Inferring Models of cis-Regulatory Modules using Information Theory

BMI/CS 776

www.biostat.wisc.edu/bmi776/

Spring 2015

Colin Dewey

cdewey@biostat.wisc.edu

- Biological Question
  - What is causing differential gene expression?
- Goal
  - Find regulatory motifs in the DNA sequence.
- Solution
  - FIRE (Finding Informative Regulatory Elements)

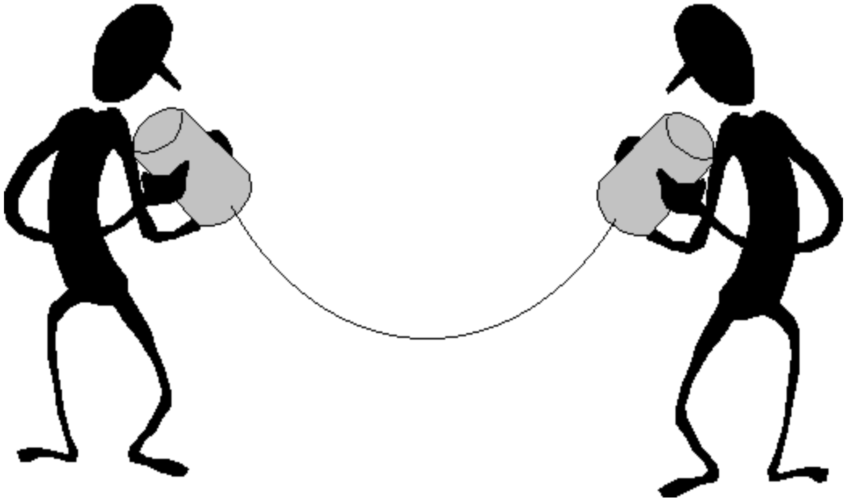
Goals for Lecture

the key concepts to understand are the following

- Entropy
- Mutual information (MI)
- Motif logos
- Using MI to identify CRM elements

Information Theory Background

- Problem
  - Create a code to communicate information
- Example
  - Need to communicate the manufacturer of each bike



Information Theory Background

- Four types of bikes
- Possible code

type	code
Trek	11
Specialized	10
Cervelo	01
Serrota	00

- expected number of bits we have to communicate: 2 bits/bike

Information Theory Background

- Can we do better?
- YES, if the bike types aren't equiprobable

Type/probability	# bits	code
$P(\text{Trek}) = 0.5$	1	1
$P(\text{Specialized}) = 0.25$	2	01
$P(\text{Cervelo}) = 0.125$	3	001
$P(\text{Serrota}) = 0.125$	3	000

- optimal code uses $-\log_2 P(c)$ bits for event with probability $P(c)$

Information Theory Background

Type/probability	# bits	code
$P(\text{Trek}) = 0.5$	1	1
$P(\text{Specialized}) = 0.25$	2	01
$P(\text{Cervelo}) = 0.125$	3	001
$P(\text{Serrota}) = 0.125$	3	000

- expected number of bits we have to communicate: 1.75 bits/bike

$$-\sum_{c=1}^{|C|} P(c)\log_2 P(c)$$

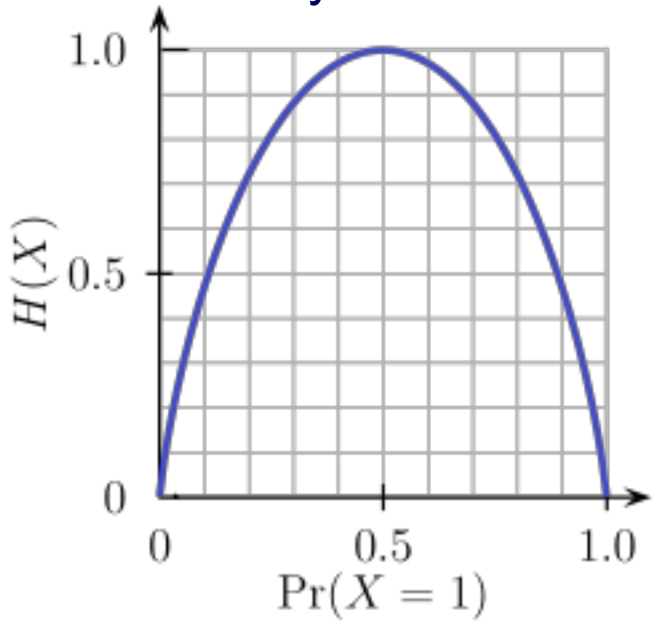
Entropy

- entropy is a measure of uncertainty associated with a random variable

- can be interpreted as the expected number of bits required to communicate the value of the variable

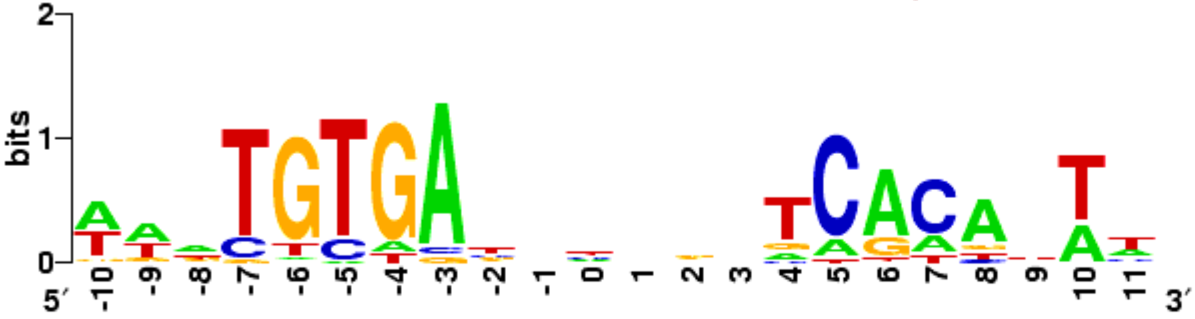
$$H(C) = -\sum_{c=1}^{|C|} P(c) \log_2 P(c)$$

entropy function for binary variable



How is entropy related to DNA sequences?

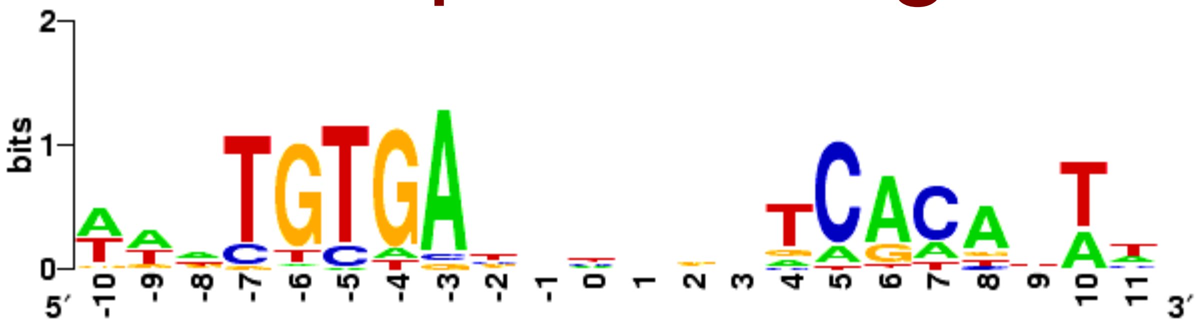
Sequence Logos



- Typically represent a binding site

- Height of each character c is proportional to $P(c)$

Sequence Logos



- height of logo at a given position determined by decrease in entropy (from maximum possible)

$$H_{\max} - H(C) = \log_2 N - \left(-\sum_{c} P(c) \log_2 P(c) \right)$$

# of characters in alphabet

Mutual Information

- *mutual information* quantifies how much knowing the value of one variable tells about the value of another

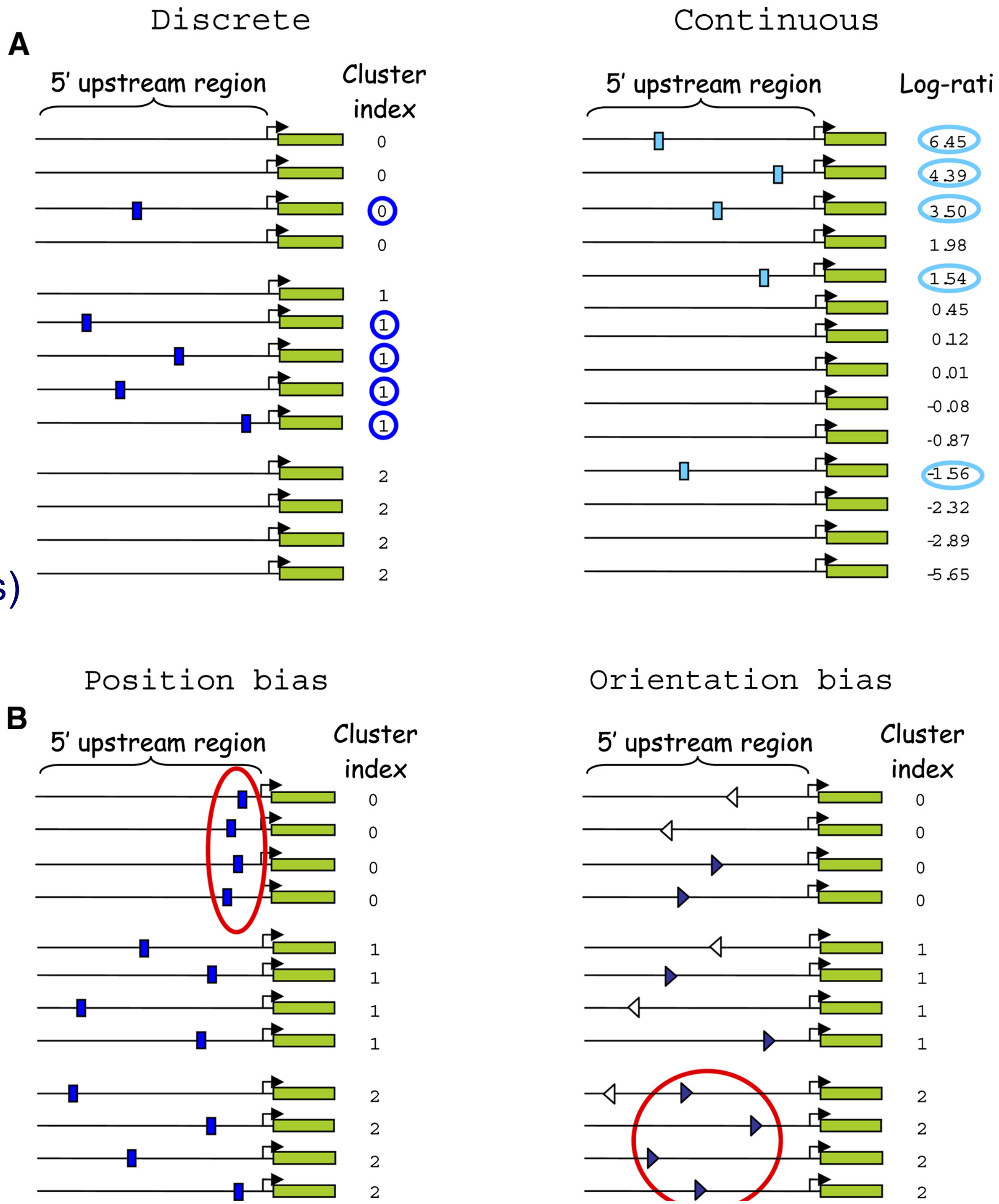
$$I(M;C) = \overset{\substack{\text{entropy of M} \\ \downarrow}}{H(M)} - \overset{\substack{\text{entropy of M} \\ \text{conditioned on C} \\ \swarrow}}{H(M \mid C)}$$

$$= \sum_{m} \sum_{c} P(m,c) \log_2 \left(\frac{P(m,c)}{P(m)P(c)} \right)$$

FIRE

Elemento et al., *Molecular Cell* 2007

- **Given** a set of sequences grouped into clusters
- **Find** motifs, and relationships, that have high *mutual information* with the clusters
- (also can do this when sequences have continuous values instead of cluster labels)



Mutual Information in FIRE

- we can compute the mutual information between a motif and the clusters as follows

$$I(M;C) = \sum_{m=0}^{1} \sum_{c=1}^{|C|} P(m,c) \log_2 \frac{P(m,c)}{P(m)P(c)}$$

$m=0, 1$ represent absence/presence of motif

c ranges over the cluster labels

Finding Motifs in FIRE

- motifs are represented by regular expressions; initially each motif is represented by a strict *k*-mer (e.g. TCCGTAC)

1. test all *k*-mers (*k*=7 by default) to see which have significant mutual information with the cluster label

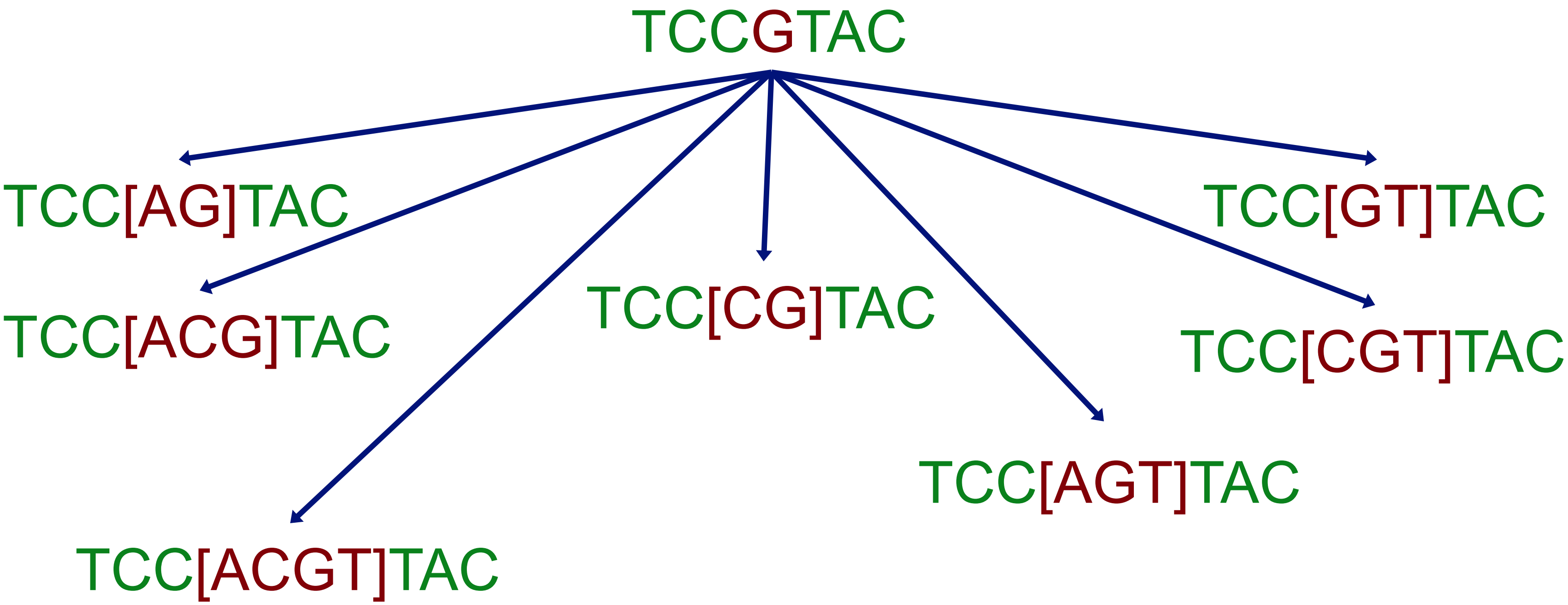
2. filter *k*-mers using a significance test

3. generalize each *k*-mer into a motif

4. filter motifs using a significance test

Key Step in Generalizing a Motif in FIRE

- randomly pick a position in the motif
- generalize in all ways consistent with current value at position
- score each by computing mutual information
- retain the best generalization



Generalizing a Motif in FIRE

```
given: k-mer, n

best ← null
repeat n times
    motif ← k-mer
    repeat
        motif ← GeneralizePosition(motif)    // shown on previous slide
    until convergence (no improvement at any position)
    if score(motif) > score(best)
        best ← motif

return: best
```

Generalizing a Motif in FIRE: Example

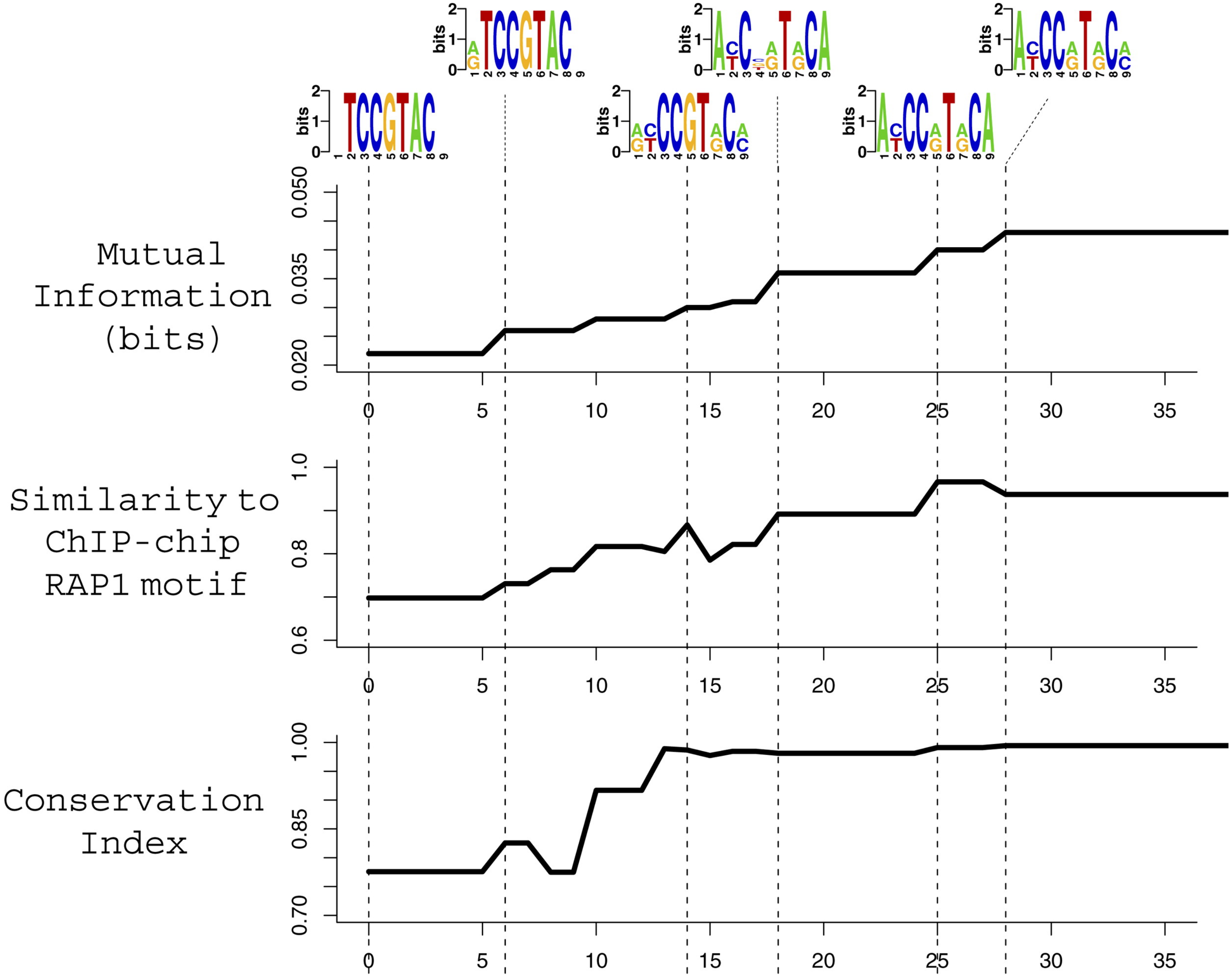


Figure from Elemento et al. *Molecular Cell* 2007

Characterizing Predicted Motifs in FIRE

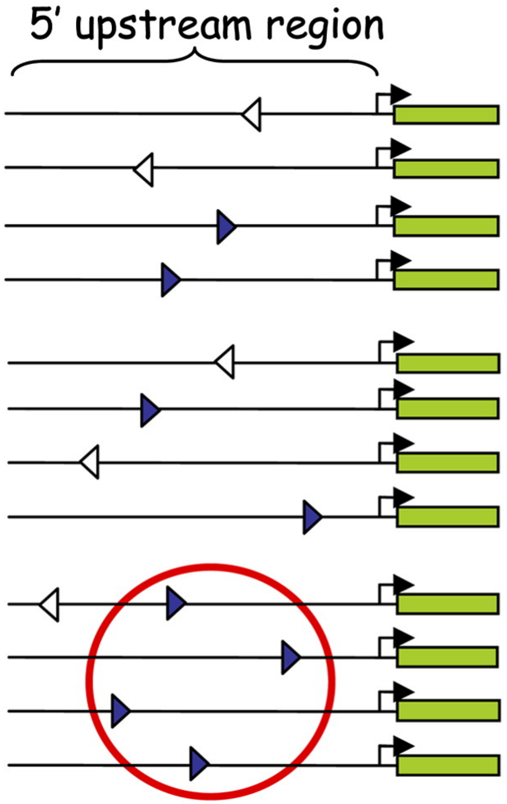
- mutual information is also used to assess various properties of found motifs
  - orientation bias
  - position bias
  - interaction with another motif

Using MI to Determine Orientation Bias

$I(S;C)$ C indicates cluster

S=1 indicates motif present on transcribed strand

S=0 otherwise (not present or not on transcribed strand)



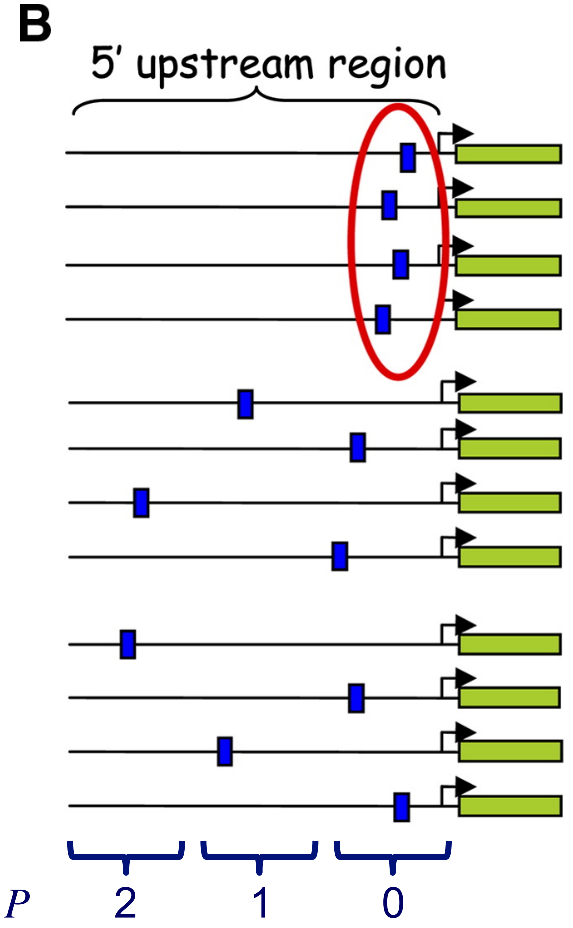
C	S
0	0
0	0
0	1
0	1
1	0
1	1
1	0
1	1
2	1
2	1
2	1
2	1

also compute MI where S=1 indicates motif present on complementary strand

Using MI to Determine Position Bias

$I(P;O)$ P ranges over position bins

O=0, 1 indicates whether or not the motif is overrepresented in a sequence’s cluster

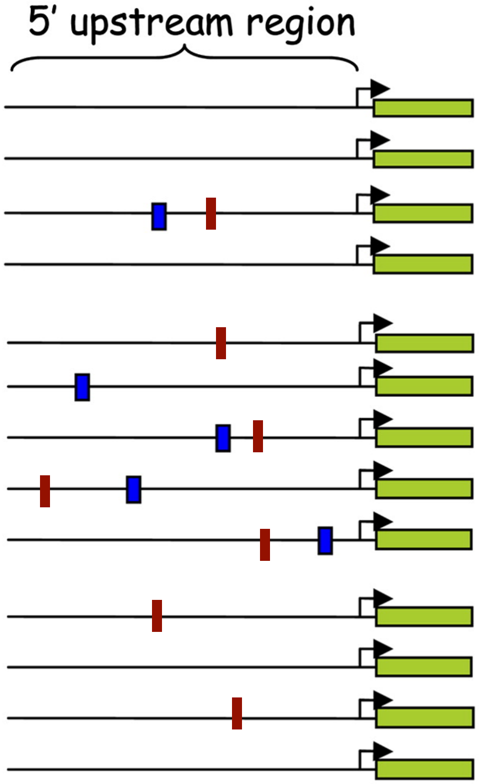


P	O
0	1
0	1
0	1
0	1
1	0
0	0
2	0
0	0
2	0
0	0
1	0
0	0

only sequences containing the motif are considered for this calculation

Using MI to Determine Motif Interactions

$I(M_1;M_2)$ $M_1=0, 1$ indicates whether or not a sequence has the motif **and** is in a cluster for which the motif is overrepresented; similarly for M_2



M_1	M_2
0	0
0	0
0	0
0	0
0	1
1	0
1	1
1	1
1	1
0	1
0	0
0	1
0	0

Discussion of CRM Finding Methods

- FIRE
  - mutual information used to identify motifs and relationships among them
  - motif search is based on generalizing informative *k*-mers

- in contrast to many motif-finding approaches, both CRM methods take advantage of *negative* sequences

- FIRE returns all informative motifs/relationships found, whereas the Noto & Craven approach returns single discriminative model