Tennessee State University

Finding Signals from the Candidates of Interaction Sites for Photosystem I Reduction

SHONDALYN A. SMITH

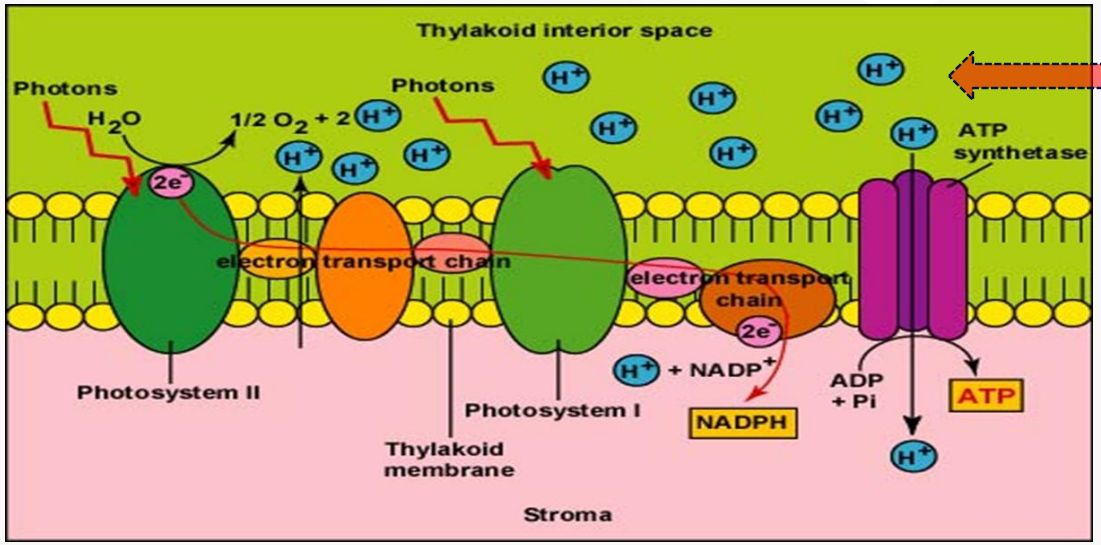
STEVYN SAWYER

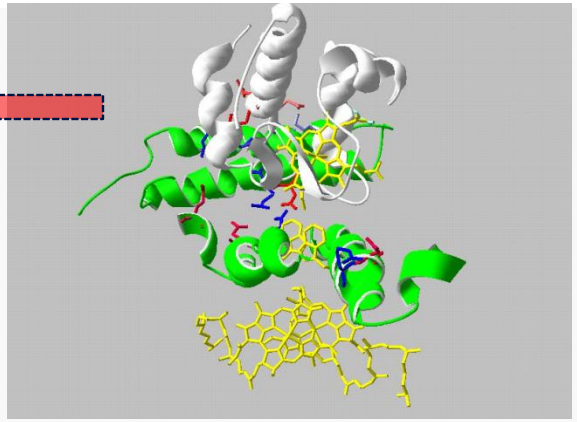
DR. WEI CHEN, ADVISOR

DEPARTMENT OF COMPUTER SCIENCE

TENNESSEE STATE UNIVERSITY

Research Background





Artificially adding c6 and psaF in Photosystem I speed up the process and produce more hydrogen

- Natural system produces hydrogen but in small amounts.
- Artificially adding c6 and psaF proteins produces more amounts quickly.
- Previous senior project group predicted the candidates for the interaction.

In order to make the prediction more preceise, we want to find more common signals from these candidates !!!

Problem Statement

Find common signal (motif) from the candidates of interaction sites in psaF and c6 protein families.

Challenge – Then same motif in different proteins can be different after have had mutation. Finding motif is computationally very expensive.

It is NP(No-deterministic Polynormial(-hard)!!!

Goal and Objectives

Goal:

Find further information (motif) from the identified candidates of interaction sites in c6 and psaF protein families.

Research Objectives:

1)Design new algorithm for finding motif effectively and efficiently.

2)Implement and evaluate the algorithm and analyze the results with existing algorithms.

Approach

- Investigate known algorithms.
- Design new algorithm.
- Compare the proposed algorithm with existing algorithms.

Functional Requirements

1. Protein pairs of c6 and psaF family from the same organism need to be provided.
2. The candidates of interaction sites of the given protein pairs need to be provided.
3. Identify and compare the existing software/tools/databases for motif/signal search of proteins for c6 and psaF.
4. Efficient algorithms and software that fits motifs/signals search from the candidates of interaction sites of c6 and psaF family are required.

Non-functional Requirements

1. Utilize standard database to find signals from the candidates of the interaction sites.
2. Compare the results from standard database and propose algorithm.
3. Algorithm implementation will utilize C# in Visual Studios.

System Design

- Dataset preparation to find signal from candidates of interaction sites.
- Utilize existing database/tools/software.
- Algorithm design.

System Design: Dataset preparation

- *Example 1: one pair of psaF and c6 proteins.*
- **psaF:**
  - DIAGLTPCSESKAYAKLEKKELKTLEKRLKQYEADSAPAVALKAT MERTKARFANYAKAGLLCGNDGLPHLIADPGLALKYGHAGEVFI PTFGFLYVAGYIGYVGRQYLIAVKGEAKPTDKEIIIDVPLATKLAW QGAGWPLAAVQELQRGTLLEKEENITVSPR
- **c6:**
  - ADLALGAQVFNGNCAACHMGGRNSVMPEKTLDKAALEQYLDGG FKVESIIYQVENGKGAMPAWADRLSEEEIQAVAEYVFKQATDAA WKY

Algorithm Design

Calculate the score of a motif

(All algorithms will use this method)

Example: Seven 36- nucleotide DNA sequences with a hid pattern TGCAACT mutated in two positions.

GGGCT**ATcCAgCT**GGGTCGTCACATTCCCCTTTCGA

TGAGGGTGCCCAATAA**ggGCAACT**CCAAAGCGGACA

ATGgAtCTGATGCCGTTTGACGACCTAAATCAACGG

GG**AaGCAACc**CCAGGAGCGCCTTTGCTGGTTCTACC

TTTTCTAAAAAGATTATAATGTCGGTCC**tTGgAACT**

GCTGTACACACTGGATCATGCTGC**ATGCcAtT**TTCA

CATGATCTTTTG**ATGgcACT**TGGATGAGGGAATGAT

GGGCT **A T c C A g C T** GGGTCGTCACATTCCCCTTTCGA

TGAGGGTGCCCAATAA **g g G C A A C T** CCAAAGCGGACA

A T G g A t C T GATGCCGTTTGACGACCTAAATCAACGG

GG **A a G C A A C c** CCAGGAGCGCCTTTGCTGGTTCTACC

TTTTCTAAAAAGATTATAATGTCGGTCC **t T G g A A C T**

GCTGTACACACTGGATCATGCTGC **A T G C c A t T** TTCA

CATGATCTTTTG **A T G g c A C T** TGGATGAGGGAATGAT

A	5	1	0	0	5	5	0	0
T	1	5	0	0	0	1	1	6
G	1	1	6	3	0	1	0	0
C	0	0	1	4	2	0	6	1
Consensus	**A**	**T**	**G**	**C**	**A**	**A**	**C**	**T**

position = (6,17,1,3,29,25,13)

Score = 5+5+6+4+5+5+6+6 = 42

Existing Algorithms for Finding Motifs

BruteForce

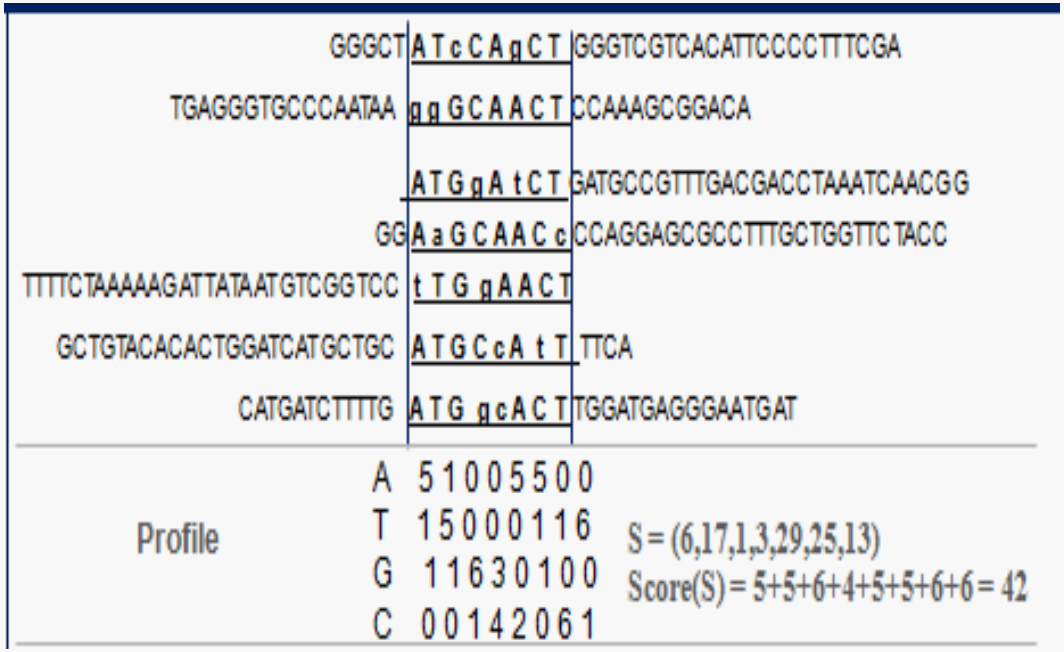
BruteForce(DNA, t, n, l)

```
bestScore := 0;
for  i1 := 1 to n-l+1
   for  i2 := 1 to n-l+1
      ……
     for  it := 1 to n-l+1
        S = (i1, i2, … , it)
        if (Score(S DNA) > bestScore)
           bestScore := Score(S, DNA
           bestMotifPosition = S
return bestScore & bestMotifPosition;
```

DNA: DNA sequences
t: number of DNA sequences
n: length of DNA sequences
l: length of the motif

Time Complexity : $O(n^t lt)$

Example: t=7, n=36, l=8,



Running Time: 7 hours

Existing Algorithms for Finding Motifs

Greedy Algorithm

Greedy-MotifFinding(DNA, t, n, *l*)
bestMotif := (1,1,…,1);
s := (1,1,…,1)
for s1 := 1 to *n*-*l*+1
  for s2 := 1 to *n*-*l*+1
    S := (s1, s2, 1, … , 1)
    if (Score(S, Seq) > bestScore)
      bestScore := Score(S, DNA);
      bestMotif Position:= S
for i := 3 to t
  for si := 1 to *n*-*l*+1
    S:= (s1, s2, … , si, 1, … , 1)
    if (Score(S, DNA) > bestScore)
      bestScore := Score(S, Seq);
      bestMotif Pos:= S;
return bestScore & bestMotifPosition

Local optimal solution for first 2 strings

Local optimal solution for 3rd - *t* th strings

Time Complexity : $O(n^2tl + nt^2l)$

DNA[0] = "GCTGTACACA**CTGGATCA**TGCTGCATGCCATTTTCA"
DNA[1] = "CATGATCTTTTGATGGCACTTGGATGA**GGGAATGA**T"
DNA[2] = "A TGGA TCTGATGCCGTTTGACGACCTAAATCAACGG"
DNA[3] = "GGAAGCAACCCCAGGAGCGCCTTTGCTGGTTCTACC"
DNA[4] = "T TT TCTAAAAAGATTATAATGTCGGTCCTTGGAACT"
…… ……
DNA[13] ="TGAGGGTGCCCAATAAGGGCAACTCCAAAGCGGACA"

DNA[0] = "GCTGTACACA**CTGGATCA**TGCTGCATGCCATTTTCA"
DNA[1] = "CATGATCTTTTGATGGCACTTGGATGA**GGGAATGA**T"
DNA[2] = "**ATGGATCT**GATGCCGTTTGACGACCTAAATCAACGG"
DNA[3] = "GGAAGCAACCCCAGGAGCGCCTTTGCTGGTTCTACC"
DNA[4] = "T TT TCTAAAAAGATTATAATGTCGGTCCTTGGAACT"
…… ……
DNA[13] ="TGAGGGTGCCCAATAAGGGCAACTCCAAAGCGGACA"

DNA[0] = "GCTGTACACACTGGATCATGCTGCATGCCATTTTCA"
DNA[1] = "CATGATCTTTTGATGGCACTTGGATGAGGGAATGAT"
DNA[2] = "ATGGATCTGATGCCGTTTGACGACCTAAATCAACGG"
DNA[3] = "**GGAAGCAA**CCCCAGGAGCGCCTTTGCTGGTTCTACC"
DNA[4] = "T TT TCTAAAAAGATTATAATGTCGGTCCTTGGAACT"
…… ……
DNA[13] ="TGAGGGTGCCCAATAAGGGCAACTCCAAAGCGGACA"

Running Time: instant

Proposed Algorithms for Finding Motifs

Improved Greedy Algorithm (using Divide-and-Conquer)

ImprovedMotifFinding(DNA[i..j], t, n, *l*)
if (j-i) < 4
  return Greedy(DNA[i..j], t, n, *l)*
else
  k =(i+j-1)/2

x = **ImprovedMotifFinding(**DNA[i..k], t, n, *l*)
y = **ImprovedMotifFinding(**DNA[k+1..j], t, n, *l*)
if x.score > y.score
  improve DNA[k+1..j] by the motifs in DNA[i..k]
else
  improve DNA[i..j] by the motifs in DNA[K_1..j]
  return bestScore and bestMotifPosition

Time Complexity :

$$T(n) = 2T(n/2) + nt^2l/2 \quad \text{if } t > 4$$

$$= n^2tl \ (\text{use greedy}) \quad \text{if } t \le 4$$

$$T(n) = O(n^3tl)$$

If score of the motifs in first part is larger

CTGTACACACTGATCATGCTGCATGCCATTTTCA
CATGATCTTTTGATGGCACTTGGATGAGGGAATGAT
ATGGATCTGATGCCGTTTGACGACCTAAATCAACGG
GGAAGCAACCCCAGGAGCGCCTTTGCTGGTTCTACC
TTTTCTAAAAAGATTATAATGTCGGTCCTTGGAACT
TTTTCTAAAAAGATTATAATGTCGGTCCTTGGAACT
GCTGTACACACTGGATCATGCTGCATGCCATTTTCA
CATGATCTTTTGATGGCACTTGGATGAGGGAATGAT
GGGCTATCCAGCTGGGTCGTCACATTCCCCTTTCGA
TGAGGGTGCCCAATAAGGGCAACTCCAAAGCGGACA
TGGATCTGATGCCGTTTGACGACCTAAATCAACGG
GGAAGCAACCCCAGGAGCGCCTTTGCTGGTTCTACC
GGGCTATCCAGCTGGGTCGTCACATTCCCCTTTCGA
TGAGGGTGCCCAATAAGGGCAACTCCAAAGCGGACA

If score of the motifs in second part is larger

Test results and algorithm comparison

Dataset 1: Number of DNA sequences = 7,
Length of DNA sequence = 32, length of motif = 8

Algorithm	Score of Motif	Position of Motif	Run Time
Brute Force	42	5, 15, 0, 2, 27, 19, 12	7 hours
Greedy	37	14, 4, 2, 13, 23, 0, 0	4.2816 ms
Improved Greedy	37	14, 4, 2, 13, 23, 0, 0	7.9553 ms

Test results and algorithm comparison

Dataset 2: Number of DNA sequences = 14,

Length of DNA sequence = 32, length of motif = 8

GCTGTACACACTGGATCATGCTGCATGCCATTTTCA
CATGATCTTTTGATGGCACTTGGATGAGGGAATGAT
ATGGATCTGATGCCGTTTGACGACCTAAATCAACGG
GGAAGCAACCCCAGGAGCGCCTTTGCTGGTTCTACC
TTTTCTAAAAAGATTATAATGTCGGTCCTTGGAACT
TTTTCTAAAAAGATTATAATGTCGGTCCTTGGAACT
GCTGTACACACTGGATCATGCTGCATGCCATTTTCA
CATGATCTTTTGATGGCACTTGGATGAGGGAATGAT
GGGCTATCCAGCTGGGTCGTCACATTCCCCTTTCGA
TGAGGGTGCCCAATAAGGGCAACTCCAAAGCGGACA
ATGGATCTGATGCCGTTTGACGACCTAAATCAACGG
GGAAGCAACCCCAGGAGCGCCTTTGCTGGTTCTACC
GGGCTATCCAGCTGGGTCGTCACATTCCCCTTTCGA
TGAGGGTGCCCAATAAGGGCAACTCCAAAGCGGACA

Improved-greedy

Algorithm	Score of Motif	Position of Motif	Run Time
Brute Force			Years
Greedy	68	10, 27, 0, 11, 8, 8, 10, 26, 0, 2, 0, 2, 1, 2	7.0717 ms
Improved Greedy	76	10, 26, 0, 11, 28, 28, 10, 26, 11, 2, 0, 11, 11,2	21.0426 ms

Future Work

- Continue to find better algorithms.
- Apply proposed algorithm to the candidates of the interaction sites in c6 and psaF protein families.

Acknowledgements

Dr. Wei Chen

Dr. Ali Sekmen