

DME: Discriminating Matrix Enumerator

Andrew D Smith

Cold Spring Harbor Laboratory

Purpose of DME

- DME discovers motifs that distinguish two sets of sequences
- Motifs are represented by position-weight matrices
- The best distinguishing motifs are those that are the most *relatively overrepresented* in one set vs the other
- The input is therefore 2 sets of sequences:
 - Foreground (FG) where we want to find the motif
 - Background (BG) where we hope it does not occur
- Intuition behind relative overrepresentation:
 - More occurrences in FG than in BG
 - Occurrences in FG are stronger (match better with the motif) than in BG

Scoring motifs in DME

- Relative overrepresentation formulated as likelihood model
- DME score function is based on the Likelihood model
- The optimal motifs under the DME score are approximately those for which the likelihood is maximized
- Let FG_{sub} be the set of substrings of sequences in FG with width same as desired motif width, similarly for BG_{sub} .
- DME score for matrix M and sequence sets FG and BG:

$$\sum_{x \in FG_{\text{sub}}} \max(\text{score}(M, x), 0) - \sum_{y \in BG_{\text{sub}}} \max(\text{score}(M, y), 0)$$

- Those x (and y) for which $\text{score}(M, x) > 0$ are **occurrences**

Log scoring and occurrences

- For matrix M and substring s , how is $\text{score}(M, s)$ calculated?
- First transform M into the log-score matrix L
- Let f be the vector of frequencies of bases in FG and BG
- Entry L_{ij} is given by

$$L_{ij} = \log \left(\frac{M_{ij}}{f_j} \right)$$

- Values of 0 in M or f are corrected (e.g. add small constant)

	Example matrix		Corresponding log-score matrix
A	$\begin{bmatrix} 0.01 & 0.88 & 0.02 & 0.20 & 0.00 & 0.00 \end{bmatrix}$	$\Rightarrow$	$\begin{bmatrix} -4.8 & 1.8 & -3.0 & -0.26 & -7.1 & -7.1 \end{bmatrix}$
C	$\begin{bmatrix} 0.97 & 0.05 & 0.74 & 0.32 & 0.00 & 0.02 \end{bmatrix}$		$\begin{bmatrix} 1.9 & -2.1 & 1.6 & 0.37 & -7.1 & -3.4 \end{bmatrix}$
G	$\begin{bmatrix} 0.00 & 0.01 & 0.18 & 0.42 & 0.00 & 0.97 \end{bmatrix}$		$\begin{bmatrix} -7.1 & -3.9 & -0.4 & 0.76 & -7.1 & 1.9 \end{bmatrix}$
T	$\begin{bmatrix} 0.01 & 0.04 & 0.04 & 0.04 & 1.00 & 0.01 \end{bmatrix}$		$\begin{bmatrix} -3.9 & -2.5 & -2.5 & -2.49 & 2.0 & -4.8 \end{bmatrix}$

Log scoring and occurrences

- Example of scoring a substring:

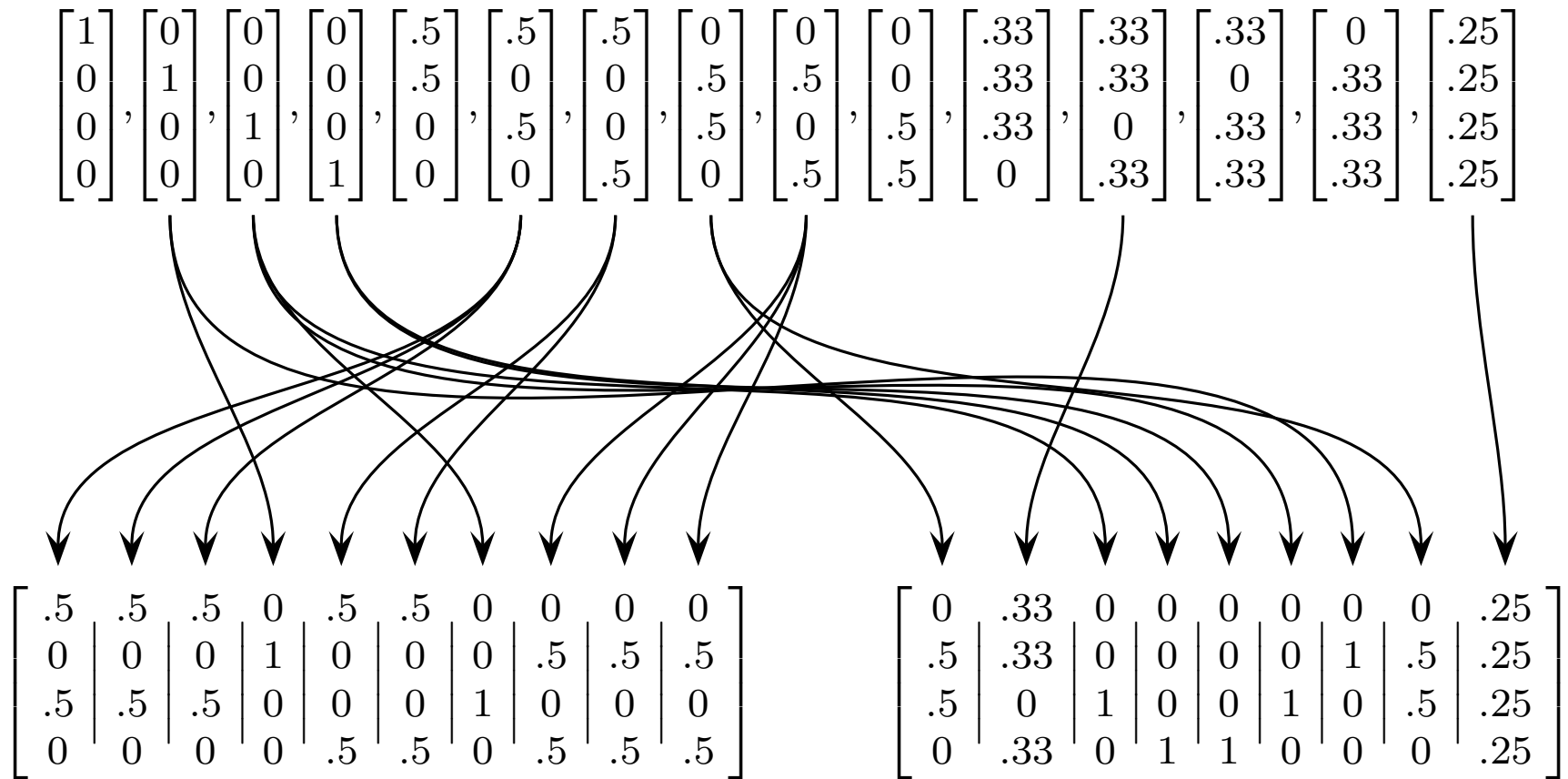
$$\begin{array}{c}
 A \\
 C \\
 G \\
 T
 \end{array}
 \begin{bmatrix}
 -4.8 & 1.8 & -3.0 & -0.26 & -7.1 & -7.1 \\
 1.9 & -2.1 & 1.6 & 0.37 & -7.1 & -3.4 \\
 -7.1 & -3.9 & -0.4 & 0.76 & -7.1 & 1.9 \\
 -3.9 & -2.5 & -2.5 & -2.49 & 2.0 & -4.8
 \end{bmatrix}$$

GCTGTAAGCCAGCCTGTGGTGGCCTG **CAGCTG** CTGAACACTCTGTTGCTGTAAGCTGA

$$\text{score} = 1.9 + 1.8 - 0.4 + 0.37 + 2.0 + 1.9 = 7.57$$

Discrete sets of column types

- Use set of “column types” to build matrices:



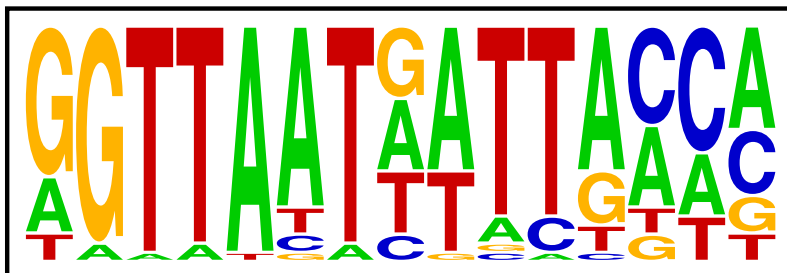
Discrete sets of column types

Logos for actual motifs and approximations made from column type set on previous slide

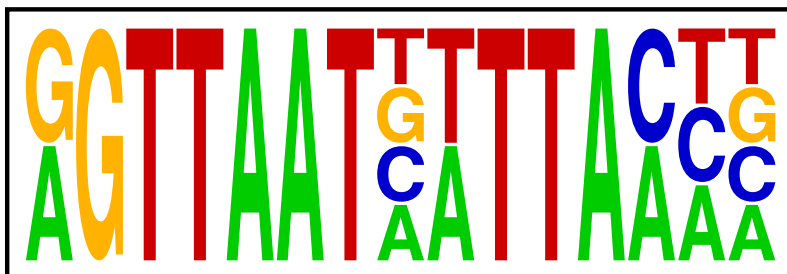
HNF-1

NF- κ B

Real motifs



Using
Column Types



Bound on information content

- DME requires the user bound information content of motifs
- Information content measures how conserved is a matrix
- Given a matrix M and a base composition f , information content is defined as:

$$I = \sum_{i=1}^w \sum_{j=A}^T M_{ij} \log(M_{ij}/f_j),$$

- In random sequences, matrices with lower information content have more occurrences
- Matrices with lower information content, in general, have higher scores
- User sets lower bound on information content of solution matrix to control expected frequency of occurrences of motif

Example

- Foreground: CATAGC $\Rightarrow$ CATAGC, GCTATG
- Background: CATGCA $\Rightarrow$ CATGCA, TGCATG
- Desired motif width = 3, sets of substrings to be considered:

Foreground: $\left\{ \begin{array}{ccc} \text{CAT} & \text{ATA} & \text{TAG} \\ \text{AGC} & \text{GCT} & \text{CTA} \\ \text{TAT} & \text{ATG} & \end{array} \right\}$ Background: $\left\{ \begin{array}{ccc} \text{CAT} & \text{ATG} & \text{TGC} \\ \text{GCA} & \text{TGC} & \text{GCA} \\ \text{CAT} & \text{ATG} & \end{array} \right\}$

- Base composition of FG + BG: (0.25, 0.25, 0.25, 0.25)
(not always uniform)

Example

- Foreground: CATAGC $\Rightarrow$ CATAGC, GCTATG
- Background: CATGCA $\Rightarrow$ CATGCA, TGCATG
- Desired motif width = 3, sets of substrings to be considered:

$$\text{Foreground: } \left\{ \begin{array}{ccc} \text{CAT} & \text{ATA} & \text{TAG} \\ \text{AGC} & \text{GCT} & \text{CTA} \\ \text{TAT} & \text{ATG} & \end{array} \right\} \quad \text{Background: } \left\{ \begin{array}{ccc} \text{CAT} & \text{ATG} & \text{TGC} \\ \text{GCA} & \text{TGC} & \text{GCA} \\ \text{CAT} & \text{ATG} & \end{array} \right\}$$

- Base composition of FG + BG: (0.25, 0.25, 0.25, 0.25)
(not always uniform)
- Column type set:

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} .5 \\ .5 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} .5 \\ 0 \\ .5 \\ 0 \end{bmatrix}, \begin{bmatrix} .5 \\ 0 \\ 0 \\ .5 \end{bmatrix}, \begin{bmatrix} 0 \\ .5 \\ .5 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ .5 \\ 0 \\ .5 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ .5 \\ .5 \end{bmatrix}$$

Example

- Foreground: CATAGC $\Rightarrow$ CATAGC, GCTATG
- Background: CATGCA $\Rightarrow$ CATGCA, TGCATG
- Desired motif width = 3, sets of substrings to be considered:

$$\text{Foreground: } \left\{ \begin{array}{ccc} \text{CAT} & \text{ATA} & \text{TAG} \\ \text{AGC} & \text{GCT} & \text{CTA} \\ \text{TAT} & \text{ATG} & \end{array} \right\} \quad \text{Background: } \left\{ \begin{array}{ccc} \text{CAT} & \text{ATG} & \text{TGC} \\ \text{GCA} & \text{TGC} & \text{GCA} \\ \text{CAT} & \text{ATG} & \end{array} \right\}$$

- Base composition of FG + BG: (0.25, 0.25, 0.25, 0.25)
(not always uniform)
- Column type set:

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} .5 \\ .5 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} .5 \\ 0 \\ .5 \\ 0 \end{bmatrix}, \begin{bmatrix} .5 \\ 0 \\ 0 \\ .5 \end{bmatrix}, \begin{bmatrix} 0 \\ .5 \\ .5 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ .5 \\ 0 \\ .5 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ .5 \\ .5 \end{bmatrix}$$

- Specified bound on information content (bits/column) = 1.5

Score and information content tables

2.00	-31.22	-31.22	-31.22	1.00	1.00	1.00	-31.22	-31.22	-31.22
-31.22	2.00	-31.22	-31.22	1.00	-31.22	-31.22	1.00	1.00	-31.22
-31.22	-31.22	2.00	-31.22	-31.22	1.00	-31.22	1.00	-31.22	1.00
-31.22	-31.22	-31.22	2.00	-31.22	-31.22	1.00	-31.22	1.00	1.00

$$\log \left(\frac{0 + \epsilon}{0.25} \right) = -31.22$$

Score table used to
calculate scores for matrices
during the search

Column Types

$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} .5 \\ .5 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} .5 \\ 0 \\ .5 \\ 0 \end{bmatrix}$	$\begin{bmatrix} .5 \\ 0 \\ 0 \\ .5 \end{bmatrix}$	$\begin{bmatrix} 0 \\ .5 \\ .5 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ .5 \\ 0 \\ .5 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ .5 \\ .5 \end{bmatrix}$
--	--	--	--	--	--	--	--	--	--

Base Composition

(0.25, 0.25, 0.25, 0.25)

Score and information content tables

2.00	-31.22	-31.22	-31.22	1.00	1.00	1.00	-31.22	-31.22	-31.22
-31.22	2.00	-31.22	-31.22	1.00	-31.22	-31.22	1.00	1.00	-31.22
-31.22	-31.22	2.00	-31.22	-31.22	1.00	-31.22	1.00	-31.22	1.00
-31.22	-31.22	-31.22	2.00	-31.22	-31.22	1.00	-31.22	1.00	1.00

$$\log \left(\frac{0.5}{0.25} \right) = 1$$

Score table used to
calculate scores for matrices
during the search

Column Types

$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} .5 \\ .5 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} .5 \\ 0 \\ .5 \\ 0 \end{bmatrix}$	$\begin{bmatrix} .5 \\ 0 \\ 0 \\ .5 \end{bmatrix}$	$\begin{bmatrix} 0 \\ .5 \\ .5 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ .5 \\ 0 \\ .5 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ .5 \\ .5 \end{bmatrix}$
--	--	--	--	--	--	--	--	--	--

Base Composition

(0.25, 0.25, 0.25, 0.25)

Score and information content tables

2.00	-31.22	-31.22	-31.22	1.00	1.00	1.00	-31.22	-31.22	-31.22
-31.22	2.00	-31.22	-31.22	1.00	-31.22	-31.22	1.00	1.00	-31.22
-31.22	-31.22	2.00	-31.22	-31.22	1.00	-31.22	1.00	-31.22	1.00
-31.22	-31.22	-31.22	2.00	-31.22	-31.22	1.00	-31.22	1.00	1.00

2 2 2 2 1 1 1 1 1 1

$$\log \left(\frac{1}{0.25} \right) = 2$$

Information content table
used to calculate bits/column
for matrices during search

Column Types

$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} .5 \\ .5 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} .5 \\ 0 \\ .5 \\ 0 \end{bmatrix}, \begin{bmatrix} .5 \\ 0 \\ 0 \\ .5 \end{bmatrix}, \begin{bmatrix} 0 \\ .5 \\ .5 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ .5 \\ 0 \\ .5 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ .5 \\ .5 \end{bmatrix}$

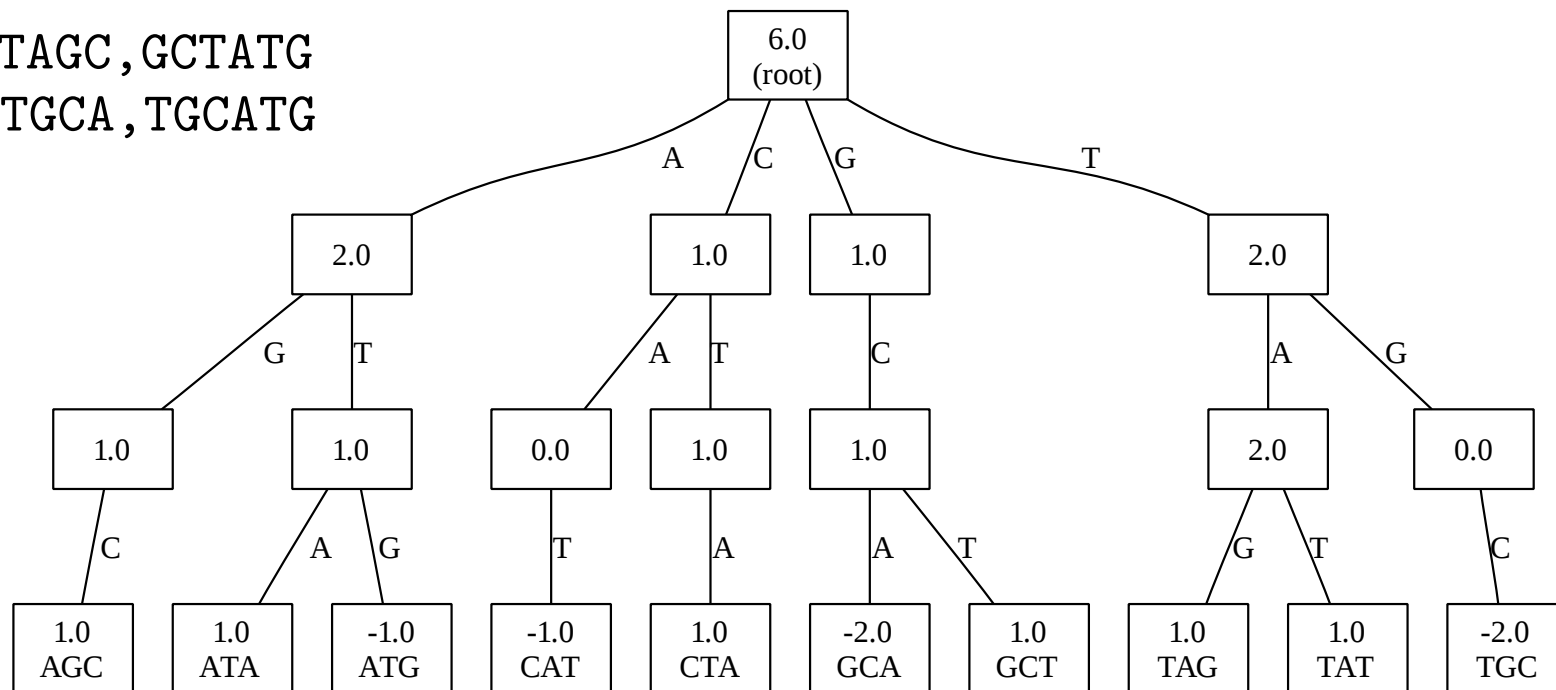
Base Composition

(**0.25**, 0.25, 0.25, 0.25)

Lexicographic trees for indexing

FG: CATAGC, GCTATG

BG: CATGCA, TGCATG



- Each substring of length 3 (for our example) is encoded
- All encoded strings spelled on some path from root to leaf
- Nodes store **maximum difference** values to indicate counts of substrings in foreground and background

The search strategy

- Search proceeds by extending matrix prefixes
- Add columns to current matrix until desired width is reached
- Only extend current matrix if it could lead to best solution
- If current matrix cannot lead to best solution, backtrack
- Backtracking: remove last column, and try a different one
- **Full solution:** matrix with desired width (3 in example)
- **Partial solution:** shorter matrices we try to extend
- Partial/full solutions are points in the search space
- Only full solutions are actually solutions
- For partial solution A and full solution B, if B is prefixed by A, B is a **full extension** of A

The search strategy

For each partial/full solution, must maintain:

Best score	Score of best full solution encountered so far (initial value is 0)
Surplus information	Difference between maximum bits/column of full extension of current matrix, and specified minimum bits/column of a solution
Upper bound	Upper bound on greatest score for some full extension of the current partial solution
Match score	For lexicographic tree nodes, match score is the value of the match between the path label of the node and the current partial solution
Frontier	For partial solution M , the set of lexicographic tree nodes whose path label is prefix of a string with match score > 0 w.r.t. a full extension of M

Representing points in search space

Current matrix:
(partial solution or full solution)

Upper bound on
best score for any
completion of current
partial solution

0	1	0	.33	bound: 12
0	0	0	0	bits: 0.75
0	0	.5	.33	best: 9.5
1	0	.5	.33	

Surplus information
Number of bits by
which some full extension
of current matrix exceeds
specified minimum

Best score attained
so far for a full
solution

When the search begins

- **Best score** obtained so far = 0 (no full solutions yet)
- Only node in **frontier** is the root, with empty string as label
- This partial solution has consumed no **surplus information**:

$$(\text{max bits/column} - \text{specified min}) \times (\text{full width} - \text{current width}) =$$

$$(2 - 1.5) \times (3 - 0) = 1.5$$

- **Upper bound** on score of best possible extension:

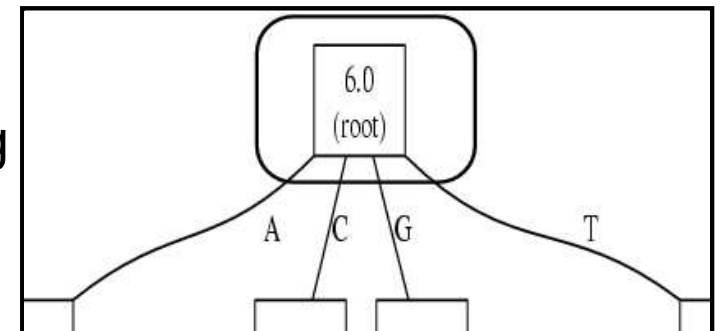
$$\sum_{x \in \text{frontier}} \text{max_diff}(x) (\text{match_score}(x) + (\text{max col. score} \times \text{remaining cols.})) =$$

$$6(0 + (2.0 \times 3)) = 36$$

Current
partial
solution

root (empty)	bound: 36
	bits: 1.5
	best: 0

Corresponding
frontier



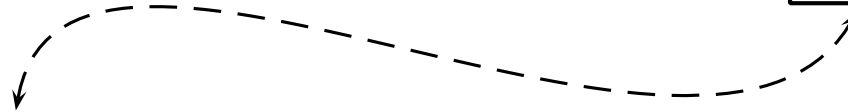
Example: first extension step

- Extend current (empty) matrix with first column type
- **Surplus info** unchanged: first column type has max
- No need to update **best score**:
Only do that for full solutions (width 3 in example)
- Update the frontier (next slide)
- Calculate new **upper bound** on score (2 ahead)

root	bound: 36
	bits: 1.5
	best: 0



1	bound: 12
0	bits: 1.5
0	best: 0
0	



Column Type Set:

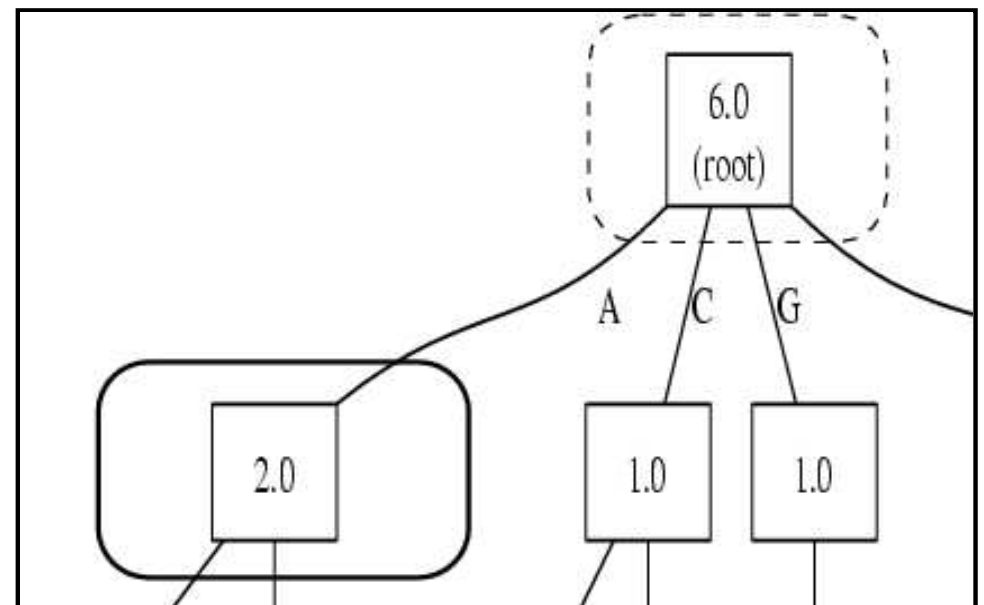
1	0	0	0	.5	.5	.5	0	0	0
0	1	0	0	.5	0	0	.5	.5	0
0	0	1	0	0	.5	0	.5	0	.5
0	0	0	1	0	0	.5	0	.5	.5

First frontier update

- One child of root has label scoring positively w.r.t. extended matrix
- Extended matrix has only one column: $(1, 0, 0, 0)^T$
- Scoring matrix entries for this column: $(2.0, -31.22, -31.22, -31.22)^T$
- Match score > 0 for child with path label A $\Rightarrow$ it goes in new frontier
- No others go in new frontier

Even in best case scenario
value of -31.22 cannot be overcome!

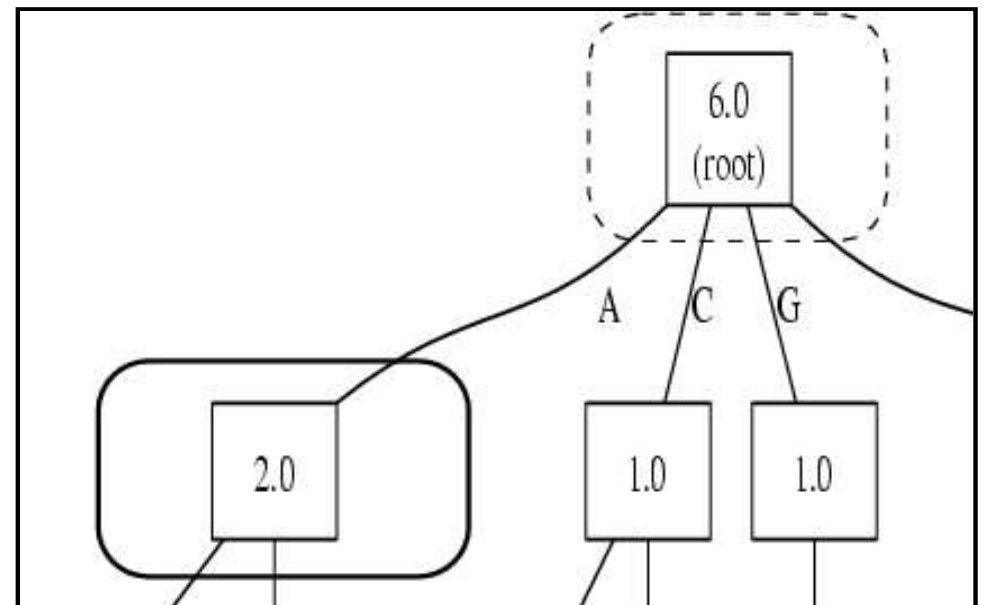
Descendants of those nodes can't have
positive match w.r.t. any further
extension of current matrix



Calculating the new upper bound

- Plug new values into formula for upper bound:

$$\sum_{x \in \text{frontier}} \text{max_diff}(x) (\text{match_score}(x) + (\text{max col. score} \times \text{remaining cols.})) =$$
$$2.0(2.0 + (2.0 \times 2)) = 12$$



A step extending to a full solution

- Previous frontier: single node, labeled with TA
- Extended partial solution with column $(0, 0, 0.5, 0.5)^T$:
 - Bits/col. of 1 below max $\Rightarrow$ subtract 1 bit/col., result = 0.5
 - Match scores > 0 for T and G children $\Rightarrow$ put in new frontier
- Completed matrix score: $1.0 \times \text{score}(\text{TAG}) + 1.0 \times \text{score}(\text{TAT}) = 10$

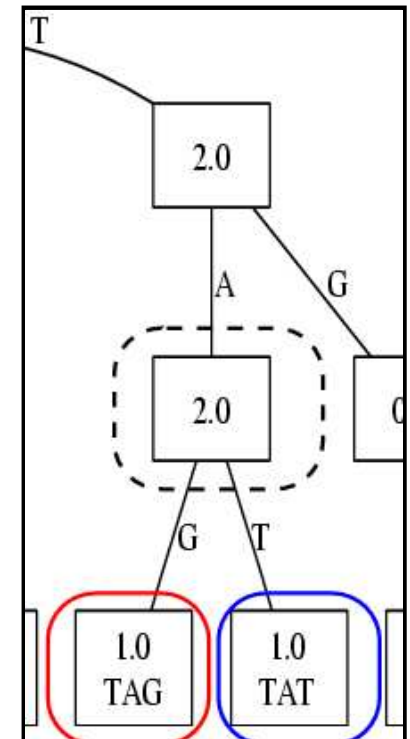
0	1	bound: 12	
0	0	bits: 1.5	
0	0	best: 6	
1	0		



0	1	0	bound: 10	
0	0	0	bits: 0.5	
0	0	.5	best: 10	
1	0	.5		

Corresponding scoring matrix:

-31.22	2.00	-31.22
-31.22	-31.22	-31.22
-31.22	-31.22	1.00
2.00	-31.22	1.00



A step extending to a full solution

- Previous frontier: single node, labeled with TA
- Extended partial solution with column $(0, 0, 0.5, 0.5)^T$:
 - Bits/col. of 1 below max $\Rightarrow$ subtract 1 bit/col., result = 0.5
 - Match scores > 0 for T and G children $\Rightarrow$ put in new frontier
- Completed matrix score: $1.0 \times \text{score}(\text{TAG}) + 1.0 \times \text{score}(\text{TAT}) = 10$

0	1	bound: 12
0	0	bits: 1.5
0	0	best: 6
1	0	

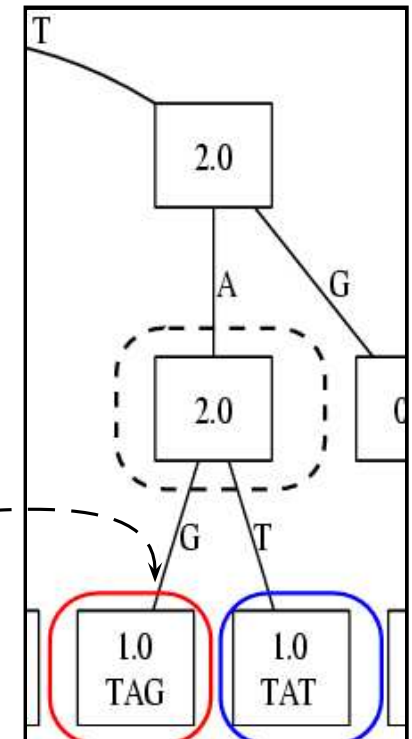
0	1	0	bound: 10
0	0	0	bits: 0.5
0	0	.5	best: 10
1	0	.5	

Corresponding scoring matrix:

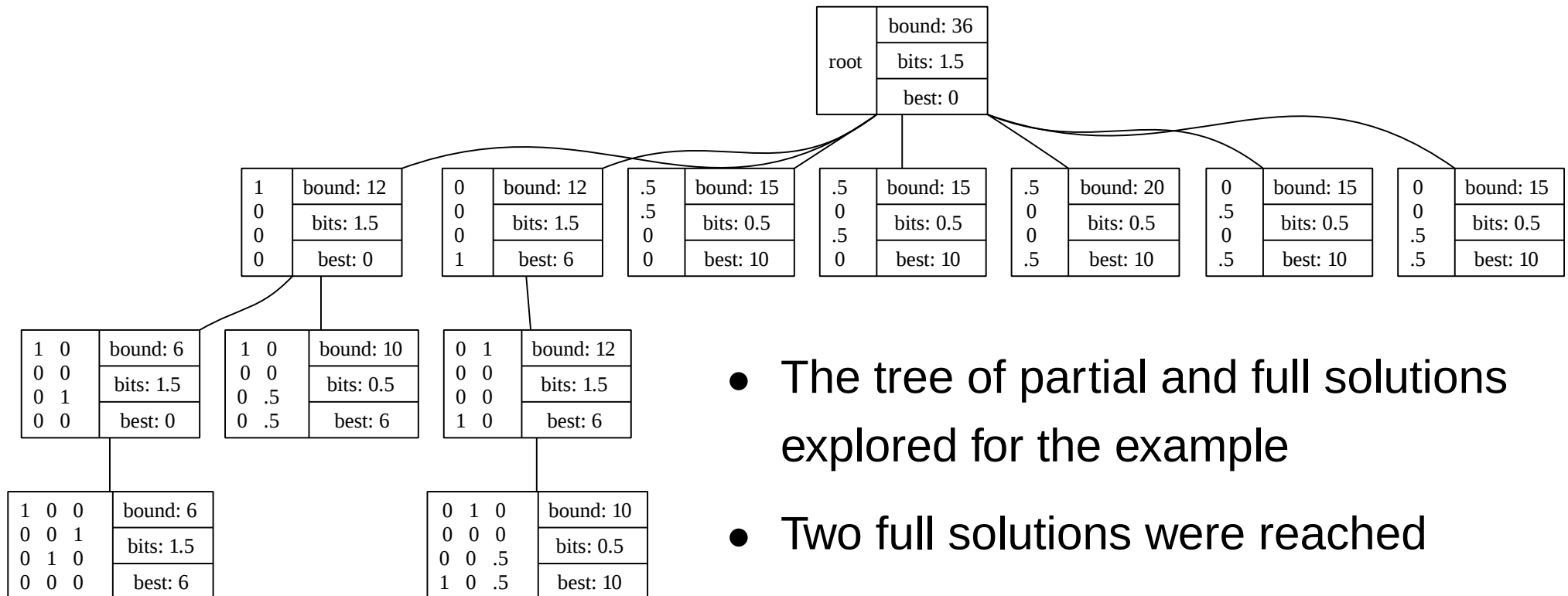
-31.22	(2.00)	-31.22
-31.22	-31.22	-31.22
-31.22	-31.22	(1.00)
(2.00)	-31.22	(1.00)

$$\text{score}(\text{TAG}) = \text{score}(\text{TA}) + 1 \times 1$$

$$\text{score}(\text{TAT}) = \text{score}(\text{TA}) + 1 \times 1$$



Implicit search tree for the example

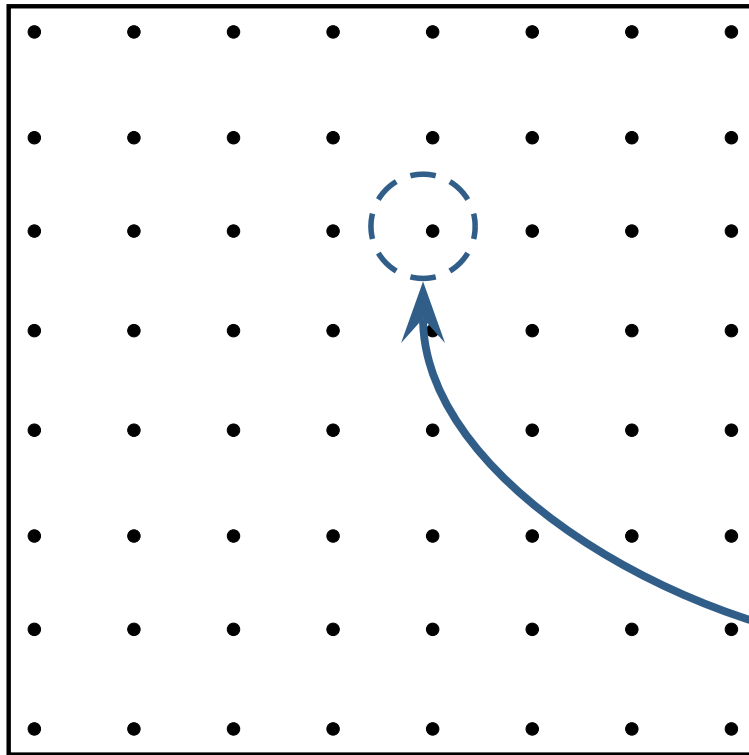


- The tree of partial and full solutions explored for the example
- Two full solutions were reached

- Each leaf at depth < 3 represents abandoned partial solutions
- Extensions not explored for 2 possible reasons:
 - Information content bound would have been violated
 - Upper bound on best extension score less than best so far

Motif Refinement

- Given optimal matrix built from original column types
- Make new set of column types “close to” columns of matrix



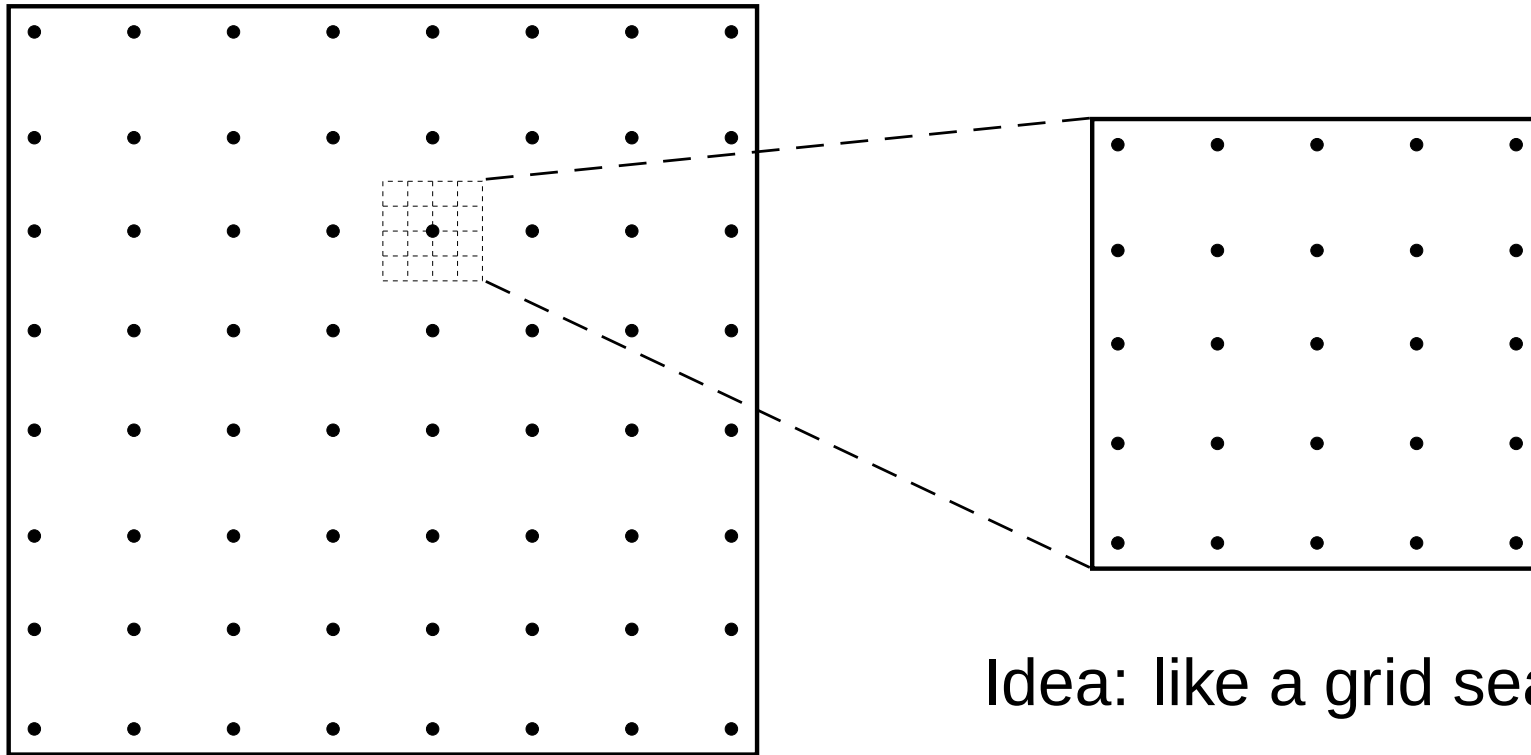
Square represents all
(continuous) matrices

Points represent matrices
built from column types

Highest scoring matrix

Motif Refinement

- Given optimal matrix built from original column types
- Make new set of column types “close to” columns of matrix



- Repeat the process $\Leftrightarrow$ zoom in on solution
- Different set of column types for each column of the original matrix

Details

- Keep frontier for all prefixes of current matrix (backtracking)
- Frontier as lists: repeatedly allocate/free $\Rightarrow$ too slow
- Instead, pre-allocate tables, store ptrs to frontier nodes
- DME can't use edge-compression in lexicographic trees
- Column types kept sorted according to bits/column:
If extending by a column type results in too few bits/column,
same will happen for all subsequent column types
- To search for more than one matrix, once a matrix is found
we remove all occurrences before searching for the next one

Reference

Smith AD, Sumazin P and Zhang MQ (2005)
Identifying tissue-specific transcription factor binding sites in
vertebrate promoters.

PNAS, 102(5):1560-1565