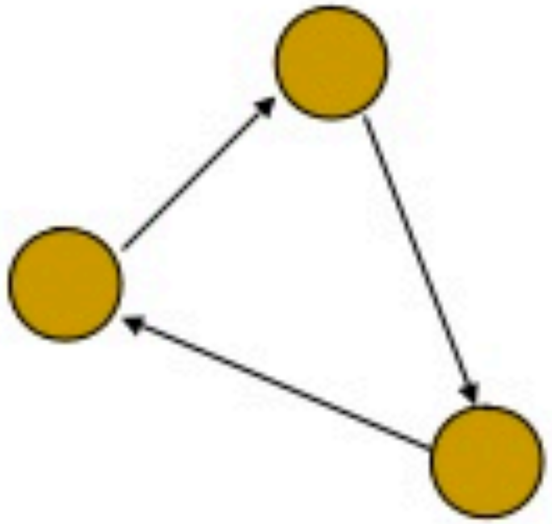
- Irreducible: There is a path from every node to every other node.

Irreducible

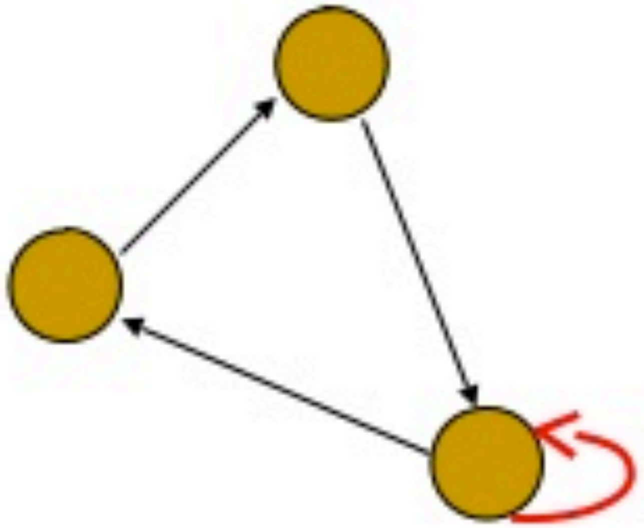
Not irreducible

Source: Random walk on graphs: an overview

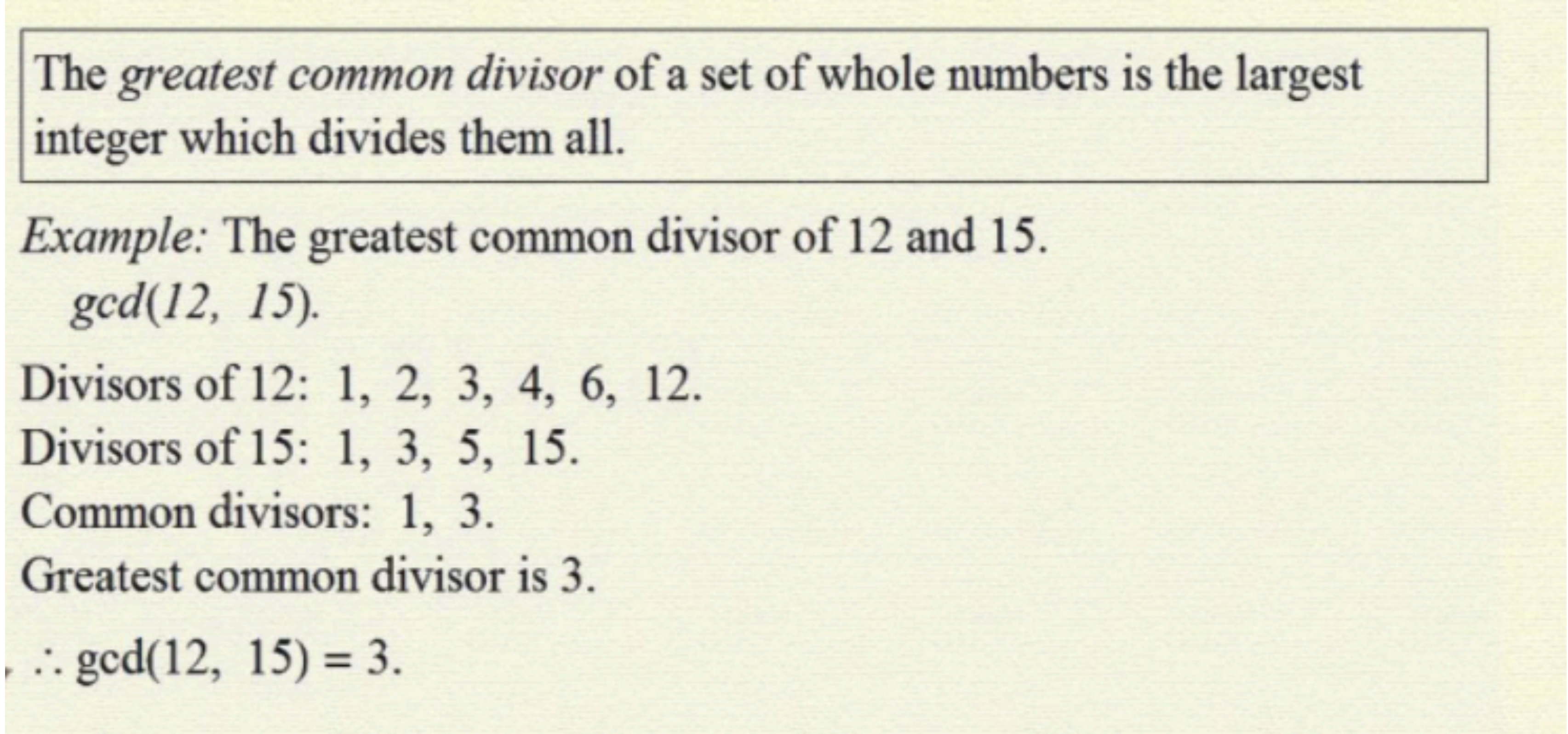
- Aperiodic: The GCD of all cycle lengths is *1*. The GCD is also called period.



Periodicity is 3



Aperiodic



The *greatest common divisor* of a set of whole numbers is the largest integer which divides them all.

Example: The greatest common divisor of 12 and 15.

gcd(*12*, *15*).

Divisors of 12: 1, 2, 3, 4, 6, 12.

Divisors of 15: 1, 3, 5, 15.

Common divisors: 1, 3.

Greatest common divisor is 3.

$\therefore$ gcd(12, 15) = 3.

Source: Random walk on graphs: an overview

Solution: jump to a random node

At each time step, the random surfer has two options

- With prob. β, follow a link at random
- With prob. $1-\beta$, jump to a random page
- Common values for β are in the range 0.8 to 0.9

$$r_j = \sum_{i \to j} \beta \frac{r_i}{d_i} + (1-\beta)\frac{1}{N}$$

$$\begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_j \\ \vdots \\ r_N \end{bmatrix} = \beta \begin{bmatrix} 1/d_1 & , & 0 & , & \dots \\ 1/d_1 & , & 1/d_2 & , & \dots \\ \vdots & & \vdots & & \\ 0 & , & 1/d_2 & , & \dots \\ \vdots & & \vdots & & \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_j \\ \vdots \\ r_N \end{bmatrix} + (1-\beta) \begin{bmatrix} 1/N \\ 1/N \\ \vdots \\ 1/N \\ \vdots \\ 1/N \end{bmatrix}$$

Difference from random walk

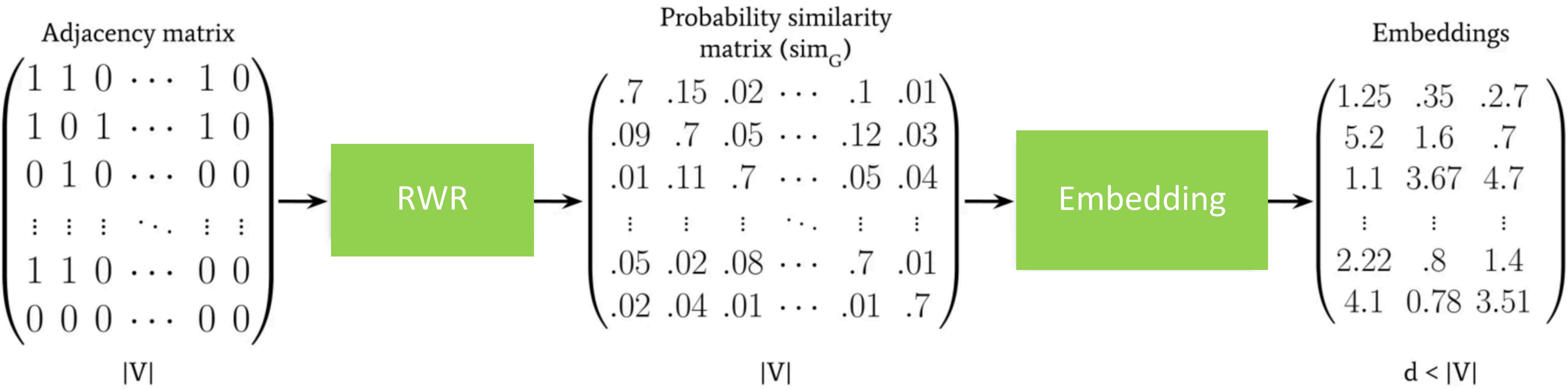
Random walk

$$\begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_j \end{pmatrix} = \beta \begin{pmatrix} 1/d_1 & , \; 0 & , \; \dots \\ 1/d_1 & , \; 1/d_2 & , \; \dots \\ \vdots & \vdots & \\ 0 & , \; 1/d_2 & , \; \dots \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_j \end{pmatrix} + (1-\beta) \begin{pmatrix} 1/N \\ 1/N \\ \vdots \\ 1/N \end{pmatrix}$$

Random walk with restart

$$\begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_j \end{pmatrix} = \beta \begin{pmatrix} 1/d_1 & , \; 0 & , \; \dots \\ 1/d_1 & , \; 1/d_2 & , \; \dots \\ \vdots & \vdots & \\ 0 & , \; 1/d_2 & , \; \dots \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_j \end{pmatrix} + (1-\beta) \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_j \end{pmatrix} \Rightarrow \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}$$

From advanced matrix to random walk probability matrix



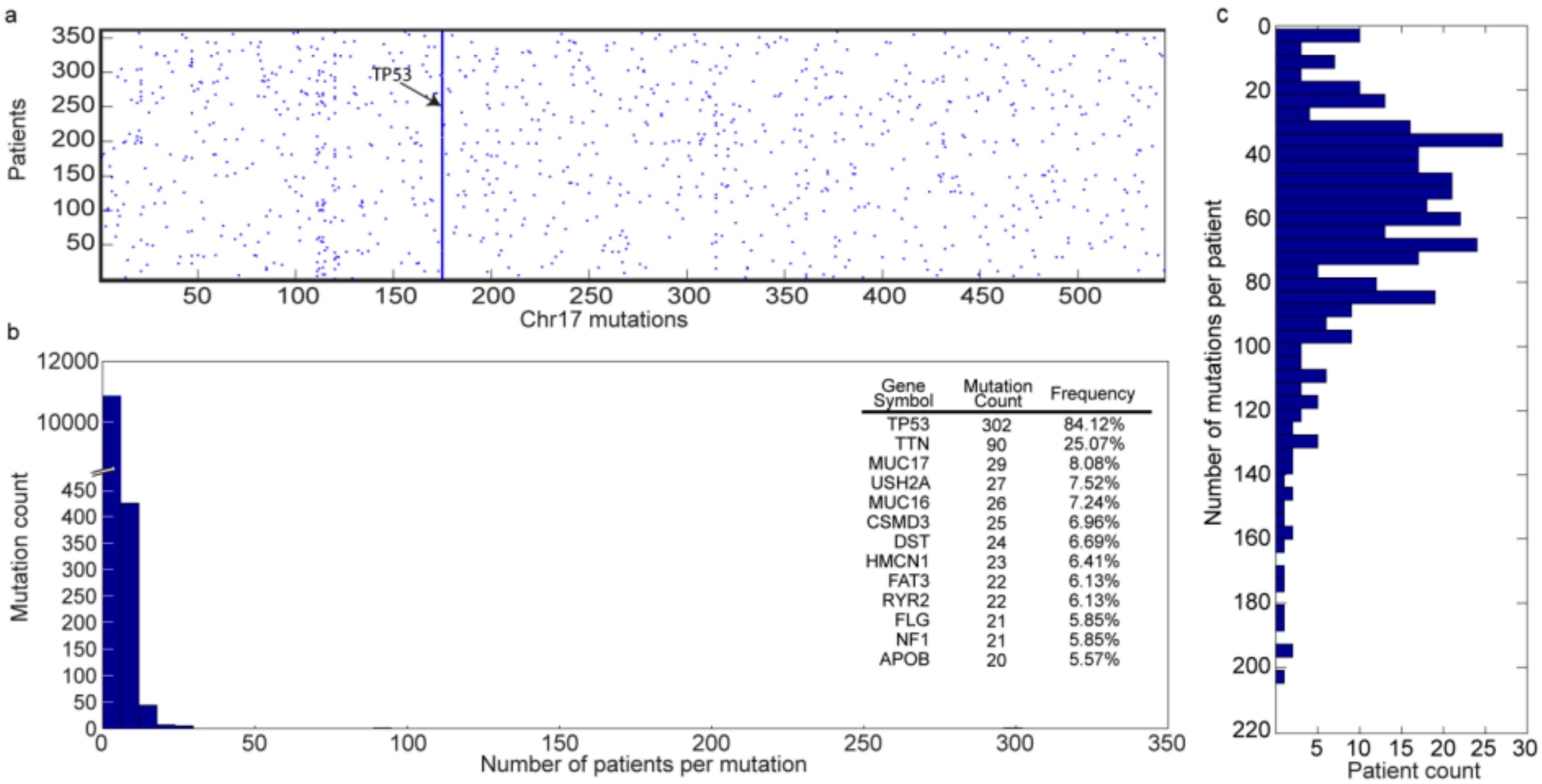
$$\begin{pmatrix} 1 & 1 & 0 & \cdots & 1 & 0 \\ 1 & 0 & 1 & \cdots & 1 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{pmatrix} \rightarrow \boxed{\text{RWR}} \rightarrow \begin{pmatrix} .7 & .15 & .02 & \cdots & .1 & .01 \\ .09 & .7 & .05 & \cdots & .12 & .03 \\ .01 & .11 & .7 & \cdots & .05 & .04 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ .05 & .02 & .08 & \cdots & .7 & .01 \\ .02 & .04 & .01 & \cdots & .01 & .7 \end{pmatrix} \rightarrow \boxed{\text{Embedding}} \rightarrow \begin{pmatrix} 1.25 & .35 & .2.7 \\ 5.2 & 1.6 & .7 \\ 1.1 & 3.67 & 4.7 \\ \vdots & \vdots & \vdots \\ 2.22 & .8 & 1.4 \\ 4.1 & 0.78 & 3.51 \end{pmatrix}$$

Image adapted from MultiVERSE: a multiplex and multiplex-heterogeneous network embedding approach

Somatic mutation profile

- Compare the mutations of tumors
- Sparse

Supplementary Figure 1

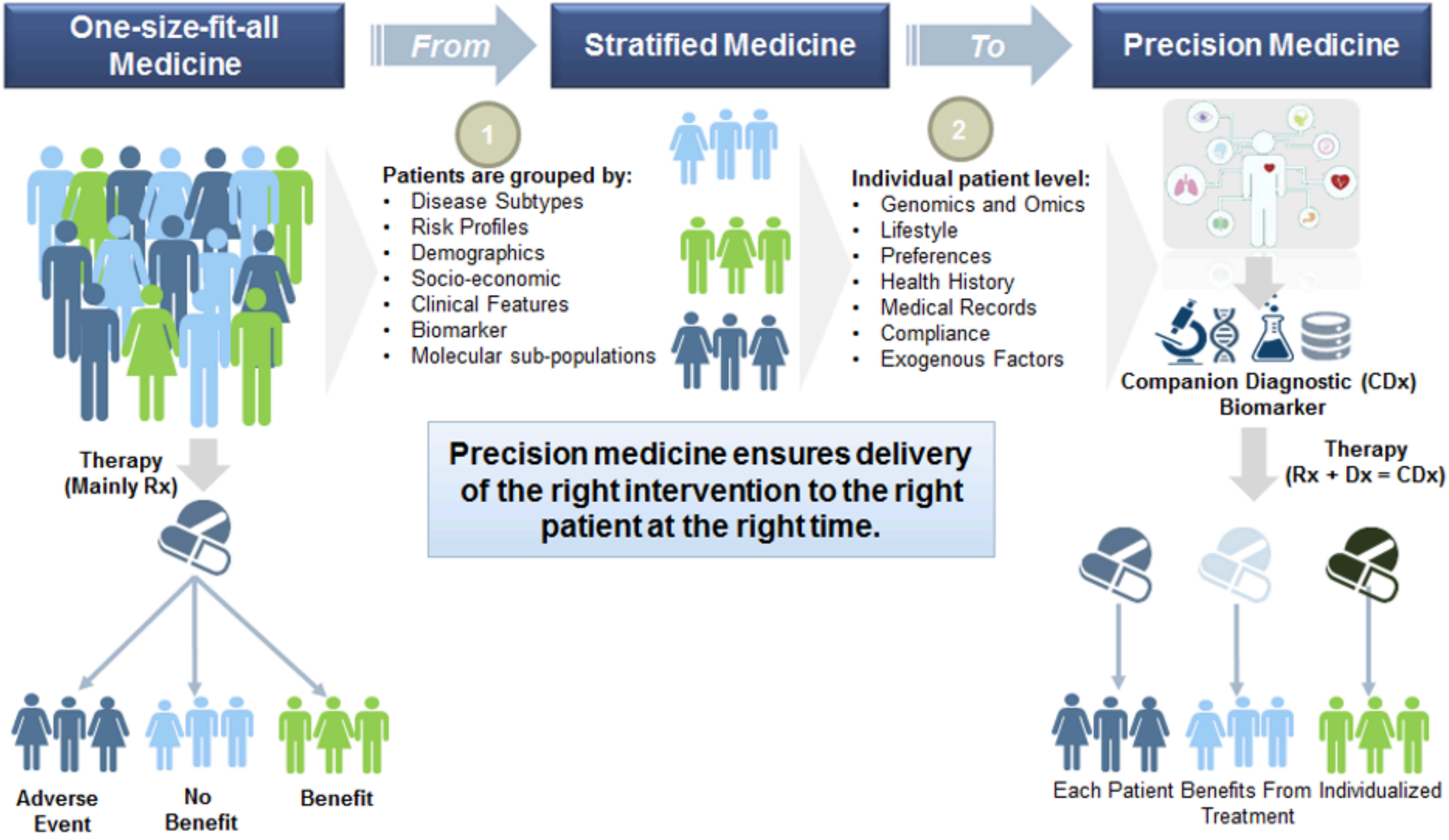


Gene Symbol	Mutation Count	Frequency
TP53	302	84.12%
TTN	90	25.07%
MUC17	29	8.08%
USH2A	27	7.52%
MUC16	26	7.24%
CSMD3	25	6.96%
DST	24	6.69%
HMCN1	23	6.41%
FAT3	22	6.13%
RYR2	22	6.13%
FLG	21	5.85%
NF1	21	5.85%
APOB	20	5.57%

Source: Network-based stratification of tumor mutations

Precision medicine:

the right patient, the right drug, the right time, the right dose



One-size-fit-all Medicine

From

Stratified Medicine

To

Precision Medicine

1

Patients are grouped by:

- Disease Subtypes
- Risk Profiles
- Demographics
- Socio-economic
- Clinical Features
- Biomarker
- Molecular sub-populations

2

Individual patient level:

- Genomics and Omics
- Lifestyle
- Preferences
- Health History
- Medical Records
- Compliance
- Exogenous Factors

Companion Diagnostic (CDx) Biomarker

Therapy (Mainly Rx)

Adverse Event

No Benefit

Benefit

Precision medicine ensures delivery of the right intervention to the right patient at the right time.

Therapy (Rx + Dx = CDx)

Each Patient Benefits From Individualized Treatment

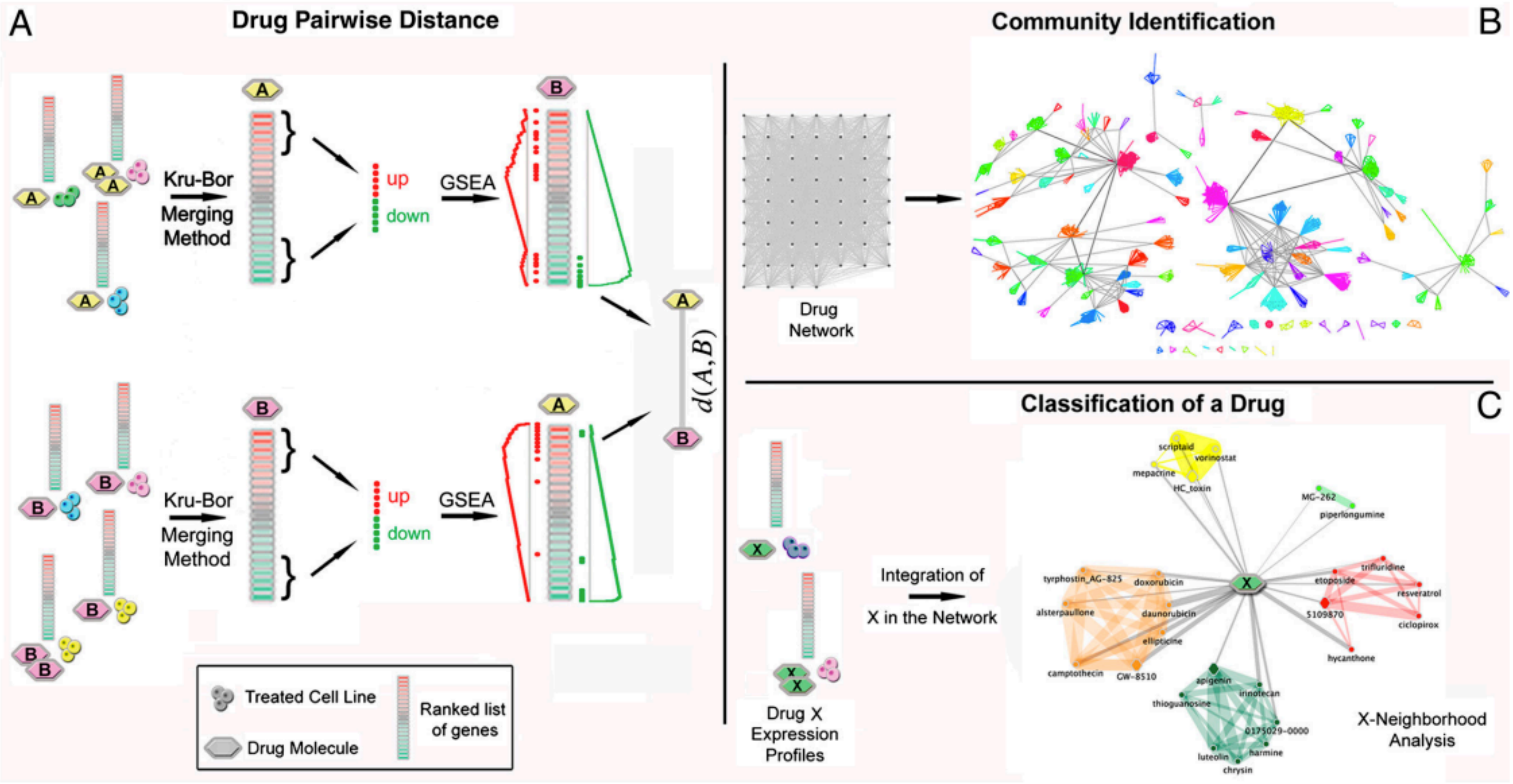
Frost and Sullivan: new paradigm shift in treatment.

We don’t have so many “drugs”

- Discovery new drug?
  - Often not in the scope of precision medicine
  - New patient cannot wait for a new drug
- Drug repurposing
  - Drug A, which is used to treat disease X, is later used to treat disease Y
  - Well-documented side effects and less restriction from FDA
- Drug combination
  - Drug A is not effective. Drug B is not effective. Durg A and B used together is effective.
- Personalized dosage
  - Widely used in clinics. Use genomics data to determine dosage (regression).

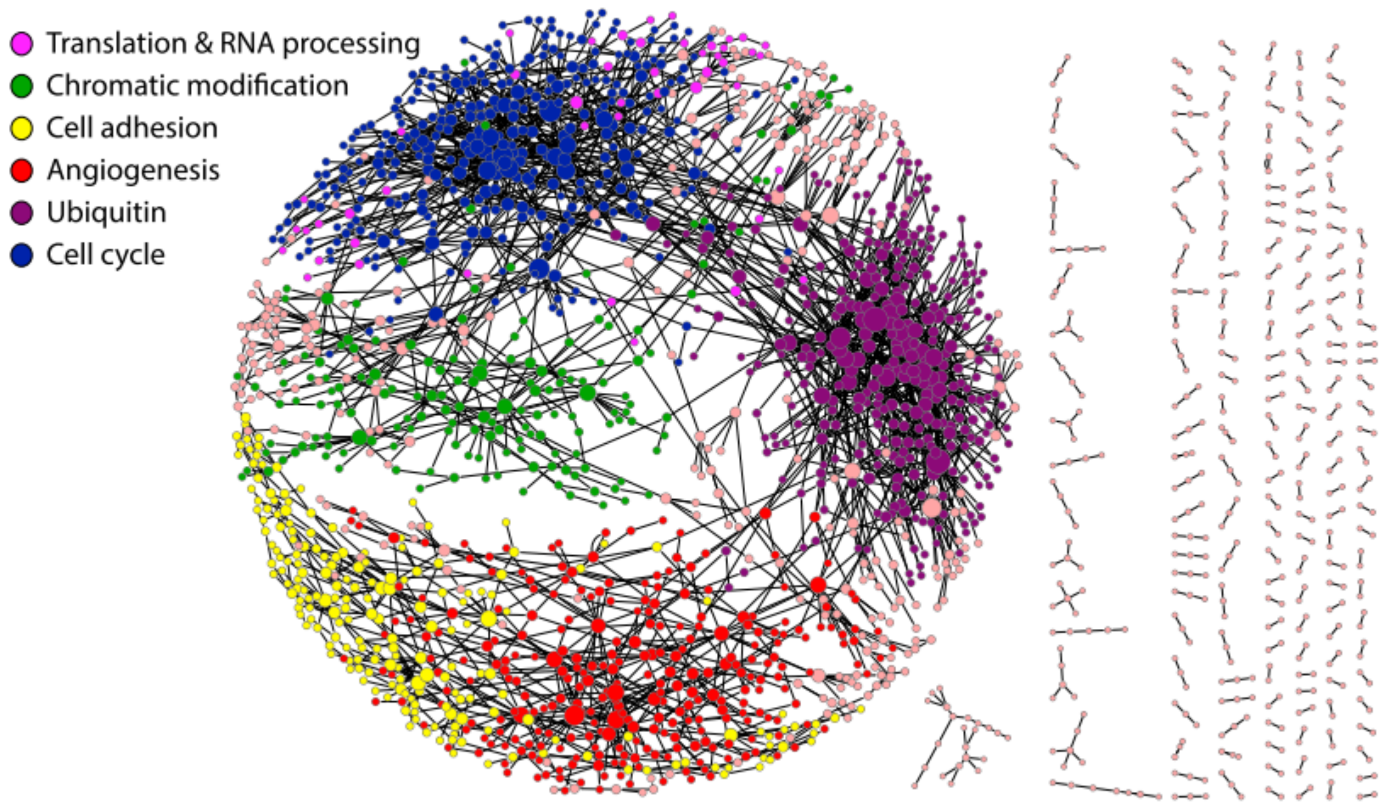
Use gene expression after treatment

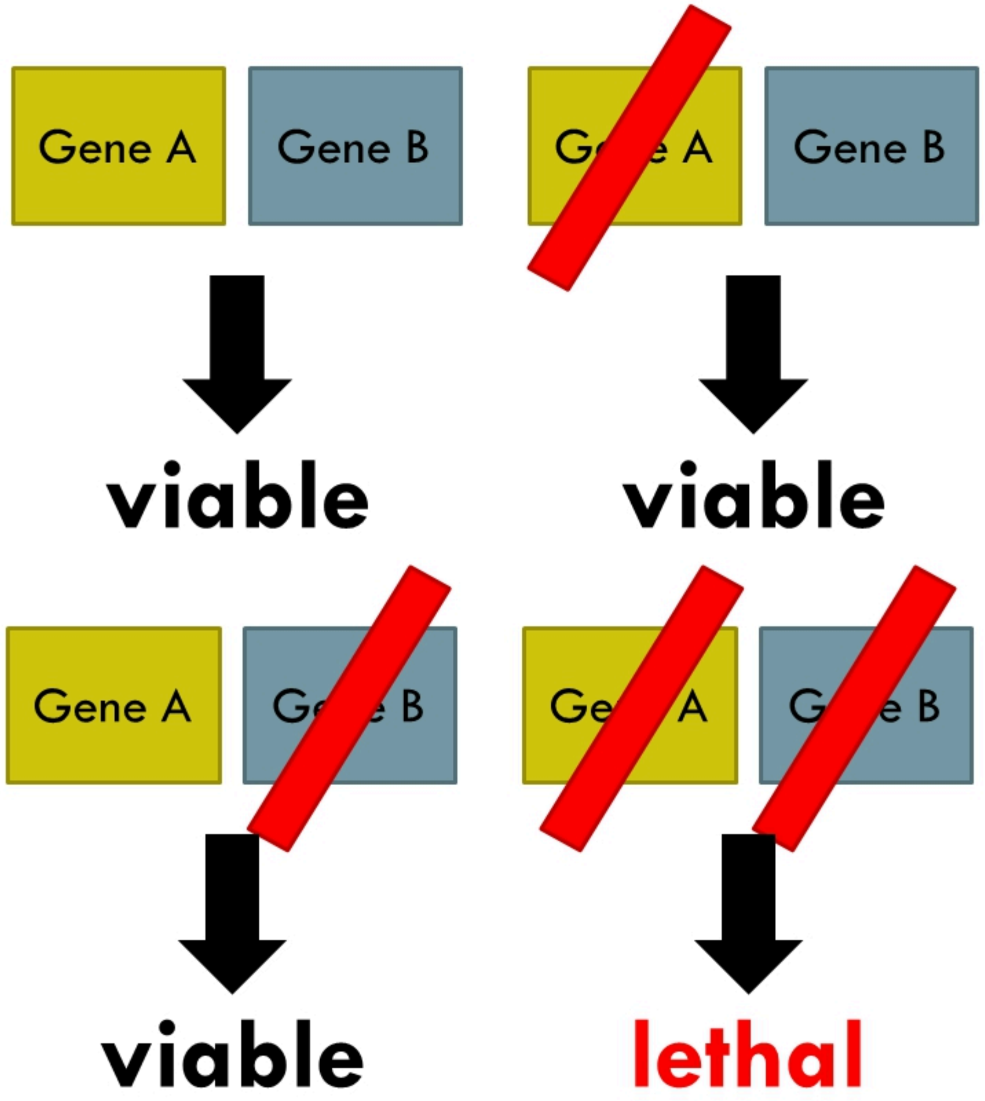
Drugs target on similar proteins or have similar Mode of Actions have similar (after treatment) expression.



Iorio et al. Discovery of drug mode of action and drug repositioning from transcriptional responses

Synthetic lethality: Gene A **OR** Gene B





Gene A Gene B → viable

~~Gene A~~ Gene B → viable

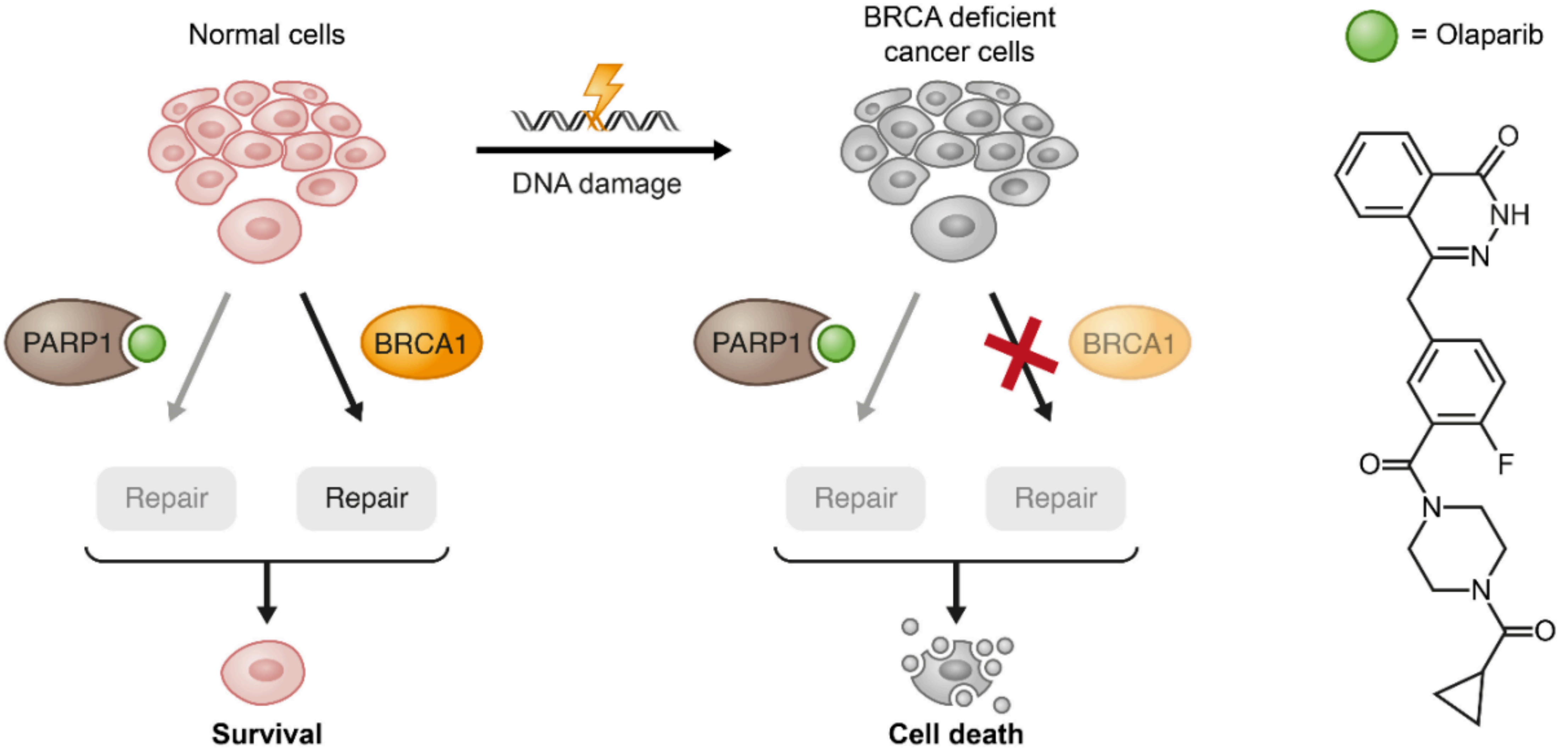
Gene A ~~Gene B~~ → viable

~~Gene A~~ ~~Gene B~~ → lethal

Question: how to leverage SL in drug combination discovery?

Source: wikipedia, Jerby-Arnon et al. Predicting Cancer-Specific Vulnerability via Data-Driven Detection of Synthetic Lethality

Drug treatment based on synthetic lethality



Goal: We want to make normal cells survive and kill cancer cells (BRCA deficient cancer cells)

Prior knowledge: PARP1 (off) + BRCA1 (off) -> cell death

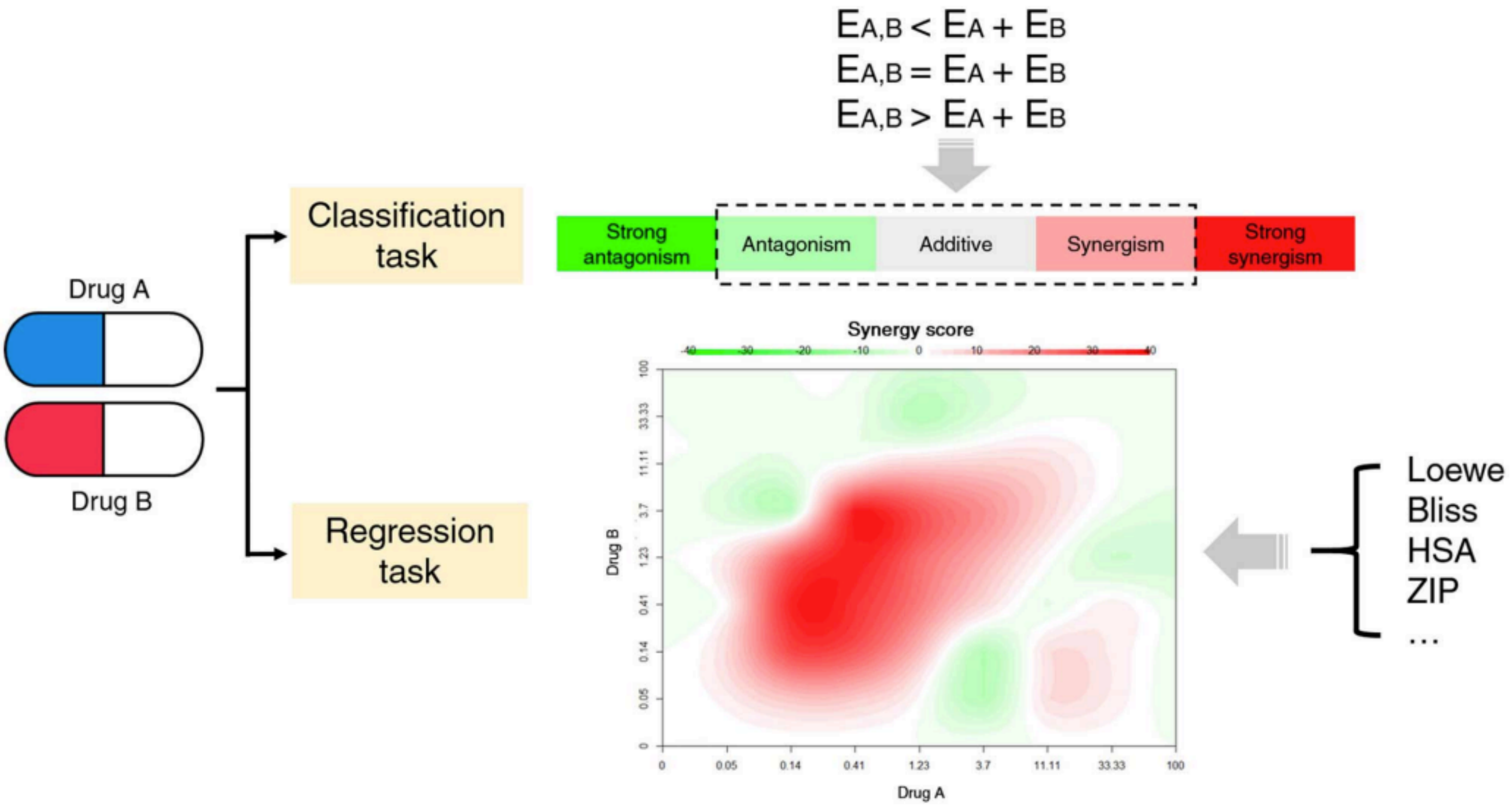
Solution: Turn off PARP1 using Olaparib

Results:

- Normal cells: PARP1 (off) + BRCA1 (on) -> cell survive
- Cancer cells: PARP1 (off) + BRCA1 (off) -> cell death

Gilad et al. Drug Combination in Cancer Treatment—From Cocktails to Conjugated Combinations

Drug combination prediction



$E_{A,B} < E_A + E_B$

$E_{A,B} = E_A + E_B$

$E_{A,B} > E_A + E_B$

Drug A

Drug B

Classification task

Regression task

Strong antagonism | Antagonism | Additive | Synergism | Strong synergism

Loewe

Bliss

HSA

ZIP

…

E(A) is the efficacy of using drug A (e.g., IC50)

Wu et al. Machine learning methods, databases and tools for drug combination prediction