Application of the Minimum Description Length Principle to Graphical Models

Anthony Almudevar
Department of Biostatsitics and Computational Biology
University of Rochester

Kolmogorov Complexity and Ideal MDL

Given a string of bits

a) 0010010010010010 01,
b) 0010001000010000100001, or
c) 0110010011010010100101

How do we characterize and measure *regularity?*

Kolmogorov Complexity:

The length of the shortest program that prints the sequence.

Invariance:

The length of two shortest programs written in distinct programming languages differ by no more than a constant C, which is not dependent on the string length (Kolmogorov 1965, Chaitin 1969, Solomonoff 1964).

Ideal and Practical MDL

- This gives an objective measure of regularity.
- Unfortunately, it can be shown that there cannot exist a computer program which can calculate Kolmogorov Complexity for every set of data (Li and Vitanyi 1997).
- This leads to the *practical MDL principle* (Rissanen 1978, 1983):
  - Suppose we have data *X* and model *M*.
  - *M* explains the regularity of *X*, therefore *X* can be encoded with the help of *M*.
  - We may also encode *M*.
  - Suppose the complete code is *L(X, M)*.
  - The model *M** which best explains the regularity in *D* is the one resulting in the code *L(X, M*)* with the shortest length.
  - Note that codes must be invertible (decodable).

Poem 1

Me, me and none but me, dart home, O gentle Death,
And quickly, for I draw too long this idle breath.
O how I long till I may fly to heav'n above,
Unto my faithful, unto my faithful beloved turtle dove.

Like to the silver swan, before my death I sing,
And yet alive my fatal knell I help to ring.
Still I desire from earth and earthly joys to fly,
He never happy liv'd, never happy liv'd that cannot love to die.

16th century, anonymous author

Poem 2

I like rats, cats, bats and hats.
I like bogs, hogs, logs and dogs.
I like mice, rice, lice and dice.
I like figs, jigs, wigs and pigs.

I like cans, fans, pans and tans.
I like hens, pens, dens and tens.
I like rocks, docks, locks and socks.
I like dads, fads, lads and pads.

21st century, author in hiding

I like rats, cats, bats and hats.
I like bogs, hogs, logs and dogs.
I like mice, rice, lice and dice.
I like figs, jigs, wigs and pigs.

I like cans, fans, pans and tans.
I like hens, pens, dens and tens.
I like hips, lips, tips and sips.
I like dads, fads, lads and pads.

Code consists of
1) Template with substitution characters
2) Lengths of substitute words
3) Data
Segments of code are delimited by %

Poem is reassembled by repeating template with data substitutions until data is exhausted.

Model consists of template and word lengths

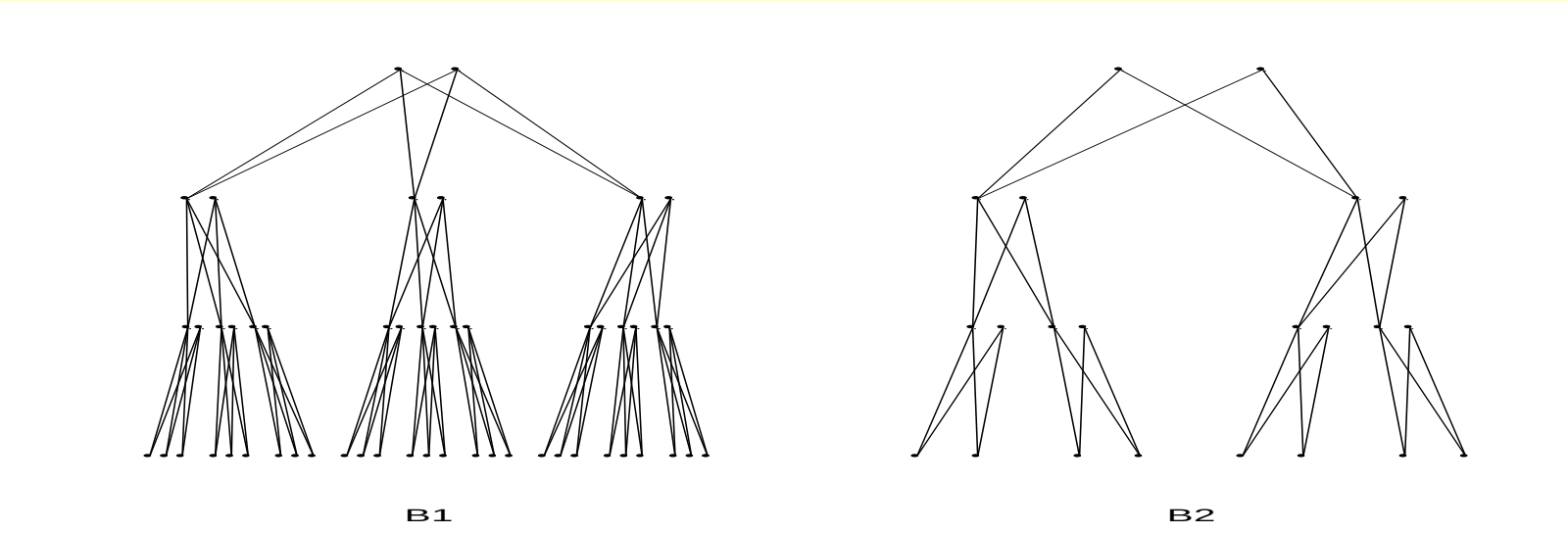
I like *a*z, *b*z, *c*z and *d*z. %
abcd1z3 %
rcbhatsbhldogsmrldicefjwpigscfptanshpdtenshltsipsdflpads %

Poem = 8 lines x 33 characters/line = **264 characters**

Compressed poem =

3 delimiters (%)
33 characters for template
7 characters for substitution code lengths
+ 56 characters for data (8 lines x 7 characters/line)
99 characters

Pedigree Reconstruction



- Suppose, given a family tree *M*, we have genotype data *X* for each individual.
- The conditional distribution of offspring genotypes given parental genotypes is well known (ie Mendelian Laws) and useful for the inference of parent-offpring relationships.
- Given data *X*, can we use the MDL principle to determine the family tree (model) *M* with useful accuracy?

MDL and the Bayesian Analogy

- The MDL principle is often expressed ("+" = concatenation)
  - $L(X, M) = L(X \mid M) + L(M)$ where
    - $L(X \mid M)$ is the data coded with the benefit of M
    - $L(M)$ is the encoded model.
- This resembles a Bayesian posterior density
  - $g(m|x) = (1/2)^{-L(x \mid m)} (1/2)^{-L(m)}$
- This is known as "crude" MDL. In fact, it can be shown that, formulated this way, the code length can be minimized, in an average sense, by setting $-L(X \mid M)$ = log likelihood.
- This leaves the problem of finding the minimum coding for model M, which then functions as a prior density.

Bayesian Averaging

- Posterior density for model m, data X
  - $g(m \mid X) = g(X \mid m)g(m)$
  - $g(X \mid m)$ is data conditional on model (likelihood)
  - $g(m)$ is prior density
- Inference is of the form (say, using MCMC)
  - $\mathrm{P}(m \in E \mid X) = \sum_{m \in E} g(X \mid m)g(m)$

Coding a Graph

- To code one of k objects requires no more than $\log(k)$ bits.
- The code length may be expressed in the number of *labels* required to code the graph, which identifies a node label and requires $\log(n)$ bits for n nodes.
- We may code a graph by coding for each node the set of parents.
- Suppose for node i, there are $n(k,i)$ possible parent sets of size k.
- To code the graph requires

$$L(M) = \Sigma_i \, l(k) + \log(\, n(\, | \, \mathrm{Pa}_i \, | , \, i) \,)$$

bits where Pa_i is the parent set of i, and $l(k)$ is the number of bits required to code k, which depends on any restrictions on $| \, \mathrm{Pa}_i \, |$.

- Current work (Almudevar 2007) shows that under ergodic type conditions

$$L(M) = N_E \log(n) + o(N_E \log(n))$$

where N_E is the number of edges.

Block Coding

- An *n*-graph can be expressed as a 0-1 *n* x *n* adjacency matrix.
- This can always be partitioned into *blocks,* defined here as any submatrix consisting only of 1's.
- The graph can then be coded by identifying all blocks.
- To identify a block, we only need to identify the subsets of edges which make up the "sides" of the block.
- One of these edges is included in both sides, and need be specified only once.
- Current work (Almudevar 2007) shows that a graph can be coded based on blocks which achieves the efficiency implied by the blocks, that is, let

$$B(M) = \Sigma_b R_b + C_b - 1$$

where the summation is over all block b, R_b is the number of rows of block b, and C_b is the number of columns of block b. Then a code exists which achieves code length

$$L(M) = B(M)\log(n) + o(B(M)\log(n))$$

Example

1	1	0	1	0	0
1	1	0	1	0	0
1	1	0	1	0	0
0	0	0	0	0	0
0	0	0	0	0	1
0	0	0	0	0	1

The matrix can be decomposed into 2 blocks, one 3 x 3, the other 2 by 1.

Edge code: $L(M) = 11 \log(n) + \varepsilon$

Block code: $L(M) = (3+3-1)\log(n) + (2+1-1)\log(n) + \varepsilon$

$= 7 \log(n) + \varepsilon$

•The edge code is additive across blocks.

•Also, within a block, the block code never requires more labels, and if both sides are > 1 requires strictly fewer, that is

$$R_b + C_b - 1 \leq R_b C_b$$

•Any matrix can be partitioned into disjoint blocks.

•*Therefore, asymptotically, the block code is never longer than the edge code, and sometimes strictly shorter.*

Bayesian Implication

- Prior is $g(m) = (1/2)^{-L(m)}$
- Suppose for models M_1, M_2

  $$L(M_1) < L(M_2),$$

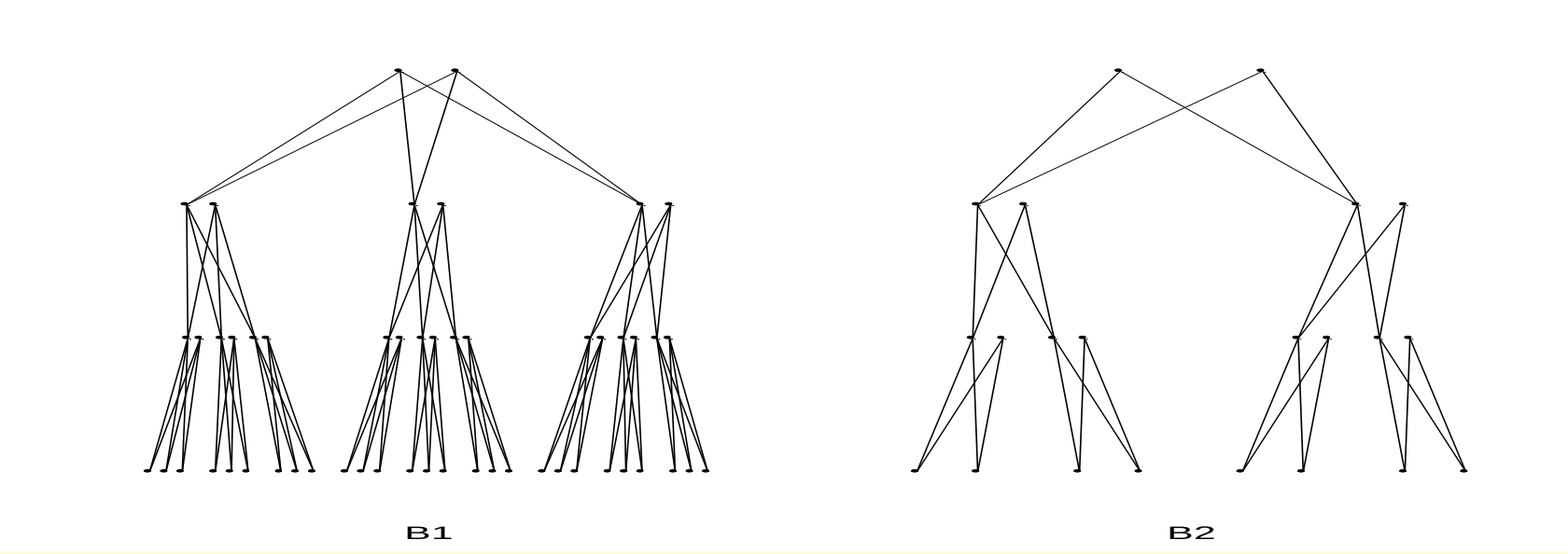
  and that there exists a binary string B of length $L(M_1)$ to which no model is assigned.
- Then the code is not efficient, since we can always assign M_2 to B.
- On the other hand if the code is efficient, then
  - $L(M)$ serves as an index of model complexity
  - $g(m)$ is a "density", that is the frequency of models with that complexity.
- Then $g(m)$, interpreted as a prior density on the model space, induces a marginal uniform prior on model complexity.

Scale Invariance

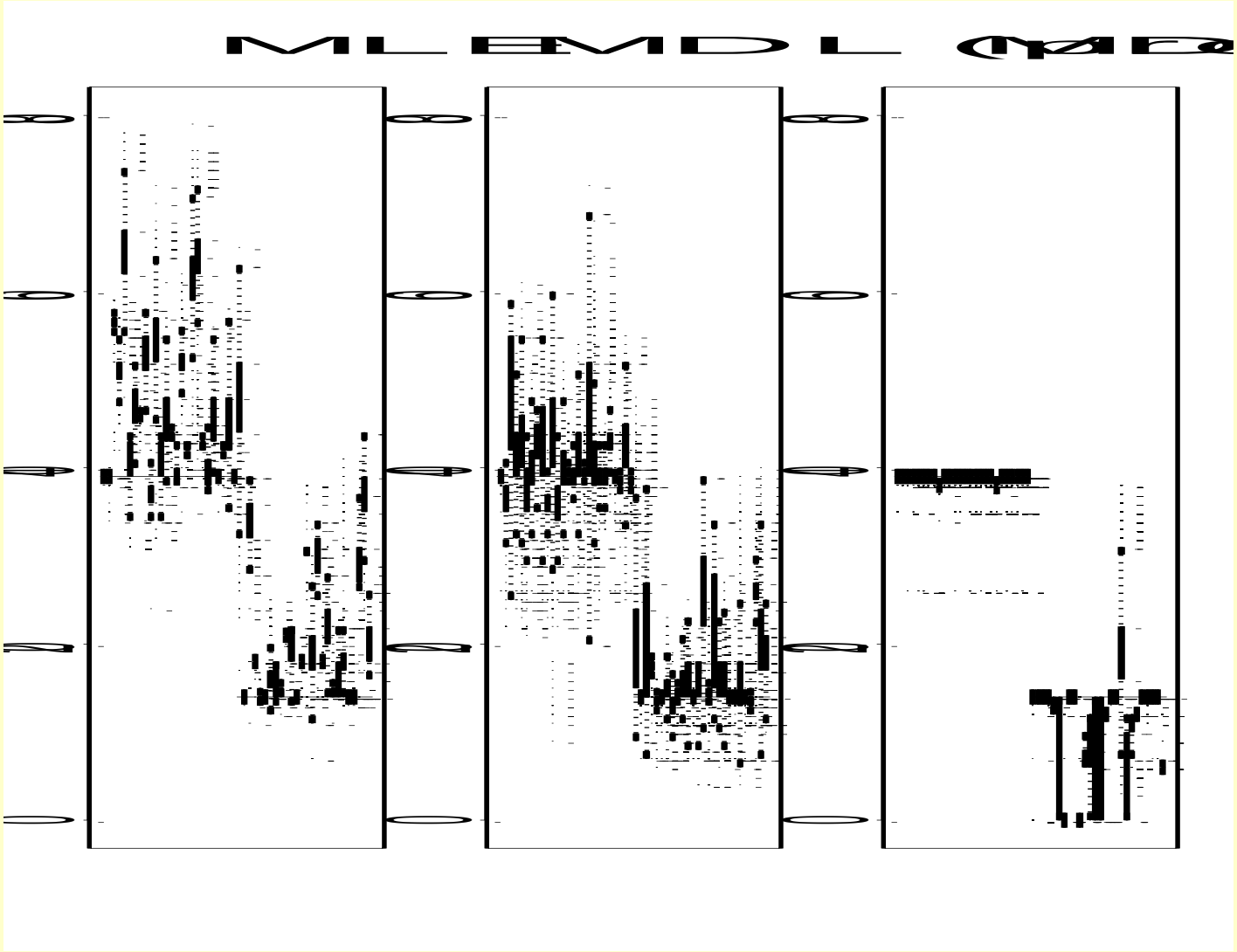
- If we wish to choose the best estimator $T(X)$ for the mean μ of a normal random variable with known variance, we could argue that $T(X) \equiv 0$ should be considered, since no other estimator could do as well when μ really is 0.
- One method of settling on a rational choice is to adopt an **invariance principle**.
- We may insist that any inference be, for example, location-scale invariant, reasoning that any inference should not depend which units are used (feet or inches, Celsius or Fahrenheit). When we do so, we are able to conclude that the sample mean is the best choice for $T(X)$.

Scale Invariance cont'd

- It can be shown that a prior $g(m)$ based on either the edge code or the block code has a **scale invariance** property.
- The local characteristics of certain types of graphs should not depend on the graph size.
- In a pedigree, the offspring numbers or mating behavior should not depend on the size of the sample.
- Therefore, we should require that under $g(m)$ the marginal distribution of these local characteristics for a fixed individual should not depend on the graph size.
- This will be true of a prior based on the edge or block code, but will not be true of most other commonly used priors (uniform, BIC, etc).

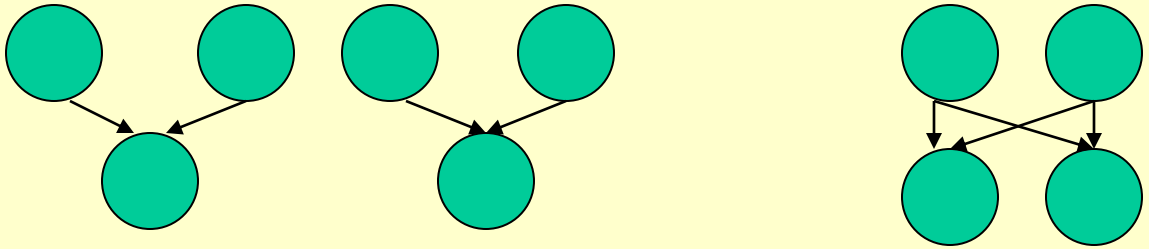


- Recall pedigrees B1 (founders with 39 descendents) B2 (founders with 14 descendents)
- Replicate these pedigrees to give n=1950 total individuals. Simulate genotypic data.
- Use
  - 1) Likelihood (uniform posterior)
  - 2) Partial MDL (likelihood code with edge code posterior)
  - 3) Full MDL (empirical entropy code with block code posterior)
- Generate posterior densities of number of descendents using MCMC.



Boxplots represent marginal posterior densities (Bayesian averaging) of descendant numbers of selected founders from pedigree types B1 and B2.

Implications of Block Code

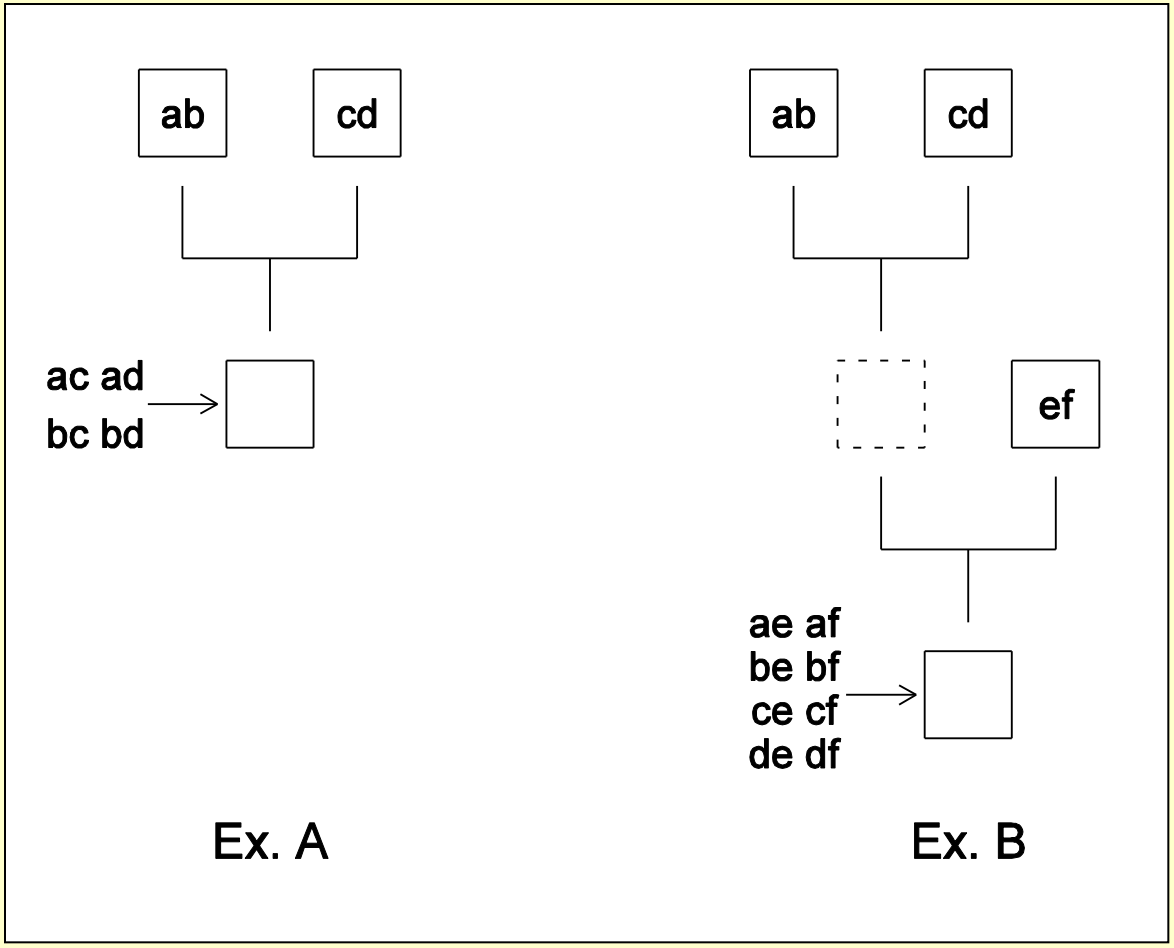


A B

- An edge code assigns the same code length to each pedigree structure (4 labels).
- A block code requires 4 labels to code structure A but only 3 labels to code structure B.
- The block code in general tends to ‘reward’ family structure.

Full MDL Approach

- Under the full MDL approach no probability model is used, but we have retained
  - $L(X \mid M)$ = -log likelihood
- as an approximation, based on coding theory.
- Is there some advantage to dispensing entirely with the likelihood?
- Can we express $L(X \mid M)$ as a code length without the probability model?



To code an offspring genotype, we make use of the parent genotypes.

This is much more efficient then coding the genotypes without this information.

In this case, the dashed line represents a missing individual, whose presence we wish to infer.

Ex. A: The genotype must be one of 4 types, so we only need 2 bits to code it. Add 2 to $L(X \mid M)$.

Ex. B: The genotype must be one of 8 types, so we only need 3 bits to code it. Add 3 to $L(X \mid M)$.

Thank you …