SUNY Korea
The State University of New York
한국뉴욕주립대학교

Stony Brook University

Instructor: Sael Lee

CS549 Spring – Computational Biology

1. INTRODUCTION TO COMPUTATIONAL BIOLOGY

Resources used: Lecture slides from Steven Skiena’s Computational Biology class and Daisuke Kihara’s Protein Bioinformatics class

WHY COMPUTATIONAL BIOLOGY?

- Computational biology is particularly exciting today because:
  - the problems are large enough to motivate efficient algorithms,
  - the problems are accessible, fresh and interesting,
  - biology is increasing becoming a computational science
- Computational biology is increasing of interest in both life science and computational science departments.
- Source of complex questions and real-life data.
  - Many problem ideas go from biology to CS: e.g. fragment assembly, sequence analysis, algorithms for phylogenic trees.
  - Many problem ideas go from CS to biology: e.g. sequencing by hybridization, DNA computing.

COMPUTER SCIENTIST VS BIOLOGIST

- Similarity:
  - There are many different types of life scientists (biologists, ecologists, medical doctors, etc.), just as there are many different types of computational scientists (algorists, software engineers, statisticians, etc.).
- Many cultural differences
  - ***Nothing is ever completely true or false in biology***, *where everything is either true or false in computer science / mathematics.*
  - Biologists are comfortable with the idea that all data has errors; computer scientists are not.
  - Biologists strive to understand the very complicated, very messy natural world; computer scientists seek to build their own clean and organized virtual worlds.

* Information extracted from Steve Skiena's slide. Thanks, Steve. ☺

- Biologists are *data driven;* while computer scientists are *algorithm driven*. Although nowadays cs are becoming more data driven.
- Biotechnology/drug companies are largely **science driven**, while the computer industry is **more engineering/marketing driven.**
- The Platonic ideal of a biologist runs a big laboratories with many people. The Platonic ideal of a computer scientists is a hacker in garage.
- Biologists are much more obsessed with being the **first to discover something**; computer scientists **invent** more than discover.
- Biologists can get/spend **infinitely more research money** than computational scientists.
- Biologists seek to publish in prestigious **journals** like *Science and Nature.* Computer scientists seek to publish in prestigious refereed **conference proceedings.**
  - One consequence is life science journals get refereed faster than computational science journals.

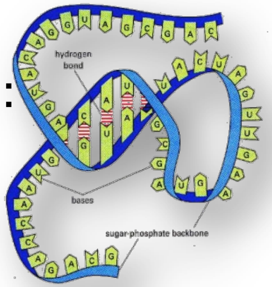
INFORMATION CONTENT IN BIOLOGY

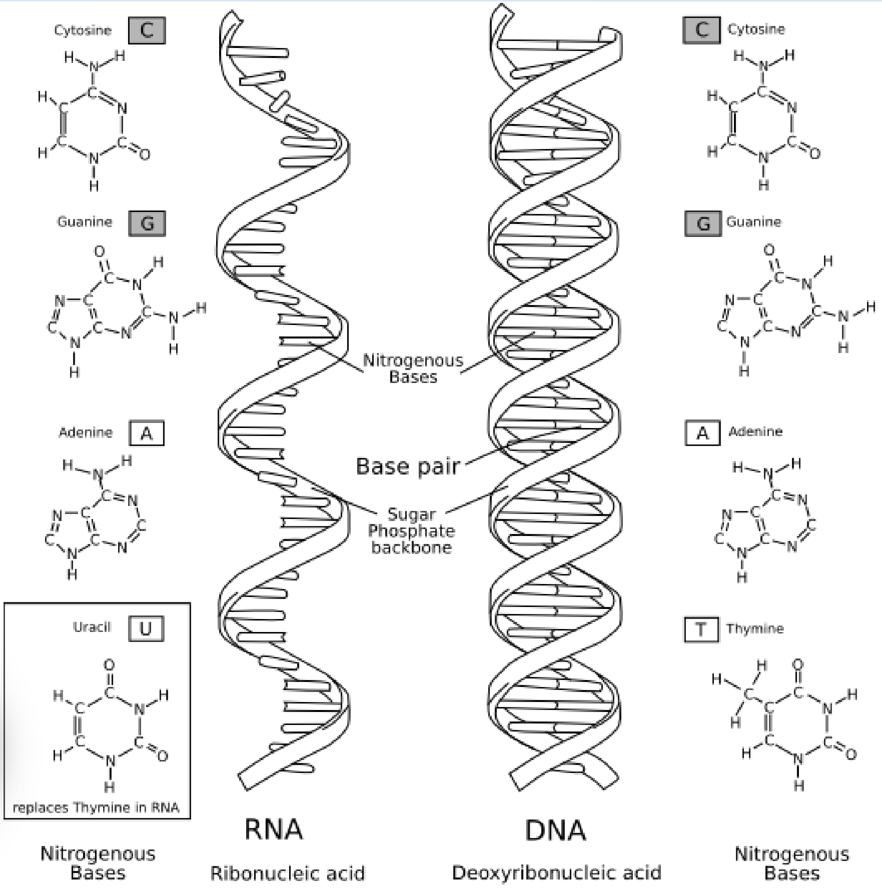
DNA

- DNA sequences can be thought of as strings of bases on a four-letter alphabet, {A,C,G,T}, called nucleic acids.
- Binding: A=T; C-G
- Stable structural form : **double helix**

RNA

- RNA sequences can also be thought of as strings of bases on a four-letter alphabet, {A,C,G,U}.
- Binding: A=U; C-G
- Stable structural form:





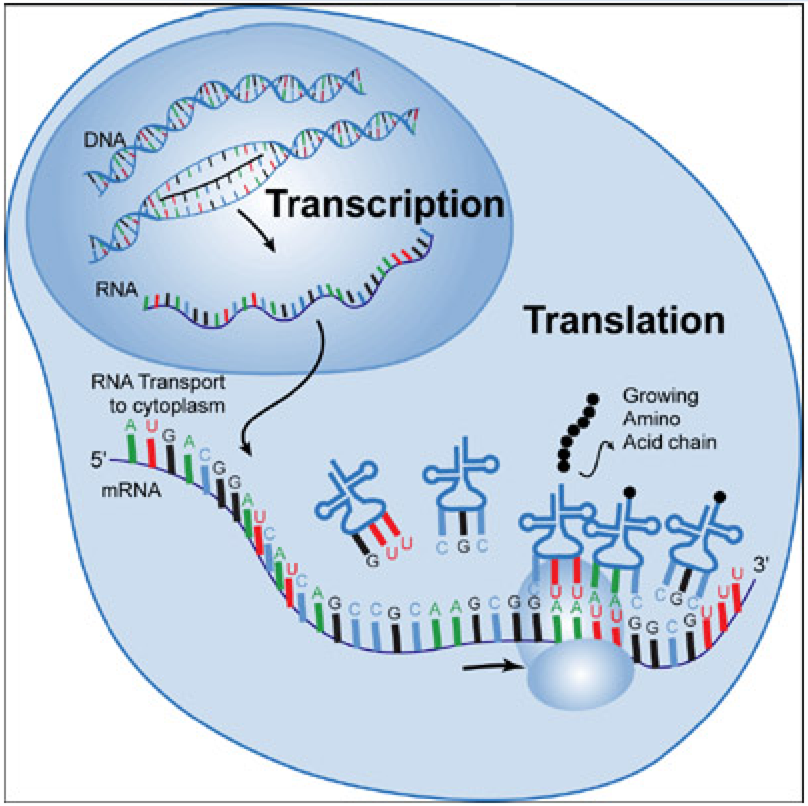
http://en.wikipedia.org/wiki/Nucleic_acids

Proteins

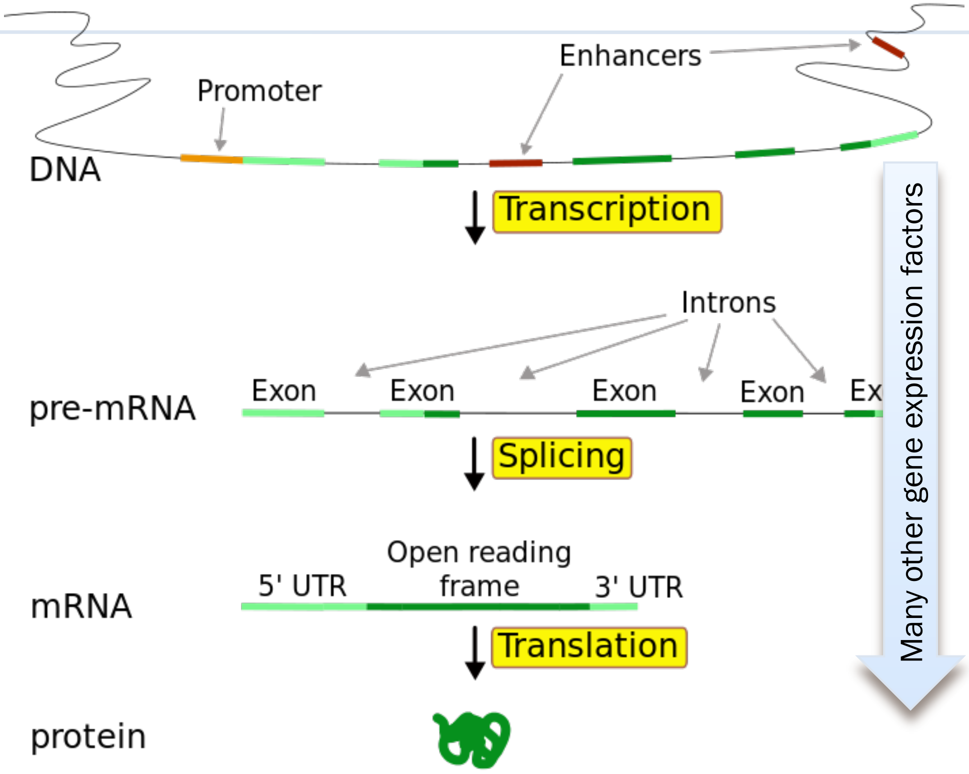
- Proteins sequence can be thought of as string of 21-letter alphabet
- Binding: covalent bonding, van der Waals force, hydrophobicity, etc.
- Stable structural form:

A LQNHTFLHTVYCQDGSPSVGLSEA …
DIFSCIVTHEPDRYTAIAYWVPRNALPS

CENTRAL DOGMA OF BIOLOGY



http://www.tokresource.org/tok_classes/biobiobio/biomenu/transcription_translation/transcription_2.jpg



youtube: From RNA to Protein Synthesis [3min]

Common Abbreviations

- DNA: Deoxyribonucleic acid
- RNA: Ribonucleic acid
- mRNA: messenger RNA
- tRNA: transfer RNA
- rRNA: ribosomal RNA
- siRNA: Small interfering RNA

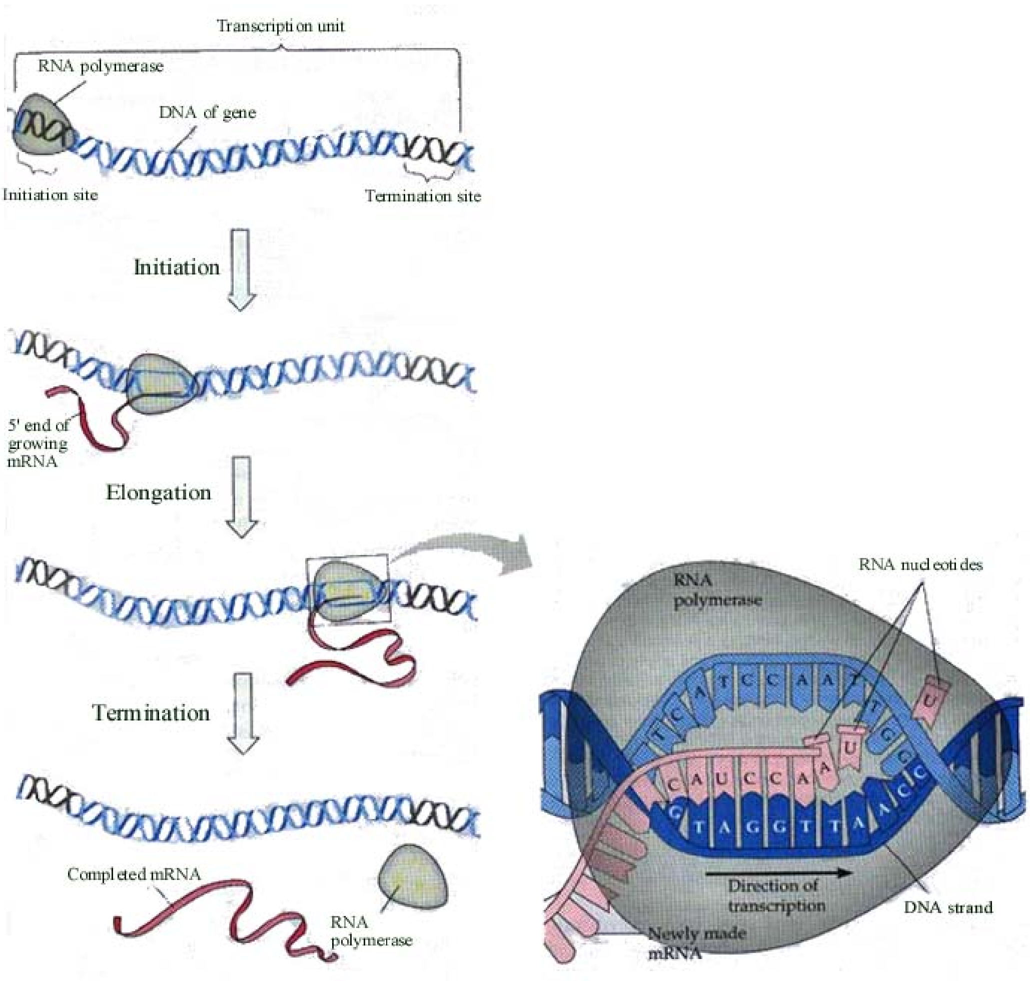
TRANSCRIPTION PROCESS

RNA polymerase ‘unzips’ the DNA from **initiation site**.

Elongation: create a RNA strand by coping DNA strand

Stops at **termination site**

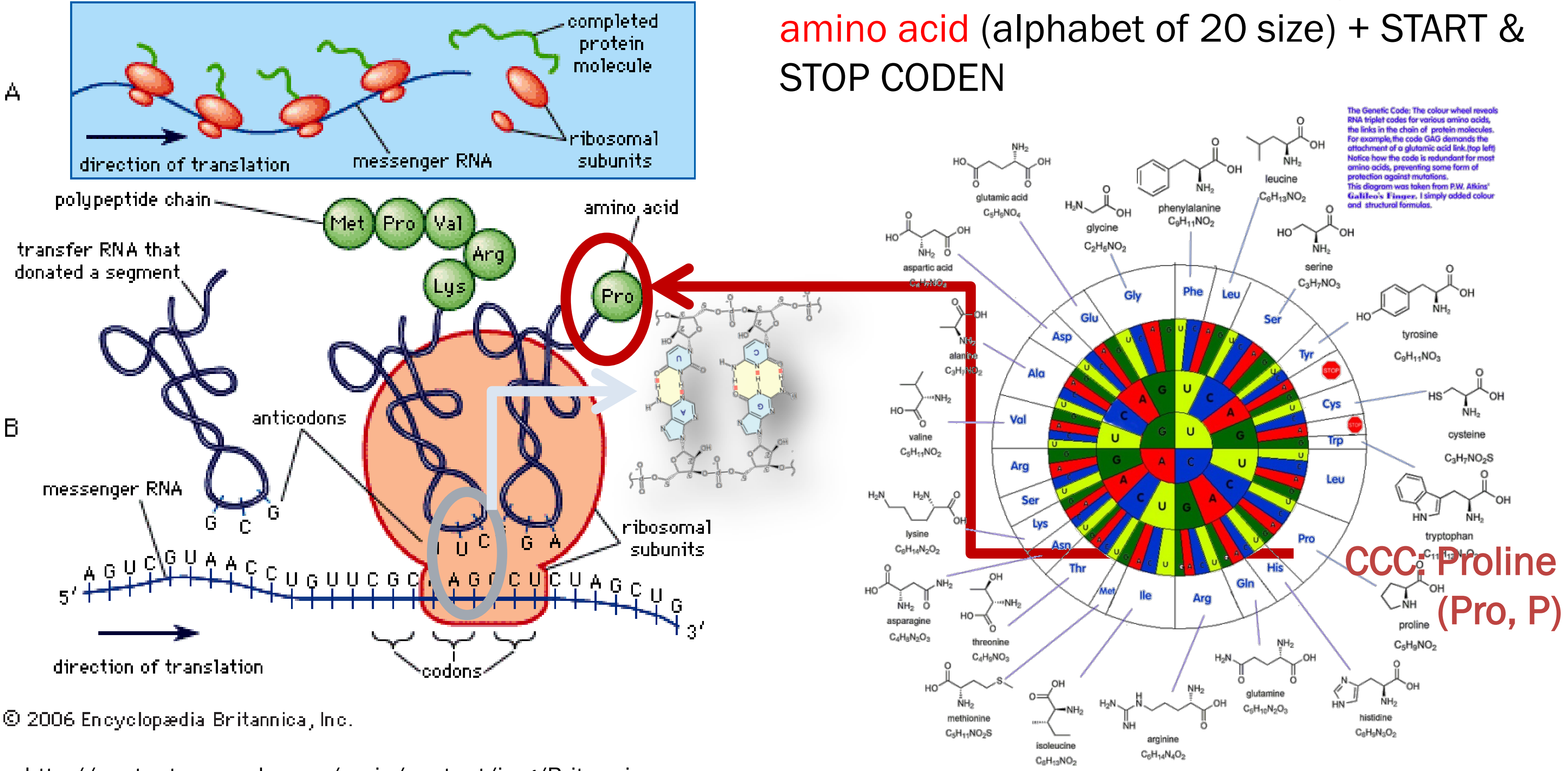
Posttranslational modification



http://biology12-lum.wikispaces.com/file/view/transcription,nbk.jpg/177394793/transcription,nbk.jpg

TRANSLATION PROCESS

Codon: Three nucleic acid coding one of 20 amino acid (alphabet of 20 size) + START & STOP CODEN



© 2006 Encyclopædia Britannica, Inc.

The Genetic Code: The colour wheel reveals RNA triplet codes for various amino acids, the links in the chain of protein molecules. For example,the code GAG demands the attachment of a glutamic acid link.(top left) Notice how the code is redundant for most amino acids, preventing some form of protection against mutations. This diagram was taken from P.W. Atkins' **Galileo's Finger.** I simply added colour and structural formulas.

CCC: Proline (Pro, P)

http://content.answcdn.com/main/content/img/BritannicaConcise/images/780.gif

Start codon: AUG (also Methionine (Met, M))
Stop codon: UAA, UAG, UGA

Instructor: Sael Lee

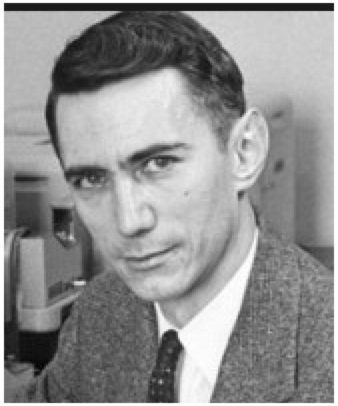
CS549 Spring – Computational Biology

LECTURE 2:
INFORMATION CONTENT IN BIOLOGY & DNA BINDING

Resources from:
1) Lecture Notes of Natasha Devroye devroye@ece.uic.edu http://www.ece.uic.edu/~devroye
2) F. Fabris “Shannon Information Theory and Molecular Biology” *JIM*, vol.12, n.1, february 2009, pp. 41-87.
3) T Cover & J Thomas “Elements of Information Theory 2nd ed.” 2006

THE MATHEMATICS THEORY OF COMMUNICATION

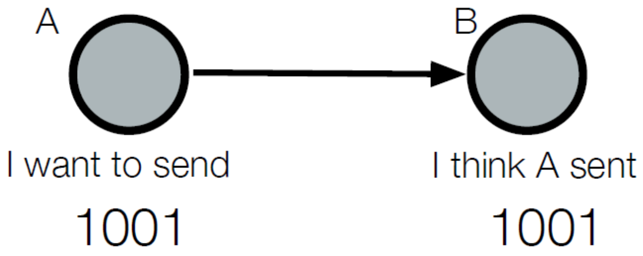
Claude E. Shannon

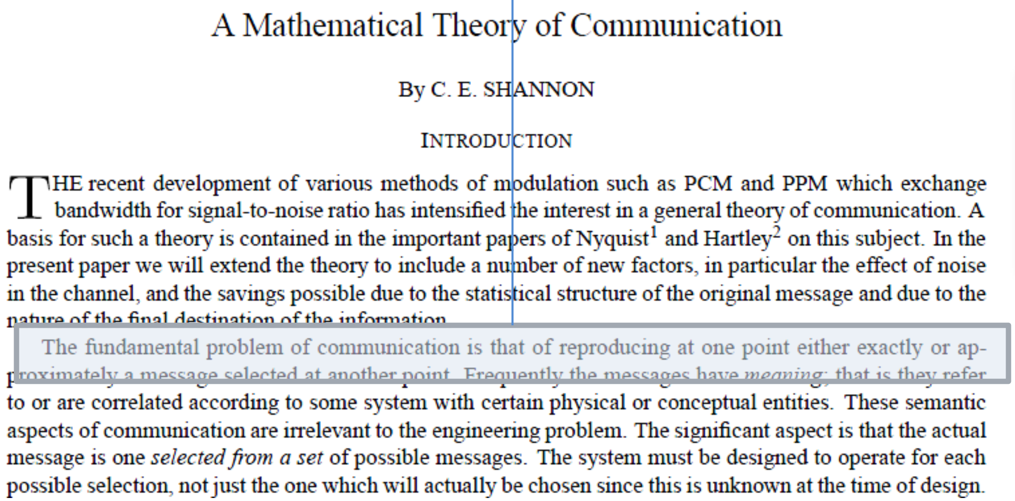


Reprinted with corrections from *The Bell System Technical Journal,*
Vol. 27, pp. 379–423, 623–656, July, October, 1948.

"The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point."

C.E. Shannon, 1948





A Mathematical Theory of Communication

By C. E. SHANNON

INTRODUCTION

THE recent development of various methods of modulation such as PCM and PPM which exchange bandwidth for signal-to-noise ratio has intensified the interest in a general theory of communication. A basis for such a theory is contained in the important papers of Nyquist[1] and Hartley[2] on this subject. In the present paper we will extend the theory to include a number of new factors, in particular the effect of noise in the channel, and the savings possible due to the statistical structure of the original message and due to the nature of the final destination of the information.

The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point. Frequently the messages have *meaning*; that is they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem. The significant aspect is that the actual message is one *selected from a set* of possible messages. The system must be designed to operate for each possible selection, not just the one which will actually be chosen since this is unknown at the time of design.

Introduced a new field: Information Theory

SHANNON'S FINDINGS

- Source Coding Problem:
  - Source = random variables
  - Ultimate **data compression** limit is the source's entropy H
- Channel Coding Problem:
  - Channel = conditional distributions
  - **Ultimate transmission rate** is the channel capacity C
- Relationship between input and output
  - Mutual Information
- Reliable communication possible ↔ H<C

GENERAL COMMUNICATION SYSTEM

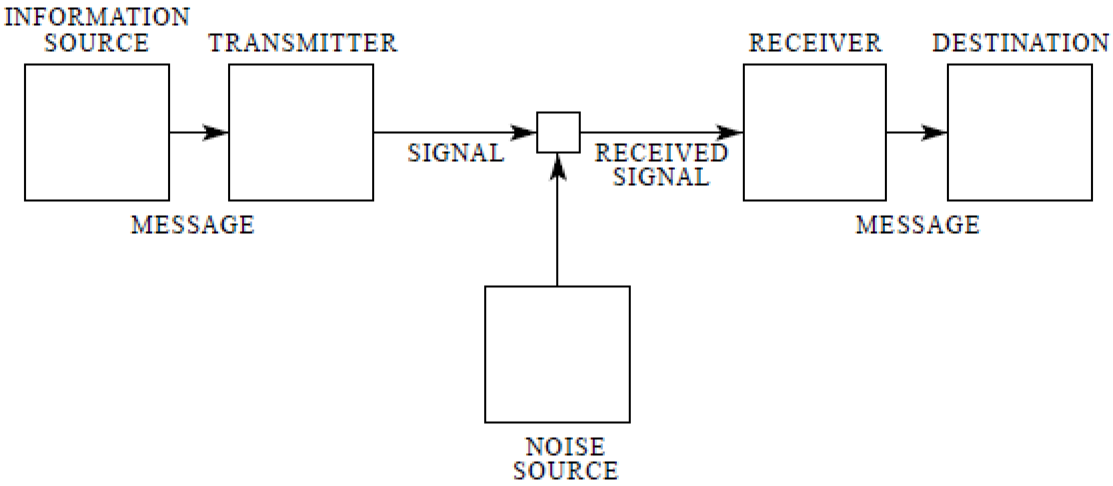
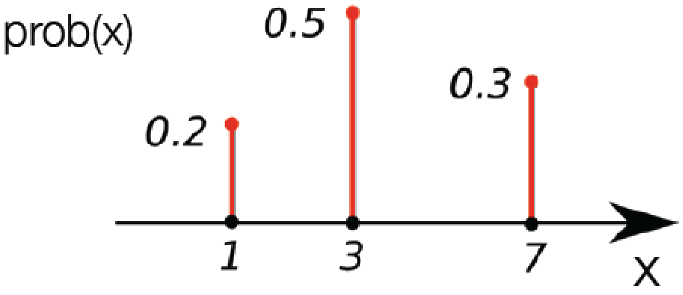


Fig. 1—Schematic diagram of a general communication system.

- ***Information source***: "produces a message or sequence of messages to be communicated to the receiving terminal"
- ***Transmitter:*** "operates on the message in some way to produce a signal suitable for transmission over the channel"
- ***Channel :*** "the medium used to transmit the signal from transmitter to receiver"
- ***Receiver:*** *"performs the inverse operation of that done by the transmitter reconstructing the message from the signal"*
- ***Destination:*** "person (or thing) for whom the message is intended"

SHANNON'S ENTROPY

- Entropy is the measure of **average uncertainty** in the random variable
- Entropy is the **average number of bits** needed to describe the random variable
- Entropy is a lower bound on the **average length of the shortest description** of the random variable
- Entropy of a deterministic value is 0



What is the **entropy** of a random variable X with distribution p(x)?

$$H(X) = -\sum_{x} p(x) log_2 (p(x))$$

Entropy measured in bits

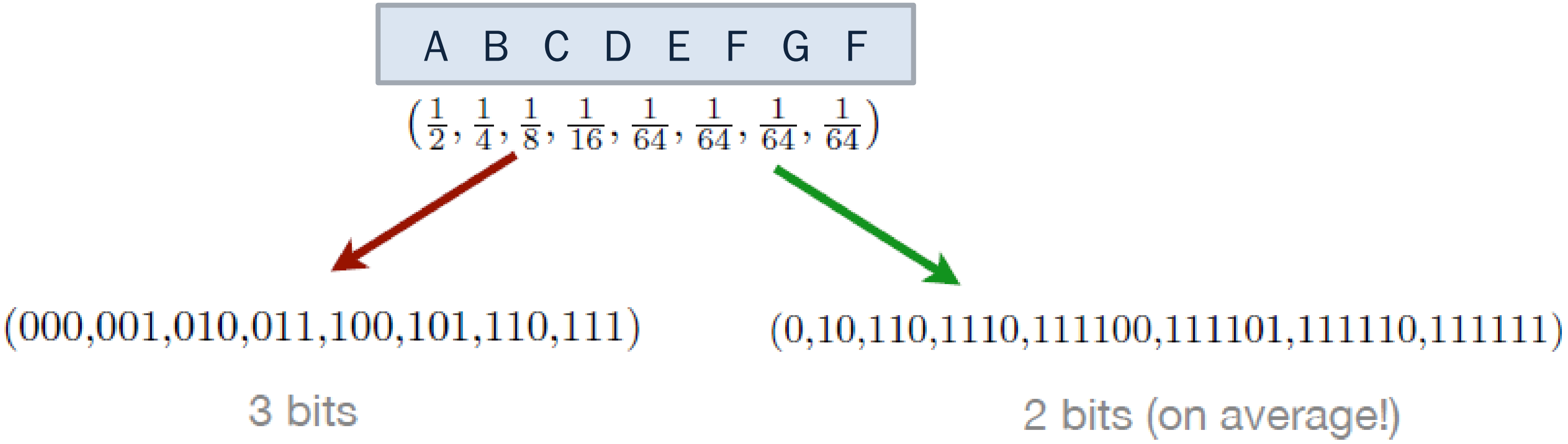
ENTROPY OF A NON-UNIFORM DISTRIBUTION

- Suppose X represents the outcome of a horse race with 8 horses, which win with probabilities $\left(\frac{1}{2},\frac{1}{4},\frac{1}{8},\frac{1}{16},\frac{1}{64},\frac{1}{64},\frac{1}{64},\frac{1}{64}\right)$

$$H(X) = -\frac{1}{2}log_2\left(\frac{1}{2}\right) - \frac{1}{4}log_2\left(\frac{1}{4}\right) - \frac{1}{8}log_2\left(\frac{1}{8}\right) - \frac{1}{16}log_2\left(\frac{1}{16}\right) - 4\frac{1}{64}log_2\left(\frac{1}{64}\right)$$

$$= \frac{1}{2} + \frac{2}{4} + \frac{3}{8} + \frac{4}{16} + 4\frac{6}{64} = 2\text{(bits)}$$

- 8 outcomes, 3 bits? But on average can represent with 2 bits.



A B C D E F G F

$\left(\frac{1}{2},\frac{1}{4},\frac{1}{8},\frac{1}{16},\frac{1}{64},\frac{1}{64},\frac{1}{64},\frac{1}{64}\right)$

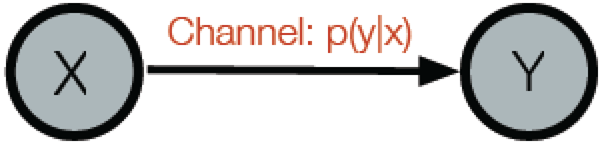
(000,001,010,011,100,101,110,111)

3 bits

(0,10,110,1110,111100,111101,111110,111111)

2 bits (on average!)

MUTUAL INFORMATION BETWEEN 2 RANDOM VARIABLES:



- Mutual Information I(X;Y) is the **reduction** in the uncertainty about X due to knowledge of Y
- if X, Y are independent I(X;Y) = 0
- if X=Y then I(X;Y) = H(X)
- I(X;Y) is non-negative

$$\mathrm{I}(X;Y) = -\sum_x p(x,y) log_2\left(\frac{p(x,y)}{p(x)p(y)}\right)$$

THE DNA-TO-PROTEIN BIO-MOLECULAR CHANNEL

- Central Dogma of Molecular Biology states there is a flow of “biologic information” from DNA towards proteins:
- -> that the DNA carries information that, after transcription and translation, drives the synthesis of the proteins.

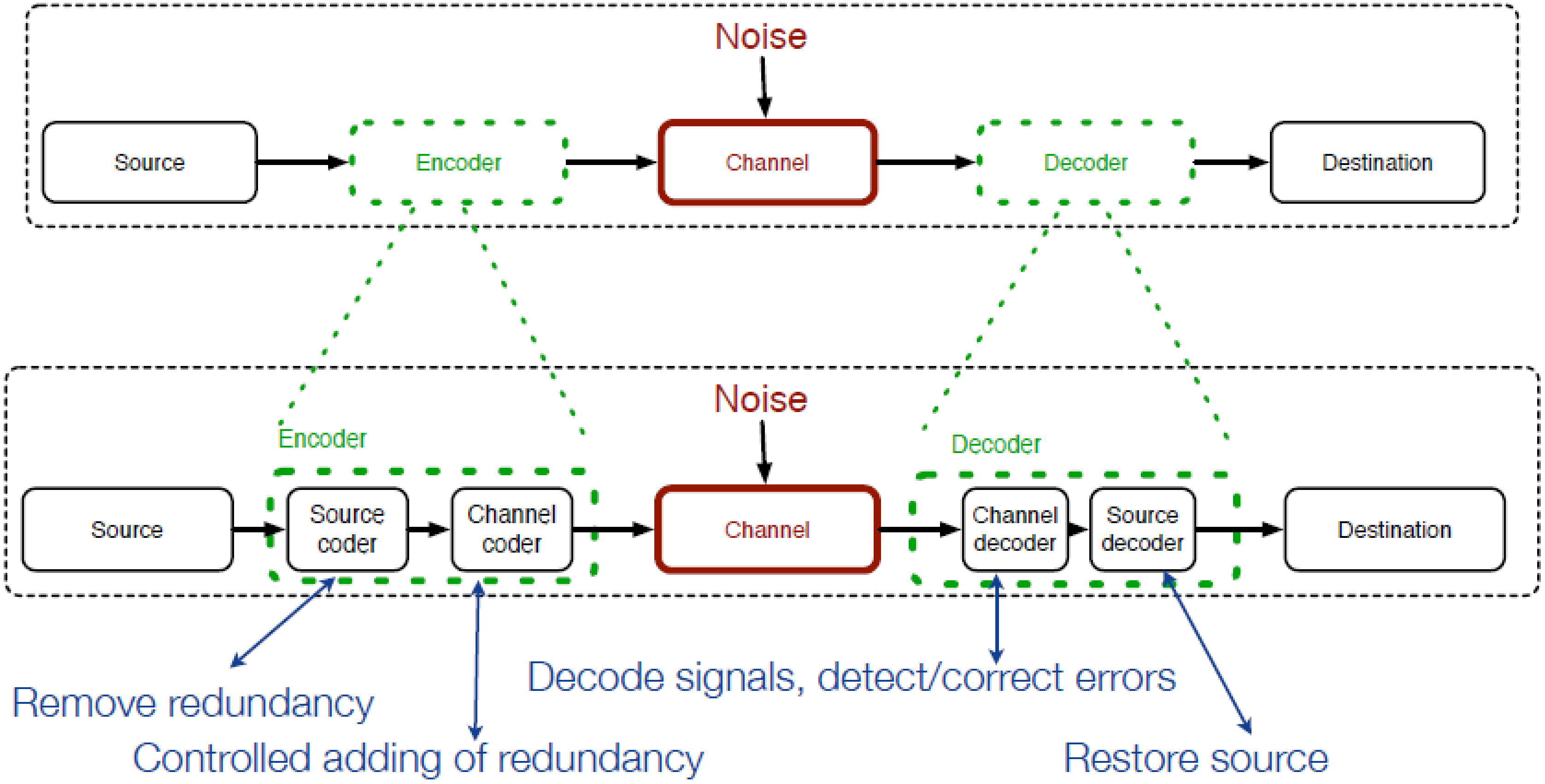
APPEALING METAPHOR

the flow of information that starts from DNA and reaches the proteins, in the biological communication system outlined by the Central Dogma, is analogous to the flow of information that starts from the sender and reaches the receiver (at the other side of the channel) in the communication system.

- DNA: interpreted as a sequence based on a 4-letters alphabet,
  - a sequence of nucleotides - Adenine, Thymine, Cytosine and Guanine (A, T,C,G),
- Protein: interpreted as a 20-letters alphabet sequence.
  - a sequence based on 20 amino acids (Metionine, Serine, Threonine etc.),

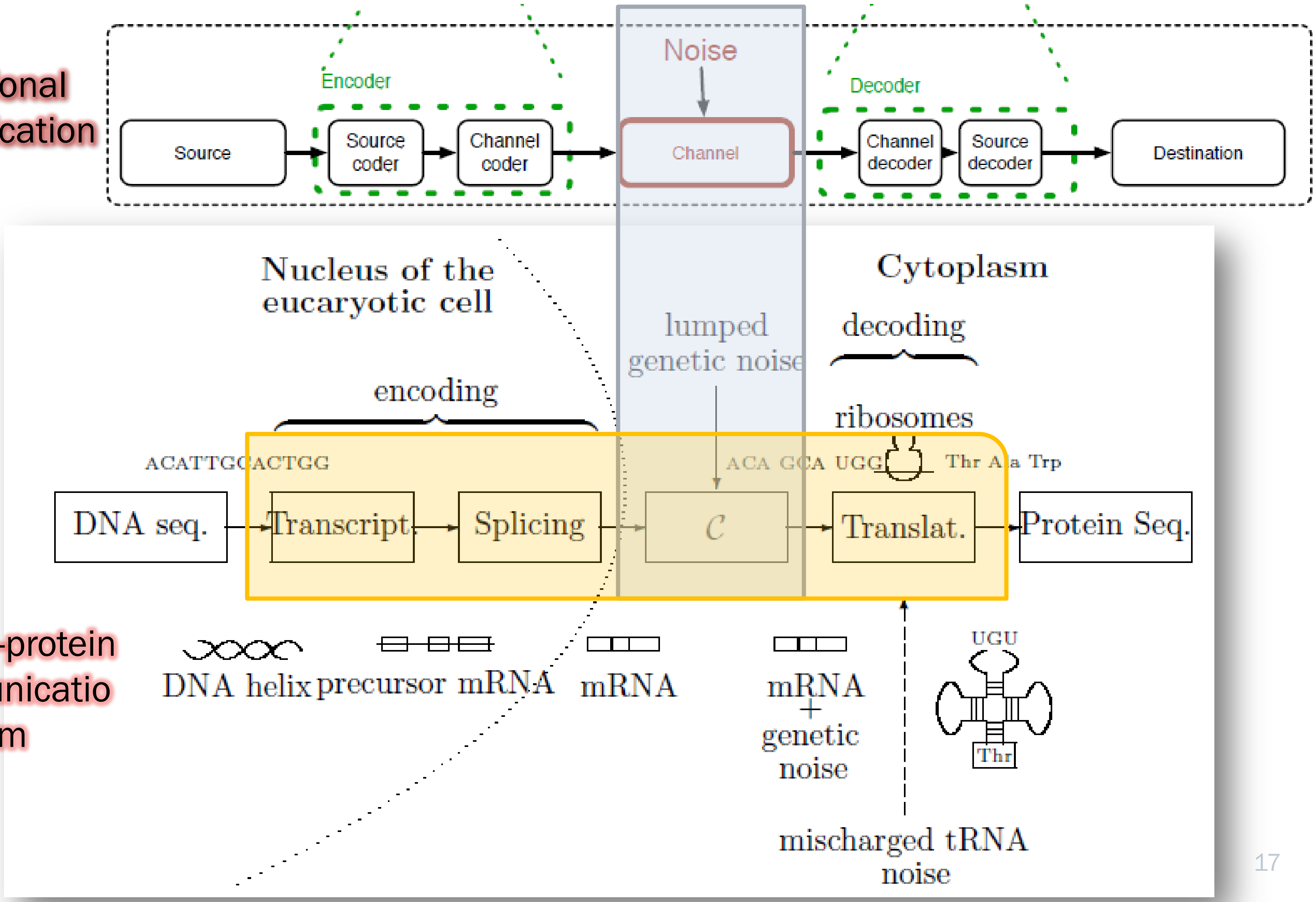
This approach seems to offer the opportunity of using Information Theory as a tool to build a model of biological information transmission and correction.

GENERIC COMMUNICATION BLOCK DIAGRAM



THE DNA-TO-PROTEIN BIO-MOLECULAR CHANNEL

Shannon unidirectional communication system



DNA-to-protein communicatio n system

APPEALING BUT HAS LIMITS

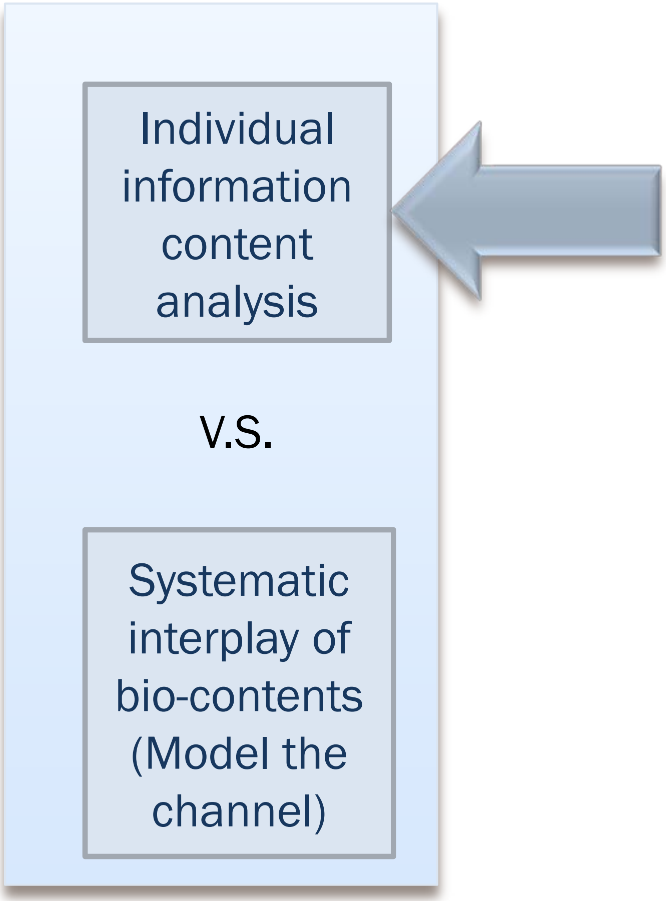
- Biology is much complex compared to general communication system.
  - Systematically complex: Feedback loops, granularity, multiple players
  - Model incomplete: Many biological relations yet to be learned

DNA / RNA / PROTEINS; GENE

Single “word” in genome

“A gene is a molecular unit of heredity of a living organism. It is widely accepted by the scientific community as a name given to some stretches of DNA and RNA that code for a polypeptide (protein) or for an RNA chain that has a function in the organism.”
[http://en.wikipedia.org/wiki/Gene]

* The concept of genes preceded the knowledge of DNA. So, there is some controversies in linking genes to DNA.

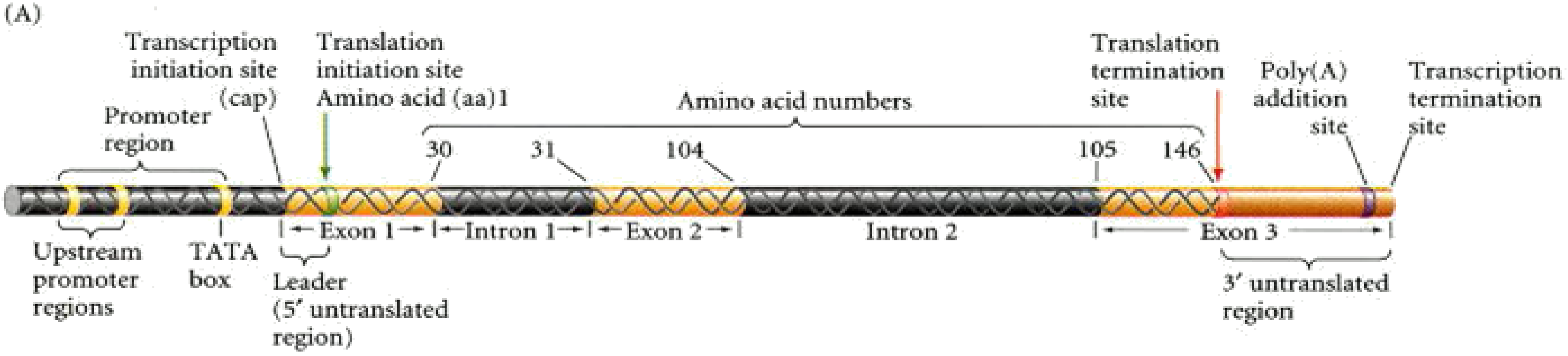


Individual information content analysis

V.S.

Systematic interplay of bio-contents (Model the channel)

ANATOMY OF THE (EUKARYOTIC) GENE


(A)
Transcription initiation site (cap)
Promoter region
Translation initiation site Amino acid (aa)1
Amino acid numbers
30
31
104
105
146
Translation termination site
Poly(A) addition site
Transcription termination site
Upstream promoter regions
TATA box
Leader (5′ untranslated region)
Exon 1
Intron 1
Exon 2
Intron 2
Exon 3
3′ untranslated region

Promoters Exons Introns

- **Promoters** are the sites where RNA polymerase binds to the DNA to initiate transcription.
- **Enhancer** is a DNA sequence that can activate the utilization of a promoter, controlling the efficiency and rate of transcription from that particular promoter. Located geometrically close to the promoter and gene but may not be close in sequence.
- **Exons**—are intervening sequences
- **Introns**—that have nothing whatsoever to do with the amino acid sequence of the protein.

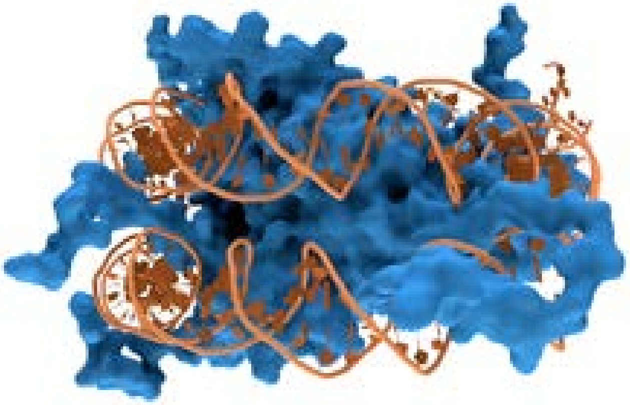
* Father Reading: Differential Gene Transcription http://www.ncbi.nlm.nih.gov/books/NBK10023/

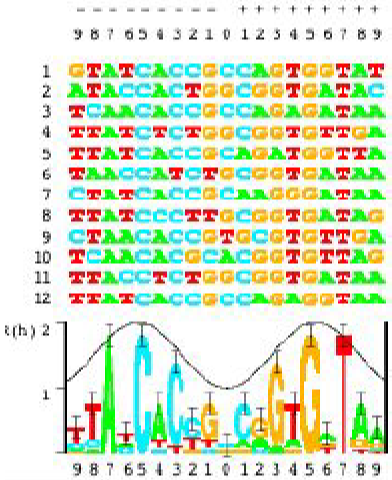
DNA-BINDING PROTEIN

http://en.wikipedia.org/wiki/DNA-binding_protein

Proteins that are composed of DNA-binding domains and thus have a specific or general affinity for either single or double stranded DNA.

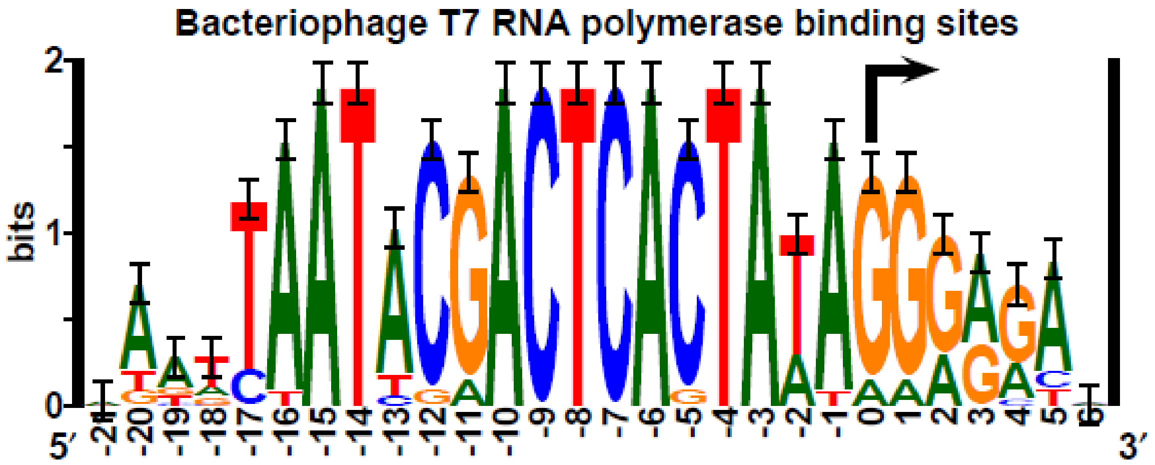
- Types of Binding
  - **Sequence-specific DNA-binding**
    - generally interact with the major groove of DNA
  - Non-specific DNA-protein interactions
  - DNA-binding proteins that specifically bind single-stranded DNA





SEQUENCE LOGO

- Sequence logo is a graphical representation of the sequence **conservation** of nucleotides (in a strand of DNA/RNA) or amino acids (in protein sequences)



Schneider & Stephens Nucl. Acids Res. 18: 6097-6100 1990

1 ttattaatacaactcactataaggagag
2 aaatcaatacgactcactatagagggac
3 cggttaatacgactcactataggagaac
4 gaagtaatacgactcagtatagggacaa
5 taattaattgaactcactaaagggagac
6 cgcttaatacgactcactaaaggagaca

Sequence Alignment

6 of 17 sites

Instructor: Sael Lee

CS549 Spring – Computational Biology

LECTURE 3 & 4
INTRODUCTION TO INFORMATION THEORY

BASIC PROBABILITY RULES

Marginalization

$$p(y) = \sum_x p(x,y) = \sum_x p(y|x)p(x)$$

$$p(y) = \int_x p(x,y) = \int_x p(y|x)p(x)$$

Bayes' Rule

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}$$

Product Rule

$$p_{X,Y}(x,y) = p_{Y|X}(y|x)p_X(x)$$
$$=p_{X|Y}(x|y)p_Y(y)$$

Convention

- $0 \log 0 = 0$
- $\mathrm{a} \log \frac{a}{0} = \infty$, if $a > 0$
- $0 \log \frac{0}{0} = 0$

INDEPENDENCE REVIEWED

The events *X* = *x* and *Y* = *y* are *statistically independent* if

$$\boxed{p(x, y) = p(x)p(y).}$$

The random variables X and Y defined over the alphabets χ and ψ , resp. are statistically independent if

$$p_{X,Y}(x,y) = p_X(x)p_Y(y), \text{ for } \forall(x,y) \in \chi \times \psi$$

The variables $X_1, X_2, \dots, X_N$ are called independent if for all $(x_1, x_2, \dots, x_N\,) \in \chi_1 \times \chi_x \times \cdots \times \chi_N$

$$p(x_1, x_2, \dots, x_N) = \prod_{i=1}^{N} p_{X_i}(x_i)$$

They are furthermore called identically distributed if all variables X_i have the same distribution $p_X(x)$.

EXPECTED VALUE

1 Discrete random variable, finite case, taking $x_1, x_2, \ldots, x_N$ with prob. $p_1, p_2, \ldots, p_N$

$$\boxed{E[X] = \frac{x_1 p_1 + x_2 p_2 + \cdots + x_k p_N}{p_1 + p_2 + \cdots + p_N}}$$

Sum to 1 if probability

2 Discrete random variable X, countable case, taking $x_1, x_2, \ldots$ with prob. $p_1, p_2, \ldots$

$$\boxed{E[X] = \sum_{i=1}^{\infty} x_i p_i}$$

3 Univariate continuous random variable:

$$E[X] = \int_{-\infty}^{\infty} x f(x)\,\mathrm{d}x$$

General definition: random variable defined on a probability space (Ω, Σ, P), then the expected value of X, denoted by E[X], ⟨X⟩, X̅ or **E**[X], is defined as the Lebesgue integral

$$E[X] = \int_{\Omega} X\, dP \; = \int_{\Omega} X(\omega)\, P(\mathrm{d}\omega)$$

ENTROPY

Definition:

The **entropy** *H*(*X*) of a discrete random variable *X* with pmf $p_X(x)$ is given by

$$H(X) = -\sum_{x} p_X(x) \log p_X(x) = -E_{p_X(x)}[\log p_X(X)]$$

The **entropy** *H*(*X*) of a continuous random variable *X* with pdf $f_X(x)$ in support set S is given by

$$h(X) = -\int_S f_X(x) \log f_X(x) = -E_{f_X(x)}[\log f_X(X)]$$

Meaning:

- Measure of the uncertainty of the r.v.
- Measure of the amount of information required on the average to describe the r.v.

Denote H(X) and H(p) as same when X is binary rv
Use log base 2

JOINT ENTROPY

Definition:

The **joint entropy** *H(X,Y)* on a pair of discrete r.v. *(X,Y)* with a joint distribution *p(x,y)* is defined as

$$H(X,Y) = -\sum_x \sum_y p(x,y)\log p(x,y)$$
$$= -E_{p(x,y)} \log p(x,y)$$

CONDITIONAL ENTROPY

Definition:

The **conditional entropy** *H(Y|X)* on a pair of discrete r.v. *(X,Y)* with a joint distribution *p(x,y)* is defined as

$$H(Y|X) = -\sum_x p(x)H(Y|X=x)$$
$$= \sum_x p(x)\sum_y p(y|x)\log p(y|x)$$
$$= -\sum_x \sum_y p(x,y)\log p(y|x)$$
$$= -E_{p(x,y)} \log p(y|x)$$

CHAIN RULE

Theory (**Chain Rule**)

$$\begin{aligned} H(X,Y) &= H(X) + H(Y|X) \\ &= H(Y) + H(X|Y) \end{aligned}$$

proof

Corollary

$$H(X,Y|Z) = H(X|Z) + H(Y|X,Z)$$

Remark

$$H(Y|X) \neq H(X|Y)$$
$$H(Y) - H(Y|X) = H(X) - H(X|Y)$$

RELATIVE ENTROPY

Definition:

The **relative entropy** (**Kullbuck-Leibler distance, K-L divergence**) between two probability mass function *p(x)* and *q(x)* is defined as

$$D(p||q) = \sum_{x \in \chi} p(x) \log \frac{p(x)}{q(x)} = E_p \log \frac{p(X)}{q(X)}$$

Meaning:

- **Distance** between two distributions
- A measure of the **inefficiency** of assuming that the distribution is *q* when the true distribution is *p*

Properties:

- Is non-negative
- $D(p||q) = 0$ if and only if *p*=*q*
- Is asymmetric : $D(p||q) \neq D(q||p)$
- Does not satisfy triangle inequality

Definition:

The **conditional relative entropy** between two probability mass function *p(x,y)* and *q(x,y)* is defined as

$$D(p(y|x)||q(y|x)) = \sum_{x \in \chi} p(y|x) \log \frac{p(y|x)}{q(y|x)} = E_{p(x,y)} \log \frac{p(Y|X)}{q(Y|X)}$$

MUTUAL INFORMATION

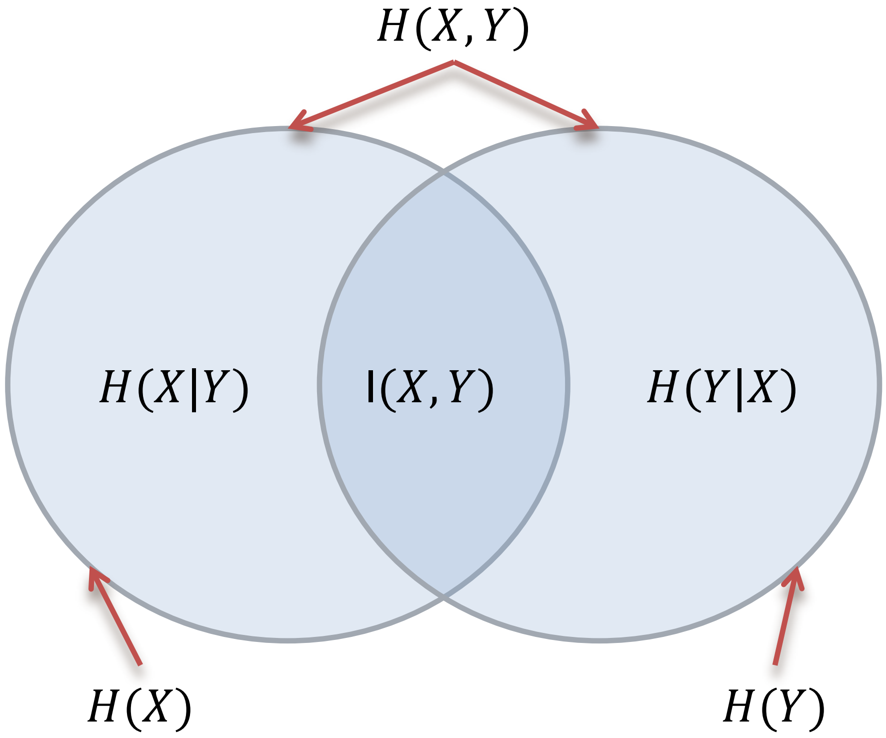
Definition:
Mutual information I(X;Y) is the relative entropy between the joint distribution p(x,y) and the product distribution p(x)p(y)

$$I(X;Y) = D(p(x,y)||p(x)p(y))$$
$$= \sum_x \sum_y p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$$
$$= E_{p(x,y)} \log \frac{p(X,Y)}{p(X)p(Y)}$$

Definition:
Conditional mutual information I(X;Y|Z) is the reduction in the uncertainty of X due to knowledge of Y when Z is given

$$I(X;Y|Z) = D(p(x,y|z)||p(x|z)p(y|z))$$
$$= \sum_x \sum_y p(x,y|z) \log \frac{p(x,y|z)}{p(x|z)p(y|z)}$$
$$= E_{p(x,y,z)} \log \frac{p(X,Y|Z)}{p(X|Z)p(Y|Z)}$$
$$= \mathrm{H(X|Z) - H(X|Y, Z)}$$

RELATIONSHIP BETWEEN ENTROPY AND MUTUAL INFORMATION



Properties:

- I(X;Y) is the reduction of uncertainty of X due to the knowledge of Y (or *vise versa*)

  proof $I(X;Y) = H(X) - H(X|Y)$
  $I(X;Y) = H(Y) - H(Y|X)$

- Is symmetric: X says about Y as much and Y says about X

- $I(X;Y) = H(Y) + H(X) - H(X,Y)$
  since $H(X,Y) = H(X) + H(Y|X)$
  by chain rule

- $I(X;X) = H(X)$ also called **self information**

Instructor: Sael Lee

CS549 Spring – Computational Biology

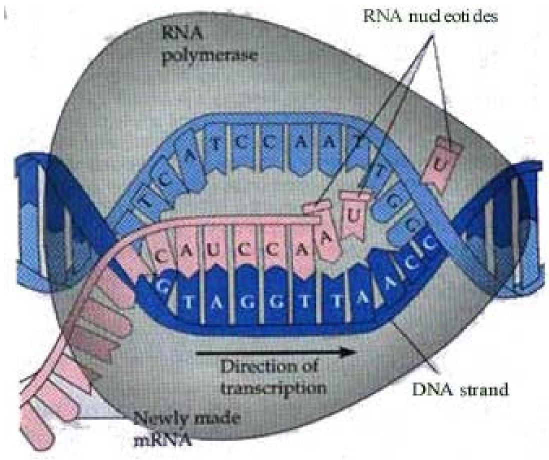
LECTURE 4:
DNA BINDING AND INFORMATION THEORY

A BRIEF REVIEW OF MOLECULAR INFORMATION THEORY.

SCHNEIDER, T. D. , (2010). *NANO COMMUNICATION NETWORKS1*(3), 173–180.

MOLECULAR INFORMATION THEORY

- **Molecular information theory:** Using information theory to measure states and patterns of molecules.
- Problem we focus on: **Interaction between DNA and Protein**

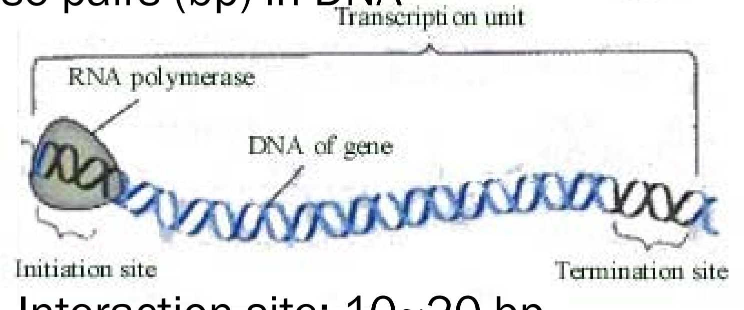


Transcription process:
RNA Polymerase (protein) binding to DNA

PROBLEM:
Analysis of interaction between DNA and proteins that control the expression DNA

PROPERTIES:

- Protein is a finite molecule
- Interaction content of proteins cover 10-20 base pairs (bp) in DNA

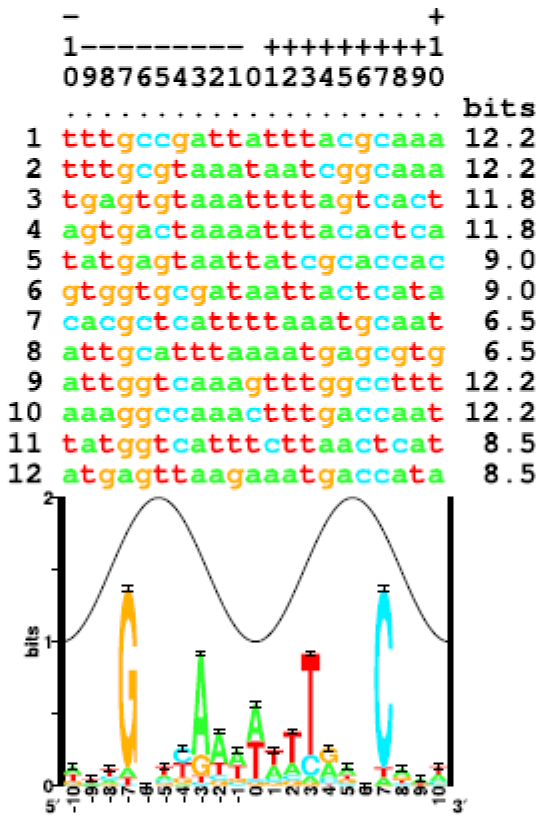


Interaction site: 10~20 bp

SEQUENCE LOGO – REVIEWED

- **Sequence logo** is a graphical representation of the sequence **conservation** of nucleotides (in a strand of DNA/RNA) or amino acids (in protein sequences)
- They can show how much pattern is in a set of binding sits.

Schneider & Stephens (1990) NAR. 18: 6097-6100



```
    -                   +
    1--------- +++++++++1
    098765432101234567890
    ..................... bits
 1  tttgccgattatttacgcaaa 12.2
 2  tttgcgtaaataatcggcaaa 12.2
 3  tgagtgtaaattttagtcact 11.8
 4  agtgactaaaatttacactca 11.8
 5  tatgagtaattatcgcaccac  9.0
 6  gtggtgcgataattactcata  9.0
 7  cacgctcattttaaatgcaat  6.5
 8  attgcatttaaaatgagcgtg  6.5
 9  attggtcaaagtttggccttt 12.2
10  aaaggccaaactttgaccaat 12.2
11  tatggtcatttcttaactcat  8.5
12  atgagttaagaaatgaccata  8.5
```

120 Fis binding sites

EX> Fis site

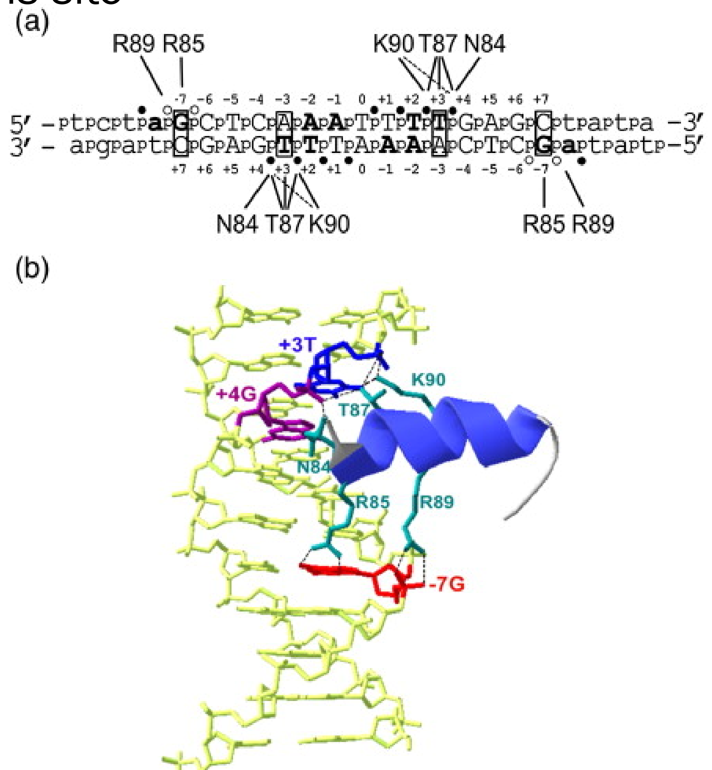
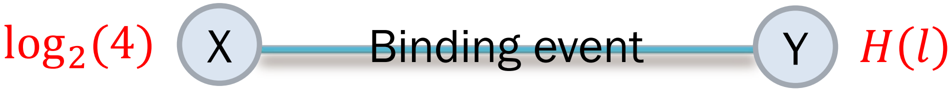


Fig. 6. Major determinants in Fis–DNA binding. [Shao et al. (2008) JMB 380:2, 327-339.]

CHARACTERIZING BINDING SITES

- Before binding, protein is uncertain as to what base it will see and that uncertainty can be measured as $\log_2(4)$
  - Before we know the binding event can occur, all four bases (A,T,C,G) can be seen in a DNA locus.
- After binding, uncertainty of what it is touching in different cases is lower.
  - If only one type of bases occur:
$$\log_2(1) = 0$$
  - If other bases occur as well: (Conditional Entropy)
$$H(l) > 0$$


$\log_2(4)$ X Binding event Y $H(l)$

The **information content** (y-axis) of position *l*:

Height in sequence logo

Four letter: A,T,C,G

Entropy

small-sample correction

$$R_{sequence}(l) = \log_2(4) - (H(l) + e_n)$$

(bits per base)

$$\boxed{\mathrm{I}(X;Y) = H(X) - H(X|Y)}$$

$\log_2(4)$: Uncertainty 'observed' by the DNA binding protein before binding to a site.

-> * maximum uncertainty possible: $\log_2|\chi|$

$H(l)$: Uncertainty 'observed' by the DNA binding protein after binding to a site.

$$H(l) = -\sum_{b\in\{A,T,G,C\}} f_{b,l}\, log_2\, f_{b,l}$$

(bits per base)

where $f_{b,l}$ are the frequency of base b at a position l.

Assuming independence between sites, **total information in a binding site.**

$$R_{sequence} = \sum_{l} R_{sequence}(l)$$

INFORMATION REQUIRED TO FIND A SET OF BINDING SITES

G = # of potential binding sites
  = genome size in some cases

γ = number of binding sites on genome

Information required to find binding sites

Uncertainty before being bound to one of the sites

Uncertainty after being bound to one of the sites

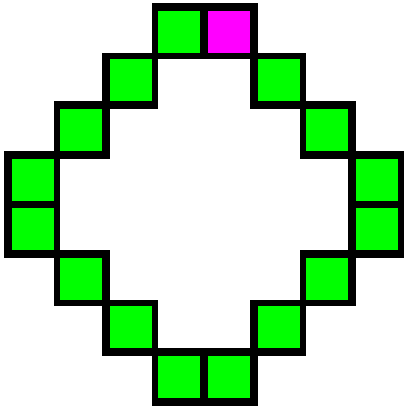
$$R_{frequency} = H_{before\ binding} - H_{after\ binding}$$

$$= \log_2 G - \log_2 \gamma$$

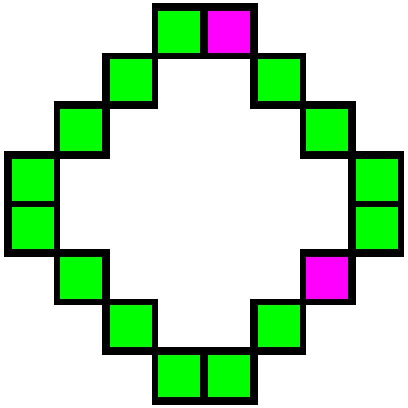
$$= -\log_2 \frac{\gamma}{G}$$

(bit per site)

INFORMATION REQUIRED TO FIND A SET OF BINDING SITES IN A GENOME



16 positions
1 site
$\log_2 16/1 = 4$ bits



16 positions
2 sites
$\log_2 16/2 = 3$ bits

Instructor: Sael Lee

CS549 Spring – Computational Biology

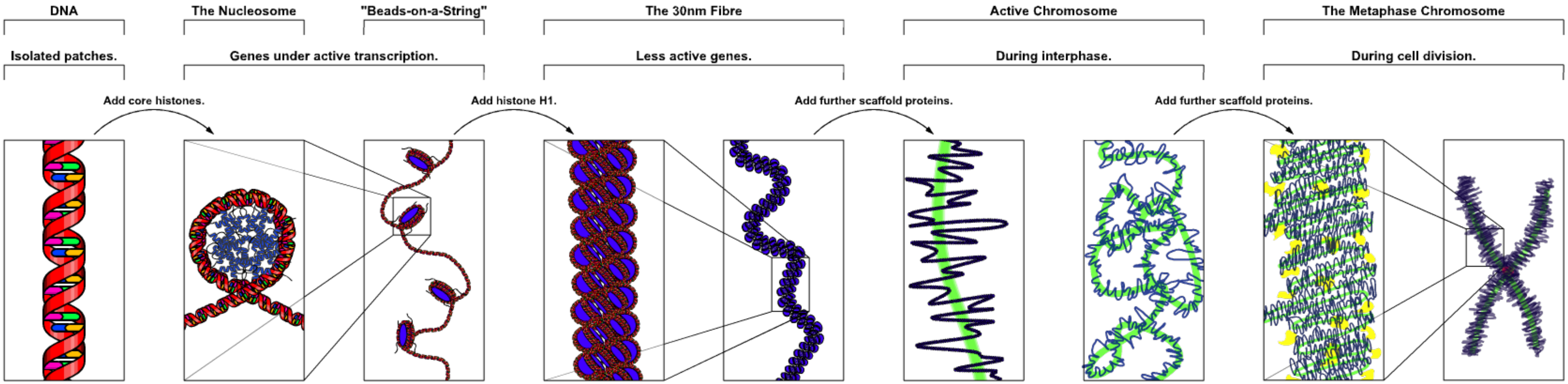
LECTURE 6:
FINDING NUCLEOSOME POSITIONS

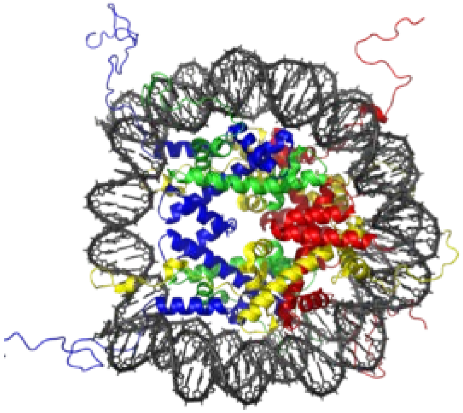
Reference:
C. Jiang and B. F. Pugh. Nucleosome positioning and gene regulation: advances through genomics.
Nature Reviews Genetics 10 161-172 (2009)

WHY NUCLEOSOMES POSITION?

- Knowing the precise locations of nucleosomes in a genome is key to understanding how genes are regulated.
- Nucleosome positions can tell us about
  - How nucleosome positioning distinguish promoter regions and transcriptional start sites, and
  - How the composition and structure of promoter nucleosomes facilitate or inhibit transcription.
  - How diverse factors, including underlying DNA sequences and chromatin remodeling complexes, influence nucleosome positioning

CHROMATIN STRUCTURES

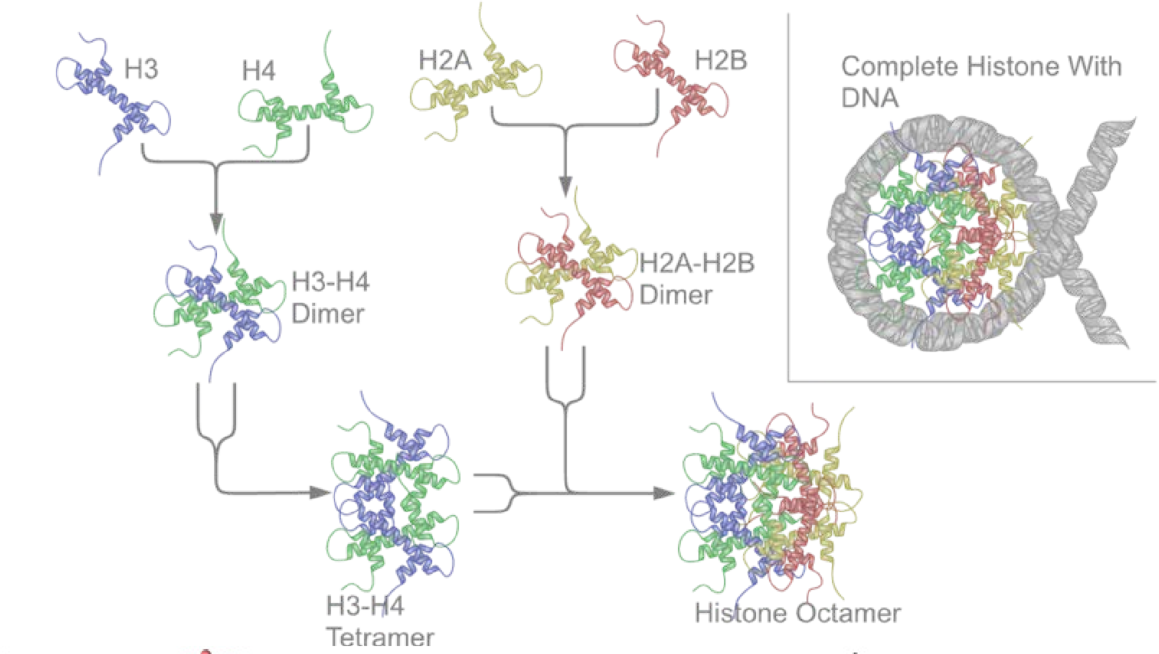


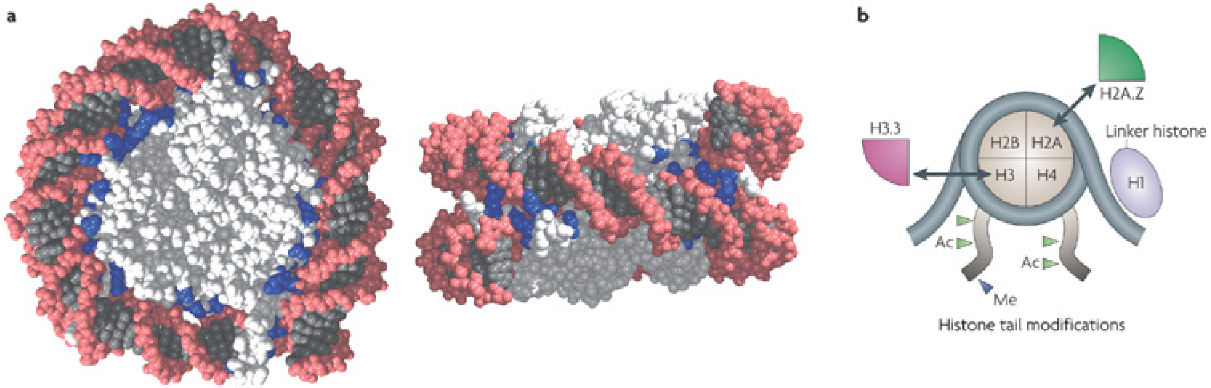


The packaging of DNA creates both a problem and an opportunity:

- Wrapping DNA around histones may be a obstacle in accessing the genetic code;
- Can be exploited so that enzymes that read, replicate and repair DNA can be directed to the appropriate entry sites

NUCLEOSOME STRUCTURE





Nature Reviews | Genetics

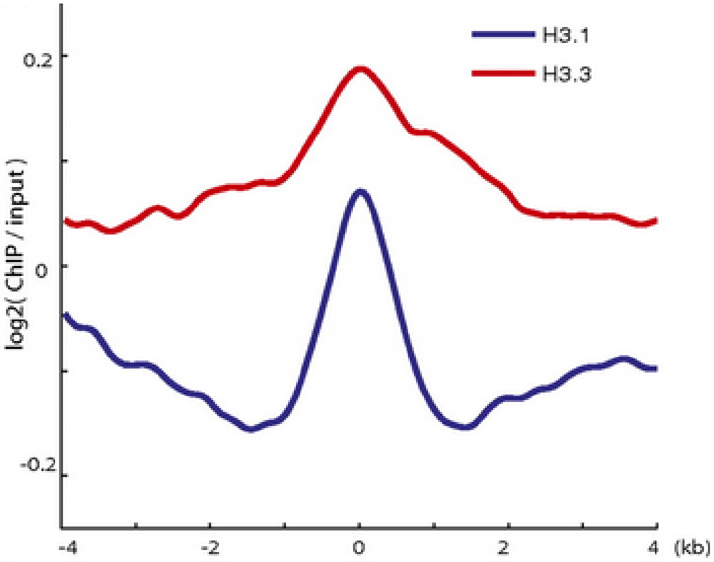
The nucleosome is the basic unit of eukaryotic chromatin, consisting of a histone core around DNA.

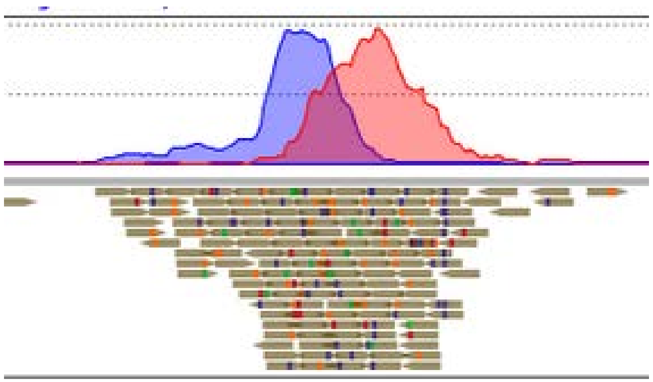
Each histone core is composed of two copies of each of the histone proteins H2A, H2B, H3 and H4. Approximately 147 bp of DNA coils 1.65 times around the histone octamer in a left-handed toroid.

GENOMEWIDE NUCLEOSOME MAPS

Allow us to explore the genomic properties of chromatin

At most loci, there is an approximately **Gaussian (normal) distribution** of nucleosome positions around particular genomic coordinates, ranging from ~30 bp for highly phased nucleosomes to a random continuous distribution throughout an array.





Cause of variation:

- Genuine positional heterogeneity
- how much is an artifact that is caused by overtrimming or undertrimming of the DNA at nucleosome borders by experiment

***Phasing**

The distribution of nucleosomes around a particular coordinate in a population of cells.

MIXTURE MODELS: INTRODUCTION

THE DENSITY ESTIMATION PROBLEM

Density Estimation Problem: (loose definition)

Given a set of N points in D dimensions, $\mathrm{x}_1, \ldots, \mathrm{x}_N \in R^D$, and a family F of probability density function on R^D, find the probability density functions (pdf) on R^D, find pdf $f(\mathrm{x}) \in F$ that is most likely to have generated the given points.

Defining F : give each of it's members the same mathematical form, and to distinguish different members by different values of a set of parameters θ.

EX> Mixture of PDFs

$$f(\mathbf{x}; \theta) = \sum_{k=1}^{K} \pi_k g(\mathbf{x}; \theta_k)$$

$$\int g(\mathbf{x}; \theta_k) d\mathbf{x} = 1$$

PDF

$$\int f(\mathbf{x};\ \theta) d\mathbf{x} = 1$$

Mixture of PDFs

$$\sum_{k=1}^{K} \pi_k\ = 1; \quad \pi_k > 0$$

Mixing probability

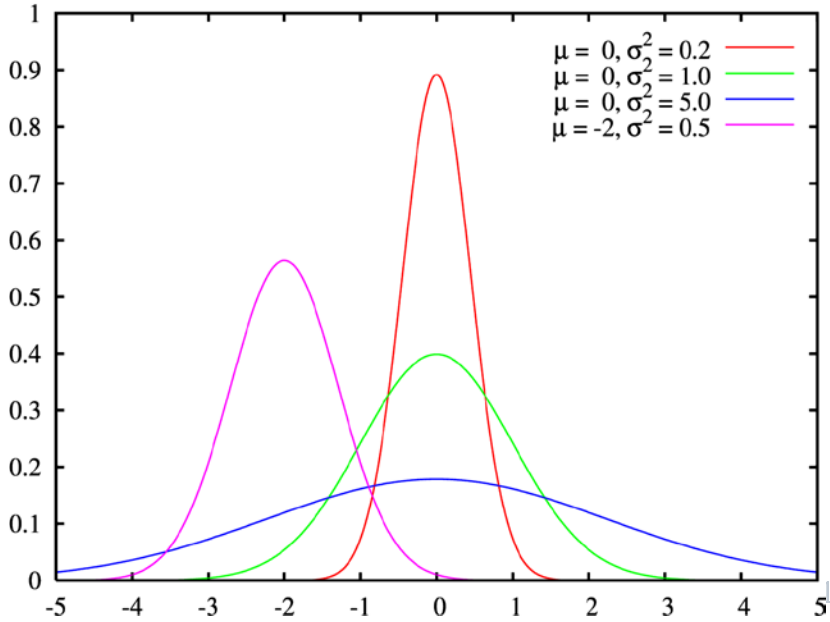
MIXTURE MODEL AND CLUSTERING

Example: Gaussian Mixture Models.

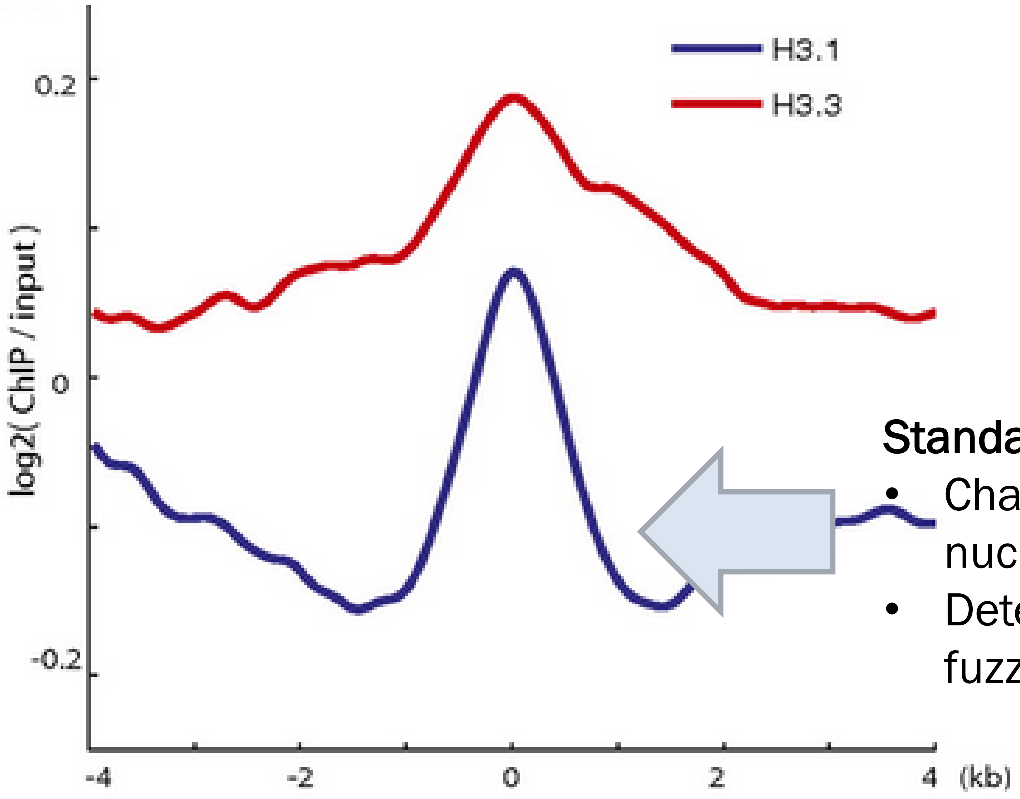
$$\sum_{k=1}^{K} \pi_k N(\mathbf{x}|\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})$$

$$N(\mathbf{x}|\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k}) = \frac{1}{(2\pi\sigma^2)^{\frac{D}{2}}} \frac{1}{|\boldsymbol{\Sigma_k}|^{\frac{1}{2}}} \exp(-\frac{1}{2}(x-\mu)^T \boldsymbol{\Sigma^{-1}}(x-\mu) \quad)$$

Each cluster is assigned a Gaussian, with **mean** being the center of cluster and **standard deviation** being the spread of data for the cluster.



GAUSSIAN MIXTURE MODEL AND NUCLEOSOME POSITION



Standard deviation:

- Characterize nucleosome stability
- Determine phased or fuzzy.

Mean:

- Determine nucleosome center position
- Determine spread of nucleosome

K-MEANS CLUSTERING: DISTORTION MEASURE

- Dataset {x1, . . . , xN}
- **Partition in K clusters**
- Cluster prototype: μk
- Binary indicator variable, 1-of-K Coding scheme

$$r_{nk} \in \{0, 1\}$$

$$r_{nk} = 1, \text{ and } r_{nj} = 0 \text{ for } j \neq k.$$

Only one is 1 and all other 0

- **Hard assignment.**
- **Distortion measure:** a measure of how much data point deviate from the center of their clusters

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \mu_k\|^2$$

K-MEANS CLUSTERING: EXPECTATION MAXIMIZATION

- Goal: Find values for $\{r_{nk}\}$ and $\{\mu_k\}$ to minimize:

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \mu_k\|^2$$

- Iterative procedure:

1. Minimize J w.r.t. r_{nk}, keep μ_k fixed (Expectation)

Calculate the membership

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|\mathbf{x}_n - \mu_k\|^2 \\ 0 & \text{otherwise} \end{cases}$$

2. Minimize J w.r.t. μ_k, keep r_{nk} fixed (Maximization)

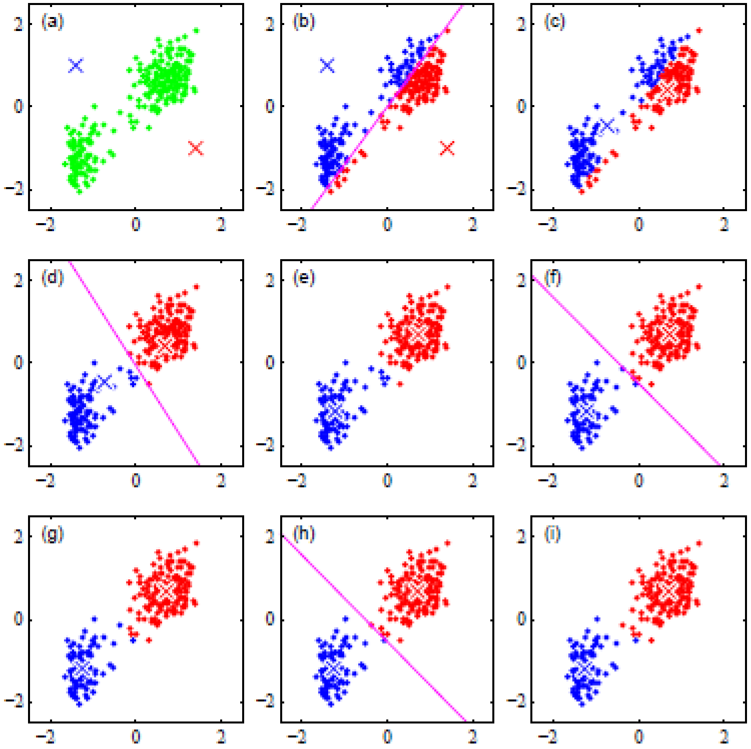
Calculate the center

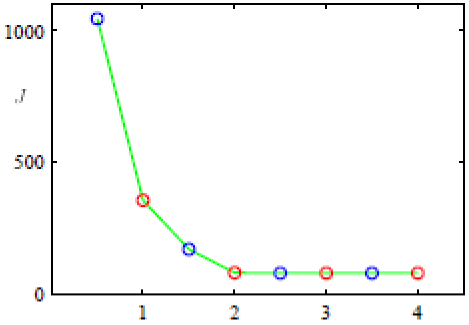
$$2 \sum_{n=1}^{N} r_{nk}(x_n - \mu_k) = 0$$

$$\mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}}$$

K-MEANS CLUSTERING: EXAMPLE

- Each E or M step reduces the value of the objective function J
- Convergence to a **local** maximum



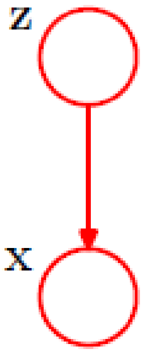


MIXTURE OF GAUSSIANS: LATENT VARIABLES

- Gaussian Mixture Distribution:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)$$

- Introduce latent variable z
  - z is binary 1-of-K coding variable
  - p(x, z) = p(z)p(x|z)



GOAL

We want to identify which data came from which source.

In probabilistic modeling words

“Evaluate the **posterior distribution** ***p*****(Z /X)** of the latent variables **Z** (which source) given the observed (visible) data variables **X**, and the evaluation of expectations computed with respect to this distribution.”

Strategy for **parametric models**

Estimate $p(\mathbf{Z}|\mathbf{X})$ by estimating it’s parameter θ

Condition we work on: The data are **independently** generated by **sources** of data (distribution functions) and there are no (or ignorable) dependency between the sources. *Mixture models*

Estimating it’s parameter θ by evaluating the **log likelihood** $\mathbf{p(x|\boldsymbol{\theta})}$

A method the solve log likelihood function is using **Expectation Maximization**

MIXTURE OF GAUSSIANS: LATENT VARIABLES (2)

The use of the joint probability p(x, z), leads to significant simplifications

- Prior probability of components

$$p(z_k = 1) = \pi_k$$
constraints: $0 \leq \pi_k \leq 1$, and $\sum_k \pi_k = 1$
$$p(\mathbf{z}) = \prod_k \pi_k^{z_k}$$

- Gaussian function of each K mixing components

$$p(\mathbf{x}|z_k = 1) = \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$$
$$p(\mathbf{x}|\mathbf{z}) = \prod_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)^{z_k}$$

- Redistribution of Gaussian mixture model

- $$p(\mathbf{x}) = \sum_z p(\mathbf{x}, \mathbf{z}) = \sum_z p(\mathbf{z})p(\mathbf{x}|\mathbf{z}) = \sum_k \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)$$

MIXTURE OF GAUSSIANS: LATENT VARIABLES (3)

- Responsibility that component k takes for “explaining” observation x:
  - the posterior probability once we observed X.

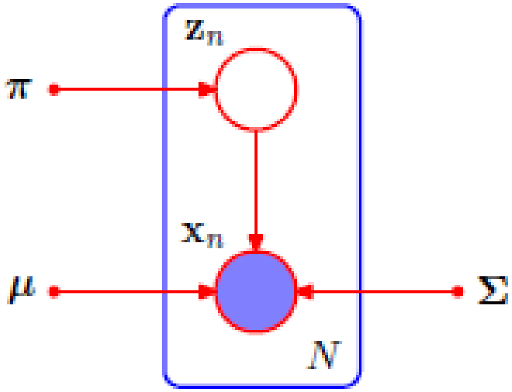
$$\gamma(z_k) \equiv p(z_k = 1|\mathbf{x}) = \frac{p(z_k = 1)p(\mathbf{x}|z_k = 1)}{\sum_k p(z_k = 1)p(\mathbf{x}|z_k = 1)}$$

$$= \frac{\pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)}{\sum_k \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)}$$

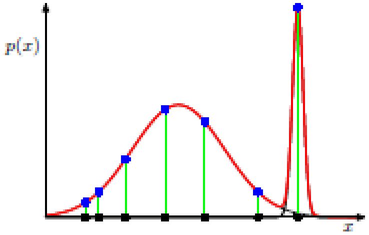
MIXTURE OF GAUSSIANS: MAXIMUM LIKELIHOOD

- Log Likelihood function of observations $X = \{x_1, \ldots, x_N\}$

$$\ln p(\mathbf{X}|\pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k) \right\}$$



- Problems with Log Likelihood
  - Singularity when a mixture component collapses on a data point
  - Identifiability for a ML solution in a K-component mixture there are K! equivalent solutions.
  - * We assume we can use heuristics to overcome these problems.



MIXTURE OF GAUSSIANS: EM FOR GAUSSIAN MIXTURES

- Informal introduction of expectation-maximization algorithm (Dempster et al., 1977).
- Maximum of log likelihood:
  - Derivatives of $\ln p(X|\pi, \mu, \Sigma)$ w.r.t parameters to 0.

$$\ln p(\mathbf{X}|\pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k) \right\}$$

MIXTURE OF GAUSSIANS: EM FOR GAUSSIAN MIXTURES SUMMARY

1. Initialize $\{\mu_k, \Sigma_k, \pi_k\}$ and evaluate log-likeihood
2. E-Step: Evaluate responsibilities $\gamma(z_k)$
3. M-Step: Re-estimate paramters θ, using current responsibilities $\gamma(z_k)$

$$\mu_k^{\text{new}} = \frac{1}{\sum_n \gamma(z_k)} \sum_n \gamma(z_k) \mathbf{x}_n$$

$$\Sigma_k^{\text{new}} = \frac{1}{\sum_n \gamma(z_k)} \sum_n \gamma(z_k) (\mathbf{x}_n - \mu_k)(\mathbf{x}_n - \mu_k)^T$$

$$\pi_k^{\text{new}} = \frac{\sum_n \gamma(z_k)}{N}$$

4. Evaluate log-likelihood $\ln p(X|\pi, \mu, \Sigma)$ and check for convergence of either the parameters or the log likelihood.

   If convergence criterion is not satisfied return to step 2.

RELATION TO *K*-MEANS

- *K*-means algorithm with the EM algorithm for Gaussian mixtures shows that there is a close similarity
  - *K*-means algorithm performs a *hard* assignment of data points to clusters, in which each data point is associated uniquely with one cluster,
  - the EM algorithm makes a *soft* assignment based on the posterior probabilities.

MIXTURE OF GAUSSIANS:

- Gaussian Mixture Distribution

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k N(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Where $p(z_k = 1) = \pi_k$: prior prob. of $z_k = 1$

$$N(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{\frac{D}{2}}} \frac{1}{|\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\}$$

- Posterior probability of z_k (responsibility) once we observed a point **x**

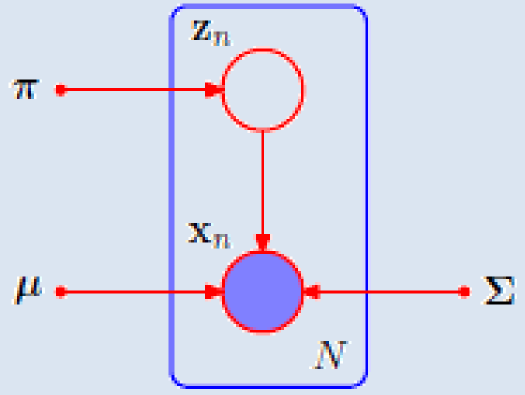
$$\gamma(z_k) \equiv p(z_k = 1|\mathbf{x}) = \frac{\pi_k N(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j N\left(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)}$$

Where $\theta_{\boldsymbol{k}} = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k\}$ and $\boldsymbol{\theta}=\{\theta_{\mathbf{1}}, \dots, \theta_K\}$

- Log Likelihood function

$$\ln p\,(\mathbf{X}|\pi, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln\{\sum_{k=1}^{K} \pi_k N(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\}$$

Mixture of Gaussians Model



$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})p(\mathbf{x}|\mathbf{z})$$

- N number of D dimension data **X**
  $\boldsymbol{X} = \{\mathbf{x_1}, \dots, \mathbf{x}_N\}, \mathbf{x} = \{\mathrm{x}_1, \dots, x_D\}$

- N number of K dim. class variable **Z**
  $\boldsymbol{Z} = \{\boldsymbol{z_1}, \dots, \boldsymbol{z}_N\}, \boldsymbol{z} = \{z_1, \dots, z_K\}$

MIXTURE OF GAUSSIANS: EM FOR GAUSSIAN MIXTURES SUMMARY

1. Initialize $\{\mu_k, \Sigma_k, \pi_k\}$ and evaluate log-likeihood
2. E-Step: Evaluate responsibilities $\gamma(z_k)$

$$\gamma(z_k) \equiv p(z_k = 1|\mathbf{x}) = \frac{\pi_k N(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j N(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

3. M-Step: Re-estimate parameters $\boldsymbol{\theta}$, using current responsibilities $\gamma(z_k)$

$$\mu_k^{\text{new}} = \frac{1}{\sum_n \gamma(z_k)} \sum_n \gamma(z_k)\mathbf{x}_n$$

$$\Sigma_k^{\text{new}} = \frac{1}{\sum_n \gamma(z_k)} \sum_n \gamma(z_k)(\mathbf{x}_n - \mu_k)(\mathbf{x}_n - \mu_k)^T$$

$$\pi_k^{\text{new}} = \frac{\sum_n \gamma(z_k)}{N}$$

Maximize log-likelihood

$$\ln p\,(\mathbf{X}|\pi, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln\{\sum_{k=1}^{K} \pi_k N(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\}$$

4. Evaluate log-likelihood $\ln p(\boldsymbol{X}|\pi, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ and check for convergence of either the parameters or the log likelihood.

   If convergence criterion is not satisfied return to step 2.

AN ALTERNATIVE VIEW OF EM: LATENT VARIABLES

- Let X observed data, Z latent variables, parameters.
- Goal: maximize marginal log-likelihood of observed data

$$\ln p(\boldsymbol{X}|\boldsymbol{\theta}) = \ln\{\sum_{\mathbf{Z}} p(\mathbf{X},\mathbf{Z}\,|\boldsymbol{\theta})\}$$

Summation over the latent variables appears inside the logarithm

Log-sum prevents the logarithm from acting directly on the joint distribution, resulting on complicated expressions for the maximum log likelihood solution.

AN ALTERNATIVE VIEW OF EM: GENERAL EM ALGORITHM

Given a joint distribution $\mathrm{p}(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})$ over observed variables **X** and latent variables **Z**, governed by parameters ***θ***, the goal is to maximize the likelihood function $\mathrm{p}(\mathbf{X} \mid \boldsymbol{\theta})$ with respect to ***θ***.

1. Initialization: Choose initial set of parameters $\boldsymbol{\theta}^{old}$
2. E-step: use current parameters $\boldsymbol{\theta}^{old}$ to compute.

$$p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{\mathbf{old}})$$

3. M-step: determine θ^{new} by maximizing $Q(\theta, \theta^{old})$

$$\boldsymbol{\theta}^{\mathbf{new}} = \arg_{\boldsymbol{\theta}} \max Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\mathrm{old}}).$$

Where

$$\mathrm{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\mathrm{old}}) = \sum_{z} p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{\mathbf{old}}) \ln p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})$$

Logarithm acts directly on the joint distribution $p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})$ so maximization is tractable

$$\ln p(\boldsymbol{X} \mid \boldsymbol{\theta}) = \ln\{\sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})\}$$

4. Check convergence either the log likelihood or the parameter values : stop, or $\boldsymbol{\theta}^{old} \longleftarrow \boldsymbol{\theta}^{new}$ and go to step 2.

AN ALTERNATIVE VIEW OF EM: GENERAL EM ALGORITHM FOR GAUSSIAN MIXTURE MODEL

Given a joint distribution $\mathrm{p}(\mathbf{X}, \mathbf{Z}\,|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ over observed variables X and latent variables Z, governed by parameters $\{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$, the goal is to maximize the likelihood function $\mathrm{p}(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ with respect to $\{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$.

1. Initialization: Choose initial set of parameters $\{\boldsymbol{\pi}^{old}, \boldsymbol{\mu}^{old}, \boldsymbol{\Sigma}^{\mathrm{old}}\}$
2. E-step: use current parameters $\{\boldsymbol{\pi}^{old}, \boldsymbol{\mu}^{old}, \boldsymbol{\Sigma}^{\mathrm{old}}\}$ to compute.

$p(\mathbf{Z}\,|\mathbf{X}, \boldsymbol{\theta}^{\mathbf{old}})$

$$E_z[z_{nk}] = \gamma(z_{nk}) \equiv \frac{\pi_k N(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j N\left(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)}$$

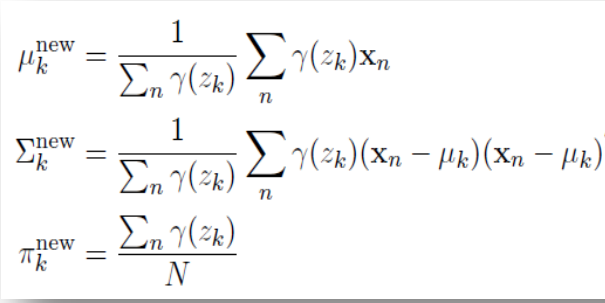
3. M-step: determine θ^{new} by maximizing $Q(\theta, \theta^{old})$

Closed form solution

$$\{\boldsymbol{\pi}_k{}^{new}, \boldsymbol{\mu}_k{}^{new}, \boldsymbol{\Sigma}_k{}^{new}\} = \arg_{\{\boldsymbol{\pi}_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}} \max \mathrm{E}_{\mathrm{Z}}[\,\ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\pi}_{\mathbf{k}}, \boldsymbol{\mu}_{\mathbf{k}}, \boldsymbol{\Sigma}_{\mathbf{k}})].$$

$$= \arg_{\{\boldsymbol{\pi}_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}} \max \sum_{n=1}^{N} \gamma(z_{nk})\{\ln \pi_k + \ln N(\mathbf{x}_{\boldsymbol{n}}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\}$$

$$\boldsymbol{\theta}^{\mathbf{new}} = \arg_{\boldsymbol{\theta}} \max \textstyle\sum_{z} p(\mathbf{Z}\,|\mathbf{X}, \boldsymbol{\theta}^{\mathbf{old}}) \ln p(\mathbf{X}, \mathbf{Z}\,|\boldsymbol{\theta}).$$



$$\mu_k^{\text{new}} = \frac{1}{\sum_n \gamma(z_k)} \sum_n \gamma(z_k) \mathbf{x}_n$$

$$\Sigma_k^{\text{new}} = \frac{1}{\sum_n \gamma(z_k)} \sum_n \gamma(z_k)(\mathbf{x}_n - \mu_k)(\mathbf{x}_n - \mu_k)$$

$$\pi_k^{\text{new}} = \frac{\sum_n \gamma(z_k)}{N}$$

4. Check convergence either the log likelihood or the parameter values : stop, or $\boldsymbol{\theta}^{old} \longleftarrow \boldsymbol{\theta}^{new}$ and go to step 2.

Instructor: Sael Lee

CS549 Spring – Computational Biology

LECTURE 11: BIOMARKER DISCOVERY

Resources: Steven Skiena's CSE 549 lecture 15-18 slides

WHAT IS A BIOMARKER?

- **Biomarker**, or biological marker, is any type of indicator of biological state.
  - "cellular, biochemical or molecular alterations that are measurable in biological media such as human tissues, cells, or fluids." - [B S Hulka (1990) New York: Oxford University Press]
- It objectively measures the states of biology in medicine, cell biology, geology, ecotoxicology, etc.
- The most popular uses are in medicine to measure states in:
  - Normal biological process
  - Pathogenic process
  - Pharmacological responds to therapeutics

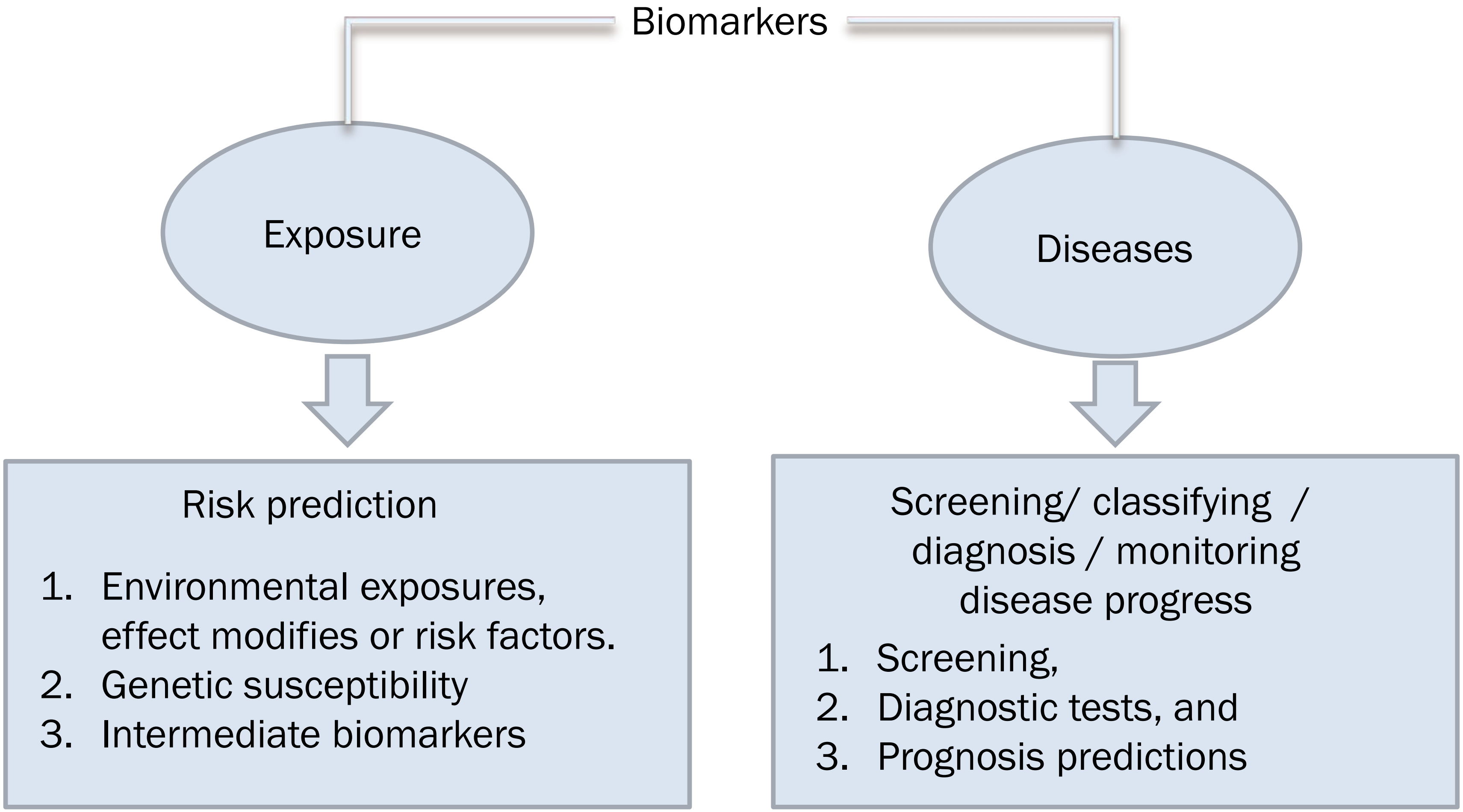
http://www.news-medical.net/health/Biomarker-What-is-a-Biomarker.aspx

CAPABILITIES OF BIOMARKERS [TABLE 1 OF MAYEUX, R. 2004]

- Delineation of events between exposure and disease
- Establishment of dose-response
- Identification of early events in the natural history
- Identification of mechanisms by which exposure and disease are related
- Reduction in misclassification of exposures or risk factors and disease
- Establishment of variability and effect modification
- Enhanced individual and group risk assessments

Mayeux, R. (2004). Biomarkers: potential uses and limitations. *NeuroRx : the journal of the American Society for Experimental NeuroTherapeutics*, *1*(2), 182–8.

TYPES OF BIOMARKERS



Biomarkers

Exposure

Diseases

Risk prediction

1. Environmental exposures, effect modifies or risk factors.
2. Genetic susceptibility
3. Intermediate biomarkers

Screening/ classifying / diagnosis / monitoring disease progress

1. Screening,
2. Diagnostic tests, and
3. Prognosis predictions

POSSIBLE SHORT COMES OF BIOMARKERS

Short Comes of Biomarkers

Variability

Validity

1. Difference in amount of an external exposure
2. Difference in the way a putative toxin is metabolized
3. Personal difference / Group difference / measurement error

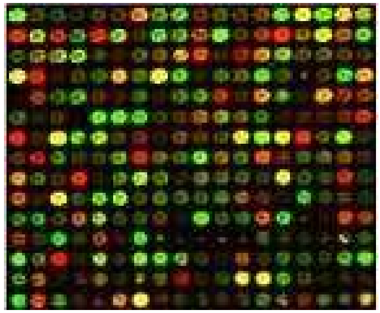
1. Content validity
   - degree to which a biomarker reflects the study
2. Construct validity
   - relevant characteristics of the disease or trait
3. Criterion validity
   1. sensitivity,
   2. specificity, and
   3. predictive power

DATA USED FOR BIOMARKER DISCOVERY

- Bio-specimens used:
  - Blood, brain, cerebrospinal fluid, spinal fluid, muscle, nerve, skin, and other body fluids
  - In both the healthy and diseased state
- **DNA, RNA, or protein**
  - **EX> Microarray chips, Genome sequences,**
- Cytogenetic markers
  - ex> chromosome structure
- Tissue markers
  - Microscope level visible differences
- Behavior markers
- Measure toxicants in body fluids & tissues
- Death of marker animals
  - Ex> environmental conditions.

FOCUSING ON GENE EXPRESSION

- Certain technologies have been developed where different compounds are anchored to tiny *beads*, so reacting beads can be labeled, isolated, and identified.
- But the best solution is to attach distinct compounds to different regions of a solid substrate so you know *where* they are.



WHAT DOES MICROARRAY MEASURE

- Analysis of post translational modifications in genes
  - ex.> methylation states.
- Sequencing variants of a *known* genome
  - detecting single nucleotide polymorphisms (SNPs)
- Identifying a specific strain of virus
  - (e.g. the Affymetrix HIV-1 array).
- Measuring differential expression of all genes in tumor and normal cells,
  - to determine which genes may cause/cure cancer

- Identify which treatment a specific tumor should respond best to.
  - Paired treatment
- Measuring differential expression of all genes in different tissue types,
  - to determine what makes one cell type different than another.
- Measuring differential expression of all genes in different time
  - Circadian rhythm
- Measuring copy number variants from chromosomal anomalies or cancer.
- Obtaining individual's genotype / SNP data, e.g. 23andMe

DNA MICROARRAY

cDNA microarray YouTube 1. – Gabriel Mckinsey

DNA Microarray YouTube 2.

- Single stranded DNA/RNA molecules are anchored by one end to the plate/substrate.
  - These molecules will seek to hybridize with complementary strands floating in solution.
- The target molecules are fluorescently labeled,
  - so that the spots on the *chip/array* where hybridization occurs can be identified.
- The strength of the detected signal somewhat reflects the amount of stuff which binds to it,
  - and thus the amount of the target in solution.
- Such *quantitative* expression data is not very reliable, however.

http://www.3d-gene.com/en/about/abo_001.html

COMPLEXITY IN ANALYSIS OF MICROARRAY DATA

- Underlying biological processes being investigated are often not understood and are almost certainly complex
- Measures the steady-state level of an unstable molecule , mRNA
  - Depends on the rate of transcription and degradation of the mRNA.

CLASSIFICATION AND CLUSTERING PROBLEM

- Finding Biomarkers using microarray data becomes **feature selection** (gene selection) problem in **classification** (supervised learning) and **clustering** (unsupervised learning)

FEATURE SELECTION AND BIOMARKER DISCOVERY

- Feature selection challenge specific to microarray data:
  - Large feature (gene) and small number of data (samples)
  - Reproducibility is low
    - need stable feature selection method.

- Cause of instability
  - Algorithm design without considering stability
  - The existence of multiple sets of true markers
  - Small number of samples in high dimensional data

FEATURE SELECTION

- Selected features can be singular or form groups.
  - Singular: early onset genetic diseases
  - **Group feature: complex diseases**
    - cancer, diabetes, etc

- Incorporation of prior-knowledge in to feature selection.
  - **Best to incorporate all we know** esp. since variable samples are always small
    - Interaction between genes

Instructor: Sael Lee

CS549 Spring – Computational Biology

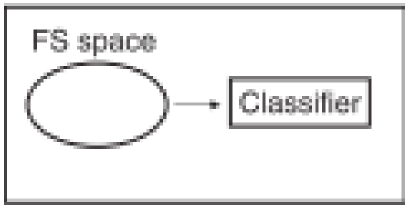
LECTURE 12-13: FEATURE SELECTION

Ref.

1. C. M. Bishop “Pattern Recognition and Machine Learning” 2nd ed. & provided sides

TYPES OF FEATURE SELECTION METHOD

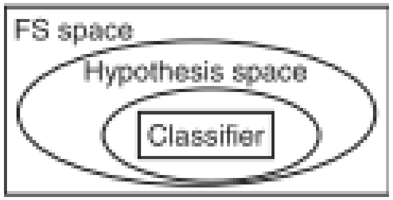
Filtering Methods



relevance of features is evaluated by looking only at the intrinsic properties of the data

* Often **feature relevance score** is used to evaluate each feature (gene)

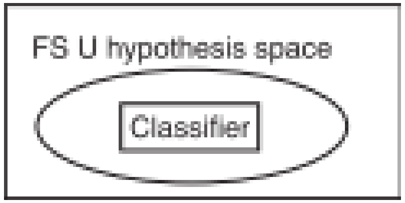
Wrapper Methods



model hypothesis search is embed within the feature subset search

-> **various subsets of features** are generated and evaluated

Embedded Method



optimal feature subset search is built into the classifier construction

-> a search in the **combined space of feature subsets and hypotheses**

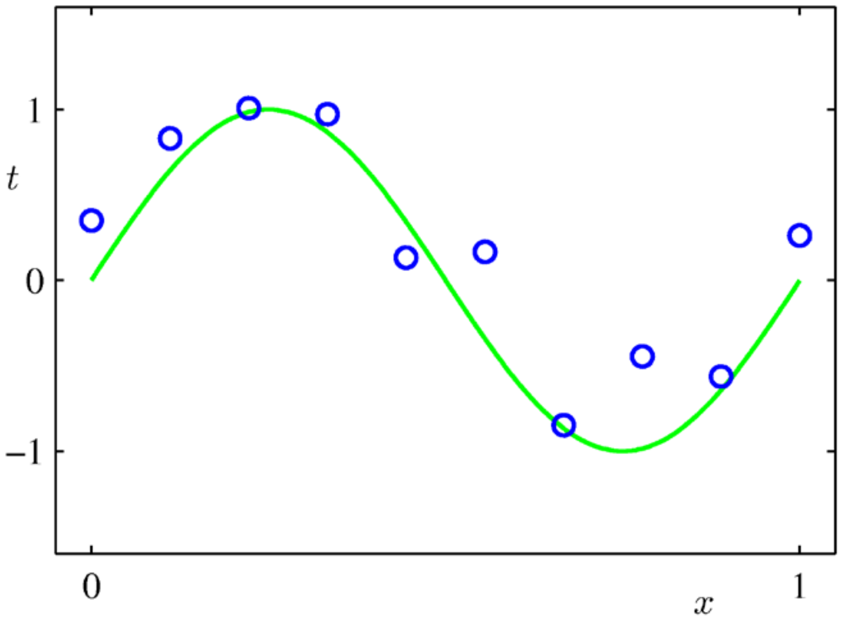
Saeys, Y., Inza, I., & Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics*, *23*(19), 2507–17.

Chapter 3 of PRML

FEATURE SELECTION WITH LASSO REGRESSION MODEL

LINEAR BASIS FUNCTION MODELS (1)

- Example: Polynomial Curve Fitting



$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

LINEAR BASIS FUNCTION MODELS (2)

- Generally

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x})$$

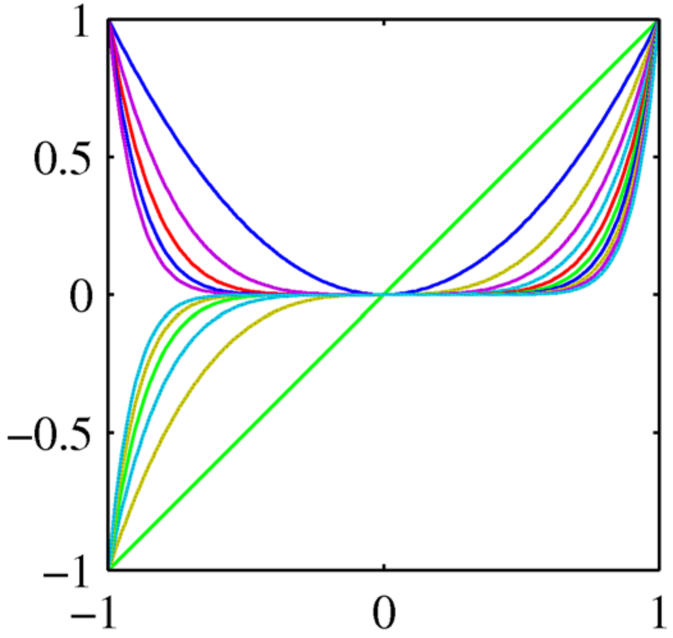
- where $\phi_j(\mathrm{x})$ are known as *basis functions*.
- Typically, $\phi_0(\mathrm{x}) = 1$, so that w_0 acts as a bias.
- In the simplest case, we use linear basis functions : $\phi_d(x) \; = \; x_d$.

LINEAR BASIS FUNCTION MODELS (3)

- Polynomial basis functions:

$$\phi_j(x) = x^j.$$

- These are global; a small change in x affect all basis functions.

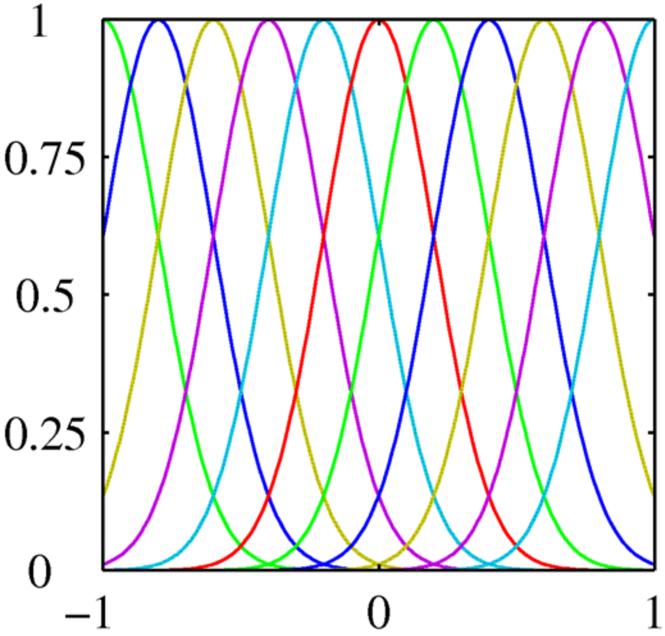


LINEAR BASIS FUNCTION MODELS (4)

- Gaussian basis functions:

$$\phi_j(x) = \exp\left\{-\frac{(x-\mu_j)^2}{2s^2}\right\}$$

- These are local;
  - a small change in x only affect nearby basis functions.
  - μ_j and s control location and sc ale (width).



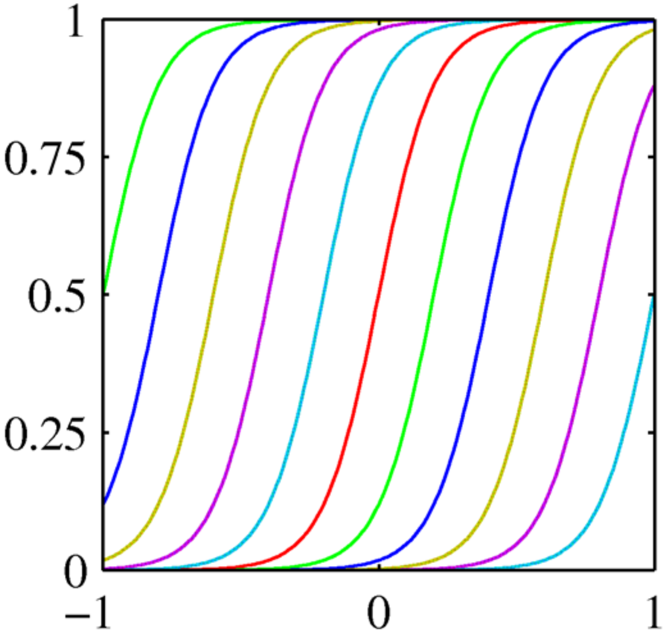
LINEAR BASIS FUNCTION MODELS (5)

- Sigmoidal basis functions:

$$\phi_j(x) = \sigma\left(\frac{x - \mu_j}{s}\right)$$

where

$$\sigma(a) = \frac{1}{1 + \exp(-a)}.$$



- Also these are local;
  - a small change in x only affect nearby basis functions.
  - μ_j and s control location and scale (width).

MAXIMUM LIKELIHOOD AND LEAST SQUARES (1)

- Assume observations from a deterministic function with added Gaussian noise:

$$t = y(\mathbf{x}, \mathbf{w}) + \epsilon \qquad \text{where} \qquad p(\epsilon|\beta) = \mathcal{N}(\epsilon|0, \beta^{-1})$$

- which is the same as saying,

$$p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1}).$$

- Given observed inputs, $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and targets, $\mathbf{t} = [t_1, \ldots, t_N]^{\mathrm{T}}$, we obtain the likelihood function

likelihood function

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}).$$

MAXIMUM LIKELIHOOD AND LEAST SQUARES (2)

$$N\left(t_n \middle| \boldsymbol{w}^{\boldsymbol{T}} \boldsymbol{\phi}(\mathbf{x_n}), \beta^{-1}\right) =$$
$$\left(\frac{\beta}{2\pi}\right)^{\frac{1}{2}} \exp\left(-\frac{\beta}{2}\left(t_n - \boldsymbol{w}^{\boldsymbol{T}} \boldsymbol{\phi}(\mathbf{x_n})\right)^2\right)$$

- Log likelihood:

$$\begin{aligned} \ln p(\mathbf{t}|\mathbf{w}, \beta) &= \sum_{n=1}^{N} \ln \mathcal{N}(t_n|\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}) \\ &= \boxed{\frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi) - \beta E_D(\mathbf{w})} \end{aligned}$$

Relationship of log likelihood and sum-of-squares error in univariate Gaussian noise model.

where

$$E_D(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2$$

is the sum-of-squares error.

REGULARIZED LEAST SQUARES (1)

- Consider the error function:

$$E_D(\mathbf{w}) + \lambda E_W(\mathbf{w})$$

Data term + Regularization term

- With the sum-of-squares error (SSE) function and a quadratic regularizer, we get

$$\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

- which is minimized by

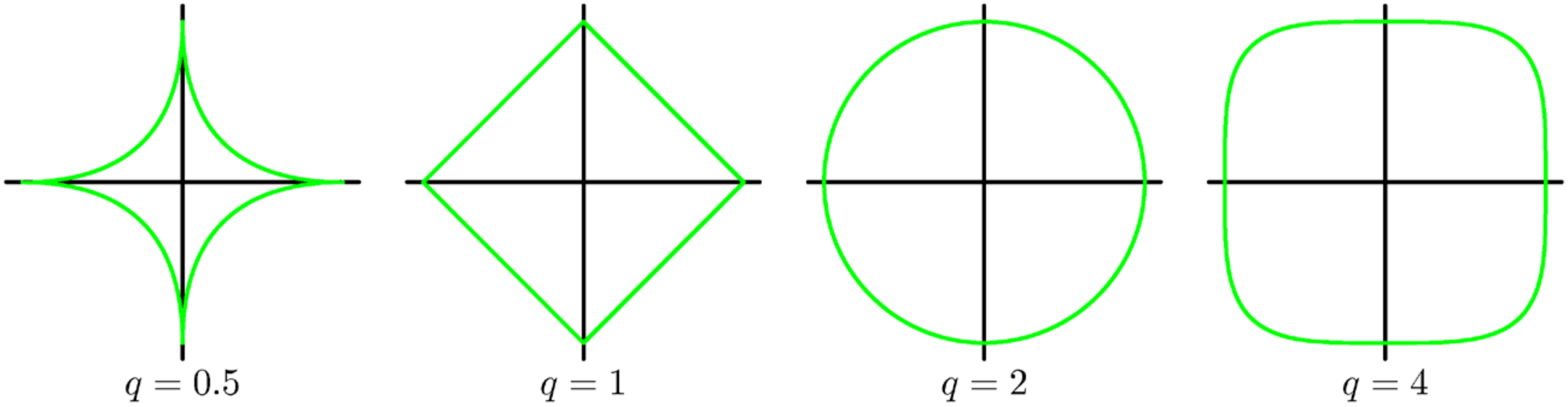
$$\mathbf{w} = \left(\lambda\mathbf{I} + \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}.$$

λ is called the regularization coefficient.

REGULARIZED LEAST SQUARES (2)

- With a more general regularizer, we have

$$\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\sum_{j=1}^{M}|w_j|^q$$



Lasso Quadratic

Fig: Contours of the regularization terms

USING LASSO FOR FEATURE SELECTION

Lasso tends to generate sparser solutions

- If λ is sufficiently large, some of the coefficients w_j are driven to zero, leading to a sparse model in which the corresponding basis function pays no role.

Minimizing

general regularizer

$$\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\sum_{j=1}^{M}|w_j|^q$$

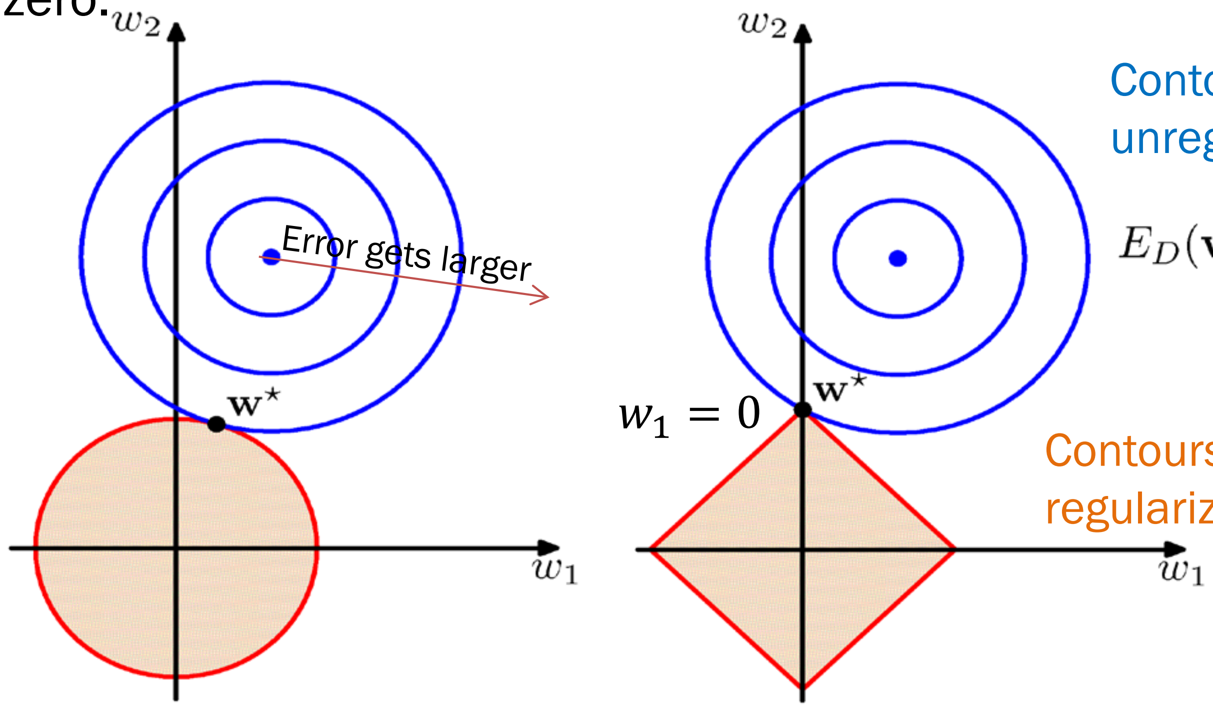
is equivalent to minimizing the unregularized SSE subjected to constraint

Lagrangian Multiplier

$$E_D(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n)\}^2 \text{ Subjected to } \sum_{j=1}^{M}|w_j|^q \leq \eta$$

REGULARIZED LEAST SQUARES (3)

Figure shows the minimum of the error function, subjected to constraint. As λ is increased, so an increasing number of parameters are driven to zero.



Contours of unregularized SSE

$$E_D(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2$$

Contours of the regularization terms

$$\sum_{j=1}^{M} |w_j|^q \leq \eta$$

Quadratic

Lasso

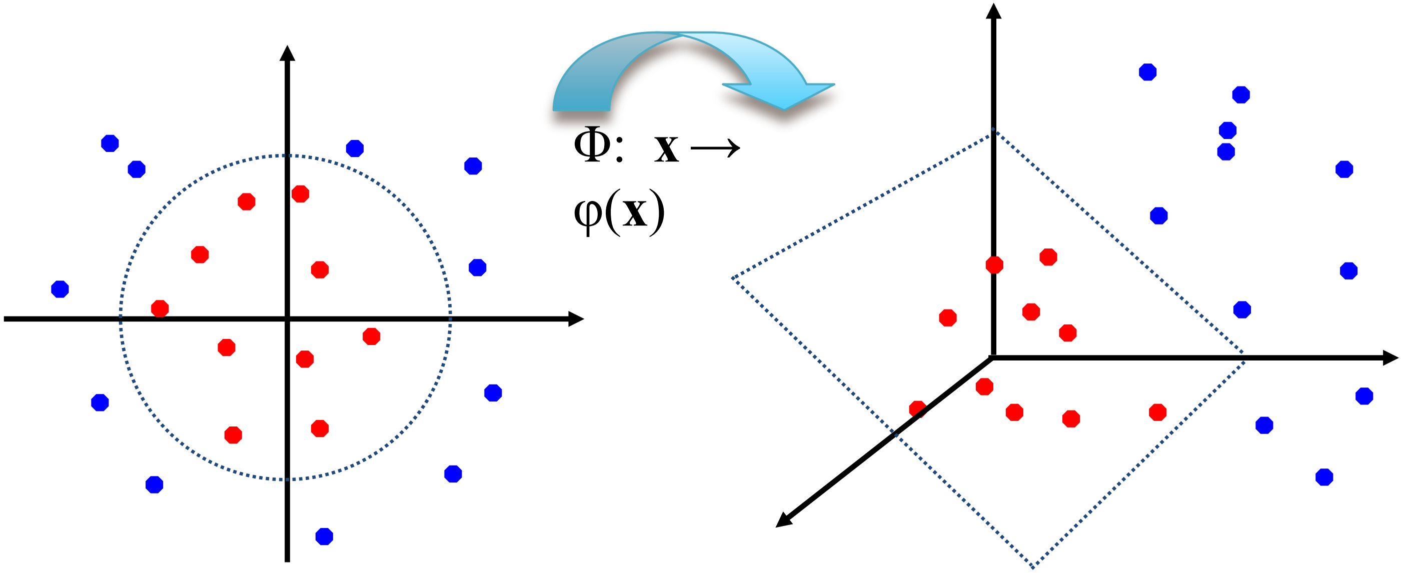
Lasso give sparse solution in which $w^* = 0$.

Q: So, how do we find the right λ?

SUPPORT VECTOR MACHINES

KERNELS

- The original feature space can always be mapped to some higher-dimensional feature space (even infinite) where the training set is separable



Φ: $\mathbf{x} \rightarrow \varphi(\mathbf{x})$

KERNELS

- The linear classifier relies on an inner product between vectors $K(\mathbf{x}_i,\mathbf{x}_j)=\mathbf{x}_i^T\mathbf{x}_j$
- If every data point is mapped into high-dimensional space via some transformation Φ: $\mathbf{x} \rightarrow \varphi(\mathbf{x})$, the inner product becomes:
$$K(\mathbf{x}_i,\mathbf{x}_j)= \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j)$$
- A *kernel function* is some function that corresponds to an inner product in some expanded feature space.

- Kernel function should measure some similarity between data
- kernel must be positive semi-definite

- You should scale the features to have same scale!!

- Most widely used is **linear kernels** and **Gaussian kernels**

GAUSSIAN KERNELS

$$k(x_i, x_j) = \exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right) = \exp\left(-\frac{\sum_{k=1}^{n}(x_{ik} - x_{jk})^2}{2\sigma^2}\right)$$

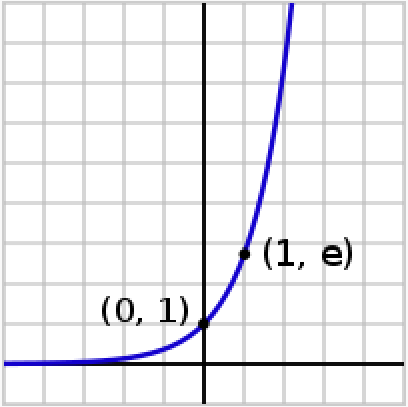
If x_i and x_j is similar:

$$k(x_i, x_j) \approx \exp\left(-\frac{0^2}{2\sigma^2}\right) \approx 1$$

If x_i and x_j is different:

$$k(x_i, x_j) \approx \exp\left(-\frac{(large\ number)^2}{2\sigma^2}\right) \approx 0$$

If you use Gaussian kernel,
You will need to pick σ



The natural exponential function
$y = e^x$

SUPPORT VECTOR MACHINES

- SVMs constructs a maximum margin separator
- SVMs create a linear separating hyperplane
  - But have ability to embed that in to higher-dimensional space (via **Kernel trick**)
- SVM are a nonparametric method
  - Retain training examples an potentially need to store all or part of the data
  - Some example are more important then others (support vectors)

SVM TERMS

- Distance from example $\mathbf{x}_i$ to the separator is

$$r = {}^{(\mathrm{w}^T\mathrm{x}+\mathrm{b})}\!/_{||\mathrm{w}||}$$

- Examples closest to the hyperplane are *support vectors*.
- *Margin* ρ of the separator is the distance between support vectors

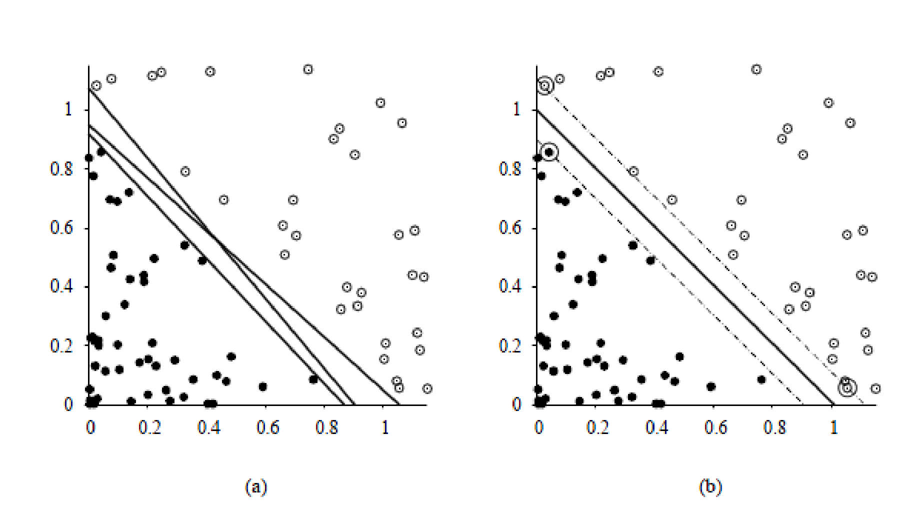
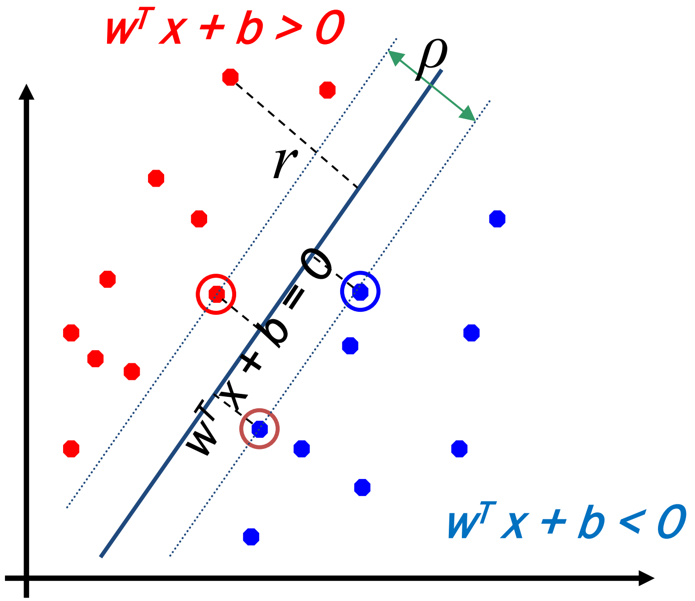


Figure 18.30 FILES: . Support vector machine classification: (a) Two classes of points (black and white circles) and three candidate linear separators. (b) The maximum margin separator (heavy line), is at the midpoint of the **margin** (area between dashed lines). The **support vectors** (points with large circles) are the examples closest to the separator.



MARGINS

Instead of minimizing expected empirical loss in the training data,
SVM attempts to minimize expected generalization loss.

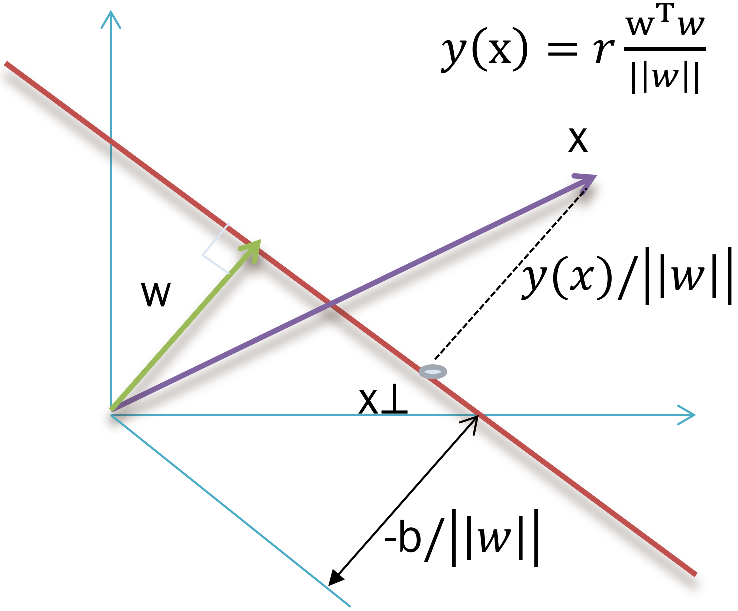
$y(\mathrm{x}) = \mathrm{w}^{\mathrm{T}}\mathrm{x} + b$ where w is weight vector and b is bias

$\mathrm{x} = \mathrm{x}\perp + r\frac{w}{||w||}$ (multiply w^{T} and add b)

$\mathrm{w}^{\mathrm{T}}\mathrm{x} + b = \mathrm{w}^{\mathrm{T}}(\mathrm{x}\perp + r\frac{w}{||w||}) + b$ $(y(\mathrm{x}) = \mathrm{w}^{\mathrm{T}}\mathrm{x} + b)$

$y(\mathrm{x}) = \mathrm{w}^{\mathrm{T}}\mathrm{x}\perp + r\frac{\mathrm{w}^{\mathrm{T}}w}{||w||} + b$ $(y(\mathrm{x}\perp) = \mathrm{w}^{\mathrm{T}}\mathrm{x}\perp + b = 0)$

$y(\mathrm{x}) = r\frac{\mathrm{w}^{\mathrm{T}}w}{||w||}$ = I



or $r = {}^{(\mathrm{w}^{T}\mathrm{x+b})}/_{||\mathrm{w}||}$

MAXIMUM MARGINS

$\phi(\mathrm{x_n})$ in the feature space

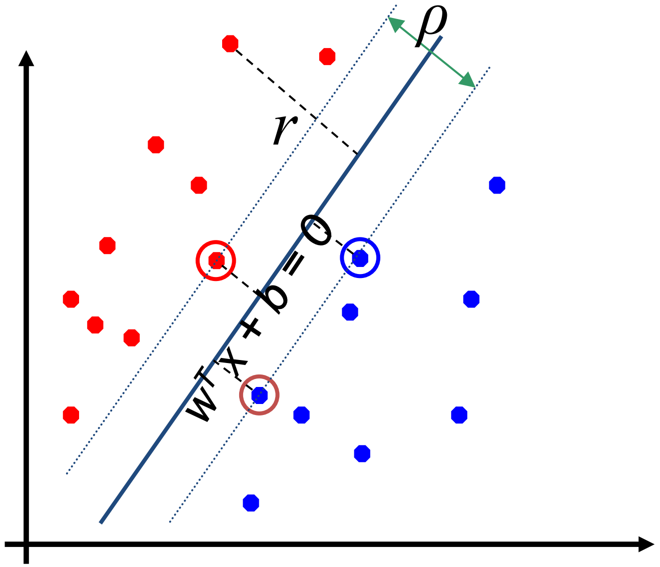
$$r = {}^{(\mathrm{w}^T\mathrm{x+b})}/_{||\mathrm{w}||}$$

$$argmax_{w,b}\,\{\frac{1}{||w||}\,min_n[t_n(\mathrm{w^T x_n} + b)]\}$$

$$\boxed{argmin_{w,b}\,\frac{1}{2}||w||^2}$$

$$\mathrm{w} = \sum\nolimits_{n=1}^{N} a_n t_n\,\phi(\mathrm{x_n})$$

$$\sum\nolimits_{n=1}^{N} a_n t_n = 0$$



Solving this is non-trivial and will not be discussed in class

SOFT MARGINS

Idea: Allow data point to be in the wrong side of the margin boundary, but with a penalty that increases with the distance from that boundary.

Penalty for each data point : slack variable ξ

$\xi_n = 0$ if point is on the right side

$\xi_n = |t_n - y(\mathrm{x}_n)|$ if point is on the wrong side

Such that

$t_n y(\mathrm{x}_n) \geq 1 - \xi_n$ for n = 1, …, N and $\xi_n \geq 0$

- $0 < \xi_n \leq 1$ for points inside the margin
- $\xi_n = 1$ for points on the margin
- $\xi_n > 1$ for points that are on the wrong side

Goal now is to maximize the margin while softly penalizing points that lie on the wrong side of the margin boundary

$$\boxed{argmin_{w,b}\ C \sum_{n}^{N} \xi_n \quad + \frac{1}{2}||w||^2}$$

OPTIMIZATION ON SOFT MARGINS

$$argmin_{w,b}\ C\sum_{n}^{N}\xi_n \quad + \frac{1}{2}||w||^2$$

subjected to $t_n y(\mathrm{x}_n) \geq 1 - \xi_n$ for n = 1, …, N and $\xi_n \geq 0$

ξ_n: slack variable for training data x_n

Complex calculations
Lagrangian
Etc.

$$\mathrm{w} = \sum_{n=1}^{N} a_n t_n\,\phi(\mathrm{x_n})$$

$$\sum_{n=1}^{N} a_n t_n = 0$$

a_n = C - μ_n

a_n is Lagrangian multiplier related to w_n

μ_n is Lagrangian multiplier related to ξ_n

$$\mathrm{b} = \frac{1}{N_M}\sum_{n\in M}(t_n - \sum_{n\in S}(a_m t_m\,k(x_n x_m))$$

PREDICTION USING KERNELS

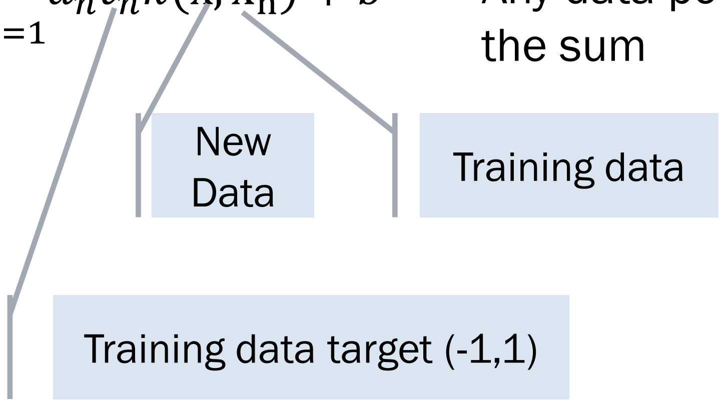
$$y(x) = \ \mathrm{w}^T \phi(\mathrm{x_n}) + b$$

$$\mathrm{w} = \sum_{n=1}^{N} a_n t_n \, \phi(\mathrm{x_n})$$

a_n is a Lagrangian multiplier

$$y(x) = \sum_{n=1}^{N} a_n t_n k(\mathrm{x}, \mathrm{x_n}) + b$$

Any data point $a_n = 0$ will not appear in the sum



New Data

Training data

Training data target (-1,1)

Instructor: Sael Lee

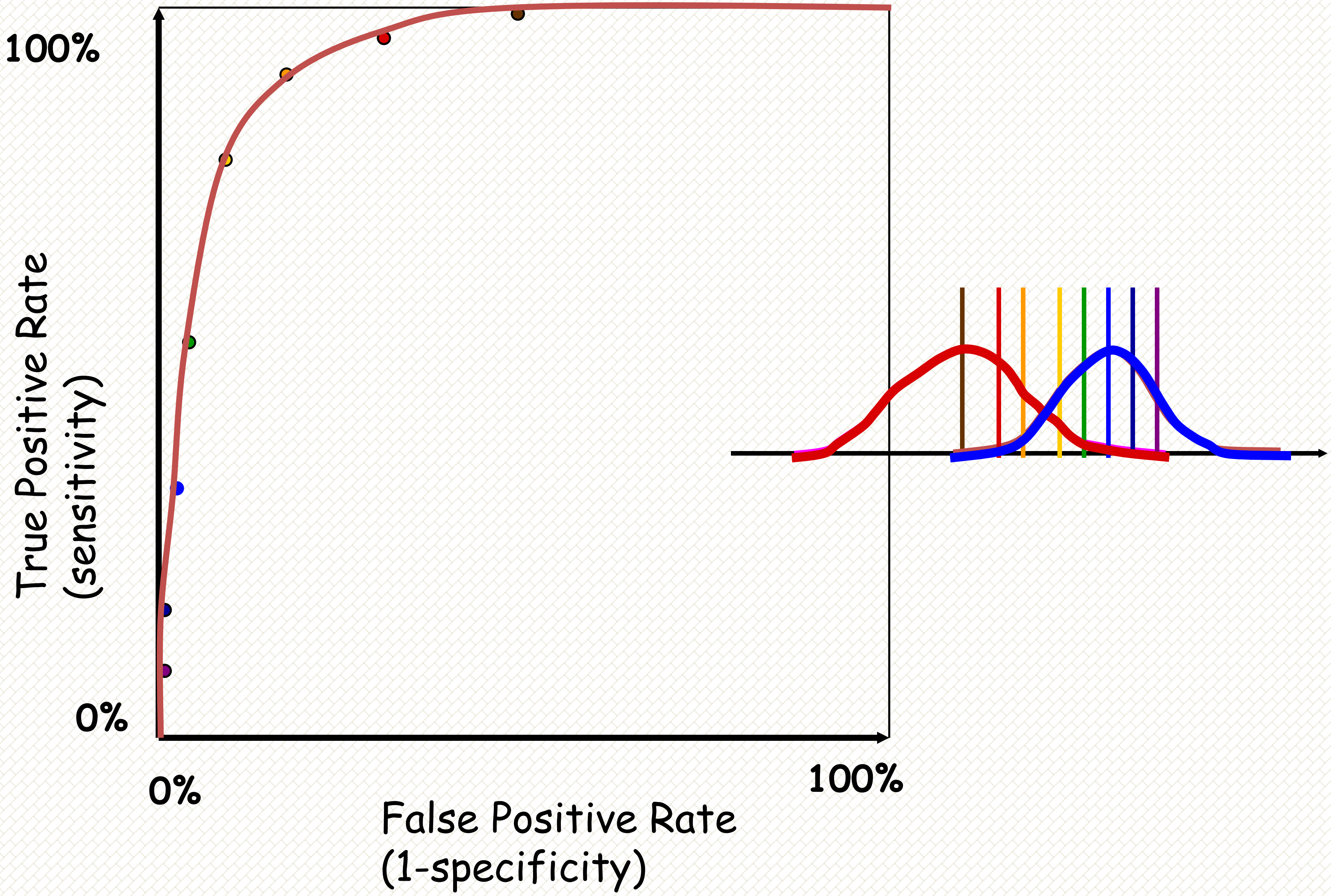
CS549 Spring – Computational Biology

LECTURE 14: BIOMARKER DISCOVERY WITH FEATURE SELECTION METHODS

Resources: .

- Abeel, T., Helleputte, T., Van de Peer, Y., Dupont, P., & Saeys, Y. (2010). Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics .26*(3), 392–8.
- Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene Selection for Cancer Classification using Support Vector Machines. *Machine Learning*, *46*(1-3), 389–422.

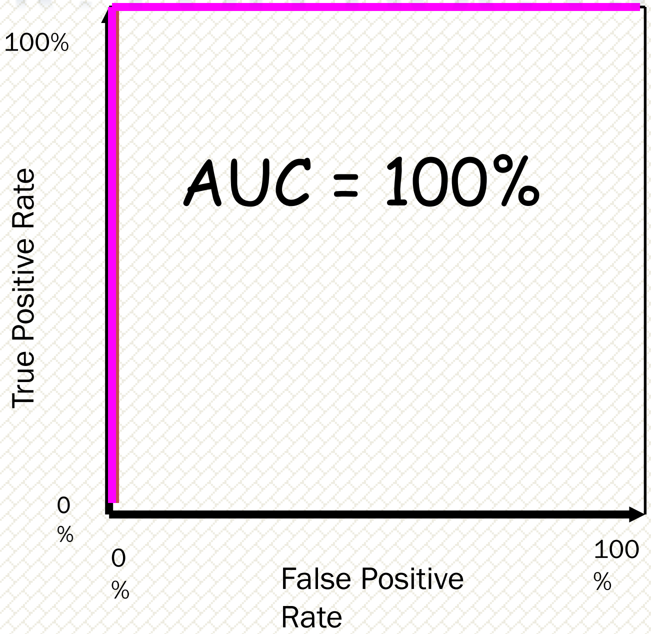
ROC CURVE

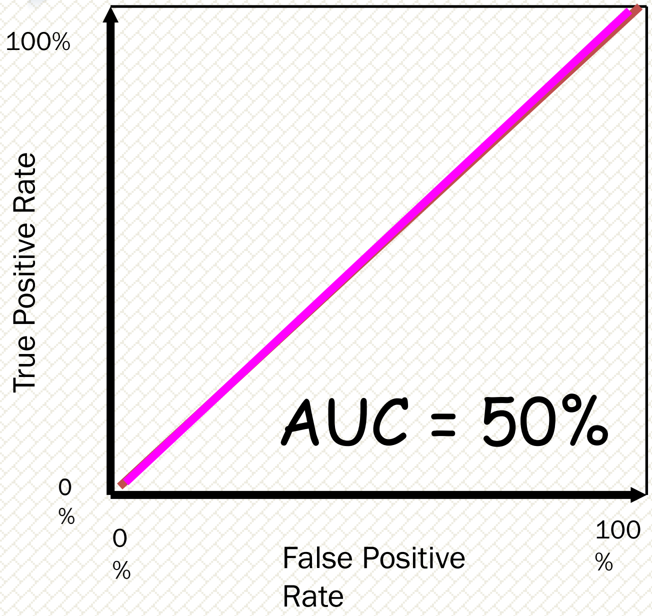

100%
True Positive Rate
(sensitivity)
0%
0%
100%
False Positive Rate
(1-specificity)

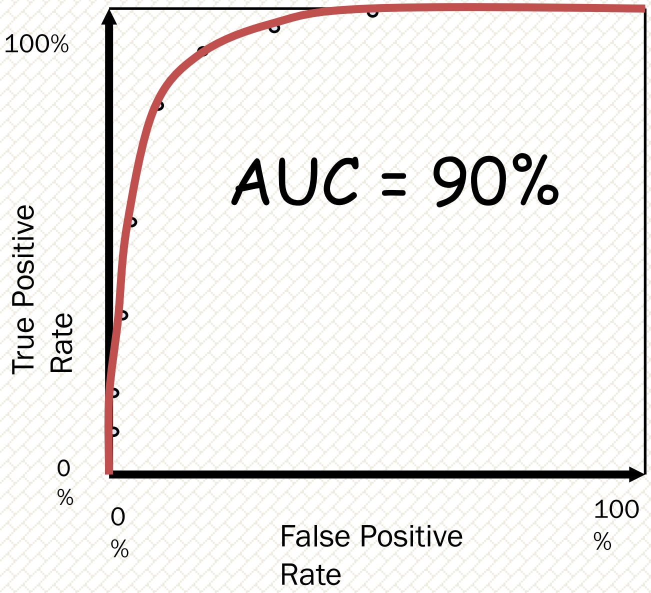
AREA UNDER ROC CURVE (AUC)

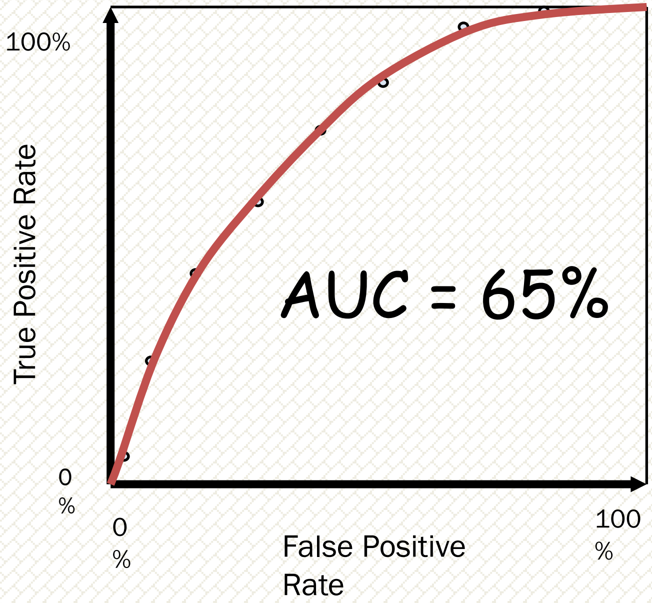
- *Overall measure* of test performance
- *Comparisons* between two tests based on differences between (estimated) AUC
- For continuous data, AUC equivalent to *Mann-Whitney U-statistic* (nonparametric test of difference in location between two populations)

AUC FOR ROC CURVES









PROBLEMS WITH AUC

- *No clinically relevant meaning*
- A lot of the area is coming from the range of *large false positive* values, no one cares what's going on in that region (need to examine restricted regions)
- The curves might *cross*, so that there might be a meaningful difference in performance that is not picked up by AUC

Instructor: Sael Lee
CS549 Spring – Computational Biology

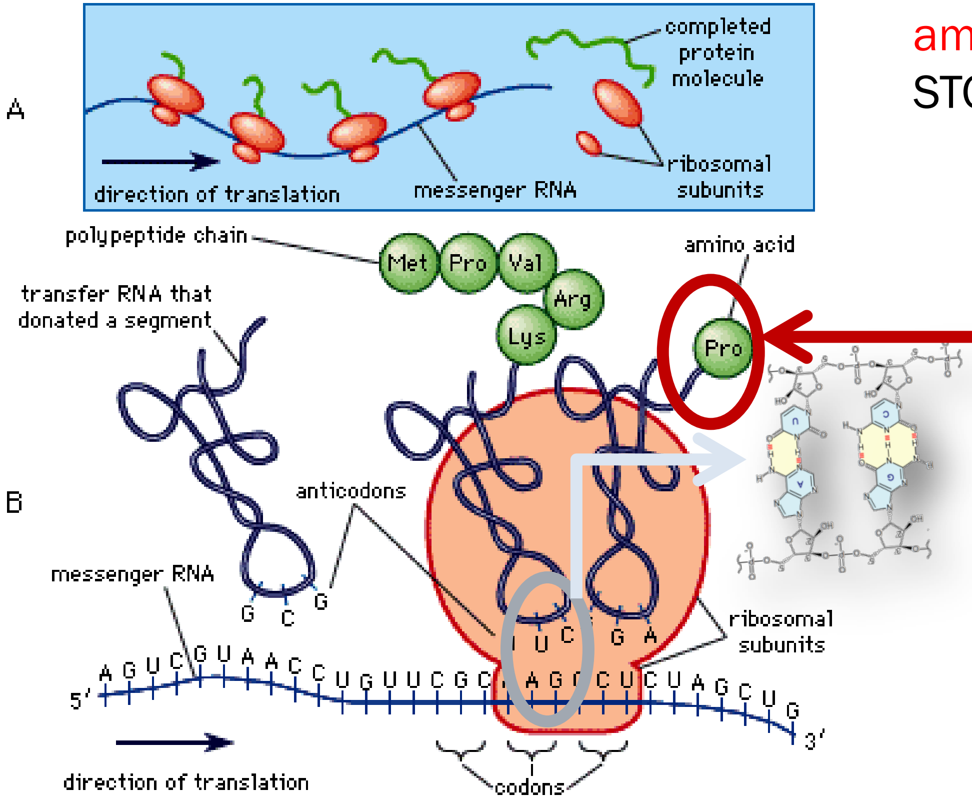
LECTURE 15:
ANALYZING PROTEIN STRUCTURE AND DYNAMICS

Resources:

- Slide 9 of Protein Bioinformatics, Spring 2013 Daisuke Kihara
- Wikipedia

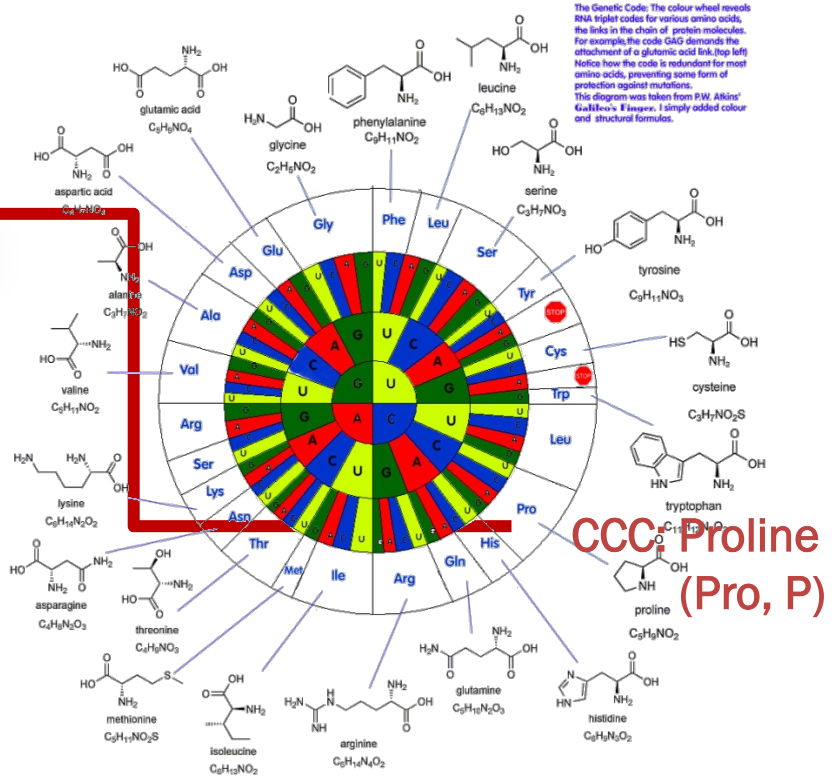
TRANSLATION PROCESS

Codon: Three nucleic acid coding one of 20 amino acid (alphabet of 20 size) + START & STOP CODEN



© 2006 Encyclopædia Britannica, Inc.

http://content.answcdn.com/main/content/img/BritannicaConcise/images/780.gif



The Genetic Code: The colour wheel reveals RNA triplet codes for various amino acids, the links in the chain of protein molecules. For example,the code GAG demands the attachment of a glutamic acid link.(top left) Notice how the code is redundant for most amino acids, preventing some form of protection against mutations.
This diagram was taken from P.W. Atkins' *Galileo's Finger*. I simply added colour and structural formulas.

CCC: Proline (Pro, P)

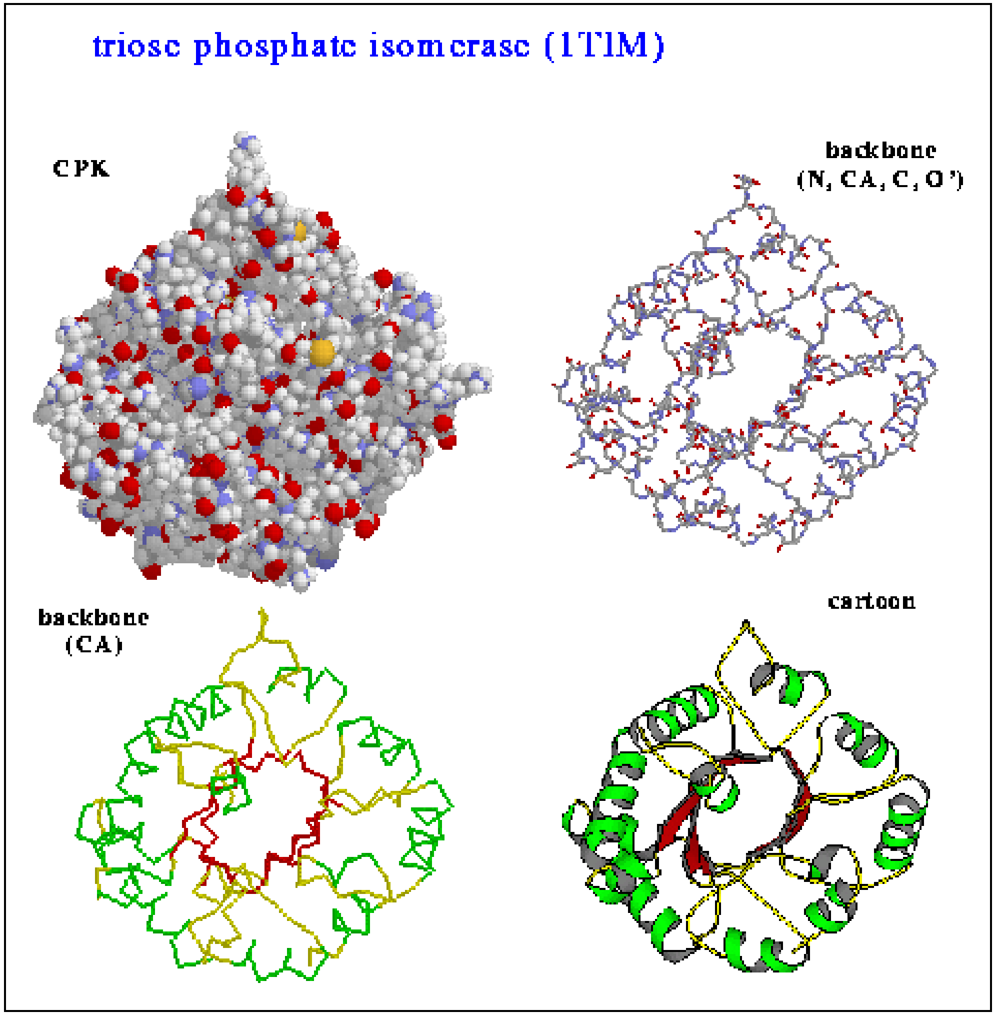
Start codon: AUG (also Methionine (Met, M))
Stop codon: UAA, UAG, UGA

PROTEIN STRUCTURE

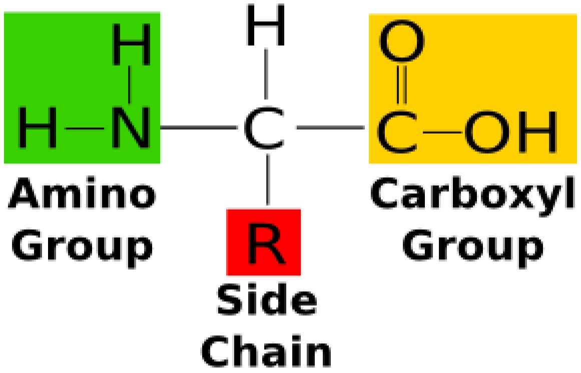
Amino Acid Sequence

```
APRKFFVGGNWKMNGDKKSLGELIHTLNGAKL
SADTEVVCGAPSIYLDFARQKLDAKIGVAAQN
CYKVPKGAFTGEISPAMIKDIGAAWVILGHSE
RRHVFGESDELIGQKVAHALAEGLGVIACIGE
KLDEREAGITEKVVFEQTKAIADNVKDWSKVV
LAYEPVWAIGTGKTATPQQAQEVHEKLRGWLK
SHVSDAVAQSTRIIYGGSVTGGNCKELASQHD
VDGFLVGGASLKPEFVDIINAKH
```

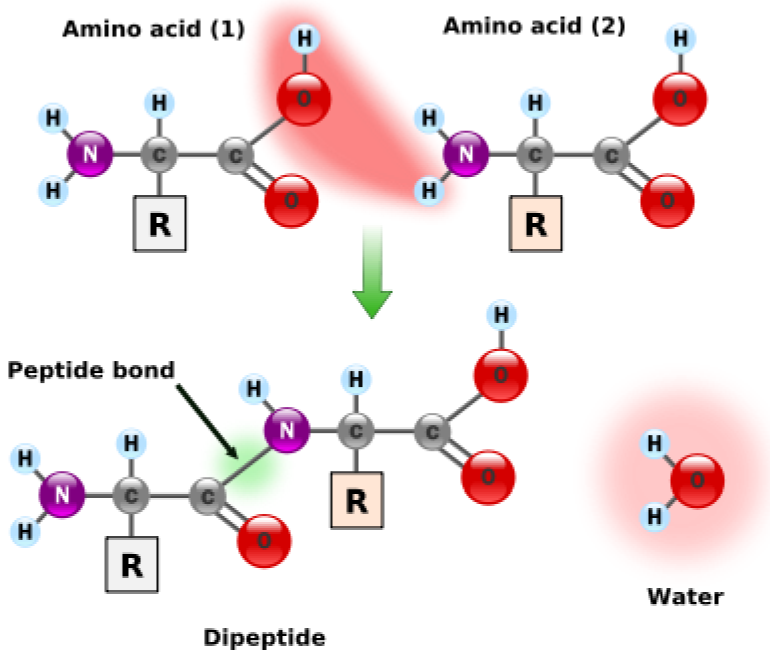
=



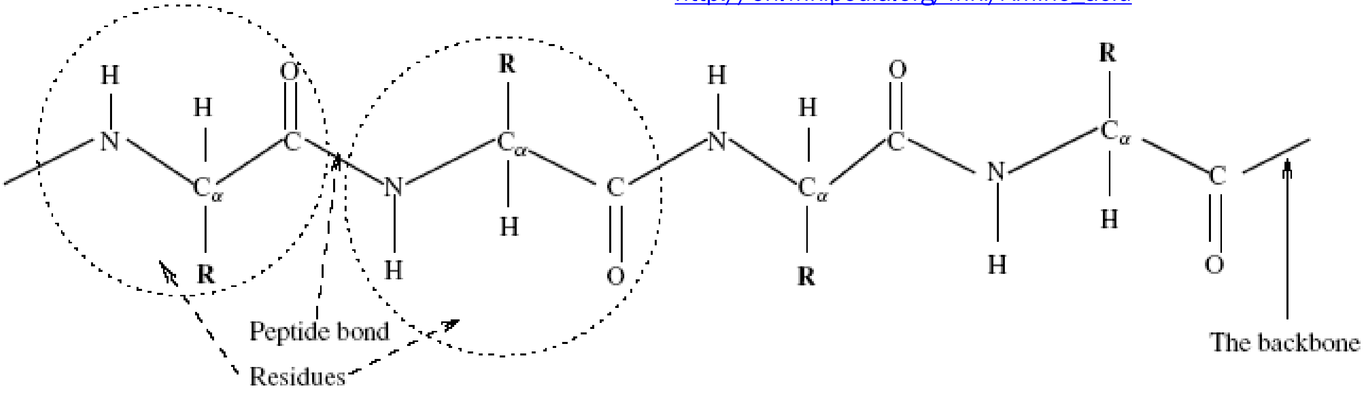
General Structure of AA



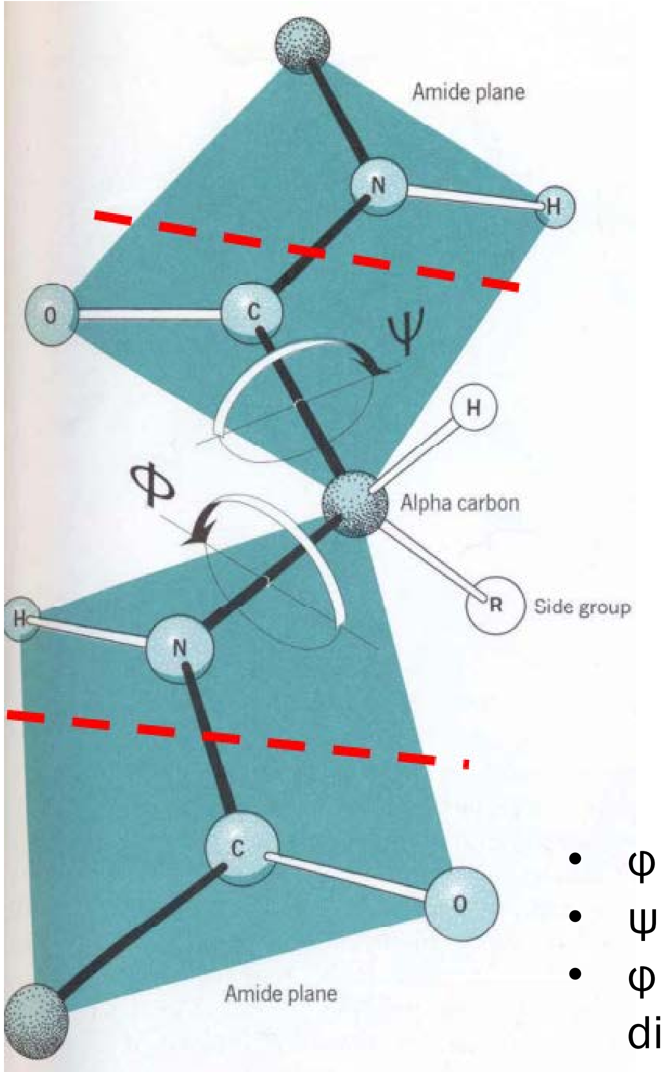
AMINO ACID AND MAIN CHAIN



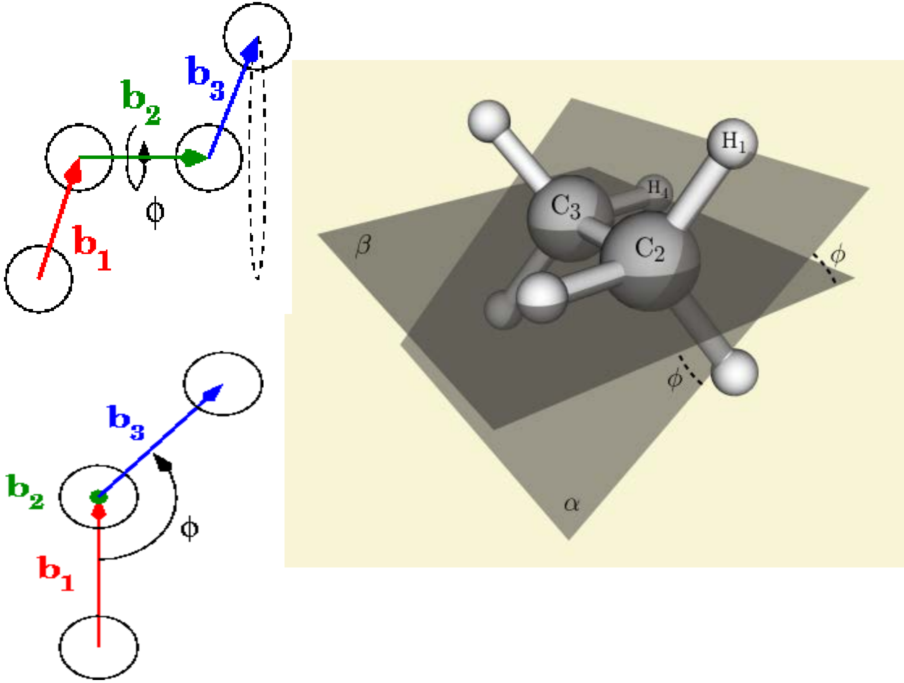
http://en.wikipedia.org/wiki/Amino_acid



DIHEDRAL ANGLES

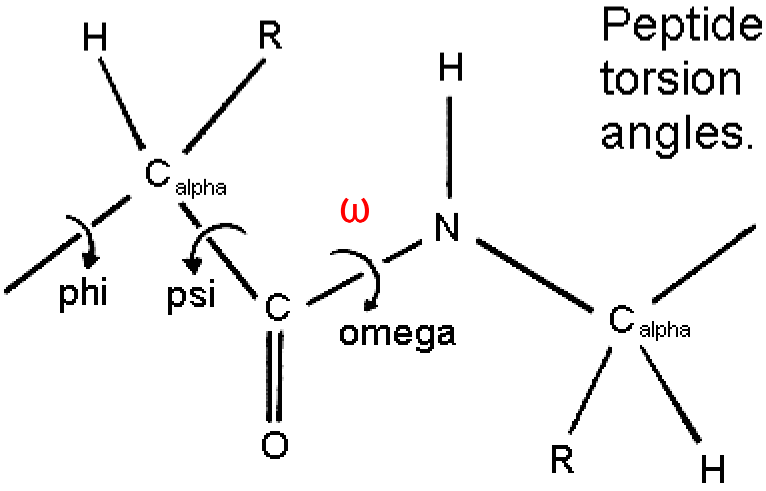


- Dihedral Angles (Torsion angles):
  Angels between two planes.

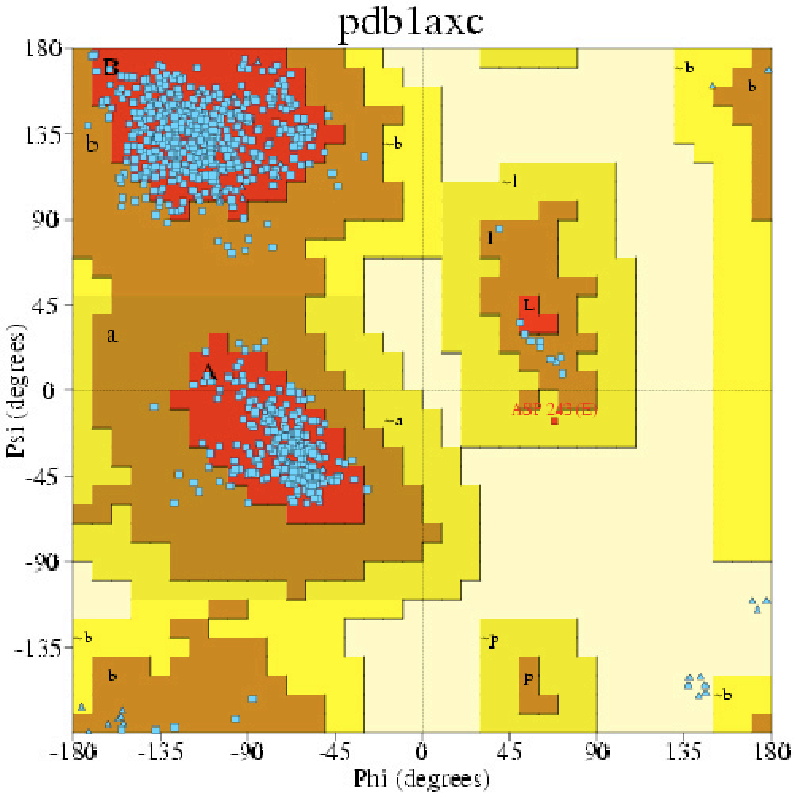


- φ (*phi*, involving the backbone atoms C'-N-C$^{\alpha}$-C')
- ψ (*psi*, involving the backbone atoms N-C$^{\alpha}$-C'-N)
- φ controls the C'-C' distance, ψ controls the N-N distance
- rotations about φ and ψ angles are the softest

- ω (*omega*, involving the backbone atoms C^{α}-C'-N-C^{α}).
- ω controls the C^{α}-C^{α} distance
- Peptide bond usually restricts ω to be 180° (the typical trans case) or 0° (the rare cis case).



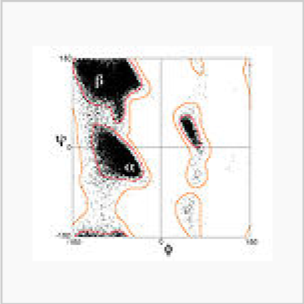
RAMACHANDRAN PLOT



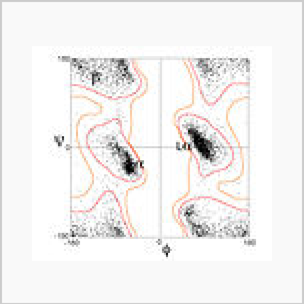
The red, brown, and yellow regions represent the favored, allowed, and "generously allowed" regions as defined by ProCheck

A **Ramachandran plot**

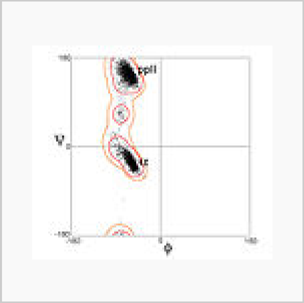
Is a visualization tools for visualizing backbone dihedral angles ψ against φ of amino acid residues in protein structure.



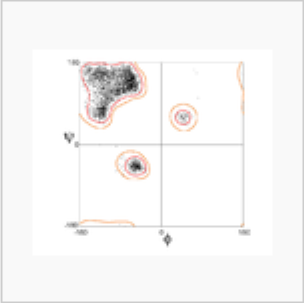
Ramachandran plot for the general case; data from Lovell 2003



Ramachandran plot for Glycine



Ramachandran plot for Proline



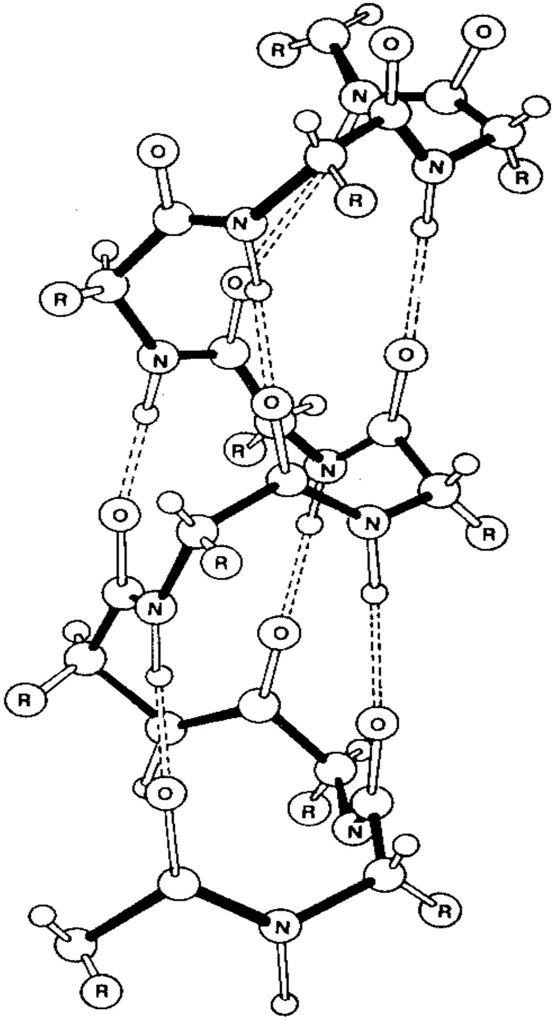
Ramachandran plot for pre-Proline

https://en.wikipedia.org/wiki/Ramachandran_plot

PROTEIN SECONDARY STRUCTURES

- Proteins packs the hydrophobic side chains inside the molecule.
- Proteins have hydrophobic kernel and hydrophilic surface.
- The backbone is polar, hence hydrophilic.
- To neutralize this hydrophility there are hydrogen bindings between NH and CO on the backbone.
- This is done by constructing regular *secondary structures*
  - *Helices, alpha most usual*
  - *Beta sheets*

ALPHA HELIX



Alpha-helix:

- Right-handed helix
- 3.6 residues per helix turn
- Hydrogen bond between n and n+4

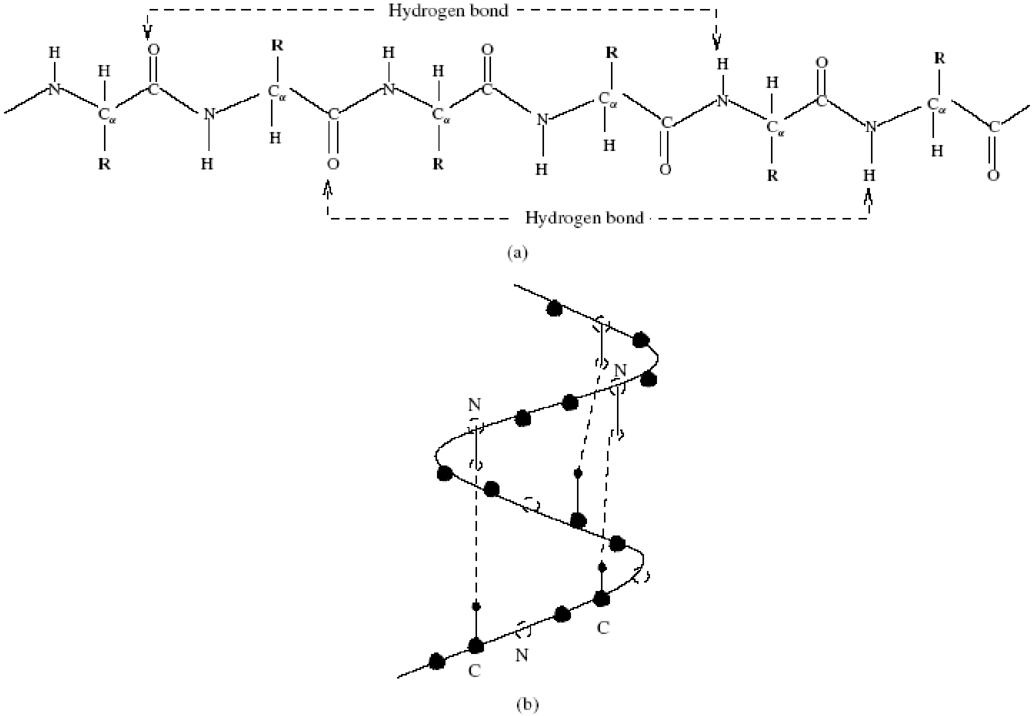


Figure B.6 (a) Schematic of the hydrogen bonding forming an α-helix. (b) For the hydrogen bonding to take place, the sequence must be formed as a helix in the space.

BETA SHEETS

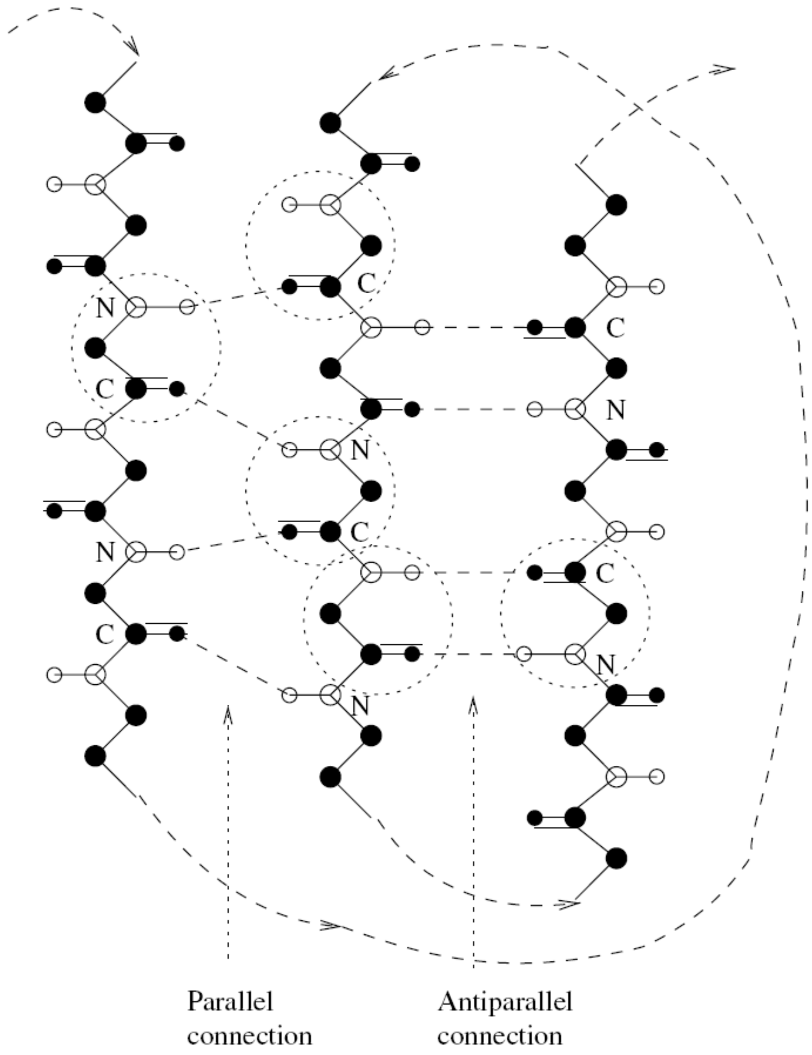
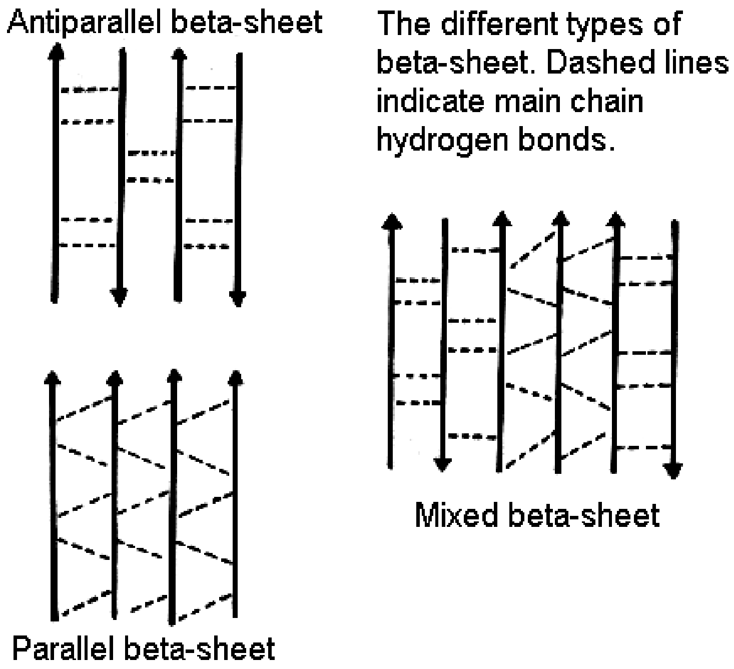


Figure B.7 A β-sheet formed of three β-strands, with one parallel and one antiparallel set of H-bonds. Note that strands near in space do not need to be near in sequence.



The different types of beta-sheet. Dashed lines indicate main chain hydrogen bonds.

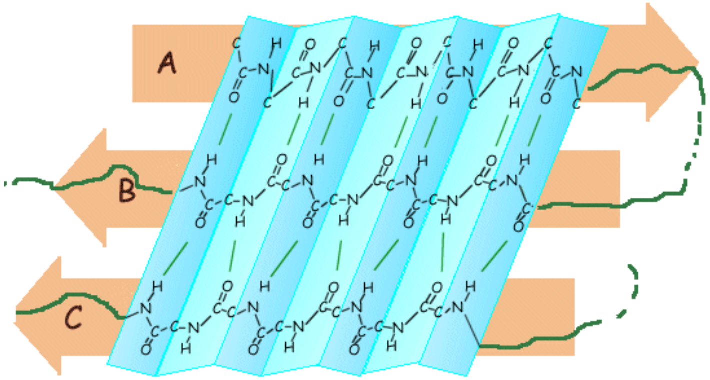
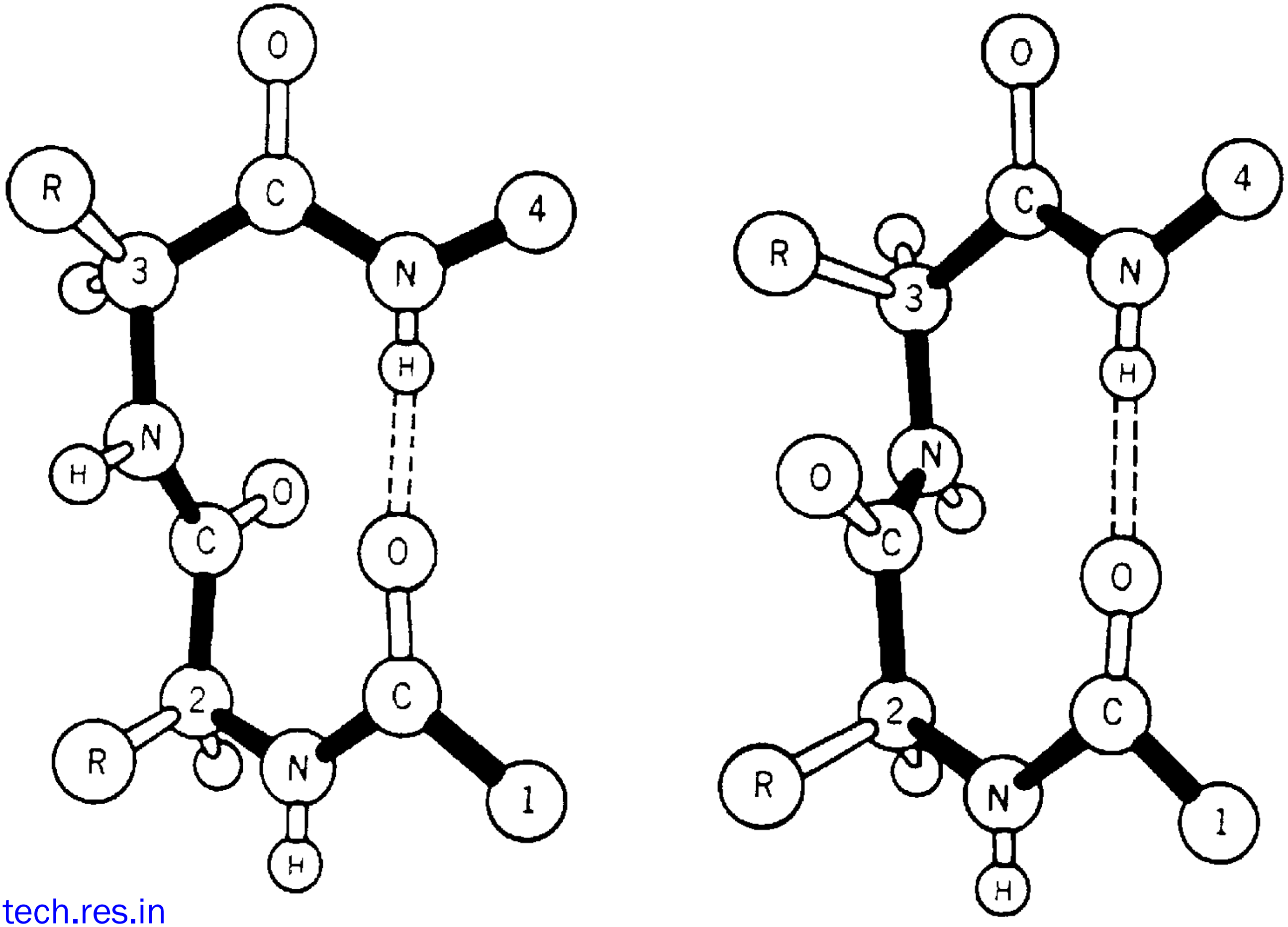


Diagram 1: Beta pleated sheet. The lateral groups (R) are not shown.

BETA-TURN

- 4 residues in length
- Enables structure to have an 180 degree turn



imtech.res.in

PROTEIN TERTIARY STRUCTURE

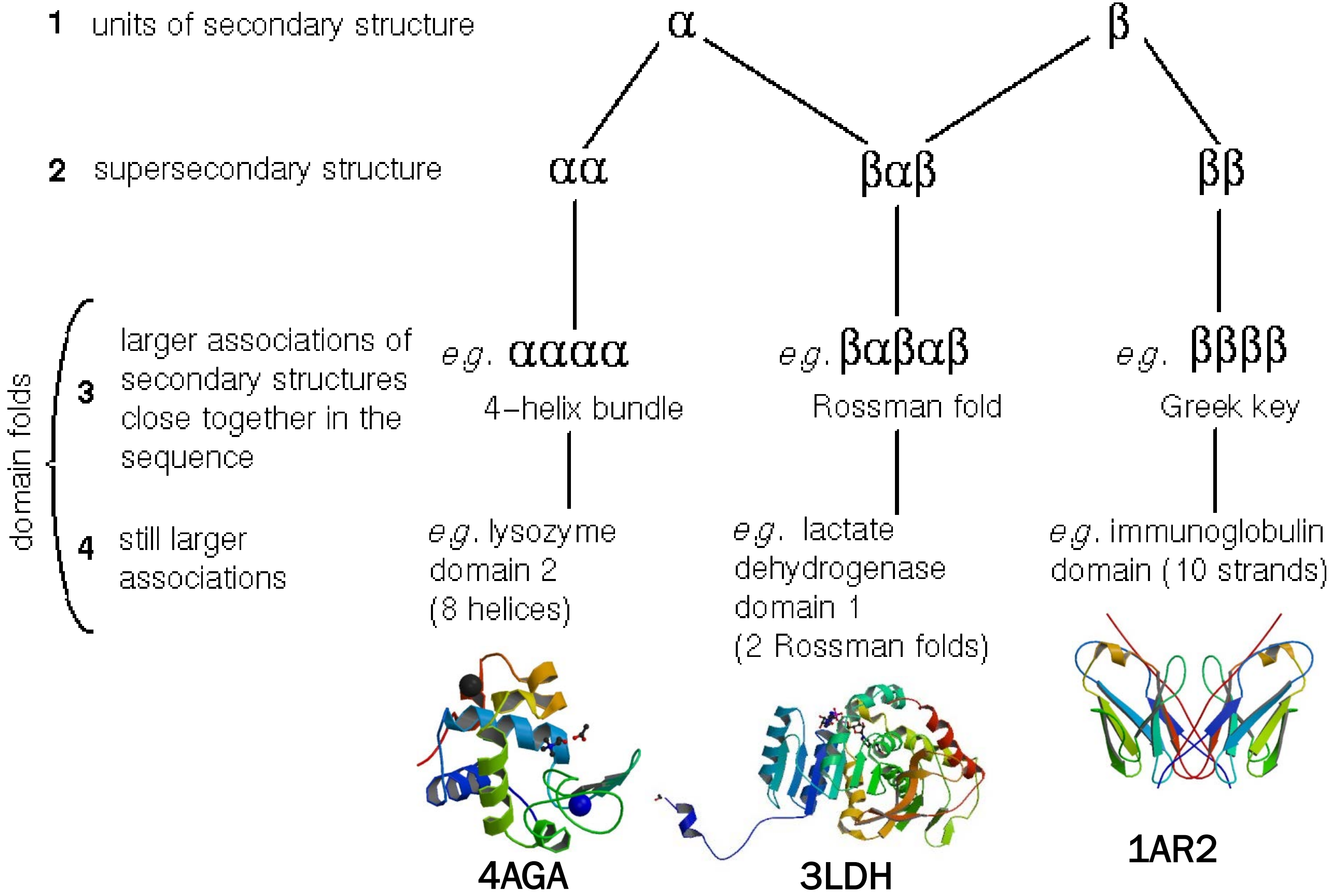
Driving force for folding:

- Hydrophobic effect
- Electrostatic
- Hydrogen bond
- Disulfide bond

PROTEIN STRUCTURE CLASSIFICATION - SCOP (STRUCTURAL CLASSIFICATION OF PROTEINS)

- Classes:
  - All alpha proteins (126)
  - All beta proteins(81)
  - Alpha and beta proteins (a/b) (87)
    Mainly parallel beta sheets (beta-alpha-beta units)
  - Alpha and beta proteins (a+b) (151)
    Mainly antiparallel beta sheets (segregated alpha and beta regions)
  - Multi-domain proteins (alpha and beta) (21)
    Folds consisting of more than one domain of different classes
  - Membrane and cell surface proteins and peptides (10)
    Does not include proteins in the immune system
  - Small proteins (44)
    Usually dominated by metal ligand, heme, and/or disulfide bridges
  - Coiled coil proteins (4)
  - Low resolution protein structures (4)
  - Peptides (61)
    Peptides and fragments
  - Designed proteins (17)
    Experimental structures of proteins with essentially non-natural sequences

SCOP CONT.



1 units of secondary structure

2 supersecondary structure

domain folds

3 larger associations of secondary structures close together in the sequence

4 still larger associations

α

β

αα

βαβ

ββ

e.g. αααα

4–helix bundle

e.g. βαβαβ

Rossman fold

e.g. ββββ

Greek key

e.g. lysozyme domain 2 (8 helices)

e.g. lactate dehydrogenase domain 1 (2 Rossman folds)

e.g. immunoglobulin domain (10 strands)

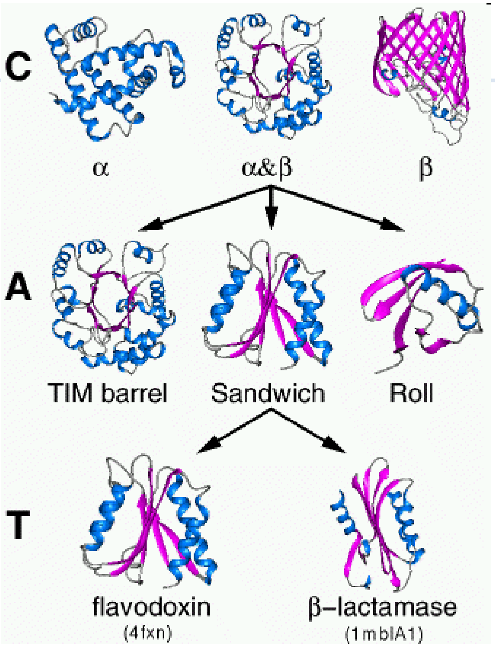
4AGA

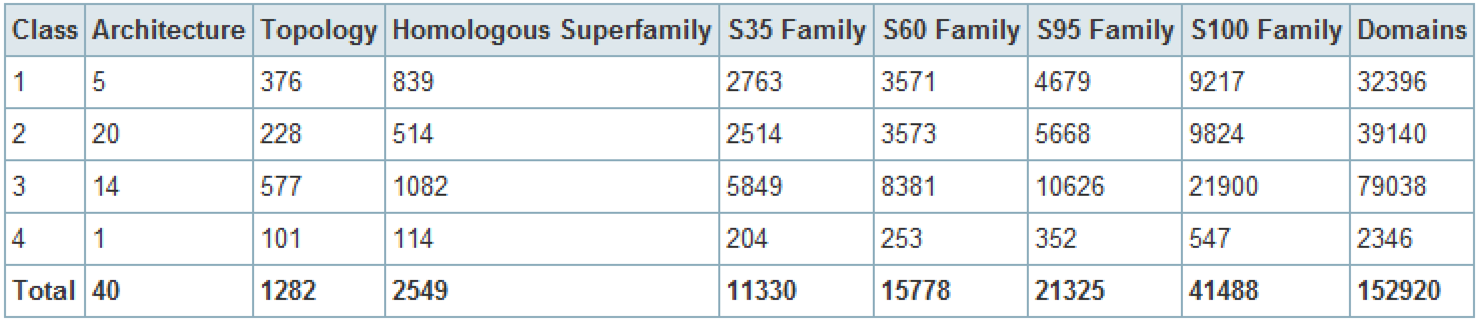
3LDH

1AR2

PROTEIN STRUCTURE CLASSIFICATION – CATH DATABASE

- Class, Architecture, Topology, Homology
  - Architecture: the global spatial arrangement of 2ndary structure segments
  - Topology: connectivity of the 2ndary structure segments is also counted
- Protein structure comparison program, SSAP is used





Class	Architecture	Topology	Homologous Superfamily	S35 Family	S60 Family	S95 Family	S100 Family	Domains
1	5	376	839	2763	3571	4679	9217	32396
2	20	228	514	2514	3573	5668	9824	39140
3	14	577	1082	5849	8381	10626	21900	79038
4	1	101	114	204	253	352	547	2346
Total	**40**	**1282**	**2549**	**11330**	**15778**	**21325**	**41488**	**152920**

CANNOT USE PURE DYNAMIC PROGRAMMING FOR STRUCTURE COMPARISON

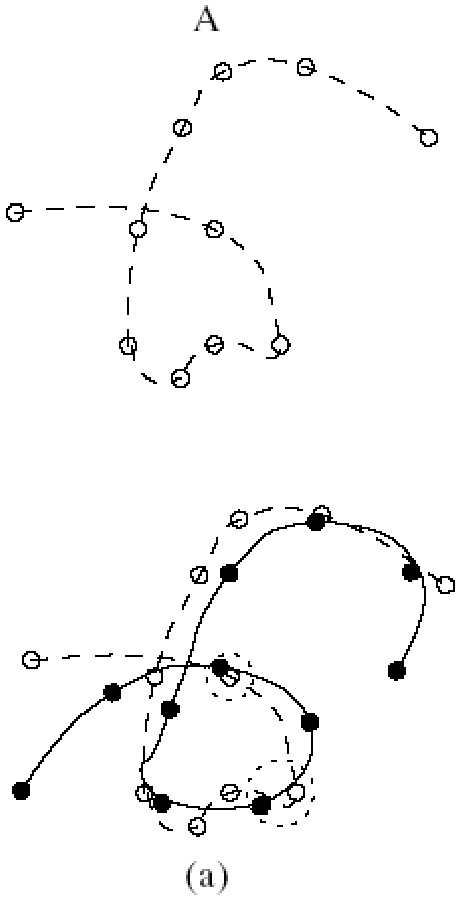
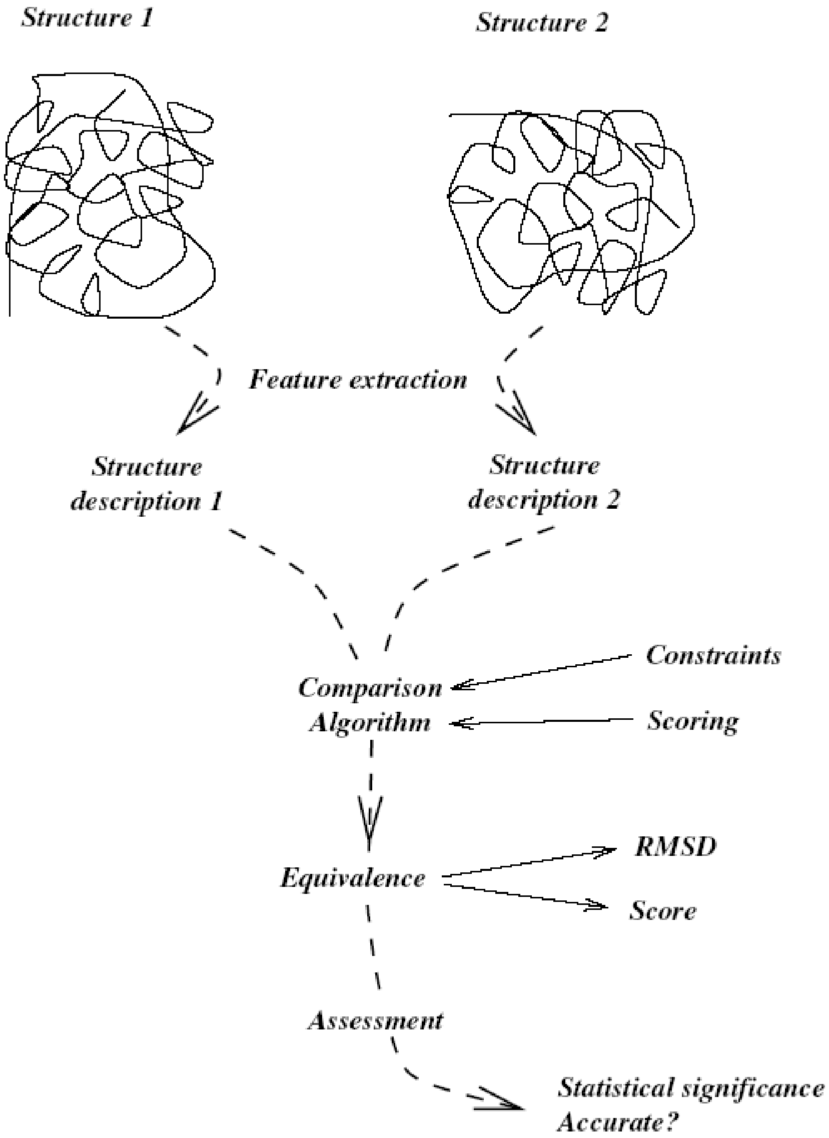


Figure 8.18 Illustration that dynamic programming cannot be used directly for structure alignment (see the text).

FRAMEWORK FOR PAIRWISE STRUCTURE COMPARISON



Structure 1

Structure 2

Feature extraction

Structure description 1

Structure description 2

Comparison Algorithm

Constraints

Scoring

Equivalence

RMSD

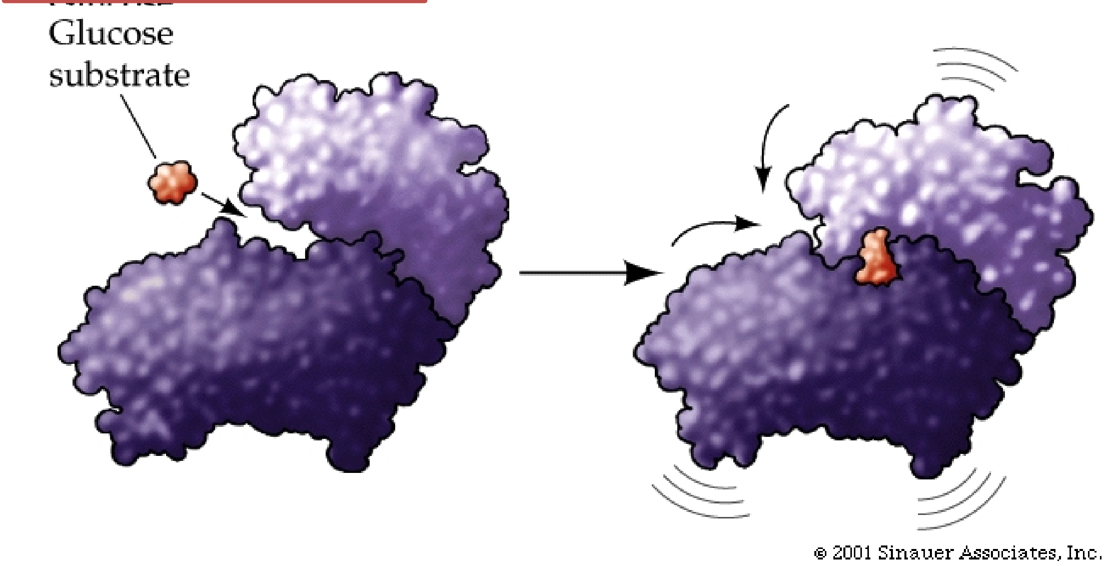
Score

Assessment

Statistical significance
Accurate?

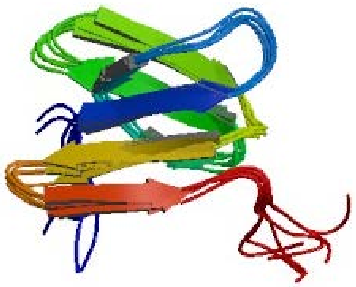
PROTEIN DYNAMICS

Induced fit model:

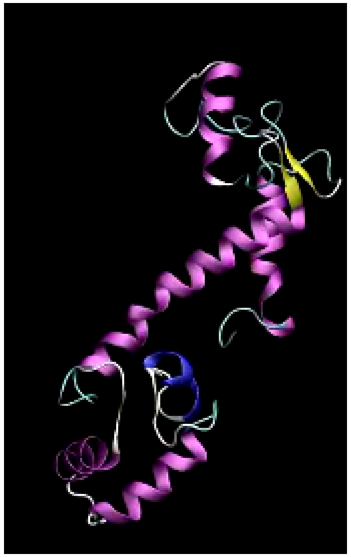


© 2001 Sinauer Associates, Inc.

1OQK



1AHR



Molecular Dynamics Extended Library:
http://mmb.pcb.ub.es/MoDEL/ :
test searching **1e5w** & **1AHR**

Instructor: Sael Lee

CS549 – Computational Biology

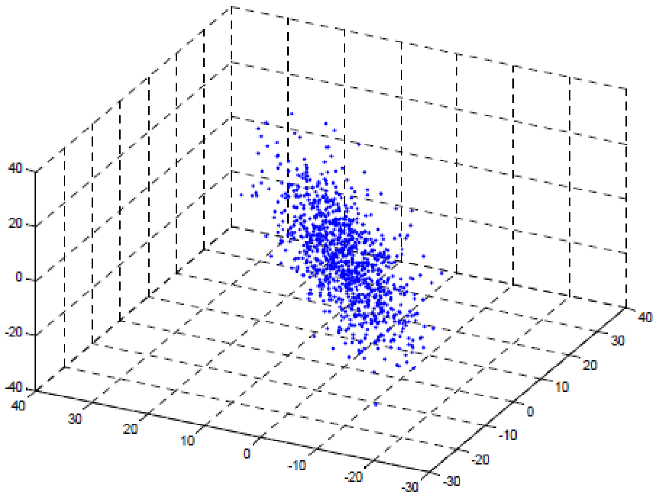
LECTURE 16: PCA AND SVD

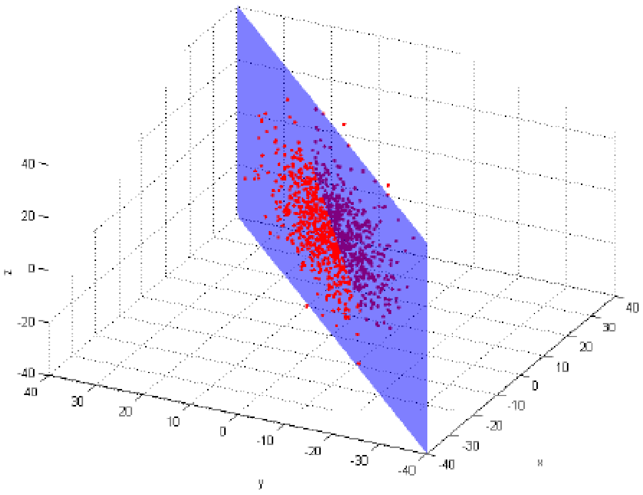
Resource:

- PCA Slide by Iyad Batal
- Chapter 12 of PRML
- Shlens, J. (2003). A tutorial on principal component analysis.

PRINCIPLE COMPONENT ANALYSIS

- PCA finds a **linear** projection of high dimensional data into a lower dimensional subspace such as:
  - The variance retained is maximized.
  - The least square reconstruction error is minimized





PCA STEPS

Linearly transform an $N \times d$ matrix X into an $N \times m$ matrix Y

- Centralized the data (subtract the mean).
- Calculate the $d \times d$ covariance matrix: $C = \frac{1}{N-1} X^T X$
  - $C_{i,j} = \frac{1}{N-1} \sum_{q=1}^{N} X_{q,i} X_{q,i}$
  - $C_{i,i}$ (diagonal) is the variance of variable i.
  - $C_{i,j}$ (off-diagonal) is the covariance between variables i and j.
- Calculate the eigenvectors of the covariance matrix (orthonormal).
- Select *m* eigenvectors that correspond to the largest *m* eigenvalues to be the new basis.

EIGENVECTORS

- If *A* is a **square matrix**, a non-zero vector **v** is an eigenvector of *A* if there is a scalar *λ* (eigenvalue) such that

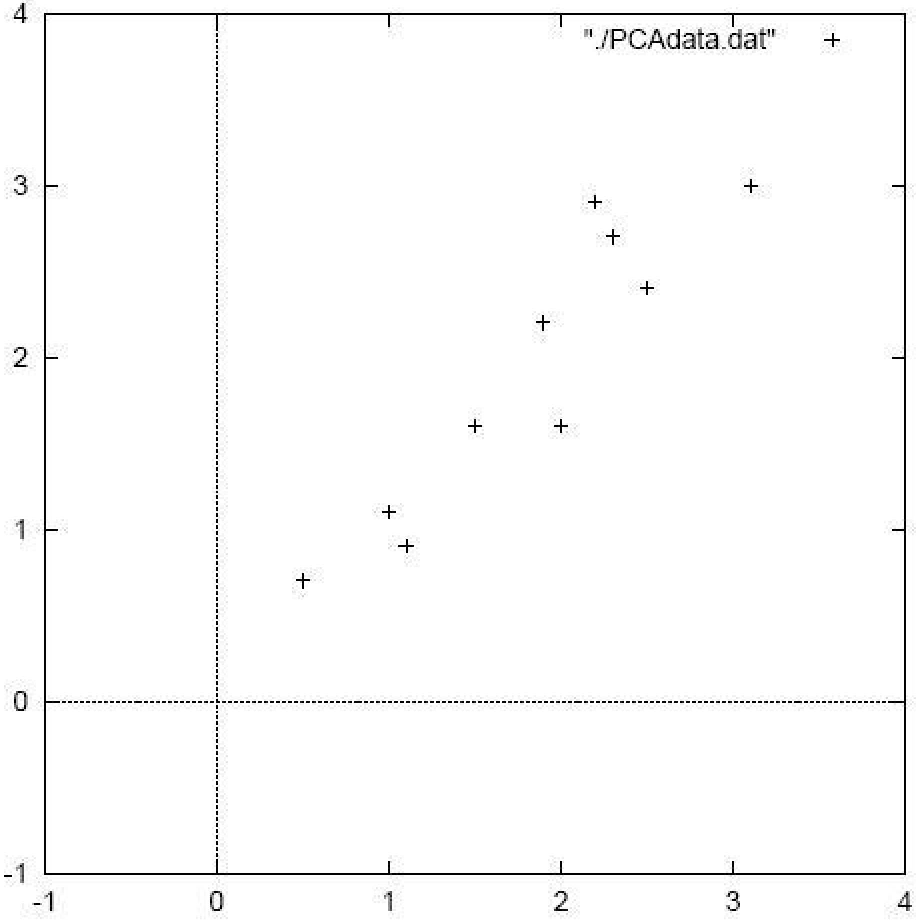
$$Av = \lambda v$$

- Example: $\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix}\begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 12 \\ 8 \end{pmatrix} = 4\begin{pmatrix} 3 \\ 2 \end{pmatrix}$
-

- If we think of the squared matrix *A* as a transformation matrix, then multiply it with the eigenvector do not change its direction.

- Step 1: subtract the mean and calculate the covariance matrix C.

$$C = \begin{pmatrix} 0.716 & 0.615 \\ 0.615 & 0.616 \end{pmatrix}$$



Step 2: Calculate the eigenvectors and eigenvalues of the covariance matrix:

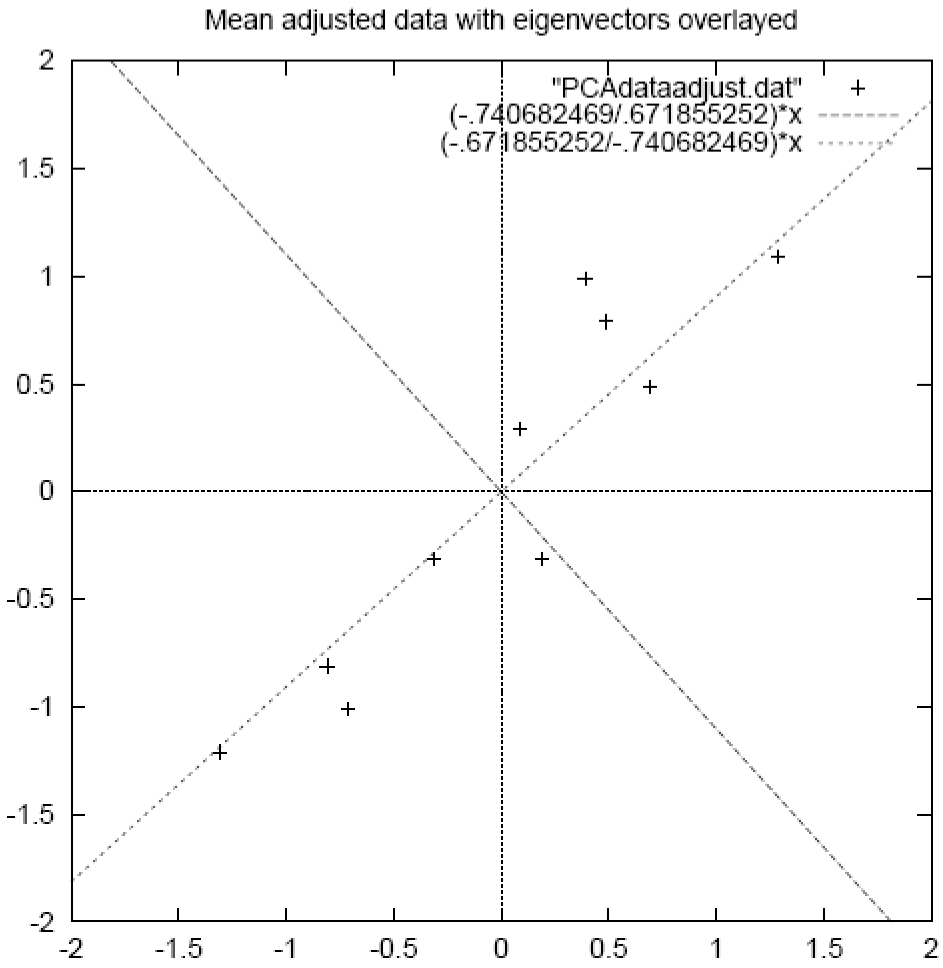
$\lambda_1 \approx 1.28$, $v_1 \approx [-0.677\ \ -0.735]^T$, $\lambda_2 \approx 0.49$, $v_2 \approx [-0.735\ \ 0.677]^T$

Notice that v1 and v2 are orthonormal:

$|v_1|=1$

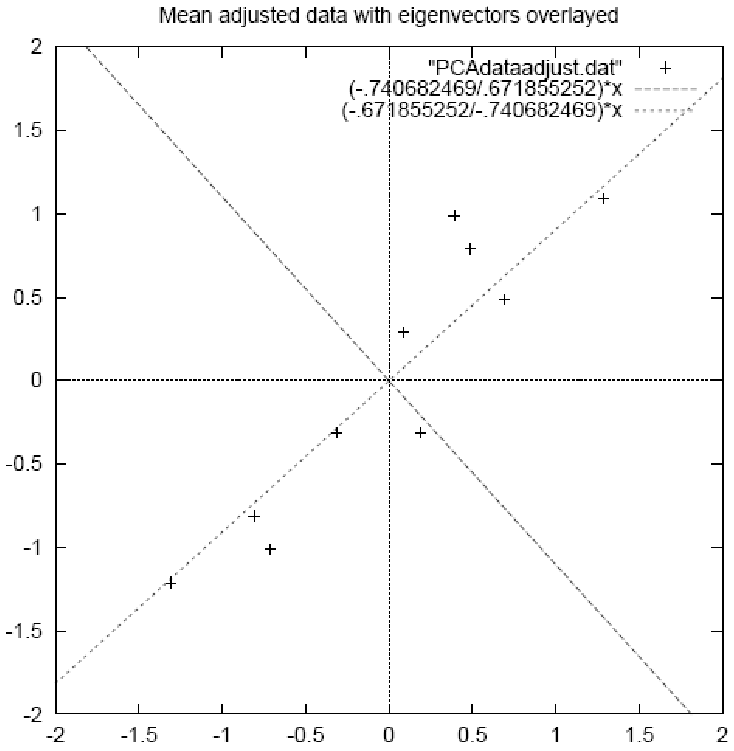
$|v_2|=1$

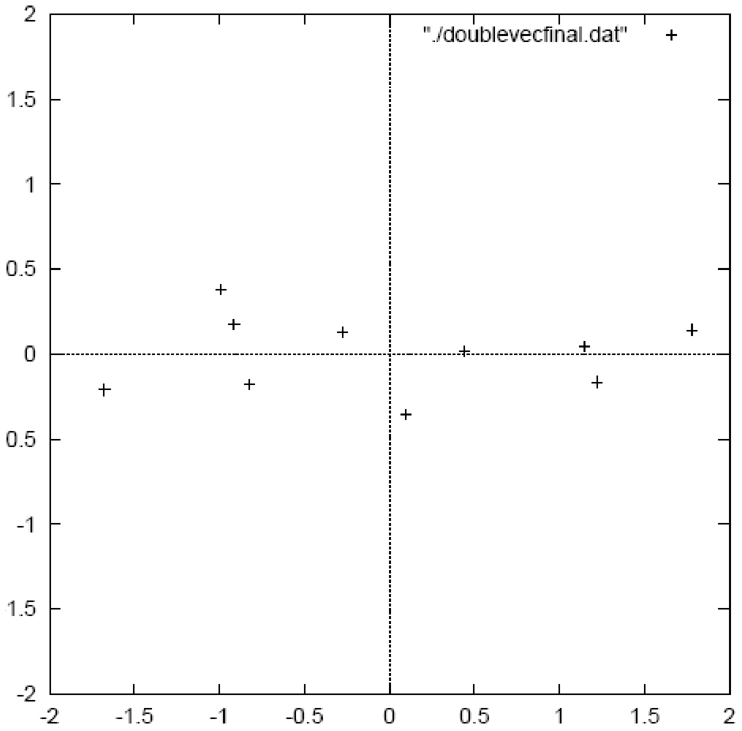
$v_1 . v_2 = 0$



Step 3: project the data

- Let $V = [v_1, \ldots\ v_m]$ is $d \times m$ matrix where the columns vi are the eigenvectors corresponding to the largest m eigenvalues
- The projected data: $Y = X\ V$ is $N \times m$ matrix.
- If m=d (more precisely rank(X)), then there is no loss of information!





- Step 3: project the data

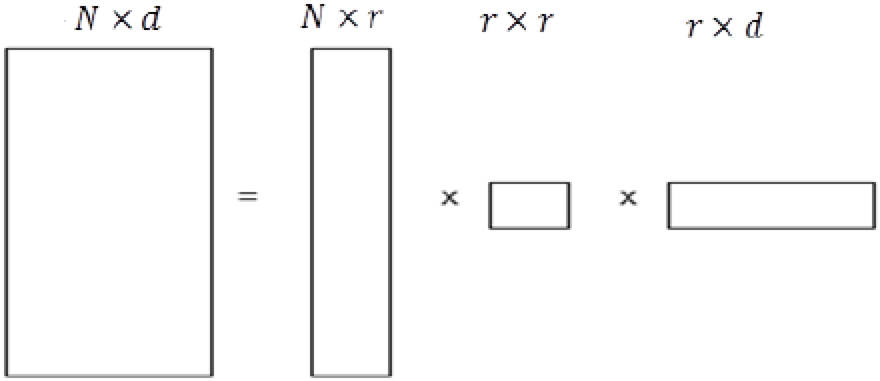
$$\lambda_1 \approx 1.28,\ v_1 \approx [-0.677\ \ -0.735]^T,\ \lambda_2 \approx 0.49,\ v_2 \approx [-0.735\ \ 0.677]^T$$

- The eigenvector with the highest eigenvalue is the **principle component** of the data.
- *if we are allowed to pick only one dimension, the principle component is the best direction (retain the maximum variance).*
- Our PC is $v_1 \approx [-0.677\ -0.735]^T$

SINGULAR VALUE DECOMPOSITION(SVD)

- Any $N \times d$ matrix X can be uniquely expressed as:

$$\mathbf{X} = \mathbf{U} \times \mathbf{\Sigma} \times \mathbf{V}^{\mathbf{T}}$$



- r is the rank of the matrix X (# of linearly independent columns/rows).
  - U is a column-orthonormal $N \times r$ matrix.
  - Σ is a **diagonal $r \times r$ matrix** where the singular values σi are sorted in descending order.
  - V is a column-orthonormal $d \times r$ matrix.

PCA AND SVD RELATION

Theorem:

Let $X = U\,\Sigma\,V^T$ be the SVD of an $N \times d$ matrix X and

$C = \frac{1}{N-1} X^T X$ be the $d \times d$ covariance matrix.

The eigenvectors of C are the same as the right singular vectors of X.

Proof:

$$X^T X = V\,\Sigma\,U^T\,U\,\Sigma\,V^T = V\,\Sigma\,\Sigma\,V^T = V\,\Sigma^2\,V^T$$

$$C = V\frac{\Sigma^2}{N-1}V^T$$

But C is symmetric, hence $C = V\,\Lambda\,V^T$

Therefore, the eigenvectors of the covariance matrix C are the same as matrix V (right singular vectors) and

the eigenvalues of C can be computed from the singular values $\lambda_i = \frac{\sigma_i^2}{N-1}$

ASSUMPTIONS OF PCA

- I. Linearity
- II. Mean and variance are sufficient statistics.
  - Gaussian distribution assumed
- III. Large variances have important dynamics.
- IV. The principal components are orthogonal

PCA WITH EIGENVALUE DECOMPOSITION

```
function [signals,PC,V] = pca1(data)

% PCA1: Perform PCA using covariance.
% data - MxN matrix of input data
% (M dimensions, N trials)
% signals - MxN matrix of projected data
% PC - each column is a PC
% V - Mx1 matrix of variances

[M,N] = size(data);

% subtract off the mean for each dimension
mn = mean(data,2);
data = data - repmat(mn,1,N);

% calculate the covariance matrix
covariance = 1 / (N-1) * data * data’;
```

```
% find the eigenvectors and eigenvalues
[PC, V] = eig(covariance);

% extract diagonal of matrix as vector
V = diag(V);

% sort the variances in decreasing order
[junk, rindices] = sort(-1*V);
V = V(rindices);
PC = PC(:,rindices);

% project the original data set
signals = PC’ * data;
```

Shlens, J. (2003). A tutorial on principal component analysis.

PCA WITH SVD

```
function [signals,PC,V] = pca2(data)

% PCA2: Perform PCA using SVD.
% data - MxN matrix of input data
% (M dimensions, N trials)
% signals - MxN matrix of projected data
% PC - each column is a PC
% V - Mx1 matrix of variances

[M,N] = size(data);

% subtract off the mean for each dimension
mn = mean(data,2);
data = data - repmat(mn,1,N);

% construct the matrix Y
Y = data’ / sqrt(N-1);

% SVD does it all
[u,S,PC] = svd(Y);

% calculate the variances
S = diag(S);
V = S .* S;

% project the original data
signals = PC’ * data;
```

Shlens, J. (2003). A tutorial on principal component analysis.

Instructor: Sael Lee

CS549 Spring – Computational Biology

LECTURE 17:
KERNEL PCA

KERNEL-BASED FEATURE EXTRACTION

- PCA can only extract a linear projection of the data
  - To do so, we first compute the covariance matrix

$$S = \frac{1}{N}\sum_{n=1}^{N} \mathrm{x}_n \mathrm{x}_n^T$$

  - Then, we find the eigenvectors and eigenvalues

$$Su_i = \lambda_i u_i \text{ and } u_i^T u_i = 1$$
$$SU = \lambda U$$

  - And, finally, we project onto the eigenvectors with largest eigenvalues

$$y = \mathrm{Ux}$$

- Can the kernel trick be used to perform this operation implicitly in a higher-dimensional space?
  - If so, this would be equivalent to performing non-linear PCA in the feature space

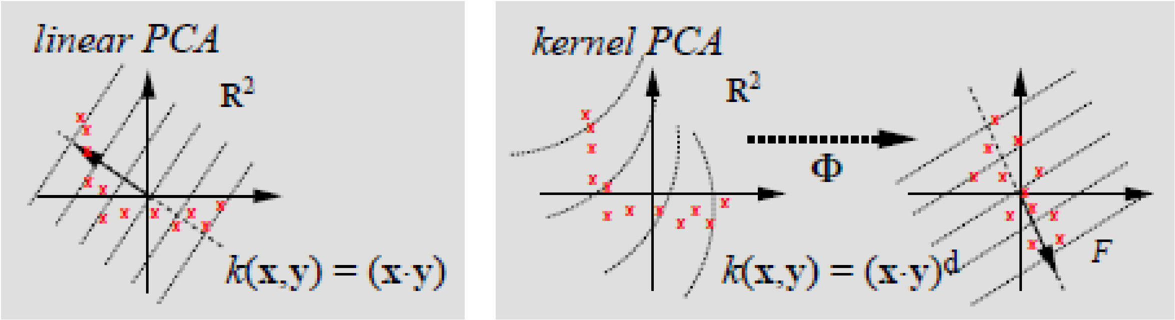

linear PCA
R^2
$k(\mathbf{x},\mathbf{y}) = (\mathbf{x}\cdot\mathbf{y})$
kernel PCA
R^2
Φ
$k(\mathbf{x},\mathbf{y}) = (\mathbf{x}\cdot\mathbf{y})^d$
F

Fig. 1. Basic idea of kernel PCA: by using a nonlinear kernel function k instead of the standard dot product, we implicitly perform PCA in a possibly high–dimensional space F which is nonlinearly related to input space. The dotted lines are contour lines of constant feature value.

Scholkopf, B., Smola, A., Muller, K. R., & Kybernetik, M. (n.d.). Kernel Principal Component Analysis, 2–7.

DERIVING KERNEL-PCA

* Assume zero mean data (centralized data points)

1. Project the data into the high-dim feature space M

$$\phi: R^D \rightarrow R^M; \mathrm{x} \rightarrow \phi(\mathrm{x})$$

2. Compute the covariance matrix

* Assume that projected data has zero mean (we will deal with it later)

$$C = \frac{1}{N} \sum_{n=1}^{N} \phi(\mathrm{x}_n)\phi(\mathrm{x}_n)^T$$

3. Compute the principal components by solving the eigenvalue problem

$$Cv_i = \lambda_i v_i \qquad \text{where } i = 1 \dots M$$

$$\text{or } Cv = \lambda v$$

- **The challenge is... how do we do this implicitly?**

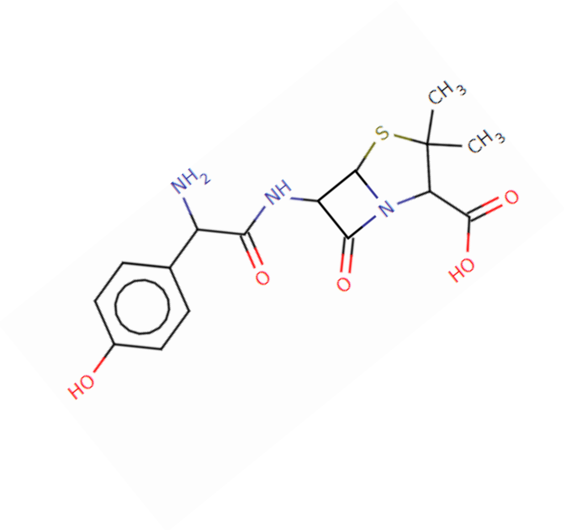
Schölkopf et al., (Neural Computation, 1998)

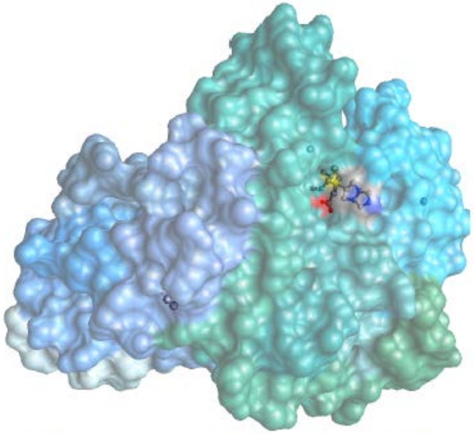
Instructor: Sael Lee

CS549 Spring – Computational Biology

LECTURE 19: DRUG DISCOVERY & CHEMOINFORMATICS

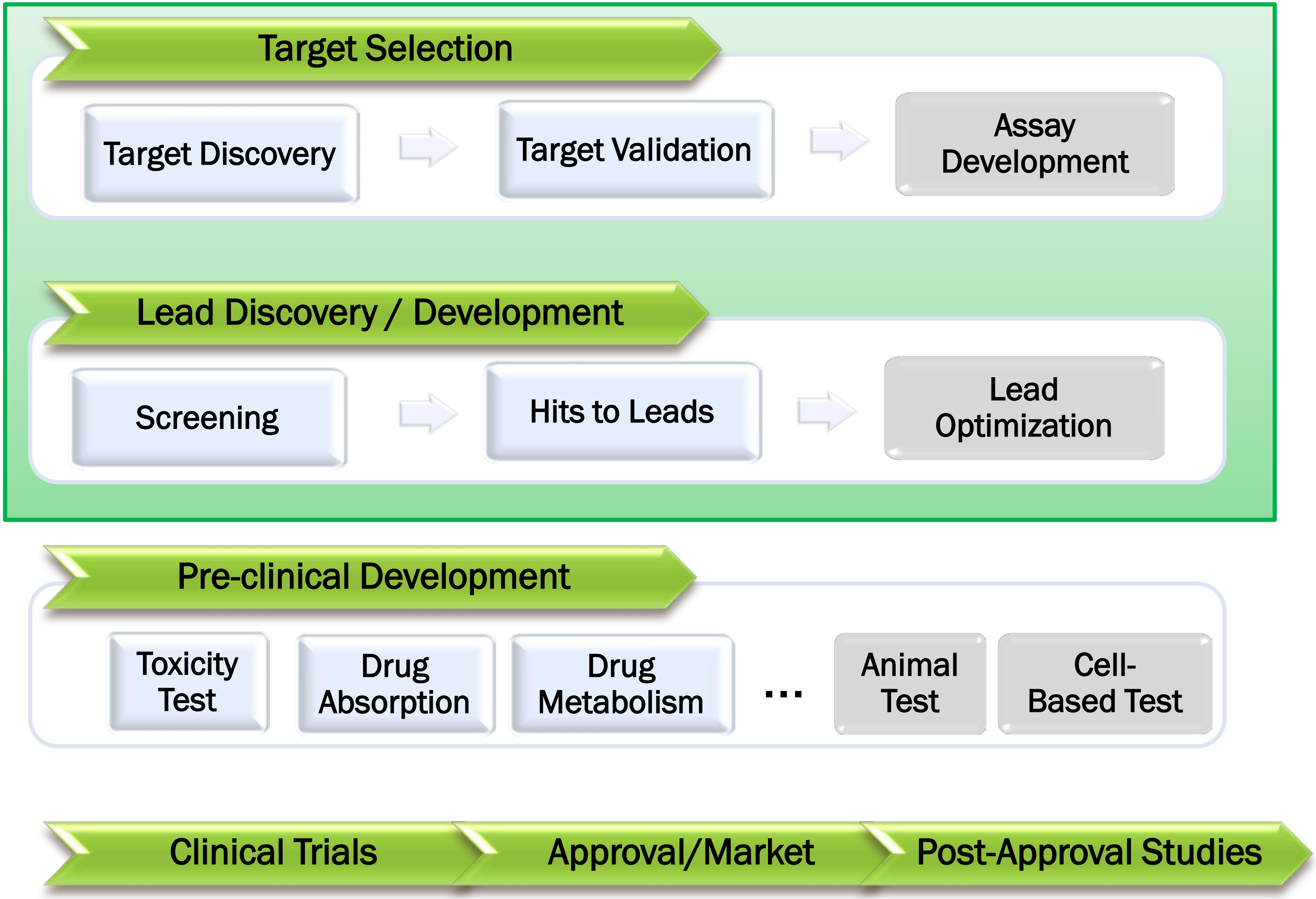
RATIONAL DRUG DISCOVERY





Biapenem in PBP-1A

TYPICAL RATIONAL DRUG DISCOVERY PROCEDURE



Target Selection

Target Discovery → Target Validation → Assay Development

Lead Discovery / Development

Screening → Hits to Leads → Lead Optimization

Pre-clinical Development

Toxicity Test

Drug Absorption

Drug Metabolism

…

Animal Test

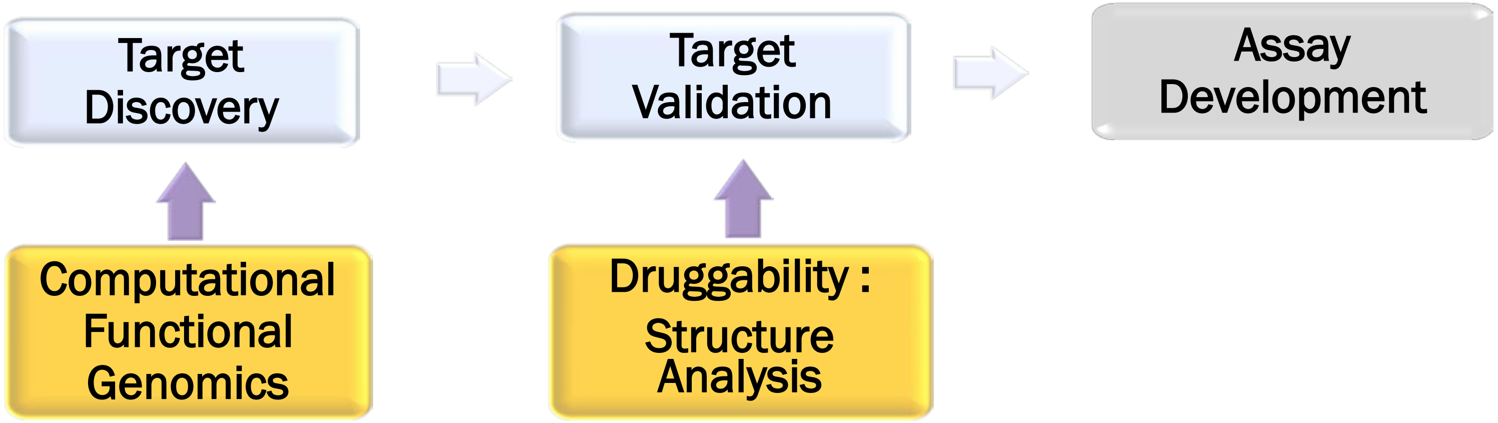
Cell-Based Test

Clinical Trials

Approval/Market

Post-Approval Studies

Target Selection



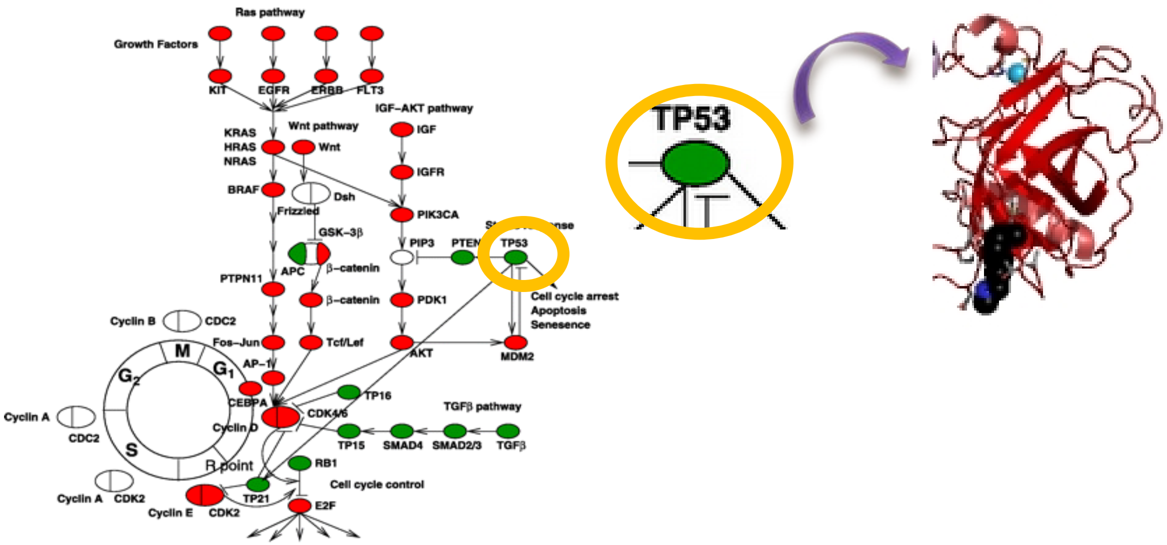
Target Discovery

Target Validation

Assay Development

Computational Functional Genomics

Druggability : Structure Analysis



[Yeang et al. *The FASEB Journal* 2008;22:2605-262]

PDBID: 2VUK
Cellular tumor antigen p53 core domain

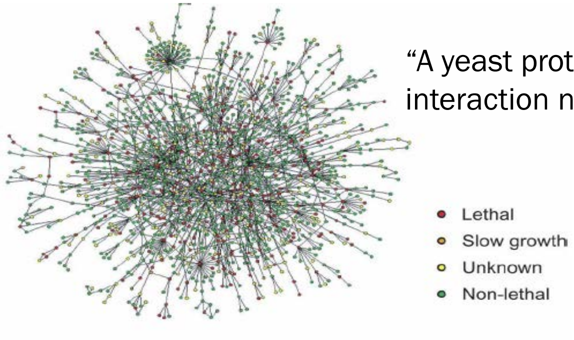
🟨 Computational Study

⬜ Experimental Study

Computational Functional Genomics

DEF.: Computational methods that make s use of the large scale genomic data to describe gene (and protein) functions and their interactions.

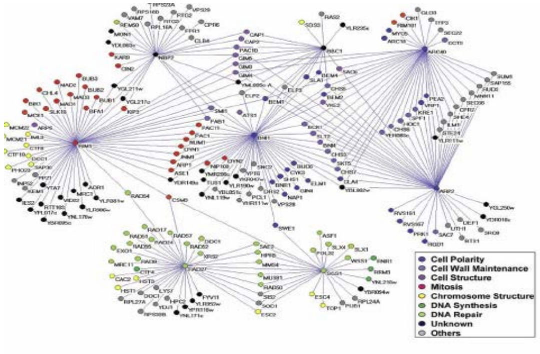
Protein-protein interaction network



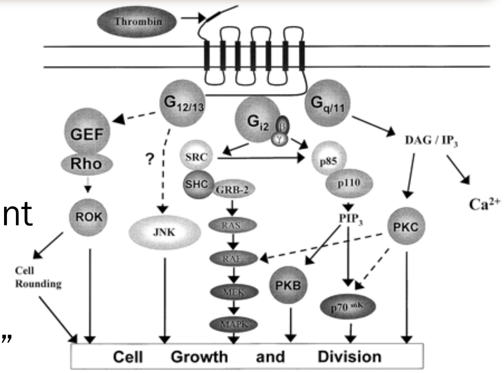
“A yeast protein–protein interaction network”

Gene association networks

“A yeast genetic network “



Regulatory pathways



“G-protein-dependent signaling pathways regulated through activation of PAR-1.”

Metabolic pathways

“An E. coli metabolic network with 574 reactions and 473 metabolites colored according to their modules”

Druggability : Structure Analysis

DEF.: The suitability of a portion of a protein or protein complex to be targeted by a drug, especially by a small molecule drug.

Protein structure prediction

Prediction of the three-dimensional structure of a protein from its amino acid sequence

Protein-ligand/drug binding site prediction

Identification of potential interaction sites such as cavities or pockets on the structure

Protein surface analysis & searching

Calculation and comparison of physicochemical and geometric properties of the potential interaction sites

Protein structure prediction

- Computational determination of three dimensional structure of macro-molecules given their primary structure (amino acid sequence/DNA sequence/RNA sequence)
- Types of structure prediction
  - Protein structure prediction
    - Ab-initial structure prediction
    - Homology modeling
    - Threading
  - RNA structure prediction
  - DNA structure prediction

} Structural searching is important

Protein-ligand / drug binding site prediction

Identifying potential ligand/drug binding sites in proteins using geometric properties such as pocket-like shape and evolutionally conservation information.

Some methods using geometric properties:

•**SURFNET** searches for a gap in a protein surface by fitting spheres inside the convex hull. [Laskowski RA. J Mol Graph1995;13:323–328]

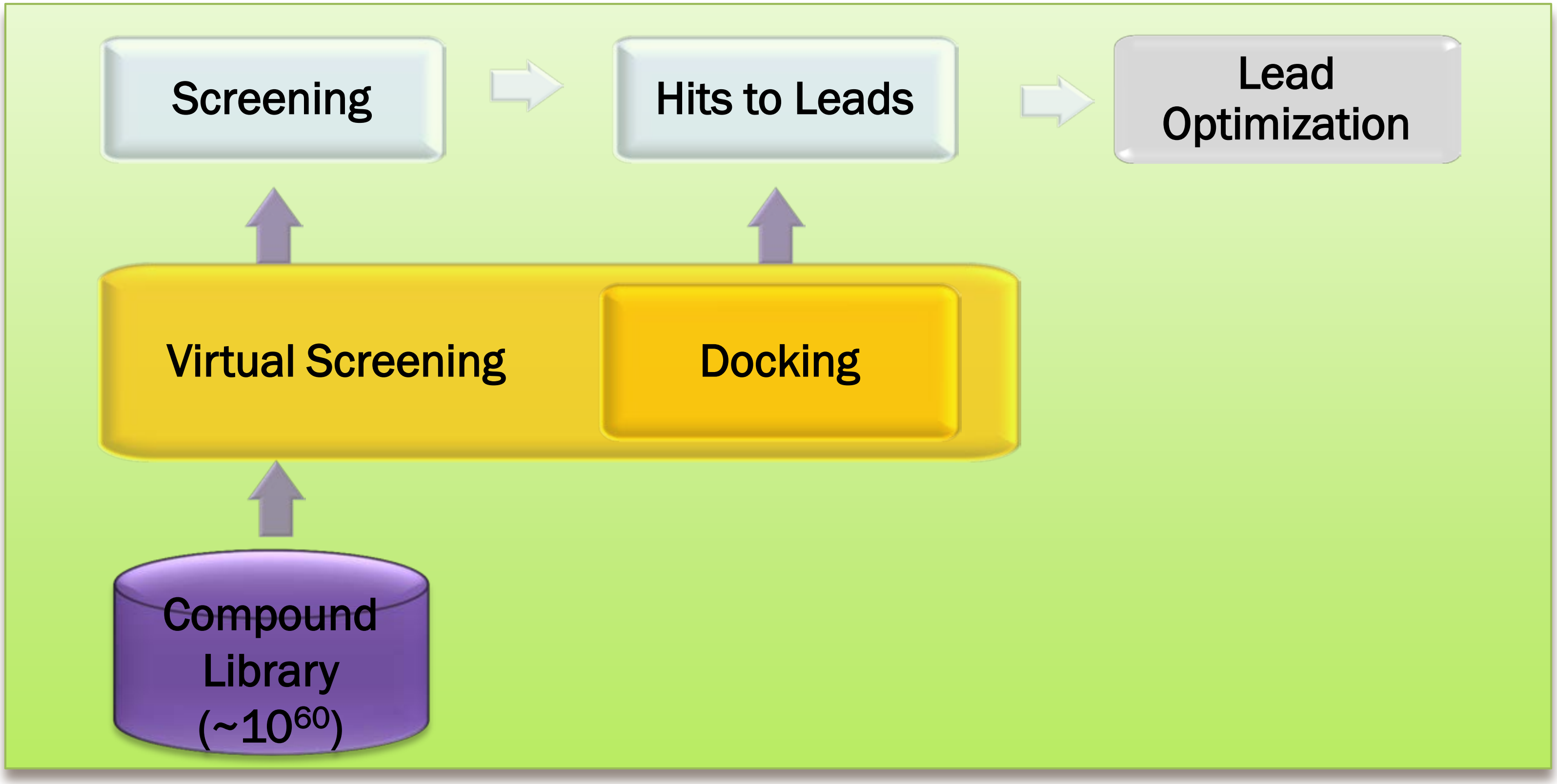
•**PocketPicker** and **LIGSITE** locate a protein onto a three-dimensional (3D) grid and scan it for protein-void-protein events in many directions [Weisel et al. Chem Cent J 2007;1:7, Hendlich et al. J Mol Graph Model 1997;15:359–363]

•**VisGrid** uses the visibility of surface points to find pockets.

•**PocketDepth** clusters grid cells using information of the depth of the grid cells. [Kalidas & Chandra J Struct Biol 2008;161:31–42]

** Several methods consider additional information, such as sequence **conservation** and **energetics** which are often combined while considering geometrical shape.

Lead Discovery / Development



Screening

Hits to Leads

Lead Optimization

Virtual Screening

Docking

Compound Library (~10^{60})

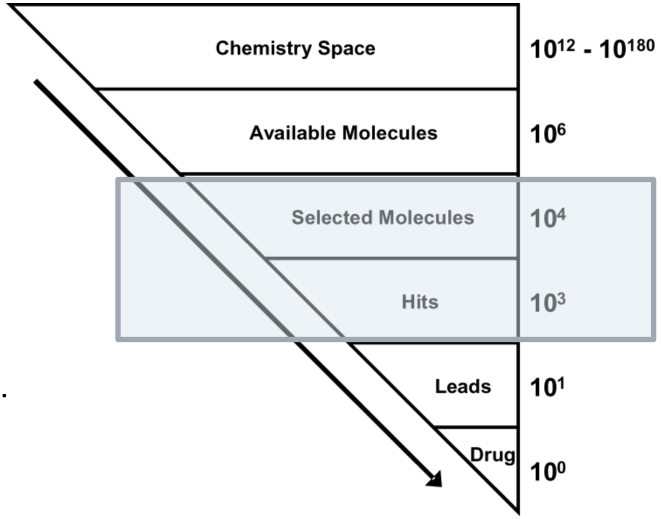
Protein Design and Optimization

Virtual screening

Computational quick search of large compound libraries in order to identify those structures which are most likely to bind to a drug target, typically a protein receptor or enzyme.

Chemoinformatics

- Similarity between known drugs or ones that have predefined properties



Molecular interaction predict

- Computational determination of whether a interact.

Types of interaction prediction

- Protein-small molecule interaction prediction
- Protein-protein interaction prediction

virtual screening are generally good at eliminate the bulk of inactive compounds (negative design). Actual selection of bioactive molecules for a given target requires more improvement(positive design).

Docking

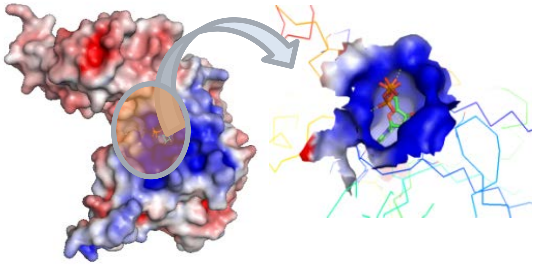
"Computational methods that predict the preferred orientation of one molecule to a second when bound to each other to form a stable complex."[Lengauer & Rarey *Curr. Opin. Struct. Biol.* 1996; **6** (3): 402–6]

Protein-ligand docking

Catalyze enzymatic reactions

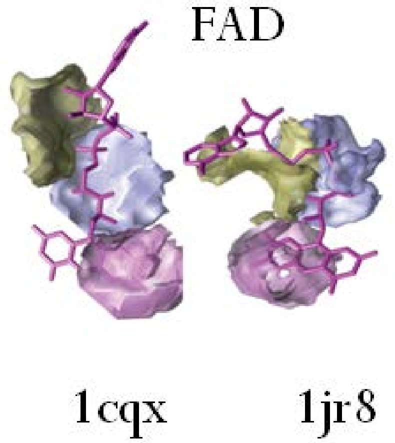
Metabolic processes

Pocket like shapes



1AOI: ATP binding protein

[Chikhi et al *Proteins* 2010]



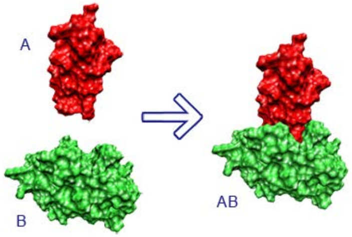
[Sael et al. *IJMS* 2010]

Protein-protein docking

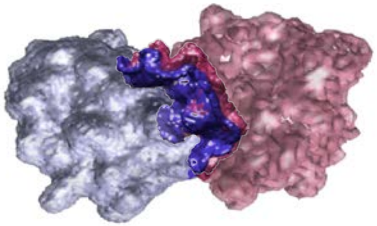
Permanent complex

Transient interaction

Mostly flat region



Z-Dock; LZerD;



1AY7: Ribonuclease Sa/Barstar complex

[Venkatraman et al. *BMC Bioinformatics* 2009]

Many of these problems deals with bio-molecular surface comparison.

CHEMOINFORMATICS & LIGAND-BASED VIRTUAL SCREENING

Resource:

- Brown, N. (2009). Chemoinformatics—an introduction for computer scientists. *ACM Computing Surveys*, *41*(2), 1–38.
- Karsten Borgwardt and Xifeng Yan | Part 8 I: Graph Mining
- Takigawa, I., & Mamitsuka, H. (2013). Graph mining: procedure, application to drug discovery and recent advances. *Drug discovery today*, *18*(1-2), 50–7.

THE SIMILAR-STRUCTURE, SIMILAR-PROPERTY PRINCIPLE

The fundamental assertion of chemoinformatics is the *similar-structure, similar-property principle* (*similar property principle*)

- similar molecules will also tend to exhibit similar properties; this is known as
- "*. . .* the so-called principle of similitude, which states that systems constructed similarly on different scales will possess similar properties." [Johnson and Maggiora 1990, page 18]

Problems are solved by determining of structural similarity between two molecules, or a larger set of molecules.

Similarity searching in virtual screening from a problem-centric rather than a method centric perspective is needed, **depending on <u>what is already known about a target and its ligands</u>.**

CHEMICAL SEARCH SPACE

Chemistry space is the term given to the space that contains all of the theoretically possible molecules and is therefore theoretically infinite.

Druglike chemistry space : a set of empirically derived rules is used to define molecules that are more likely to be orally available as drugs.
Reduced druglike chemistry space is estimated to **contain anything from 10^{12} to 10^{180} molecules**

Goal of chemoinformatics is to assist in 1) **filtering** the space of available molecules to something more manageable while also 2) **maximizing** the chances of a) covering the molecules with the most potential to enter the clinic and b) maintaining some degree of structural diversity to avoid prospective redundancies or premature convergence.

Brown, N. (2009).

CHEMISTRY AND GRAPH THEORY

The **molecular graph** is a type of graph that is undirected and where the nodes are colored and edges are weighted where the **nodes** are the **atoms** of a molecule and the **edges** are the **bonds.**

- The individual nodes are colored according to the particular atom type: carbon (C), oxygen (O), nitrogen (N), chlorine (Cl),etc.,
- The edges are assigned weights according to the bond order: single, double, triple, and aromatic.

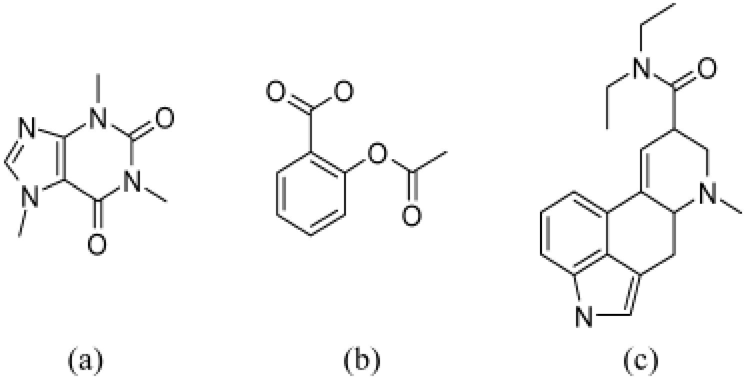
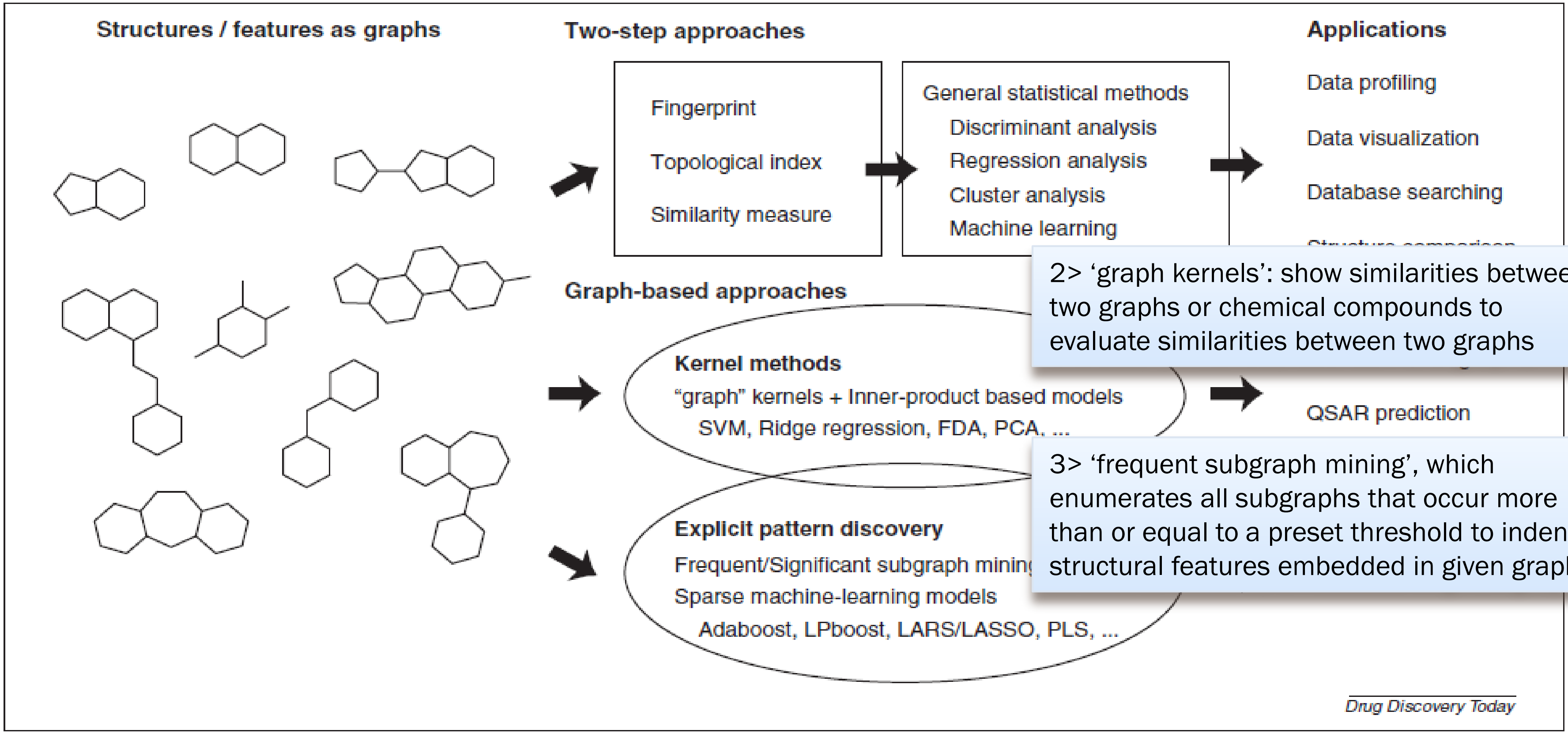


Fig. 5. The hydrogen-depleted molecular graphs of (a) caffeine, (b) aspirin, and (c) D-lysergic acid diethylamide. (N Brown 2009)

VARIOUS GRAPH MINING-BASED APPROACHES



Structures / features as graphs

Two-step approaches

Fingerprint

Topological index

Similarity measure

General statistical methods

Discriminant analysis

Regression analysis

Cluster analysis

Machine learning

Applications

Data profiling

Data visualization

Database searching

Structure comparison

QSAR prediction

Graph-based approaches

Kernel methods

"graph" kernels + Inner-product based models

SVM, Ridge regression, FDA, PCA, ...

Explicit pattern discovery

Frequent/Significant subgraph minin

Sparse machine-learning models

Adaboost, LPboost, LARS/LASSO, PLS, ...

Drug Discovery Today

2> 'graph kernels': show similarities between two graphs or chemical compounds to evaluate similarities between two graphs

3> 'frequent subgraph mining', which enumerates all subgraphs that occur more than or equal to a preset threshold to indentify structural features embedded in given graphs.

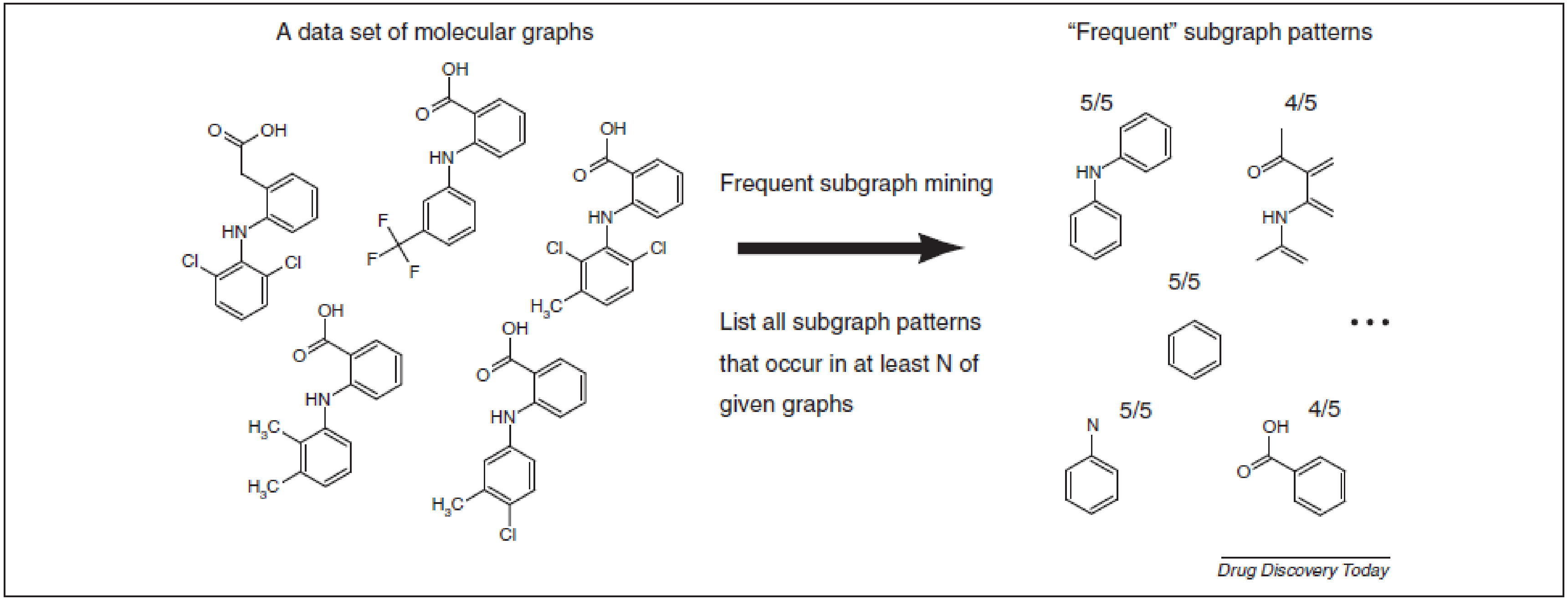
FIGURE 1

Three types of graph mining approaches. *Abbrevations*: ADME/Tox: absorption, distribution, metabolism, excretion and toxicology; LARS: least square regression; LASSO: least absolute shrinkage and selection operator; PCA: principal component analysis; PLS: partial least squares; QSAR: quantitative structure–activity relationship; SVM: support vector machines.

Fig. from Takigawa, I., & Mamitsuka, H. (2013).

FREQUENT SUBGRAPH MINING

Frequent subgraph mining is used for analyzing structural fragments or partial structures and molecular graphs.



Frequent subgraph mining

List all subgraph patterns that occur in at least N of given graphs

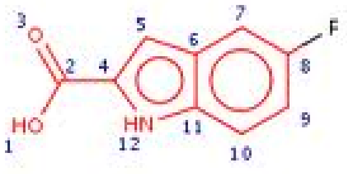
FIGURE 3

Frequent subgraph mining. Note that this example does not consider aromaticity, however, it can be incorporated.

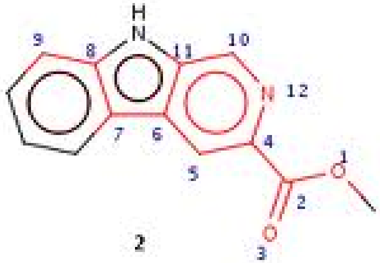
Fig. from Takigawa, I., & Mamitsuka, H. (2013).

SUBGRAPH ISOMORPHISM

Problem: Given two graphs G and H as input, determine whether G contains a subgraph G' that is where two vertices u and v of G' are adjacent in G' if and only if f(u) and f(v) are adjacent in H (isomorphic to H)



1



2

"subgraph isomorphism problem' is theoretically proven to be NP-complete.

Instructor: Sael Lee

CS549 Spring – Computational Biology

LECTURE 20: GRAPH KERNELS

Resources:

- Shervashidze, N., et al. (2011). Weisfeiler-Lehman Graph Kernels. *Journal of Machine Learning Research*, *12*, 2539–2561.
- “Graph Mining and Graph Kernels” *K. Borgwardt and X. Yan* KDD2008 Tutorial
- Vishwanathan, S. V. N., et al. (2010). Graph Kernels. *Journal of Machine Learning Research*, *11*, 1201–1242.
- “Graph kernels and chemoinformatics” Jean-Philippe Vert. Slides from Gbr’2007

GRAPH ISOMORPHISM

Graph isomorphism

Find a mapping f of the vertices of G_1 to the vertices of G_2 such that G_1 and G_2 are identical; i.e. (x,y) is an edge of G_1 iff (f(x),f(y)) is an edge of G_2. Then f is an **isomorphism**, and G_1 and G_2 are called **Isomorphic**

- No polynomial-time algorithm is known for graph isomorphism
- Neither is it known to be NP-complete

Subgraph isomorphism

G_1 and G_2 are **isomorphic** if there exists a subgraph isomorphism from G_1 to G_2 and from G_2 to G_1

- Subgraph isomorphism is NP-complete

We want polynomial-time similarity measure for graphs

MEASURING GRAPH SIMILARITY 1

Graph Edit Distances

Principle

- Count operations that are necessary to transform G1 into G2
- Assign costs to different types of operations (edge/node insertion/deletion, modification of labels)

Advantages

- Captures partial similarities between graphs
- Allows for noise in the nodes, edges and their labels
- Flexible way of assigning costs to different operations

Disadvantages

- Contains subgraph isomorphism check (NP-complete) as one intermediate step
- Choosing cost function for different operations is difficult

MEASURING GRAPH SIMILARITY 2

Topological Descriptors

Principle

- Map each graph to a feature vector (ex> finger printing methods)
- Use distances and metrics on vectors for learning on graphs

Advantages

- Reuses known and efficient tools for feature vectors

Disadvantages

- Most feature vector transformation leads to loss of topological information
- Or includes subgraph isomorphism as one step

MEASURING GRAPH SIMILARITY 3:

Graph Kernels: Kernels on pairs of graphs

Principle

- Let $\phi(\mathrm{x})$ be a vector representation of the graph x
- The kernel between two graphs is defined by:

$$K(x, x') = \phi(x)^T \, \phi(x')$$

- To solve convex optimization with kernels, kernels needs to be
  - Symmetric, that is, $k(x, x') = k(x', x)$, and
  - Positive semi-definite (p.s.d.)
- Comparing nodes in a graph involves constructing a kernel between nodes
- Comparing graphs involves constructing a kernel between graphs.

Advantages

- Similarity of two graphs are inferred through kernel function

Disadvantages

- Defining a kernel that captures the semantics inherent in the graph structure and is reasonably efficient to evaluate is the key challenge.

GRAPH KERNELS TERMINOLOGY

- A **graph** G as a triplet (V, E, l), where *V* is the set of vertices, *E* is the set of undirected edges, and $l : V \rightarrow \Sigma$ is a function that assigns labels from an alphabet Σ to nodes in the graph.
- The **neighborhood** *N* (*v*) of a node *v* is the set of nodes to which *v* is connected by an edge, that is $N(v) = \{v' | (v, v') \in E\}$.

For simplicity, we assume that every graph has ***n*** nodes, ***m*** edges, and a maximum degree of ***d***. The **size of *G*** is defined as the cardinality of *V.*

- A **path** is a walk that consists of distinct nodes only.
- A **walk** is a sequence of nodes in a graph, in which consecutive nodes are connected by an edge. walk extends the notion of path by allowing nodes to be equal
- A ***(rooted) subtree*** is a subgraph of a graph, which has no cycles, but a designated root node.
- The **height of a subtree** is the maximum distance between the root and any other node in the subtree.

GRAPH KERNELS TERMINOLOGY CONT.

Complete graph kernels

A graph **kernel is complete** if it separates non-isomorphic graphs, i.e.:

$$\forall G_1, G_2 \in X, d_K\,(G_1, G_2) = 0 \Rightarrow G1 \cong G2\,.$$

Equivalently, $\phi(G_1) \neq \phi(G_1)$ if G_1 and G_2 are not isomorphic.

- If a graph kernel is not complete, then there is cannot cover all possible functions over X: the kernel is not expressive enough.
- On the other hand, kernel computation must be tractable, i.e., no more than polynomial (with small degree) for practical applications.
- Can we define tractable and expressive graph kernels?

Computing any complete graph kernel is at least as hard as the graph isomorphism problem. (Gärtner et al., 2003)

GRAPH KERNELS TERMINOLOGY CONT.

subtree patterns (also called ***tree-walks***, Bach, 2008) can have nodes that are equal .

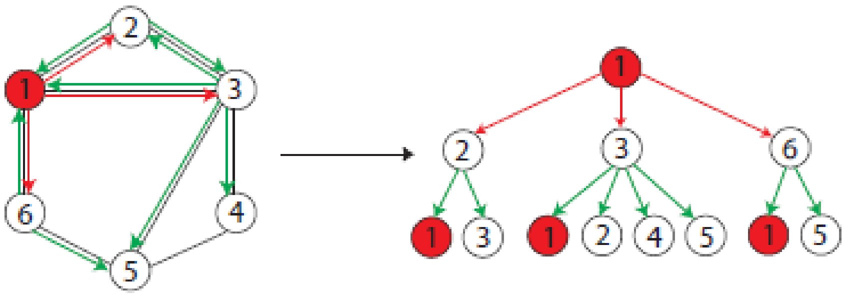


Figure 1: A subtree pattern of height 2 rooted at the node 1. Note the repetitions of nodes in the unfolded subtree pattern on the right.

Note that all **subtree kernels** compare **subtree** ***patterns*** in two graphs, not (strict) subtrees.

PATH KERNEL

A **path** of a graph (V,E) is sequence of **distinct vertices**

$v_1, \ldots, v_n \in V \ (i \ \neq j \Rightarrow \ v_i \ \neq v_j$) such that $(v_i \ , v_{i+1}) \in E$ for i = 1, . . . , n − 1.

Equivalently the paths are the **linear subgraphs.**

The **path kernel** is the subgraph kernel restricted to paths, i.e.,

$$K_{path}(G_1, G_2) \ = \sum_{H \in P} \lambda_H \ \phi_H(G_1)\phi_H(G_2)$$

where P ⊂ X is the set of path graphs.

NOTE: Computing the path kernel is NP-hard. (Gärtner et al., 2003)

EXPRESSIVENESS VS COMPLEXITY TRADE-OFF

- It is **intractable** to compute **complete graph kernels.**
- It is **intractable** to compute the **subgraph kernels**.
- Restricting subgraphs to be linear does not help:
  - it is **intractable** to compute the **path kernel**.
- One approach to define polynomial time computable graph kernels is to have the feature space be made up of graphs **homomorphic to subgraphs**, e.g., to consider walks instead of paths.

RANDOM WALKS

Principle (Kashima et al., ICML 2003, Gaertner et al., COLT 2003)

- Compare walks in two input graphs G and G’
- Walks are sequences of nodes that allow repetitions of nodes

Computation

- Walks of length k can be computed by looking at the k-th power of the adjacency matrix
- Construct direct product graph of G and G’
- Count walks in this product graph $\mathrm{G}_\times = (V_\times, E_\times)$
- <u>Each walk in the product graph corresponds to one walk in G and G’</u>

$$k_\times(G,G') = \sum_{i,j=1}^{|V_\times|} [\sum_{k=0}^{\infty} \lambda^k A_\times^k]_{ij}$$

Runtime in $O(n^6)$

Some proposed speed up:

- Fast computation of random walk graph kernels (Vishwanathan et al., NIPS 2006)
- Label enrichment and preventing tottering (Mahe et al., ICML 2004)
- Graph kernels based on shortest paths(Kriegel, ICDM 2005)

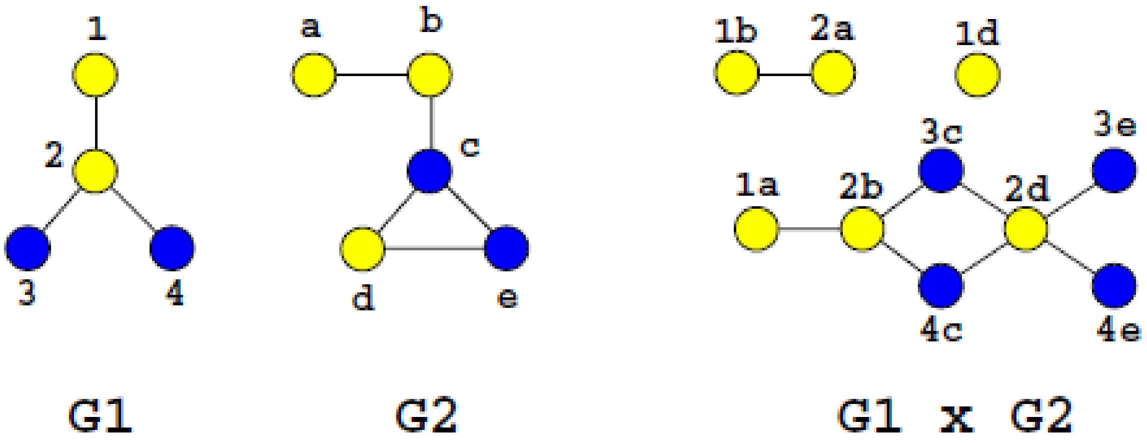
PRODUCT GRAPH

Let $G1 = (V1, E1)\ and\ G2 = (V2, E2)$ be two graphs with labeled vertices. The $product\ graph\ G = G1 \times G2$ is the graph G = (V,E) with:

$$V = \{(v_1, v_2) \in V_1 \times V_2 : v_1 \text{ and } v_2 \text{ have the same label}\},$$

$$E = \{((v_1, v_2), (v_1', v_2')) \in V\text{x}V : (v_1, v_1') \in E_1\ and (v_2, v_2') \in E_2\}$$

.

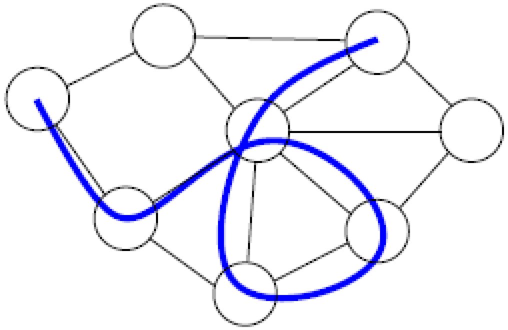


- Product graph consists of pairs of identically labeled nodes and edges from G1 and G2

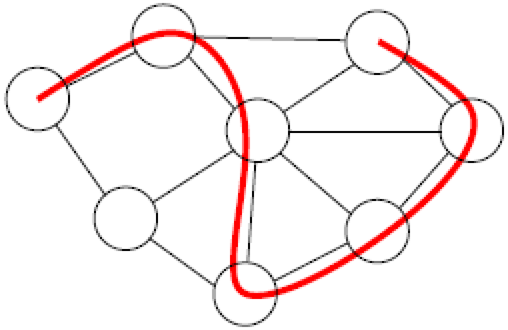
WALKS

A **walk** of a graph (V,E) is sequence of $v_1,\ldots,v_n \in V$ such that $(vi\ , vi+1) \in E$ for $i = 1,\ldots,n-1$.

We note $\boldsymbol{W_n(G)}$ the set of walks with n vertices of the graph G, and $\boldsymbol{W(G)}$ the set of all walks.

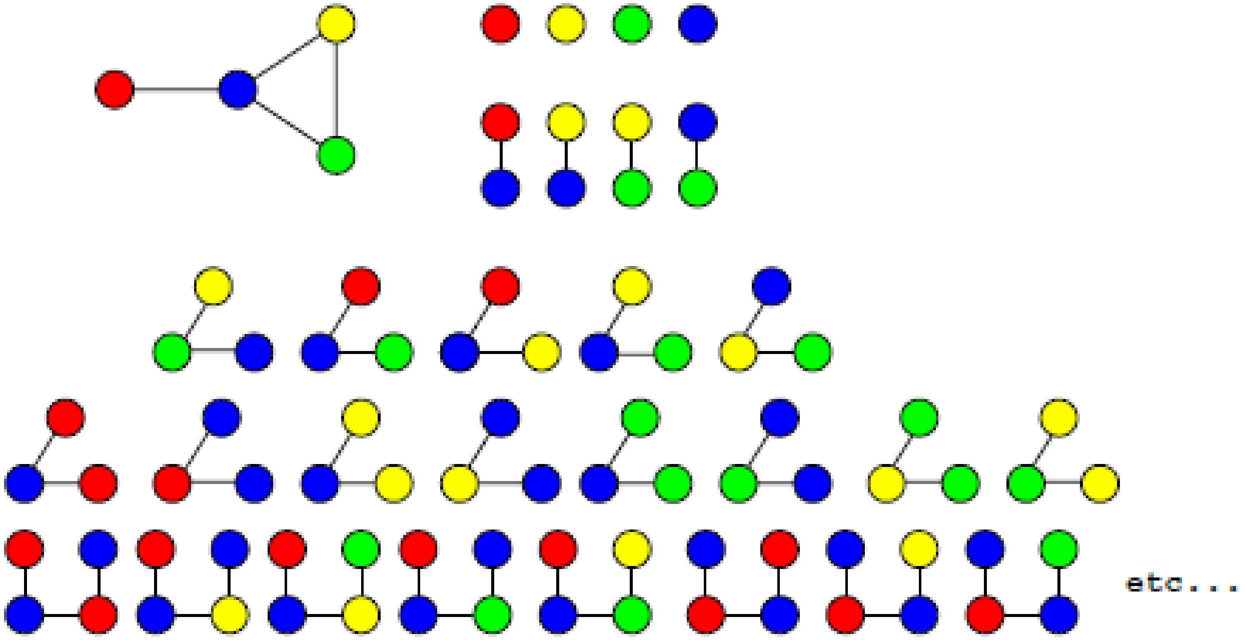


walks



Paths

WALK KERNELS

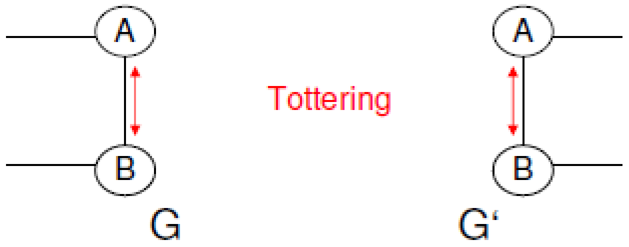


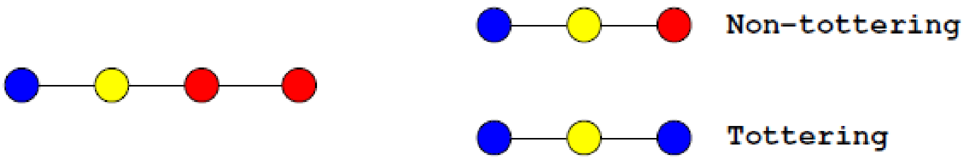
TOTTERING

Tottering (Mahe et al., ICML 2004)

A **tottering walk** is a walk $w = v_1 \ldots v_n$ with $v_i = v_i + 2$ for some i.

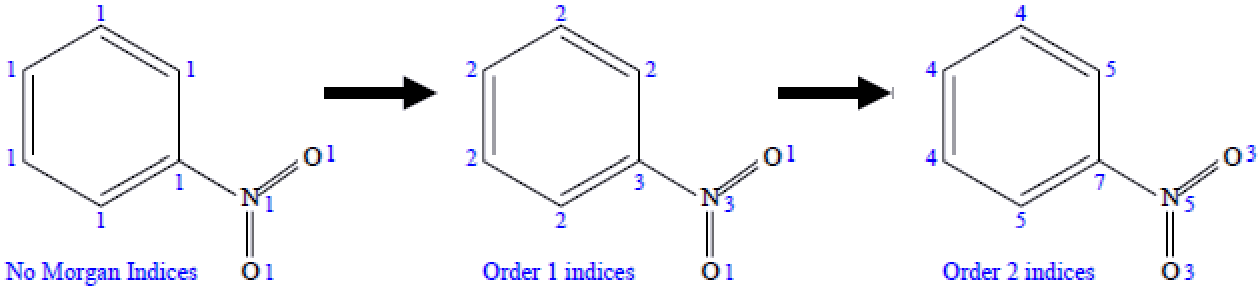
- A walk can visit the same cycle of nodes all over again
- Kernel measures similarity in terms of common walks
- Hence a small structural similarity can cause a huge kernel value
- Focusing on non-tottering walks is a way to get closer to the path kernel (e.g., equivalent on trees).



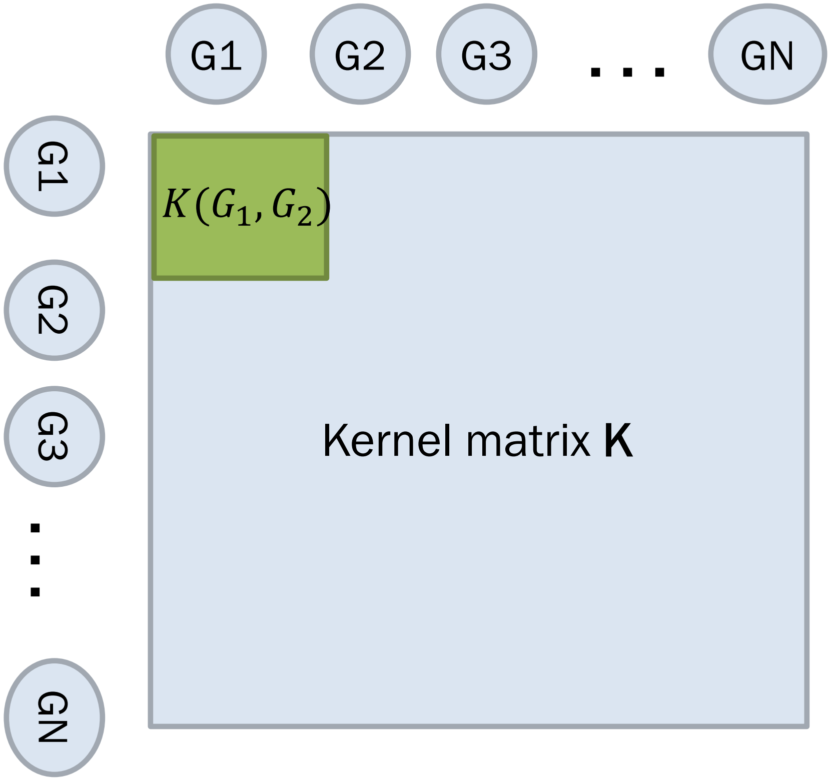


LABEL ENRICHMENT: MORGAN INDEX (1965)

- Size of product graph affects runtime of kernel computation
- The **more node labels, the smaller the product graph**
- Trick: Introduce new artificial node labels
- Topological descriptors of nodes are natural extra labels
- For instance, the Morgan Index that counts k-th order neighbours of a node:



GRAPH KERNELS



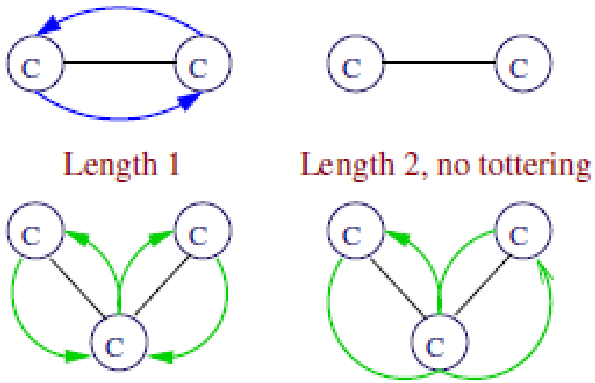
How to define a **valid kernel** function $K(G_j, G_j)$, between two graphs G_j and G_j.

- $K(G_j, G_j)$ should provide relationship (similarity / dissimilarity / correlation etc.) measure for between two graphs.
- $K(G_j, G_j)$ should be able to be applied in kernel based machine learning methods such that it provide optimal classification / clustering performance.

We will look at graph kernels that states similarity between kernels.

PREVENTING TOTTERING CONT.

- Motivation:



⋆ Only "real" chemical path are matched

⋆ ⇒ Compounds are now seen as different

- Solution : increase the order of the random walk model :
  $\Rightarrow p_G(h) = p_s(v_1)p_t(v_2|v_1)\prod_{i=3}^{n} p_t(v_i|v_{i-2}, v_{i-1})$

2ND ORDER MARKOV RANDOM WALK

$$p_G(h) = p_s(v_1)p_t(v_2|v_1)\prod_{i=3}^{n} p_t(v_i|v_{i-2}, v_{i-1})$$

$$\begin{cases} p_s(v) = p_0(v)p_q^{(0)}(v), \\ p_t(u|v) = \frac{1-p_q^{(0)}(v)}{p_q^{(0)}(v)} p_a(u|v)p_q(u), \\ p_t(u|w,v) = \frac{1-p_q(v)}{p_q(v)} p_a(u|w,v)p_q(u). \end{cases}$$

The function is still a valid kernel but the implementation described for the first order Markov random walk cannot be directly used anymore.

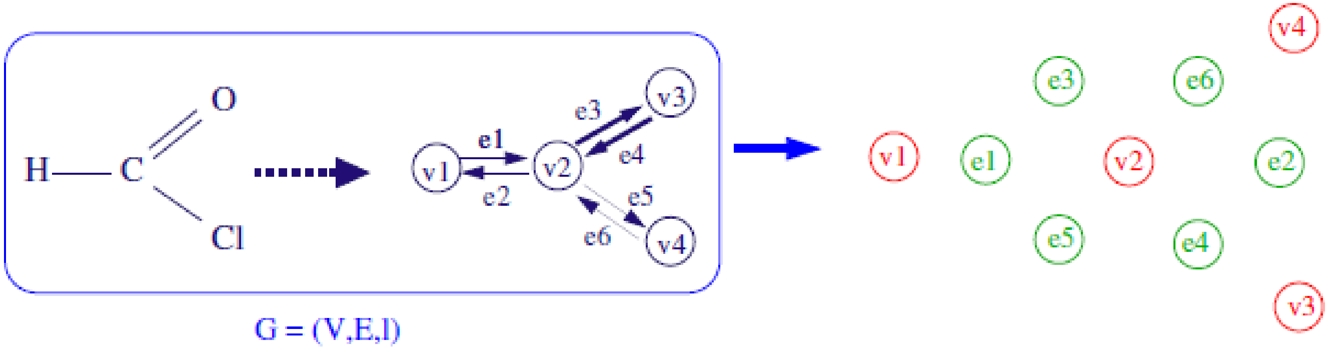
=> Instead of explicitly working with 2nd Order Markov Random walk, transform the original graph G to G' such that G' contains the look ahead information.

GRAPH TRANSFORMATION CONT.

* Don’t confuse G’ used in the last notation for compared Graph

Transformation : $G = (V, E, l) \Rightarrow G' = (V', E', l')$ where :

- $V' = V \cup E$
- $E' = \{(v, (v, t)) \mid v \in V, (v, t) \in E\}$
  $\cup \{((u, v), (v, t)) \mid (u, v), (v, t) \in E, u \neq t\}$

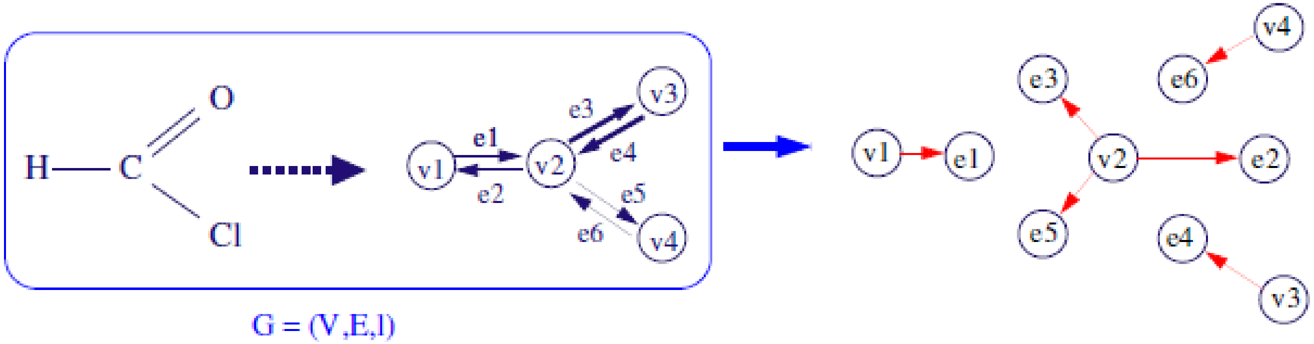


G = (V,E,l)

GRAPH TRANSFORMATION CONT.

Transformation : $G = (V, E, l) \Rightarrow G' = (V', E', l')$ where :

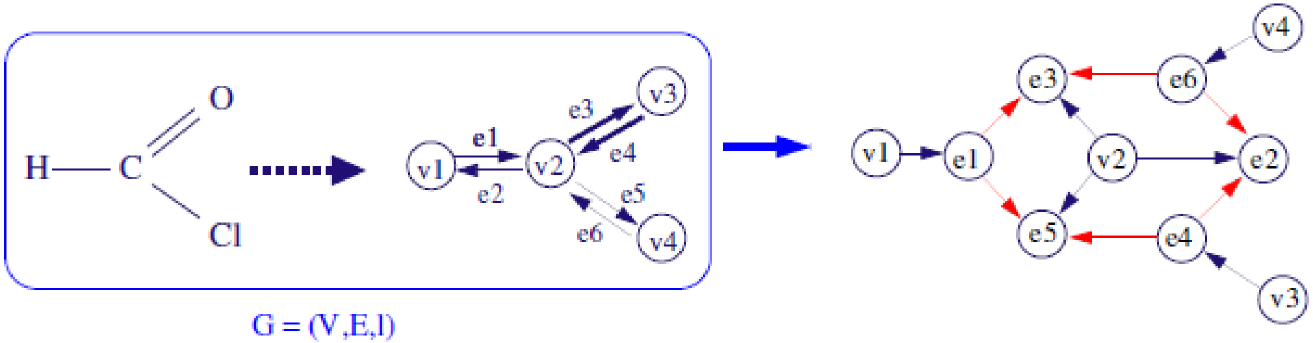
- $V' = V \cup E$
- $E' = \{(v, (v,t)) \,|\, v \in V, (v,t) \in E\}$
  $\cup \{((u,v), (v,t)) \,|\, (u,v), (v,t) \in E, u \neq t\}$



GRAPH TRANSFORMATION CONT.

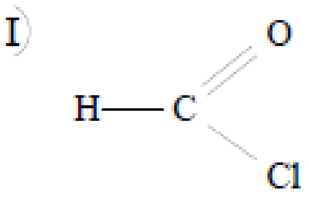
Transformation : $G = (V, E, l) \Rightarrow G' = (V', E', l')$ where :

- $V' = V \cup E$
- $E' = \{(v, (v, t)) \,|\, v \in V, (v, t) \in E\}$
  $\cup \; \{((u, v), (v, t)) \,|\, (u, v), (v, t) \in E, u \neq t\}$

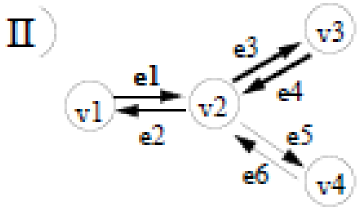


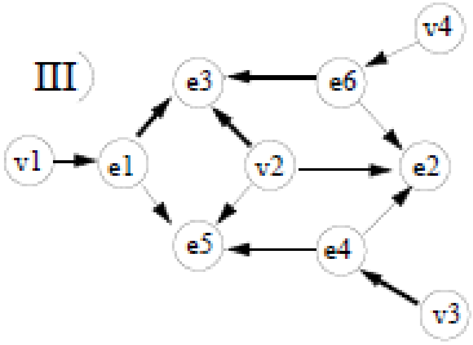
G = (V,E,l)

Original Graph

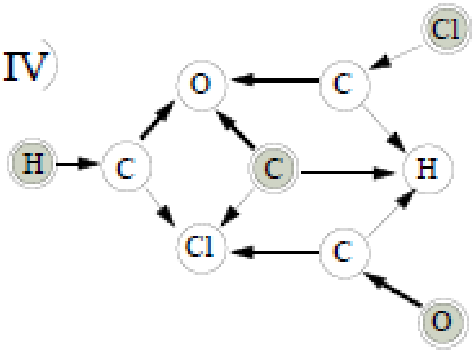


Corresponding directed graph G = (V,E,l)





Transformed Graph



Labels in the transformed graph

MODIFIED KERNEL COMPUTATION CONT.

- Consider : $\begin{cases} H_0(G) = \{\text{Non tottering paths of G}\} \\ H_1(G') = \{\text{Paths of } G' \text{ starting from a node } v \in V \ \} \end{cases}$

- Theorem: p' factorizes as

$$p'(h') = p'_s(v'_1) \prod_{i=2}^{n} p'_t(v'_i|v'_{i-1})$$

$\star$ $p'_s(v') = p_s(v')$

$\star$ $p'_t(u'|v') = \begin{cases} p_t(u|v') \text{ if } v' \in V \text{ and } u' = (v', u) \in E \\ p_t(u|v, w) \text{ if } v' = (v, w) \text{ and } u' = (w, u) \in E \end{cases}$

- Corollary :

$$\left. \begin{array}{l} \text{- graph transformation} \\ \text{- original graph kernel} \end{array} \right\} \Rightarrow \text{tottering paths removed}$$

MODIFIED KERNEL COMPUTATION

- Consider : $\begin{cases} H_0(G) = \{\text{Non tottering paths of G}\} \\ H_1(G') = \{\text{Paths of } G' \text{ starting from a node } v \in V\ \} \end{cases}$

- The mapping $f : H_0(G) \to H_1(G')$ defined by

$$h = (v_1, ..., v_n) \mapsto h' = (v'_1, ..., v'_n) \text{ such that } \begin{cases} v'_1 = v_1 \\ v'_i = (v_{i-1}, v_i) \end{cases}$$

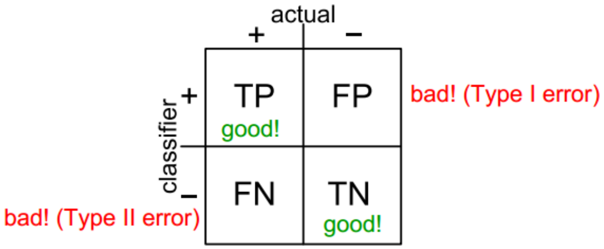
establishes a bijection between $H_0(G)$ and $H_1(G')$

one-to-one correspondence

- Let p' be the image of p_G by f:

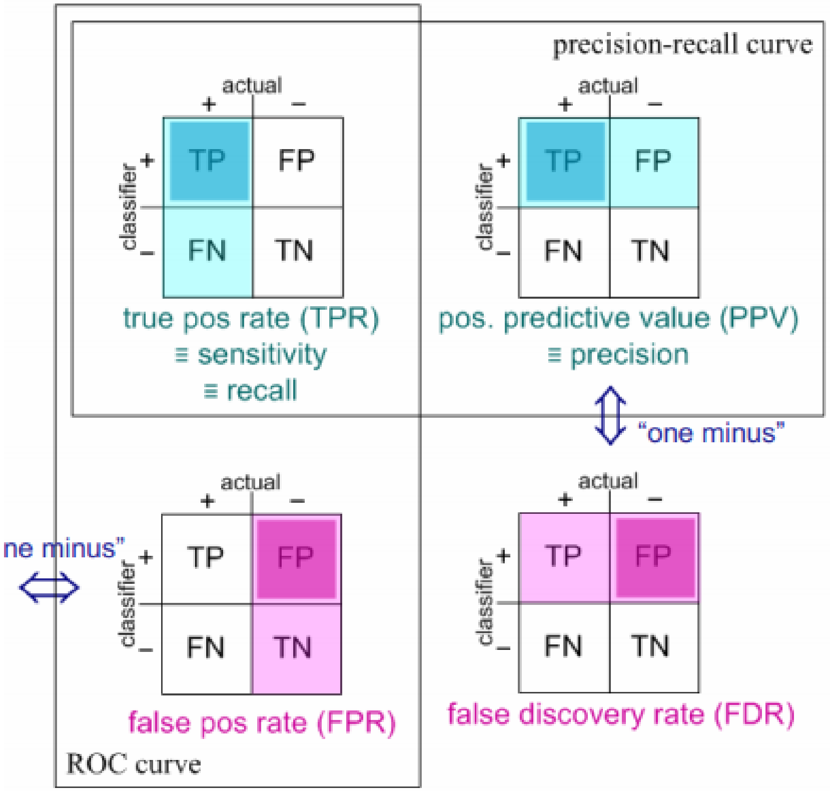
$$\forall h' \in H_1(G'), \quad p'(h') := p_G\left(f^{-1}(h')\right)$$

ROC VS PRECISION RECALL



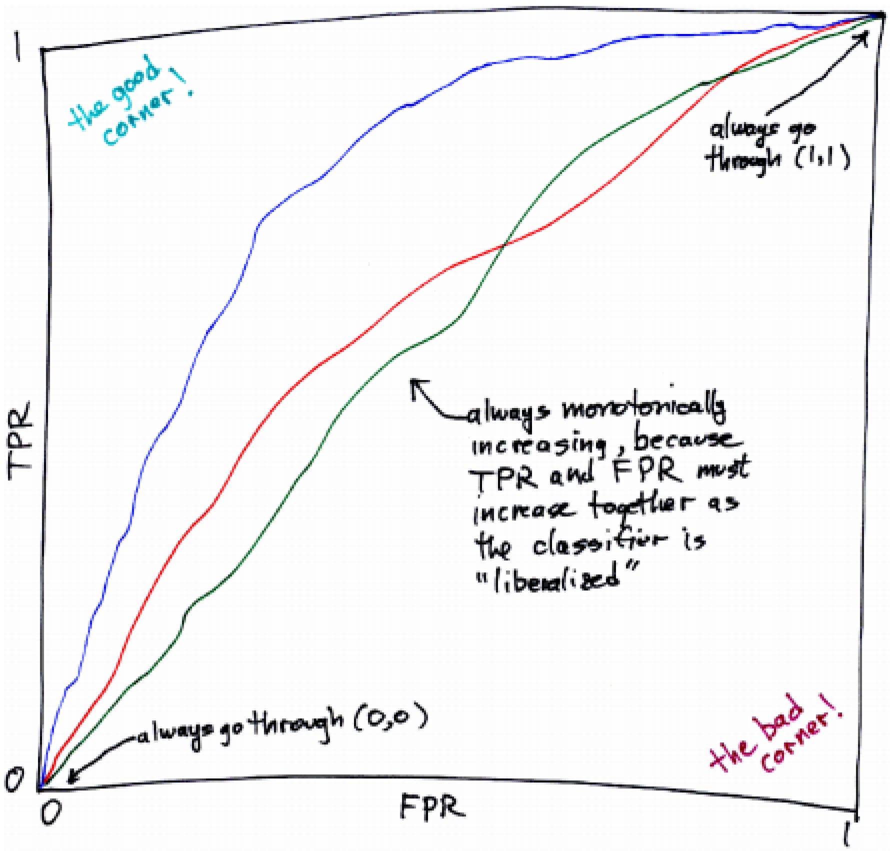
precision recall curve

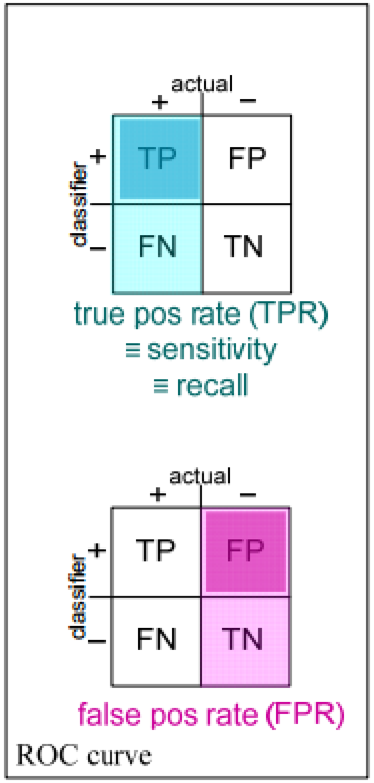
ROC (AUC)



Slide from The UT Austin, CS 395T, Spring 2008, Prof. William H. Press

ROC ("Receiver Operating Characteristic") curves plot TPR vs. FPR as the classifier goes from "conservative" to "liberal"





		actual +	actual −
classifier	+	TP	FP
classifier	−	FN	TN

true pos rate (TPR) ≡ sensitivity ≡ recall

		actual +	actual −
classifier	+	TP	FP
classifier	−	FN	TN

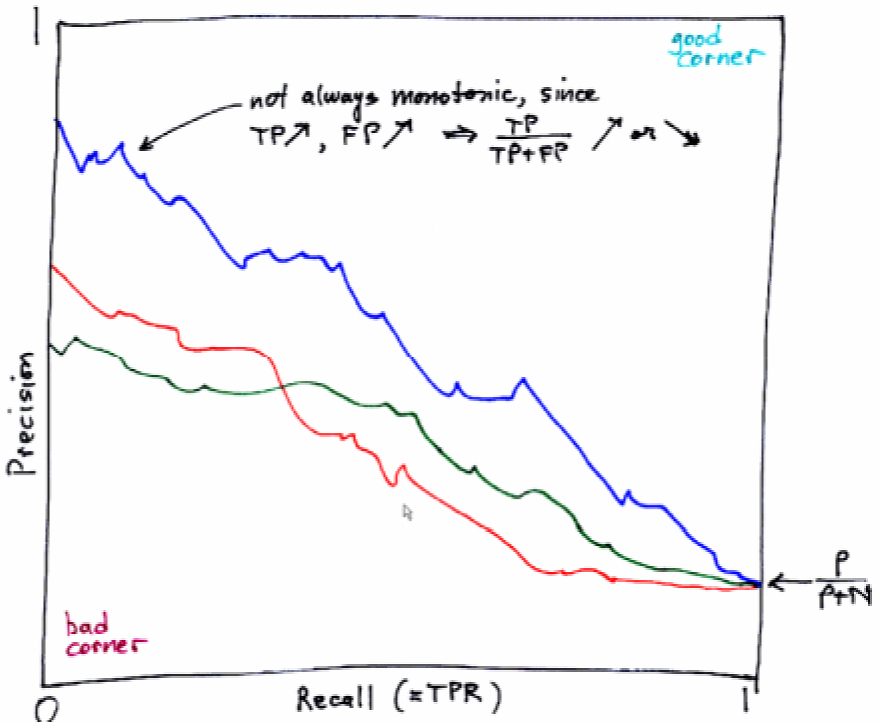
false pos rate (FPR)

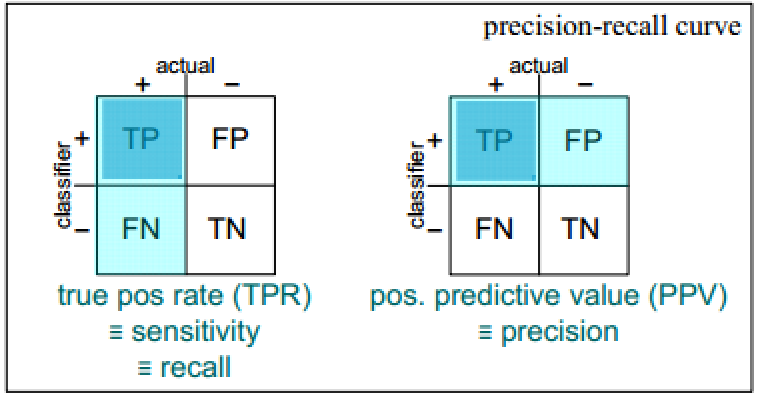
ROC curve

blue dominates red and green
neither red nor green dominate the other

You could get the best of the red and green curves by making a hybrid or "Frankenstein" classifier that switches between strategies at the cross-over points.

Precision-Recall curves overcome this issue by comparing TP with FN and FP




precision-recall curve

actual + −
classifier + −

TP	FP
FN	TN

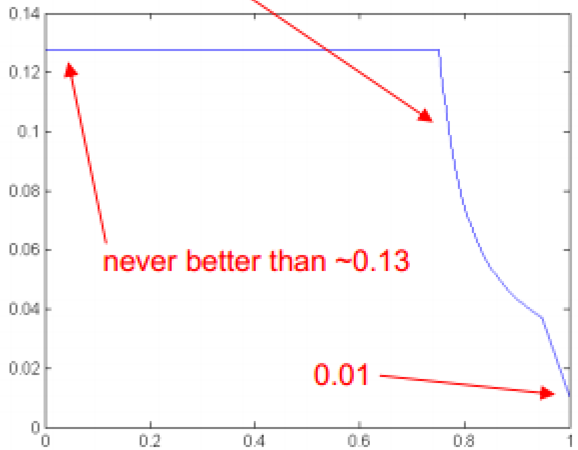
true pos rate (TPR)
≡ sensitivity
≡ recall

actual + −
classifier + −

TP	FP
FN	TN

pos. predictive value (PPV)
≡ precision

By the way, this shape "cliff" is what the ROC convexity constraint looks like in a Precision-Recall plot. It's not very intuitive.



never better than ~0.13

0.01

Continue our toy example:

note that P and N now enter

```
prec = tpr*100./(tpr*100+fpr*9900);
prec(1) = prec(2); % fix up 0/0
reca = tpr;
plot(reca,prec)
```

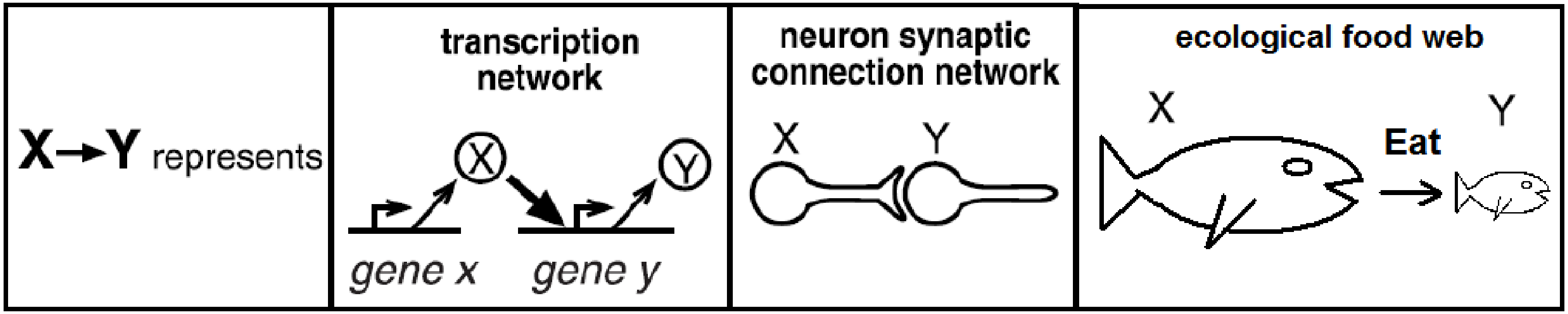
Selected slides from BioNetwork slide by Dr. Nataša Pržulj

Dr. Nataša Pržulj
Department of Computing
Imperial College London
natasha@imperial.ac.uk
www.doc.ic.ac.uk/~natasha/

Introduction: biological networks

- **Biological nets**

  Other network types

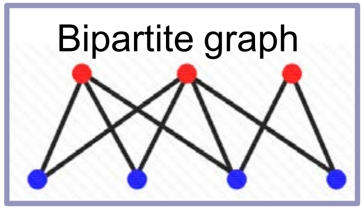


Metabolic networks

- Used for studying and modeling ***metabolism***
  - Biochemical reactions in cells that allow an organism to:
    - Respond to the environment
    - Grow
    - Reproduce
    - Maintain its structure
    -
  - i.e., the main biochemical reactions needed to keep an organism in *homeostasis*
    - An internal regulation that maintains a stable, constant condition of a living system

Metabolic networks

- ***Metabolites***
  - Small molecules such as glucose and amino acids
  - Also, macromolecules such as polysaccharides and glycans (carbohydrates)
- ***Metabolic pathways***
  - Series of successive biochemical reactions for a specific metabolic function, e.g., glycolysis, or penicillin synthesis, that convert one metabolite into another
  - ***Enzymes***: proteins that catalyze (accelerate) chem. reactions
- Thus, in a metabolic pathway:
  - Nodes correspond to metabolites and enzymes
    - In an alternate order → ***bipartite graphs***
  - Directed edges correspond to metabolic reactions
  - Simpler approaches: **nodes** are metabolites, directed edges are reactions that convert one metabolite into the other; or **nodes** are enzymes and metabolites as edges

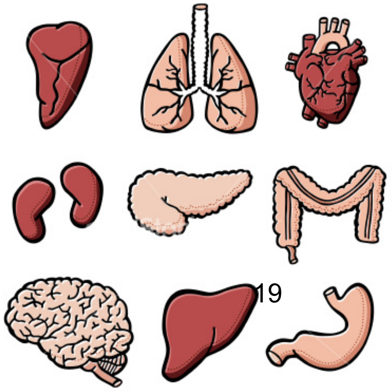

Bipartite graph

Metabolic networks

- All metabolic pathways of a cell form a ***metabolic network***
  - Complete view of cellular metabolism and material/mass flow through the cell
  - Cell relies on this network to digest substrates from the environment, generate energy, and synthesize components needed for its growth and survival
  - Insights from analyzing them <u>used to</u>, for example:
    - Cure human metabolic diseases through better understanding of the metabolic mechanisms
    - Control infections of pathogens by understanding the metabolic differences between human and pathogens

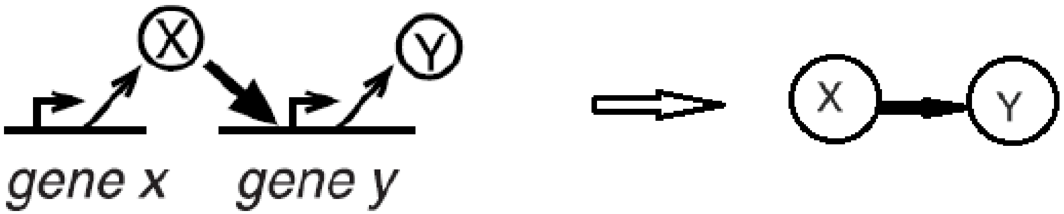
Transcriptional regulation networks

- Model regulation of *gene expression*
  - Recall: gene → mRNA → protein
- Gene regulation
  - Gives a cell control over its structure and function, e.g.:
    - *Cellular differentiation* – a process by which a cell turns into a more specialized cell type
    - *Morphogenesis* (a process by which an organism develops its shape)
    - ...



Transcriptional regulation networks

- Nodes correspond to genes
  - DNA sequences which are transcribed into mRNAs that translate into proteins
- Directed edges correspond to interactions through which the products of one gene affect those of another
  - Protein-protein, protein-DNA and protein-mRNA interactions



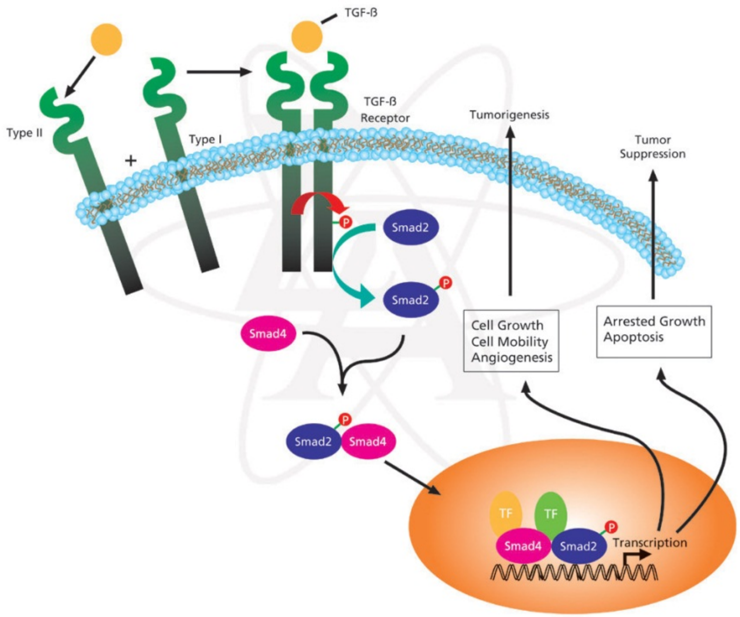
- *Transcription factor* X (protein product of gene X) binds regulatory DNA regions of gene Y to regulate the production rate (i.e., stimulate or repress transcription) of protein Y
  - Note: proteins are products of gene expression that play a key role in regulation of gene expression

Transcriptional regulation networks

- Problem
  - *Stimulation* and *repression* of gene transcription are both represented the same way in the network
- Available for *model organisms*
  - Non-human species manipulated and studied to get insights into workings of other organisms, e.g.:
    - Baker's yeast, *S. cerevisiae* (Milo et al., 2002)
    - *E. coli* (Shen-Orr et al., 2002)
    - Sea urchin (Davidson et al., 2002)
    - Fruitfly, *D. melanogaster*
  - Available from dBs: EcoCyc, GeneNet, KEGG, RegulonDB, Reactom, TRANSPATH, TRANSFAC

Cell signaling networks

- ***Cell signaling***
  - ○ Complex communication system that governs basic cellular activities, e.g., development, repair, immunity
- Errors in signaling cause diseases
  - ○ E.g., cancer, autoimmune diseases, diabetes



E.g.: **Transforming growth factor beta** (TGF-β) is a protein that controls proliferation, cellular differenciation, and other functions in most cells.

Cell signaling networks

- ***Signaling pathways***
  - Ordered sequences of signal transduction reactions in a cell, as shown in the previous figure
  - Cascade of reversible chemical modifications of proteins
    - E.g., phosphorylation catalyzed by *protein kineases:* enzymes that modify other proteins by adding phosphate groups to them (process called *phosphorylation*)
- Signaling pathways in the cell form the ***cell signaling network***
  - Nodes are proteins and edges are directed

Cell signaling networks

Famous examples (lots of literature on them):

- *Mitogen-activated protein kinase (MAPK)* pathway
  - Originally called "ERK" pathway
  - MAPK protein: an enzyme, a *protein kinase,* which can attach *phosphate groups* to a target protein, causing its spatial reorganization and affecting its function
    - Other enzymes can restore protein's initial function
  - E.g.:
    - MYC
      - An *oncogene* transcription factor expressed in a wide range of human cancers (oncogene – when mutated or over-expressed, the gene helps turn a normal into a tumor cell)
      - MAPK can *phosphorylate* (attach phosphate group to) MYC and alter gene transcription and cell cycle progression
    - EGFR = "epidermal growth factor receptor"
      - Activates MAPK pathway
      - Mutations affecting its expression/activity can result in cancer

Cell signaling networks

Famous examples (lots of literature on them) cont'd:

- *Hedgehog signaling pathway*
  - One of the key regulators of animal development
  - Conserved from fly to human
  - Establishes basis of fly body plan
  - Important during *embryogenesis* (the process by which the embryo develops) and *metamorphosis* (from larva to pupa to adult)
- *TGF-beta signaling pathway*
  - The "transforming growth factor" (TGF) signaling pathway
  - Involved in:
    - Cell growth
    - Cell differentiation
    - *Apoptosis* (programmed cell death)

Cell signaling networks

- Compared to metabolic networks:
  - Limited mass flow
  - Instead, sig. nets provide information transmission along a sequence of reactions – one enzyme modulates the activity of another one, which then modulates the activity of the third enzyme, etc., but *enzymes are not consumed* in the reactions they catalyze
- Compared to transcriptional reg. networks:
  - They overlap, but gene expression, i.e., transcription factors, can be seen as the "final targets" of signaling pathways
- Compared to PPI networks:
  - Signal transduction is indeed mediated between proteins, but PPIs are undirected without a defined input and output (as we will discuss soon)
  - Not all PPIs are involved in chemical reactions, or part of signal transduction
  - Also, many components of signaling are not proteins
- These networks have much in common
- At the same time, they reflect different aspects of cellular activity

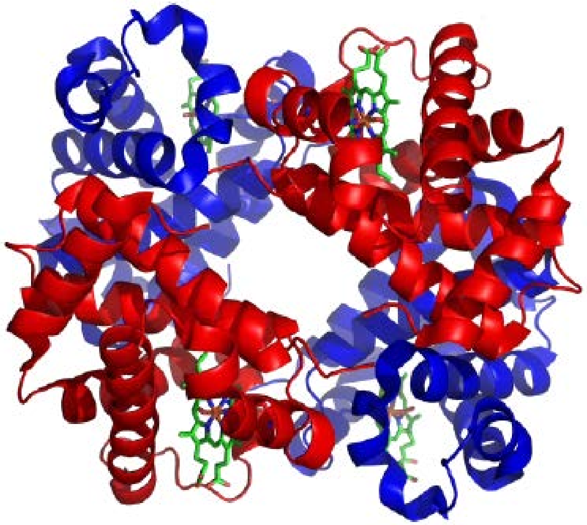
Protein-protein interaction (PPI) networks

- A *protein-protein interaction (PPI)* usually refers to a physical interaction, i.e., binding between proteins
- Can be other associations of proteins such as functional interactions – e.g., synthetic lethality: type of a “genetic interaction” (will introduce later)

Protein-protein interaction (PPI) networks

- PPIs are very important for structure and function of a cell:
  - Participate in signal transduction (*transient interactions*)
    - Play a role in many diseases (e.g., cancer)
  - Can be *stable interactions* forming a *protein complex*

    (a form of a quaternary protein structure, set of proteins which bind to do a particular function, e.g., ribosome, hemoglobin – illustrated below)



Protein-protein interaction (PPI) networks

- PPIs are very important for structure and function of a cell:
  - Can be *transient interactions*
    - Brief interactions that modify a protein that can further change PPIs e.g., protein kineases (add a phosphate group to a target protein)
    - A protein can carry another protein, e.g., *nuclear pore importins* (proteins that carry other proteins from cytoplasm to nucleus and vice versa)
    - Transient interactions form the *dynamic part of PPI networks*
  - Some estimates state that about 70% of interactions are stable and 30% are dynamic (transient)
- PPIs are essential to almost every process in a cell
- Thus, understanding PPIs is crucial for understanding life, disease, development of new drugs (most drugs affect PPIs)

Protein-protein interaction (PPI) networks

Methods to detect PPIs

- Biological and computational approaches
- None are perfect
  - High rates of *false positives*
    - Interactions present in the data sets that are not present in reality
  - High rates of *false negatives*
    - Missing true interactions

Protein-protein interaction (PPI) networks

Methods to detect PPIs

- PPIs initially studied individually by small-scale biochemical techniques (SS)
- However, large-scale (high-throughput) interaction detection methods (HT) are needed for high discovery rates of new protein interactions
- SS of better “quality,” i.e., less noisy than HT
- However, HT are more standardized, while SS are performed differently each time
- SS are biased – the focus is on the subsets of proteins interesting to particular researchers
- HT – view of the entire proteome

Protein-protein interaction (PPI) networks

Methods to detect PPIs

- Physical binding
  - Yeast 2-hybrid (Y2H) screening
  - Mass spectrometry of purified complexes
- Functional associations
  - Correlated mRNA expression profiles
  - Genetic interactions
  - In silico (computational) methods
- In many cases, functional associations do take the form of physical binding

Protein-protein interaction (PPI) networks

Functional associations

- Correlated mRNA expression profiles (Dr. Rice’s lectures)
  - Results in a gene expression correlation network
- Co-expression means that resulting proteins *could* interact
- Co-expression overlaid over PPI data, e.g. tool KeyPathwayMiner

Protein-protein interaction (PPI) networks

Functional associations

- Genetic interactions
  - Two non-essential genes that cause lethality when mutated at the same time form a *synthetic lethal* interaction
  - Such genes are often functionally associated and their encoded proteins may also interact physically
  - Charles Boone’s group from University of Toronto published genetic interaction networks

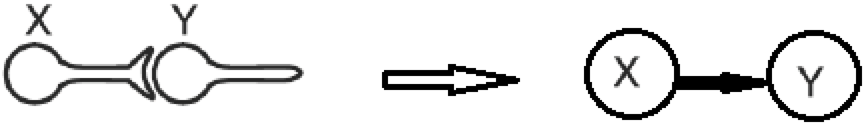
Protein-protein interaction (PPI) networks

Functional associations

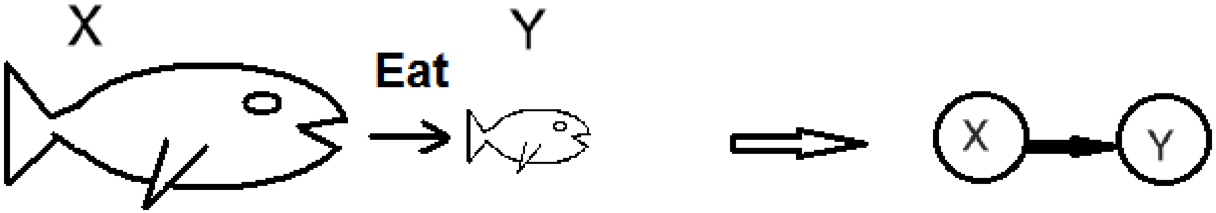
- In silico (computational) methods
  - Gene fusion (if two genes are present in one species and fused in another)

Other biological networks

- Neuronal synaptic connection networks



- Brain functional networks
  - Simultaneous (correlated) activities of brain regions during a task
- Ecological food webs



- Phylogenetic networks (trees)
  - Evolutionary relationships between species

Other biological networks

- Correlation networks (e.g., gene co-expression)
  - Different from transcriptional regulation networks
  - Not a direct result of experiments
  - Determined by:
    - Collecting large amounts of high-throughput data
    - Calculating the correlations between all elements
  - Biolayout Express 3-D: a tool for generating correlation networks

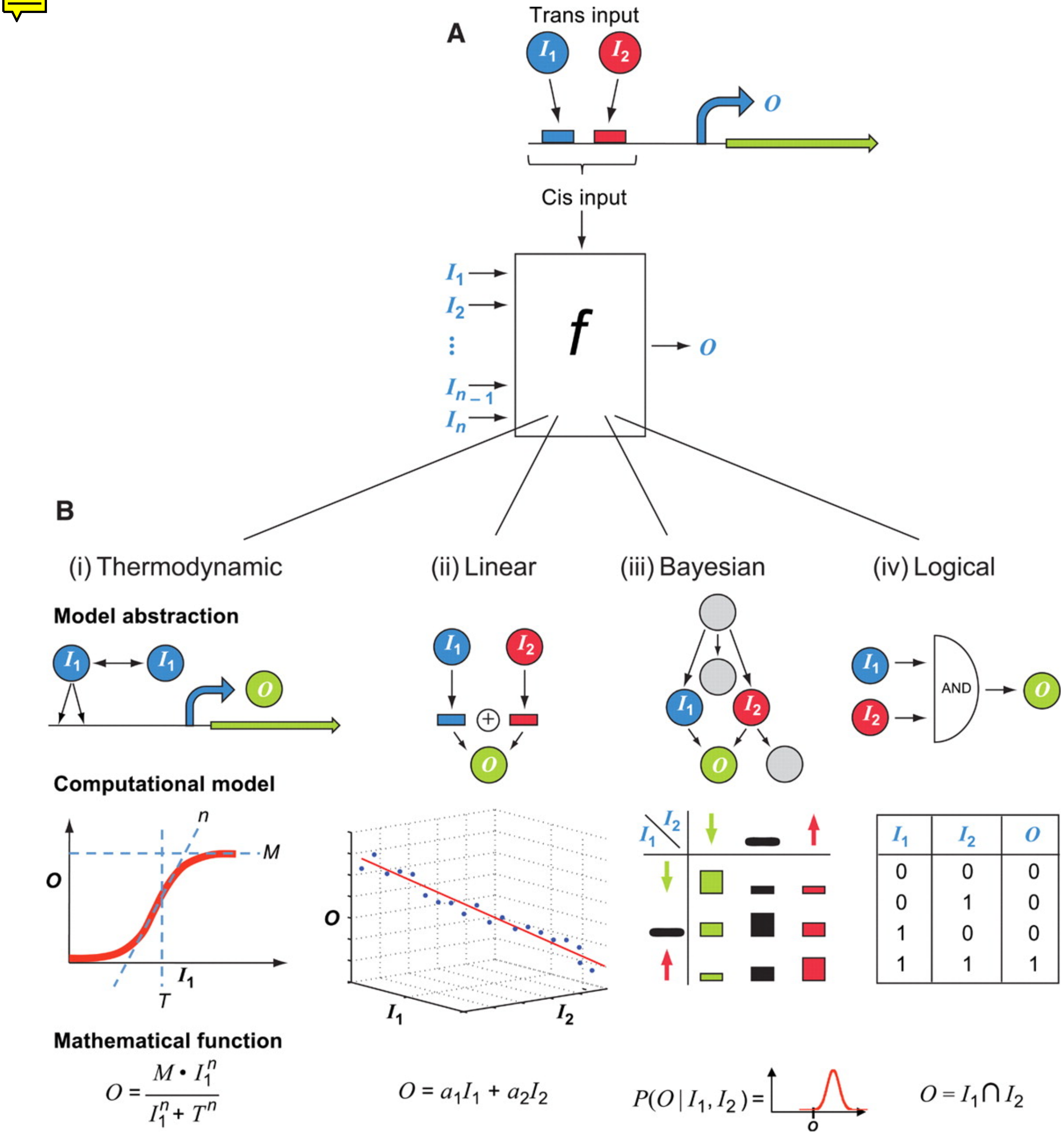
Other biological networks

- Disease – “disease gene” association networks
  - Link diseases that are caused by the same gene
  - Link genes if they cause the same disease
- Drug – “drug target” association networks
  - Link drugs if they target the same gene (protein)
  - Link genes (protiens) if they are targeted by the same drug

Systems Biology: The inference of networks from high dimensional genomics data

Ka Yee Yeung

Nov 3, 2011



I_1	I_2	O
0	0	0
0	1	0
1	0	0
1	1	1

$$O = \frac{M \cdot I_1^n}{I_1^n + T^n}$$

$$O = a_1 I_1 + a_2 I_2$$

$$P(O \mid I_1, I_2) =$$

$$O = I_1 \cap I_2$$

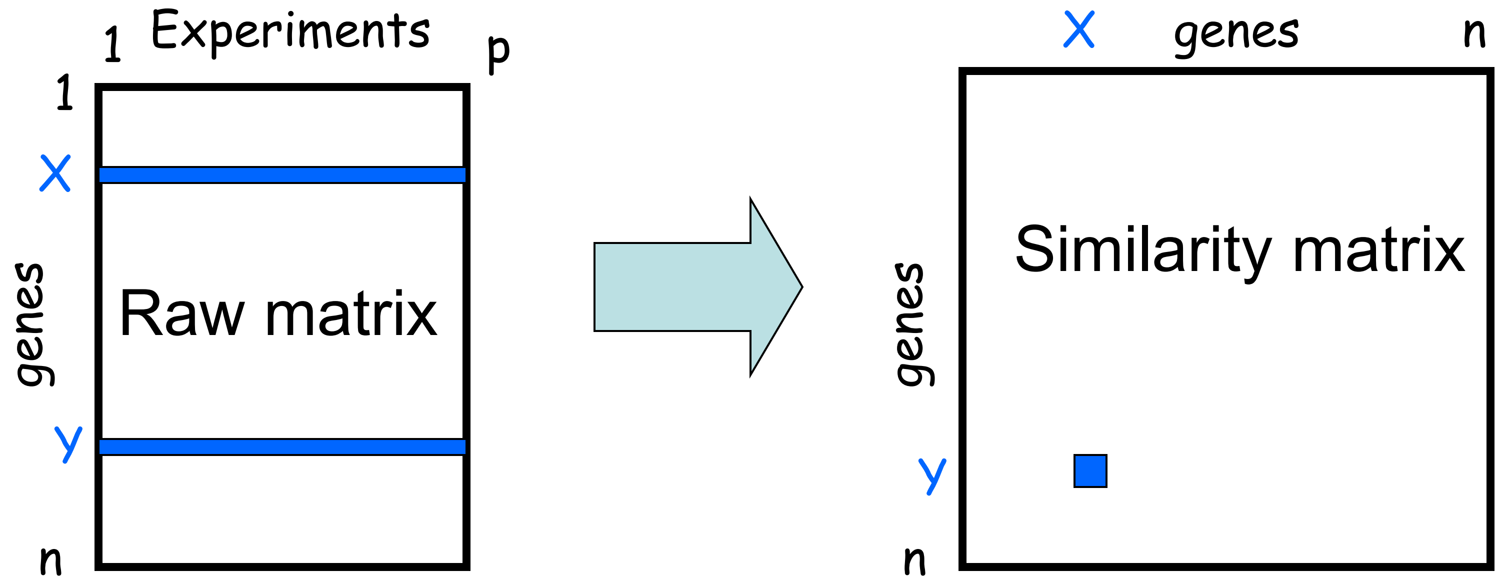
A **gene-regulation function** describes how inputs such as transcription factors and regulatory elements, are transformed into a gene’s mRNA level.

Kim et al. Science 2009

Network construction methods

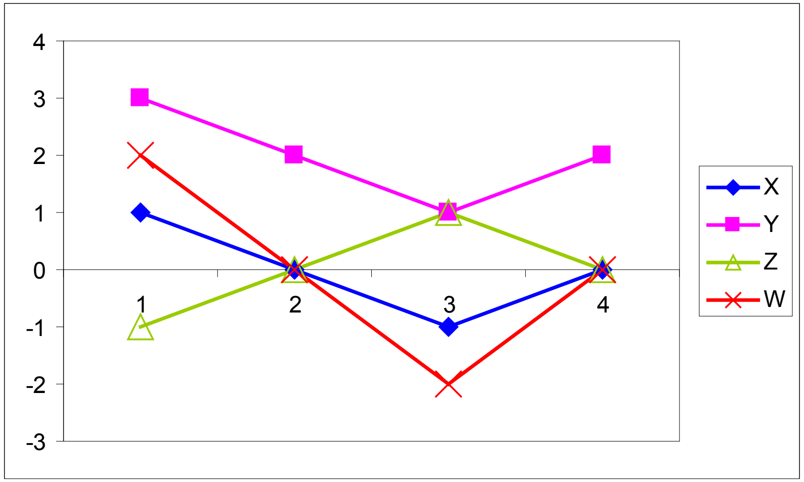
- Co-expression networks
- Bayesian networks
- Regression-based methods

Correlation: pairwise similarity



Raw matrix

Similarity matrix



Correlation (X,Y) = 1

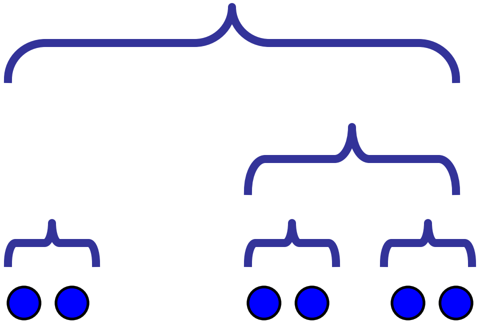
Correlation (X,Z) = -1

Correlation (X,W) = 1

Clustering algorithms

- Inputs:
  - Similarity matrix
  - Number of clusters or some other parameters
- Many different classifications of clustering algorithms:
  - Hierarchical vs partitional
  - Heuristic-based vs model-based
  - Soft vs hard

Hierarchical Clustering

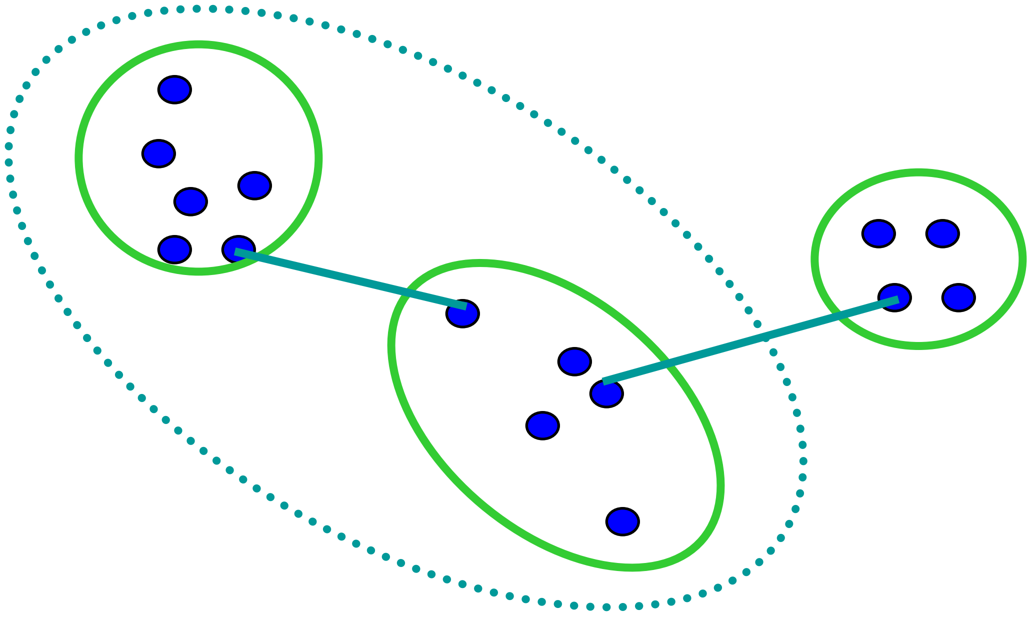


dendrogram

- Agglomerative (bottom-up)
- Algorithm:
  - Initialize: each item a cluster
  - Iterate:
    - select two most similar clusters
    - merge them
  - Halt: when required number of clusters is reached

Hierarchical: Single Link

- cluster similarity = similarity of two most similar members

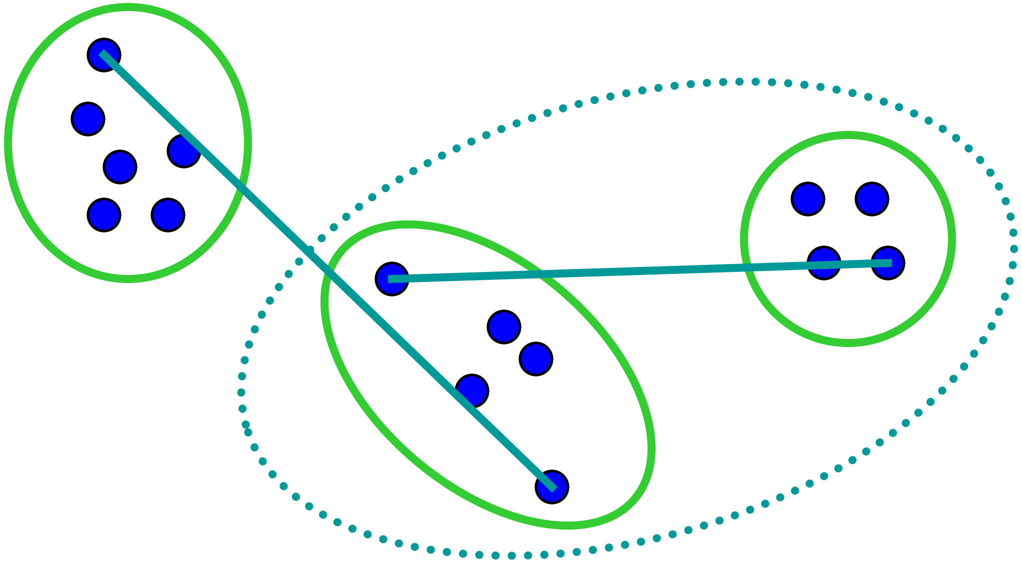


- Potentially long and skinny clusters

+ Fast

Hierarchical: Complete Link

- cluster similarity = similarity of two least similar members

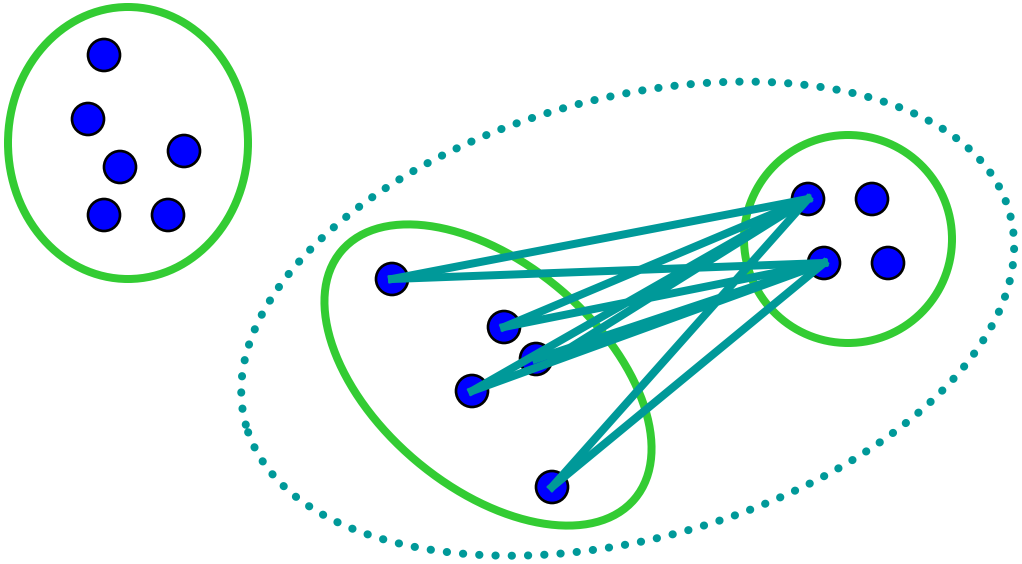


+ tight clusters

- slow

Hierarchical: Average Link

- cluster similarity = average similarity of all pairs



+ tight clusters

- slow

CSE 549

Lecturer: Sael Lee

AMINO ACID SEQUENCE ALIGNMENT

Slides provided by courtesy of Dr. D. Kihara @ Purdue

KEY ISSUES IN SEQUENCE ALIGNMENT

- What sort of alignment should be considered?

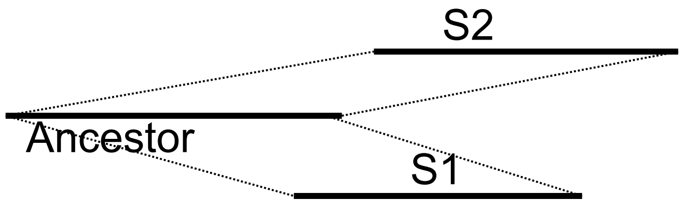
- What scoring system should be used to rank alignment ?

- What algorithm should be used to find optimal (or good) scoring alignments ?

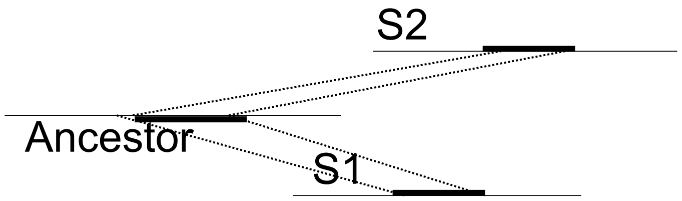
- What statistical methods should be used to evaluate the significance of an alignment score?

TYPES OF ALGINMENT

- Global Alignment
  - Assuming that the *complete* sequences are the results of evolution from the same ancestor sequence



- Local Alignment
  - Align segments of the sequences so that the segments are evolutionarily related



SCORING (1)

- Match – mismatch
  - Match : +1, mismatch: 0
  - Identity matrix (often used for DNA sequences)

DNA

	a	c	g	t
a	1	0	0	0
c	0	1	0	0
g	0	0	1	0
t	0	0	0	1

Amino acid

	A	R	N	D	…
A	1	0	0	0	0
R	0	1	0	0	0
N	0	0	1	0	0
D	0	0	0	1	0
…	0	0	0	0	..

ALIGNMENT SCORE

- Add up the terms (assume independence).
- DNA

```
atgatcaagtactttaagaagcagaagcggc
||||| ||| |||||||||| || ||| |||
atgataaagcactttaagaaacaaaagaggc
```

  - 26 matches / 31 nt (= 83.9%) **(identity)**

- Protein

```
SSWRVISSIEQKTER
.:::..:::::: : 
ASWRILSSIEQKEEA
```

  - 10 matches / 15 aa (= 66.7%) **(identity)**

SCORING (2)

- Amino acid **substitution (similarity) matrix**
  - Counting similarity of amino acids
  - Analyze statistics of known alignments
  - PAM, BLOSUM series, matrices specific for a certain type of proteins, e.g. membrane proteins

	A	R	N	D	…
A	5	-2	-1	-2	
R	-2	7	0	-1	
N	-1	0	6	2	
D	-2	-1	2	7	
…					..

(BLOSUM45)

SCORING MATRICES FOR PROTEIN SEQUENCE ALIGNMENT

- Define scores for amino acid pairs in sequence alignments
- Reflect “similarity” of amino acid residues
- Most often a substitution matrix is used.
- *Amino acid substitution matrix* is not necessarily symmetric,
  - Reflecting the difference of the mutation probability of A > B from B > A (A, B: two different amino acids)
  - Correspond to the logarithm of the relative likelihood that the sequences are related, compared to being unrelated.

ALIGNMENT SCORE: SMITH-WATERMAN SCORE

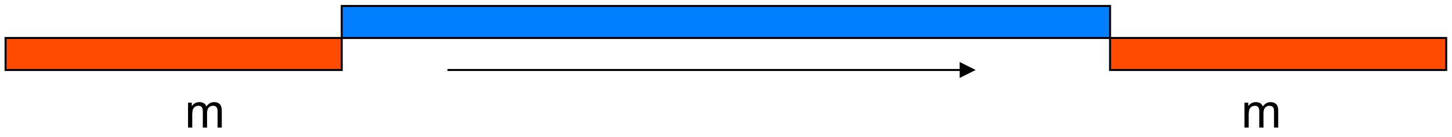
- BLOSUM45, Gap penalty: -12/-2
- Add up each term.
- Sequence identity: 15/29 = 51.7%
- Smith-Waterman Score: 63

```
SSWRVISSIEQKTER--NEKKQQ-GKEYR
.:::..:::::: :   :. . .  ::::
ASWRILSSIEQKEEAKGNDVSVKRIKEYR
```

1+4+15+7+3+2+4+4+5+6+…. +6-2-12-2+6… … -12…

GAPLESS ALIGNMENT

- Gaps not allowed in the middle
  - Scan one sequence along the other one
- Number of possible alignments
  - Sequence length: m, n



  - m + n + 1 , If m=n, 2n+1; i.e O(n)
- Application: finding a known motif in a sequence
- How to choose the “best” alignment?
  - Scoring scheme

ALIGNMENT WITH GAPS

- Scoring: AA matrix + gap penalty
- Gap penalty for a gap of length g:
  - Linear model: -gd (d : gap penalty, d>0)
  - Affine model: -d – (g-1)e

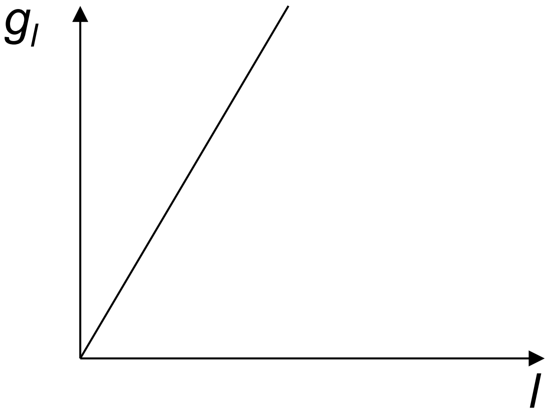
    (d: opening penalty, e: extension penalty. d > e > 0)
- Number of possible alignments

  $\binom{m+n}{m}$ If m=n, $\binom{m+n}{m} = \binom{2n}{n} = \frac{(2n)!}{(n!)^2} \cong \frac{2^{2n}}{\sqrt{2\pi n}}$ i.e. $O(4^n)$
- Algorithmic challenge: Given AA matrix, gap penalty, find th e alignment with the best score.

LINEAR AND AFFINE GAP PENALTIES

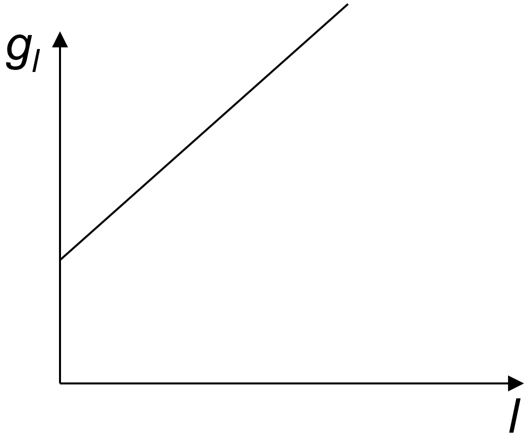
- Linear:

$$g_l = g \cdot l$$



- Affine:

$$g_l = g_{open} + (l-1) \cdot g_{extend}$$



GLOBAL ALIGNMENT

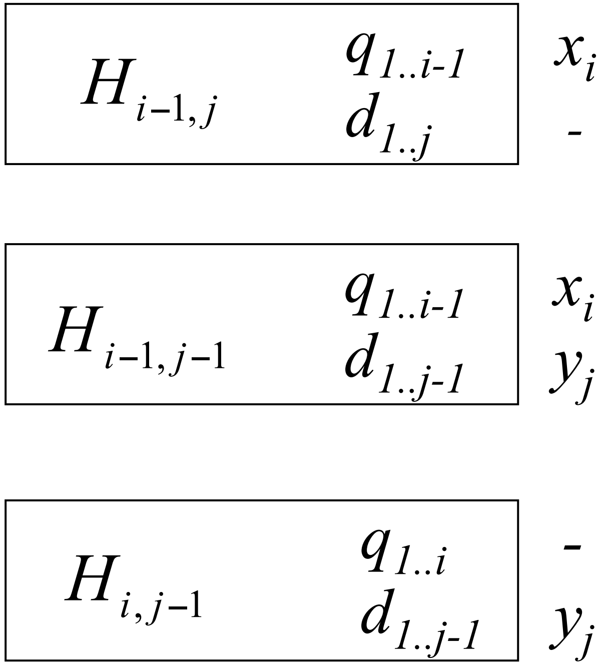
FINDING THE HIGHEST SCORING ALIGNMENT

- Problem:
  - Given two sequences, a scoring matrix, and a gap penalty, find the alignment with the highest score
- Large number of possible alignments
  - Cannot generate all and score them to find the best
  - Algorithm: **dynamic programming (DP) algorithm**

    (Needleman-Wunsch Algorithm)

RECURRENCE RELATION IN DP

- Assume that, $H_{i-1,j-1}$, $H_{i-1,j}$, $H_{i,j-1}$ are known



$H_{i-1,j}$ $q_{1..i-1}$ x_i
$d_{1..j}$ -

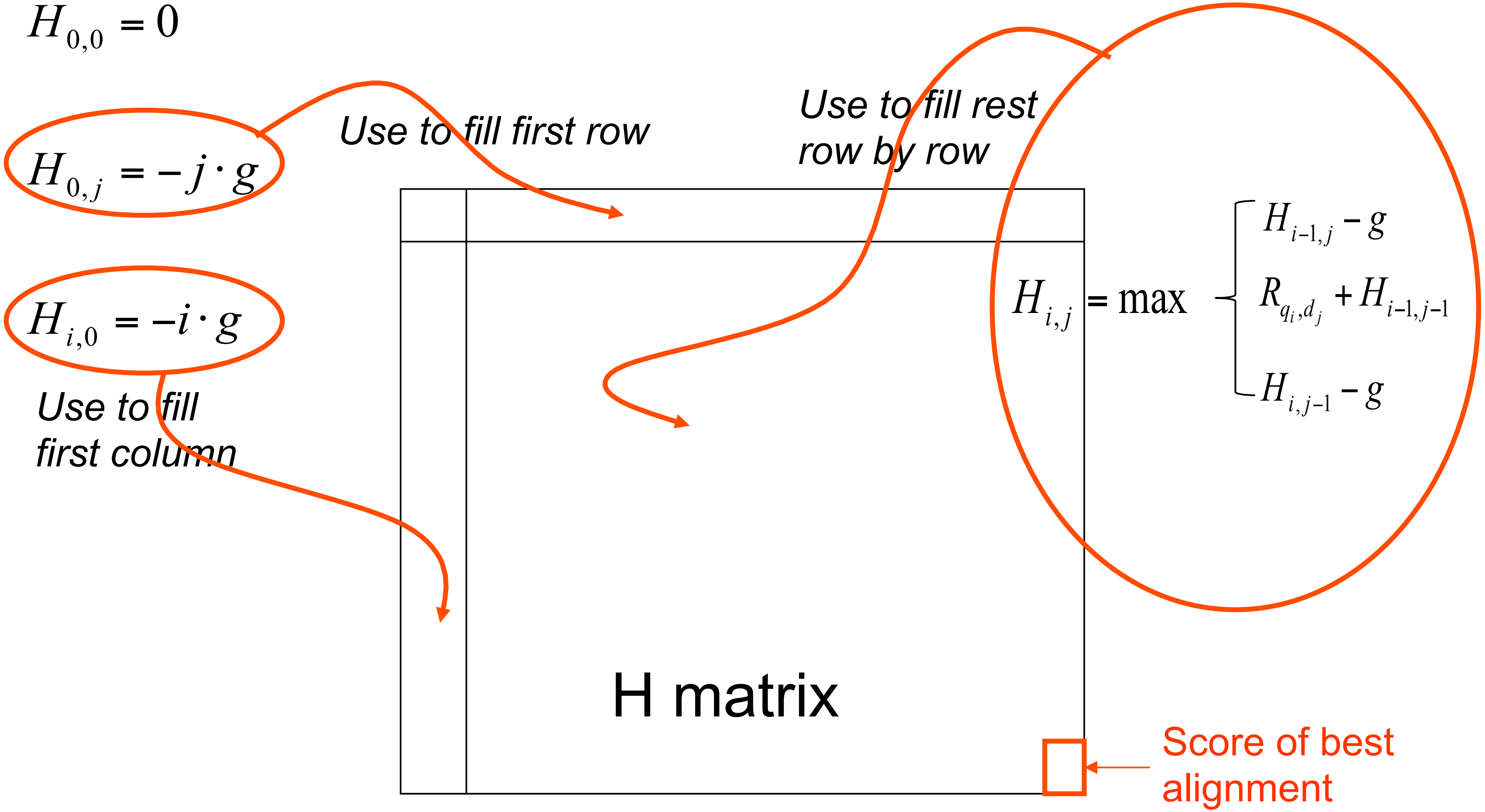
$H_{i-1,j-1}$ $q_{1..i-1}$ x_i
$d_{1..j-1}$ y_j

$H_{i,j-1}$ $q_{1..i}$ -
$d_{1..j-1}$ y_j

$$H_{i,j} = \max \begin{cases} H_{i-1,j} - g \\ S(x_i, y_j) + H_{i-1,j-1} \\ H_{i,j-1} - g \end{cases}$$

Where g is the gap open penalty and $S(x_i, y_j)$ is the similarity score obtained from substitution matrix for residue type of x_i and y_j

CALCULATING SCORE OF BEST ALIGNMENT USING MATRIX



$$H_{0,0} = 0$$

$$H_{0,j} = -j \cdot g$$

Use to fill first row

$$H_{i,0} = -i \cdot g$$

Use to fill first column

Use to fill rest row by row

$$H_{i,j} = \max \begin{cases} H_{i-1,j} - g \\ R_{q_i,d_j} + H_{i-1,j-1} \\ H_{i,j-1} - g \end{cases}$$

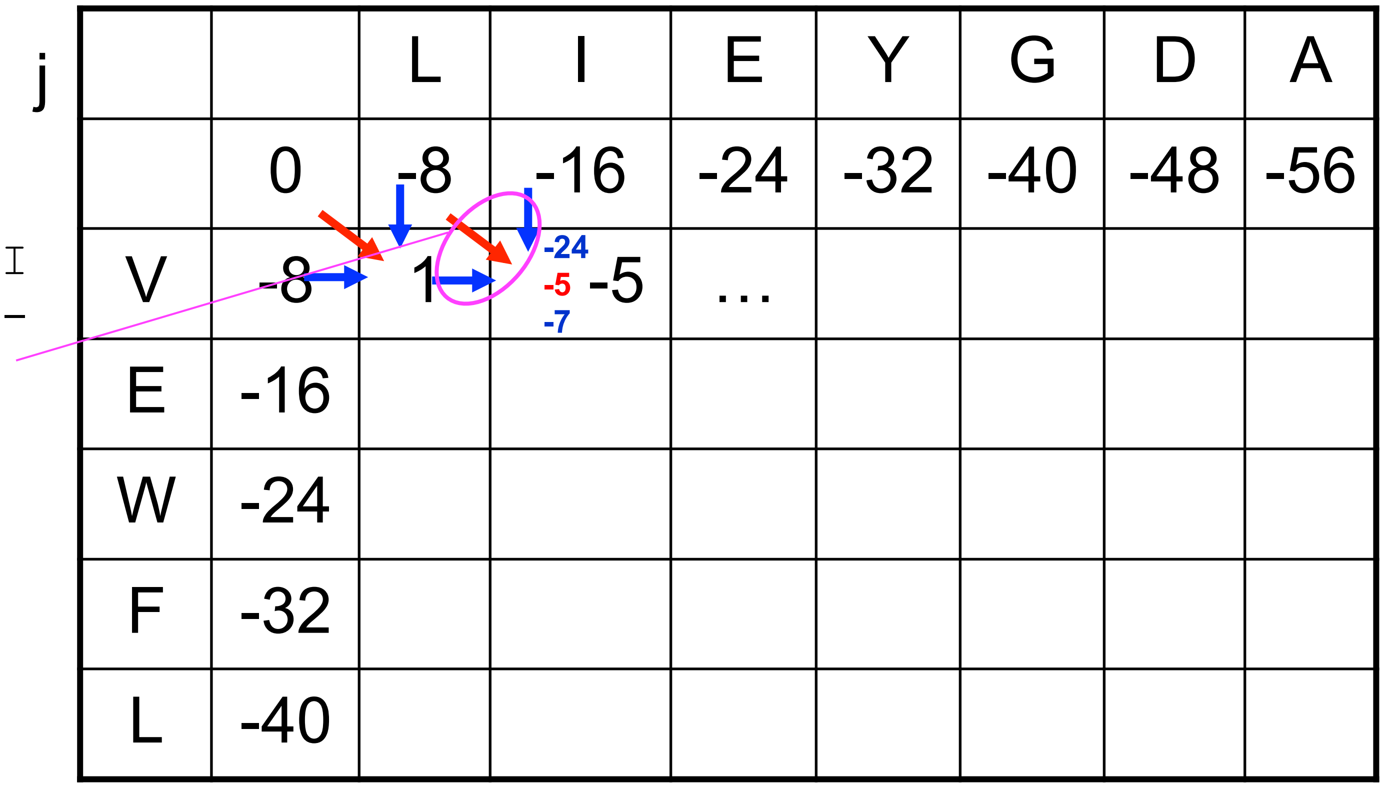
Score of best alignment

GLOBAL DP MATRIX, H(I,J)

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

BLOSUM45

```
LI   LI-    LI
-V   --V    V-
   or     or
```



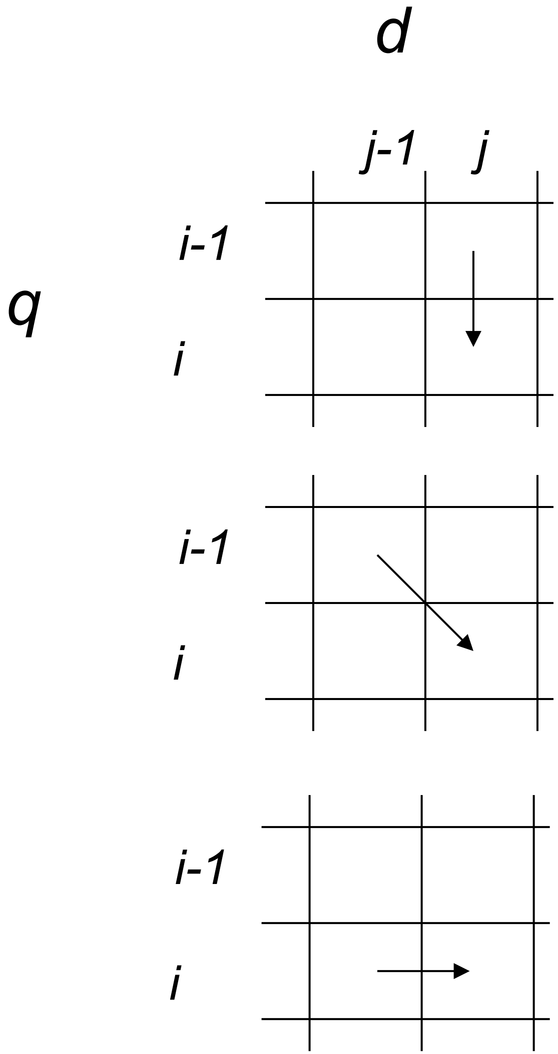
i

j

		L	I	E	Y	G	D	A
	0	-8	-16	-24	-32	-40	-48	-56
V	-8	1	-24 -5 -5 -7	…				
E	-16							
W	-24							
F	-32							
L	-40							

BLOSUM45
S(V,L) = 1
S(V,I) = 3
Gap = -8

TRACEBACK, ALIGNMENT



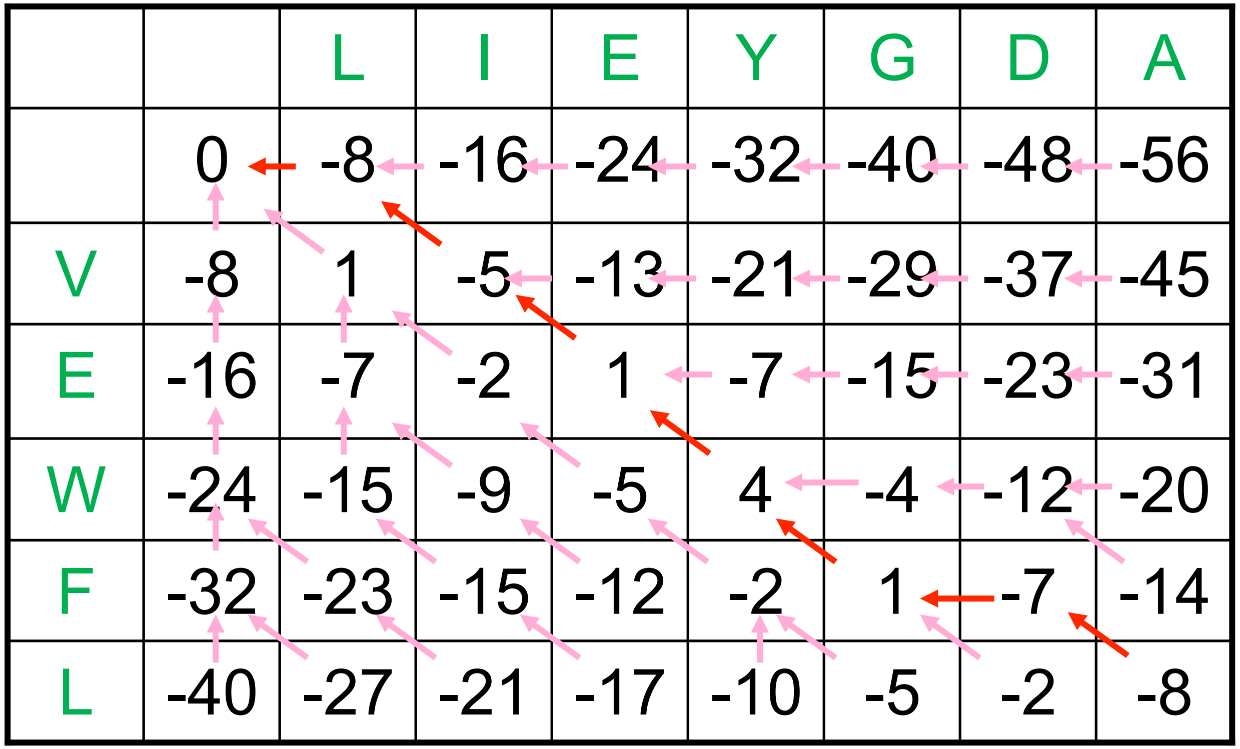
$$\begin{array}{ll} q' & q_i \\ d' & - \end{array}$$

$$\begin{array}{ll} q' & q_i \\ d' & d_i \end{array}$$

$$\begin{array}{ll} q' & - \\ d' & d_j \end{array}$$

GLOBAL DP MATRIX, H(I,J)

Fill this table from top-left to bottom-right
Trace back to get the alignment!



		L	I	E	Y	G	D	A
	0	-8	-16	-24	-32	-40	-48	-56
V	-8	1	-5	-13	-21	-29	-37	-45
E	-16	-7	-2	1	-7	-15	-23	-31
W	-24	-15	-9	-5	4	-4	-12	-20
F	-32	-23	-15	-12	-2	1	-7	-14
L	-40	-27	-21	-17	-10	-5	-2	-8

```
LIEYGDA
-VEWF-L
```

TIME COMPLEXITY

- Sequences of lengths *n* and *m*

$$O(nm)$$

- Two sequences of length *l*

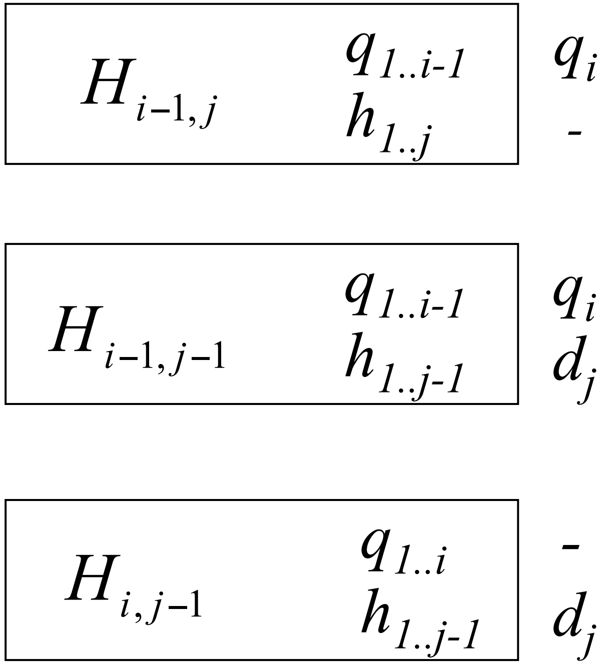
$$O(l^2)$$

LOCAL ALGINMENT

THE LOCAL ALIGNMENT

- Aims to identify only very similar region of two protein sequences
- Should ignore negatively contributing suffixes of align ments
- Score of best local alignment – highest value in dyna mic programming matrix
- Alignment found by tracing back from maximum value until cell with value 0 (zero) has been reached

DP RECURRENCE RELATION



$H_{i-1,j}$ $\quad q_{1..i-1}$ / $h_{1..j}$ $\quad q_i$ / -

$H_{i-1,j-1}$ $\quad q_{1..i-1}$ / $h_{1..j-1}$ $\quad q_i$ / d_j

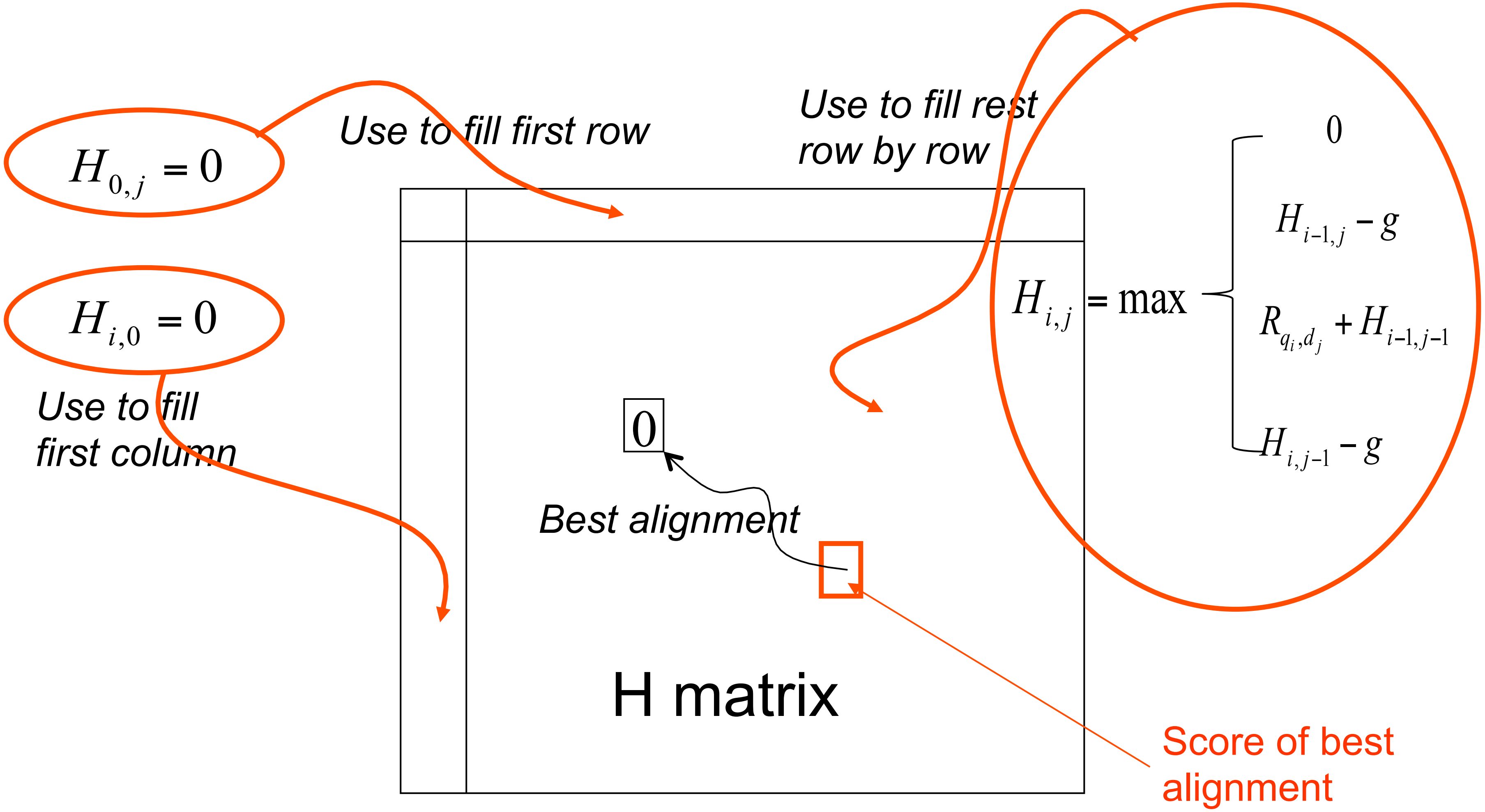
$H_{i,j-1}$ $\quad q_{1..i}$ / $h_{1..j-1}$ $\quad$ - / d_j

Empty alignment

$$H_{i,j} = \max \begin{cases} H_{i-1,j} - g \\ R_{q_i,d_j} + H_{i-1,j-1} \\ H_{i,j-1} - g \\ 0 \end{cases}$$

Effectively allows for removal of negatively contributing prefixes.

CALCULATING BEST LOCAL ALIGNMENT



$H_{0,j} = 0$

Use to fill first row

$H_{i,0} = 0$

Use to fill first column

Use to fill rest row by row

$$H_{i,j} = \max \begin{cases} 0 \\ H_{i-1,j} - g \\ R_{q_i,d_j} + H_{i-1,j-1} \\ H_{i,j-1} - g \end{cases}$$

$\boxed{0}$

Best alignment

H matrix

Score of best alignment

EXAMPLE OF LOCAL DP MATRIX, H(I,J)

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

		L	I	E	Y	G	D	A
	0	0	0	0	0	0	0	0
V	0	1	3	0	0	0	0	0
E	0	0	0	9	1	0	2	0
W	0	0	0	1	12	4	0	0
F	0	1	0	0	4	9	1	0
L	0	5	3	0	0	1	6	0

BLOSUM45
Gap penalty = -8

```
IEY
VEW
```

TIME COMPLEXITY OF LOCAL ALIGNMENT

- Sequences of lengths *n* and *m*

$$O(nm)$$

- Two sequences of length *l*

$$O(l^2)$$

AFFINE GAP ALGORITHM 2

M:

S/T		L	I	E	Y	G	D	A
	0	-12	-14	-16	-18	-20	-22	-24
V	-12							
E	-14							
W	-16							
F	-18							
L	-20							

Gap:
Opening: -12
Extension: -2

$$M(i, j) = \max \begin{cases} M(i-1, j-1) + s(S_i, T_j) & S_i \text{ align with } T_j \\ I(i-1, j-1) + s(S_i, T_j) & \text{Si align with gap} \\ J(i-1, j-1) + s(S_i, T_j) & \text{gap align with } T_j \end{cases}$$

AFFINE GAP ALGORITHM 2

Tj:

I:

Si:

S/T		L	I	E	Y	G	D	A
	0	(-∞)	(-∞)	(-∞)	(-∞)	(-∞)	(-∞)	(-∞)
V	-12							
E	-14							
W	-16							
F	-18							
L	-20							

$$I(i, j) = \max \begin{cases} M(i-1, j) - d & S_i \text{ align with initial gap} \\ I(i-1, j) - e & S_i \text{ align with extension gap} \end{cases}$$

AFFINE GAP ALGORITHM 2

Tj:

J:

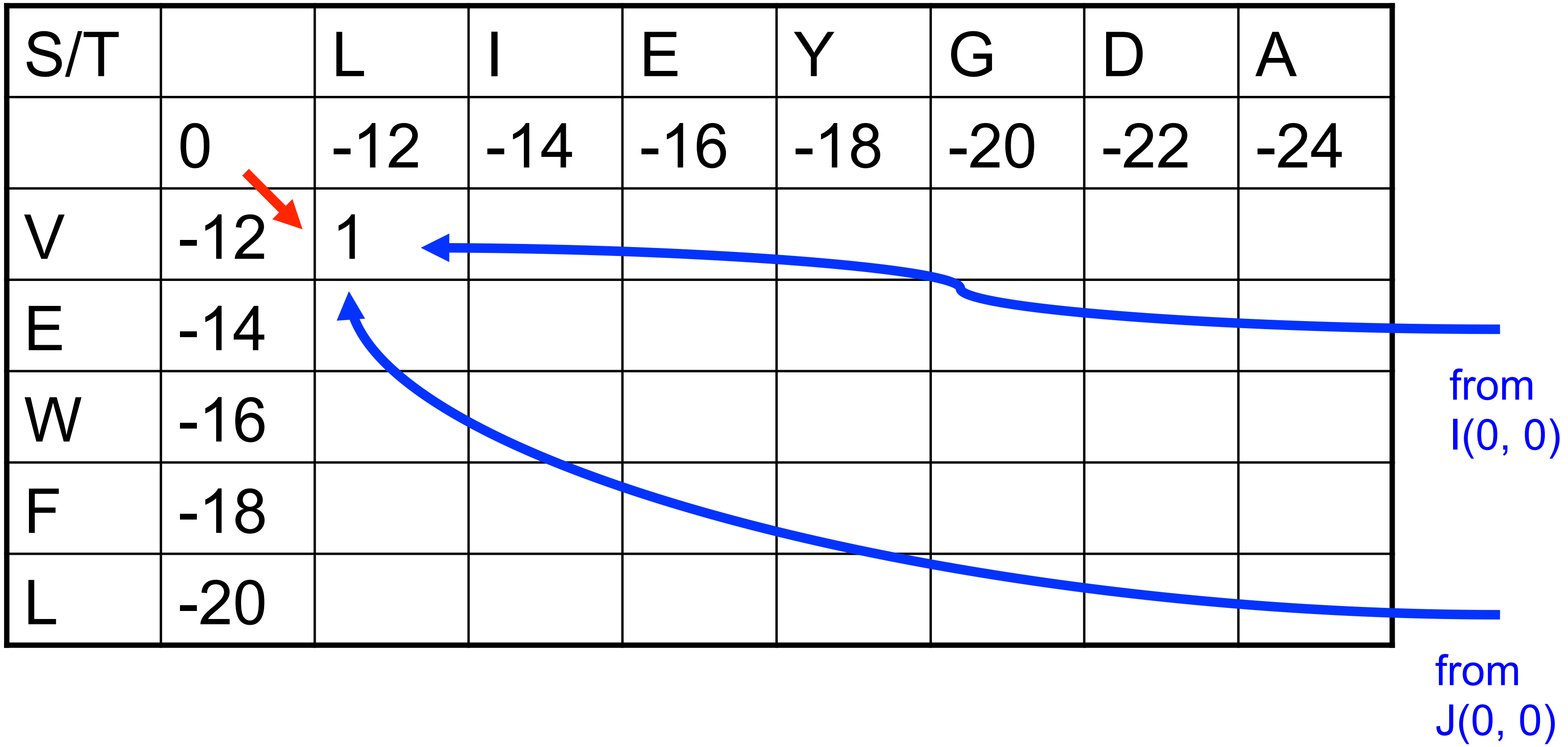
Si:

S/T		L	I	E	Y	G	D	A
	0	-12	-14	-16	-18	-20	-22	-24
V	(-∞)							
E	(-∞)							
W	(-∞)							
F	(-∞)							
L	(-∞)							

$$J(i, j) = \max \begin{cases} M(i, j-1) - d & \text{initial gap align with } T_j \\ J(i, j-1) - e & \text{extension gap align with } T_j \end{cases}$$

AFFINE GAP ALGORITHM 2

M:



S/T		L	I	E	Y	G	D	A
	0	-12	-14	-16	-18	-20	-22	-24
V	-12	1						
E	-14							
W	-16							
F	-18							
L	-20							

$$M(1, 1) = \max \begin{cases} 0 + 1 = 1 \\ 0 + 1 = 1 \\ 0 + 1 = 1 \end{cases}$$

AFFINE GAP ALGORITHM 2

I:

S_i/T_j		L	I	E	Y	G	D	A
	0	(-∞)	(-∞)	(-∞)	(-∞)	(-∞)	(-∞)	(-∞)
V	-12	-24	-26	-28	-30	-32	-34	-36
E	-14	-11						
W	-16							
F	-18							
L	-20							

$$I(1, 1) = \max \begin{cases} 1 - 12 = -11 \\ -24 - 2 = -26 \end{cases}$$

from M(1, 1)

AFFINE GAP ALGORITHM 2

J:

S_i/T_j		L	I	E	Y	G	D	A
	0	-12	-14	-16	-18	-20	-22	-24
V	(-∞)	-24	-11					
E	(-∞)	-26						
W	(-∞)	-28						
F	(-∞)	-30						
L	(-∞)	-32						

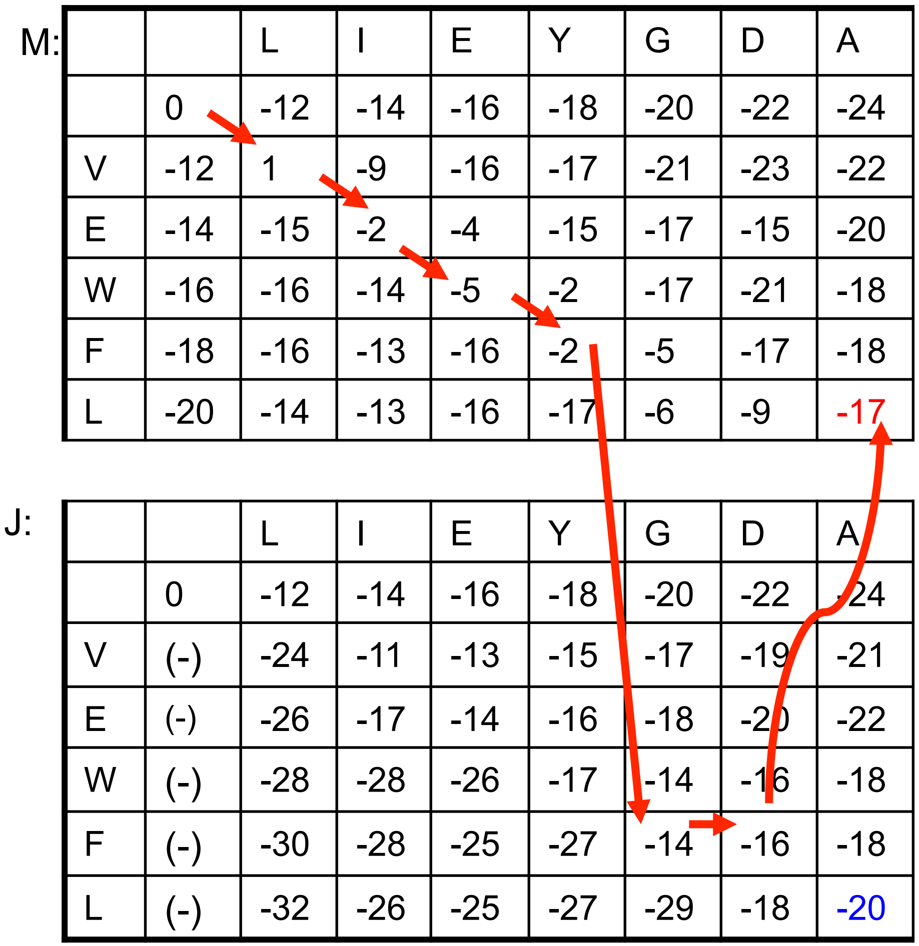
from M(1, 1)

$$J(1, 1) = \max \begin{cases} 1 - 12 = -11 \\ -24 - 2 = -26 \end{cases}$$

BLOSUM62 SCORE MATRIX

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0
R	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3
N	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3
D	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3
C	0	-3	-3	-3	9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2	-1
Q	-1	1	0	0	-3	5	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2
E	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2
G	0	-2	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-2	-3	-3
H	-2	0	1	-1	-3	0	0	-2	8	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-3	-1	3
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	-2	2	0	-3	-2	-1	-2	-1	1
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6	-4	-2	-2	1	3	-1
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7	-1	-1	-4	-3	-2
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4	1	-3	-2	-2
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-2	-2	0
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11	2	-3
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

AFFINE GAP ALGORITHM 2



M:

		L	I	E	Y	G	D	A
	0	-12	-14	-16	-18	-20	-22	-24
V	-12	1	-9	-16	-17	-21	-23	-22
E	-14	-15	-2	-4	-15	-17	-15	-20
W	-16	-16	-14	-5	-2	-17	-21	-18
F	-18	-16	-13	-16	-2	-5	-17	-18
L	-20	-14	-13	-16	-17	-6	-9	-17

I:

		L	I	E	Y	G	D	A
	0	(-)	(-)	(-)	(-)	(-)	(-)	(-)
V	-12	-24	-26	-28	-30	-32	-34	-36
E	-14	-11	-21	-30	-32	-33	-35	-34
W	-16	-13	-13	-16	-27	-29	-27	-32
F	-18	-15	-15	-17	-14	-29	-29	-30
L	-20	-17	-17	-19	-16	-17	-29	-30

J:

		L	I	E	Y	G	D	A
	0	-12	-14	-16	-18	-20	-22	-24
V	(-)	-24	-11	-13	-15	-17	-19	-21
E	(-)	-26	-17	-14	-16	-18	-20	-22
W	(-)	-28	-28	-26	-17	-14	-16	-18
F	(-)	-30	-28	-25	-27	-14	-16	-18
L	(-)	-32	-26	-25	-27	-29	-18	-20

V E W F - - L
L I E Y G D A

TIME COMPLEXITY

- Sequences of lengths *n* and *m*

$$O(nm)$$

- Two sequences of length *l*

$$O(l^2)$$

SUNY Korea
The State University of New York
한국뉴욕주립대학교

Stony Brook University

CSE 549

Lecturer: Sael Lee

AMINO ACID SEQUENCE ALIGNMENT II

Slides provided by courtesy of Dr. D. Kihara @ Purdue

SCORING MATRICES

SCORING MATRICES FOR AA SEQUENCE ALIGNMENT

- Define scores for amino acid pairs in sequence alignments
- Reflect "similarity" of amino acid residues

- *Amino acid scoring matrix/Amino acid similarity matrix* => symmetric
- *Amino acid substitution matrix* => not necessarily symmetric,
  - reflecting the difference of the mutation probability of A to B from B to A (A, B: two different amino acids)

PAM MATRICES (DAYHOFF, 1978)

- PAM: A Point Accepted Mutations.
  - Models the replacement of a single AA in the primary structure of a protein with another single AA that is accepted by natural selection.
    - Does not include silent mutations , mutations which are lethal, or mutations which are rejected by natural selection in other ways.
- PAM matrix: 20x20 AA substitution matrix
  - Each entry indicates the likelihood of the AA of that row being replaced with the AA of that column through a series of one or more PAM during a specified evolutionary interval, compared to these two AA being aligned by chance.

PAM MATRIX CONT.

- Different PAM matrices correspond to different lengths of time in the evolution of the protein sequence.
  - EX> PAM1: one accepted mutation per 100 residues
  - (n in the PAM_n matrix represents the number of mutations per 100 amino acids,)
- Start from a set of well manually curated sequence alignments
  - >85% sequence identity
  - 71 groups of homologous sequences
- Construct phylogenetic trees and estimate the history of the mutation events in the family
  - 1572 observed mutations in the phylogenetic trees of 71 families of closely related proteins.

PAM: COLLECTION OF DATA FROM PHYLOGENETIC TREES

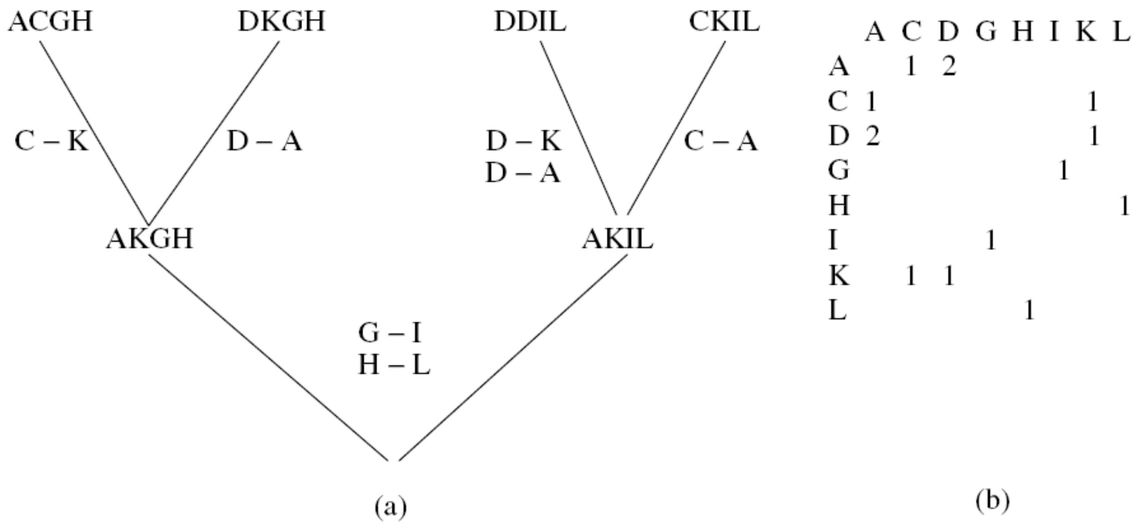
ACGH

DKGH

DDIL

CKIL

PAM SCORING MATRICES



	A	C	D	G	H	I	K	L
A		1	2					
C	1						1	
D	2						1	
G						1		
H								1
I				1				
K		1	1					
L					1			

Figure 5.4 (a) A small phylogenetic tree of four observed sequences, and two derived parent sequences. (b) The mutations are on the edges. The numbers of different mutations are shown in the table.

COMPUTING PROBABILITY OF *A* CHANGING TO *B* IN A CERTAIN TIME Τ

- Count for each branch in the phylogenetic trees, the number of mismatches recorded and compute fequencey
  - f_{ab} : frequency of mutation from *a* => *b or b* => *a* (assume symmetry i.e. $f_{ab} = f_{ba}$)
- Compute mutability of *a:* $f_a = \Sigma_{b\neq a} f_{ab}$
  - the total number of mutation involving *a*
- Compute $f = \Sigma_a f_a$:
  - <u>twice</u> the total number of mutations
- Compute p_a where $\Sigma_a p_a = 1$:
  - the frequency of amino acid *a*,
- Compute m_a : the relative mutability of *a*
  - the probability that *a* will mutate in the evolutionary time τ

CALCULATING M_A AND M_{AB} IN THE TIME τ

- Consider the time τ = 1 PAM
  - the time while one mutation is accepted per 100 res.
- The probability that mutation is from *a* is:

  $\frac{1}{2} f_a/(f/2) = f_a/f$,

  (1/2 comes from $f_{ab} = f_{ba}$)
- Among 100 res., there are $100p_a$ occurrences of *a*
- The relative mutability of *a* is
  - $m_a = (1/ 100p_a) f_a/f$
- The prob. that *a* will be mutated to *b* in the time τ
  - $M_{ab} = m_a (f_{ab}/f_a)$ for $a \neq b$; $M_{aa} = 1 - m_a$

SUBSTITUTION MATRIX M^1

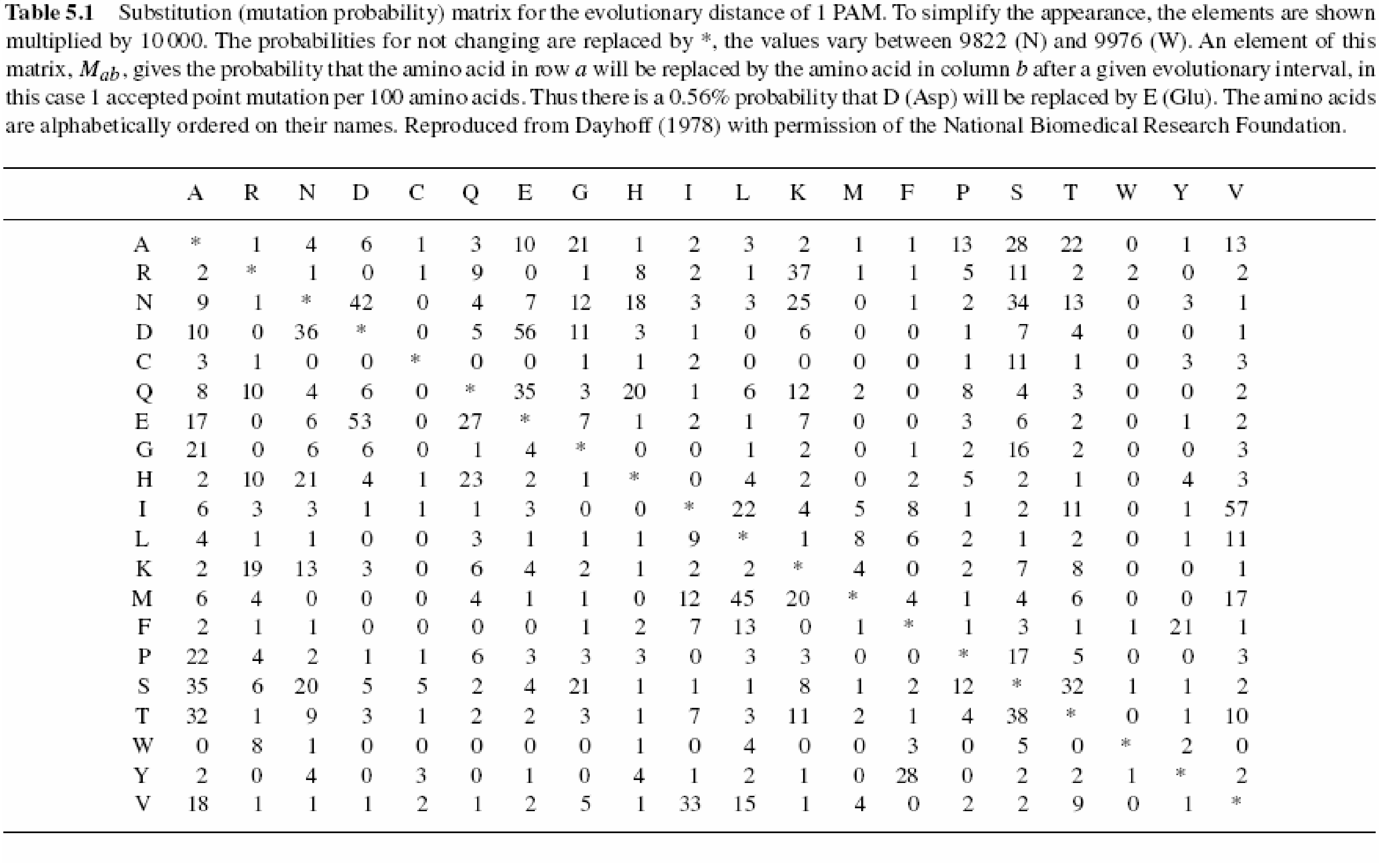


Table 5.1 Substitution (mutation probability) matrix for the evolutionary distance of 1 PAM. To simplify the appearance, the elements are shown multiplied by 10 000. The probabilities for not changing are replaced by *, the values vary between 9822 (N) and 9976 (W). An element of this matrix, M_{ab}, gives the probability that the amino acid in row a will be replaced by the amino acid in column b after a given evolutionary interval, in this case 1 accepted point mutation per 100 amino acids. Thus there is a 0.56% probability that D (Asp) will be replaced by E (Glu). The amino acids are alphabetically ordered on their names. Reproduced from Dayhoff (1978) with permission of the National Biomedical Research Foundation.

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	*	1	4	6	1	3	10	21	1	2	3	2	1	1	13	28	22	0	1	13
R	2	*	1	0	1	9	0	1	8	2	1	37	1	1	5	11	2	2	0	2
N	9	1	*	42	0	4	7	12	18	3	3	25	0	1	2	34	13	0	3	1
D	10	0	36	*	0	5	56	11	3	1	0	6	0	0	1	7	4	0	0	1
C	3	1	0	0	*	0	0	1	1	2	0	0	0	0	1	11	1	0	3	3
Q	8	10	4	6	0	*	35	3	20	1	6	12	2	0	8	4	3	0	0	2
E	17	0	6	53	0	27	*	7	1	2	1	7	0	0	3	6	2	0	1	2
G	21	0	6	6	0	1	4	*	0	0	1	2	0	1	2	16	2	0	0	3
H	2	10	21	4	1	23	2	1	*	0	4	2	0	2	5	2	1	0	4	3
I	6	3	3	1	1	1	3	0	0	*	22	4	5	8	1	2	11	0	1	57
L	4	1	1	0	0	3	1	1	1	9	*	1	8	6	2	1	2	0	1	11
K	2	19	13	3	0	6	4	2	1	2	2	*	4	0	2	7	8	0	0	1
M	6	4	0	0	0	4	1	1	0	12	45	20	*	4	1	4	6	0	0	17
F	2	1	1	0	0	0	0	1	2	7	13	0	1	*	1	3	1	1	21	1
P	22	4	2	1	1	6	3	3	3	0	3	3	0	0	*	17	5	0	0	3
S	35	6	20	5	5	2	4	21	1	1	1	8	1	2	12	*	32	1	1	2
T	32	1	9	3	1	2	2	3	1	7	3	11	2	1	4	38	*	0	1	10
W	0	8	1	0	0	0	0	0	1	0	4	0	0	3	0	5	0	*	2	0
Y	2	0	4	0	3	0	1	0	4	1	2	1	0	28	0	2	2	1	*	2
V	18	1	1	1	2	1	2	5	1	33	15	1	4	0	2	2	9	0	1	*

CALCULATE M^{Z} BY MATRIX MULTIPLICATION

Example Z=2

- 2 mutations per 100 residues
- A residue *a* can be changed to residue *b* after 2 PAM of following reasons:

1. *a* is mutated to *b* in first PAM, unchanged in the next, with probability $M_{ab}M_{bb}$
2. *a* is unchanged in first PAM, changed in the next, probability $M_{aa}M_{ab}$
3. *a* is mutated to an amino acid *x* in the first PAM, and then to *b* in the next, probability $M_{ax}M_{xb,}$ *x* being any amino acid unequal (*a*,*b*)

These three cases are disjunctive, hence

$$M_{ab}^{2} = M_{ab}M_{bb} + M_{aa}M_{ab} + \sum_{x \notin \{a,b\}} M_{ax}M_{xb} = \sum_{x \in M} M_{ax}M_{xb}$$

CONVERTING FROM A SUBSTITUTION MATRIX TO A SCORING MATRIX

- In a substitution matrix not symmetric in general,
  - $M_{ab} \neq M_{ba}$ (*a* in sequence q, *b* in sequence d)
- To remove the effect of the frequent occurrence of *b* in sequence d, the **odds scoring matrix** is
  - $O_{ab} = M_{ab}/p_b$
  - O_{ab} is symmetric ($O_{ab} = O_{ba}$, p. 110, middle)
- **Log-odds matrix R:**
  - $R_{ab} = log\ O_{ab}$

BLOSUM (HENIKOFF & HENIKOFF)

- BLOSUM (BLOcks SUbstitution Matrix) matrix is a substitution matrix used to score alignments between evolutionarily divergent protein sequences introduced by Henikoff and Henikoff in 1992

- Make multiple alignments consist of sequences sharing more than X% sequence identity
- Discover blocks not containing gaps (used over 2,000 blocks)

```
 ...KIFIMK.......GDEVK...
...NLFKTR       GDSKK...
   KIFKTK       GDPKA
   KLFESR       GDAER
   KIFKGR       GDAAK
```

- For each column in each block, counted the number of occurrences of each pair of AA
  - 210 different pairs (combination with repetition: (20+2-1)! /(2!(20-1)!))

BLOSUM CONT

- A block of length *w* from an alignment of *n* sequences has T=$w*n(n-1)/2$ possible occurrences of amino acid pairs
  - Let h_{ab} be the number of occurrences of the pair (*ab*) in all blocks ($h_{ab}=h_{ba}$)
  - *T* total number of pairs
  - $f_{ab}=h_{ab}/T$
- Constructing logodds matrix : $R_{ab}=log(f_{ab}/e_{ab})$
  - with background probabilities of finding the amino acids a and in any protein sequence as p_a
  - $e_{aa}=p_a p_a$
  - $e_{ab}=p_a p_b + p_b p_a = 2\, p_a p_b$ *for* $a \neq b$

COMPARING PAM AND BLOSUM

- PAM: based on an evolutionary model (tree)
- PAM1 is multiplied to obtain PAMx (the larger x, the more distant)

- BLOSUM: Based on common regions in protein families
- Simple to compute
- BLOSUMx (e.g. x=45, 62, 80, the larger more closer)

ANALYSIS OF SCORING MATRICES

- PAMx or BLOSUMy is designed for aligning sequences of that range
  - i.e. BLOSUM50 cannot align very distantly related sequences by definition
- Starts from a set of pairwise (multiple) alignments
  - alignments > scoring matrix > alignment
- Can develop a scoring matrix from any set of alignments following the BLOSUM's method
- There are many AAindex database

  http://www.genome.ad.jp/dbget/aaindex.html

MULTIPLE ALIGNMENT

USE OF ALIGNMENTS

- High sequence similarity usually means significant structural and/or functional similarity.
- Homolog proteins (common ancestor) can vary significantly in large parts of the sequences, but still retain common 2D-patterns, 3D-patterns or common active site or binding site.
- Comparison of several sequences in a family can reveal what is common for the family. Conserved regions can be significant when regarding all of the sequences, but need not if regarding only two.

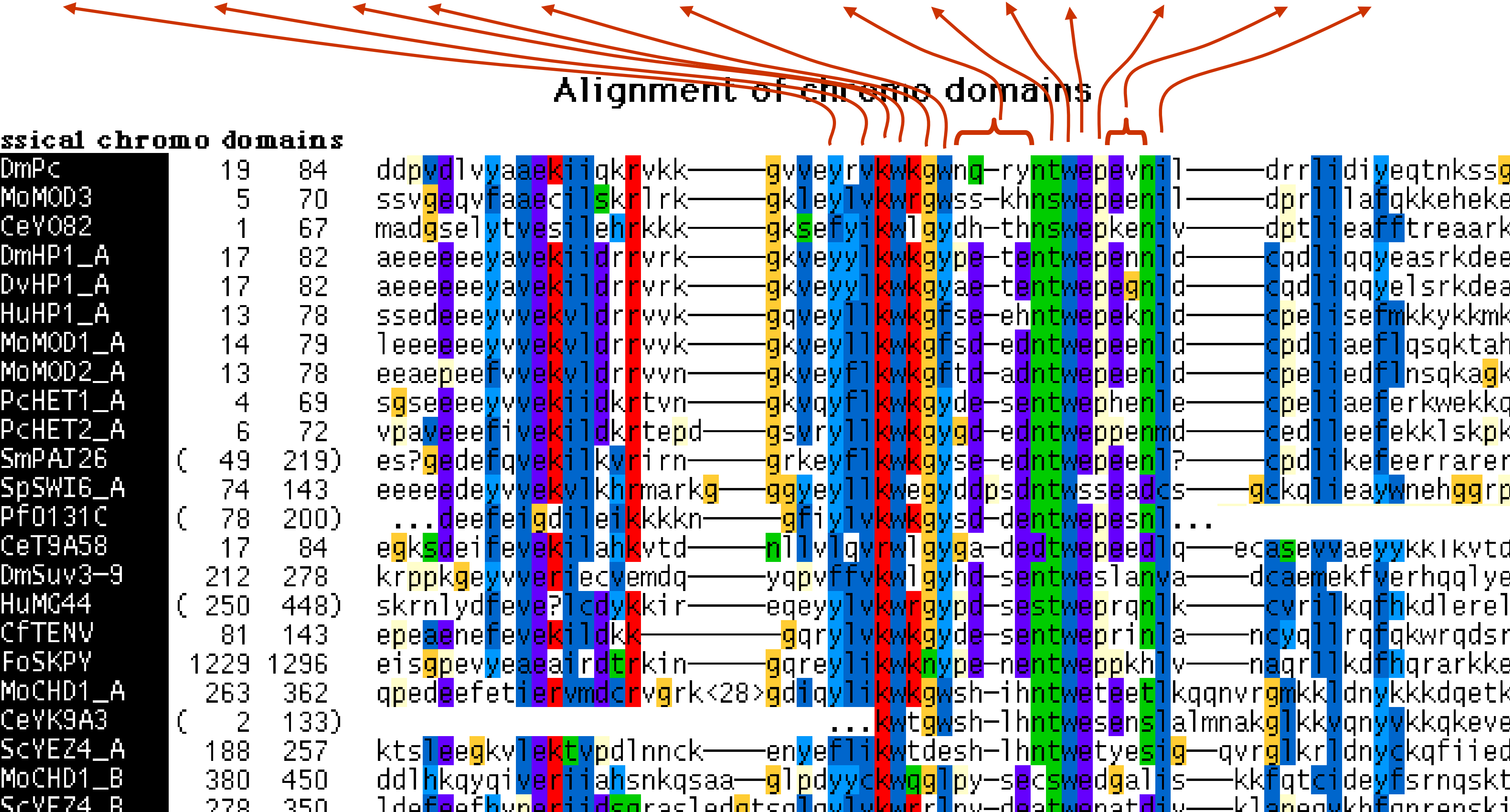
- Multiple alignment can be used to derive evolutionary history.
- Conserved positions : structurally/functionally important

USE OF ALIGNMENTS
- MAKE PATTERNS/PROFILES

- Can make a *profile* or a *pattern* that can be used to match against a sequence database and identify *new family members*
- Profiles/patterns can be used to predict family membership of *new* sequences
- *Databases* of profiles/patterns
    - PROSITE
    - PFAM
    - PRINTS
    - …

PATTERN FROM ALIGNMENT

[FYL]-x-[LIVMC]-[KR]-W-x-[GDNR]-[FYWLE]-x(5,6)-[ST]-W-[ES]-[PSTDN]-x(3)-[LIVMC]



Alignment of chromo domains

ssical chromo domains

```
DmPc          19   84   ddpvdlvyaaekiiqkrvkk————gvveyrvkwkgwnq-ryntwepevnil————drrlidiyeqtnkssg
MoMOD3         5   70   ssvgeqvfaaecilskrlrk————gkleylvkwrgwss-khnswepeenil————dprlllafqkkeheke
CeY082         1   67   madgselytvesilehrkkk————gksefyikwlgydh-thnswepkeniv————dptlieafftreaark
DmHP1_A       17   82   aeeeeeeyavekiidrrvrk————gkveyylkwkgype-tentwepennld————cqdliqqyeasrkdee
DvHP1_A       17   82   aeeeeeeyavekildrrvrk————gkveyylkwkgyae-tentwepegnld————cqdliqqyelsrkdea
HuHP1_A       13   78   ssedeeeyvvekvldrrvvk————gqveyllkwkgfse-ehntwepeknld————cpelisefmkkykkmk
MoMOD1_A      14   79   leeeeeeyvvekvldrrvvk————gkveyllkwkgfsd-edntwepeenld————cpdliaeflqsqktah
MoMOD2_A      13   78   eeaepeefvvekvldrrvvn————gkveyflkwkgftd-adntwepeenld————cpeliedflnsqkagk
PcHET1_A       4   69   sgseeeeyvvekiidkrtvn————gkvqyflkwkgyde-sentwephenle————cpeliaeferkwekkq
PcHET2_A       6   72   vpaveeefivekildkrtepd———gsvryllkwkgygd-edntweppenmd————cedlleefekklskpk
SmPAJ26     (  49  219) es?gedefqvekilkvrirn————grkeyflkwkgyse-edntwepeenl?————cpdlikefeerrarer
SpSWI6_A      74  143   eeeeedeyvvekvlkhrmarkg——ggyeyllkwegyddpsdntwsseadcs————gckqlieaywnehggrp
Pf0131C     (  78  200)   ...deefeigdileikkkkn———gfiylvkwkgysd-dentwepesnl...
CeT9A58       17   84   egksdeifevekilahkvtd————nllvlqvrwlgyga-dedtwepeedlq——ecasevvaeyykklkvtd
DmSuv3-9     212  278   krppkgeyvveriecvemdq————yqpvffvkwlgyhd-sentweslanva——dcaemekfverhqqlye
HuMG44      ( 250  448) skrnlydfeve?lcdykkir————eqeyylvkwrgypd-sestweprqnlk————cvrilkqfhkdlerel
CfTENV        81  143   epeaenefevekildkk——————gqrylvkwkgyde-sentweprinla——ncyqllrqfqkwrqdsr
FoSKPY      1229 1296   eisgpevyeaeairdtrkin————gqreylikwknype-nentweppkhlv——nagrllkdfhqrarkke
MoCHD1_A     263  362   qpedeefetiervmdcrvgrk<28>gdiqylikwkgwsh-ihntweteetlkqqnvrgmkkldnykkkdgetk
CeYK9A3     (   2  133)                          ...kwtgwsh-lhntwesenslalmnakglkkvqnyvkkqkeve
ScYEZ4_A     188  257   ktsleegkvlektvpdlnnck———enyeflikwtdesh-lhntwetyesig—qvrglkrldnyckqfiied
MoCHD1_B     380  450   ddlhkqyqiveriiahsnkqsaa—glpdyyckwqglpy-secswedgalis——kkfqtcideyfsrnqskt
ScYEZ4_B     278  350   ldefeefhvneriidsgrasledgtsglqvlvkwrrlpy-deatwepatdiv——klaneqvkhfqnrenski
```

ALIGN BY USE OF DYNAMIC PROGRAMMING

- Dynamic programming finds *best* alignment of *k* sequences with given scoring scheme

- For two sequences there are three different column types

- For three sequences there are seven different column types

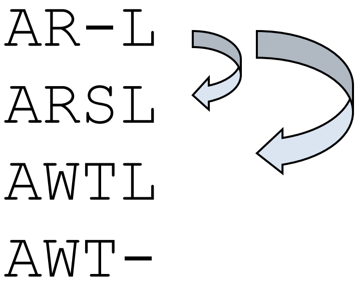
  x means an amino acid,   - a blank

```
Sequence1     x   -   x   x   -   -   x
Sequence2     x   x   -   x   -   x   -
Sequence3     x   x   x   -   x   -   x
```

- Time complexity of $O(n^k)$ (sequence lengths = n)

SCORING MULTIPLE SEQUENCE ALIGNMENTS

Alignment



```
AR-L
ARSL
AWTL
AWT-
```

- Sum of the pairwise sequence score

$$S(MSA) = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} S(s_i, s_j)$$

m: the number of sequences
s_i, s_j: sequence i, j
$S(s_i,s_j)$ = score of s_i,s_j

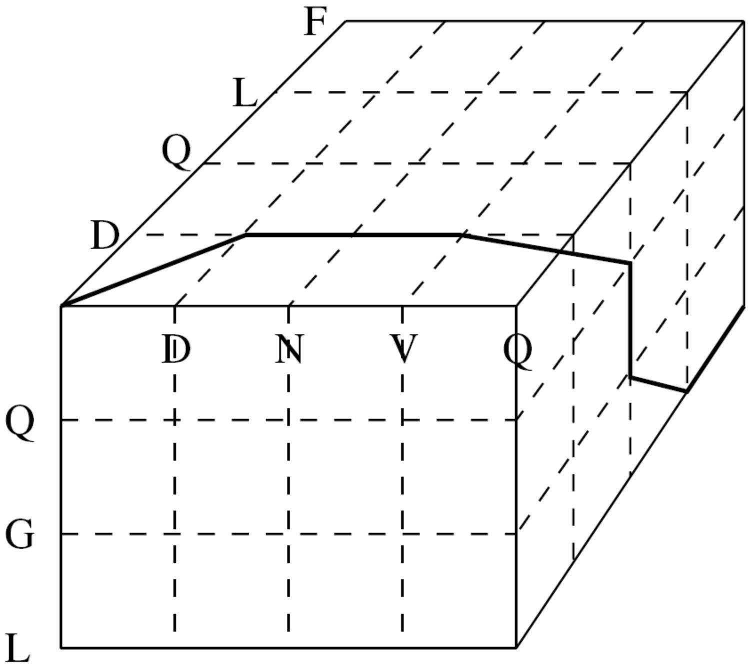
- Sum of scores for each row

$$S(MSA) = \sum_{k=1}^{r} \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} R_{S_k^i S_k^j}$$

r: number of columns

USE OF K-DIMENTIONAL DYNAMIC PROGRAMMING

- Dynamic programming finds *best* alignment of *k* sequences given a scoring scheme

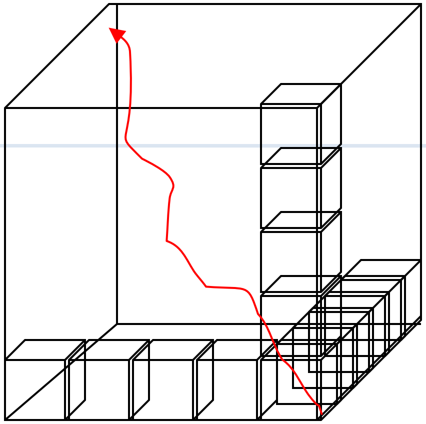


(a)

```
D--Q-LF
DNVQ---
---QGL-
```

(b)

MULTI-DIMENSIONAL DP



- 3 sequences:
  - Linear gap cost: γ(d) = -gd
  - Score of the whole MSA: $S(m) = \sum_i S(m_i)$

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

$-d-d+s(y_j,z_k)$
or
$-d+s(y_j,z_k)$ etc.

-d-d+"s(-,-)"
or
-d + 0 etc.

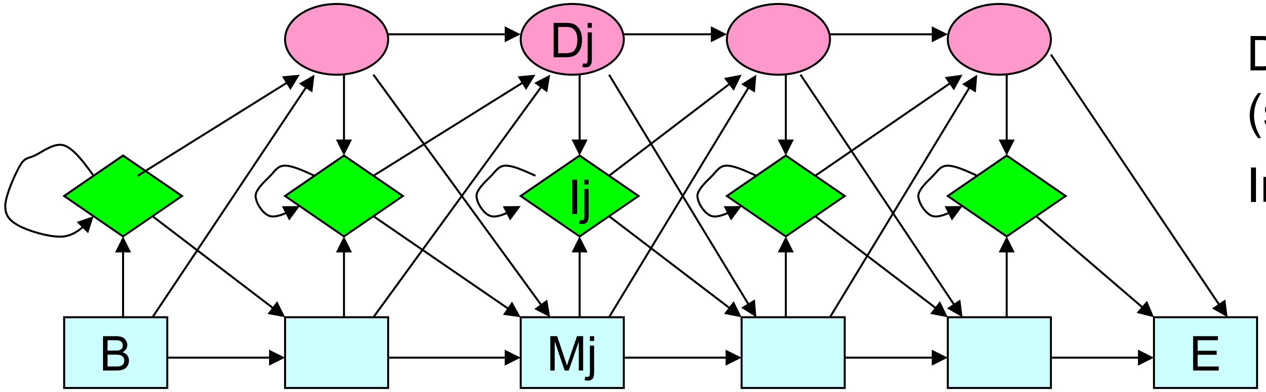
PROFILE HIDDEN MARKOV MODEL

REF: Biological sequence analysis: Probabilistic models of proteins and nucleic acids Richard Durbin et al.

Slides by SNU BioIntelligence Lab. (http://bi.snu.ac.kr)
Sildes by D. Kihara @ Purdue

PROFILE HMM

- An HMM which model a multiple sequence alignment of a protein family
- Concentrate on features that are conserved in the wh ole family (consensus modeling):
  - Improves alignment of distantly related sequence of the sa me family.
  - Able to characterize the family.

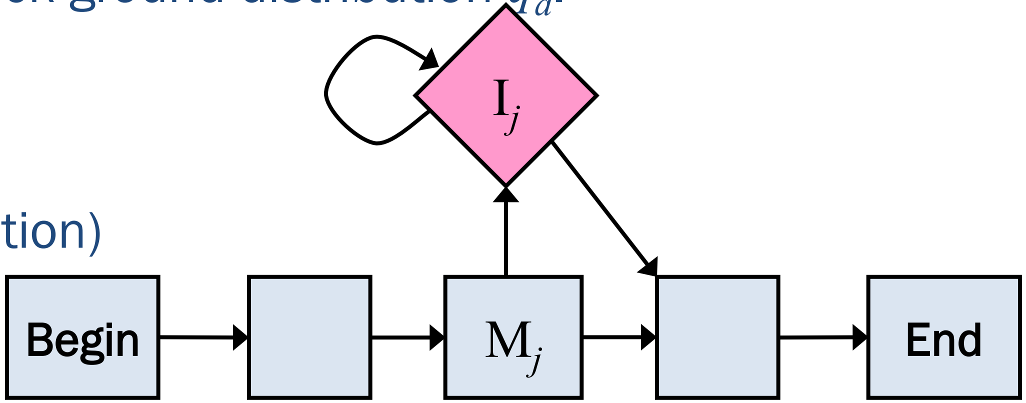


Deletion (silent states)

Insertion

ADD INSERTIONS

- Introduce insert states I_i
  - Emission prob. $e_{I_i}(a)$
    - Normally set to equal back ground distribution q_a.
  - Transition prob. For
    - M_i to I_i,
    - I_i to itself (multiple insertion)
    - I_i to M_{i+1}

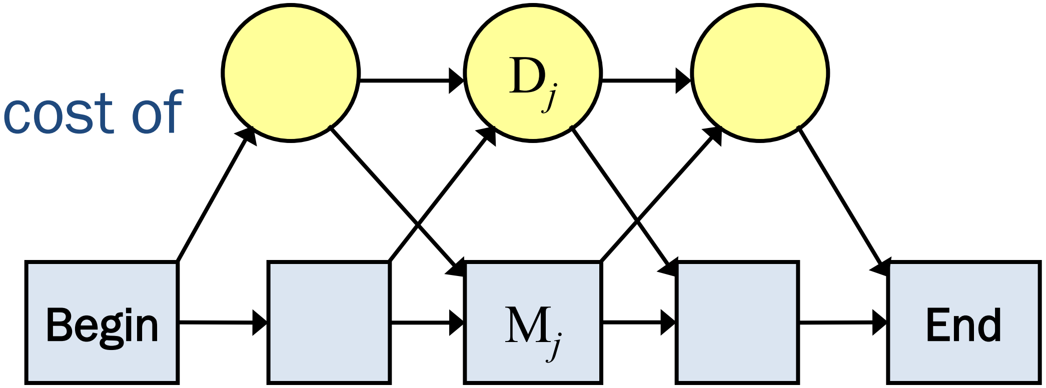


  - Log-odds score of a gap of length k
    - Assuming that $e_{I_i}(a) = q_{a\_}$ there is no logg-odds from emission

$$\log a_{M_j I_j} + \log a_{I_j M_{j+1}} + (k-1)\log a_{I_j I_j}$$

ADD DELETION

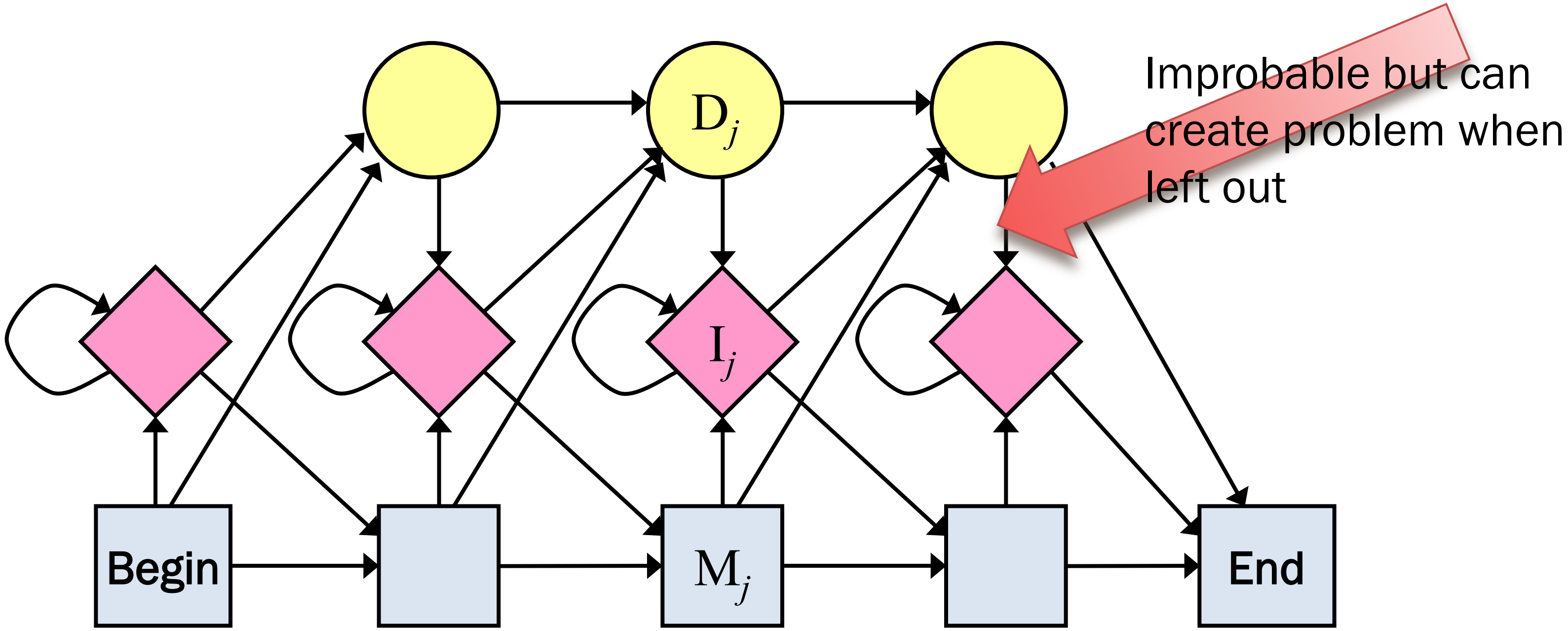
- Introduce delete states (silent state)
  - No emission prob.
  - Cost of a deletion sum cost of
    - M→D transition
    - D→D transitions
    - D→M transition



  - Each D→D might be different prob. Unlike I->I that have same prob.

COMPONENTS OF PROFILE HMMS (5)

- Combining all parts

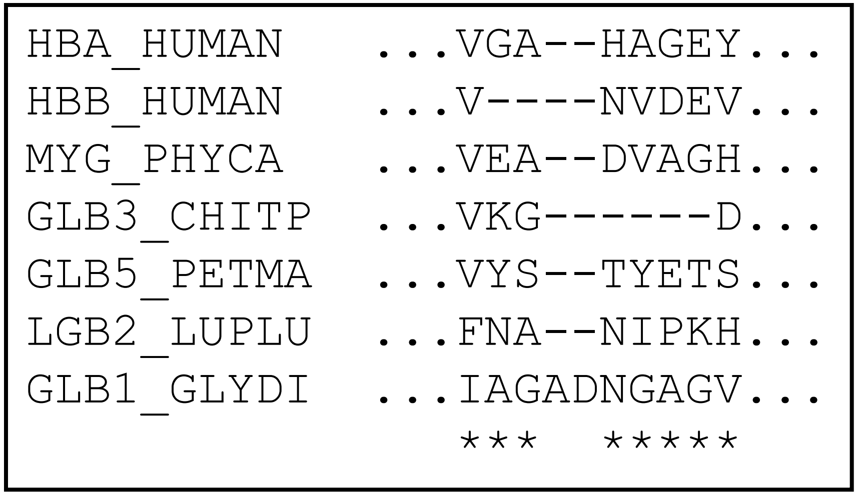


Improbable but can create problem when left out

Figure 5.2 *The transition structure of a profile HMM*.

DERIVING PROFILES HMM FROM MSA

- Assume correct multiple seq. alignment is given



```
HBA_HUMAN    ...VGA--HAGEY...
HBB_HUMAN    ...V----NVDEV...
MYG_PHYCA    ...VEA--DVAGH...
GLB3_CHITP   ...VKG------D...
GLB5_PETMA   ...VYS--TYETS...
LGB2_LUPLU   ...FNA--NIPKH...
GLB1_GLYDI   ...IAGADNGAGV...
                ***  *****
```

Figure 5.3 Ten columns from the multiple alignment of seven globin protein sequences shown in Figure 5.1 The starred columns are ones that will be treated as ‘matches’ in the profile HMM.

HMMS FROM MULTIPLE ALIGNMENTS

- Basic profile HMM parameterization
  - Aim: generate distribution peak around members of the family
- Parameters
  - Probabilities values: various ways to do it but let assume independent samples aligned independently to the HMM

$$a_{kl} = \frac{A_{kl}}{\sum_{l'} A_{kl'}} \qquad e_k(a) = \frac{E_k(a)}{\sum_{a'} E_k(a')}$$

  - Length of the model: heuristics or systematic way
    - Deciding which MSA columns to assign to match states and which to insert states.
    - One Heuristics: columns that are more than half gap should be modelled buy inserts.

SEARCHING WITH PROFILE HMMS (1)

- Main usage of profile HMMs
  - Detecting potential membership in a family
  - By (global) matching a sequence to the profile HMMs
  - Scoring a match:
    - Viterbi equations – gives h most probable alignment of a seq together with its probability
    - Forward equation – calculates the full probabilities of seq summed overall possible paths.
  - Either case, what we want is the log-odd ratio x being the family compared to the random model

$$P(x \mid R) = \prod_i q_{x_i}$$

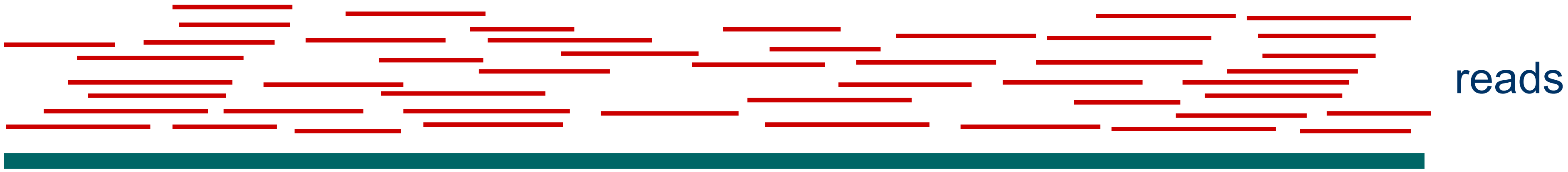
DNA Sequencing

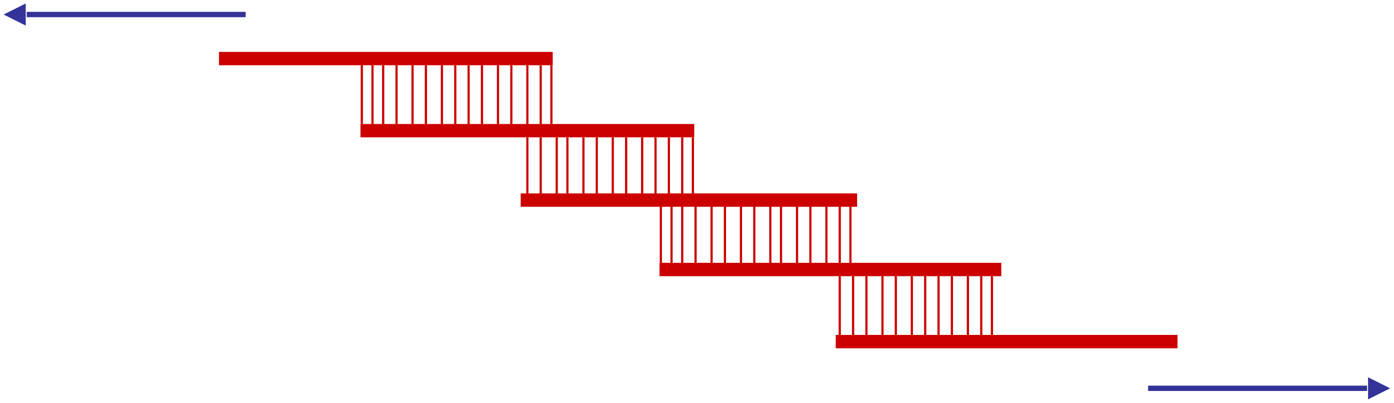
Two main assembly problems

- De Novo Assembly

- Resequencing

Reconstructing the Sequence (De Novo Assembly)

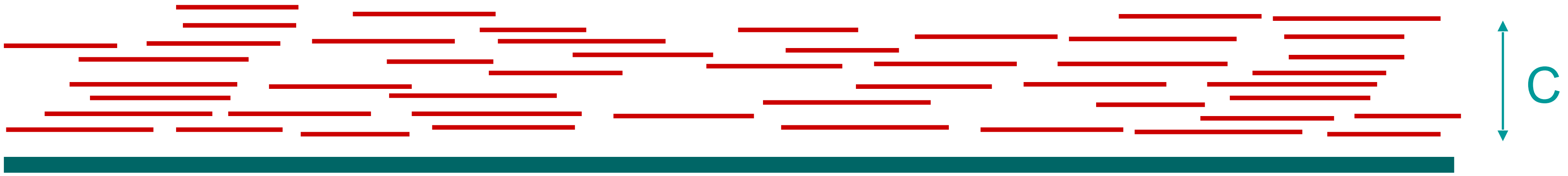




Cover region with high redundancy

Overlap & extend reads to reconstruct the original genomic region

Definition of Coverage



Length of genomic segment:	**G**
Number of reads:	**N**
Length of each read:	**L**

Definition: Coverage **C = N L / G**

How much coverage is enough?

Lander-Waterman model: **Prob[not covered bp] = e^{-C}**

Assuming uniform distribution of reads, C=10 results in 1 gapped region /1,000,000 nucleotides

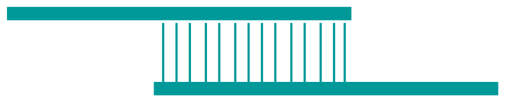
Fragment Assembly

(in whole-genome shotgun sequencing)

Steps to Assemble a Genome

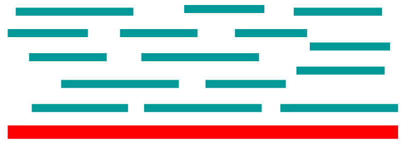
Some Terminology

read a 500-900 long word that comes out of sequencer

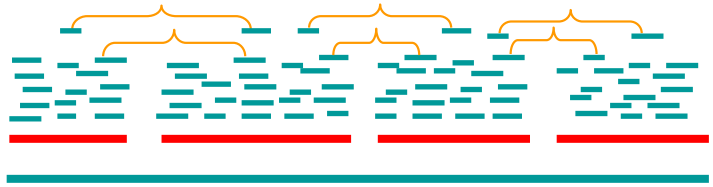


mate pair a pair of reads from two ends of the same insert fragment

contig a contiguous sequence formed by several overlapping reads with no gaps



supercontig (scaffold) an ordered and oriented set of contigs, usually by mate pairs



consensus sequene sequence derived from the multiple alignment of reads in a contig

..ACGATTACAATAGGTT..

1. Find Overlapping Reads

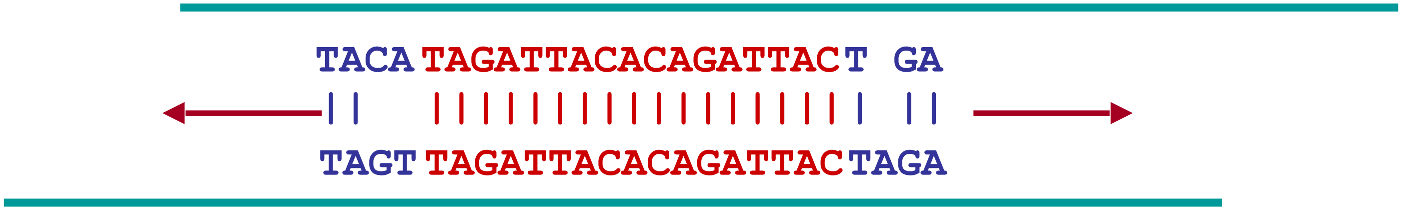
```
aaactgcagtacggatct
aaactgcag
 aactgcagt
…
        gtacggatct
         tacggatct
gggcccaaactgcagtac
gggcccaaa
 ggcccaaac
…
        actgcagta
         ctgcagtac
gtacggatctactacaca
gtacggatc
 tacggatct
…
        ctactacac
         tactacaca
```

```
(read, pos., word, orient.)
aaactgcag
aactgcagt
actgcagta
…
gtacggatc
tacggatct
gggcccaaa
ggcccaaac
gcccaaact
…
actgcagta
ctgcagtac
gtacggatc
tacggatct
acggatcta
…
ctactacac
tactacaca
```

```
(word, read, orient., pos.)
aaactgcag
aactgcagt
acggatcta
actgcagta
actgcagta
cccaaactg
cggatctac
ctactacac
ctgcagtac
ctgcagtac
gcccaaact
ggcccaaac
gggcccaaa
gtacggatc
gtacggatc
tacggatct
tacggatct
tactacaca
```

1. Find Overlapping Reads

- Find pairs of reads sharing a k-mer, k ~ 24
- Extend to full alignment – throw away if not >98% similar

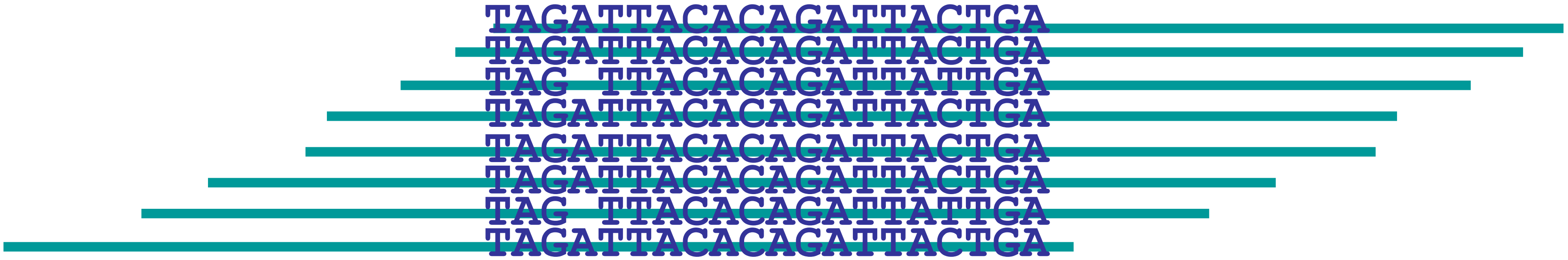


```
TACA TAGATTACACAGATTACT  GA
||   ||||||||||||||||||  ||
TAGT TAGATTACACAGATTAC TAGA
```

- Caveat: repeats
  - A k-mer that occurs N times, causes O(N^2) read/read comparisons
  - ALU k-mers could cause up to 1,000,000^2 comparisons
- Solution:
  - Discard all k-mers that occur “too often”
    - Set cutoff to balance sensitivity/speed tradeoff, according to genome at hand and computing resources available

1. Find Overlapping Reads

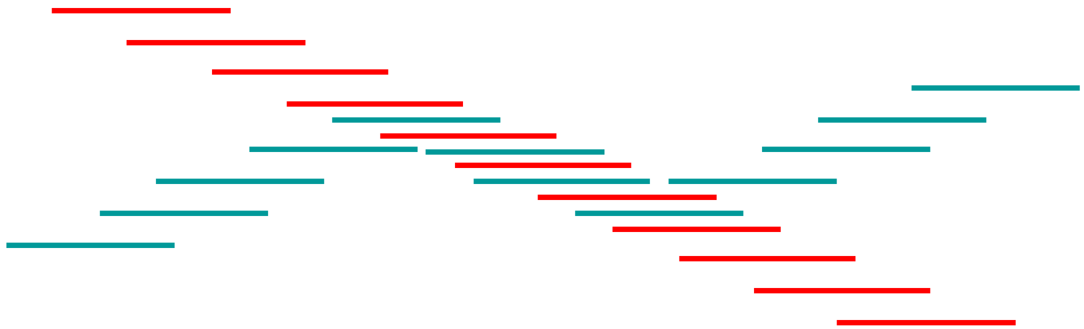
Create local multiple alignments from the overlapping reads



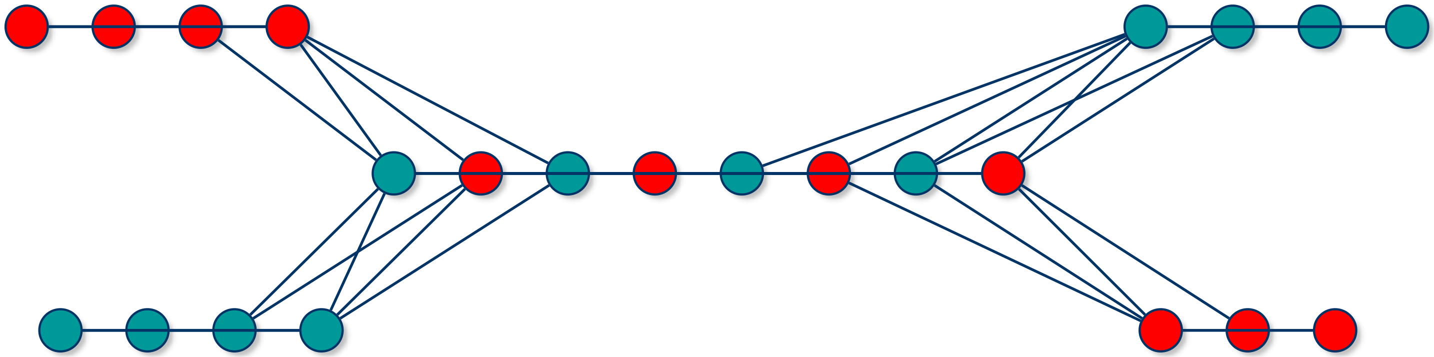
```
TAGATTACACAGATTACTGA
TAGATTACACAGATTACTGA
TAG TTACACAGATTATTGA
TAGATTACACAGATTACTGA
TAGATTACACAGATTACTGA
TAGATTACACAGATTACTGA
TAG TTACACAGATTATTGA
TAGATTACACAGATTACTGA
```

2. Merge Reads into Contigs

- Overlap graph:
  - Nodes: reads $r_1 \ldots\ldots r_n$
  - Edges: overlaps (r_i, r_j, shift, orientation, score)

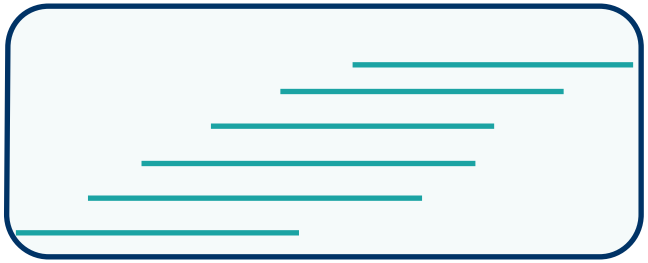


Reads that come from two regions of the genome (blue and red) that contain the same repeat

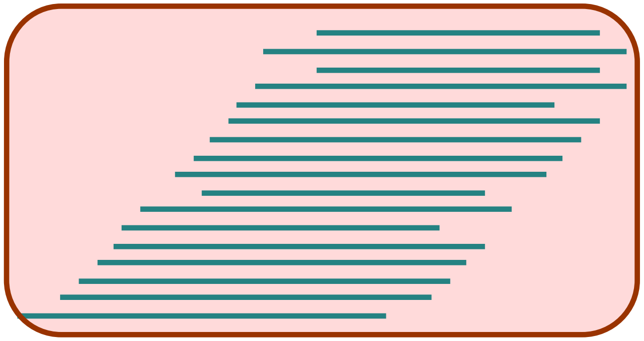


Note:
of course, we don't know the "color" of these nodes

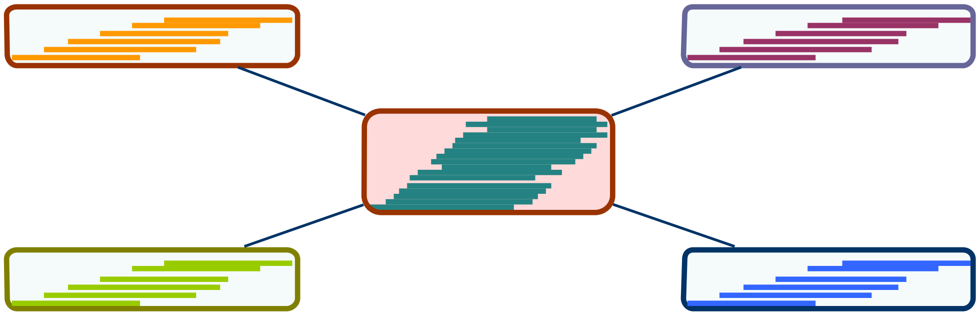
3. Link Contigs into Supercontigs



Normal density



Too dense
⇒ Overcollapsed

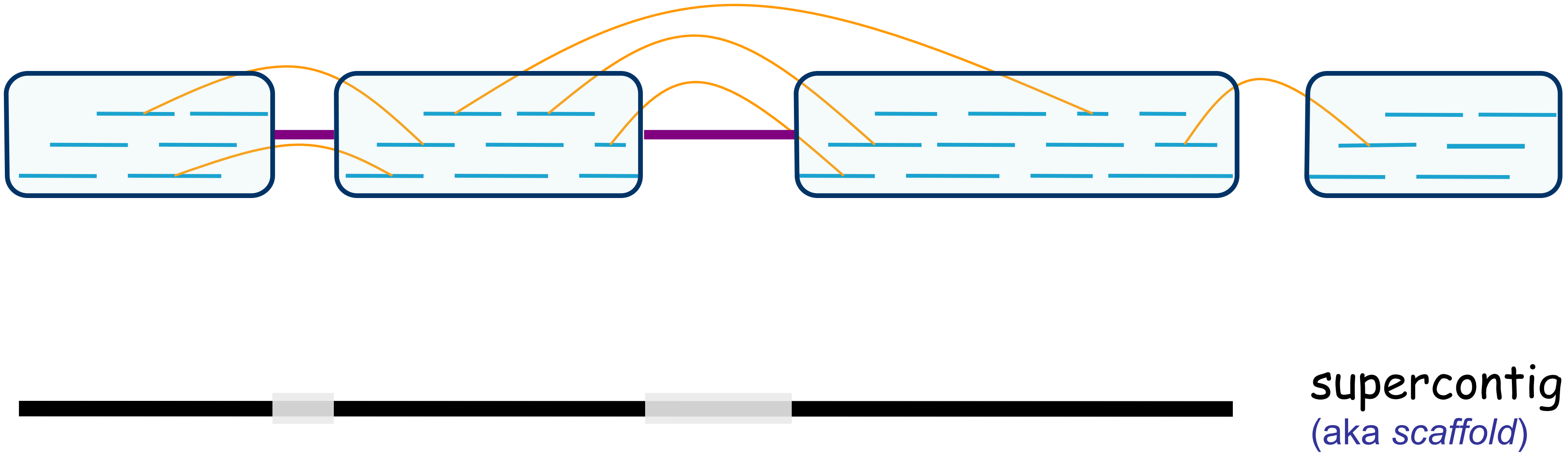


Inconsistent links
⇒ Overcollapsed?

3. Link Contigs into Supercontigs

Find all links between unique contigs

Connect contigs incrementally, if ≥ 2 forward-reverse links

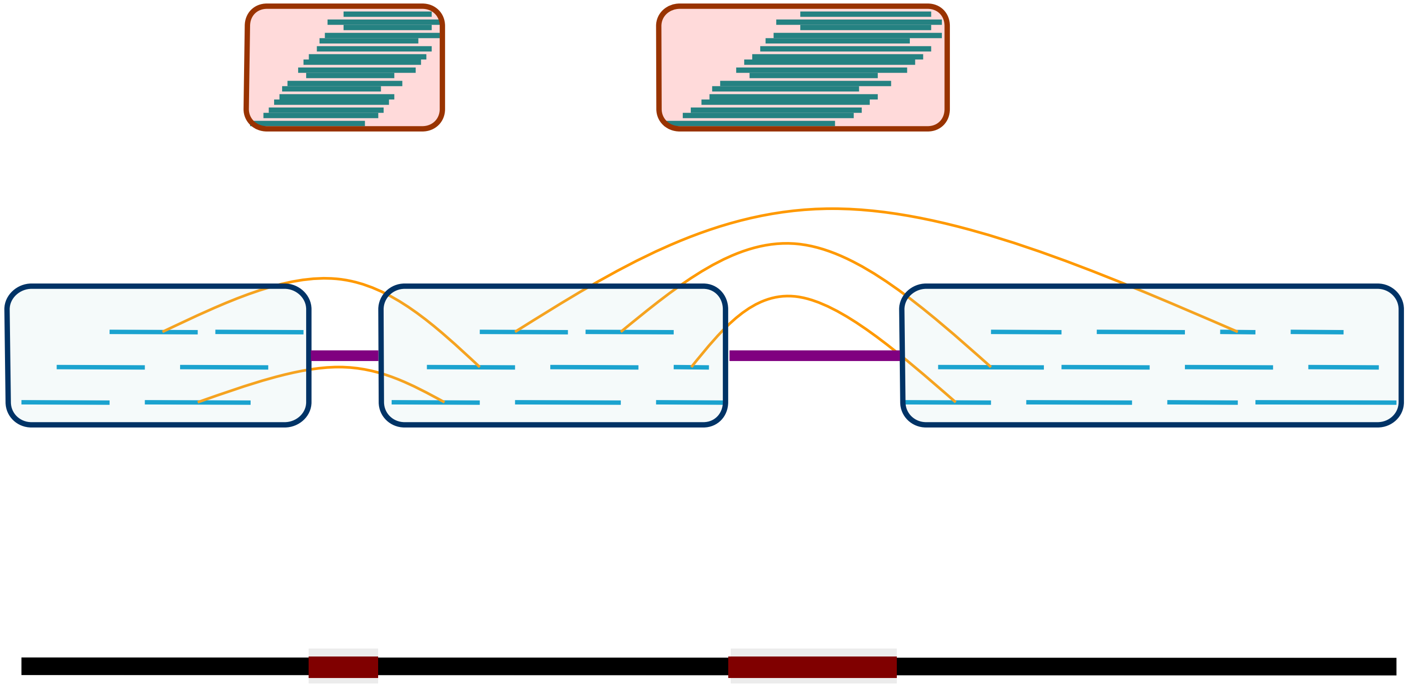


3. Link Contigs into Supercontigs

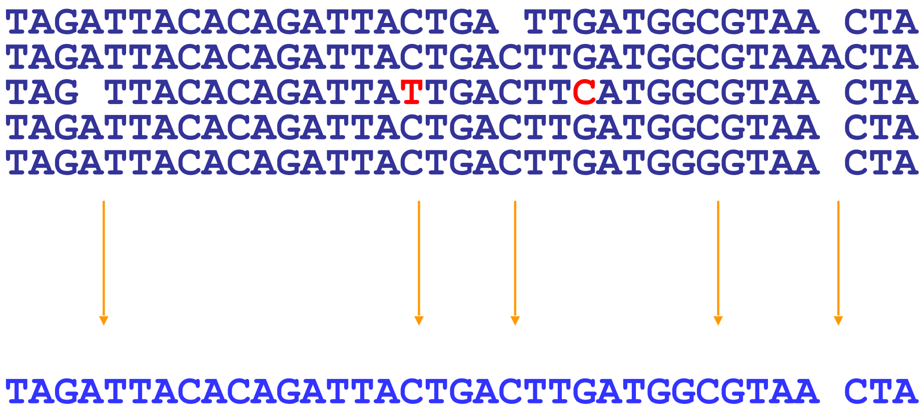
Fill gaps in supercontigs with paths of repeat contigs

Complex algorithmic step

- Exponential number of paths
- Forward-reverse links



4. Derive Consensus Sequence



```
TAGATTACACAGATTACTGA TTGATGGCGTAA CTA
TAGATTACACAGATTACTGACTTGATGGCGTAAACTA
TAG TTACACAGATTATTGACTTCATGGCGTAA CTA
TAGATTACACAGATTACTGACTTGATGGCGTAA CTA
TAGATTACACAGATTACTGACTTGATGGGGTAA CTA

TAGATTACACAGATTACTGACTTGATGGCGTAA CTA
```

Derive multiple alignment from pairwise read alignments

Derive each consensus base by weighted voting

(Alternative: take maximum-quality letter)

CSE 549

Sael Lee

WHOLE GENOME SEQ. ALIGNMENT

Slides Courtesy of Michael Schatz
Quantitative Biology Class @ CSHL

EXACT MATCHING

Slide extracts from Michael Schatz's Quantitative Biology Class @ CSHL
http://schatzlab.cshl.edu/teaching/2010

EXACT MATCHING OVERVIEW

Where is GATTACA in the human genome?

Brute Force (3 GB)

```
BANANA
BAN
 ANA
  NAN
   ANA
```

Naive

Slow & Easy

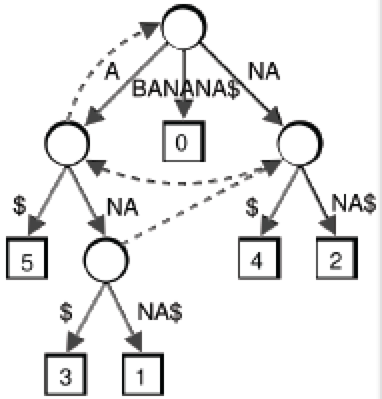
Suffix Array (>15 GB)

6	$
5	A$
3	ANA$
1	ANANA$
0	BANANA$
4	NA$
2	NANA$

Vmatch, PacBio Aligner

Binary Search

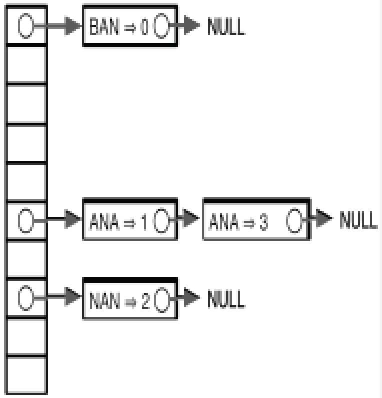
Suffix Tree (>51 GB)



MUMmer, MUMmerGPU

Tree Searching

Hash Table (>15 GB)



BLAST, MAQ, ZOOM, RMAP, CloudBurst

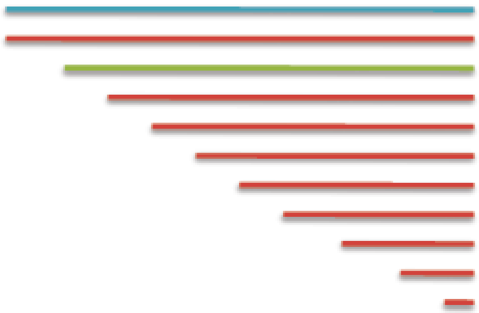
Seed-and-extend

BRUTE FORCE ANALYSIS

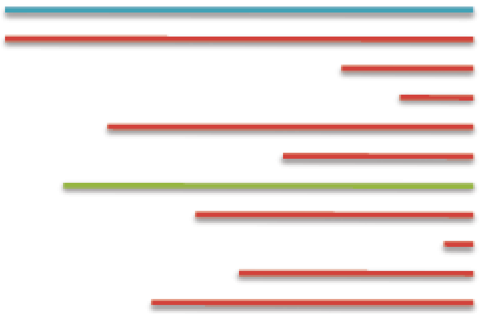
- Brute Force:
  - At every possible offset in the genome:
    - Do all of the characters of the query match?
- Analysis
  - Simple, easy to understand
  - Genome length = n
  - Query length = m
  - Comparisons: (n-m+1) * m
- Overall runtime: O(nm)
  - If we double genome or query size, takes twice as long
  - If we double both, takes 4 times as long

SUFFIX ARRAYS

- What if we need to check many queries?
  - Sorting alphabetically lets us immediately skip through the data *without any loss in accuracy*
- Sorting the genome: Suffix Array (Manber & Myers, 1991)
  - Sort every suffix of the genome



Split into n suffixes



Sort suffixes alphabetically

SEARCHING THE INDEX

- Strategy 2: Binary search
  - Compare to the middle, refine as higher or lower

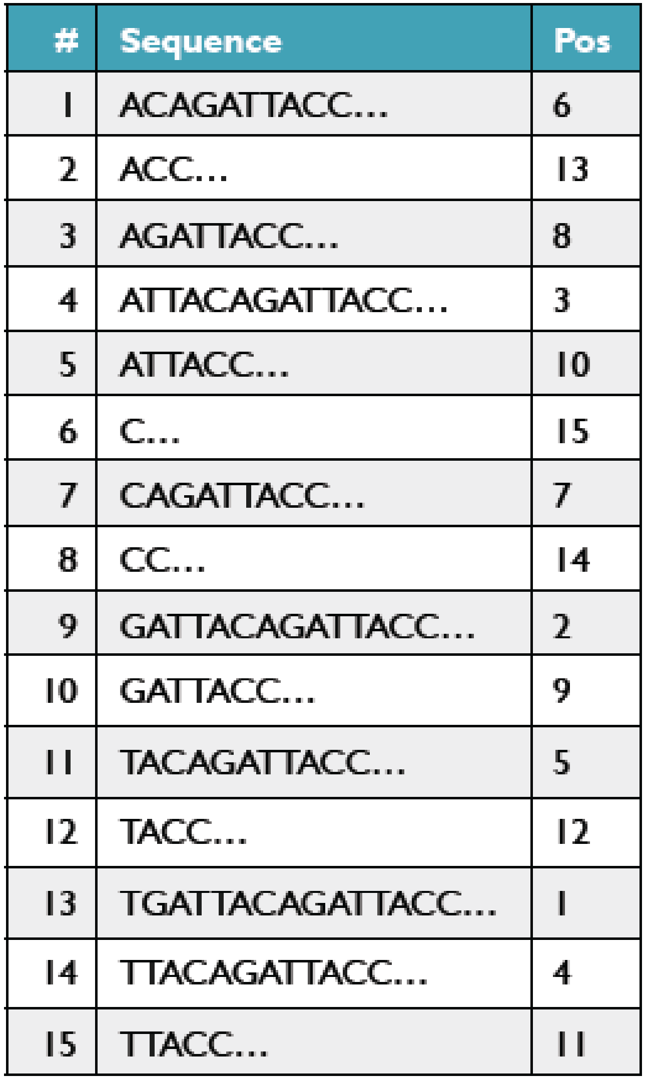
- Searching for GATTACA
  - Lo = 1; Hi = 15; Mid = (1+15)/2 = 8
  - Middle = Suffix[8] = CC
    => Higher: Lo = Mid + 1

  - Lo = 9; Hi = 15; Mid = (9+15)/2 = 12
  - Middle = Suffix[12] = TACC
    => Lower: Hi = Mid - 1

  - Lo = 9; Hi = 11; Mid = (9+11)/2 = 10
  - Middle = Suffix[10] = GATTACC
    => Lower: Hi = Mid - 1

  - Lo = 9; Hi = 9; Mid = (9+9)/2 = 9
  - Middle = Suffix[9] = GATTACA…
    => Match at position 2!

Lo →

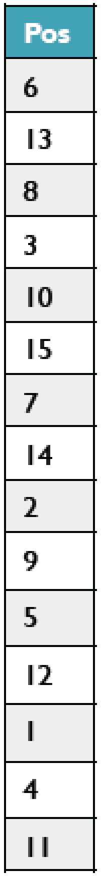


#	Sequence	Pos
1	ACAGATTACC…	6
2	ACC…	13
3	AGATTACC…	8
4	ATTACAGATTACC…	3
5	ATTACC…	10
6	C…	15
7	CAGATTACC…	7
8	CC…	14
9	GATTACAGATTACC…	2
10	GATTACC…	9
11	TACAGATTACC…	5
12	TACC…	12
13	TGATTACAGATTACC…	1
14	TTACAGATTACC…	4
15	TTACC…	11

Hi →

SUFFIX ARRAY CONSTRUCTION

- Searching the array is very fast, but it takes time to construct
  - This time will be amortized over many, many searches
  - Run it once "overnight" and save it away for all future queries
- How do we store the suffix array?
  - Explicitly storing all n strings is not feasible $O(n^2)$
- Instead use implicit representation
  - Keep 1 copy of the genome, and a list of sorted offsets
  - Storing 3 billion offsets requires a big server (12GB)
    - Build a separate index for each chromosome

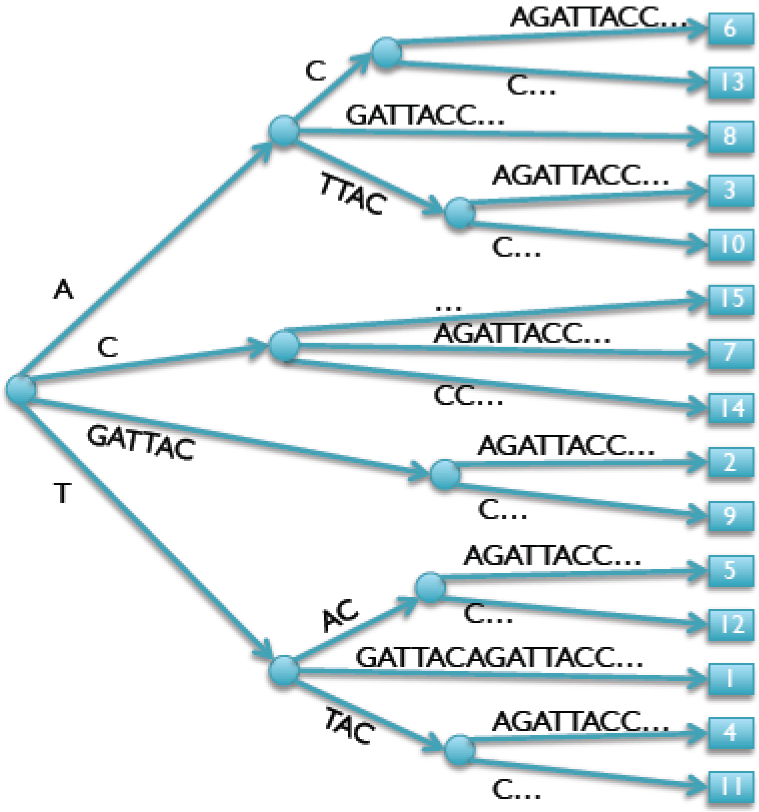


Pos
6
13
8
3
10
15
7
14
2
9
5
12
1
4
11

TGATTACAGATTACC

SUFFIX TREES

#	Sequence	Pos
1	ACAGATTACC…	6
2	ACC…	13
3	AGATTACC…	8
4	ATTACAGATTACC…	3
5	ATTACC…	10
6	C…	15
7	CAGATTACC…	7
8	CC…	14
9	GATTACAGATTACC…	2
10	GATTACC…	9
11	TACAGATTACC…	5
12	TACC…	12
13	TGATTACAGATTACC…	1
14	TTACAGATTACC…	4
15	TTACC…	11



Suffix Tree = Tree of suffixes (indexes **all** substrings of a sequence)

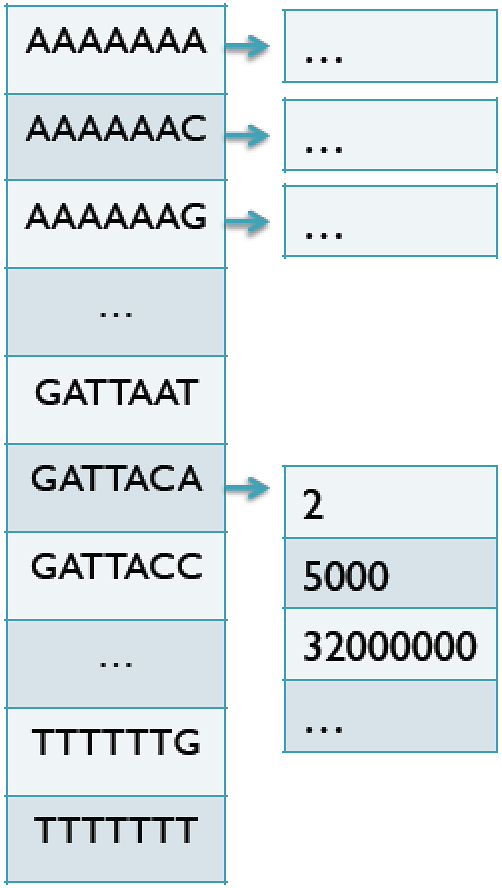
- 1 Leaf ($) for each suffix, path-label to leaf spells the suffix
- Nodes have at least 2 and at most 5 children (A,C,G,T,$)

SUFFIX TREE PROPERTIES & APPLICATIONS

- Properties
  - Number of Nodes/Edges: O(n)
  - Tree Size: O(n)
  - Max Depth: O(n)
  - Construction Time: O(n)
    - Uses suffix links to jump between nodes without rechecking
    - Tricky to implement, prove efficiency
- Applications
  - Sorting all suffixes: O(n)
  - Check for query: O(m)
  - Find all z occurrences of a query O(m + z)
  - Find maximal exact matches O(m)
  - Longest common substring O(m)
- Used for many string algorithms in linear time
  - Many can be implemented on suffix arrays using a little extra work

HASHING

- Where is GATTACA in the human genome?
  - Build an inverted index (table) of every k-mer in the genome
- How do we access the table?
  - We can only use numbers to index
    - table[GATTACA] <- error, does not compute
  - Encode sequences as numbers
    - Easy: A = 110, C = 210, G = 310, T = 410
      - GATTACA = 314412110
    - Smart: A = 002, C = 012, G = 102, T = 112
      - GATTACA = 100011110001002 = 915610
  - Running time
    - Construction: O(n)
    - Lookup: O(1) + O(z)
    - Sorts the genome mers in linear time



IN-EXACT ALIGNMENT

Slide extracts from Michael Schatz's Quantitative Biology Class @ CSHL
http://schatzlab.cshl.edu/teaching/2010

IN-EXACT ALIGNMENT

- Where is GATTACA *approximately* in the human genome?
  - And how do we efficiently find them?
- It depends…
  - Define 'approximately'
    - Hamming Distance, Edit distance, or Sequence Similarity
    - Ungapped vs Gapped vs Affine Gaps
    - Global vs Local
    - All positions or the single 'best'?
- Efficiency depends on the data characteristics & goals
  - Smith-Waterman: Exhaustive search for optimal alignments
  - BLAST: Hash based homology searches
  - MUMmer: Suffix Tree based whole genome alignment
  - Bowtie: BWT alignment for short read mapping

SEED-AND-EXTEND ALIGNMENT

- Theorem: An alignment of a sequence of length *m* with at most *k* differences ***must*** contain an exact match at least *s=m/(k+1)* bp long (*Baeza-Yates* and Perleberg, 1996)
  - Proof: Pigeon hole principle

- Search Algorithm
  - Use an index to rapidly find short exact alignments to seed longer in-exact alignments
    - RMAP, CloudBurst, …
  - Specificity of the seed depends on length
  - Length s seeds can also seed some lower quality alignments
    - Won't have perfect sensitivity, but avoids very short seeds

HAMMING DISTANCE LIMITATIONS

- Hamming distance measures the number of substitutions (SNPs)
  - Appropriate if that’s all we expect/want to find
    - Illumina sequencing error model
    - Other highly constrained sequences
- What about insertions and deletions?
  - At best the **indel** will only slightly lower the score
  - At worst highly similar sequences will fail to align

```
ACGTCTAG
||*****^
ACTCTAG-
```

Hamming distance=5 : 2 matches, 5 mismatches, 1 not aligned

```
ACGTCTAG
||^|||||
AC-TCTAG
```

Edit Distance = 1 : 7 matches, 0 mismatches, 1 not aligned

EDIT DISTANCE EXAMPLE

TGCATAT → ATCCGAT in 4 steps

TGCATAT → (insert A at front)

ATGCATAT → (delete 6th T)

ATGCATA → (substitute G for 5th A)

ATGCGTA → (substitute C for 3rd G)

ATCCGAT (Done)

Can it be done in 3 steps???

bioalgorithms.info