

# Motif Finding Problem.

$D$ : Alphabet set.  $\mathcal{C} \equiv \{A, T, C, G\}$

$l$ -mer: A sequence  $x_1, x_2, \dots, x_l$ , s.t.  $x_i \in \{A, T, C, G\}$   
(A sequence of  $l$ -nucleotides).

Let  $D \equiv \{A, T, C, G\}$   
An  $l$ -mer is an element of  $D^l$ .

Let  $Z$  be the set of  $t$ - $l$ -mers i.e.

Consensus score & pattern:  
 $Z = \{\bar{z}_1, \bar{z}_2, \dots, \bar{z}_t\}$  where,  $\bar{z}_i \in D^l$ .

Let  $m_j$  = maximally occurred alphabet at  $j$ th position over the set  $Z$ .

i.e.

$$\{z_{ij}, i=1, \dots, t\}$$

$C_j$  = no. of times  $m_j$  occurs.

$$\text{Consensus score}(Z) = \sum_{j=1}^l C_j$$

$$P = \text{Consensus Pattern}(Z) = m_1 m_2 \dots m_l$$

It may be noted that 'P' may not be unique.

Profile Matrix:

$$\begin{matrix} A \\ T \\ C \\ G \end{matrix} \begin{bmatrix} 1 & 2 & \dots & l \\ x & & & \end{bmatrix} \equiv [f_{ij}]$$

where,  $f_{ij}$  denotes no. of times  $i$ th nucleotide occurs at  $j$ th position of  $l$ -mer.

$$\therefore C_j = \max_{i \in \{A, T, C, G\}} \{f_{ij}\}$$

$$m_j = \underset{i \in \{A, T, C, G\}}{\text{argmax}} \{f_{ij}\}$$



R.G.

5 mer

$$Z = \begin{Bmatrix} A & T & C & C & A \\ G & G & G & C & A \\ A & T & G & G & A \\ T & A & G & C & T \\ A & T & G & G & C \end{Bmatrix} \quad \begin{array}{l} \text{t-no. of 5-mers.} \\ = 6 \end{array}$$

Profile Matrix  $(Z) =$

|   |               |               |               |               |
|---|---------------|---------------|---------------|---------------|
|   | $\frac{1}{2}$ | $\frac{2}{3}$ | $\frac{3}{4}$ | $\frac{4}{5}$ |
| A | 2             | 1             | 0             | 0             |
| T | 1             | 3             | 0             | 0             |
| C | 0             | 0             | 1             | 3             |
| G | 1             | 1             | 4             | 2             |

$p =$  A T G C A

Consensus Score  $(Z) = 2 + 3 + 4 + 3 + 3$   
 $= 15.$

Highest Possible Consensus Score:  $t + t + \dots + t = t \cdot l.$

Smallest " " " :  $\frac{t}{4} + \frac{t}{4} + \dots + \frac{t}{4} = \frac{t \cdot l}{4}$

Time complexity for CS:  $O(t \cdot l).$

Modif Finding Problem:

Let  $X =$  set of  $n$ -mers,  $n \gg l.$   
 $= \{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_t\}$

Let  $S = \{s_1, s_2, \dots, s_t\}, 1 \leq s_i \leq n-l+1$

denote  $s_i$  positions on each  $n$ -mers to extract an  $l$ -mer.

$l\text{-l}(X, S) \triangleq$   $t$   $l$ -mers



Motif = Consensus Pattern

$$S_m = \underset{S}{\operatorname{argmax}} \{ \text{Consensus-Score}(X, S) \}$$

$$\text{Motif} = \text{Consensus-Pattern}(S_m)$$

$$\text{Score} = \max_S \{ CS(X, S) \}$$

Problem statement: Given a set of DNA sequences, find a set of  $l$ -mers, one from each sequence, that maximises the consensus score.

⑦

Exhaustive approach:

→ Search over all possible 'S' to ~~maximize~~ maximise consensus score.

→ generation of all possible 'S' (Permutation).

```

for i = t to 1,
  if (s_i == n-l+1)
    s_i = 1;
  else
    s_i = s_i + 1;
  return S;
}
return S;
  
```

at →

Cyclic generation

$1, 1, \dots, 1, 1$   
 $1, 1, \dots, 1, 2$   
 $\vdots$   
 $1, 1, \dots, 1, n-l+1$   
 $1, 1, \dots, 2, 1$   
 $\vdots$   
 $n-l+1, n-l+1, \dots, n-l+1, 1$   
 $\vdots$   
 $n-l+1, n-l+1, \dots, n-l+1, 1$   
 $1, 1, \dots, 1, 1$

Similar to number generation of  $t$  digits with 'n-l+1' number of alphabets.

Total No. of elements =  $(n-l+1)^t$

→ Time complexity:  $O((n-l+1)^t \cdot t \cdot l) \approx O(l \cdot n^t \cdot t)$

→ Enumeration policy (Let  $S = s_1, s_2, \dots, s_t$ )

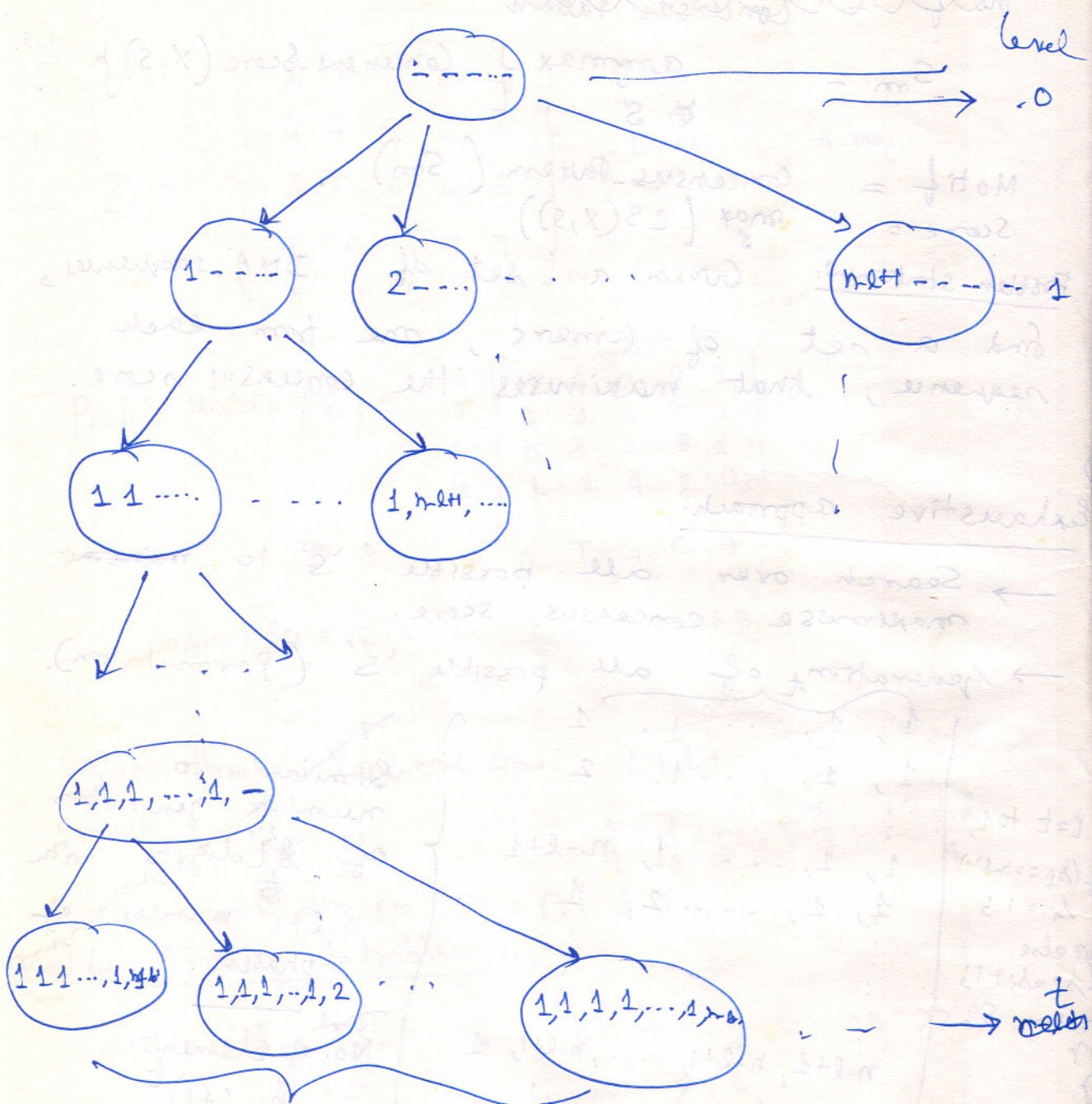
next element:

```

for i = t to 1,
  if (s_i == n-l+1)
    s_i = 1;
  else
    s_i = s_i + 1;
  return S;
  
```



# → Search Tree



leaf nodes (Generated <sup>as</sup> during number enumeration)

However it may be helpful to generate these elements following the search tree hierarchy. In such case, initial checking (with substring of 1-n) may mean, if the sub-tree may be pruned from the search space.



→ Search Tree Node Generator policy (Following DFS principle).

Level  $i$  : (First  $i$  digits & ignore ~~the last~~  $t-i$  digits).  
 $s_1, s_2, \dots, s_i$   
 ↓ if  $(i < t)$  else  
 Level  $(i+1)$  so that  $a_{(i+1)} = 1$ .  
 Node with enumeration of  ~~$s_1, s_2, \dots, (n-l+1)$~~  digits for the  $(n-l+1)$ th digit are with  $(n-l+1)$  go back to its parent level  $(i)$  and increase the value  $t$  so on.

Next vertex  $(\bar{a}, i, t, n-l+1)$

```

if (i < t)
{
  a_{i+1} ← 1
  return (ā, i+1)
}
else
{
  for j ← t to 1
  {
    if a_j < (n-l+1)
    {
      a_j ← a_j + 1
      return (ā, j)
    }
  }
  return (ā, 0)
}

```

$(\bar{a}, i)$

denotes only for 'i' digits are considered for  $\bar{a} \in [t-digits, t]$

Returns next search vertex.

→ Search steps:

1. Initialize  $S = (-, -, \dots, -) [\bar{a}, 0]$  (mostly)
2. If ~~it is~~ go to next node, if it is a leaf node compute consensus score & store the maximum score & consensus str.



## → Branch & Bound Strategies:

→ Partial score at intermediate node:

$$PS = CS (s_1, s_2, \dots, s_i, \dots)$$

+ Most optimistic score for the rest  $(t-i)$  str  
(=  $l \cdot (t-i)$ )

? If  $PS < \text{Current\_Max\_Score}$ , one ~~may~~ should  
not proceed further, reject the subtree &  
go for <sup>testing the</sup> next sibling (or ~~go back to parent~~  
~~node~~)

→ Bypassing strategies:

Bypass  $(\bar{a}, i, t, n-l+1)$

```
{
  for  $j \leftarrow 0$  to 1
  {
    if  $a_j < (n-l+1)$ 
    {
       $a_j \leftarrow a_j + 1$ 
      return  $(\bar{a}, j)$ 
    }
  }
  return  $(\bar{a}, 0)$ 
}
```

Returns  
next  
search  
vertex.



④ An approximate algorithm based on greedy strategy. (Gary Stormo & G. Hertz, 1989)  
the motif CONSENSUS

→ Consider a pair of  $n$ -mer & compute best  $(s_1, s_2)$  for maximum consensus score  $(n-l+1)^2 \cdot 2 \cdot l$

→ Iteratively compute ~~the~~  <sup>$i$ th</sup> ~~each~~  $n$ -mer ~~(from~~  $s_i$ , (by keeping  $(s_1, s_2, \dots, s_{i-1})$  fixed) for ~~max~~ the best consensus score.

for  $i$ th step:  $(n-l+1) \cdot i \cdot l$ .

$$\therefore \text{Time Complexity: } (n-l+1)^2 \cdot 2 \cdot l + \sum_{i=3}^t (n-l+1) \cdot i \cdot l$$

$$= (n-l+1)^2 \cdot 2 \cdot l + (n-l+1) \cdot l \cdot \left[ \frac{t(t+1)}{2} - 3 \right]$$

$$\approx O(ln^2 + lnt).$$

→ In CONSENSUS, ~~the~~ <sup>one</sup> sequence (n-mer) is selected randomly & also stores some other high scoring combinations of  $(s_1, s_2, \dots, s_{i-1})$  for the purpose of searching.  $(s_1, s_2, \dots, s_{i-1})$  ~~is~~  $(s_1, s_2, \dots, s_{i-1})$

This provides almost optimal performance in most cases.

## ② Median String Problem (Equivalent to Motif Finding)

\* Hamming distance between two  $l$ -mers :  
number positions they differ.

$$d_H(\text{ATTGTC}, \text{ACTCTC}) = 2$$

|   |                |   |                |   |   |
|---|----------------|---|----------------|---|---|
| A | T              | T | G              | T | C |
| A | <sup>x</sup> C | T | <sup>x</sup> C | T | C |

\* Total Hamming distance between a  $l$ -mer  $\bar{v}$  and a set of  $l$ -mers  $Z$  (t no. of  $l$ -mers)  
o.e.  $\{\bar{z}_1, \bar{z}_2, \dots, \bar{z}_t\}$

$$d_H(\bar{v}, Z) = \sum_{i=1}^t d_H(\bar{v}, \bar{z}_i)$$

\* Total Hamming Distance between a  $l$ -mer  $\bar{v}$  and a set of  $t$   $n$ -mers  $X$  o.e.  $\{x_1, x_2, \dots, x_t\}$

$$d_H(\bar{v}, X) = \min_{\substack{Z = \{\bar{z}_1, \bar{z}_2, \dots, \bar{z}_t\}, \\ 1 \leq s_i \leq n-l+1}} (d_H(\bar{v}, Z|(X, S)))$$

$Z|(X, S) =$  set of  $t$   $l$ -mers from  $X$  given starting positions at  $S \equiv (s_1, s_2, \dots, s_t)$ .

\* Median String for  $X$  is the  $l$ -mer  $\bar{v}^*$  which minimize  $d_H(\bar{v}^*, X)$ . o.e.

$$\begin{aligned} d_H(\bar{v}^*, X) &= \min_{\bar{v}} (d_H(\bar{v}, X)) \\ &= \min_{\bar{v}} \left( \min_{Z} (d_H(\bar{v}, Z|(X, S))) \right) \end{aligned}$$



$$\bar{v}^* = \underset{\bar{v}}{\operatorname{argmin}} \left( \min_{\bar{s}} \left( d_H(\bar{v}, \bar{z} | (x, \bar{s})) \right) \right)$$

\* Computation of Median String is equivalent to Motif finding:

i)  $0 \leq d_H(\bar{v}_1, \bar{v}_2) \leq l$

ii)  $0 \leq d_H(\bar{v}, \bar{z}) \leq l.t$

iii)  $d_H(\bar{v}, \bar{x}) = l.t - \underset{s}{\max} CS(x, s)$

$\downarrow$   
 Consensus score

$\therefore$  Minimization of LHS  $\Rightarrow$  maximization of  $CS(x, s)$  for all possible  $s$ .

\* Search Space of  $\bar{v}$  (l-mer):

alphabets:  $\{A, T, C, G\} \triangleq \{1, 2, 3, 4\}$  (let)

$4^l$

$$\begin{cases} 1 & 1 \dots 1, 1 \\ 1 & 1 \dots 1, 2 \\ 1 & 1 \dots 1, 3 \\ 1 & 1 \dots 1, 4 \\ 1 & 1 \dots 2, 1 \\ \vdots \\ 4 & 4 \dots 4, 4 \end{cases}$$

Time complexity for

Computing:  $d_H(\bar{v}, x) \Rightarrow (n-l+1) \cdot t$

For each  $\bar{v}$ , find  $s_i$  giving minimum  $d_H(\bar{v}, \bar{x}_i)$  with respect to  $(n-l+1)$   $\Rightarrow$   $(n-l+1) \cdot t$

$\therefore$  Exhaustive search for all  $\bar{v}$ ,  $4^l \cdot (n-l+1) \cdot t \Rightarrow O(4^l \cdot n \cdot t)$

which may be efficient than  $O(l \cdot t \cdot n^2)$



\* Use of exhaustive approach as before.

\* Use of branch & bound method by computing partial hamming distance with  $q$ ,  $k$ -mer ( $k$  at  $k^{\text{th}}$  level of the search tree)  $(v_1, v_2, \dots, v_k)$ ,  $k \leq l$ , if the distance is greater than ~~already~~ current best distance, bypass the subtree & move to next sibling.

Prob. Derive a tighter bound for the branch & bound strategy.

Proof: split an  $l$ -mer  $\bar{w}$  into two parts  $\bar{u}$  &  $\bar{v}$ .

use  $d_H(\bar{u}, x) + d_H(\bar{v}, x)$  to bound  $d_H(\bar{w}, x)$ .

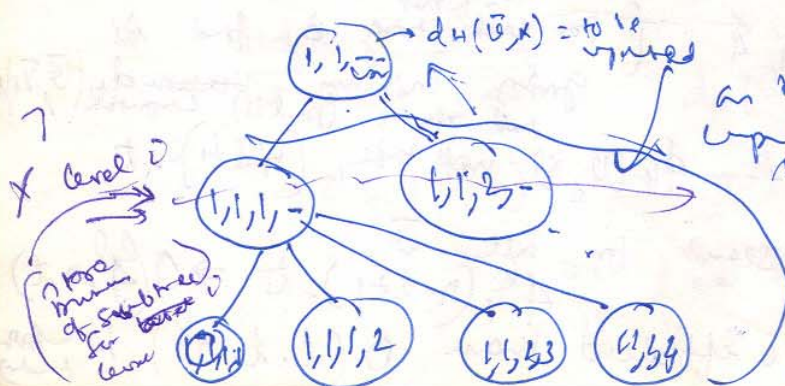
$$\Rightarrow d_H(\bar{w}, x) = \min_z \{d_H(\bar{w}, z | (x, S))\}$$

$$\phi_H(\bar{u}, x) =$$

$\phi_H(\bar{u}, x) = \dots$   
 $\therefore \phi_H(\bar{w}, x) \leq \phi_H(\bar{u}, x) + \underbrace{d_H(\bar{v}, x)}_{\text{minimum at subtree level}} \leq d_H(\bar{w}, x).$

$$\therefore d_H(\bar{v}, x) \leq d_H(\bar{w}, x) - d_H(\bar{v}, x).$$

Initialise,  $d_H(\bar{U}, x) = 0$ , but update as soon as the value is available  $\Gamma_{y_i}$  need to



[ You need to store  
~~compute~~ these  
values as well  
(Assigning my  
own capital)

1) Across the globe  
(one may store the  
num at level 1) ~~2~~ ~~10~~



## ⑦ Randomized Algorithms:

Let  $\bar{s} = (s_1, s_2, \dots, s_t)$  [starting positions of 'i' <sup>sequence strategy</sup>]  
 produce the profile matrix  $P(\bar{s}) = [p_{ij}]_{4 \times l}$ .  
 [for an l-mer consensus string]  $p_{ij} = \text{prob. of } i^{\text{th}} \text{ nucleotide at } j^{\text{th}} \text{ loc.}$

$$\text{Hence, } p(\bar{s} | \bar{v} (= v_1, v_2, \dots, v_l)) = \prod_{i=1}^l p(v_i, i)$$

Hence, most probable l-mer for  $v_i$  sequence is the one whose  $p(\bar{s} | P(\bar{s}))$  is maximum.

One may randomly select 'i' <sup>th</sup> seq. & iteratively update  $s_i$  <sup>leading</sup> based on this observation. However, the choice of initial seed is ~~difficult~~ arbitrary, it is unlikely to get optimal solution quickly.

### Gibbs sampling:

1. Randomly select  $\bar{s} = (s_1, s_2, \dots, s_t)$  &  $i^{\text{th}}$  (seq.)
2. Create profile matrix, leaving the  $i^{\text{th}}$  seq.  
 calculate  $p_j$  of l-mers starting at  $j^{\text{th}}$  location for the  $i^{\text{th}}$  seq.,  $1 \leq j \leq n-l+1$ .
3. Choose  $s_i$  in proportion to  $(p_1, p_2, \dots, p_{n-l+1})$
4. Iterate till the convergence achieved.

⇒ @ local optimum, works poorer in case of skewed distribution of nucleotides (some are more likely to occur).

e.g. If A has a freq. of 70% & T, G & C have freq. 10%, then Poly(A) may be the most freq. motif.



(From: "Computational Genome Analysis: An Intro", Deonier, Travare & Waterman (Springer, 2005)).

1. Profile matrix is also known as positional weight matrix or PWM in a probabilistic framework, where,

$$p_{ai} = \frac{f_{ai}}{\sum_{a=A,C,G,T} f_{ai}}, \quad a \in \{A, T, C, G\}$$

2. Let  $A = a_1 a_2 \dots a_w$  be the consensus string.

$$\therefore P(A | \text{Hypothesis}) = \prod_{i=1}^w p_{ai}$$

(or B)  $\downarrow$  of declares it as a motif.

3. Let the same sequence is chosen with a random collection of bases (say hypothesis R), with prob.  $q_{ai}$  (the random model in mind). Then the prob. of the seq. is given by:

$$P(A | R) = \prod_{i=1}^w q_{ai}$$

The "score" can be defined as:

$$S = \log_2 \frac{P(A | B)}{P(A | R)} = \sum_{i=1}^w \log_2 \left( \frac{p_{ai}}{q_{ai}} \right) = \sum_{i=1}^w s_i$$

Rep Multiple representation:  $S$  is represented in no. of bits (let)

Position-specific scoring matrix (PSSM) or "possum" =  $\left[ \log_2 \left( \frac{p_{ai}}{q_{ai}} \right) \right]_{4 \times w}$    
 $p_{ai} \Rightarrow$  Prob of a at  $i^{th}$  pos. of motifs across all the seqs.   
 $q_{ai} \Rightarrow$  Prob of a at  $i^{th}$  pos. of random seqs.   
 Note that:  $\frac{P(A | B)}{P(A | R)} = 2^S$    
 Problem in repn if  $p_{ai} = 0$  ??   
 m. of a bases in seq

Relative entropy of A:

$$H(A | q) = \sum_{a \in \{A, C, T, G\}} p_{ai} \log_2 \left( \frac{p_{ai}}{q_{ai}} \right)$$



# Determining threshold of significance:

