The Expectation Maximization (EM) algorithm and some of its applications in Molecular Biology

Frédérique Galisson

Swiss Institute of Bioinformatics & University of Lausanne,
Switzerland
Introduction

When analysing biological data like sequences or gene expression measurements, many questions can be formalized as the clustering of the data into several groups, corresponding to different biological features. When these features can be represented through probabilistic models (like statistical profiles for modeling sequence motifs, or Markov chains for coding regions), mathematical tools are available, enabling one to compute the probability of any model given any observed data. If the models which are expected to be found in the data are known (supervised clustering), it is then easy to score each model with respect to the data, and classify the data according to these scores. When the model parameters are not known, and have to be guessed from the data (unsupervised clustering), the problem can be solved using the Expectation-Maximization (EM) algorithm. Developed in 1977 by Dempster et al., it can be used for a great variety of problems with missing (incomplete or hidden) data. It has been applied to several computational biology problems like the sequence motif inference problem, the unsupervised search for coding regions in sequences, or more recently, to the clustering of array data.

The objective of this tutorial is to give a unified presentation of the application of EM to these different problems, by first explaining a form of EM which can be used for clustering: "EM for Mixture Models", and then showing through three examples (motif inference, search for coding regions, and microarray data clustering) how different problems can been formalized the same way, with mixtures of probabilistic models specific to each problem, used in combination with the same EM algorithm.

The tutorial is thus organized into four parts:

1. **EM for Mixture Models**: the longest and most theoretical part of the tutorial, explaining the EM algorithm, and more specifically focusing on its application to the unsupervised learning of Mixture Models parameters. Then three examples of applications of this form of EM to molecular biology problems are presented:

2. **Motif inference from sets of unaligned sequence**
3. **Unsupervised search for coding regions**
4. **Clustering of micro-array data**

The handouts are also divided into the same four parts, each containing first a few notes, mainly pointing to bibliographic references, and then the set of slides corresponding to that part.

The EM algorithm has also been applied to other biological problems than the three mentioned above, such as phylogenetic inference, protein structure modeling, and others - see for instance (Huelsenbeck et al., 2001; Friedman et al., 2002; Rantanen et al., 2001; Holmes and Rubin, 2002). Since the topics of the tutorial is focused on the computational biology applications of EM which make explicit use of Mixture Models, for obvious reasons of time, important applications of other forms of EM are not presented here, as is also the case of related work using probabilistic graphical models and the Bayesian framework for inference (using EM, or other learning algorithms like the closely related Gibbs sampler). The notes try to compensate a bit for this by including references to some interesting, sometimes very sophisticated, related approaches.
The EM algorithm
EM = Expectation Maximization

Developed by Dempster and Rubin in 1977.
It belongs to the class of Machine Learning algorithms.

Machine learning:
- is the same as Statistical Model Fitting,
- aim: extract information from the data automatically (inference) via a process of model fitting (learning from examples).
- most useful in areas where there is a lot of data and little theory.
- theoretical background = Bayesian statistics, Probabilistic graphical models.

The EM algorithm

- Useful in a variety of model fitting problems with missing (incomplete, or hidden) data.
- Proceeds through iterative computation of Maximum Likelihood (ML) estimates.
- Main idea: to associate with the given incomplete-data problem, a complete-data problem for which ML estimation is computationally tractable.
- Some forms of EM, for Mixture Models, can be used for model-based clustering.
Biological Data

- Huge amount of data: sequences, expression measurements, others,
- We wish to extract biological information from these data: search for genes, pattern detection, clustering, etc
- The data are noisy, for experimental or biological reasons: evolution, superposition of signals (in sequences), biological variability and complexity...
- Bayesian statistics offer a rigorous framework to deal with noise, missing information, and prior knowledge ("working hypothesis"), and have been applied to a wide range of problems in computational biology.

Organization of the tutorial

The EM algorithm and some of its applications in Molecular Biology

Introduction (20 minutes)
- Part 1: EM for Mixture Models (70 min)

Coffee Break
- Part 2: Motif inference from sets of sequences (30 min)
- Part 3: Unsupervised search for coding regions (30 min)
- Part 4: Clustering of array data (30 min)

Conclusion and questions
Part I

The EM algorithm for Mixture Models

Theoretical background: the Bayesian framework

This introductory section (slides 5 to 9) presents the theoretical framework on which the EM algorithm is grounded - as many other machine learning approaches used in computational biology. The unification of many models and algorithms under the Bayesian framework is presented in (Baldi and Brunak, 2001), and the first part of the tutorial refers extensively to chapters 2, 3, 4 and appendices of this book. The use of the Bayesian framework for estimating and comparing models is specifically developed in (Kass and Raftery, 1995). The formalization of bayesian inference with biopolymer sequences is also developed in (Durbin et al., 1998) and (Liu and Lawrence, 1999).

Maximum Likelihood parameter estimation

This section (slides 10 to 15) briefly recalls the problem of Maximum Likelihood parameter estimation, and focuses on a class of problems which can be formalized using mixtures of probabilistic models. It uses a simple example of univariate Gaussian mixtures, inspired from (Hastie et al., 2001) where a full treatment of the ML estimation problem can be found.

The EM algorithm

The EM algorithm finds locally optimal solutions to the ML estimation problem. The most fundamental references about the core original algorithm are its many variations are (Dempster et al., 1977, its first publication), and (McLachlan and Krishnan, 1997). The presentation of EM which is given in slides 16 to 22 mainly refers to an excellent tutorial about EM (Bilmes, 1997). The proof of EM convergence, and the description of its behavior (slides 20 to 22) have been developed from (Russell, 1998) and from the "statistics" appendix of (Durbin et al., 1998).

Model-based clustering = EM for Mixture Models

The formalization of EM for Mixture Models presented through the slides 23 to 31 refers to (Bilmes, 1997), and also to (Banfield and Raftery, 1993), and (Fraley and Raftery, 1998; Fraley and Raftery, 2000) where developments about the use of EM for Mixture Models as a Model-Based clustering method and its relationships with other clustering methods, can be found.
Machine learning

Extract information from the data through a process of statistical model fitting (= model inference), and evaluate, compare, select models:

- Probabilistic graphical models: Neural Networks, Hidden Markov Models, others.
- Learning algorithms: Gradient descent, Monte Carlo methods, . . . , Expectation Maximization

Theoretical framework: Bayesian statistics

Bayesian Inference

Deduction and inference:

- Deduction: if $H \rightarrow D$, and $H$ is true, then $D$ is true.
- Induction: if $H \rightarrow D$, and $D$ is true, then $H$ is more plausible...

Bayesian framework for induction:

- starting with a space of alternative hypothesis (or models), which may have generated the data,
- we wish to express our relative preferences (or degrees of belief) with respect to the different hypothesis, in terms of the available information (the data, and possibly some prior knowledge).

---- The Cox-Jaynes axioms relate these degrees of plausibility to probabilities, thus providing us with a rigorous mathematical framework for model selection, evaluation, comparison, etc (Baldi and Brunak, 2001)
Bayes Theorem

For any two events \( A \) and \( B \),
\[
p(A, B) = p(A \mid B) p(B) = p(B \mid A) p(A),
\]
from which we derive:
\[
p(A \mid B) = \frac{p(B \mid A) p(A)}{p(B)}
\]

Inductive form:
\[
p(\text{Model} \mid \text{Data}) = \frac{p(\text{Data} \mid \text{Model}) p(\text{Model})}{p(\text{Data})}
\]

Let \( H_1, H_2, \ldots, H_k \) be a set of mutually exclusive alternative hypothesis fully explaining the data. We get \( p(D) = p(D, H_1) + p(D, H_2) \cdots + p(D, H_k) \), and:
\[
p(H_k \mid D) = \frac{p(D \mid H_k) p(H_k)}{p(D)} = \frac{p(D \mid H_k) p(H_k)}{\sum_{i=1}^{k} p(D \mid H_i) p(H_i)}
\]

Model evaluation and comparison

The models are probability distributions over the data, enabling one to compute the probability of the data according to the model, also called the Likelihood of the model given the data, and noted: \( \mathcal{L}(M \mid D) = p(D \mid M) \). From Bayes theorem, one can also compute the posterior probability of the model given the data, \( p(M \mid D) \).

- For each model, the odds ratio \( \frac{p(M_1 \mid D)}{p(M_2 \mid D)} \) measures the increase in our belief in the model, from the evidence brought by the data.

- Chosing the best model in the light of the observed data is made by comparing their posterior distributions given the data:
\[
\frac{p(M_1 \mid D)}{p(M_2 \mid D)} = \frac{p(D \mid M_1) p(M_1)}{p(D \mid M_2) p(M_2)}
\]

- \( \frac{p(D \mid M_1)}{p(D \mid M_2)} \) is called the Bayes Factor.

- BIC: the Bayesian information criteria is an approximation to \( p(D \mid M) \).
Model inference

In order to find the values of the model parameters which best fit the data, MAP (Maximum a Posteriori), or ML (Maximum Likelihood) estimates can be used. They are related through Bayes theorem:

$$\log p(M \mid D) = \log p(D \mid M) + \log p(M) - \log p(D)$$

- If we assume a uniform distribution over the set of alternative models, MAP and ML estimations are equivalent.
- If the fit of a model to some data $D$ is measured with an “error” function $f$, of the model parameters $\Theta$, and the data $D$, the associated likelihood function can always be defined as:

$$p(D \mid \Theta) = \frac{\exp(-f(\Theta, D))}{Z}$$

So, minimizing the error function is equivalent to maximizing the likelihood function.

Maximum Likelihood estimation

Let $x$ be a random variable, and $p(x \mid \Theta)$ a probability density function governed by the set of parameters $\Theta$. Let $X = x_1, x_2, \cdots, x_N$ be a data set of $N$ observations independently drawn from this distribution.

The likelihood of the model parameters given the data is:

$$\mathcal{L}(\Theta \mid X) = p(X \mid \Theta) = \prod_{i=1}^{N} p(x_i \mid \Theta)$$

The Maximum Likelihood estimation problem: given the observations $X$, find the set of parameters $\Theta^*$ which maximizes $\mathcal{L}$:

$$\Theta^* = \arg\max_{\Theta} \mathcal{L}(\Theta \mid X)$$

, or equivalently,

$$\Theta^* = \arg\max_{\Theta} \log \mathcal{L}(\Theta \mid X)$$
Maximum Likelihood estimation

\[ \Theta^* = \arg \max_{\Theta} \log \mathcal{L}(\Theta \mid X) \]

In simple cases, this can be solved analytically by taking the partial derivatives of the likelihood function with respect to the parameters, and looking for the values of \( \theta \) which verify:

\[ \frac{\partial \log \mathcal{L}}{\partial \theta} = 0 \]

Maximum Likelihood estimation

Example of a simple case

Let the probability distribution be a Gaussian, parameterized by its mean and variance: \( \theta = \{\mu, \sigma^2\} \).

The density function is given by:

\[ f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left( -\frac{(x-\mu)^2}{2\sigma^2} \right) \]

Let \( X = x_1, x_2, \ldots, x_N \) be a data set having supposedly been drawn from this distribution, and from which we want to estimate the \( \mu \) and \( \sigma^2 \) parameters.

Solving the previous partial derivative equations leads to the usual formulas:

- \( \mu = \frac{1}{N} \sum_{i=1}^{N} x_i \)
- \( \sigma^2 = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2 \)
Maximum Likelihood estimation

A Mixture Model example

Suppose now that each item of the data set has been drawn from one of 3 Gaussian distributions, each with its own set of parameters:

![Graph showing a mixture of three Gaussian distributions](image)

Mixture of univariate Gaussian distributions.

Mixture of Gaussians

If we know which distribution each data item has been drawn from, the data are classified into 3 sets, and we can estimate independently the 3 sets of parameters.

![Graph showing a mixture of Gaussian distributions](image)

Supervised learning
Mixture of Gaussians

When we don’t know anything, the data look like this:

Unsupervised learning: we have to infer from the data:
- the number of model components (= underlying statistical distributions),
- their respective proportions (the amount of data each of them generated),
- the ML parameter estimates for each model component ($\mu$ and $\sigma$ for univariate Gaussian distributions).

The Expectation - Maximization (EM) algorithm

The ML problem: solve $\Theta^* = \arg \max_{\Theta} \log \mathcal{L}(\Theta \mid \mathcal{X})$, $\mathcal{X}$ being a data set of $N$ independent observations drawn from the distribution $p(X \mid \Theta)$

Let us assume that:
- the observed data set $\mathcal{X}$ is incomplete, i.e. there are missing or hidden data $\mathcal{Y}$,
- there is a joint probability distribution $p(X, Y \mid \Theta)$ which we could compute if we knew the values of the $\mathcal{Y}$ data.

The likelihood of the complete data set can be written:

$$\mathcal{L}(\Theta \mid \mathcal{X}, \mathcal{Y}) = p(\mathcal{X}, \mathcal{Y} \mid \Theta) = \mathcal{L}(\Theta \mid \mathcal{X}) p(\mathcal{Y} \mid \mathcal{X}, \Theta)$$
The EM algorithm

Computation of the complete data log-likelihood:

The problem in order to compute (and maximize with respect to $\Theta$) the joint likelihood $L(\Theta \mid \mathcal{X}, \mathcal{Y})$ is that the values of the $\mathcal{Y}$ data are unknown, presumably governed by a probability distribution, conditioned by $\mathcal{X}$ and $\Theta$. If we can compute the conditional distribution, of the $\mathcal{Y}$ data, given the observed data ($\mathcal{X}$) and an estimate of the model parameters ($\Theta^g$), we can integrate over it to compute the expected value of the complete data log-likelihood:

$$E[\log L(\Theta \mid \mathcal{X}, \mathcal{Y}) \mid \mathcal{X}, \Theta^g] = \int_{y \in \mathcal{Y}} \log p(\mathcal{X}, y \mid \Theta)p(y \mid \mathcal{X}, \Theta^g)dy$$

or, for discrete data:

$$E[\log L(\Theta \mid \mathcal{X}, \mathcal{Y}) \mid \mathcal{X}, \Theta^g] = \sum_{y \in \mathcal{Y}} \log p(\mathcal{X}, y \mid \Theta)p(y \mid \mathcal{X}, \Theta^g)$$

The EM algorithm

iterates these two steps until convergence:

**E step:** computes the conditional expectation of the complete data log-likelihood, given the observed data and a previous estimate of the model parameters:

$$Q(\Theta, \Theta^g) = E[\log L(\Theta \mid \mathcal{X}, \mathcal{Y}) \mid \mathcal{X}, \Theta^g]$$

**M step:** maximizes the expectation computed at the E step with respect to the model parameters, i.e computes the next estimates:

$$\Theta^{g+1} = \arg\max_{\Theta} Q(\Theta, \Theta^g)$$

By iteratively maximising the conditional expectation of the complete data likelihood, EM finds local optima of the incomplete data likelihood, $L(\Theta \mid \mathcal{X})$. 
EM iterations

M step: \( \Theta^{g+1} = \text{argmax}_\Theta Q(\Theta, \Theta^g) \)

E step: \( Q(\Theta, \Theta^g) = E[\log \mathcal{L}(\Theta \mid X, Y) \mid X, \Theta^g] \)

Data:
\( X = (x_1, x_2, \ldots, x_N) \)
\( Y = (y_1, y_2, \ldots, y_M) \)

Model:
\( P(X, Y \mid \Theta) \)

EM, why does it work?

We wish to optimize \( l(\Theta) = \log \mathcal{L}(\Theta \mid X) = \log p(X \mid \Theta) \)

From \( p(X, Y \mid \Theta) = p(Y \mid X, \Theta)p(X \mid \Theta) \), we can write:
\( l(\Theta) = \log p(X, Y \mid \Theta) - \log p(Y \mid X, \Theta) \)

Multiplying each term with \( p(y \mid X, \Theta^g) \) and summing (or integrating) over \( Y \), yields:
\( l(\Theta) = \sum_y p(y \mid X, \Theta^g) \log p(X, y \mid \Theta) - \sum_y p(y \mid X, \Theta^g) \log p(y \mid X, \Theta) \)

\( \Leftrightarrow l(\Theta) = Q(\Theta, \Theta^g) - \sum_y p(y \mid X, \Theta^g) \log p(y \mid X, \Theta) \)
and \( l(\Theta^g) = Q(\Theta^g, \Theta^g) - \sum_y p(y \mid \mathcal{X}, \Theta^g) \log p(y \mid \mathcal{X}, \Theta^g) \)

Recall we want \( l(\theta) \geq l(\theta^g) \)

We can write:

\[
\begin{align*}
l(\Theta) - l(\Theta^g) &= Q(\Theta, \Theta^g) - Q(\Theta^g, \Theta^g) + \sum_y p(y \mid \mathcal{X}, \Theta^g) \log \frac{p(y \mid \mathcal{X}, \Theta^g)}{p(y \mid \mathcal{X}, \Theta)} \\
&= h(\Theta, \Theta^g) + \sum_y p(y \mid \mathcal{X}, \Theta^g) \log \frac{p(y \mid \mathcal{X}, \Theta^g)}{p(y \mid \mathcal{X}, \Theta)}
\end{align*}
\]

where the last term is a relative entropy (always positive), which implies that:

\[
l(\Theta) - l(\Theta^g) \geq Q(\Theta, \Theta^g) - Q(\Theta^g, \Theta^g)
\]

So, choosing at the M step \( \Theta^{g+1} \) such that \( Q(\Theta^{g+1}, \Theta^g) \geq Q(\Theta^g, \Theta^g) \)

(Generalized EM), suffices to guarantee that \( l(\Theta) = \log \mathcal{L}(\Theta \mid X) \) increases at each iteration, and converges towards a local optima of the \( l \) function.

**EM, how does it behave?**

\[
l(\Theta) \geq l(\Theta^g) + Q(\Theta, \Theta^g) - Q(\Theta^g, \Theta^g) = h(\Theta, \Theta^g)
\]
Mixture Models

A model $M$ made up with $C$ model components $m_1, m_2, \ldots, m_C$, is parametrized with the set $(\alpha_1, \alpha_2, \ldots, \alpha_C, \theta_1, \theta_2, \ldots, \theta_C)$, where $\alpha_j$ is the mixing coefficient for model component $m_j$, and $\theta_j$ is its set of parameters.

The probability distribution of the data is given by:

$$p(x | \Theta) = \sum_{j=1}^{C} \alpha_j p_j(x | \theta_j)$$

where $\sum_{j=1}^{C} \alpha_j = 1$, and each $p_j$ is a density function parameterized by $\theta_j$.

The log-likelihood function for this density, from a data set $X = (x_1, x_2, \ldots, x_N)$, is:

$$\log L(\Theta | X) = \log \prod_{i=1}^{N} p(x_i | \Theta) = \sum_{i=1}^{N} \log \left( \sum_{j=1}^{C} \alpha_j p_j(x_i | \theta_j) \right)$$

Mixtures Models and EM

Let us assume there are hidden data $Y$ which inform us about which model component generated each data item. The complete data set is $(X, Y) = (x_1, x_2, \ldots, x_N, y_1, y_2, \ldots, y_N)$, with $y_i = j$ if the data item $x_i$ has been generated by the model component $m_j$.

If we knew the values of the $Y$ (supervised learning), it would be easy to estimate the model parameters:

- $\alpha_j = \frac{1}{N} \sum_{i=1}^{N} n_{i,j}$, with $n_{i,j} = 1$ if $y_i = j$, $0$ otherwise.
- the $\theta_j$ are estimated independently for each model component, from its data subset: $x_i \in \mathcal{X}_j$, such that $y_i = j$. 
Mixtures Models and EM

We don’t know the \( y_i \) values (unsupervised learning), but we can compute their conditional distribution, given \( x_i \) and a previous estimate of the model parameters: 

\[
p(y_i = j \mid x_i, \Theta^g) = p(m_j \mid x_i, \Theta^g)
\]

How? \( \Rightarrow \) Through Bayes theorem...

\[
p(m_j \mid x_i, \Theta^g) = \frac{p(x_i \mid m_j, \Theta^g) p(m_j \mid \Theta^g)}{p(x_i \mid \Theta^g)} \quad \forall j
\]

\[
= \frac{p(x_i \mid m_j, \Theta^g) p(m_j \mid \Theta^g)}{\sum_{k=1}^C p(x_i \mid m_k, \Theta^g) p(m_k \mid \Theta^g)}
\]

\[
= \frac{p_j(x_i \mid \theta^g_j) \alpha_j^g}{\sum_{k=1}^C p_k(x_i \mid \theta^g_k) \alpha_k^g}
\]

EM for Mixture Models

We can know consider the \( y_i \) data as vectors of probabilities, \( y_i = (y_{i,1} \cdots y_{i,C}) \), with \( y_{i,j}^g = p(m_j \mid x_i, \Theta^g) \).

The conditional expectation of the complete data log-likelihood can be computed as:

\[
Q(\Theta, \Theta^g) = E[\log \mathcal{L}(\Theta \mid X^g) \mid X^g, \Theta^g]
\]

\[
= \sum_{j=1}^C \sum_{i=1}^N p(m_j \mid x_i, \Theta^g) \log \alpha_j + \sum_{j=1}^C \sum_{i=1}^N p(m_j \mid x_i, \Theta^g) \log p_j(x_i \mid \theta_j)
\]

\[
= \sum_{j=1}^C \sum_{i=1}^N y_{i,j}^g \log \alpha_j + \sum_{j=1}^C \sum_{i=1}^N y_{i,j}^g \log p_j(x_i \mid \theta_j)
\]
EM for Mixture Models

\[ Q(\Theta, \Theta^g) = \sum_{j=1}^{C} \sum_{i=1}^{N} y^g_{i,j} \log \alpha_j + \sum_{j=1}^{C} \sum_{i=1}^{N} y^g_{i,j} \log p_j(x_i | \theta_j) \]

**E step:** calculate \( Q(\Theta, \Theta^g) \):

\[ y^g_{i,j} = p(m_j | x_i, \Theta^g) = \frac{p_j(x_i | \theta^g_j) \alpha^g_j}{\sum_{k=1}^{C} p_k(x_i | \theta^g_k) \alpha^g_k} \]

**M step:** calculate the new ML estimates \( \Theta^{g+1} \):

\[ \alpha^{g+1} = \frac{\sum_{i=1}^{N} p(m_j | x_i, \Theta^g)}{N} \]

\( \theta^{g+1} \): depends on the probabilistic model which is used.

---

**Data:**

- \( x_1, (y_{1,1}, \ldots, y_{1,C}) = y_1 \)
- \( x_2, (y_{2,1}, \ldots, y_{2,C}) = y_2 \)
- \( x_N, (y_{N,1}, \ldots, y_{N,C}) = y_N \)

**Model:**

- \( m_1 : \{ \alpha_1, \theta_1 \} \)
- \( m_2 : \{ \alpha_2, \theta_2 \} \)
- \( m_C : \{ \alpha_C, \theta_C \} \)

**E step:** \( y^g_{i,j} = p(m_j | x_i, \Theta^g) \)
EM for a Mixture of three univariate Gaussian distributions

Data:
\[x_1, y_1 = (y_{1,1}, y_{1,2}, y_{1,3})
\]
\[x_2, y_2 = (y_{2,1}, y_{2,2}, y_{2,3})
\]
\[\vdots
\]
\[x_N, y_N = (y_{N,1}, y_{N,2}, y_{N,3})
\]

Model: mixture of three Gaussian distributions

\[G_1 : \{\alpha_1, \mu_1, \sigma_1\}\]
\[G_2 : \{\alpha_2, \mu_2, \sigma_2\}\]
\[G_3 : \{\alpha_3, \mu_3, \sigma_3\}\]

E step: calculate \(y_{i,j}^g = p(G_j \mid x_i, \Theta^g)\)

M step: \(\Theta^{g+1} = \{\alpha_j^{g+1}, \mu_j^{g+1}, \sigma_j^{g+1}\}_{j=1,2,3}\)

\[Q(\Theta, \Theta^g) = \sum_{j=1}^3 \sum_{i=1}^N y_{i,j}^g \log \alpha_j + \sum_{j=1}^3 \sum_{i=1}^N y_{i,j}^g \log p_j(x_i \mid \mu_j, \sigma_j)
\]

\[\text{E step: calculate } Q(\Theta, \Theta^g): \]
\[y_{i,j}^g = p(G_j \mid x_i, \Theta^g) = \frac{p_j(x_i \mid \mu_j^g, \sigma_j^g) \alpha_j^g}{\sum_{k=1}^3 p_k(x_i \mid \mu_k^g, \sigma_k^g) \alpha_k^g}
\]

\[\text{M step: calculate } \Theta^{g+1} = \{\alpha_j^{g+1}, \mu_j^{g+1}, \sigma_j^{g+1}\}_{j=1,2,3}\]

- \(\alpha_j^{g+1} = \frac{1}{N} \sum_{i=1}^N p(G_j \mid x_i, \Theta^g) = \frac{1}{N} \sum_{i=1}^N y_{i,j}^g\)
- \(\mu_j^{g+1} = \frac{\sum_{i=1}^N y_{i,j}^g x_i}{\sum_{i=1}^N y_{i,j}^g} = \frac{1}{N \alpha_j^{g+1}} \sum_{i=1}^N x_i y_{i,j}^g\)
- \((\sigma_j^{g+1})^2 = \frac{1}{N \alpha_j^{g+1}} \sum_{i=1}^N y_{i,j}^g (x_i - \mu_j^{g+1})^2\)
CEM: "Classification EM"

CEM is an approximation to EM for Mixture Models, making "hard" assignments of each data item to one single class (model component) at each E-step:

- E-step: compute the $y_i$ vector of probabilities for each data item $x_i$, and then set $y_{i,j}$ to 1 if $y_{i,j} = \max\{y_{i,1}, \ldots, y_{i,C}\}$, and to 0 otherwise.

- M-step: update the parameters of each model component simply taking into account the data items which have been assigned to the class it models.

Remarks:
- CEM is very closely related to the k-means algorithm.
- The approximation which is made here is analogous to the Viterbi approximation of the Baum-Welch algorithm (EM for the unsupervised learning of HMM parameters).
Part II
Motif inference from sets of unaligned sequences

The motif inference problem

The slides 32 to 34 briefly review the "multiple local sequence alignment", or "motif inference" problem. It has been addressed with many different approaches, among which we will focus on those using full probabilistic models of the sequence patterns (motifs). Other, more "deterministic" methods have also been developed leading for instance to the PRATT (Jonassen et al., 1995), and SMILE (Marsan and Sagot, 2000) programs, but will not be dealt with here.

Probabilistic models of sequence patterns

Slides 35 to 38: the most simple probabilistic models for biological sequence patterns are probabilistic sequence profiles of fixed length, where the multiple alignment columns are modeled by independent multinomial distributions over the letters of the sequence alphabet (amino acids or nucleotides). Flexible length models, allowing for insertions/deletions in the multiple alignment of the pattern occurrences, and modeling dependencies between subsets of the profile positions, are natural extensions of the simplest sequence profiles in a Bayesian framework. Profile Hidden Markov Models are such extensions, fully covered as many other probabilistic models of proteins or nucleic acids in (Durbin et al., 1998; Baldi and Brunak, 2001), and references therein.

Motif inference with EM

The slides 39 to 47 present the application of EM to the discovery of sequence motifs modeled by simple fixed length probabilistic profiles with independent positions. The simplest problem of finding exactly one occurrence of the pattern per sequence (Lawrence and Reilly, 1990), and some of its variations are first presented. We then focus of the more general formalization of the same problem as a two-component Mixture Model (Bailey and Elkan, 1994), and its generalization to find multiple motifs occurring zero or any number of times in the sequences from the data set, as implemented in the MEME program (Bailey and Elkan, 1995b; Bailey and Elkan, 1995a; Bailey, 1995).

Note about the Baum-Welch algorithm: one of the most important applications of EM in biology is to the unsupervised learning of HMM parameters (notably - but not only - used for modeling multiple sequence alignments), the hidden parameters (the state sequence, with respect to each biological sequence to be aligned to the HMM) are treated as missing variables by EM. Because the tutorial focuses on applications of EM for Mixture Models, the Baum-Welch algorithm is not presented here, but fully explained elsewhere as (Durbin et al., 1998), and (Bilmes, 1997, for a more general treatment).
The "local multiple alignment" or "motif inference" problem

Questions:

- positions of the motif occurrences in the sequences?
- number of motif occurrences per sequence?
- length of the motif? (fixed or flexible?)
- degree of conservation/divergence of the motif positions?
- how to evaluate, compare different motif candidates?
Approaches to motif inference

Using "deterministic" models: PRATT, (Jonassen et al., 1995), SMILE (Marsan and Sagot, 2000), others.

Using probabilistic models:
- Greedy algorithm: CONSENSUS (Hertz and Stormo, 1999).
- Stochastic methods:
  - EM: (Lawrence and Reilly, 1990); MEME, (Bailey and Elkan, 1994, 1995b)
  - Gibbs sampling: (Lawrence et al., 1993),
  - EM with HMM: the Baum-Welch algorithm.
- Many other specific applications of probabilistic models and EM or Gibbs sampling (Thijs et al., 2002, and others...).

Probabilistic models of sequence motifs

Simple probabilistic profiles

The parameters of the model are independent vectors of emission probabilities: $P_j = (P_{1,j}, \ldots, P_{A,j})$ where $A$ is the size of the alphabet. The probability of a sequence $S = s_1, s_2, \ldots, s_L$ of length $L$ under such a model is given by $p(S \mid profile) = \prod_{j=1}^{L} P_j(s_j)$
Estimating the parameters of probabilistic sequence profiles

**Supervised learning:** if we know the positions of the pattern occurrences, we can align the sequences, and compute the model parameters from this multiple alignment.

- if we assume that we know a priori nothing about the multinomial distributions underlying the composition of the alignment columns, the probability of an alignment column is given by: \( p(column_j \mid profile) = \prod_{i=1}^{A} (p_j(i))^{n_{i,j}}, \) where \( n_{i,j} \) is the number of times the letter \( i \) is observed at position \( j \) of the motif in the aligned sequences. In that case, the maximum likelihood estimates of the profile parameters are just the relative frequencies of the letters at each position:

\[
p_j(i) = \frac{n_{i,j}}{N}
\]

Problem when the data is scarce...

- prior knowledge about the profile position-specific distributions can be integrated under the form of pseudo-counts in the estimates:

\[
p_j(i) = \frac{c_{j,i} + \alpha_i}{N + \sum_i \alpha_i}
\]

are the Posterior Mean Estimates (PME) of \( P_j \) if we consider the probability vectors as random variables governed by a prior Dirichlet distribution with parameter \( \alpha = (\alpha_1, \alpha_2, \cdots, \alpha_A) = \beta(q_1, q_2, \cdots, q_A). \)

Mixtures of Dirichlet distributions (modeling prior knowledge about several kind of alignment columns) are usually used as priors when modeling protein sequence motifs with simple probabilistic profiles, or more sophisticated graphical probabilistic models like Hidden Markov Models.
Generalized profiles & Profile Hidden Markov Models

- Each model position is modeled by a Match and an Delete state, the former being a vector of emissions probabilities.
- Insertion states allow extra residues between model positions.
- Transitions between states are parameterized by conditional probabilities, e.g. $p(M_j \mid D_{j-1})$

The "one occurrence per sequence" problem

Can be solved without formally using Mixtures Models (Lawrence and Reilly, 1990):

- The motif is modeled with an ungapped probabilistic profile of fixed length ($L$).
- Knowing the motif occurrence positions, the complete data log-likelihood for a sequence of length $K$, is given by:

$$\log \mathcal{L} = \sum_{j=1}^{L} \sum_{i=1}^{A} n_{i,j} \log P_{i,j} + (K - L) \sum_{i=1}^{A} n_{i,0} \log P_{i,0}$$

EM: The unknown motif positions in the sequences are considered as missing variables.
The "one occurrence per sequence" problem

M step: $[P_{y+1}^0, P_{y+1}^1, \ldots, P_{y+1}^L]$

E step: $p\{z_i = k\} | x_i, P_0, P_1^0, P_2^0, \ldots, P_L^0$

For each sequence position, the probability that it is the first position of a motif occurrence is computed at the E-step.

At each M step of EM, the model parameters $P_j$ are updated by counting the letter occurrences for each the motif position, considering every possible starting point for the motif in the sequences, and weighting the counts by the probability of the motif starting at that position (computed at the previous E step).
Variable length motifs, zero or multiple motifs occurrences per sequence

- Cardon and Stormo 1992: two-sites motif with a variable length spacer whose length is considered as another missing variable.
- Lawrence et al 1993: Gibbs sampling approach
- others...
- MEME (Bailey and Elkan 1994, 1995), Multiple EM for Motif Elicitation: uses a two-component Mixture Model, and a mixture of sequences derived from the original dataset.

Mixture modeling of sequence motifs

The dataset is broken into all $N$ overlapping subsequences of length $L$ that it contains:

$\mathcal{X} = \{x_1, x_2, \cdots, x_N\}$

The model is made up with two components:
- a probabilistic profile of length $L$ modeling the motif,
- a vector of $A$ background probabilities modeling the non-motif model positions.
Motif inference with a two-component Mixture Model

**Data:** mixture of sequences

$x_i, y_i = (y_{i,M}, y_{i,B})$

**Model:**

{\( \alpha_{M}, \alpha_{B}, \{P_1 \cdots P_L\} \)}

$\{\alpha_{M}, \alpha_{B}, \{P_1 \cdots P_L\}\}$

**Motif**

\( y_{i,M} = p(M | x_i, \Theta) \)

\( y_{i,B} = p(B | x_i, \Theta) \)

**M step:** \( \alpha_{M}^{g+1}, \alpha_{B}^{g+1}, \{P_1^{g+1} \cdots P_L^{g+1}\}, P_0^{g+1} \)

**E step:**

Advantages of the mixture modeling:

- the original sequences may contain any number of motif occurrences,
- the formalism can easily account for more sophisticated probabilistic graphical models of the motif,
- it can be extended to discover multiple motifs.
MEME, Multiple EM for Motif Elicitation

After each run of EM, the sequences which are classified as occurrences of the discovered motif are "erased" from the data set, and EM is applied again:

MEME: soft (probabilistic) erasure of discovered motifs

After convergence of the first EM run, one motif, $M_1$ has been discovered, and the $x_i$ sequences from the mixture data set have been each assigned a probability of being an occurrence of the discovered motif:

$$y_{i,M_1} = p(M_1 \mid x_i).$$

Before the next run of EM, each sequence is in addition assigned a weight, $w_i$, which is originally set to $w_i^0 = 1$, and represents the probability that $x_i$ is not part of a previously discovered motif (being itself an occurrence, or an overlap with such an occurrence):

$$w_i^1 = w_i^0 \prod_{k=i-L+1}^{i+L-1} (1 - y_{k,M_1}).$$

At the next EM run, the estimation of the relative frequencies at the M step are made by weighting each count with $y_{i,M_2} \cdot w_i^1$, thus lowering the contributions of the $x_i$ data to the newly discovered motif, according to their probabilities of overlapping the first motif.
Part III
Unsupervised search for coding regions

Introduction

The second illustration of EM and Mixture Models presented here (slides 48 to 64) is the unsupervised search for coding regions in sets of sequences containing both coding (in any frame) and non-coding sequences. The different approaches used in gene prediction, reviewed in (Mathe et al., 2002), are briefly recalled, before focusing on the "search by content", i.e. the modeling and prediction of regions having a "coding potential".

Modeling coding regions

The first slide (slide 50) mainly refers to pioneer work in the modeling of coding regions (Fickett, 1982; Staden, 1983; Gribskov et al., 1984) and this review (Fickett, 1992). Then, the modeling of coding regions using heterogeneous Markov chains has been developed by Mark Borodovsky (Bodorovsky et al., 1986; Borodovsky and McIninch, 1993) (slides 51 to 54).

Supervised learning, and prediction

A supervised learning approach is used in conjunction with a seven-component heterogeneous Markov model in the Genemark program (Borodovsky and McIninch, 1993) (slides 55 to 58), and such supervised approaches with similar models are also part of most gene predictions programs both for procaryotic and eucaryotic genomes - such as GLIMMER (Salzberg et al., 1998), GRAIL (Xu et al., 1996) or GENSCAN (Burge, 1997).

Unsupervised learning using EM

The previous supervised learning approach works well as long as one has a training set of known coding and non-coding sequences. Because the statistical biases which are being modeled are organism-specific (Fickett, 1992) this is not the case when one analyses the new genome of an organism which has not been extensively experimentally studied. This is the first motivation towards the development of unsupervised approaches for detecting coding regions in unknown sequence data (slide 58). Previous work (Hayes and Borodovsky, 1998; Audic and Claverie, 1998; Borodovsky and Besemer, 1999) have proven the relevance of the approach. We implemented a general version of EM for mixture models as well as a seven-component model (slides 59 to 64), and validated the algorithm on both synthetic and real-world procaryotic genomes. Further applications to eucaryotic data and towards the discovery of different classes of genes (Hayes and Borodovsky, 1998) are underway.
Search for coding regions using a Mixture Model of coding regions and EM

M step

Mixture of coding and non coding sequences:

E step

Mixture model of coding and non coding regions:

Periodic Markov chains

Approaches to gene prediction

Current approaches usually combine searches by:

- **signal**: promoters, splicing sites, RBS, etc
- **similarity**: with transcriptome data, homology of the putative protein product with known proteins, conserved regions between (close) homologous genomes.
- **content**: identifying regions having a *coding potential*, from their statistical biases induced by the coding constraints.
Modeling coding regions

Factors that affect the base and codon compositions of coding regions:
1. uneven use of amino-acids in proteins,
2. redundancy of the genetic code: unequal number of codons for different amino acids,
3. codon preference: synonymous codons are not equally used.

Effects that coding has on a DNA sequence: "signature" of coding regions:
• positional base frequencies,
• codon, di-codon, etc, frequencies, are different in each of the three possible reading frames.

Modeling the coding statistical biases:
chosen model = Inhomogeneous Markov Chains of period 3

Markov Chains

The sequence is considered as a string of random variables:
\[ S = s_1 s_2 \ldots s_i s_{i+1} \ldots s_n \]

Homogeneous Markov Chains:

order 0: \( p(S) = p(s_1) p(s_2) \ldots p(s_i) p(s_{i+1}) \ldots p(s_n) \)

order 1: \( p(S) = p(s_1) p(s_2 | s_1) \ldots p(s_{i+1} | s_i) \ldots p(s_n | s_{n-1}) \)

order k: \( p(S) = p(s_1 \ldots s_k) \prod_{i=1}^{n-k} p(s_{i+k} | s_i \ldots s_{i+k-1}) \)

In order to reflect the statistical biases at the level of codons or di-codons, the natural choice for modeling coding regions is to use Markov chains of order 2 or 5.

The tri- or hexa-nucleotide frequencies are expected to be different with respect to the reading frame which is considered, and the model also needs to take this into account...
A sequence is said to be coding in frame $j$ when its first nucleotide is at the position $j$ of a codon.

**Inhomogeneous Markov Models for coding regions**

**Heterogeneous** Markov chain of order $k$ and period $p$: there are $p$ probability distributions, modeling different (but periodic) positional oligonucleotide frequencies

\[ p(S) = p^1(s_1 \ldots s_k) \prod_{j=1}^{p}(i-1 \mod p)+1 (s_{i+k} | s_{i} \ldots s_{i+k-1}) \]

**Modeling coding frames:**

\[
\begin{align*}
 p(S \mid \text{cod}_1) &= \pi^1_{s_1 \ldots s_k} T_1 s_1 \ldots s_k, s_{k+1} T_2 s_2 \ldots s_{k+1}, s_{k+2} T_3 s_3 \ldots s_{k+2}, s_{k+3} \ldots \\
p(S \mid \text{cod}_2) &= \pi^2_{s_1 \ldots s_k} T_1 s_1 \ldots s_k, s_{k+1} T_2 s_2 \ldots s_{k+1}, s_{k+2} T_3 s_3 \ldots s_{k+2}, s_{k+3} \ldots \\
&\vdots \\
p(S \mid \text{cod}_{-3}) &= \pi^{-3}_{s_1 \ldots s_k} T^{-3} s_1 \ldots s_k, s_{k+1} T^{-1} s_2 \ldots s_{k+1}, s_{k+2} T^{-2} s_3 \ldots s_{k+2}, s_{k+3} \ldots \\
\end{align*}
\]

**Modeling non coding sequences:**

\[
\begin{align*}
 p(S \mid \text{cod}_0) &= \pi^0_{s_1 \ldots s_k} T^0 s_1 \ldots s_k, s_{k+1} T^0 s_2 \ldots s_{k+1}, s_{k+2} T^0 s_3 \ldots s_{k+2}, s_{k+3} \ldots \\
\end{align*}
\]
Model of coding and non coding sequences

The sequences are considered under seven (mutually exclusive) hypothesis:

- Coding on the direct strand
  - cod3
  - cod2
  - cod1
  - cod0

- Non coding
  - cod-1
  - cod-2
  - cod-3

- Coding on the reverse strand
  (shadow of a coding sequence)

Each model component \((\text{cod}_j)\) is parametrized with 3 probability distributions, from the set \(\left\{ T_f \right\}_{j=3}^{3+3}\), where each \(T_f\) is a table of transitions probabilities

\[
T_{w_1 \ldots w_k, w_{k+1}} = p_f(w_{k+1} | w_1 \ldots w_k).
\]

Estimation of the model parameters

Supervised learning

If we have got a training set of sequences that we know are coding (and in which frame), and a set of non-coding sequences, the ML estimates of the model parameters are given by:

\[
T_{w_1 \ldots w_k, w_{k+1}} = \frac{N_{w_1 \ldots w_k, w_{k+1}}^f}{N_{w_1 \ldots w_k}^f}
\]

where \(N_{s}^f\) is the number of times the substring \(s\) has been found as coding in frame \(f\) (i.e the first nucleotide of string \(s\) is at the position \(f\) of a codon from a coding sequence)
Predicting new coding sequences

How is this score computed?...

Through Bayes theorem:

\[ \text{sc} = p(\text{cod}_j \mid S) = \frac{p(S|\text{cod}_j)p(\text{cod}_j)}{\sum_{k=-3}^{3} p(S|\text{cod}_k)p(\text{cod}_k)} \]

Slide 56

Slide 57

Output of the Genemark program
Motivations for an unsupervised approach

Limitations of the supervised approach: one needs a training set for estimating the model parameters, which is a problem for newly sequenced genomes, from organisms for which no such data set exists. Ad Hoc solutions are being used (e.g. building a training set from “long” ORF, which are expected to be true ones).

Discovering different class of genes in a single organism.

Previous work:
- Iterative approach to gene discovery, and clustering of the predicted genes into classes using the \textit{k-means} algorithm in \textit{E. coli} (Hayes and Borodovsky, 1998).
- Iterative (“CEM-like”) approach to gene discovery in procaryotic genomes (Audic and Claverie, 1998; and Baldi, 2000, for the EM interpretation).

The data set, a mixture of coding and non coding sequences

From a whole genome, the data set is generated by sliding a window along the genomic sequence. The length of the window as well as the sliding step are parameters which can be varied.

Genomic sequence:

Mixture of sequences:
Mixture Model of coding and non-coding sequences

The sequences are considered under seven (mutually exclusive) hypothesis:

- Coding on the direct strand
  - cod3
  - cod2
  - cod1
  - cod0
- Sequence S
- Non coding
  - cod-1
  - cod-2
  - cod-3
- Coding on the reverse strand
  - (shadow of a coding sequence)

Model parameters:

- Mixing coefficients: \( \alpha = \{\alpha_j = p(\text{cod}_j)\}^{+3}_{j=-3} \)
- Transitions probabilities: \( T = \{T^f\}^{+3}_{j=-3} \)

EM with Mixture Models of coding and non-coding sequences

\[
Q(\Theta, \Theta^q) = \sum_{i=1}^{N} \sum_{j=-3}^{+3} y^n_{i,j} \log \alpha_j + \sum_{i=1}^{N} \sum_{j=-3}^{+3} y^n_{i,j} \log p(\text{seq}_i \mid \text{cod}_j)
\]

E step: calculate \( Q(\Theta, \Theta^q) \):

\[
y^n_{i,j} = p(\text{cod}_j \mid \text{seq}_i, \Theta^q) = \frac{\alpha^n_j p(S \mid \text{cod}_j)}{\sum_{h=-3}^{+3} \alpha^n_h p(S \mid \text{cod}_h)}
\]

M step: calculate \( \Theta^{q+1} \):

\[
\alpha^{q+1}_j = \frac{\sum_{i=1}^{N} p(\text{cod}_j \mid \text{seq}_i, \Theta^q)}{N}
\]

\( T \): in order to compute \( (T^f)^{q+1} \), count each word \( w \) of length \( k+1 \), each time it appears at position \( f \) of a codon in a sequence \( \text{seq}_i \), and weight this count by the appropriate \( y^n_{i,j} = p(\text{cod}_j \mid \text{seq}_i, \Theta^q) \)
Updating the model parameters at the M step

Unsupervised search for coding regions using EM and Mixture Models

(R. Greset, E. Becker, and F. Galisson)

- modular implementation of EM for Mixture Models written in Java: prototype allowing to easily test various models (for various problems).
- implementations of a mixture model of heterogeneous Markov chains for modeling coding regions.
- validation of both the supervised and unsupervised (EM) approaches with synthetic and real data.
Discovering coding regions using EM and mixture models

Data: mixture of sequences

\[ \text{seq}_i, y_i = (y_{i-3} \cdots y_{i+3}) \]

Model: \( \{\alpha_j, \text{cod}_j\}_{j=1}^{a+1} \) with parameters:

\[ \{T^f\}_{j=1}^{a+1} \]

\[ T^f = \{T_{aaaaaa,}, T_{aaaaaC}, \cdots\} \]

E step: \( y_{ij}' = p(\text{cod}_j \mid \text{seq}_i, \Theta) \)

M step: \( \{\alpha_j^{a+1}\}_{j=1}^{a+1}, \{T^{-3}\cdots T^{a+3}\}^{a+1} \)
Part IV
Clustering of array data

Introduction

The technology of microarrays allows to study simultaneously the amounts of thousands of different DNA or RNA molecules, through the measurements of their hybridization intensities with their complementary sequences which are fixed on a nylon or glass support. The different stages of a microarray experiment, from its design to its interpretation(s) through biocomputational methods, are developed in two recent books and the many references they contain (Baldi and Hatfield, 2002; Kohane et al., 2002). The introduction (slides 65 to 67) briefly recalls the kind of experiments which can be done, and the important steps in their analysis (Brazma and Vilo, 2000, and books cited above). We will then focus on the clustering of gene expression measurements, assuming the lower levels of the data analysis (bg/fg separation, data normalization and transformation) have been adequately performed.

Note: the issue of determining when two gene expression measurements are significantly different (which arises for instance when hybridizing the the same arrayed probe with two targets coming from two different samples and labeled with different dyes) is beside the main focus here and is not going to be dealt with. It has lead to several solutions (Pan, 2002), among which a Bayesian framework has been proposed (Baldi and Long, 2001).

Model-based clustering of gene expression data

When looking at the gene expression profiles from more than two samples, we enter a higher level of analysis, where we wish to make sense of high-dimensional data depending of two kind of variables (samples and genes). The first issue in order to make sense of the data is to reduce the number of variables and classify the data. A sophisticated method for reducing the dimensionality of the data is PCA – see for instance (Baldi and Hatfield, 2002, chapter6), (Yeung and Ruzzo, 2001). In the absence of any prior knowledge regarding the classification of the data, unsupervised clustering methods are required. Classical methods for classification such as hierarchical clustering algorithms or k-means have been applied to array data (Baldi and Hatfield, 2002; Brazma and Vilo, 2000, for reviews). These “classical” approaches are briefly reviewed in slides 68 to 70.

Here, we focus on the model-based approaches and in particular the use of EM and Mixture Models (slides 71, 75 and references therein), which have more recently been applied with much success to microarray data.

Other applications of EM to the clustering of array data

A rapid survey of some recent publications relative to this approach, as well as to some related ones, is given in slides 76 to 78.
**Microarray technology**

Allows to perform and monitor simultaneously thousands of nucleic acid hybrids.

**The probes:**
- oligonucleotides or PCR products,
- support: glass slides or nylon membranes,
- arraying: in-situ oligonucleotide synthesis, or spotting of pre-synthesized molecules,

**The targets:**
- cDNA (from full mRNA samples), genomic DNA, PCR products,
- labelling: radioactive or fluorescent (can allow differential measurements, using two dyes, cy-3 and cy-5).

Rapidly evolving!

---

**Microarray experiments**

The probes are usually "markers" of genes, used to ask many kinds of questions regarding the expression levels of the corresponding genes, the evolution between genomes, the growth of different populations, etc.

Examples of questions:
- which gene expression pattern for which tumor type?
- which genes are co-expressed in which condition? when?
- which mutants in a population are affected in which process?

many others and many more in the future as the technology improves.

Focus on gene expression data and related questions.
Microarray analysis

1. data acquisition (signal processing: raw measurements, image segmentation, etc)
2. data transformation
3. first level statistics: single-gene measurements, differential hybridizations,
4. higher analysis levels:
   • studying multiple genes and samples at the same time, in order to extract "patterns" from the data: clustering,
   • combining with other data analysis approaches (e.g. looking for common patterns in gene promoters)

Clustering of gene expression data

Supervised clustering: some data are "labeled" (e.g. we know which tissue samples belong to the which class of tumor), and a "classifier" is built from the features of the classified data.

PCA (Principal Component analysis): method for reducing the dimensionality of the data (not a clustering method per se),

Unsupervised clustering: SOMs, hierarchical clustering, k-means, model-based clustering,
Hierarchical clustering

- requires a distance or similarity metrics,
- starts from pairwise distances and iteratively merges (agglomerative methods) or partitions (decomposition methods) the clusters,
- uses maximum, average, or minimum linkage rules to re-compute the distances between the cluster members at each step,
- the number of clusters is determined \textit{a posteriori}, usually by setting thresholds upon the intra/inter-cluster distances (and depends on the scale at which we look at the data).

k-means

- requires a distance or similarity metrics (e.g. euclidian distance between gene expression vectors),
- the number of clusters has to be chosen \textit{a priori},
- starts with a random (or "heuristic") partitioning, and then iterates:
  1. compute a representative point ("center") for each cluster,
  2. adjust the partition so that each point belongs to the cluster the center of which is the closest,
Model based clustering of gene expression data

Samples or experiments are represented by gene expression vectors of dimension $N$, and can be viewed as points in an $N$-dimensional space.

Genes are be represented as sample expression vectors of dimension $P$, with usually $P \ll N$. In general, clustering samples with respect to their gene expression profiles first requires to reduce the number of genes to those relevant for the clustering.

**Clustering genes:** the clusters (groups of genes with similar sample expression patterns) are modeled with multivariate Gaussian distributions, each parameterized with its own mean vector $\mu$ and covariance matrix $\Sigma$, and the whole data set is modeled as a Gaussian Mixture whose **number of components**, $C$ has to be set **a priori**.

Modeling with Gaussian distributions

Representation of the covariance matrix of a multivariate Gaussian distribution and its geometric interpretation, from (Banfield and Raftery, 1993; Fraley and Raftery, 1998):

$$\Sigma_j = \lambda_j D_j A_j D_j^T$$

where:

- $D_j$ is the orthogonal matrix of eigenvectors, and determines the orientation of the component,
- $A_j$ is a diagonal matrix whose elements are proportional to the eigenvalues of $\Sigma_j$, and which determines the shape of the cluster,
- $\lambda_k$ is a scalar which determines the volume of the cluster.
Modeling with Gaussian distributions

<table>
<thead>
<tr>
<th>$\Sigma_j$</th>
<th>Distribution</th>
<th>Volume</th>
<th>Shape</th>
<th>Orientation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda I$</td>
<td>spherical</td>
<td>equal</td>
<td>equal</td>
<td></td>
</tr>
<tr>
<td>$\lambda_j I$</td>
<td>spherical</td>
<td>variable</td>
<td>equal</td>
<td></td>
</tr>
<tr>
<td>$\lambda D_j A D_j^T$</td>
<td>ellipsoidal</td>
<td>equal</td>
<td>equal</td>
<td></td>
</tr>
<tr>
<td>$\lambda D_j A D_j^T$</td>
<td>ellipsoidal</td>
<td>equal</td>
<td>equal</td>
<td></td>
</tr>
<tr>
<td>$\lambda_j D_j A D_j^T$</td>
<td>ellipsoidal</td>
<td>variable</td>
<td>equal</td>
<td></td>
</tr>
<tr>
<td>$\lambda_j D_j A D_j^T$</td>
<td>ellipsoidal</td>
<td>variable</td>
<td>variable</td>
<td>variable</td>
</tr>
</tbody>
</table>

Clustering genes with EM and Gaussian Mixtures

**Data:**
- $N$ genes, $N$ expression vectors of dimension $P$
- $x_i = (e_{i,1}, \cdots, e_{i,P}), y_i = (y_{i,1}, \cdots, y_{i,C})$

**Model:**
- $C$ Gaussian densities $\\text{cluster}_j : \{\alpha_j, \mu_j, \Sigma_j\}$

**M step:** $\Theta^{p+1} = \{\alpha_j^{p+1}, \mu_j^{p+1}, \Sigma_j^{p+1}\}_{j=1}^C$

**E step:** $y_i^p = p(\text{cluster}_j \mid x_i, \Theta^p)$
Clustering genes with EM and Gaussian Mixtures


- assume the data are distributed according to a mixture of Gaussian densities,
- explore the extent to which different transformations of the data satisfy the normality assumption,
- use previously published MCLUST model-based clustering implementation (same authors), to synthetic and real data sets,
- Evaluation of the results and model selection/comparison:
  - use Rand indices to compare results on synthetic data with the "true" answer,
  - use the BIC score to compare models, and show that it selects the number of clusters which maximizes the Rand index.

Model-based clustering of microarray data

Ghosh and Chinnaiyan, 2002: *Mixture modelling of gene expression data from microarray experiments.*

McLachlan et al, 2002: *A mixture model-based approach to the clustering of microarray expression data.* Addresses additional problems, in particular the dimensional problem when clustering tissues. (EMMIX-GENE software)

Luan and Li, 2003: *Clustering of time-course gene expression data using a mixed-effects model with B-splines.*
Other extensions of the Model-based clustering approach

- Taking into account other data types:
  Holmes and Bruno, 2000, Barash and Friedman, 2002
  Segal et al, 2001, 2003: *Rich probabilistic models for gene expression* (Segal et al., 2001), and their use with EM (Segal et al., 2003a; Segal et al., 2003b) (ISMB2003 best paper awards)

- Not specifying the number of clusters:
  Medvedovic and Sivaganesan, 2002: *Bayesian infinite mixture model based clustering of gene expression profile*. (clustering performed with a Gibbs sampling algorithm).

- Use of hierarchical mixture models to model heterogeneity in tumor samples (Ghosh, 2004).

and several others...

Moloshok et al 2002: *Application of Bayesian Decomposition for analysing microarray data*

Desmet et al, 2002: *Adaptive quality-based clustering of gene expression profiles*

Lee et al, 2003: *Gene selection: a bayesian variable selection approach*

Sabatti et al, 2003: *Co-expression pattern from DNA microarray experiments as a tool for operon prediction*

Irizarry et al, 2003: *Use of Mixture Models in a microarray-based screening procedure for detecting differentially represented yeast mutants*
Conclusion

- EM for Mixture Models: a very simple form of EM, which can be applied to many different problems, using mixture of probability distributions specific to each problem,
- Extensions and variations can be derived from this simple algorithm, allowing for instance to take into account several kinds of data simultaneously, using sophisticated probabilistic graphical models.
References


McLachlan, G. J. and Peel, D.


