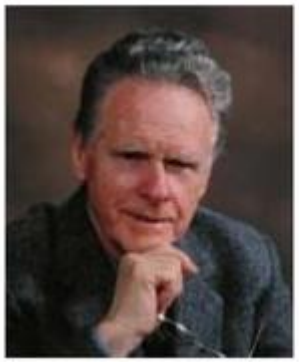
生物统计学：
生物信息中的概率统计模型

2020年秋



Dempster–Laird–Rubin,
Statistics, Havard Univ.

有关信息

- 授课教师：宁康
  - Email: ningkang@hust.edu.cn
  - Office: 华中科技大学东十一楼504室
  - Phone: 87793041, 18627968927

- 课程网页
  - http://www.microbioinformatics.org/teach/#
  - QQ群：182996651

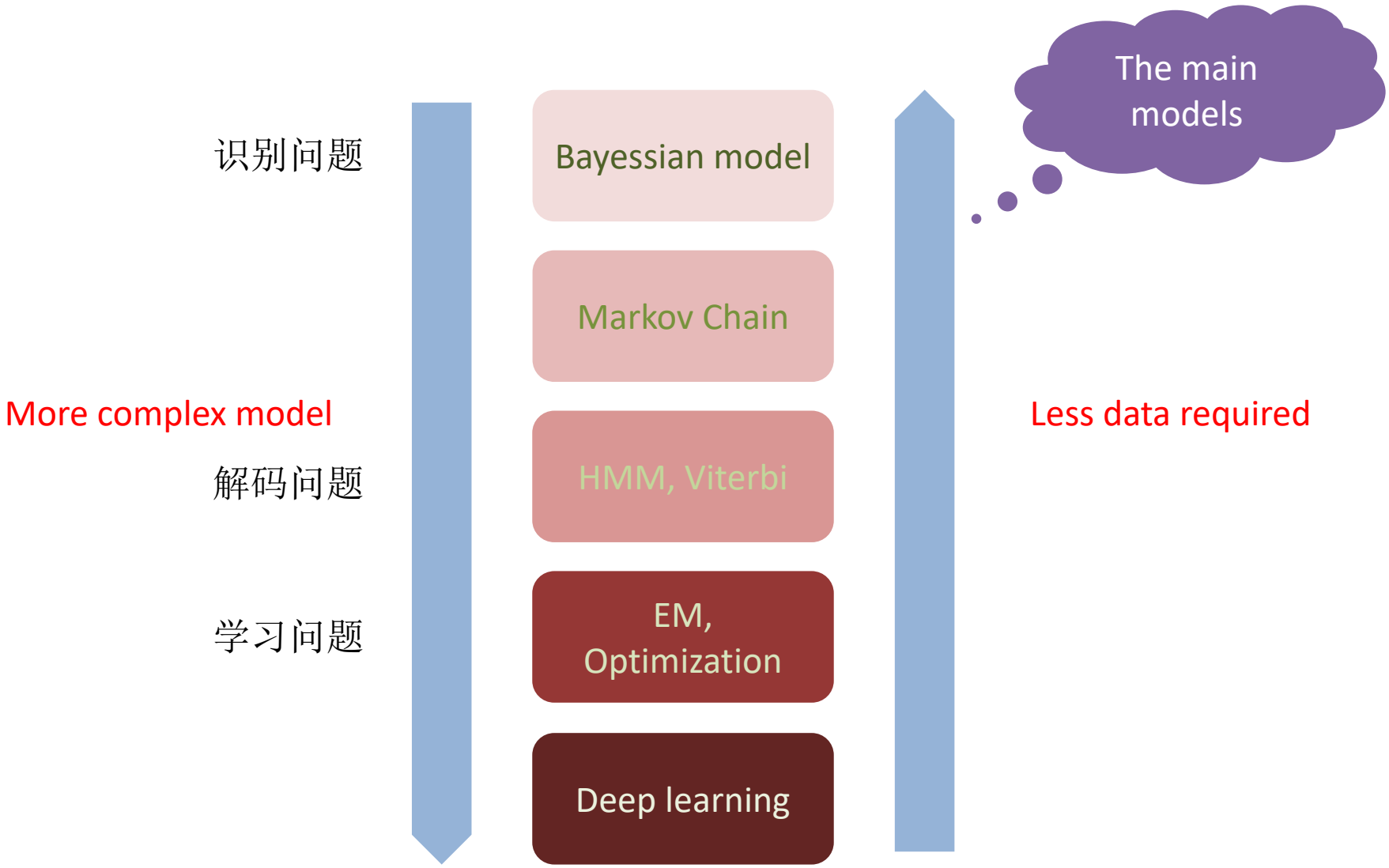
课程安排

- 生物背景和课程简介
- 传统生物统计学及其应用
- 生物统计学和生物大数据挖掘
  - Hidden Markov Model (HMM)及其应用
    - Markov Chain
    - HMM理论
    - HMM和基因识别 (Topic I)
    - HMM和序列比对 (Topic II)
  - 进化树的概率模型 (Topic III)
  - Motif finding中的概率模型 (Topic IV)
    - EM algorithm
    - Markov Chain Monte Carlo (MCMC)
  - 基因表达数据分析 (Topic V)
    - 聚类分析-Mixture model
    - Classification-Lasso Based variable selection
  - 基因网络推断 (Topic VI)
    - Bayesian网络
    - Gaussian Graphical Model
  - 基因网络分析 (Topic VII)
    - Network clustering
    - Network Motif
    - Markov random field (MRF)
  - Dimension reduction及其应用 (Topic VIII)
- 面向生物大数据挖掘的深度学习

研究对象:
生物序列,
进化树,
生物网络,
基因表达
...

方法:
生物计算与生物统计

Statistical modeling



第5-1章 EM算法

- Maximum likelihood estimation (MLE)
- EM算法
- EM for Multinomial distribution

What is MLE?

- Given
  - A sample $X=\{X_1, \ldots, X_n\}$
  - A vector of parameters θ
- We define
  - Likelihood of the data: $L(\theta)=P(X \mid \theta)$
  - Log-likelihood of the data: $l(\theta)=\log P(X|\theta)$
- Given X, find

$$\theta_{ML} = \arg\max_{\theta \in \Omega} l(\Theta)$$

MLE (cont)

- Often we assume that X_is are independently identically distributed (i.i.d.)

$$\begin{aligned}\theta_{ML} &= \arg\max_{\theta\in\Omega} l(\Theta) \\ &= \arg\max_{\theta\in\Omega} \log P(X\,|\,\Theta) \\ &= \arg\max_{\theta\in\Omega} \log \prod_i P(X_i\,|\,\Theta) \\ &= \arg\max_{\theta\in\Omega} \sum_i \log P(X_i\,|\,\Theta)\end{aligned}$$

- Depending on the form of p(x|θ), solving optimization problem can be easy or hard.

An Easy Case

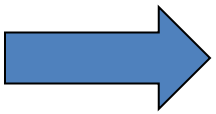
- Assuming
  - A coin has a probability p of being heads, 1-p of being tails.
  - Observation: We toss a coin N times, and the result is a set of Hs and Ts, and there are m Hs.

- What is the value of p based on MLE, given the observation?

An Easy Case (cont)

$$l(\Theta) = \log P(X \mid \Theta) = \log p^m (1-p)^{N-m}$$
$$= m \log p + (N-m)\log(1-p)$$

$$\frac{dl(\Theta)}{dp} = \frac{d(m \log p + (N-m)\log(1-p))}{dp} = \frac{m}{p} - \frac{N-m}{1-p} = 0$$



$$\hat{p} = \frac{m}{N}$$

以频率来估计概率

Basic Setting in EM

- X is a set of data points: **observed** data
- Θ is a parameter vector.
- EM is a method to find θ_{ML} where

$$\theta_{ML} = \arg\max_{\theta \in \Omega} l(\Theta)$$
$$= \arg\max_{\theta \in \Omega} \log P(X \mid \Theta)$$

- Calculating P(X | θ) directly is hard.
- Calculating P(X, Z|θ) is much simpler, where Z is “hidden” data (or “missing” data).

The Basic Setting in EM

- Y = (X, Z)
  - Y: complete data ("augmented data")
  - X: observed data ("incomplete" data)
  - Z: hidden data ("missing" data)

- Given a fixed x, there could be many possible z's.
  - Ex: given a sentence x, there could be many state sequences in an HMM that generates x.

The Iterative Approach for MLE

- When missing data is unavailable, it's hard to find the MLE directly

$$\theta_{ML} = \underset{\theta}{\text{Argmax}} \log \left(\sum_{Z} P(X, Z|\theta) \right)$$

- An alternative is to find a sequence

$$\theta^{(0)}, \theta^{(1)}, \cdots, \theta^{(t)}, \cdots,$$

$$\text{s.t.} \quad l(\theta^{(0)}) < l(\theta^{(1)}) < \cdots < l(\theta^{(t)}) < \cdots$$

$$
\begin{aligned}
l(\theta) - l(\theta^{(t)}) &= \log P(X|\theta) - \log P(X|\theta^{(t)}) \\
&= \log \left(\frac{\sum_Z P(X, Z|\theta)}{\sum_Z P(X, Z|\theta^{(t)})} \right) \\
&= \log \left(\sum_Z \frac{P(X, Z|\theta)}{\sum_{Z'} P(X, Z'|\theta^{(t)})} \right) \\
&= \log \left(\sum_Z \frac{P(X, Z|\theta)}{\sum_{Z'} P(X, Z'|\theta^{(t)})} \times \frac{P(X, Z|\theta^{(t)})}{P(X, Z|\theta^{(t)})} \right) \\
&= \log \left(\sum_Z \frac{P(X, Z|\theta^{(t)})}{\sum_{Z'} P(X, Z'|\theta^{(t)})} \times \frac{P(X, Z|\theta)}{P(X, Z|\theta^{(t)})} \right)
\end{aligned}
$$

$$
\begin{aligned}
l(\theta) - l(\theta^{(t)}) &= \log\left(\sum_Z \frac{P(X,Z|\theta^{(t)})}{\sum_{Z'} P(X,Z'|\theta^{(t)})} \times \frac{P(X,Z|\theta)}{P(X,Z|\theta^{(t)})}\right) \\
&= \log\left(\sum_Z P(Z|X,\theta^{(t)}) \times \frac{P(X,Z|\theta)}{P(X,Z|\theta^{(t)})}\right) \\
&\geq \sum_Z P(Z|X,\theta^{(t)}) \times \log\left(\frac{P(X,Z|\theta)}{P(X,Z|\theta^{(t)})}\right) \\
&= E_{P(Z|X,\theta^{(t)})}\left[\log\left(\frac{P(X,Z|\theta)}{P(X,Z|\theta^{(t)})}\right)\right] \\
&= E_{P(Z|X,\theta^{(t)})}\left[\log P(X,Z|\theta)\right] \\
&\quad - E_{P(Z|X,\theta^{(t)})}\left[\log P(X,Z|\theta^{(t)})\right]
\end{aligned}
$$

Jensen's inequality

Maximizing the Lower Bound

- The Jensen's inequality gives a lower bound to maximize,

$$\theta^{(t+1)} = \underset{\theta}{\text{Argmax}}\, E_{P(Z|X,\theta^{(t)})}\left[\log P(X, Z|\theta)\right]$$

- Q-function

$$Q(\theta|\theta^{(t)}) = E_{P(Z|X,\theta^{(t)})}\left[\log P(X, Z|\theta)\right]$$

Increasing the Likelihood

- Increasing the likelihood by maximizing the lower bound

$$l(\theta) - l(\theta^{(t)}) \geq Q(\theta|\theta^{(t)}) - Q(\theta^{(t)}|\theta^{(t)})$$

$$Q(\theta^{(t+1)}|\theta^{(t)}) > Q(\theta^{(t)}|\theta^{(t)}) \Rightarrow l(\theta^{(t+1)}) > l(\theta^{(t)})$$

- Which means that a better estimation of the parameter.

Summary: EM Algorithm

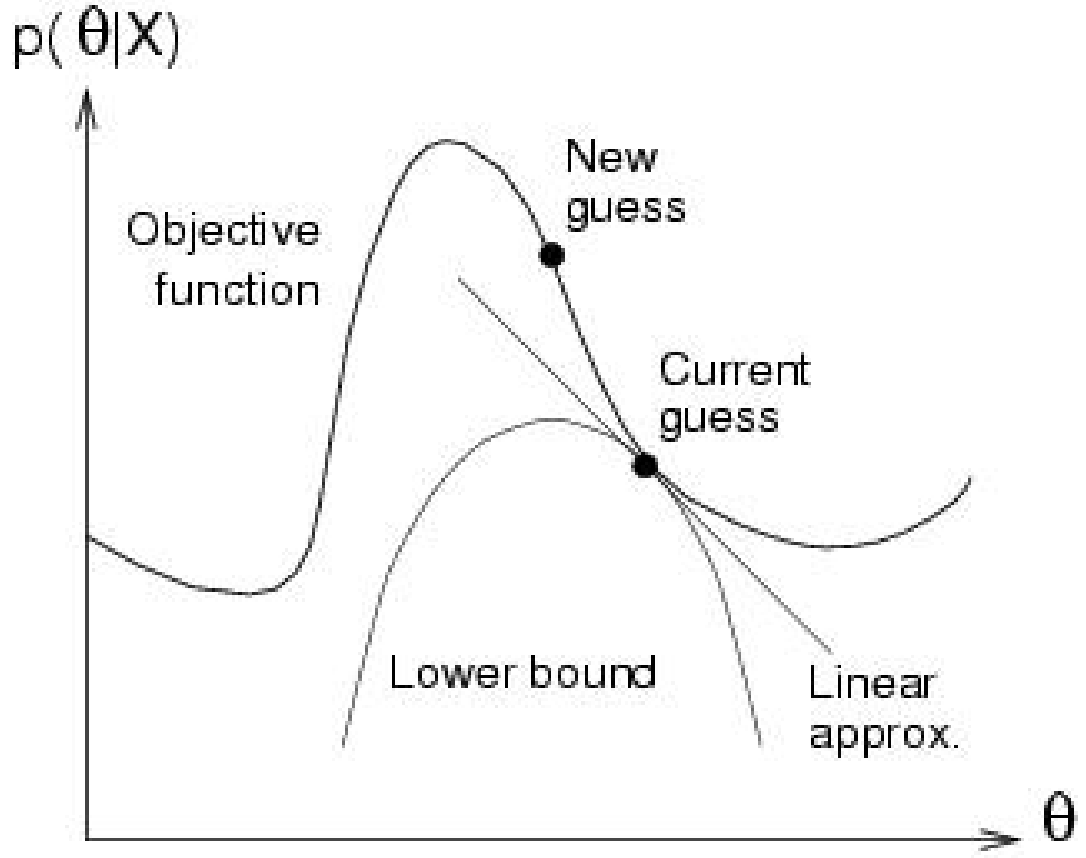
- Define a auxiliary function

$$Q(\theta|\theta') = \sum_Z P(Z|X,\theta') \log P(X,Z|\theta)$$
$$= E_{P(Z|X,\theta')} [\log P(X,Z|\theta)]$$

- EM algorithm iterates with two step
  - E-Step, compute $Q(\theta|\theta^{(t)})$
  - M-Step:

$$\theta^{(t+1)} = \underset{\theta}{\text{Argmax}}\, Q(\theta|\theta^{(t)})$$

Illustration of EM Algorithm



MLE算法 vs. EM算法（Examples）

- **MLE算法**

某位同学与一位猎人一起外出打猎，一只野兔从前方窜过。只听一声枪响，野兔应声到下，如果要你推测，这一发命中的子弹是谁打的？你就会想，只发一枪便打中，由于猎人命中的概率一般大于这位同学命中的概率，看来这一枪是猎人射中的。

- **EM算法**

小时候，老妈给一大袋糖果给你，叫你和你姐姐等分，然后你懒得去点糖果的个数，所以你也就不知道每个人到底该分多少个。咱们一般怎么做呢？先把一袋糖果目测的分为两袋，然后把两袋糖果拿在左右手，看哪个重，如果右手重，那很明显右手这代糖果多了，然后你再在右手这袋糖果中抓一把放到左手这袋，然后再感受下哪个重，然后再从重的那袋抓一小把放进轻的那一袋，继续下去，直到你感觉两袋糖果差不多相等了为止。

Reference: https://blog.csdn.net/zouxy09/article/details/8537620

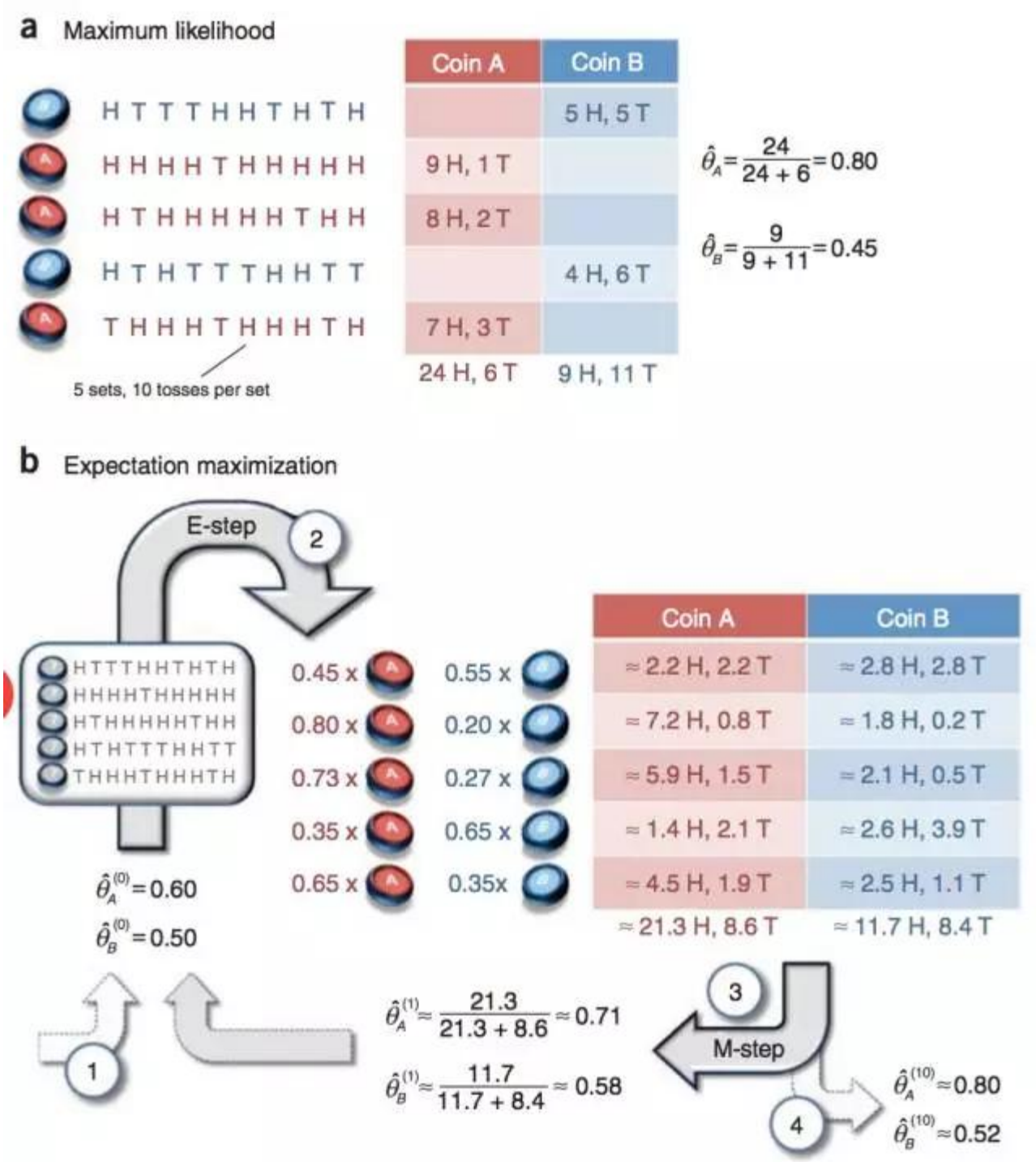
Example

抛3个硬币，抛C0硬币决定C1和C2，然后抛C1或者C2决定正反面，然后估算3个硬币的正反面概率值。

X: 观测到的正面
Z: 选取的哪一个硬币

每一次的情况：知道/不知道

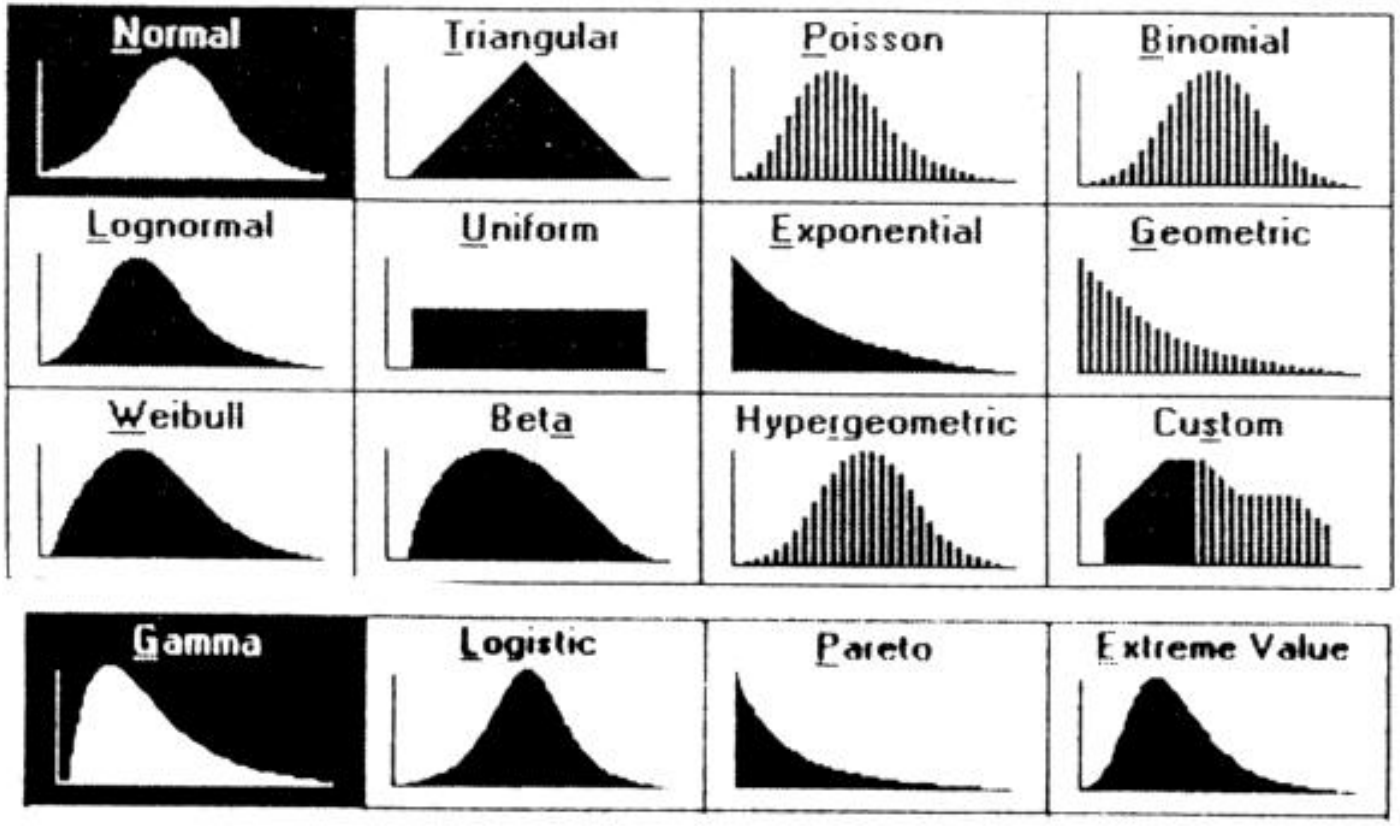
Binomial distribution assumption



Chuong B Do & Serafim Batzoglou“What is the expectation maximization algorithm?”. Nature Biotech., 2008

Think about...

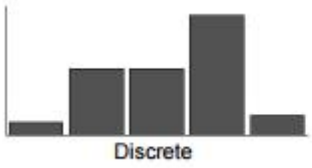
- Which distribution to be selected?

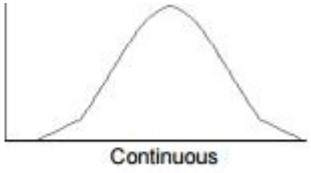


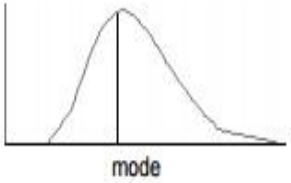
Think about...

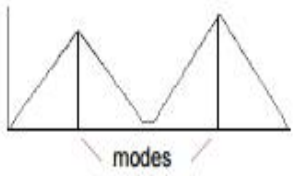
Which distribution to be selected?

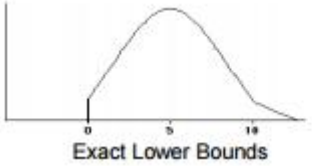
- Is the quantity discrete or continuous?
- Does the quantity have bounds?
- How many modes does it have?
- Is it symmetric or skewed?

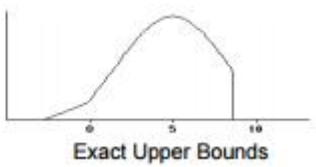


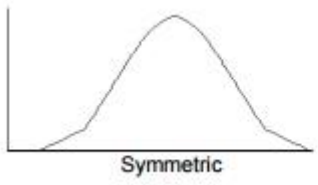


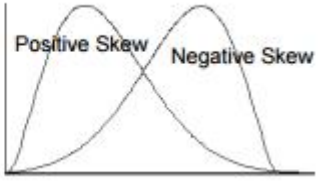












Think about...

- Which distribution to be selected?

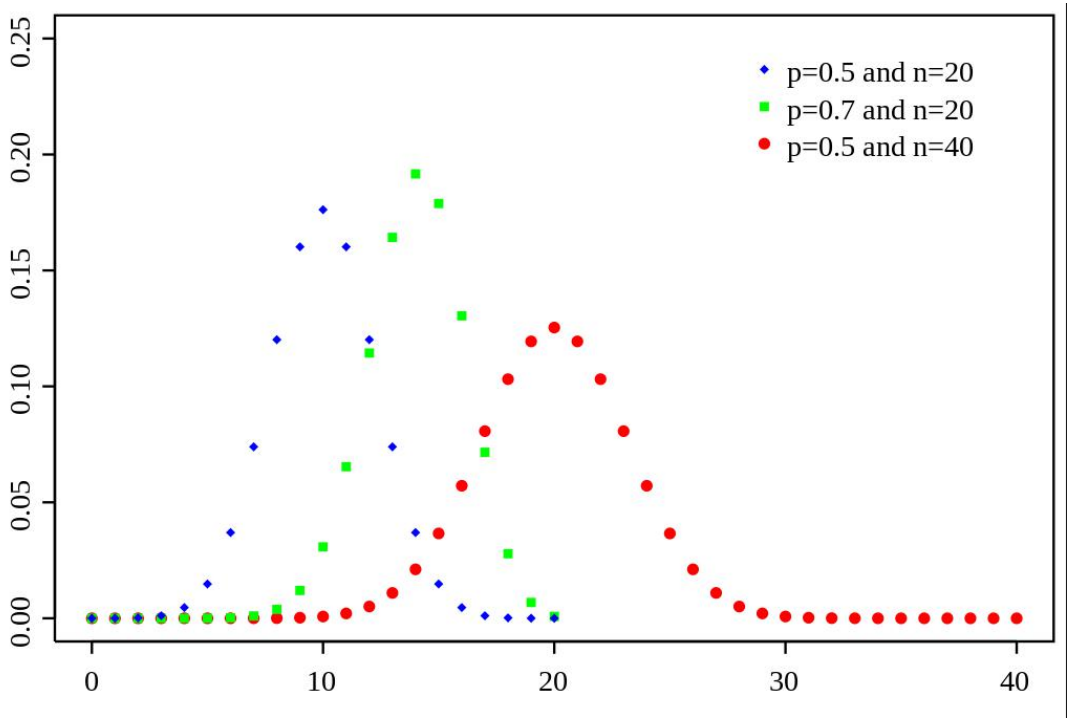
- When tossed, it can land in one of two positions: *Head* or *Tail*
- We denote by θ the (unknown) probability *P(H)*.

Estimation task:

- Given a sequence of toss samples *x[1], x[2], …, x[M]* we want to estimate the probabilities *P(H)*= θ and *P(T)* = *1* - θ

Think about...

- Which distribution to be selected?



Reference:

https://www.vosesoftware.com/riskwiki/Selectingtheappropriatedistributionsforyourmodel.php

http://wiki.analytica.com/Choosing_an_appropriate_distribution

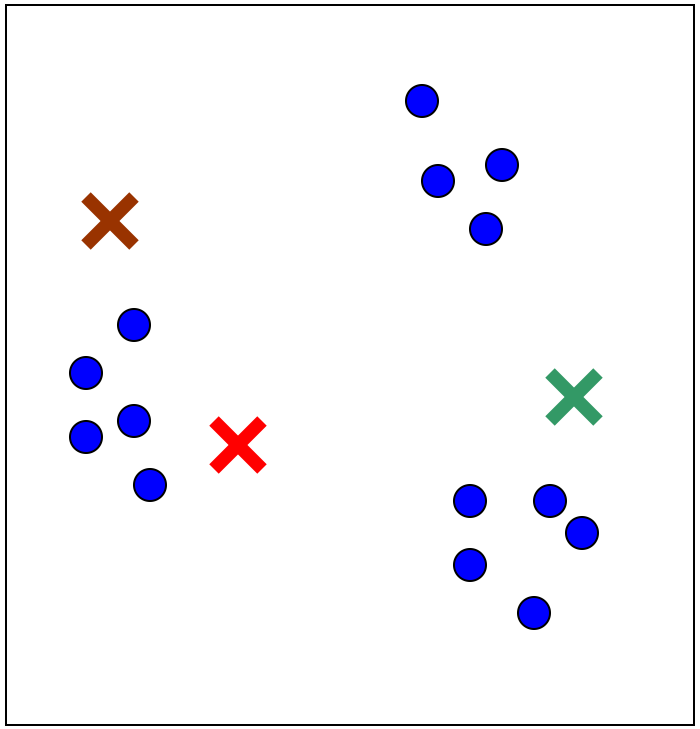
K-means Algorithm

1. Choose K centroids at random
2. Make initial partition of objects into k clusters by assigning objects to closest centroid
3. Calculate the centroid (mean) of each of the k clusters.
4. a. For object i, calculate its distance to each of the centroids.

   b. Allocate object i to cluster with closest centroid.

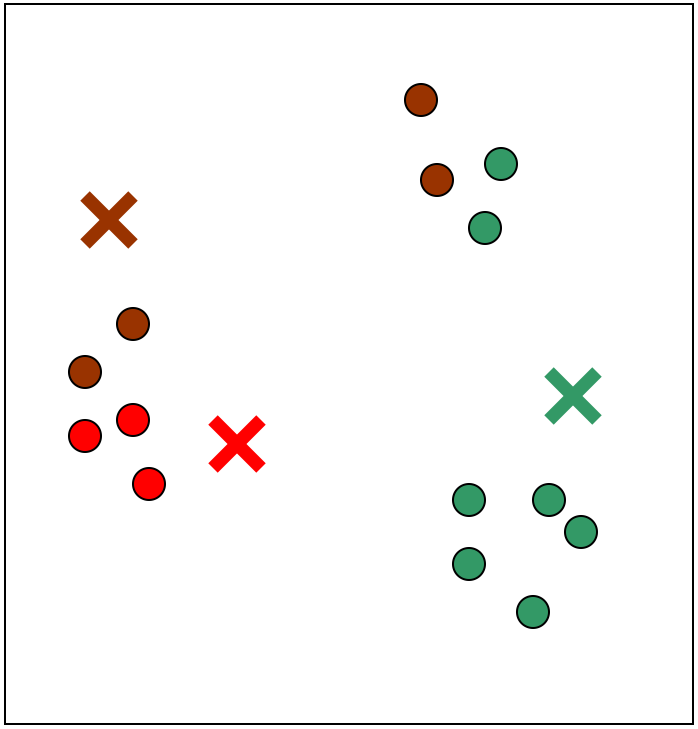
   c. If object was reallocated, recalculate centroids based on new clusters.
4. Repeat 3 for object i = 1,….N.
5. Repeat 3 and 4 until no reallocations occur.
6. Assess cluster structure for fit and stability

K-means Algorithm



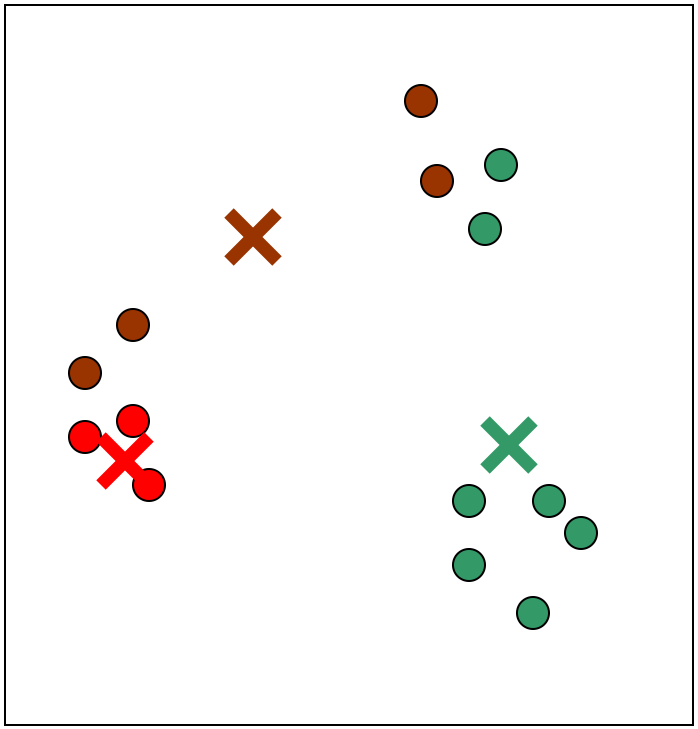
Iteration = 0

K-means Algorithm



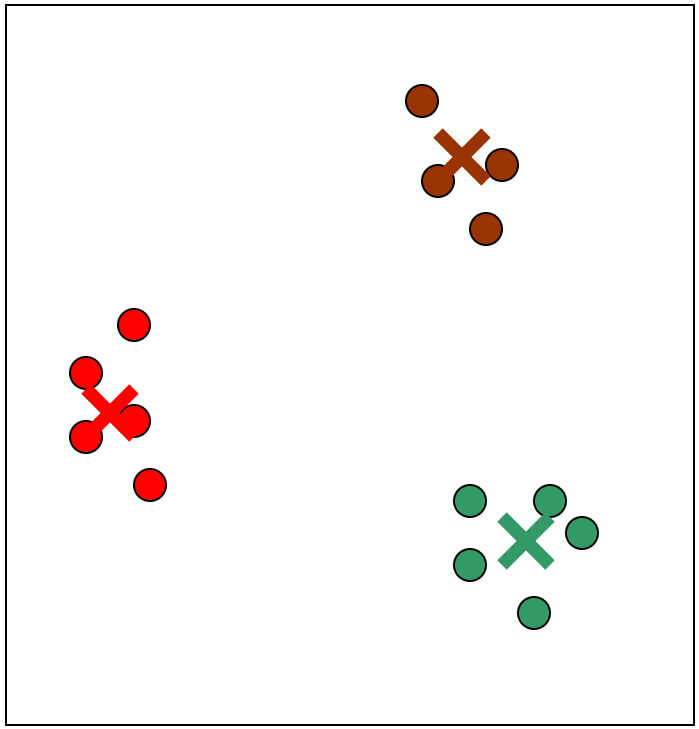
Iteration = 1

K-means Algorithm



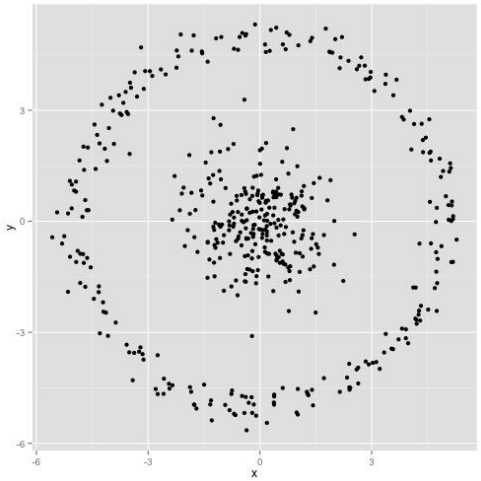
Iteration = 2

K-means Algorithm

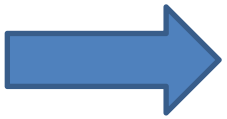


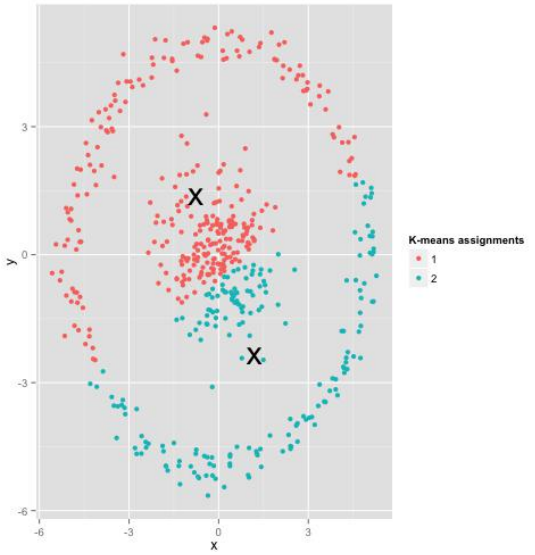
Iteration = 3

K-means problems

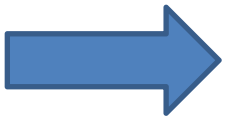


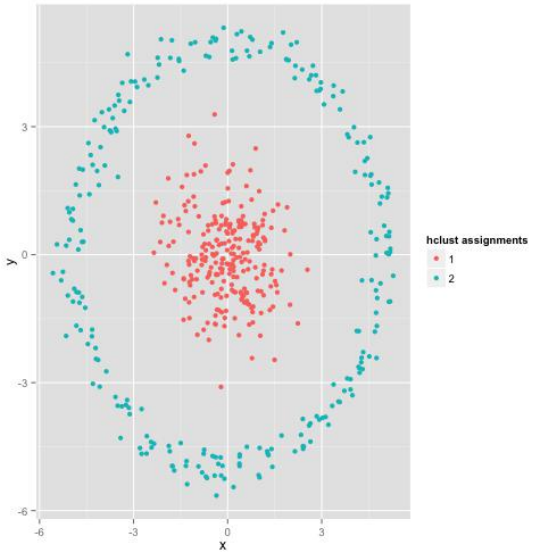
K-means



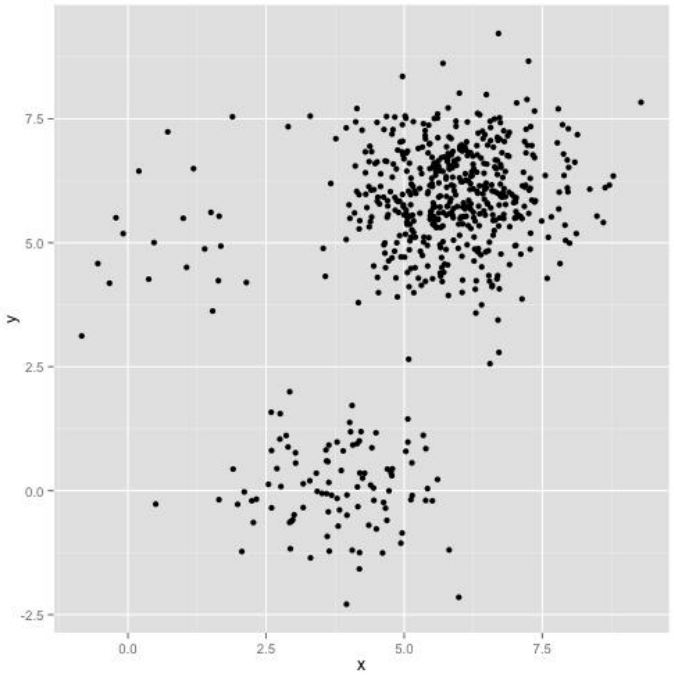


Hcluster

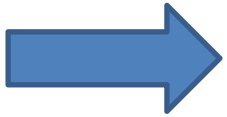


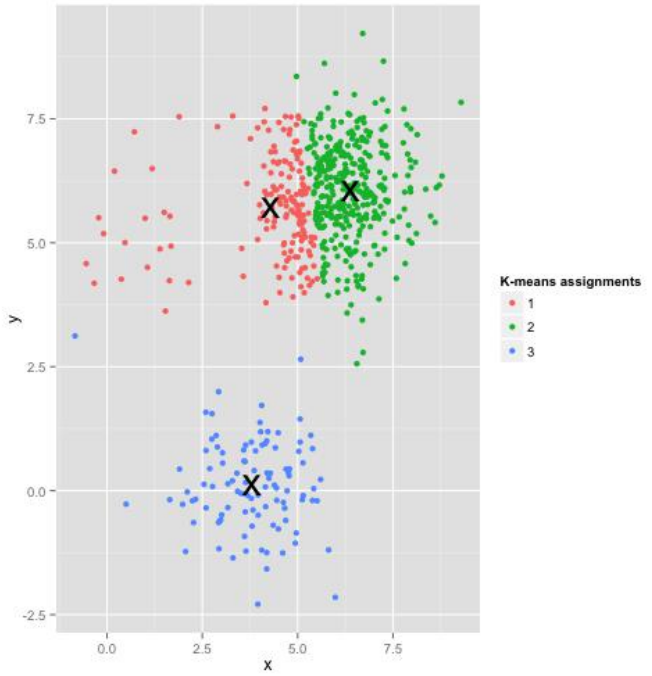


K-means problems

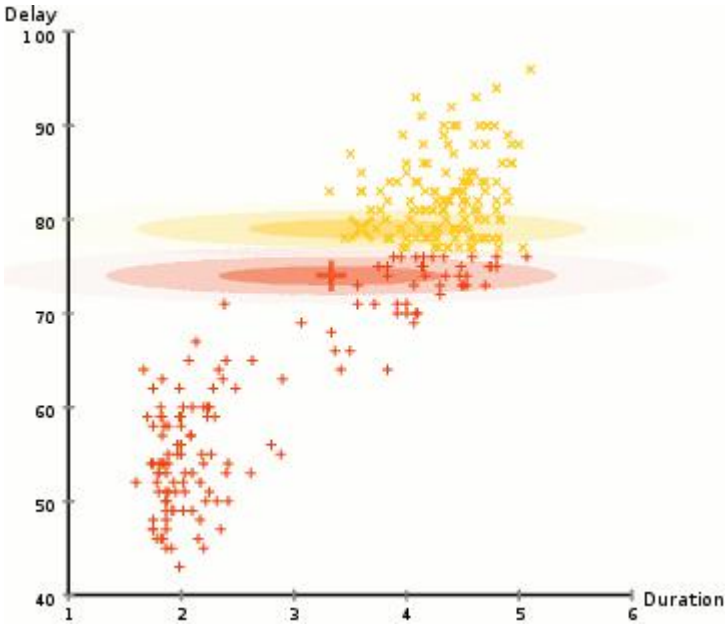


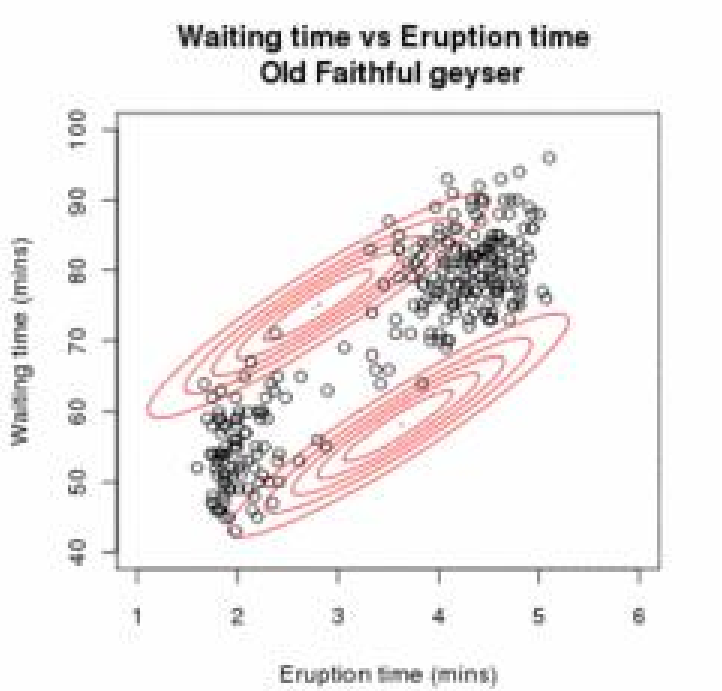
K-means





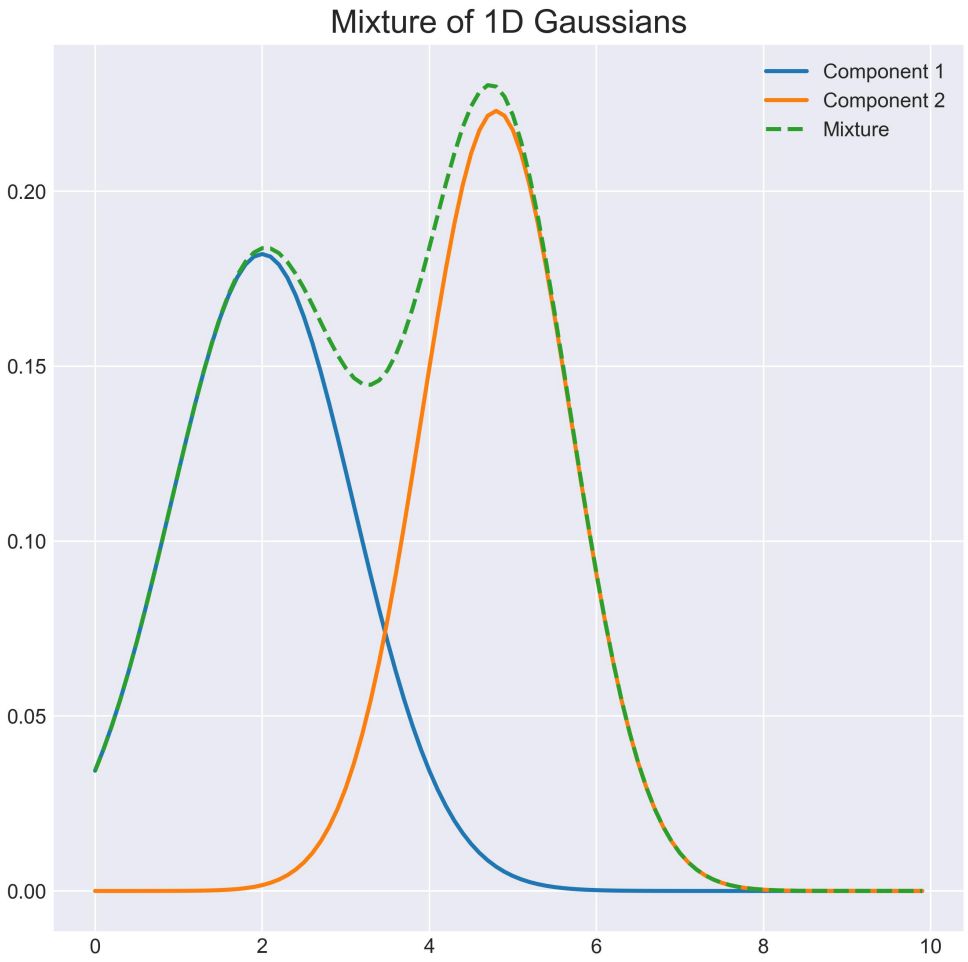
EM clustering



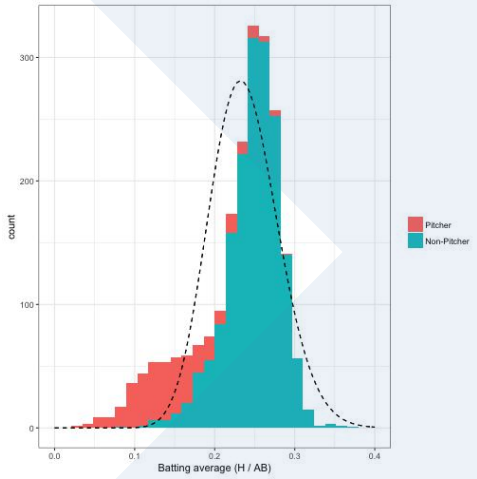


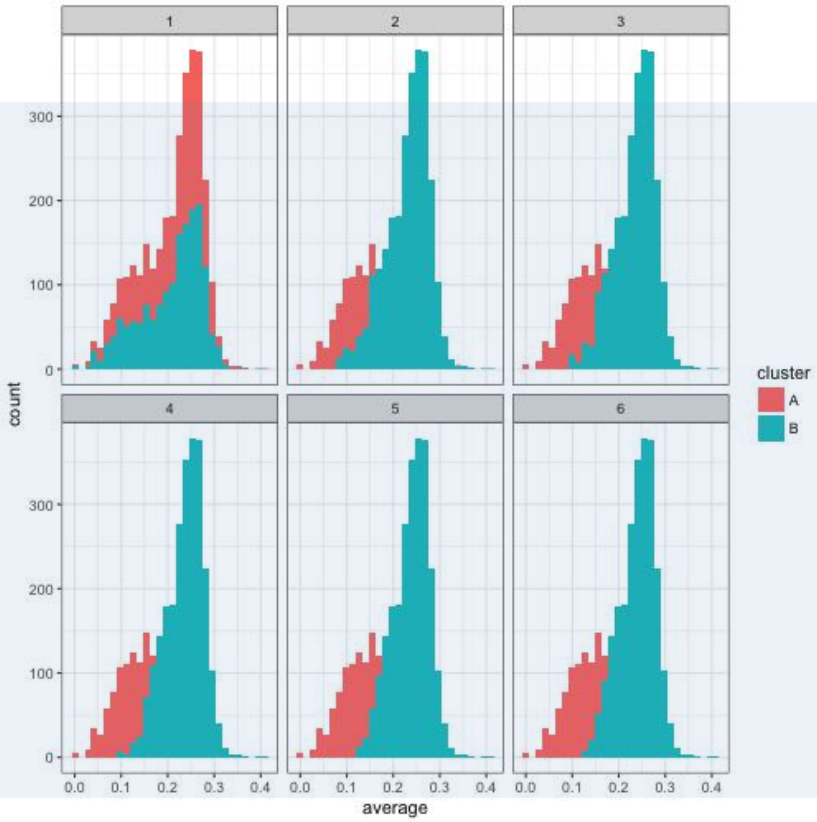
GMM-EM: Gaussian Mixture Model (GMM) using Expectation Maximization (EM)

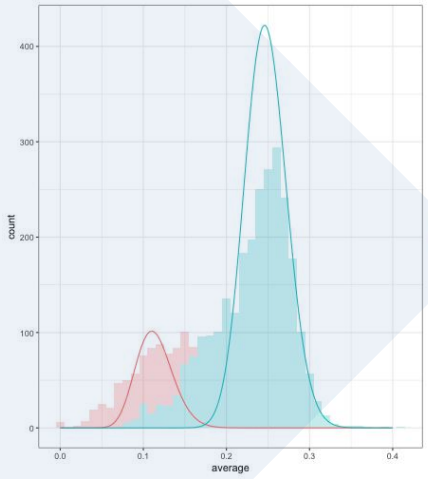
Gaussian Mixture Model



GMM-EM

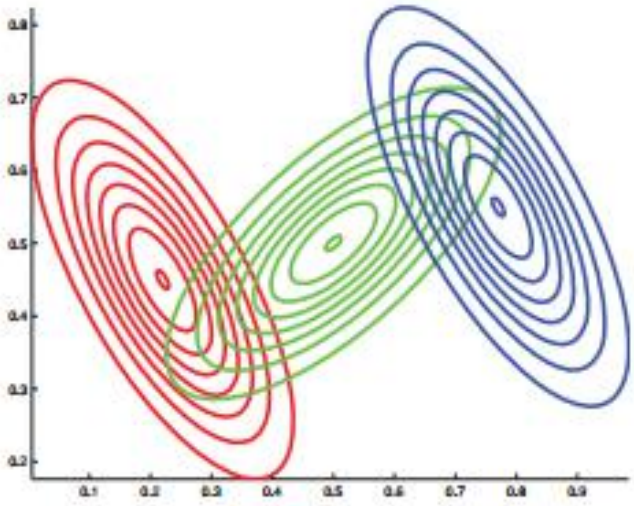






Reference: http://varianceexplained.org/r/mixture-models-baseball/

GMM-EM

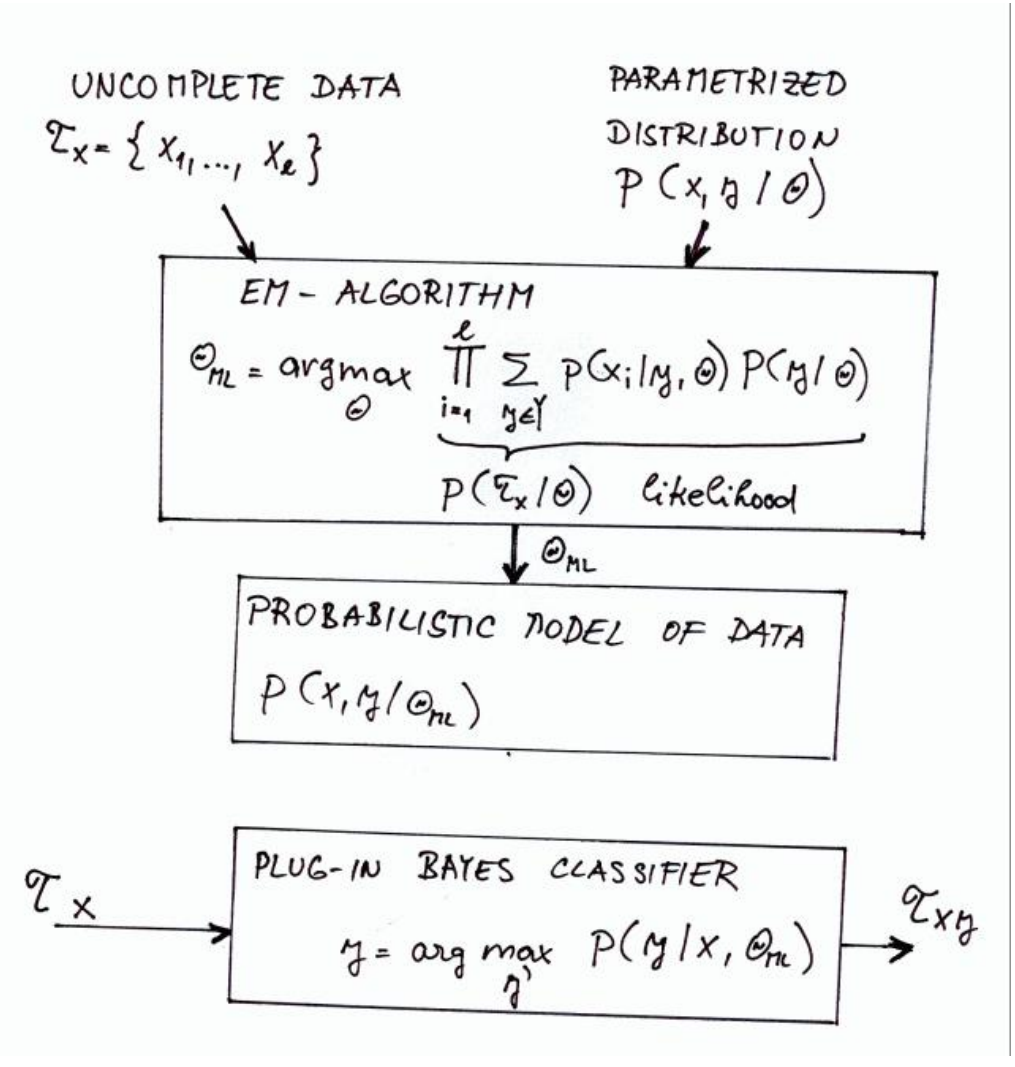


(a)

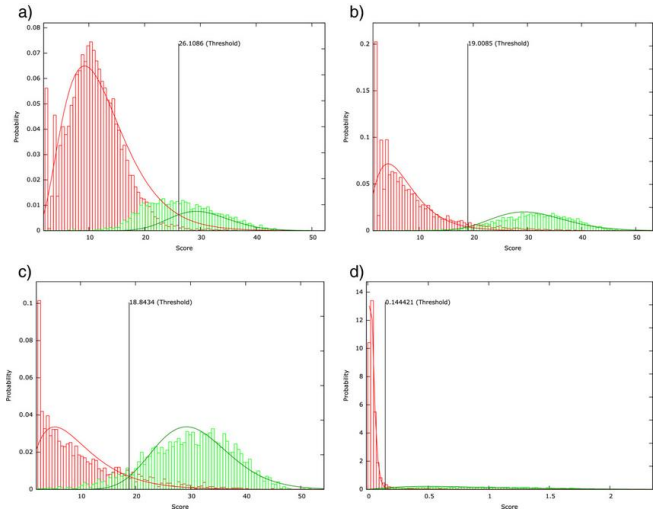
(b)

Analytical procedure

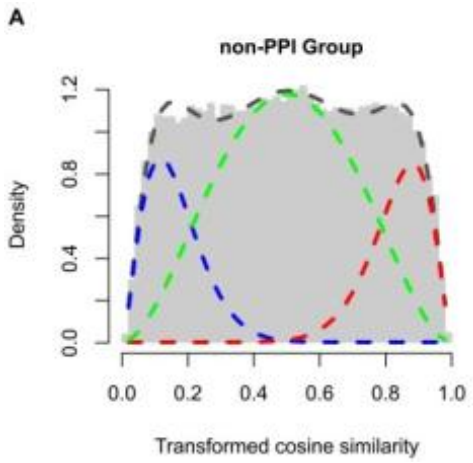
Expectation-Maximization Algorithm for clustering

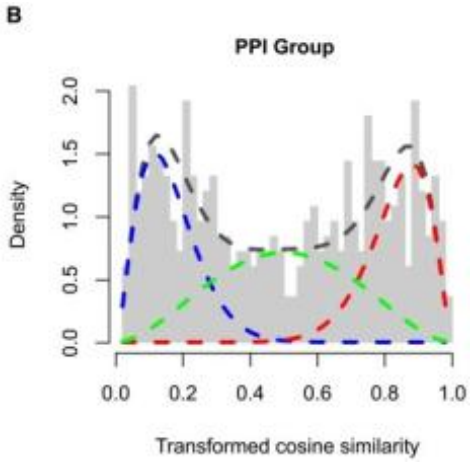


Other mixture models



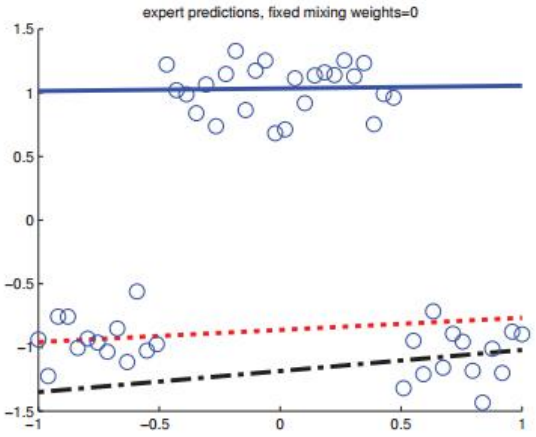
GMM: Gamma-Mixture Model



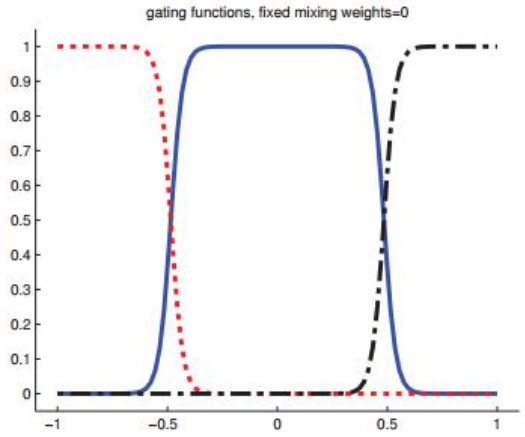


BMM: Beta-Mixture Mode

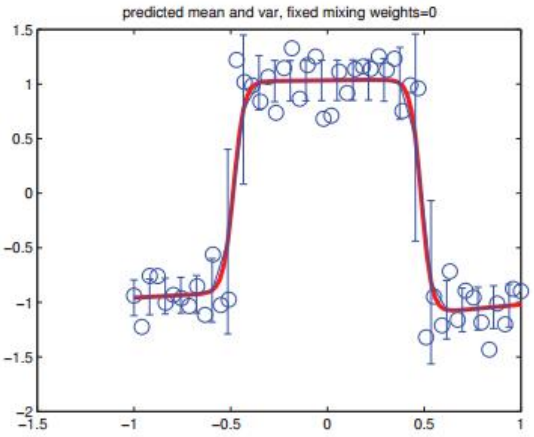
Other mixture models



(a)



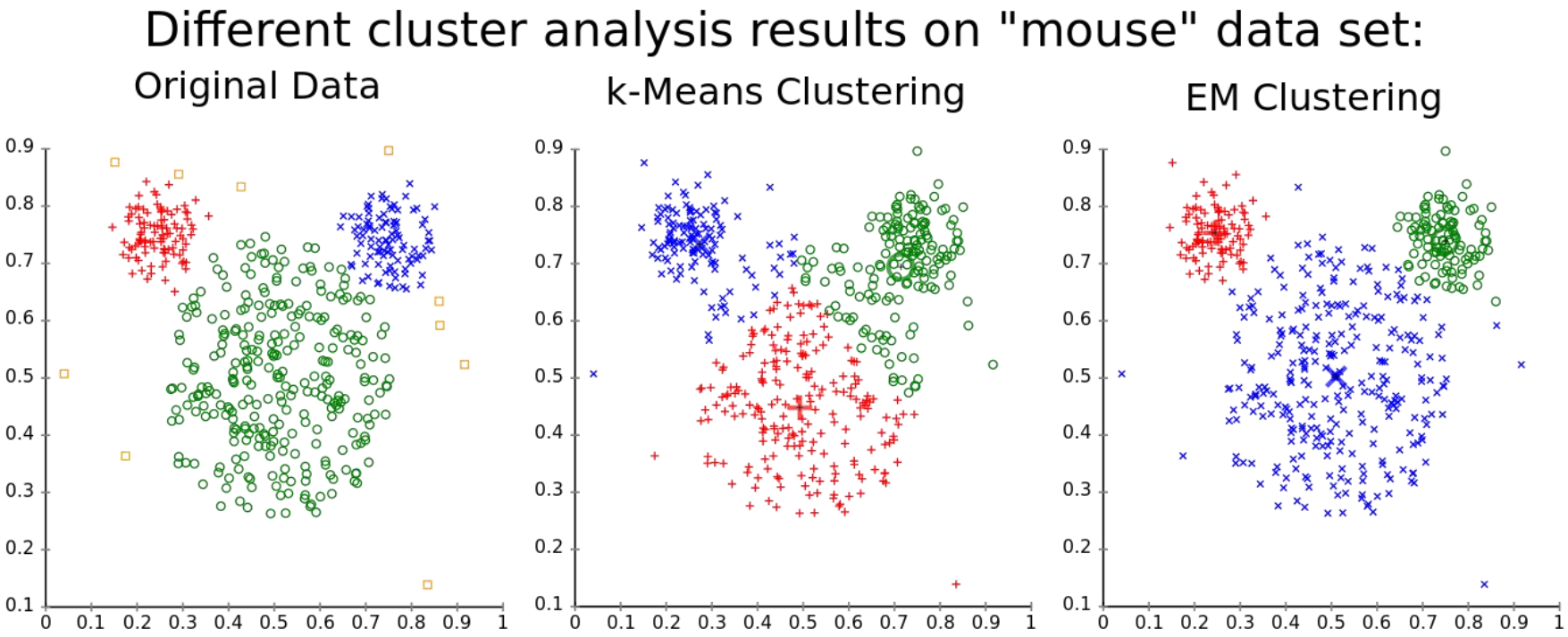
(b)



(c)

Mixtures of experts

K-means VS. EM

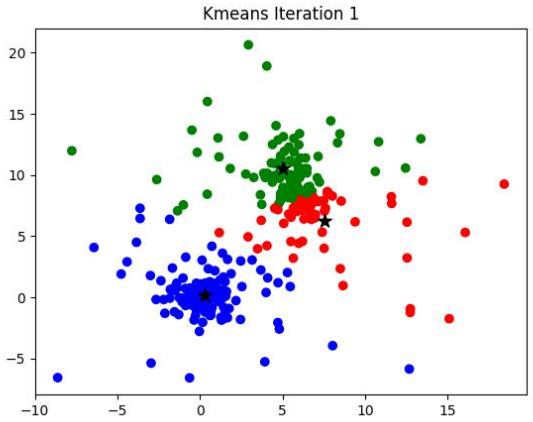


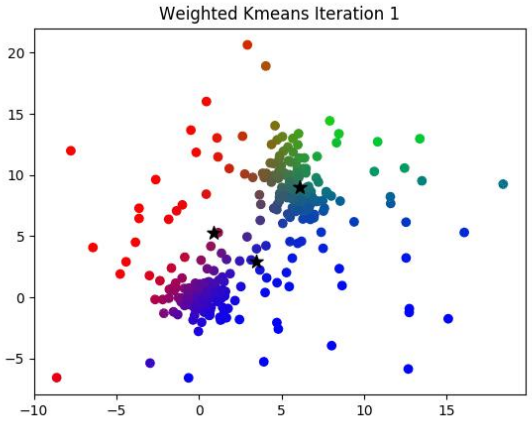
EM聚类与K-Means不同之处：并不计算距离，而是计算概率（并且明显要比K-Means复杂的多），用一个给定的多元高斯概率分布模型来估计出一个数据点属于一个聚类的概率，即将每一个聚类看作是一个高斯模型。

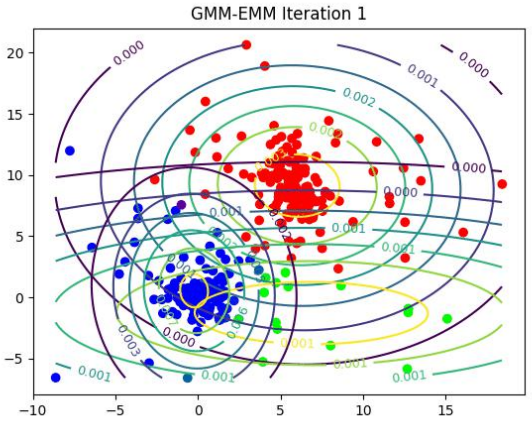
Reference: https://blog.csdn.net/eternity1118_/article/details/51516497

K-means VS. EM

GMM-EM: Gaussian Mixture Model (GMM) using Expectation Maximization (EM)



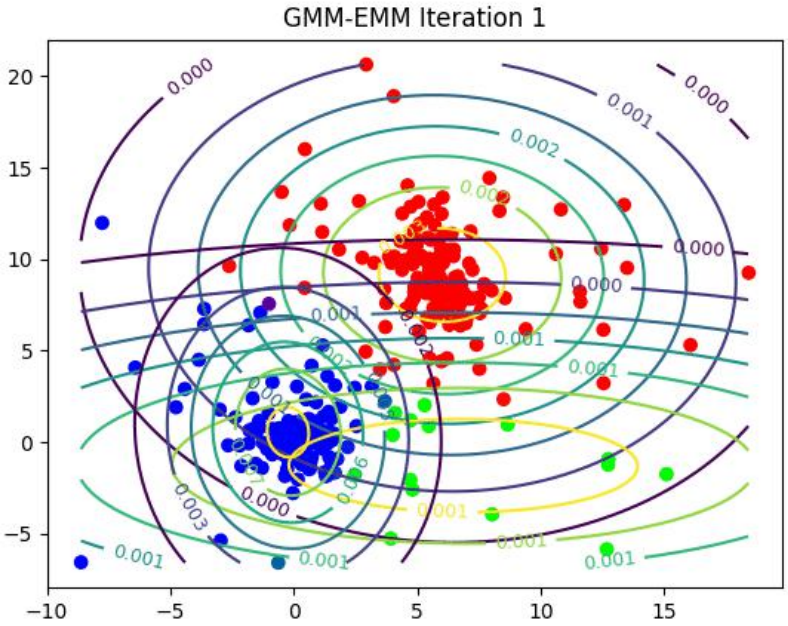


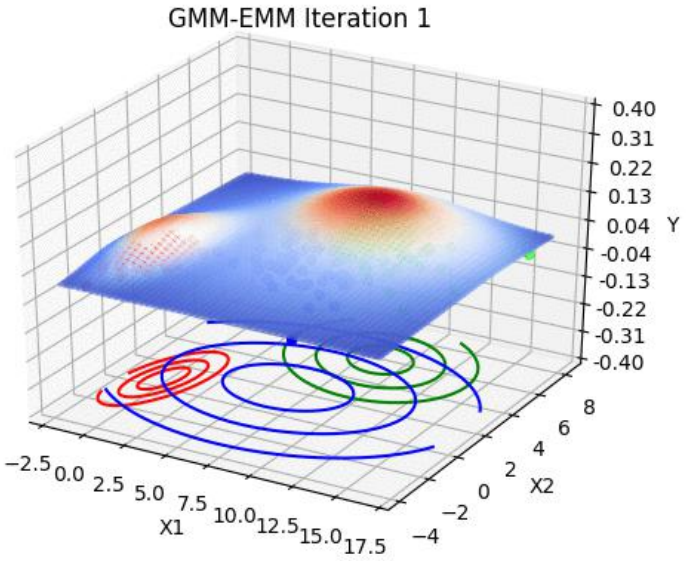


Reference: https://sandipanweb.wordpress.com/2017/03/19/hard-soft-clustering-with-k-means-weighted-k-means-and-gmm-em/

K-means VS. EM

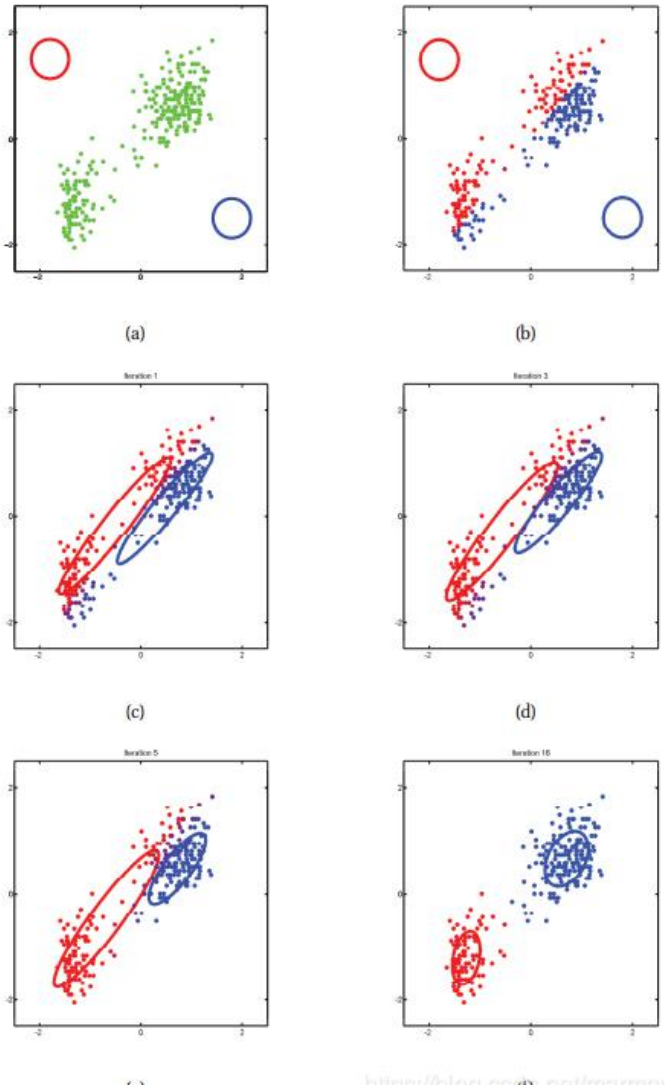
GMM-EM: Gaussian Mixture Model (GMM) using Expectation Maximization (EM)





Reference: https://sandipanweb.wordpress.com/2017/03/19/hard-soft-clustering-with-k-means-weighted-k-means-and-gmm-em/

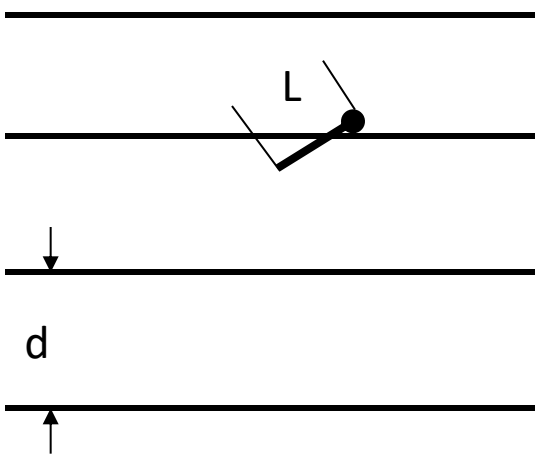
GMM-EM step-by-step



第5-2章 MCMC

- 引言
- Monte Carlo 近似计算
- 伪随机数生成
- 理论基础
- MCMC算法
- MCMC应用

Buffon投针实验(1777)



设针与平行线的夹角为ϕ，针的中点与最近的平行线的距离为X，相交条件为:

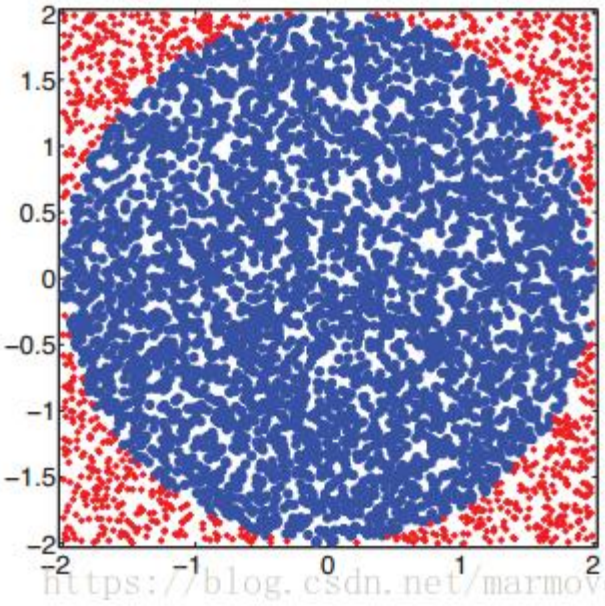
$$X \le \frac{L\sin\phi}{2}$$

$$p = \Pr\{X \le \frac{L\sin\phi}{2}\} = \frac{2}{d\pi}\int_0^{\pi}\left(\int_0^{\frac{L\sin\phi}{2}} dx\right)d\phi = \frac{2L}{d\pi}$$

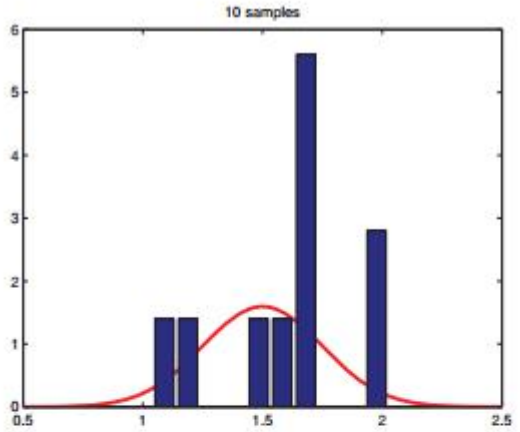
Buffon实验

学者	年代	l/d	总次数	成功次数	π 的估计
Wolf	1850	0.8	5000	2532	3.15956
Smith	1855	0.6	3204	1218	3.1554
Fox	1884	0.75	1030	489	3.1595
Lazzarini	1907	0.833	3408	1808	3.14159292

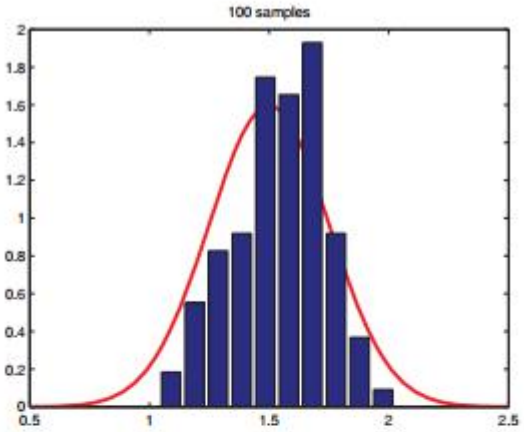
Monte Carlo方法



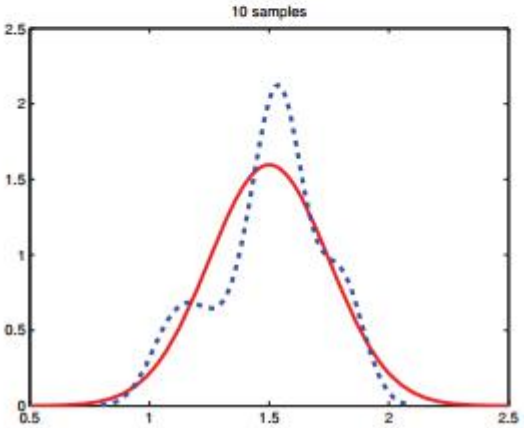
Monte Carlo方法



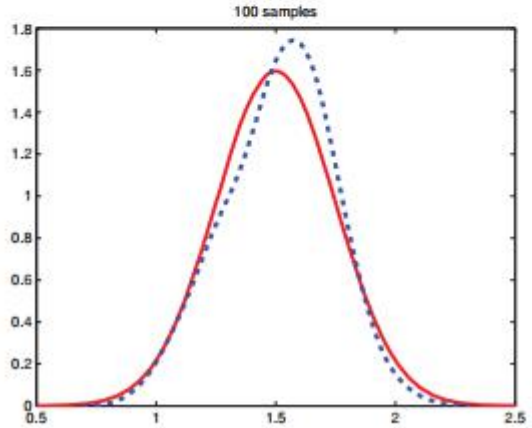
(a)



(b)



(c)



(d)

Monte Carlo方法

- Von Neumann
- S.Ulam (1946)
- N.Metropolis (1953)
- Hasting (1970)

- Monte Carlo (Monaco), famous for its gambling casino.

Pseudo-Random Numbers

- Truly random numbers can not be generated on a computer
- Pseudo-random numbers follows a well-defined algorithm, thus predictable and repeatable
- Have nearly all the properties of true random numbers

Linear congruence generator (LCG)

- One of the earliest and fastest algorithm:

$$X_{n+1} = (aX_n + c) \mod M$$

where $0 \leq X_n < M$, M is the modulus, a is multiplier, c is increment. All of them are integers. Choice of a, c, M must be done with special care.

http://random.mat.sbg.ac.at/links/rando.html

Choice of Parameters

Source	m	a	c	output bits of seed in *rand()* / *Random(L)*
Numerical Recipes	2^{32}	1664525	1013904223	
Borland C/C++	2^{32}	22695477	1	bits 30..16 in *rand()*, 30..0 in *lrand()*
glibc (used by GCC)[4]	2^{32}	1103515245	12345	bits 30..0
ANSI C: Watcom, Digital Mars, CodeWarrior, IBM VisualAge C/C++	2^{32}	1103515245	12345	bits 30..16
Borland Delphi, Virtual Pascal	2^{32}	134775813	1	bits 63..32 of *(seed * L)*
Microsoft Visual/Quick C/C++	2^{32}	214013	2531011	bits 30..16
RtlUniform from Native API [5]	$2^{31} - 1$	2147483629	2147483587	
Apple CarbonLib	$2^{31} - 1$	16807	0	see MINSTD
MMIX by Donald Knuth	2^{64}	6364136223846793005	1442695040888963407	
VAX's **MTH$RANDOM**,[6] old versions of glibc	2^{32}	69069	1	
Random class in Java API	2^{48}	25214903917	11	bits 47...16
LC53[7] in Forth (programming language)	$2^{32} - 5$	$2^{32} - 333333333$	0	

$$x_{n+1} = (ax_n + c) \mod M$$

http://en.wikipedia.org/wiki/Linear_congruential_generator
http://crypto.mat.sbg.ac.at/results/karl/server/server.html

Other Modern Generators

- Mersenne Twister (MT19937)

  Extremely long period (2^{19937}-1), fast

  (http://www.math.keio.ac.jp/~matumoto/emt.html)

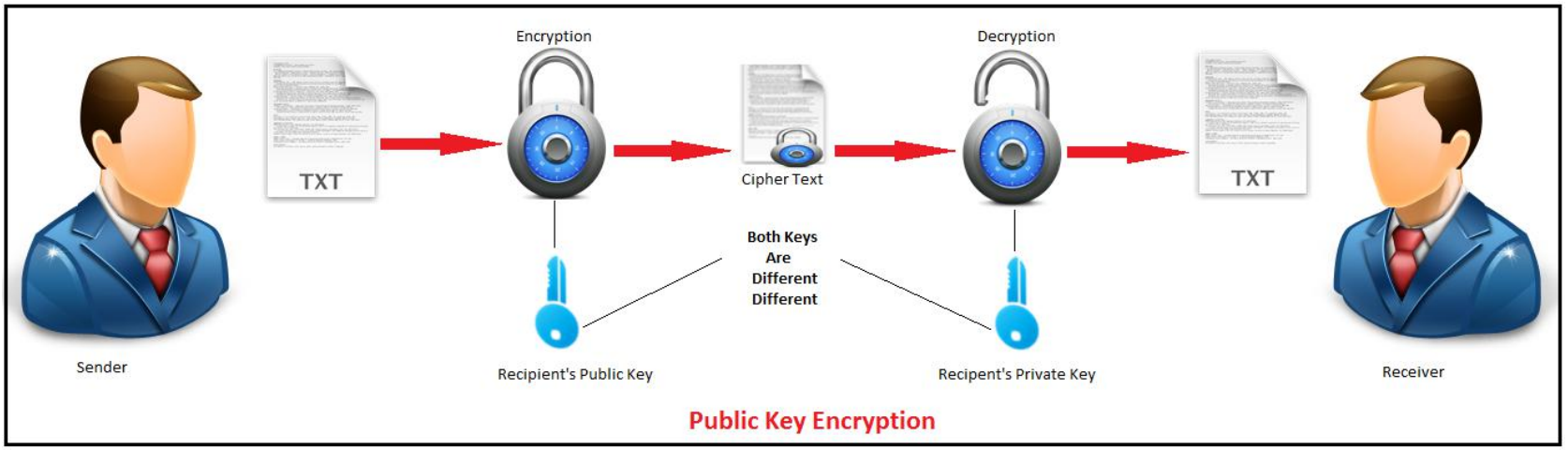
- Inversive congruential generator

$$X_{n+1} = (ax_n^{-1} + c) \mod M$$

  nonlinear, no lattice structure

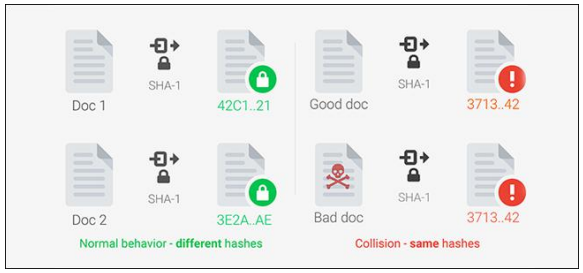
http://en.wikipedia.org/wiki/Mersenne_twister
http://en.wikipedia.org/wiki/Inversive_congruential_generator

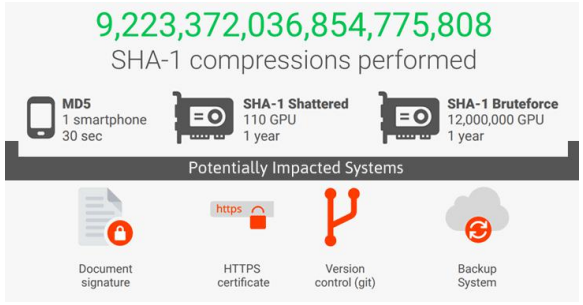
Public key system



RSA algorithm

Adi Shamir, Ron Rivest and Len Adleman
1977



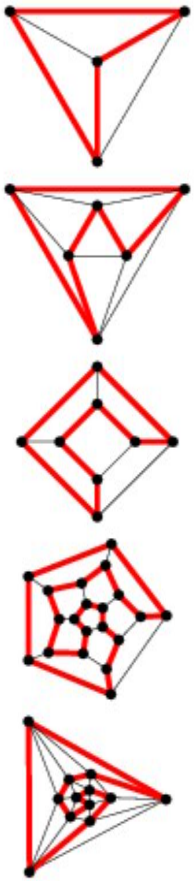


Collision attack

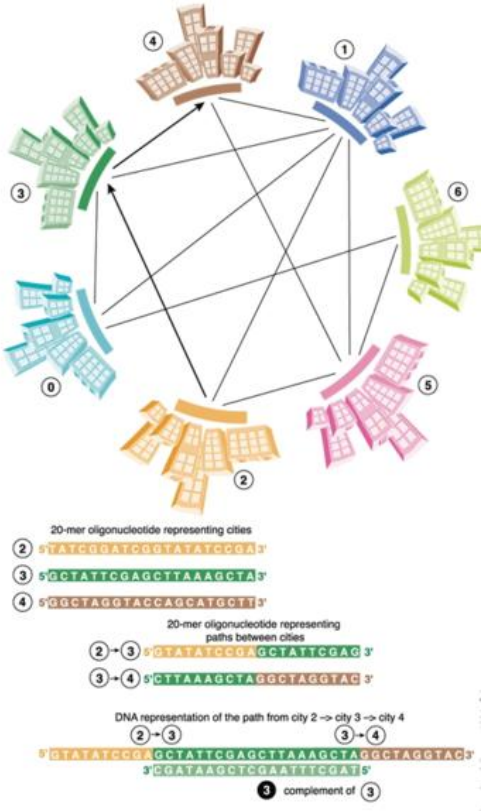
Xiaoyun Wang @ China
2004

Extension: from statistics back to DNA

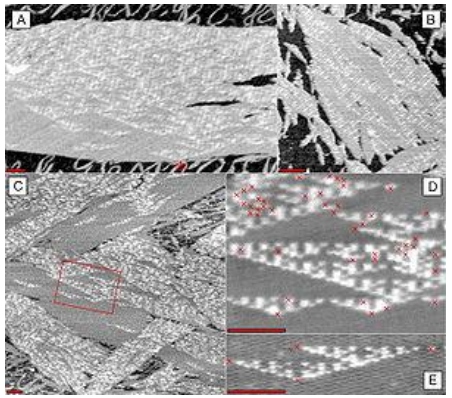
Hamilton path problem



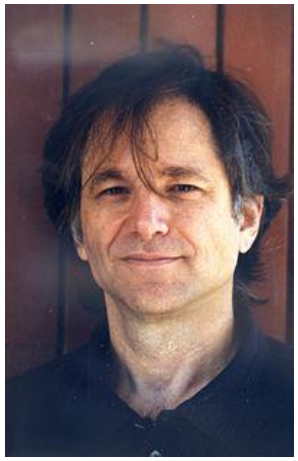
DNA computing



Solution

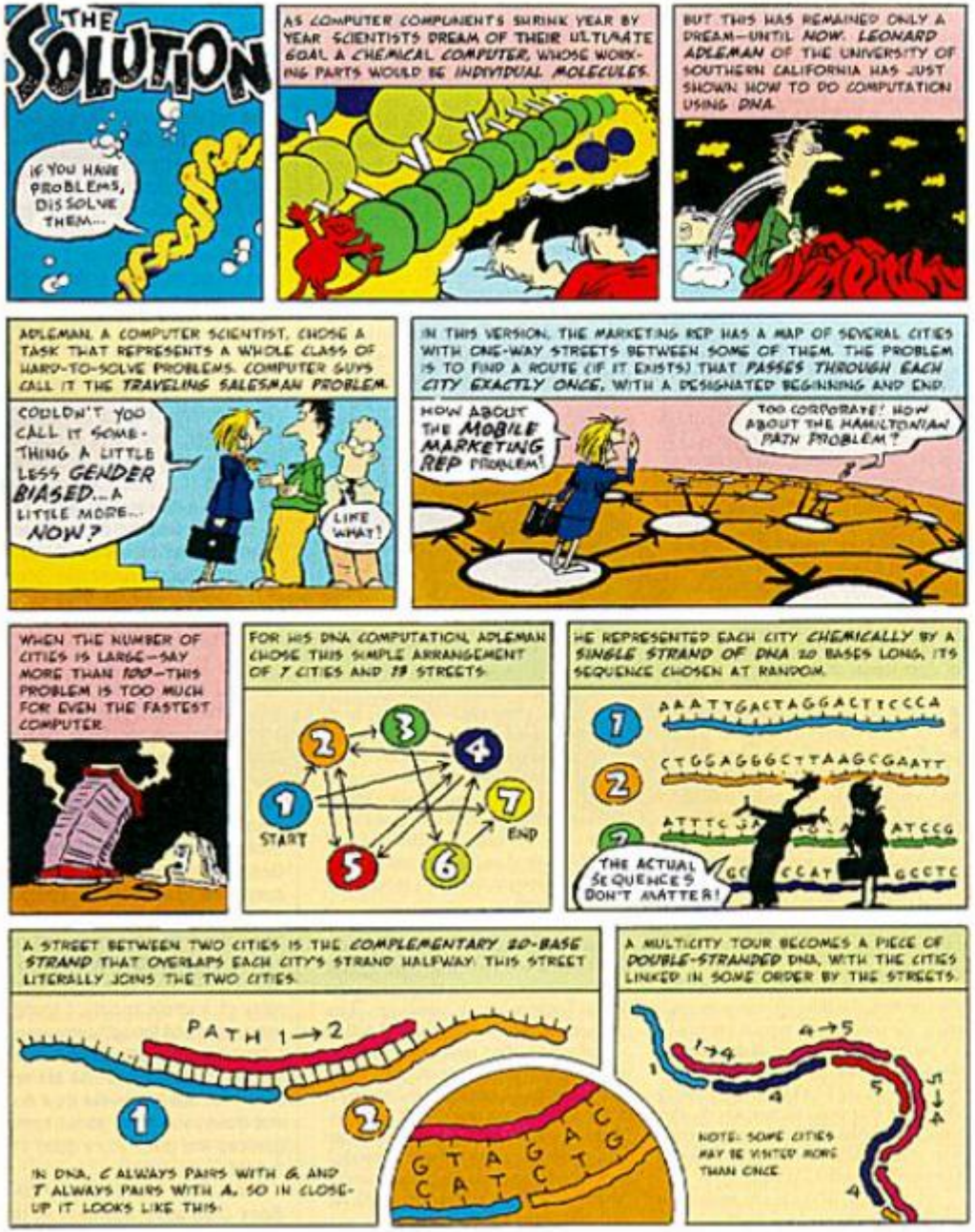


Leonard Adleman



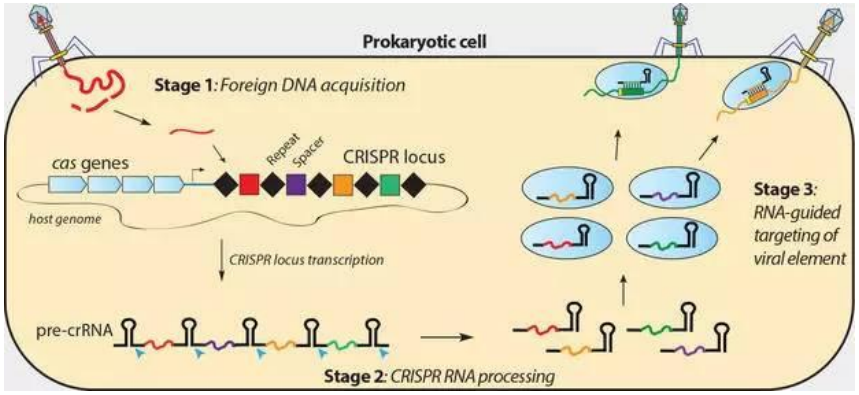
SCIENCE CLASSICS

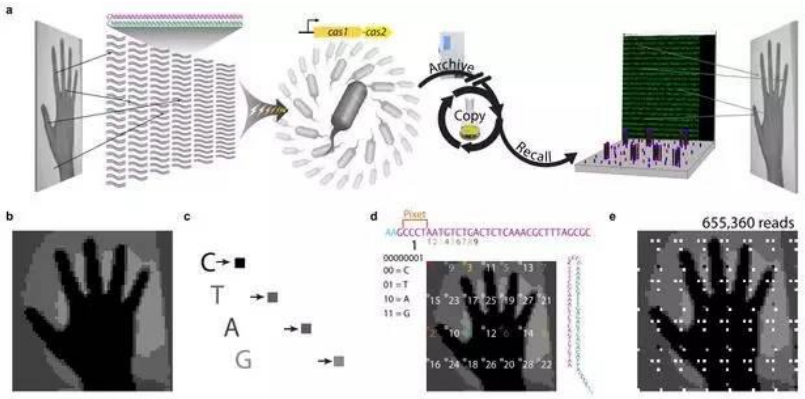
BY LARRY GONICK

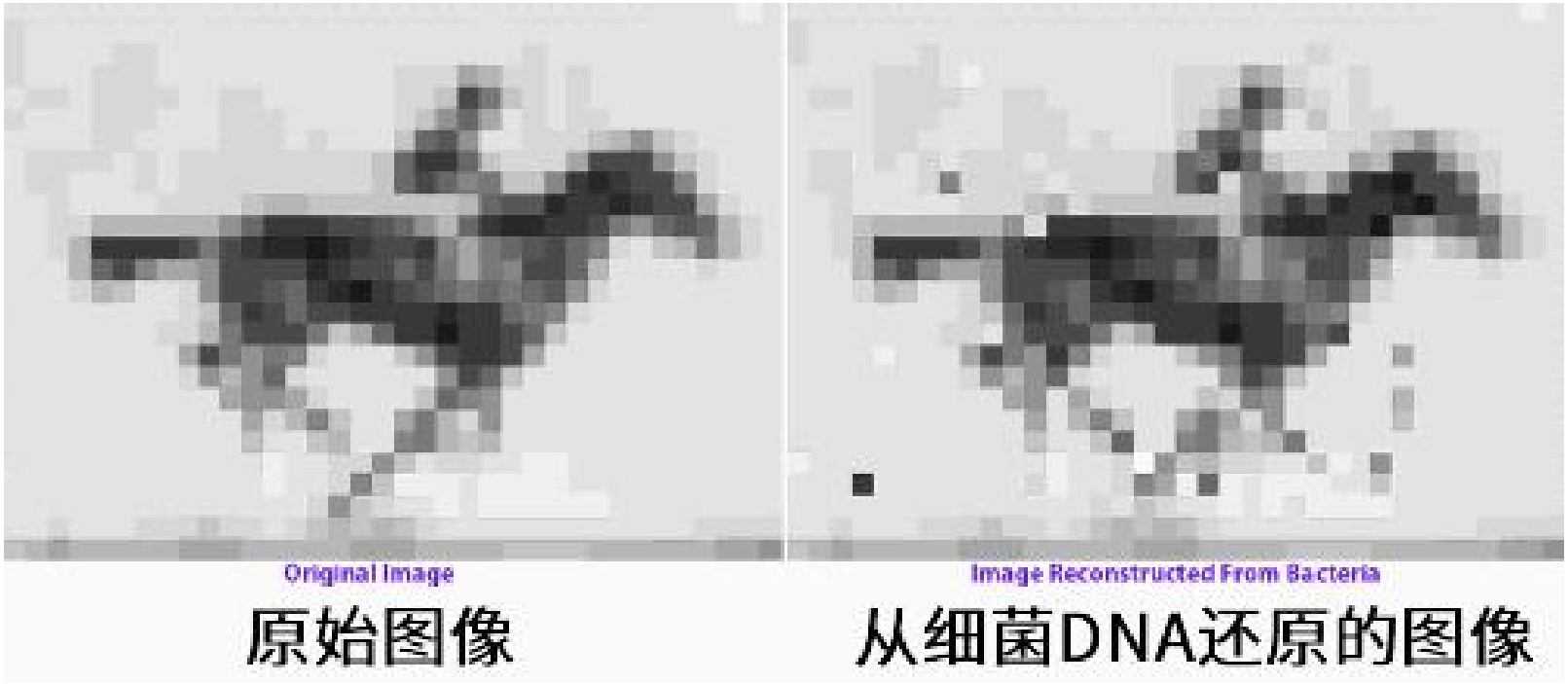


Discover magazine published an article in comic strip format about Leonard Adleman's discovery of DNA computation. Not only entertaining, but also the most understandable explanation of molecular computation I have Ever seen.

Understand it, create it!







CRISPR–Cas encoding of a digital movie into the genomes of a population of living bacteria, Nature, 2017

马氏链的遍历性与遍历极限

- **马氏链的遍历性定理：**若有限状态Markov链的所有状态都是互通的(不可约)，f是状态空间上的有界实值函数且满足 $\sum_i |f(i)|\pi_i < +\infty,$ 则

$$Pr\left(\frac{f(\xi_1)+\cdots+f(\xi_n)}{n} \to \sum_i f(\xi_i)\pi_i\right) = 1$$

其中π是MC的不变分布，而且此极限与初分布无关.

MCMC 算法的思想

- 对于非周期的MC (例如$p_{ij}>0$时)，我们还有

$$p_{ij}(n) \to \pi_j, \qquad n \to +\infty$$

- **Markov Chain Monte Carlo** 算法的思想就是：设计一个马氏链, 使得它的极限分布π与f成比例, 于是当我们模拟马氏链足够多步后, 它的分布就近似于π.

Markov Chain Monte Carlo

- Goal: $p(z^{(m)}) = p^*(z)$ as $m \to \infty$
  - MCMCs that have this property are **ergodic.**

- Transition properties that provide **detailed balance** guarantee ergodic MCMC processess.
  - Also considered **reversible**.

$$p^*(z)T(z, z') = p^*(z')T(z', z)$$

Markov Chain Monte Carlo (MCMC)

- 设计一个 Markov 链，使其不变分布为我们关心的分布, (如高维分布, 或样本空间非常大的离散分布)。用这个 Markov 链的样本, 来对该分布作采样, 并用以作随机模拟。这样的方法, 统称为 **Markov Chain Monte Carlo (MCMC)方法。**
- 由于这种方法的问世, 使随机模拟在很多领域的计算中, 相对于决定性算法，显示出它的巨大的优越性。而有时随机模拟与决定性算法的结合使用, 会显出更多的长处.

MCMC 的应用

- 用于生成较复杂的随机数： 高维分布的随机向量。
- 求复杂空间上函数的极值： 模拟退火；
- 缺失数据的参数估计：Gibbs Sampler
- Bayesian 缺失数据问题

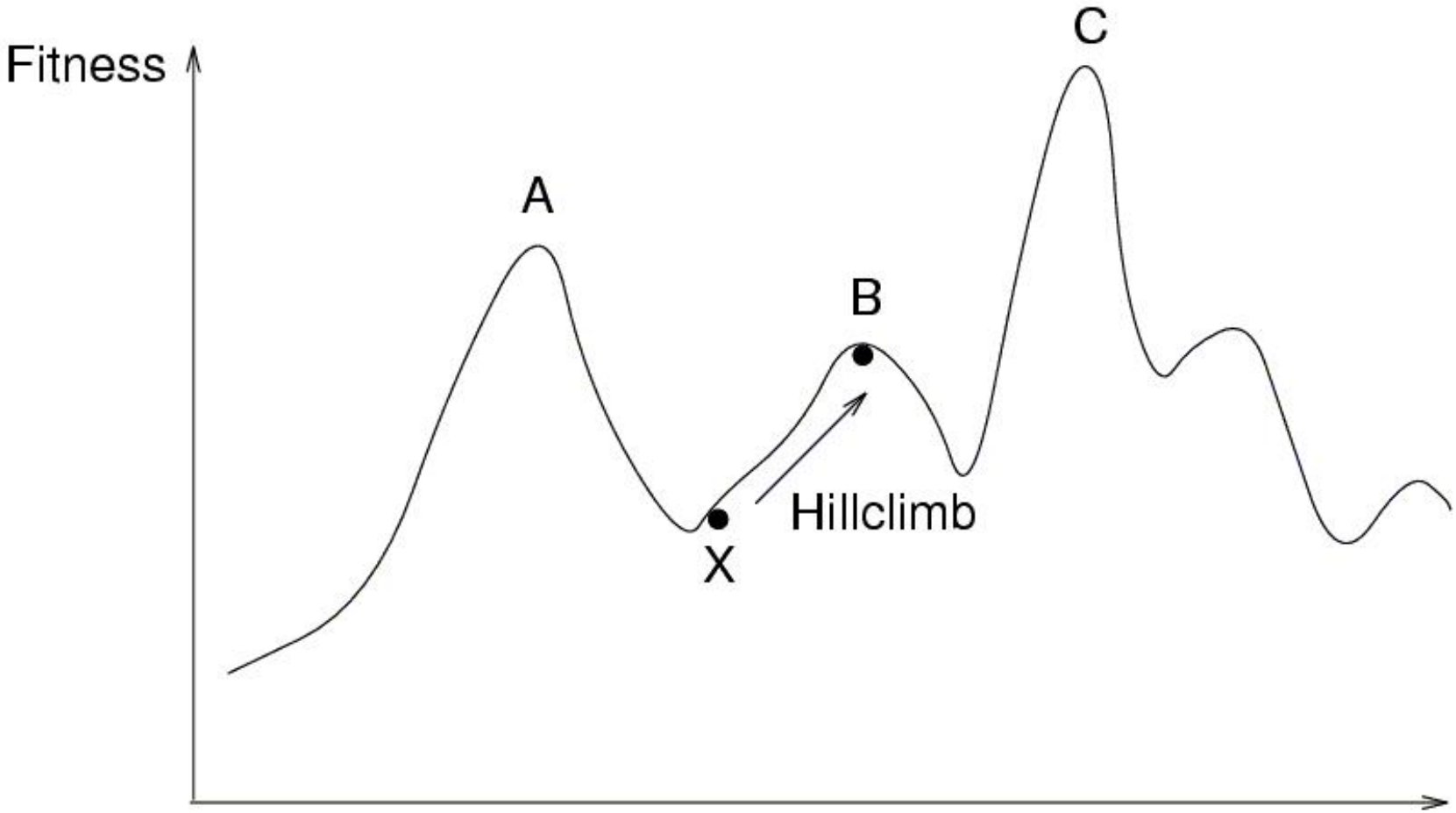
Gibbs Sampler

- 生成一元随机变量是并不困难的， 但是生成高维各分量不独立的随机向量就非常困难。
- Gibbs 采样法的思想是**通过条件分布**得到以给定分布π为不变分布的马氏链的转移概率.

Metropolis 采样法概述

- 与Gibbs采样法一样，Metropolis方法也给出了在计算机上用 马氏链近似模拟遵从一个分布 π 的随机变量(向量) 的一个算法。
- Metropolis 提出了这种采样法, 称为**Metropolis采样法**。

Hill Climbing



The Problem with Hill Climbing

- Gets stuck at local minima
- Possible solutions
  - Try several runs, starting at different positions
  - Increase the size of the neighbourhood (e.g. in TSP try 3-opt rather than 2-opt)

模拟退火

- 实质就是要构造适当的Markov链，使得相应Boltznann分布作为其极限分布

- 温度T有一个调整的过程，首先升温保证初始点几乎以均一的概率都能被访问到，而后逐渐降温，使得分布越来越集中到最大值点

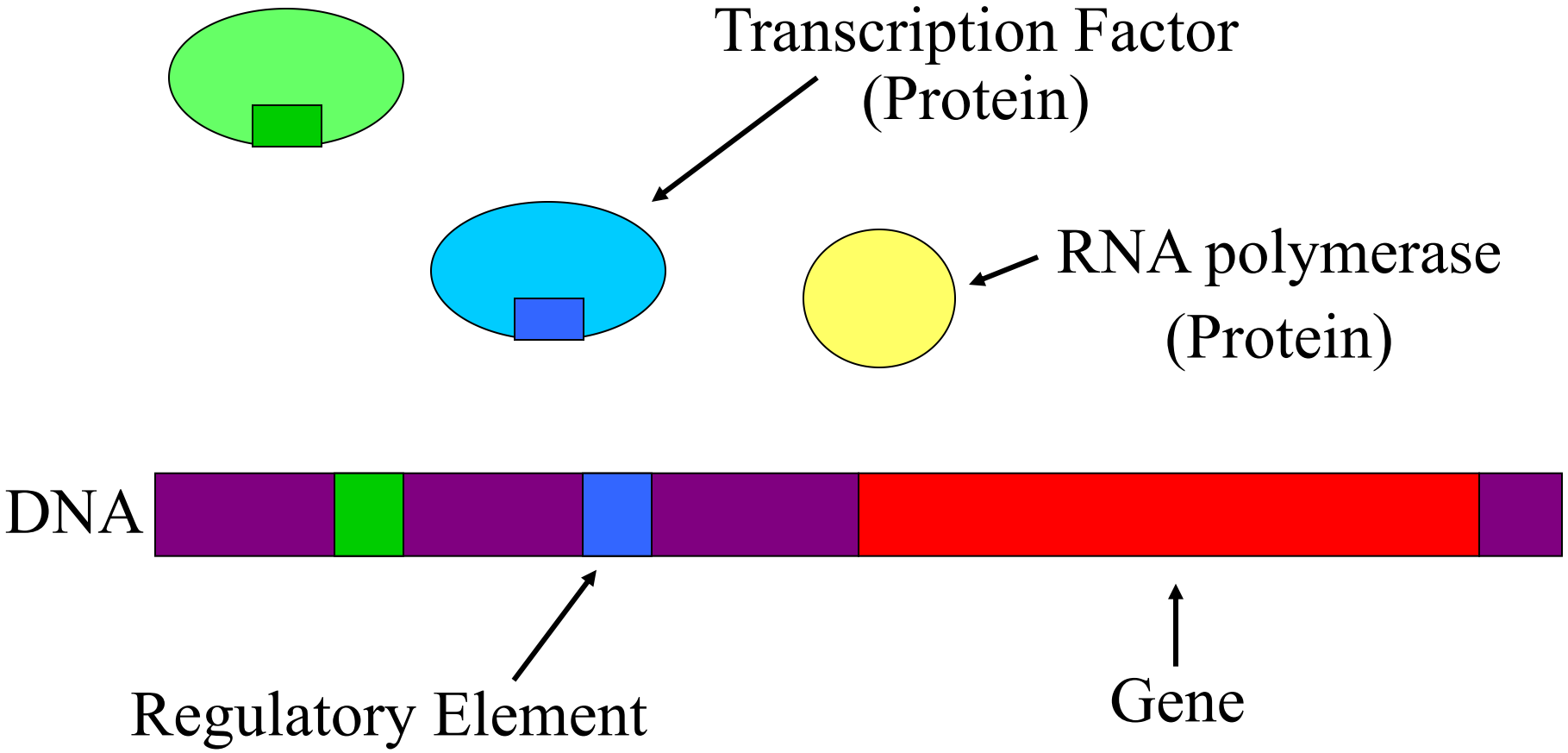
第5-2章 Motif Finding

- Motif finding problem
- EM algorithm
- Markov chain Monte Carlo (Gibbs Sampler)

Transcriptional Regulation

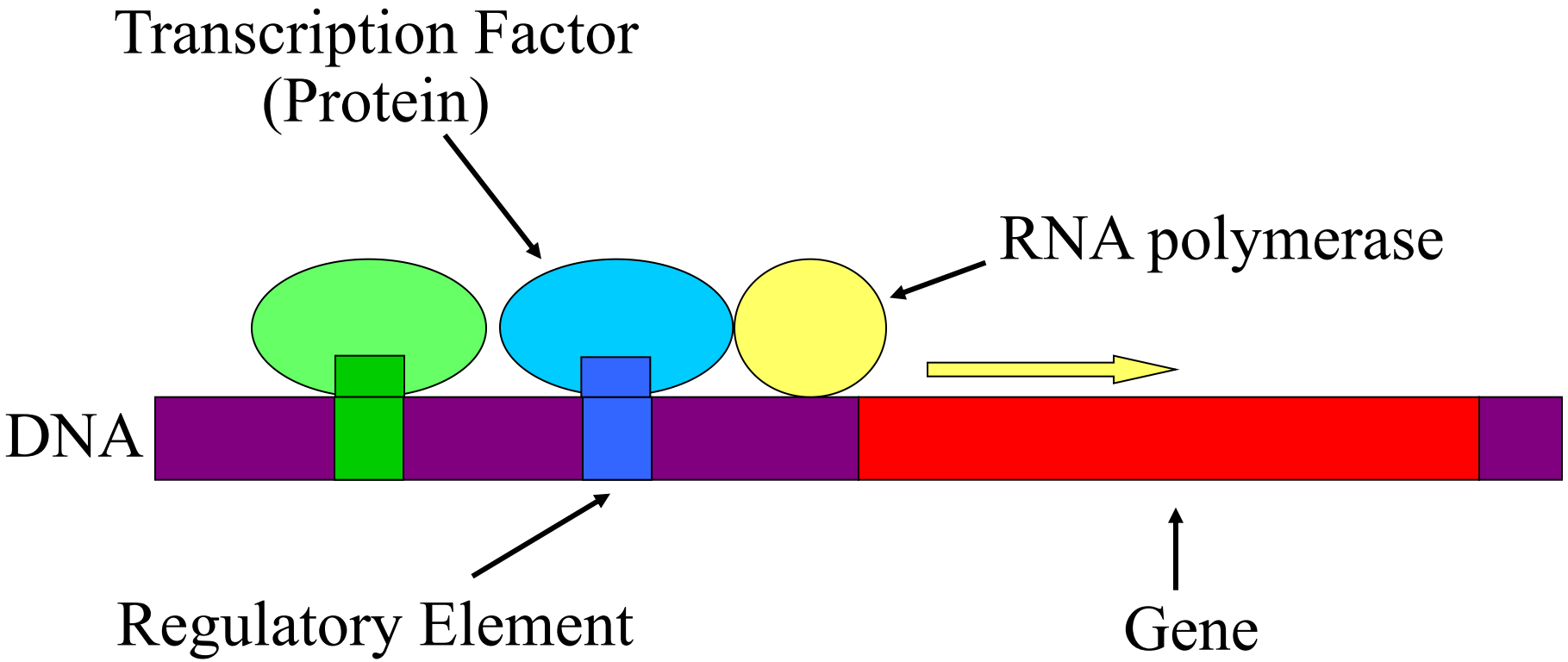
- The transcription of each gene is controlled by a regulatory region of DNA relatively near the transcription start site (TSS).
- two types of fundamental components
  - short DNA regulatory elements
  - *gene regulatory proteins* that recognize and bind to them.

Regulation of Genes



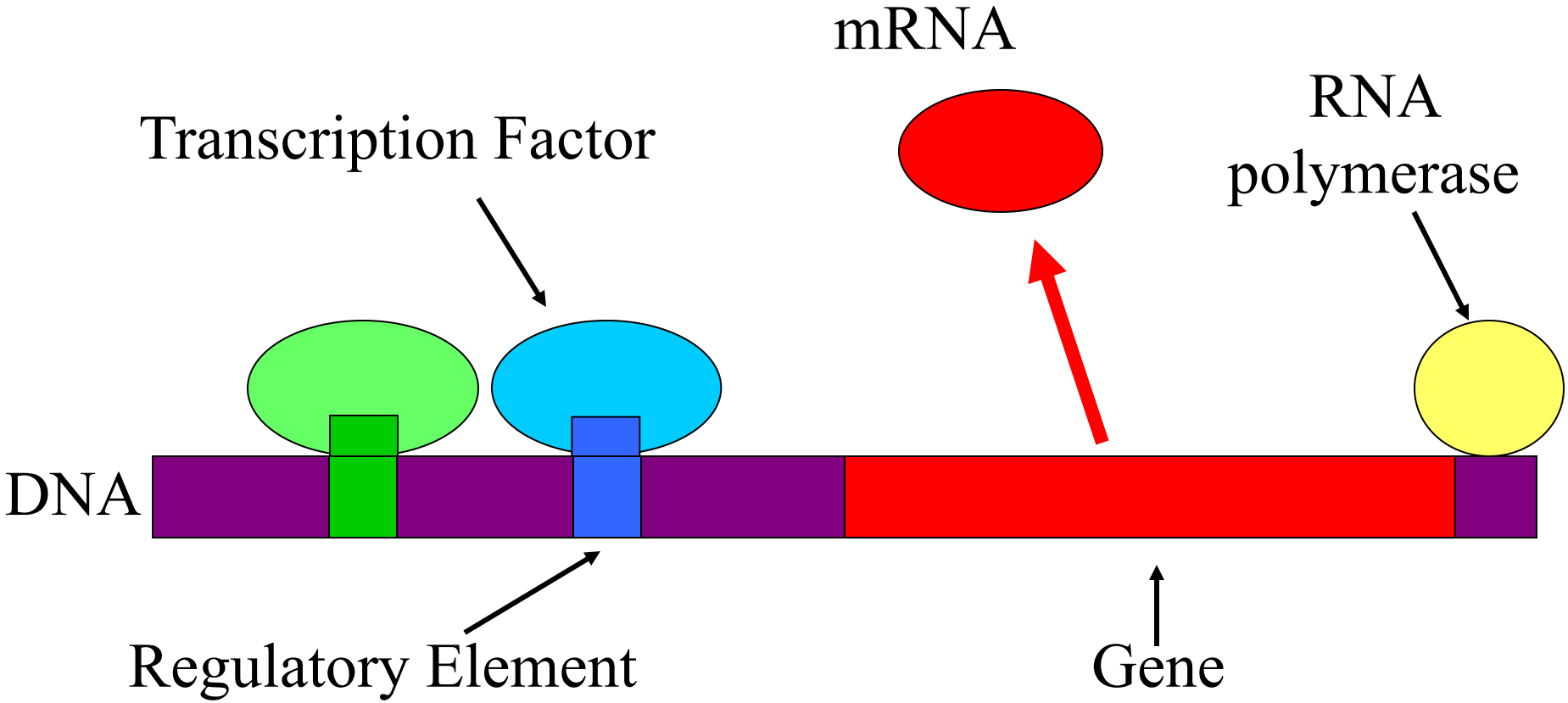
source: M. Tompa, U. of Washington

Regulation of Genes



source: M. Tompa, U. of Washington

Regulation of Genes



source: M. Tompa, U. of Washington

Transcriptional Binding Site

Wiki: DNA binding sites are a type of binding site found in DNA where other molecules may bind

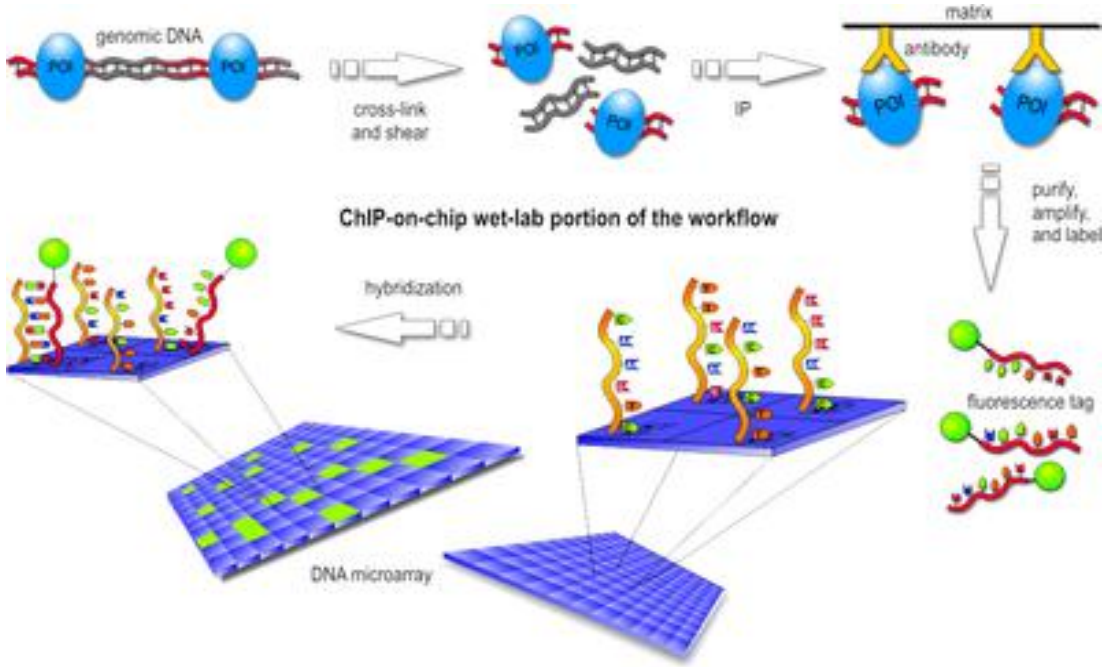
- Small (6-20bp)
- Highly variable

Experimental Method (I)

- DNase footprinting assay: The method uses an enzyme,deoxyribonuclease (DNase, for short), to cut the radioactively end-labeled DNA, followed by gel electrophoresis to detect the resulting cleavage pattern

Experimental Method (II)

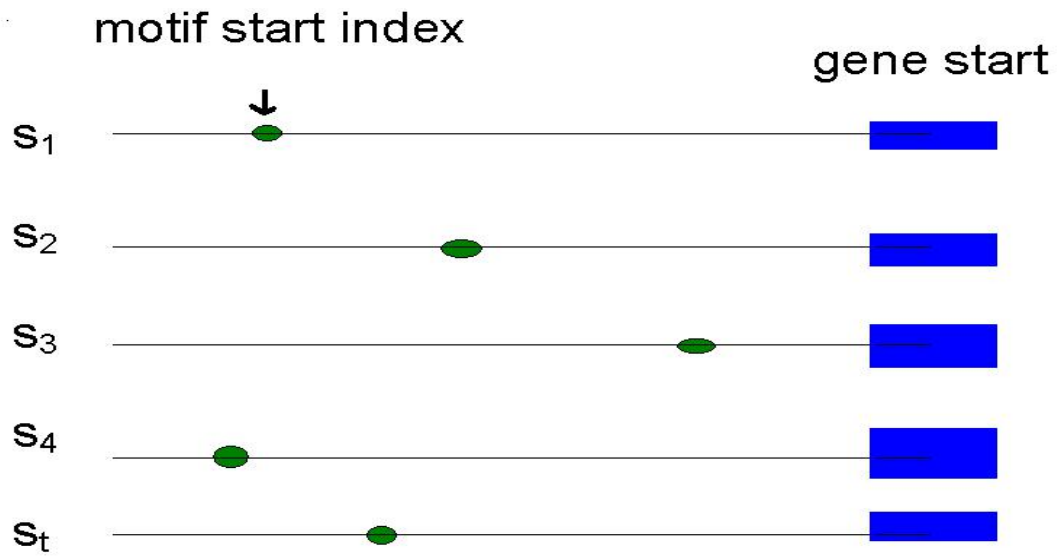
- ChIP-chip or ChIP-Seq: is a technique that combines chromatin immunoprecipitation ("ChIP") with microarray (or sequencing) technology



http://en.wikipedia.org/wiki/ChIP-chip

Motif Finding

- Find promoter motifs associated with **co-regulated** or **functionally related** genes



Genes regulated by same transcription factor

Input Sequences

- ChIP-chip experiment.
- Promoter sequences from a cluster of microarray data (or functional related genes)
- Conserved noncoding sequences among different species.

Essential Tasks

- Modeling motifs
- Visualization motifs
- Finding motif

Consensus

Nature Biotechnology 24, 423 - 425 (2006)

Probabilistic Model

- Positional weighted matrix (PWM)
  - L x 4 matrix, where L is the length of the motif
  - Each position is a probability distribution (p(A), p(C), p(G), P(T))
  - Independence between different position

PWM

HEM13 CCCATT

HEM13 TTTCTG

HEM13 TCAATT

ANB1 CTCATT

ANB1 TCCATT

ANB1 CCTATT

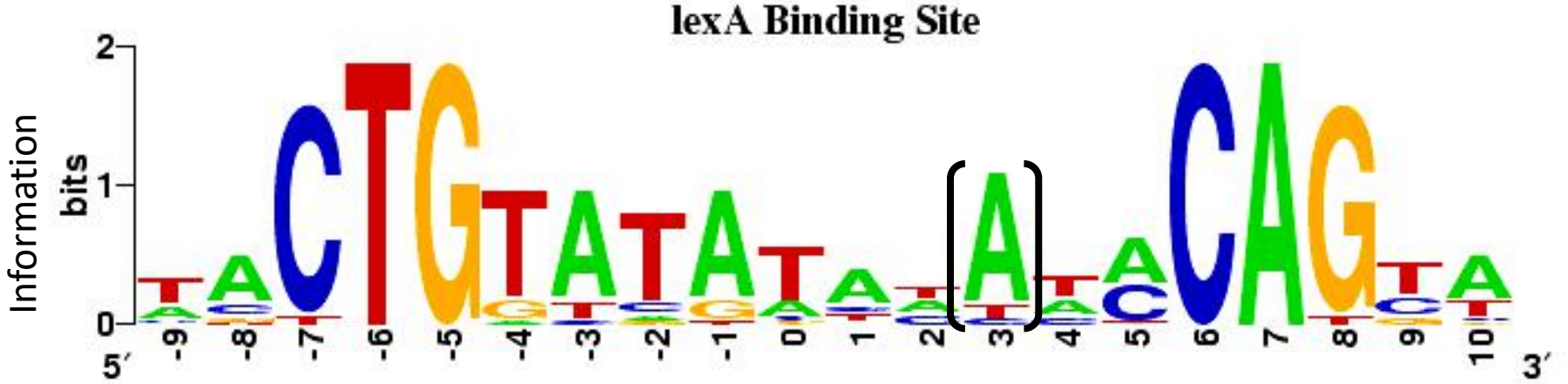
ANB1 TCCATT

ROX1 CCAATT

	1	2	3	4	5	6
A	0	0	0.25	0.875	0	0
C	0.5	0.75	0.5	0.125	0	0.125
G	0	0	0	0	0	0.875
T	0.5	0.25	0.25	0	1.0	0

Motif Information

The height of a stack is often called the motif information at that position measured in bits



$$\text{Motif Position Information} = 2 - \sum_{b=\{A,T,G,C\}} -p_b \log\, p_b$$

Why is this a measure of information?

随机事件的信息量 (I)

- 如果说“明天的太阳会从东边升起”，你会觉得这是一句废话，因为没有得到任何信息。
- 反过来，如果说“明天会发生日食”，你会觉得很吃惊，感觉到得到了很多信息。
- 因此，信息量的多少与随机事件发生的概率有关，是概率的函数 f(p).

随机事件的信息量 (II)

- 相互独立的两个随机事件同时发生引起的信息量是分别引起的信息量之和。

$$f(pq) = f(p) + f(q)$$

- 什么函数具有上述性质？可以证明，唯有对数函数具有上述性质。

$$I(p) = -\log_2(p)$$

随机分布的信息量

- 定义为每个可能的随机事件的平均信息量。

- 若离散分布S有n个取值，p_i是相应取值的概率。则分布S的熵定义为

$$\mathrm{Inf}(S) = -\sum_{i=1}^{n} p_i \log_2(p_i)$$

Entropy

Entropy measures average uncertainty

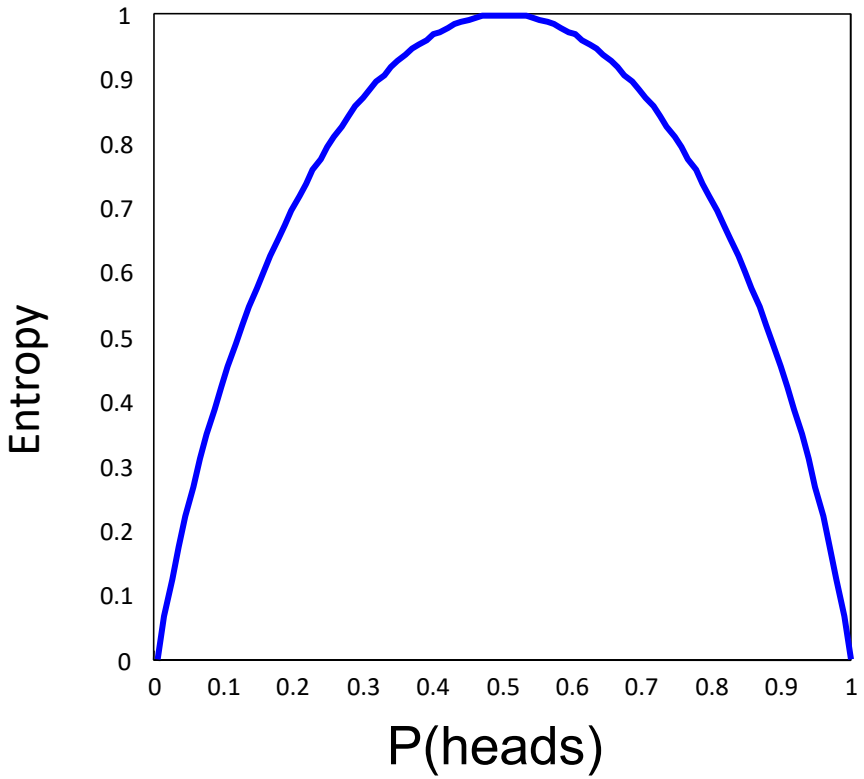
Entropy measures randomness

$$H(X) = -\sum_i p_i \log_2 p_i$$

If log is base 2, then the units are called bits

Entropy versus Randomness

Entropy is maximum at maximum randomness

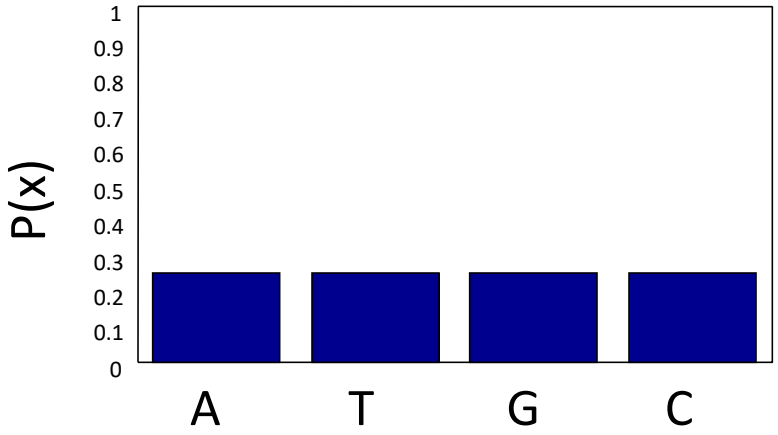


Example: Coin Toss

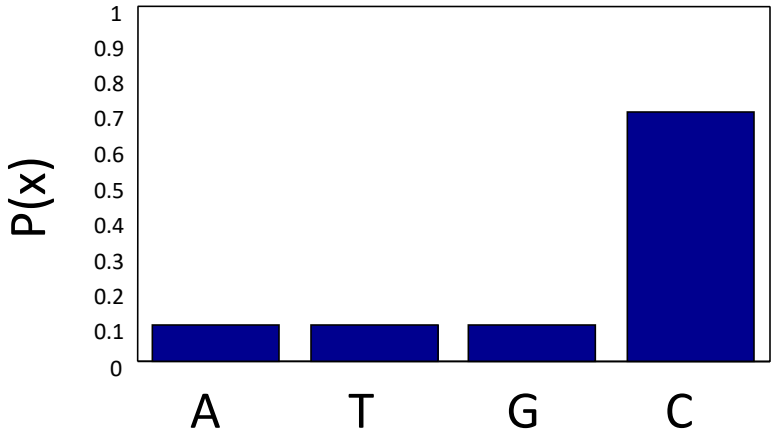
P(heads)=0.1 Not very random
H(X)=0.47 bits

P(heads)=0.5 Completely random
H(X)=1 bits

Entropy Examples



$$H(X) = -[4 * 0.25 \log_2(0.25)]$$
$$= 2(bit)$$



$$H(X) = -[3 * 0.1 \log_2(0.1)$$
$$+ 0.7 \log_2(0.7)]$$
$$= 0.63(bit)$$

Motif Information

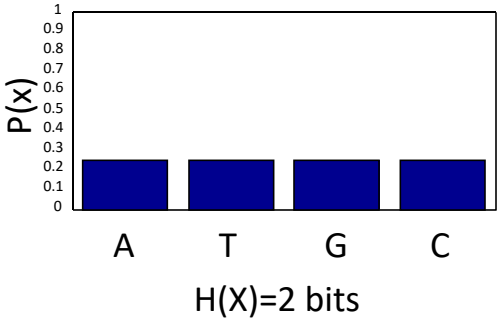
Motif Position Information = $2.0 - \sum_{b=\{A,C,G,T\}} p_b \log_2 p_b$

$H_{background}(X)$

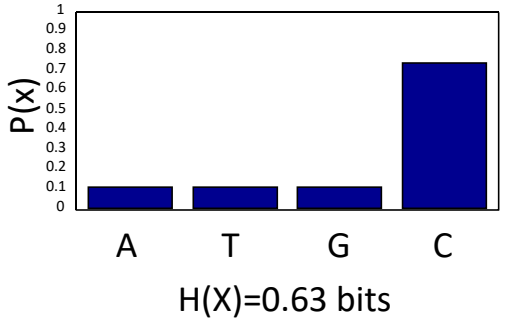
$H_{motif\_i}(X)$

Prior uncertainty about nucleotide

Uncertainty after learning it is position i in a motif



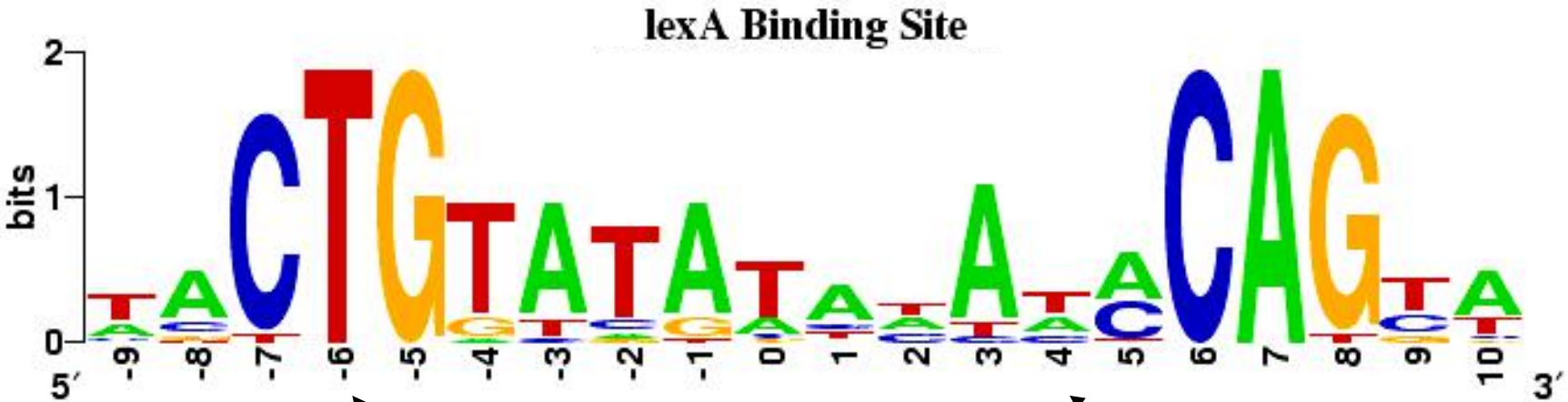
H(X)=2 bits



H(X)=0.63 bits

Uncertainty at this position has been reduced by 1.37 bits

Motif Logo



Conserved Residue
Reduction of uncertainty
of 2 bits

Little Conservation
Minimal reduction of
uncertainty

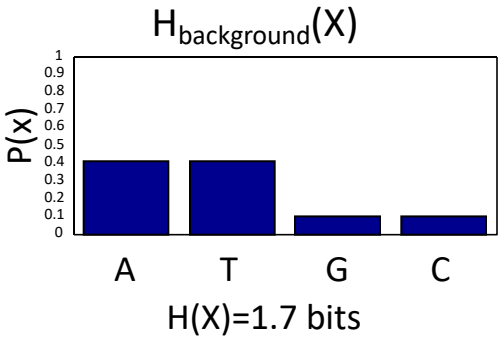
http://weblogo.berkeley.edu/

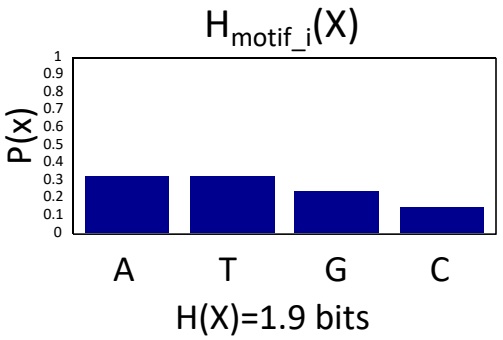
Background DNA Frequency

The definition of information assumes a uniform background DNA nucleotide frequency

What if the background frequency is not uniform?

(e.g. Plasmodium)





Motif Position Information = $1.7 - \sum_{b=\{A,C,G,T\}} p_b \log p_b = 0.2(\text{bit})$

Some motifs could have negative information!

A Different Measure

- Relative entropy or Kullback-Leibler distance (divergence)

$$D_{KL}(P_{motif}||P_{bg}) = \sum_{b=\{A,C,G,T\}} P_{motif}(b) \log \frac{P_{motif}(b)}{P_{bg}(b)}$$

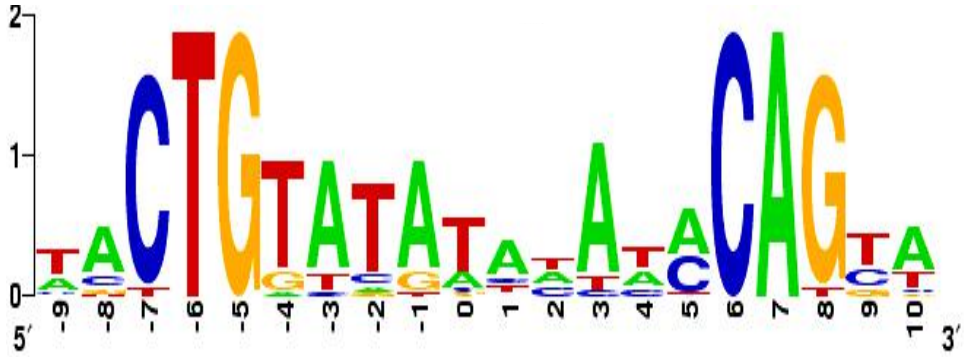
- Property

$$D_{KL} \geq 0$$

$$D_{KL} = 0 \Leftrightarrow P_{motif} = P_{bg}$$

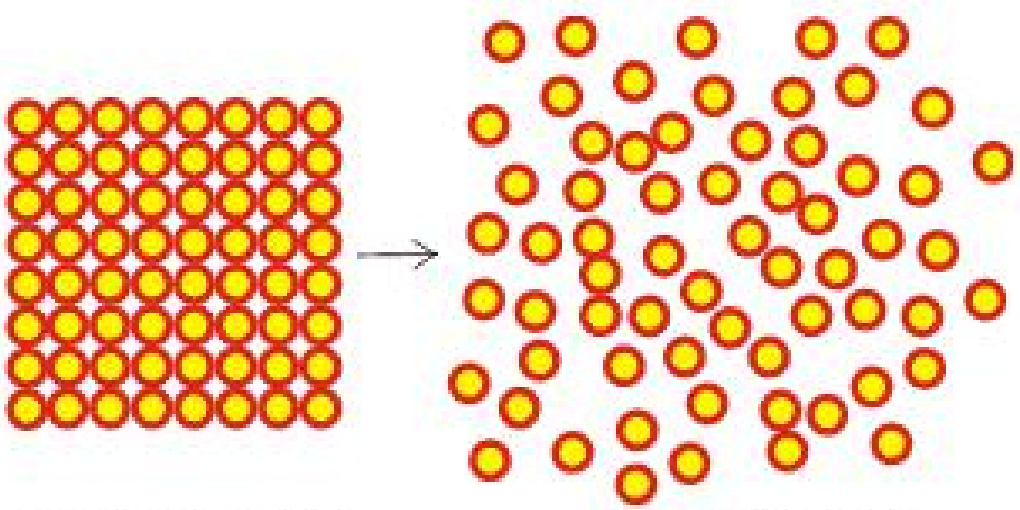
Comparing Both Methods

Information assuming uniform background DNA



KL Distance assuming 20% GC content (e.g. Plasmodium)

Entropy Examples



AAAAAAAA

Bucket 1

Low Entropy

AAAABBCD

Bucket 2

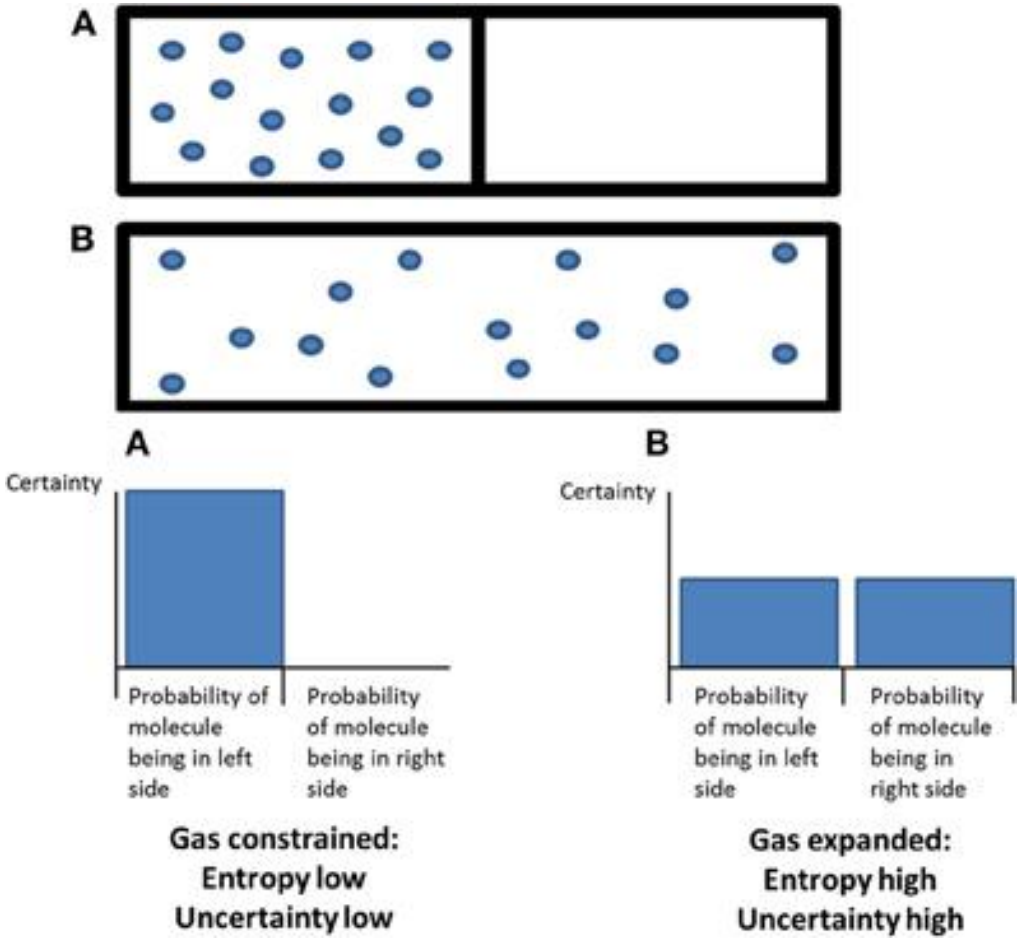
Medium Entropy

AABBCCDD

Bucket 3

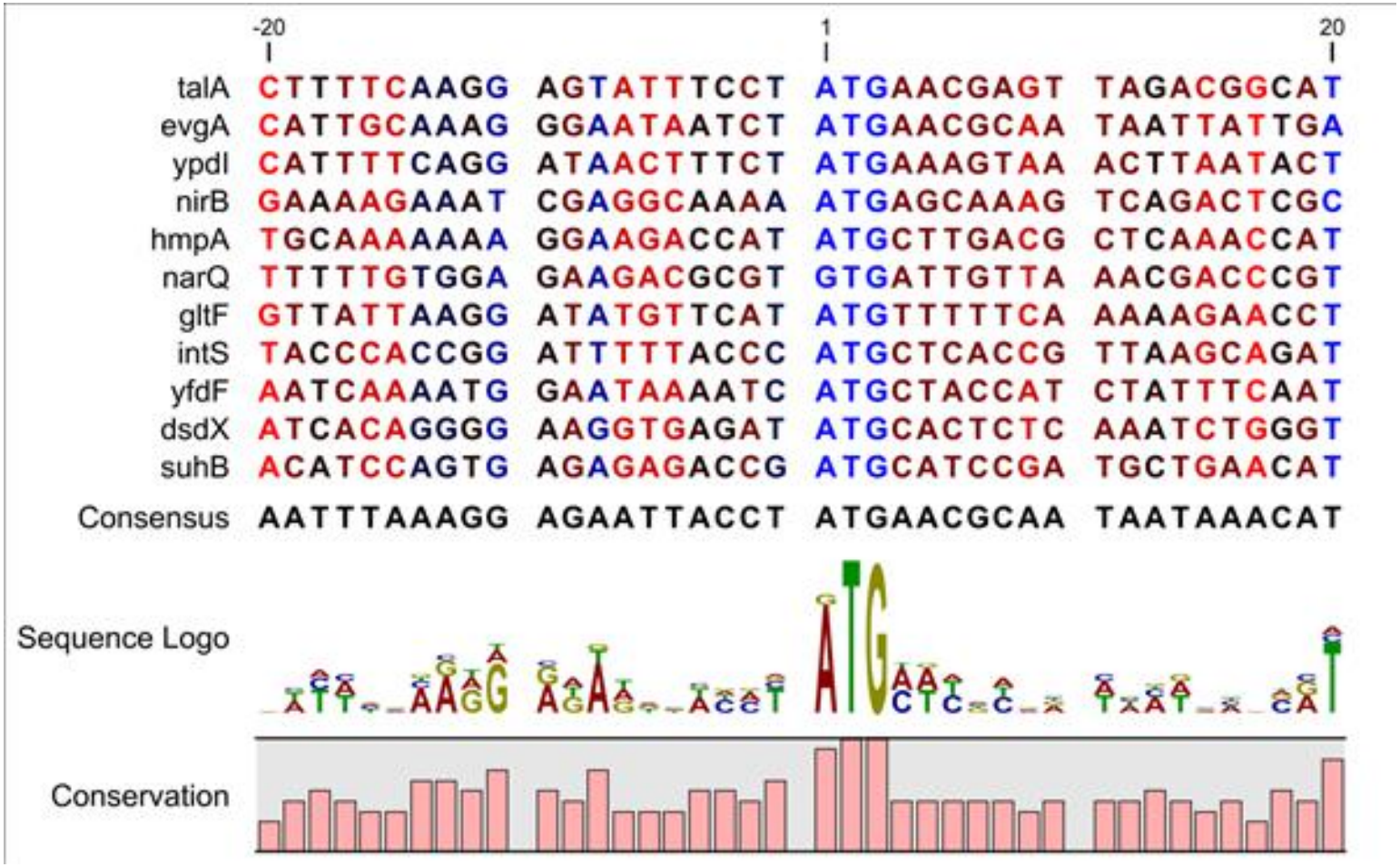
High Entropy

Entropy Examples

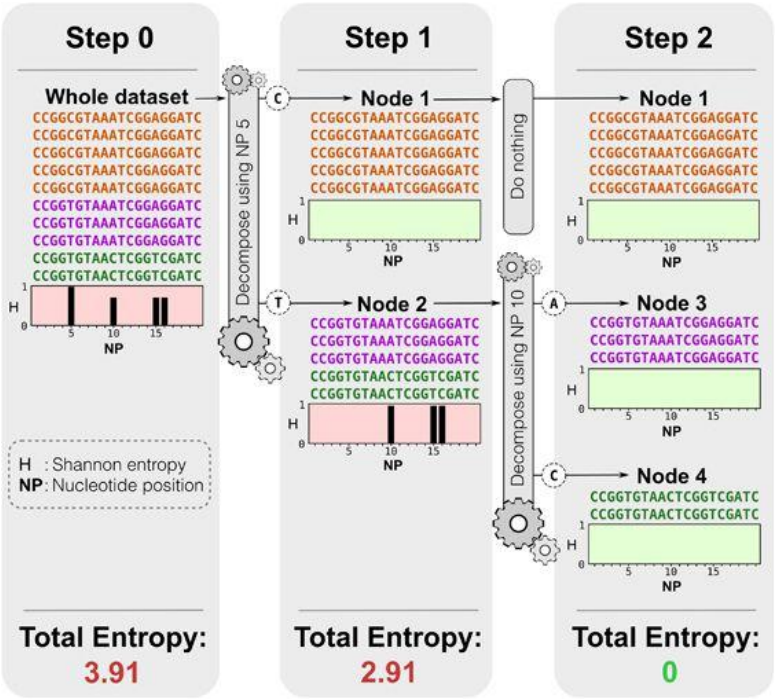


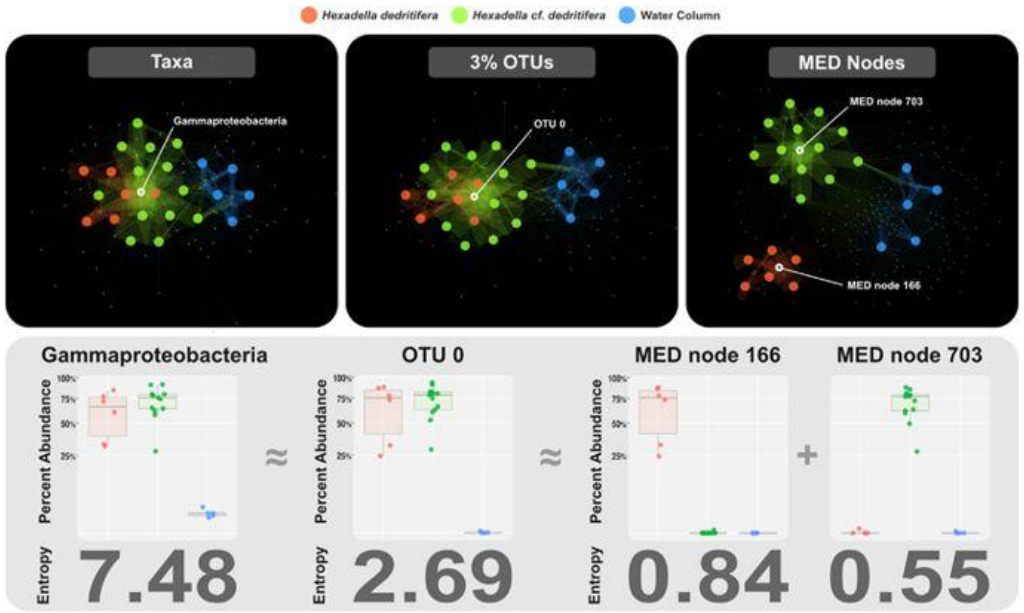
High entropy --> information --> high uncertainty --> high probability

Entropy Examples



Entropy Examples





Reference: https://www.nature.com/articles/ismej2014195

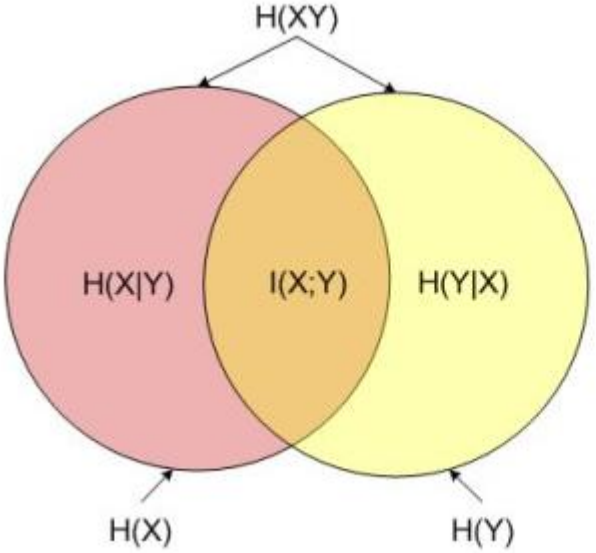
Mutual information (互信息)

$$I(A,B) = \sum_{a,b} p_{AB}(a,b) \log \frac{p_{AB}(a,b)}{p_A(a).p_B(b)}$$

互信息又可以等价地表示成

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

其中H(X)和H(Y) 是边缘熵，H(X|Y)和H(Y|X)是条件熵，而H(X,Y)是X和Y的联合熵。



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

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

I(X;Y)=H(X)+H(Y)-H(XY)

Finding New Motifs

Learning Motif Models

Motif Finding Problem

- Given a set of sequences, find the motif shared by all or most sequences, while its starting position in each sequence is unknown
- Assumption:
  - Each motif appears exactly once in one sequence
  - The motif has fixed length

Motif Finding

- Given the missing data, it's a multinomial distribution

$$\Pr(X_i \mid Z_{ij} = 1, p) = \underbrace{\prod_{k=1}^{j-1} p_{x_{ik},0}}_{\text{before motif}} \underbrace{\prod_{k=j}^{j+W-1} p_{x_{ik},k-j+1}}_{\text{motif}} \underbrace{\prod_{k=j+W}^{L} p_{x_{ik},0}}_{\text{after motif}}$$

X_i is the *i*th sequence

Z_{ij} is 1 if motif starts at position *j* in sequence *i*

Example

- Finding motif (length 3) in following sequences

A C A G C A

A G G C A G

T C A G T C

EM Updating

$$p_{A,1} = \frac{z_{11} + z_{13} + z_{21} + z_{33}}{z_{11} + z_{12} + z_{13} + z_{14} + \cdots + z_{31} + z_{32} + z_{33} + z_{34}}$$

$$p_{C,1} = \frac{z_{12} + z_{24} + z_{32}}{z_{11} + z_{12} + z_{13} + z_{14} + \cdots + z_{31} + z_{32} + z_{33} + z_{34}}$$

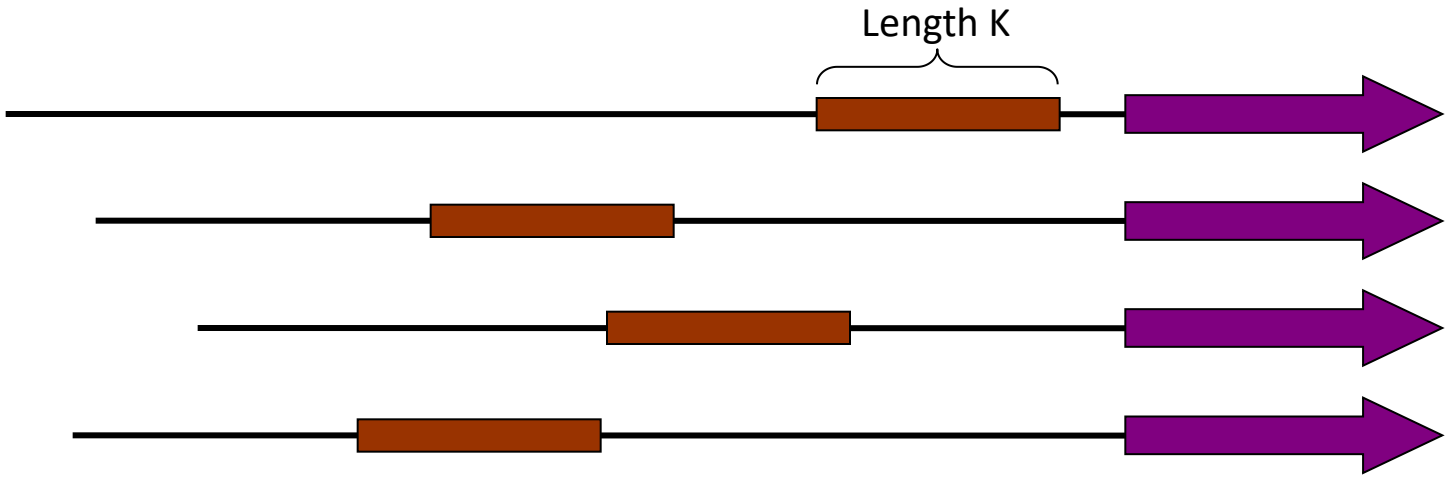
$$p_{G,1} = \frac{z_{14} + z_{22} + z_{23} + z_{32}}{z_{11} + z_{12} + z_{13} + z_{14} + \cdots + z_{31} + z_{32} + z_{33} + z_{34}}$$

$$p_{T,1} = \frac{z_{31}}{z_{11} + z_{12} + z_{13} + z_{14} + \cdots + z_{31} + z_{32} + z_{33} + z_{34}}$$

Generative Model

- Suppose the sequences are aligned, the aligned regions are generated from a motif model
- Motif model is a PWM. A PWM is a position-specific multinomial distribution.
  - For each position i, a multinomial distribution on (A,C,G,T): q_{iA},q_{iC},q_{iG},q_{iT}
- The unaligned regions are generated from a background model: p_A,p_C,p_G,p_T

A Promoter Model



Motif

Background DNA

	M_1					M_K
A	.1	.2	.1	.4	.1	.3
C	.2	.2	.2	.2	.5	.4
G	.4	.5	.4	.2	.2	.2
T	.3	.1	.2	.2	.2	.1

$P_k(S|M)$

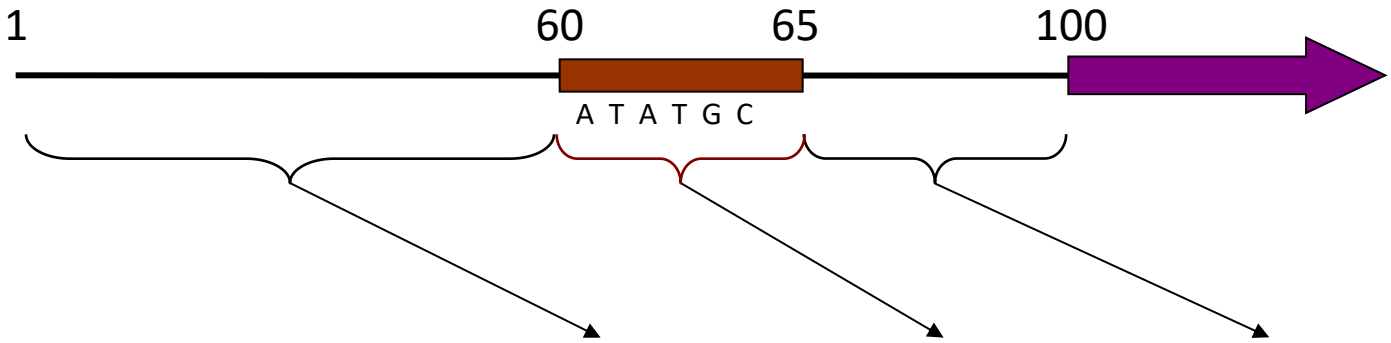
A: 0.25
T: 0.25
G: 0.25
C: 0.25

$P(S|B)$

The same motif model in all promoters

Probability of a Sequence

Given a sequence(s), motif *model* and motif *location*



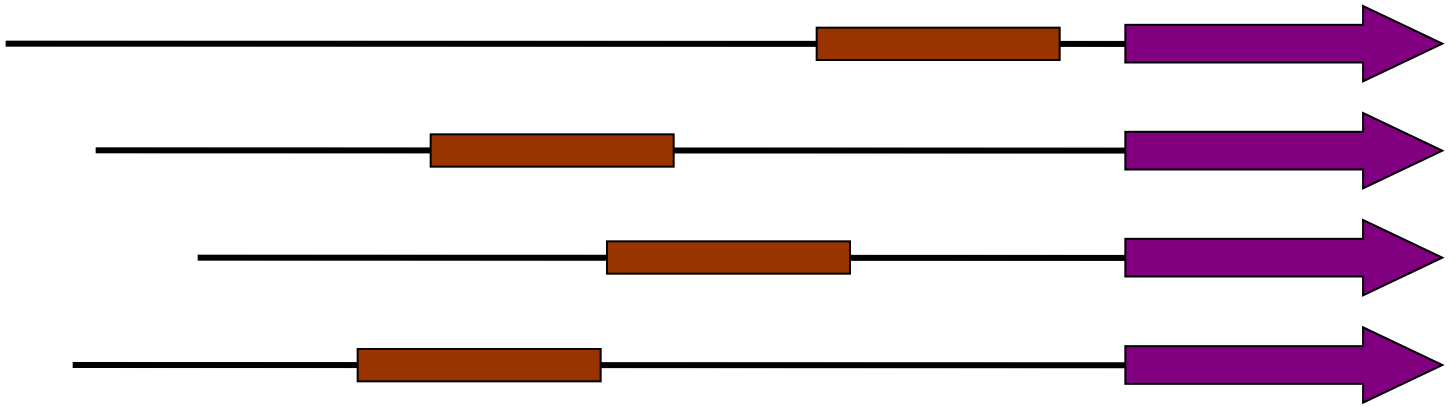
$$P(Seq \mid Mstart = 10, Model) = \prod_{i=1}^{59} P(S_i \mid B) \prod_{k=1}^{6} P_k\left(S_{k+63} \mid M\right) \prod_{i=66}^{100} P(S_i \mid B)$$

S_i = *nucleotide at position i in the sequence*

	M_1					M_K
A	.1	.2	.1	.4	.1	.3
C	.2	.2	.2	.2	.5	.4
G	.4	.5	.4	.2	.2	.2
T	.3	.1	.2	.2	.2	.1

Parameterizing the Motif Model

Given multiple sequences and motif locations but no motif *model*



AATGCG
ATATGG
ATATCG
GATGCA

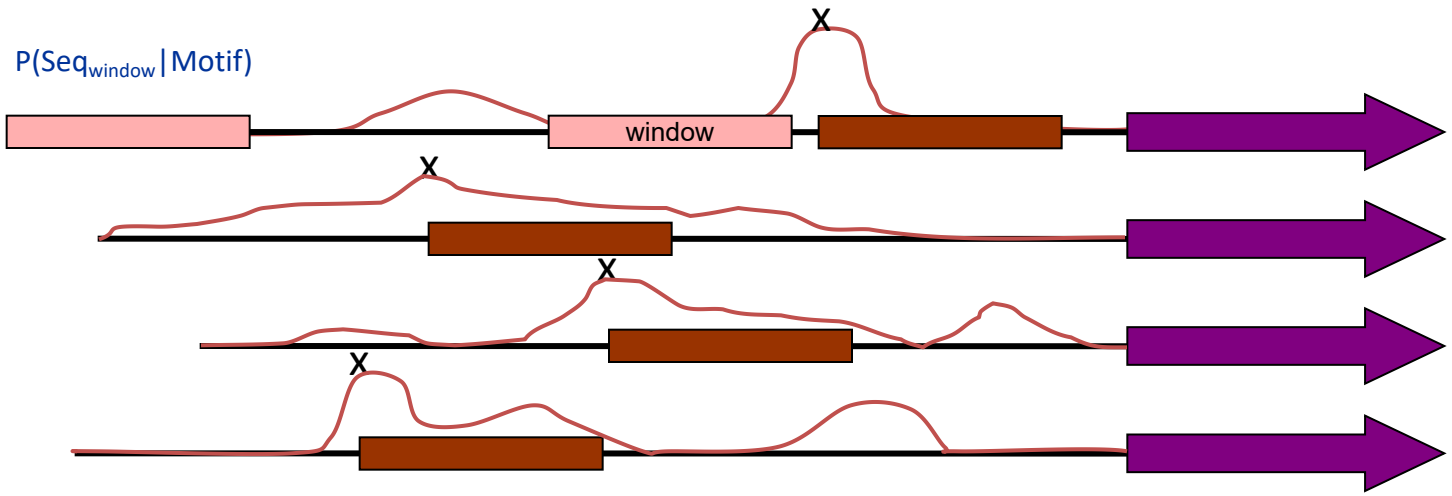
Count Frequencies

Add pseudocounts

	M_1					M_6
A	3/4	1/2	1/2			1/4
C					3/4	
G	1/4			1/2	1/4	3/4
T		1/2	1/2	1/2		

Finding Known Motifs

Given multiple sequences and motif model but no motif *locations*

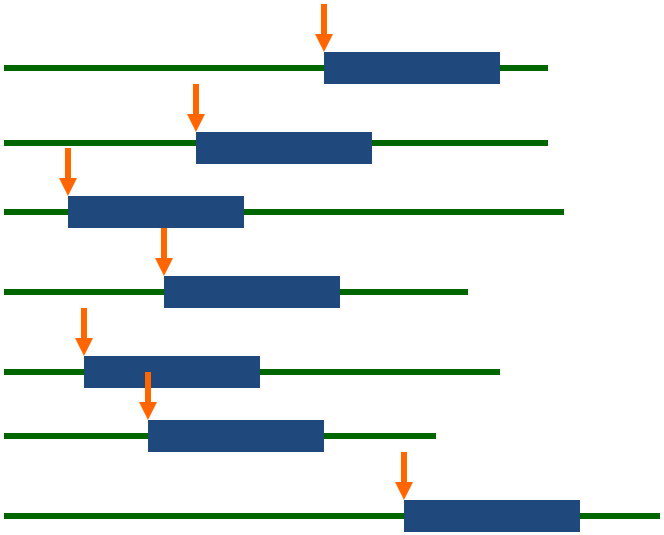


Calculate P(Seq$_{window}$|Motif) for every starting location

Choose best starting location in each sequence

The EM Approach

- EM is a family of algorithms for learning probabilistic models in problems that involve *hidden state*
- in our problem, the hidden state is where the motif starts in each training sequence



The MEME Algorithm

- Bailey & Elkan, 1993
- uses EM algorithm to find multiple motifs in a set of sequences
- first EM approach to motif discovery: Lawrence & Reilly 1990

EM Algorithm for Motif Discovery

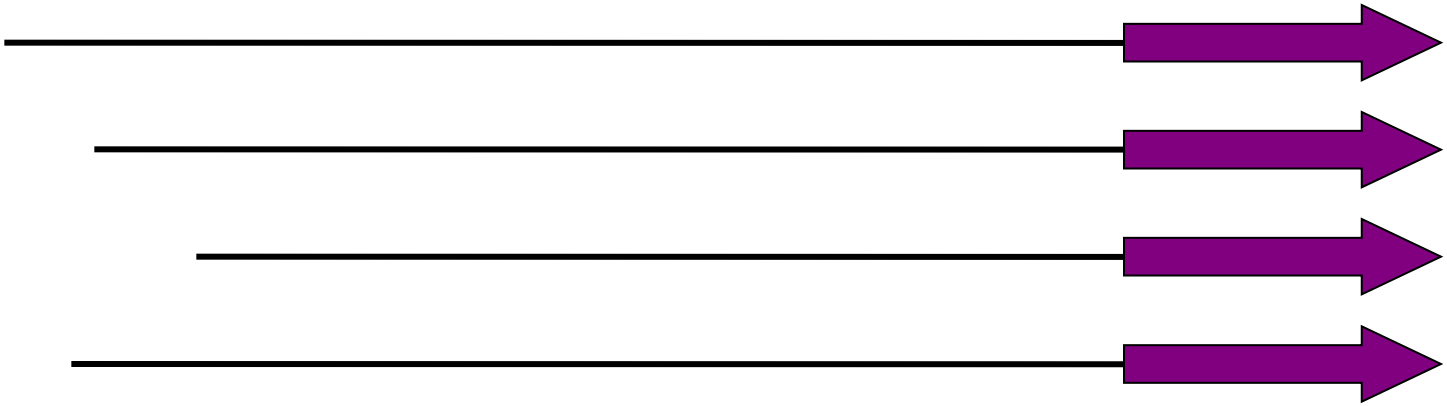
1. Start with random motif model
2. E Step: estimate probability of motif positions for each sequence
3. M Step: use estimate to update motif model
4. Iterate (to convergence)

At each iteration, P(Sequences|Model) guaranteed to increase

Demo: Initialization

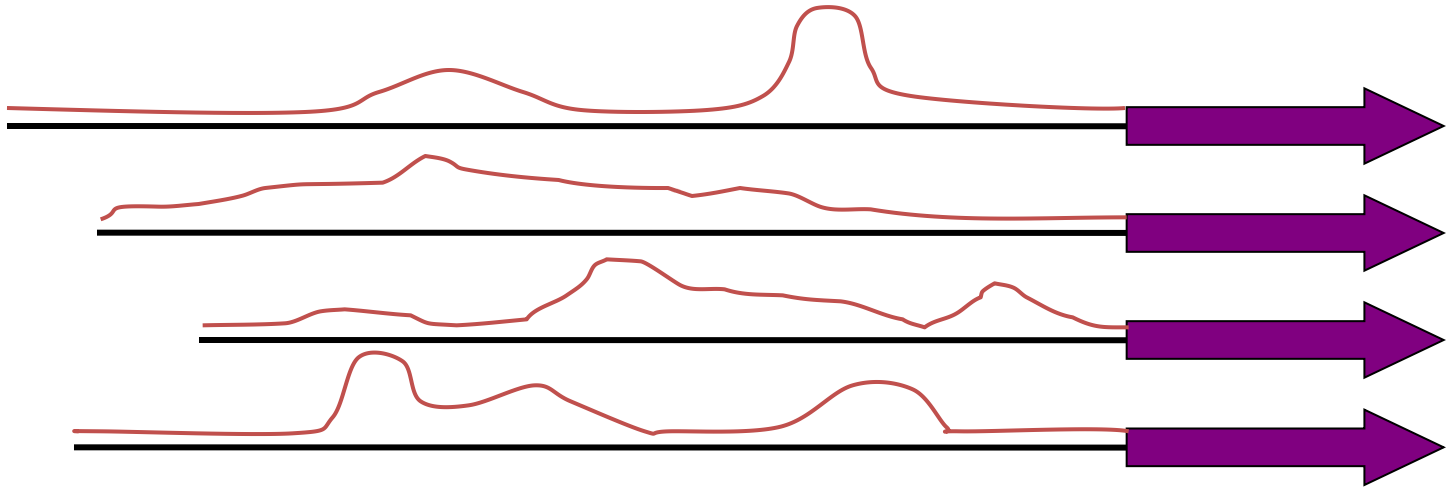
- Given a random motif model

A	0.1	0.2	0.1	0.4	0.1	0.3
C	0.2	0.2	0.2	0.2	0.5	0.4
G	0.4	0.5	0.4	0.2	0.2	0.2
T	0.3	0.1	0.2	0.2	0.2	0.1



Demo: E-Step

- E Step: estimate probability of motif positions for each sequence



Demo: M-Step

- M Step: use estimate to update motif model

A	0.1	0.1	0.1	0.1	0.1	0.3
C	0.2	0.3	0.2	0.2	0.5	0.1
G	0.4	0.5	0.4	0.5	0.2	0.1
T	0.3	0.1	0.2	0.2	0.2	0.1

Basic EM Approach

- we'll need to calculate the probability of a training sequence given a hypothesized starting position:

$$\Pr(X_i \mid Z_{ij} = 1, p) = \underbrace{\prod_{k=1}^{j-1} p_{c_k,0}}_{\text{before motif}} \underbrace{\prod_{k=j}^{j+W-1} p_{c_k,k-j+1}}_{\text{motif}} \underbrace{\prod_{k=j+W}^{L} p_{c_k,0}}_{\text{after motif}}$$

X_i is the *i*th sequence

Z_{ij} is 1 if motif starts at position *j* in sequence *i*

c_k is the character at position *k* in sequence *i*

Example

$X_i =$ **G C T G T A G**

$p =$

	0	1	2	3
A	0.25	0.1	0.5	0.2
C	0.25	0.4	0.2	0.1
G	0.25	0.3	0.1	0.6
T	0.25	0.2	0.2	0.1

$$\Pr(X_i \mid Z_{i3} = 1, p) =$$

$$p_{\mathrm{G},0} \times p_{\mathrm{C},0} \times p_{\mathrm{T},1} \times p_{\mathrm{G},2} \times p_{\mathrm{T},3} \times p_{\mathrm{A},0} \times p_{\mathrm{G},0} =$$

$$0.25 \times 0.25 \times 0.2 \times 0.1 \times 0.1 \times 0.25 \times 0.25$$

The E-step: Estimating *Z*

- To estimate the starting positions in *Z* at step *t*

$$Z_{ij}^{(t)} = \frac{\Pr(X_i \mid Z_{ij} = 1, p^{(t)}) \Pr(Z_{ij} = 1)}{\sum_{k=1}^{L-W+1} \Pr(X_i \mid Z_{ik} = 1, p^{(t)}) \Pr(Z_{ik} = 1)}$$

- This comes from Bayes' rule applied to

$$\Pr(Z_{ij} = 1 \mid X_i, p^{(t)})$$

The E-step: Estimating Z

- Assume that it is equally likely that the motif will start in any position

$$Z_{ij}^{(t)} = \frac{\Pr(X_i \mid Z_{ij} = 1, p^{(t)}) \Pr(Z_{ij} = 1)}{\sum_{k=1}^{L-W+1} \Pr(X_i \mid Z_{ik} = 1, p^{(t)}) \Pr(Z_{ik} = 1)}$$

Example: Estimating *Z*

$X_i =$ **G C T G T A G**

$p =$

	0	1	2	3
A	0.25	0.1	0.5	0.2
C	0.25	0.4	0.2	0.1
G	0.25	0.3	0.1	0.6
T	0.25	0.2	0.2	0.1

$$Z_{i1} = 0.3 \times 0.2 \times 0.1 \times 0.25 \times 0.25 \times 0.25 \times 0.25$$

$$Z_{i2} = 0.25 \times 0.4 \times 0.2 \times 0.6 \times 0.25 \times 0.25 \times 0.25$$

$$\vdots$$

- Then normalize so that $\sum_{j=1}^{L-W+1} Z_{ij} = 1$

The M-step: Estimating *p*

- recall $p_{c,k}$ represents the probability of character *c* in position *k* ; values for position 0 represent the background

$$p_{c,k}^{(t+1)} = \frac{n_{c,k} + d_{c,k}}{\sum_b (n_{b,k} + d_{b,k})}$$

pseudo-counts

$$n_{c,k} = \begin{cases} \sum_i \sum_{\{j | X_{i,j+k-1} = c\}} Z_{ij} & k > 0 \\ n_c - \sum_{j=1}^{W} n_{c,j} & k = 0 \end{cases}$$

total # of c's in data set

Example: Estimating *p*

A C A G C A

$$Z_{1,1} = 0.1, Z_{1,2} = 0.7, Z_{1,3} = 0.1, Z_{1,4} = 0.1$$

A G G C A G

$$Z_{2,1} = 0.4, Z_{2,2} = 0.1, Z_{2,3} = 0.1, Z_{2,4} = 0.4$$

T C A G T C

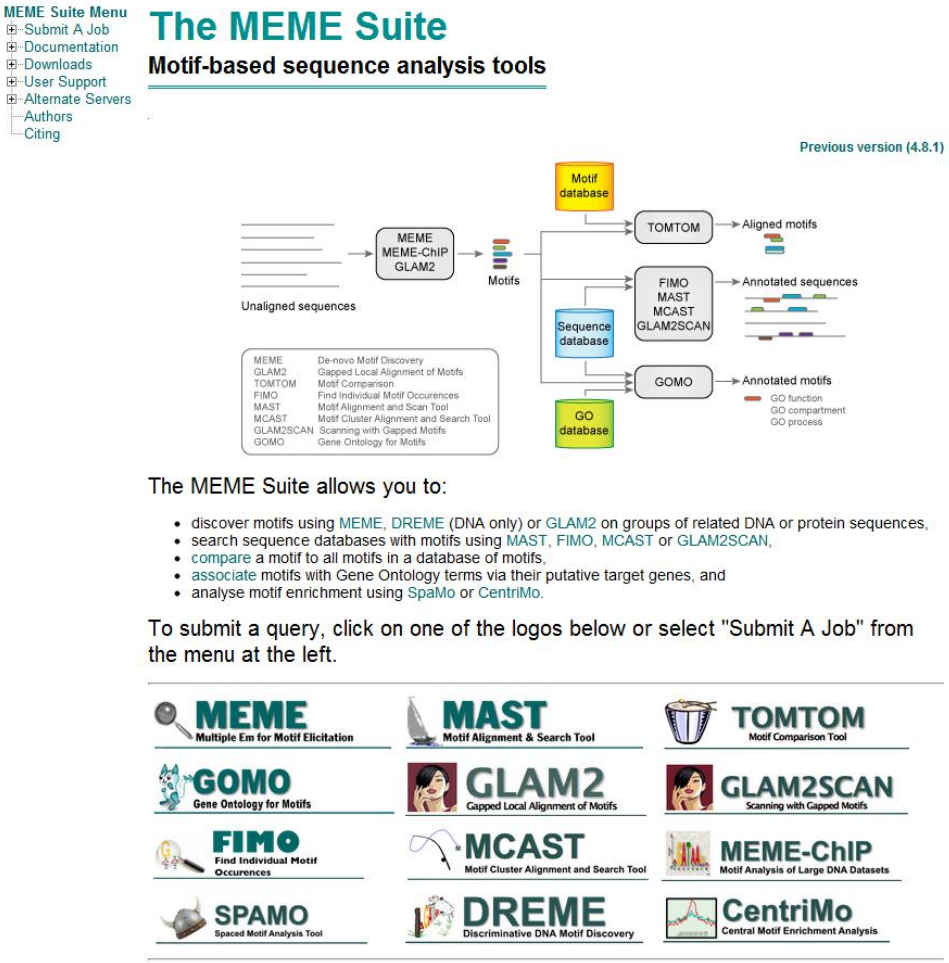
$$Z_{3,1} = 0.2, Z_{3,2} = 0.6, Z_{3,3} = 0.1, Z_{3,4} = 0.1$$

$$p_{\mathrm{A},1} = \frac{Z_{1,1} + Z_{1,3} + Z_{2,1} + Z_{3,3} + 1}{Z_{1,1} + Z_{1,2} \ldots + Z_{3,3} + Z_{3,4} + 4}$$

MEME

- MEME - implements EM for motif discovery in DNA and proteins
- MAST – search sequences for motifs given a model
- References

1. Timothy L. Bailey, Mikael Bodén, Fabian A. Buske, Martin Frith, Charles E. Grant, Luca Clementi, Jingyuan Ren, Wilfred W. Li, William S. Noble, "MEME SUITE: tools for motif discovery and searching", Nucleic Acids Research, 37:W202-W208, 2009.



http://meme.sdsc.edu/meme/

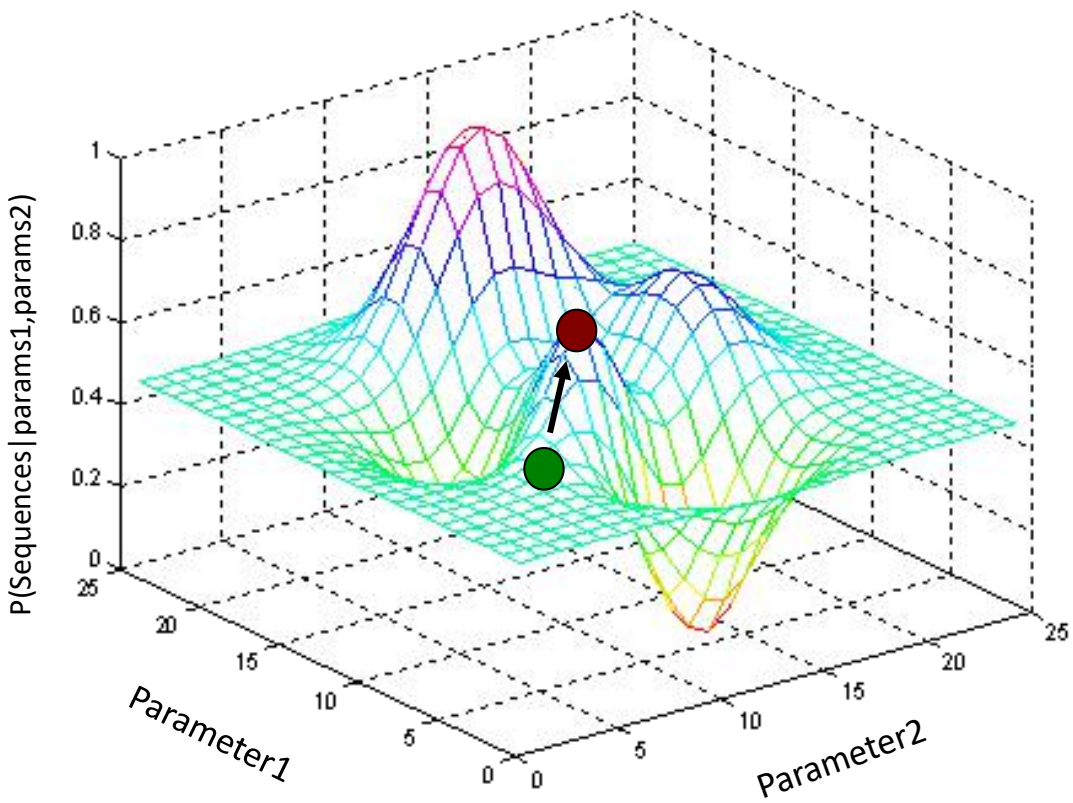
P(Seq|Model) Landscape

EM searches for parameters to increase P(seqs|parameters)

Useful to think of P(seqs|parameters) as a function of parameters

EM starts at an initial set of parameters

And then "climbs uphill" until it reaches a local maximum



Where EM starts can make a big difference

Search from Many Different Starts

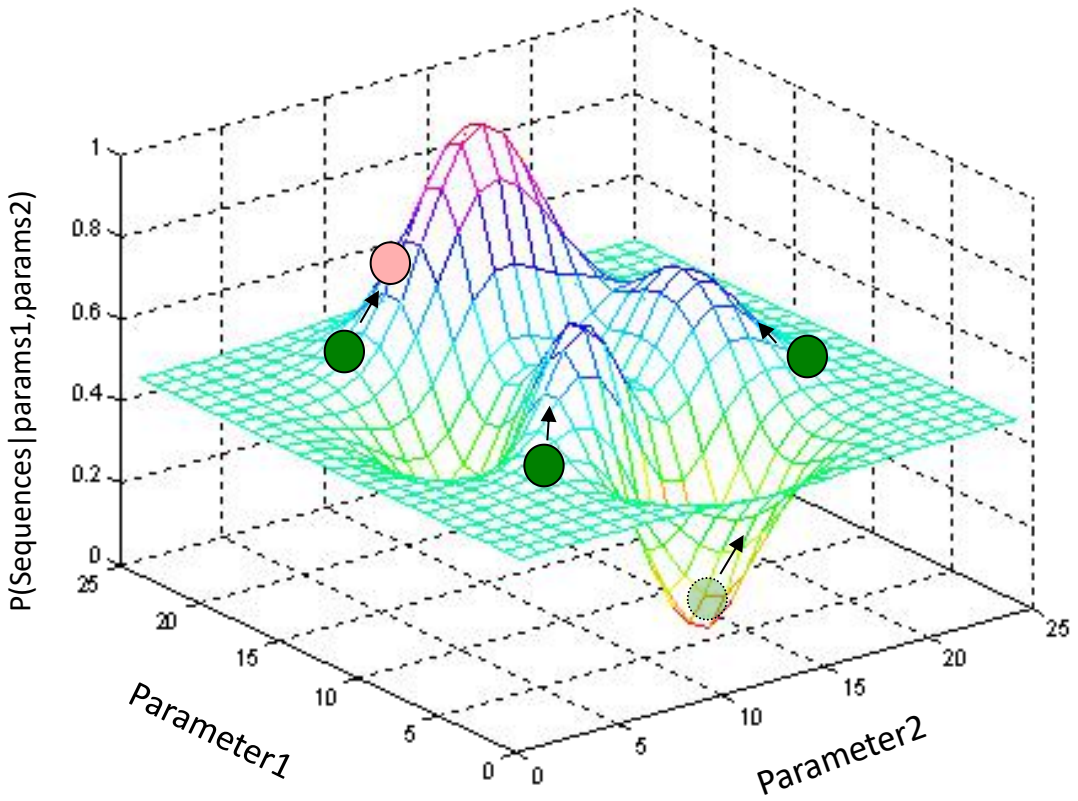
To minimize the effects of local maxima, you should search multiple times from different starting points

MEME uses this idea

Start at many points

Run for one iteration

Choose starting point that got the “highest” and continue



Gibbs Sampler

- A stochastic version of EM that differs from deterministic EM in two key ways

- At each iteration, we only update the motif position of a single sequence
- We may update a motif position to a "suboptimal" new position

Algorithm: Gibbs Sampler

1. Start with random motif locations and calculate a motif model

2. Randomly select a sequence, remove its motif and recalculate tempory model

3. With temporary model, calculate probability of motif at each position on sequence

4. Select new position based on this distribution

5. Update model and Iterate

Sampling New Motif Positions

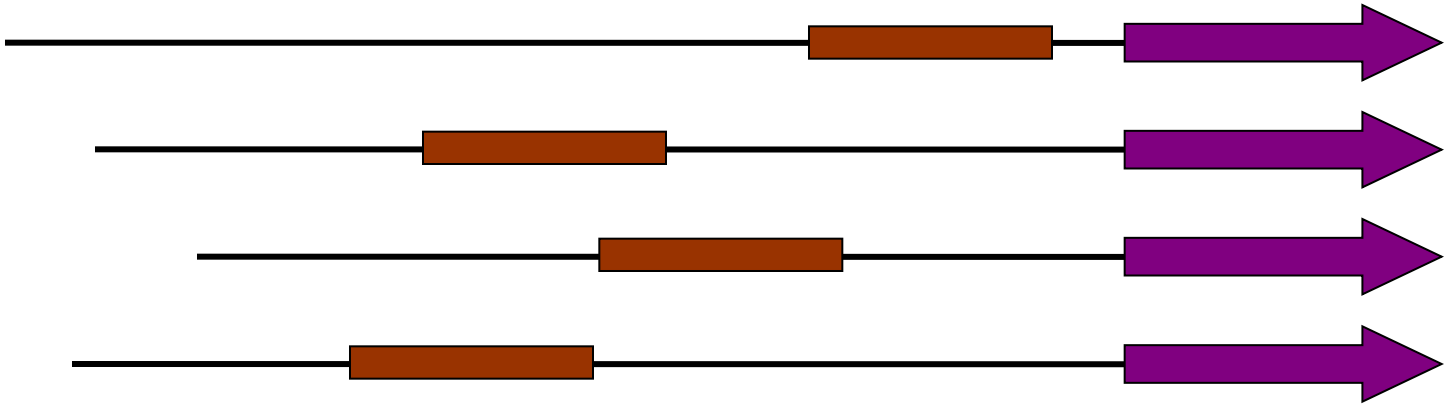
- For each possible starting position, $a_i = j$, compute a weight (likelihood ratio)

$$A_j = \frac{\prod_{k=j}^{j+W-1} p_{c_k,k-j+1}}{\prod_{k=j}^{j+W-1} p_{c_k,0}}$$

- Randomly select a new starting position a_i according to these weights

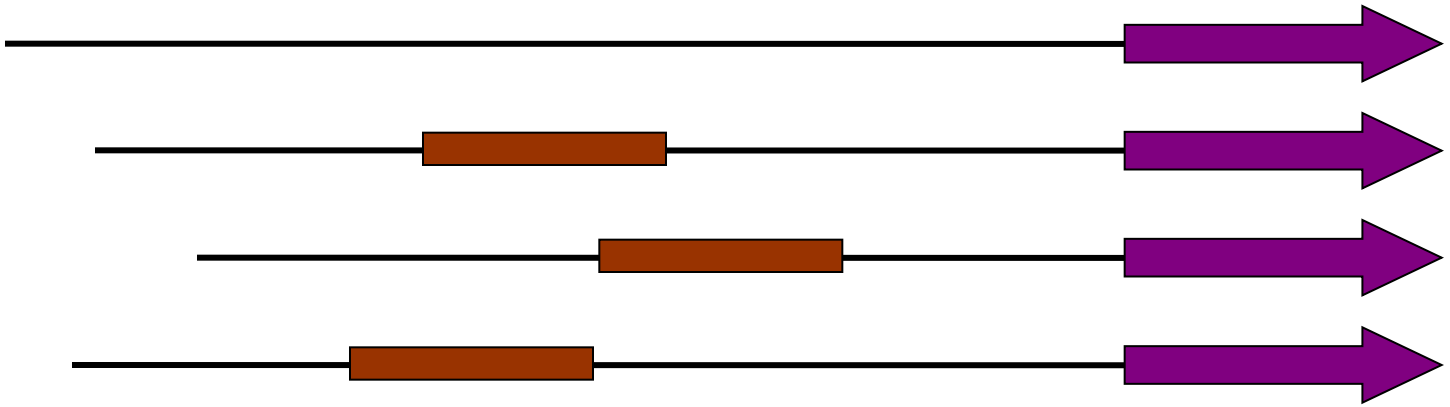
Demo: Initialization

- Random choose motif location



Demo: Step 2

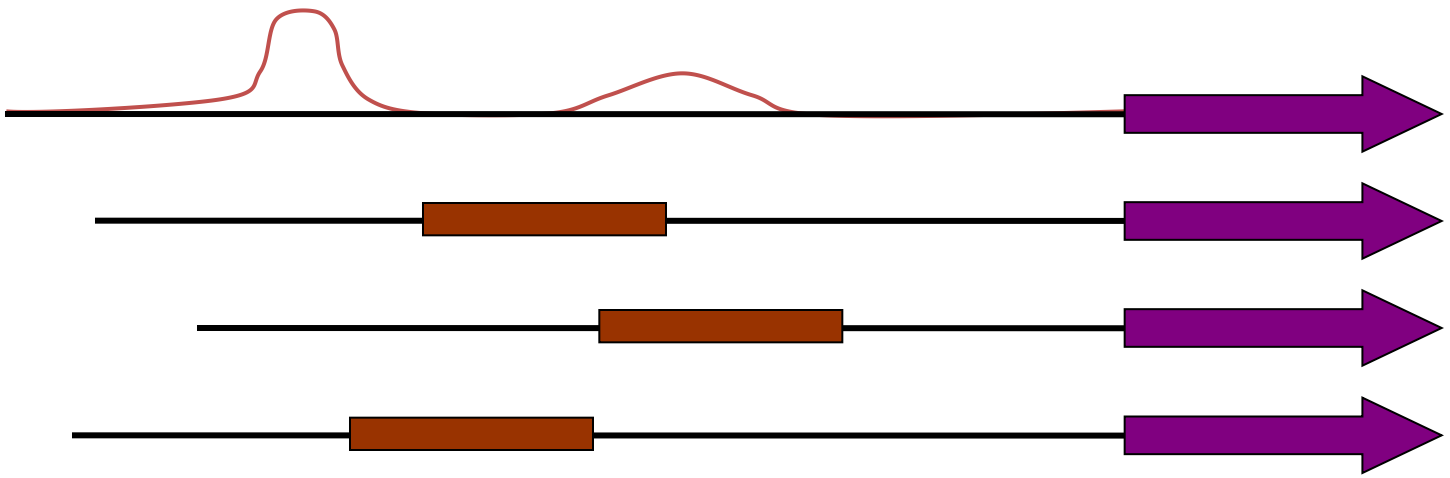
- Random Select One Sequence and Remove Its Motif. Recalculate its Temporal Model



A	0.1	0.2	0.1	0.4	0.1	0.3
C	0.2	0.2	0.2	0.2	0.5	0.4
G	0.4	0.5	0.4	0.2	0.2	0.2
T	0.3	0.1	0.2	0.2	0.2	0.1

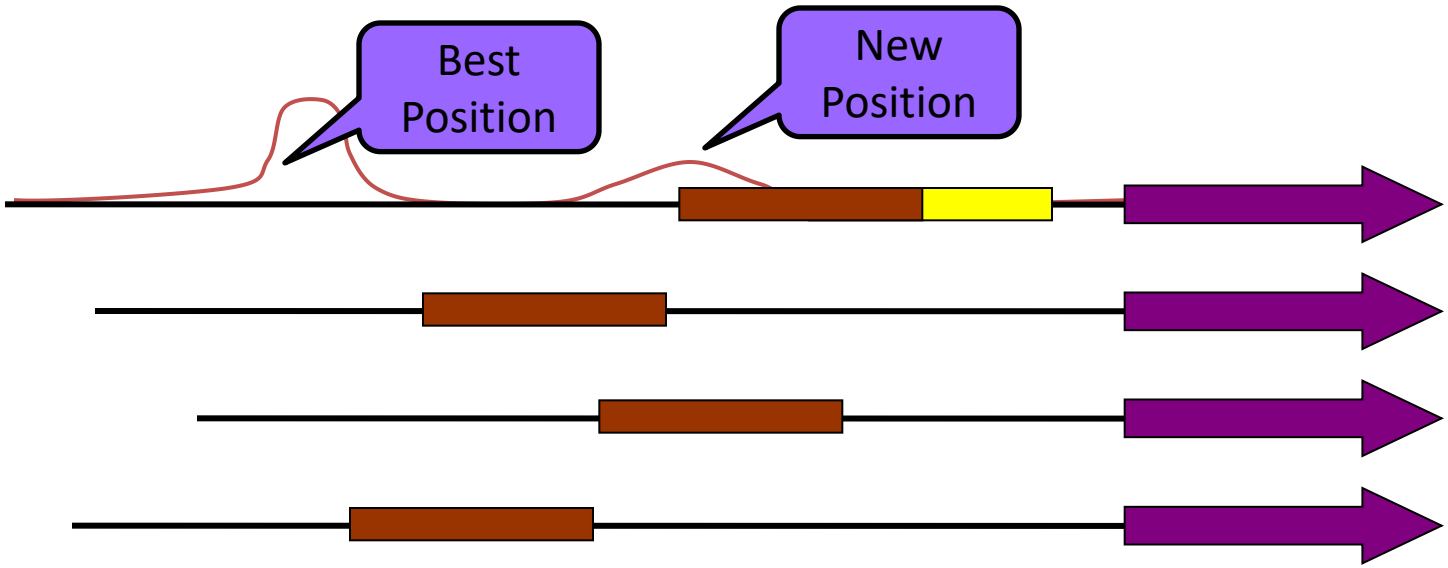
Demo: Step 3

- Calculate Probability of motif at each position on sequence



Demo: Step 4

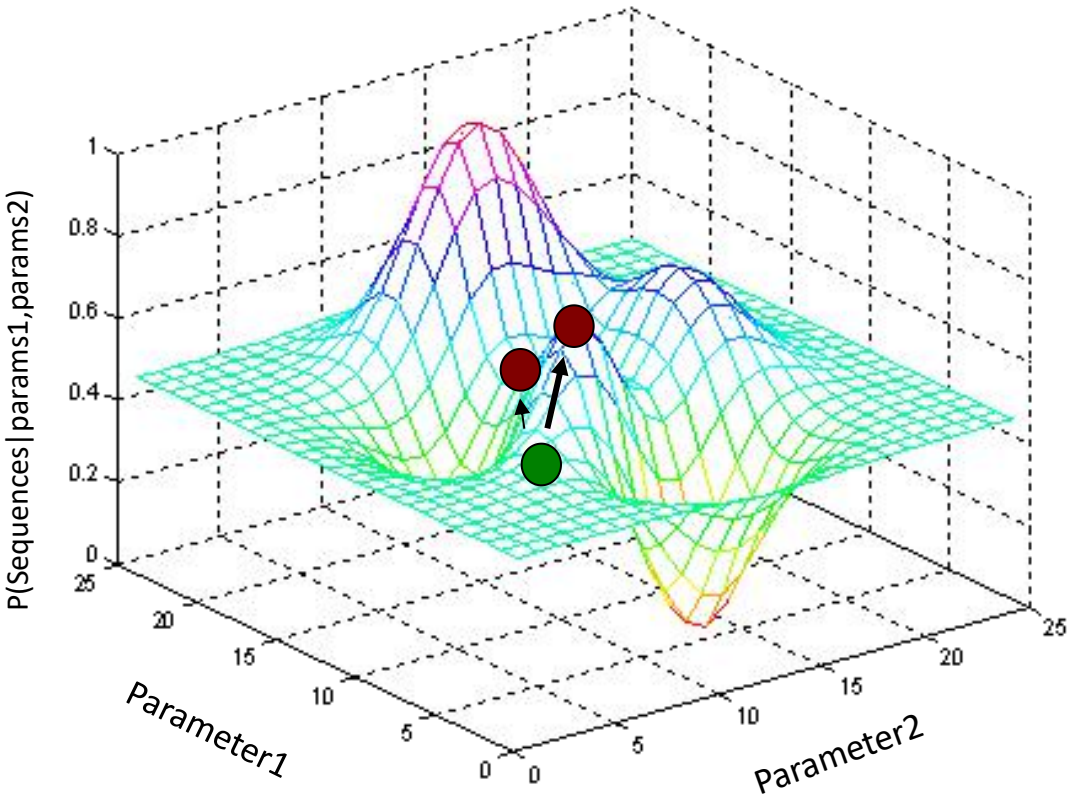
- Demo: Select New Position, Update Motif Model



A	0.1	0.1	0.1	0.1	0.1	0.3
C	0.2	0.3	0.2	0.2	0.5	0.1
G	0.4	0.5	0.4	0.5	0.2	0.1
T	0.3	0.1	0.2	0.2	0.2	0.1

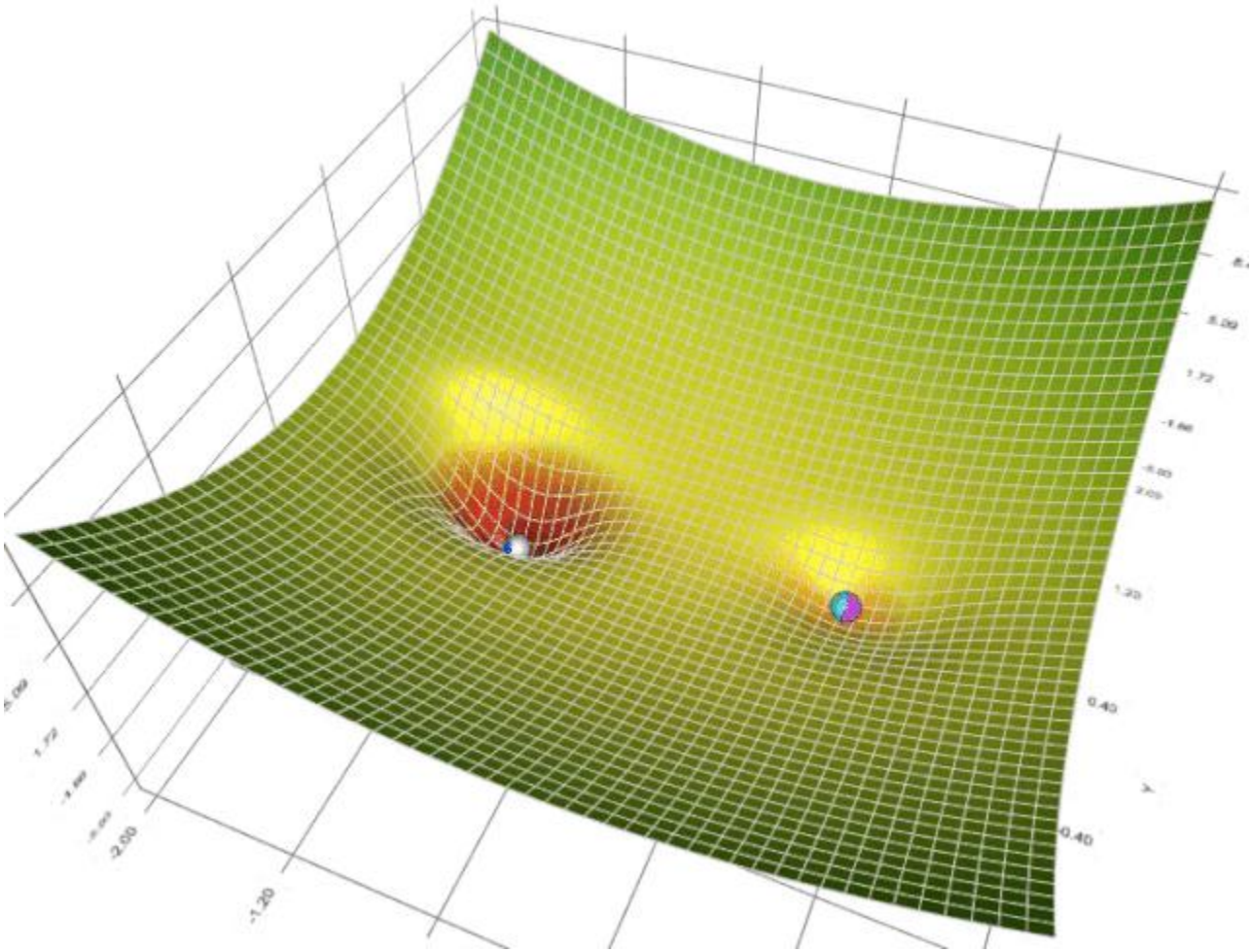
Gibbs Sampling and Climbing

Because gibbs sampling does always choose the best new location
it can move to another place not directly uphill



In theory, Gibbs Sampling less likely to get stuck a local maxima

梯度下降



AlignACE

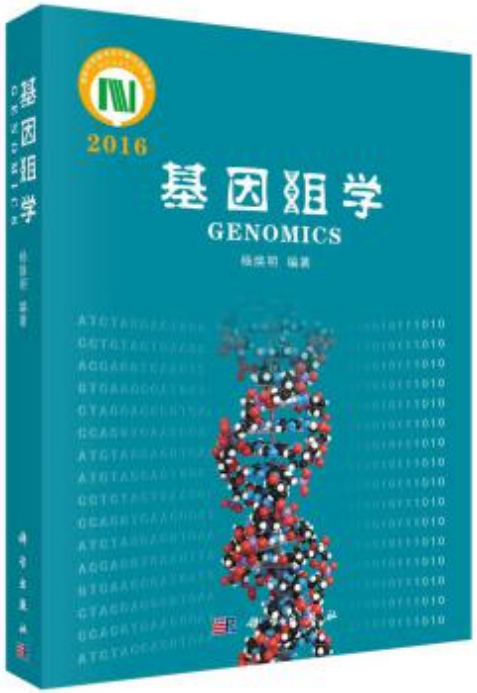
- **Implements Gibbs sampling for motif discovery**
- **ScanAce – look for motifs in a sequence given a model**
- **CompareAce – calculate "similarity" between two motifs (i.e. for clustering motifs)**

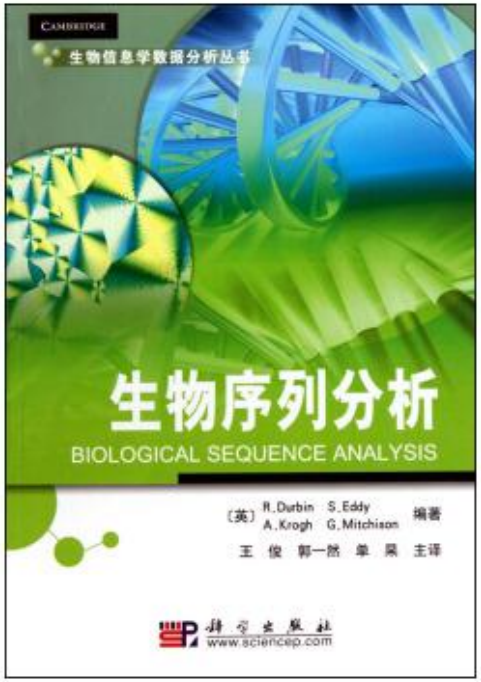
Reference

1. Roth, F.R., Hughes, J.D., Estep, P. E. & G.M. Church. Finding DNA Regulatory Motifs within Unaligned Non-Coding Sequences Clustered by Whole-Genome mRNA Quantitation. *Nature Biotechnology* 16, 939 - 945 (1998)
2. Hughes, JD, Estep, PW, Tavazoie S & GM Church. Computational identification of cis-regulatory elements associated with groups of functionally related genes in Saccharomyces cerevisiae, *Journal of Molecular Biology* 2000 Mar 10;296(5):1205-14.

http://arep.med.harvard.edu/mrnadata/mrnasoft.html

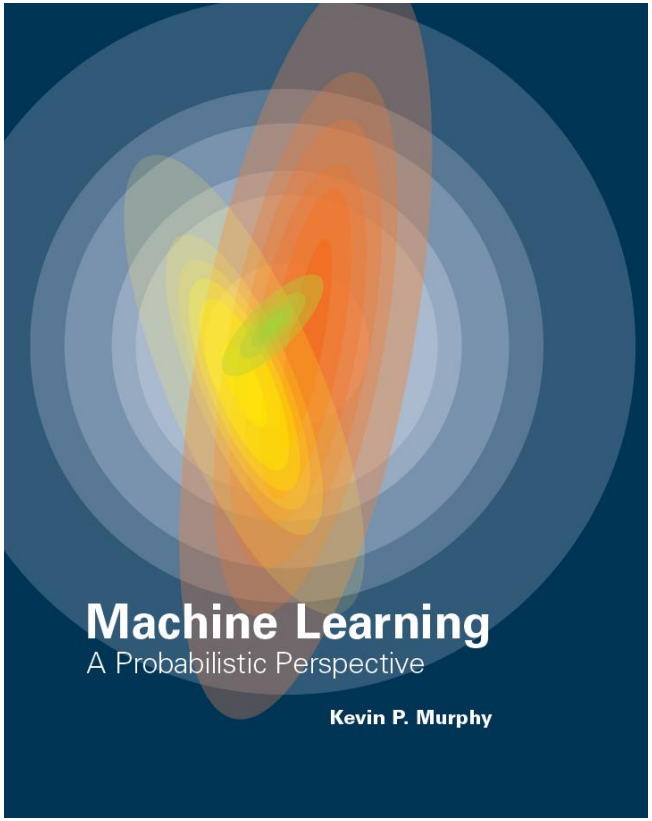
References


2016
基因组学
GENOMICS


CAMBRIDGE
生物信息学数据分析丛书
生物序列分析
BIOLOGICAL SEQUENCE ANALYSIS
R.Durbin S.Eddy
A.Krogh G.Mitchison
编著
王俊 郭一然 单昊 主译
www.sciencep.com


生物统计学
(第五版)

References



https://github.com/probml/pyprobml

Slides credits

——生物信息学研究方法概述：北京大学生物信息中心

——生物统计学：卜东波@中国科学院计算技术研究所，邓明华@北京大学

——神经网络与深度学习: 邱锡鹏@复旦大学

——Introduction to Computational Biology and Bioinformatics: Xiaole Shirley Liu Lab@Havard University

——Combinatorial Methods in Computation Biology: Ken Sung Lab@NUS

——Deep Learning in the Life Sciences: MIT

——Probabilistic Graphical Models: Eric Xing@CMU

——Numerous other leading researchers and leading labs.....

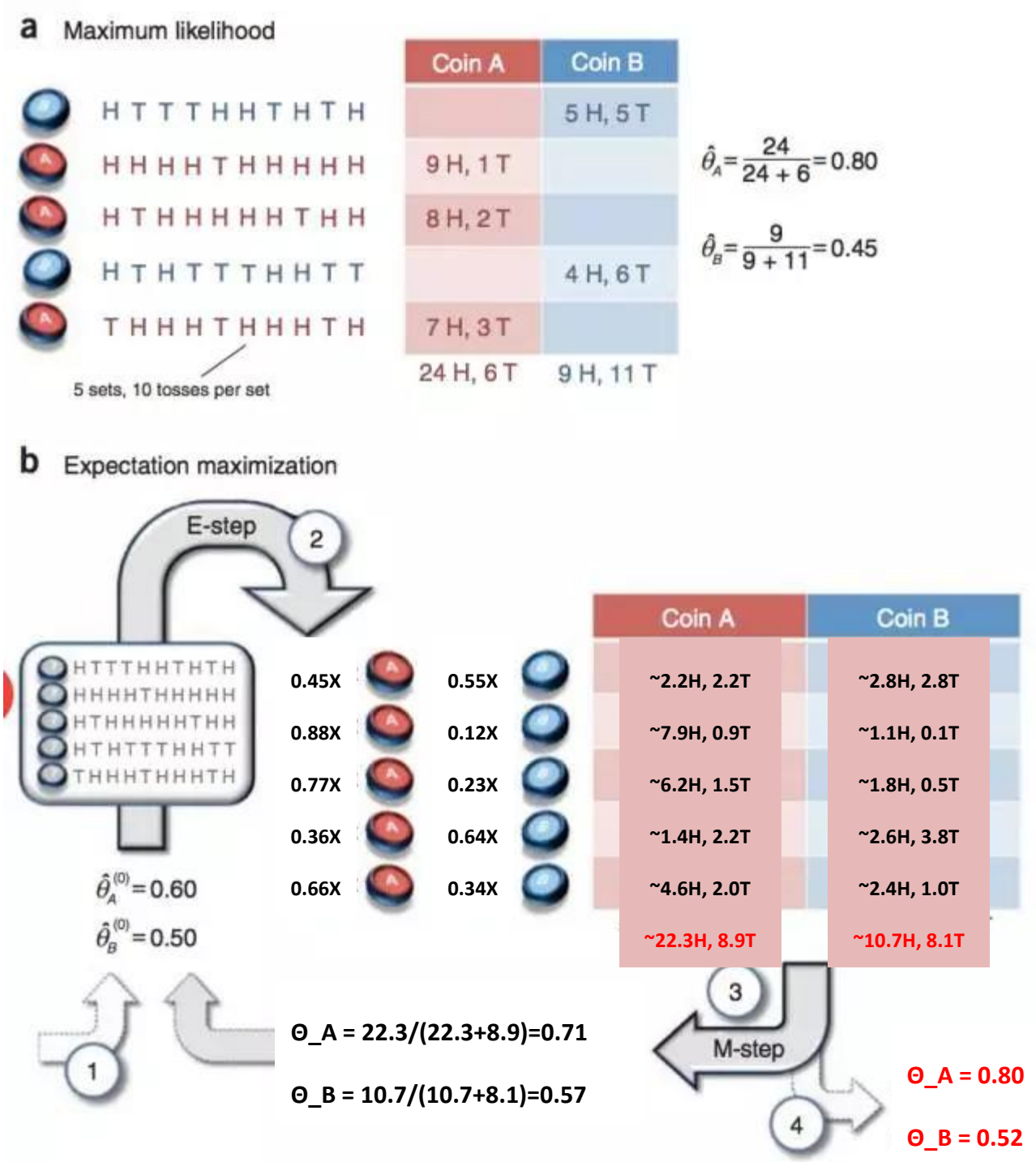
Below computation is wrong!!!

Example

抛3个硬币，抛I硬币决定C1和C2，然后抛C1或者C2决定正反面， 然后估算3个硬币的正反面概率值。

X: 观测到的正面
Z: 选取的哪一个硬币
每一次的情况：知道/不知道

Random distribution assumption



Example: Random distribution