TIN SINH HỌC ĐẠI CƯƠNG
(Introduction to Bioinformatics)

PGS.TS. Trần Văn Lăng
Email: langtv@vast.vn

Chương 4:

PHÂN TÍCH TRÌNH TỰ DNA

- Chuyển đổi trình tự DNA
- Dự đoán gene, tìm motif

PHƯƠNG PHÁP SIXFRAME

- DNA động thực vật được cấu thành chủ yếu từ 4 base cơ bản là A, T, G, C
- Chúng có khả năng tạo nên 64 codon (mỗi codon gồm 3 base),
- Được gói gọn thành 20 amino acid.
- Các amino acid này lại góp phần hình thành nên các protein đặc trưng.

- Tuy nhiên, các sinh vật không giống nhau nên sự hình thành amino acid cũng không giống nhau.
- Có những loài với codon này thì tạo nên amino acid này nhưng đối với loài khác thì lại là một acid amin khác.

Chẳng hạn

- Bộ ba (codon) “CTT” ở động vật có xương sống hình thành nên Leucine thì ở nấm men là Threonine.
- Do đó, có nhiều bảng dịch mã khác nhau cho từng loài, từng trường hợp khác nhau

Dịch mã sang Protein

- Tiến trình chuyển đổi DNA sang protein cũng có sự khác biệt.
- Bởi không thể biết chính xác trình tự DNA đưa vào có base bắt đầu chính là base đầu tiên trong một codon hay không

Six-Frame Translation

- Dùng phương pháp Six-Frame với 3 dịch chuyển, ký hiệu +1, +2 và +3.
- Ngoài ra, DNA tồn tại dạng chuỗi xoắn kép theo từng cặp A-T, C-G.

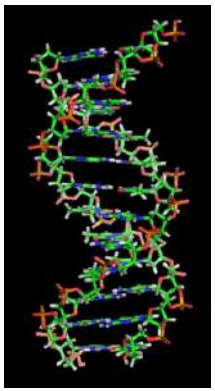
- Hơn nữa, đầu 3' của DNA không rõ là phía nào.
- Nên có 6 trường hợp cho cả 2 trình tự xoắn với nhau.

Ví dụ

- Với chuỗi xoắn kép của DNA

  5' GGTCTAT 3'

  3' CCAGATA 5'
- *Lưu ý*: Mỗi sợi đơn có một trình tự định hướng với một đầu 5 phosphate tự do, đầu kia là 3 hydroxyl tự do.



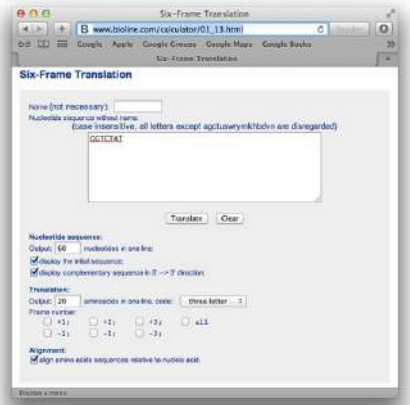
- Hướng của hai sợi đơn trong chuỗi xoắn kép ngược nhau, nên được gọi là hai sợi đối song
- Khi đó có các trường hợp xãy ra như sau theo 2 trình từ

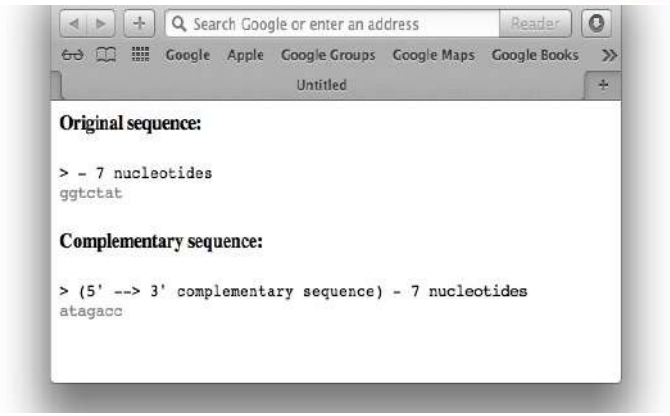
Translation in forward direction

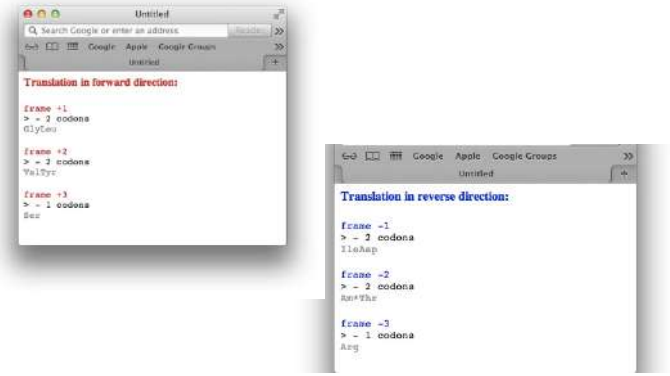
- Sợi thứ nhất: 5′ GGTCTAT 3′
  - –frame +1, có 2 codon: GGT CTA: GlyLeu
  - –frame +2 có 2 codon: GTC TAT: ValTyr
  - –frame +3 có 1 codon: TCT: Ser

- Sợi thứ hai: 5′ ATAGACC 3′
  - –frame -1, có 2 codon: ATA GAC: IleAsp
  - –frame -2, có 2 codon: TAG ACC: Am*Thr
  - –frame -3, có 1 codon: AGA: Arg

Sử dụng phần mềm

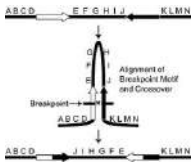






TÌM MOTIF

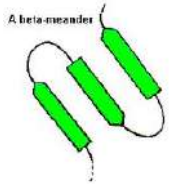
- **Motif** là một đoạn trình tự nucleotide hay amino acid phổ biến và có (hoặc cho là có) một chức năng sinh học nào đó
- Đối với protein, motif được phân thành 2 loại:
  - Motif trình tự
  - Motif cấu trúc



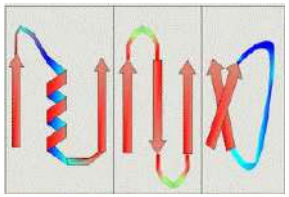
- Motif trình tự: sequence motif, motif, pattern, conserved pattern, consensus pattern, signature, fingerprint, block, feature.
- Là một vùng trình tự bảo tồn hay là một đoạn trình tự đặc trưng được tìm thấy ở 2 hay nhiều trình tự.

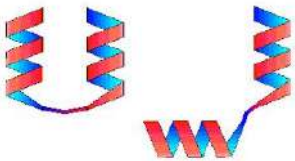
- Bản thân nó đại diện cho chức năng, cấu trúc hoặc thành viên trong họ.
- Các motif có thể được tìm thấy ở DNA, RNA, và Protein.

- Motif cấu trúc (structural motif) là một nhóm cấu trúc gần kề nhau kết hợp lại tạo thành một cấu trúc 3D chuyên biệt
- Lưu ý, với protein, motif trình tự (sequence motif) và motif cấu trúc là khác nhau hoàn toàn.



- Ví dụ: xoắn – vòng - xoắn
- Còn được gọi là siêu cấu trúc bậc 2 (supersecondary structure)





- Ứng dụng, chẳng hạn
  - Ghép mẫu motif vào trong trình tự.
  - Sau đó tìm nơi đã cấy ghép

- Có một mẫu trình tự nucleotide ngẫu nhiên

```
atgaccgggatactgataccgtatttggcctaggcgtacacattagataaacgtatgaagtacgttagactcggcgccgccg
acccctatttttgagcagatttagtgacctggaaaaaaaatttgagtacaaaacttttccgaatactgggcataaggtaca
tgagtatccctgggatgacttttgggaacactatagtgctctcccgatttttgaatatgtaggatcattcgccagggtccga
gctgagaattggatgaccttgtaagtgttttccacgcaatcgcgaaccaacgcggacccaaaggcaagaccgataaaggaga
tccctttgcggtaatgtgccgggaggctggttacgtagggaagccctaacggacttaatggcccacttagtccacttatag
gtcaatcatgttcttgtgaatggatttttaactgagggcatagaccgcttggcgcacccaaattcagtgtgggcgagcgcaa
cggttttggcccttgttagaggcccccgtactgatggaaactttcaattatgagagagctaatctatcgcgtgcgtgttcat
aacttgagttggtttcgaaaatgctctggggcacatacaagaggagtcttccttatcagttaatgctgtatgacactatgta
ttggcccattggctaaaagcccaacttgacaaatggaagatagaatccttgcatttcaacgtatgccgaaccgaaagggaag
ctggtgagcaacgacagattcttacgtgcattagctcgcttccggggatctaatagcacgaagcttctgggtactgatagca
```


- Ghép motif trình tự **AAAAAAAAGGGGGGG vào**

```
atgaccgggatactgatAAAAAAAAGGGGGGGgcgtacacattagataaacgtatgaagtacgttagactcggcgccgccg
acccctatttttgagcagatttagtgacctggaaaaaaaatttgagtacaaaacttttccgaataAAAAAAAAGGGGGGGa
tgagtatccctgggatgacttAAAAAAAAGGGGGGGtgctctcccgatttttgaatatgtaggatcattcgccagggtccga
gctgagaattggatgAAAAAAAAGGGGGGGtccacgcaatcgcgaaccaacgcggacccaaaggcaagaccgataaaggaga
tccctttgcggtaatgtgccgggaggctggttacgtagggaagccctaacggacttaatAAAAAAAAGGGGGGGcttatag
gtcaatcatgttcttgtgaatggatttAAAAAAAAGGGGGGGgaccgcttggcgcacccaaattcagtgtgggcgagcgcaa
cggttttggcccttgttagaggcccccgtAAAAAAAAGGGGGGGcaattatgagagagctaatctatcgcgtgcgtgttcat
aacttgagttAAAAAAAAGGGGGGGctggggcacatacaagaggagtcttccttatcagttaatgctgtatgacactatgta
ttggcccattggctaaaagcccaacttgacaaatggaagatagaatccttgcatAAAAAAAAGGGGGGGaccgaaagggaag
ctggtgagcaacgacagattcttacgtgcattagctcgcttccggggatctaatagcacgaagcttAAAAAAAAGGGGGGGa
```


- Vấn đề đặt ra là motif được ghép nằm ở đâu

```
atgaccgggatactgataaaaaaaaggggggggcgtacacattagataaacgtatgaagtacgttagactcggcgccgccg
acccctatttttgagcagatttagtgacctggaaaaaaaatttgagtacaaaacttttccgaataaaaaaaaggggggga
tgagtatccctgggatgacttaaaaaaaaggggggtgctctcccgatttttgaatatgtaggatcattcgccagggtccga
gctgagaattggatgaaaaaaaaggggggtccacgcaatcgcgaaccaacgcggacccaaaggcaagaccgataaaggaga
tccctttgcggtaatgtgccgggaggctggttacgtagggaagccctaacggacttaataaaaaaaaggggggcttatag
gtcaatcatgttcttgtgaatggatttaaaaaaaaggggggggaccgcttggcgcacccaaattcagtgtgggcgagcgcaa
cggttttggcccttgttagaggcccccgtaaaaaaaaggggggcaattatgagagagctaatctatcgcgtgcgtgttcat
aacttgagttaaaaaaaaggggggctggggcacatacaagaggagtcttccttatcagttaatgctgtatgacactatgta
ttggcccattggctaaaagcccaacttgacaaatggaagatagaatccttgcataaaaaaaaggggggaccgaaagggaag
ctggtgagcaacgacagattcttacgtgcattagctcgcttccggggatctaatagcacgaagcttaaaaaaaaggggggga
```


- Và rồi, giả sử có 4 đột biến trong motif đã cấy ghép **AAAAAAAAGGGGGGG.**

```
atgaccgggatactgatAgAAgAAAGGttGGGggcgtacacattagataaacgtatgaagtacgttagactcggcgccgccg
acccctatttttgagcagatttagtgacctggaaaaaaaatttgagtacaaaacttttccgaataCAAtAAAAcGGcGGGa
tgagtatccctgggatgacttAAAAtAAtGGaGtGGtgctctcccgatttttgaatatgtaggatcattcgccagggtccga
gctgagaattggatgcAAAAAAAGGGattGtccacgcaatcgcgaaccaacgcggacccaaaggcaagaccgataaaggaga
tccctttgcggtaatgtgccgggaggctggttacgtagggaagccctaacggacttaatAtAAtAAAGGaaGGGcttatag
gtcaatcatgttcttgtgaatggatttAACcAAtAAGGGctGGgaccgcttggcgcacccaaattcagtgtgggcgagcgcaa
cggttttggcccttgttagaggcccccgtAtAAACAAGGaGGGCcaattatgagagagctaatctatcgcgtgcgtgttcat
aacttgagttAAAAAtAGGGaGCCctggggcacatacaagaggagtcttccttatcagttaatgctgtatgacactatgta
ttggcccattggctaaaagcccaacttgacaaatggaagatagaatccttgcatActAAAAAGGaGcGGaccgaaagggaag
ctggtgagcaacgacagattcttacgtgcattagctcgcttccggggatctaatagcacgaagcttActAAAAAGGaGcGGa
```

- Motif ở đâu trong trình tự này

```
atgaccgggatactgatagaagaaaggttgggggcgtacacattagataaacgtatgaagtacgttagactcggcgccgccg
acccctattttttgagcagatttagtgacctggaaaaaaaatttgagtacaaaacttttccgaatacaataaaacggcggga
tgagtatccctgggatgacttaaaataatggagtggtgctctcccgatttttgaatatgtaggatcattcgccagggtccga
gctgagaattggatgcaaaaaaagggattgtccacgcaatcgcgaaccaacgcggacccaaaggcaagaccgataaaggaga
tccctttttgcggtaatgtgccgggaggctggttacgtagggaagccctaacggacttaatataataaaggaagggcttatag
gtcaatcatgttcttgtgaatggatttaacaataagggctgggaccgcttggcgcacccaaattcagtgtgggcgagcgcaa
cggttttggcccttgttagaggcccccgtataaacaaggagggccaattatgagagagctaatctatcgcgtgcgtgttcat
aacttgagttaaaaaatagggagccctggggcacatacaagaggagtcttccttatcagttaatgctgtatgacactatgta
ttggcccattggctaaaagcccaacttgacaaatggaagatagaatccttgcatactaaaaaggagcggaccgaaagggaag
ctggtgagcaacgacagattcttacgtgcattagctcgcttccggggatctaatagcacgaagcttactaaaaaggagcgga
```


- Tại sao tìm motif (15,4) này lại khó

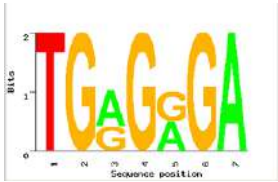
```
atgaccgggatactgatAgAAgAAAGGttGGGggcgtacacattagataaacgtatgaagtacgttagactcggcgccgccg
acccctattttttgagcagatttagtgacctggaaaaaaaatttgagtacaaaacttttccgaatacAAtAAAACGGCGGGa
tgagtatccctgggatgacttAAAAtAAtGGaGtGGtgctctcccgatttttgaatatgtaggatcattcgccagggtccga
gctgagaattggatgCAAAAAAAGGGattGtccacgcaatcgcgaaccaacgcggacccaaaggcaagaccgataaaggaga
tccctttttgcggtaatgtgccgggaggctggttacgtagggaagccctaacggacttaatAtAAtAAAGGaaGGGcttatag
gtcaatcatgttcttgtgaatggatttAACAAtAAGGGctGGgaccgcttggcgcacccaaattcagtgtgggcgagcgcaa
cggttttggcccttgttagaggcccccgtAtAAACAAGGaGGGCcaattatgagagagctaatctatcgcgtgcgtgttcat
aacttgagttAAAAAtAGGGaGCCctggggcacatacaagaggagtcttccttatcagttaatgctgtatgacactatgta
ttggcccattggctaaaagcccaacttgacaaatggaagatagaatccttgcatActAAAAAGGaGCGGaccgaaagggaag
ctggtgagcaacgacagattcttacgtgcattagctcgcttccggggatctaatagcacgaagcttActAAAAAGGaGCGGa
```

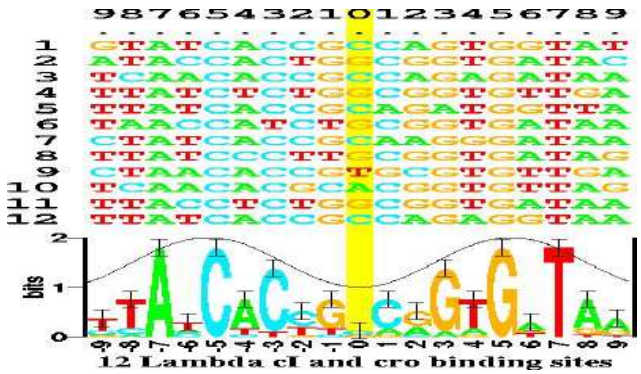
AgAAgAAAGGttGGG
CAAAAAAACGGCGGG

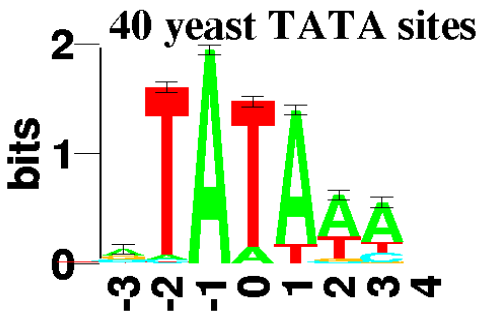
Biểu tượng motif (motif logo)

- Motif có thể đột biến trên những base không quan trọng.
- Năm motif trong 5 gen khác nhau có đột biến ở vị trí thứ 3 và 5 như hình.
- Motif logo minh họa vùng bảo tồn và vùng thay đổi của một motif

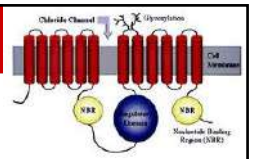
TGGGGGA
TGAGAGA
TGGGGGA
TGAGAGA
TGAGGGA







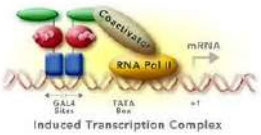
Việc định danh motif



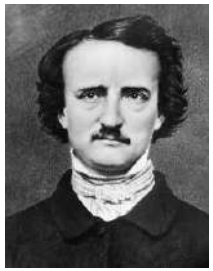
- Ta biết gen được bật tắt bởi protein điều chỉnh (regulator protein)
- Các protein điều chỉnh này liên kết với trình tự DNA ngắn gọi là motif
- Vì vậy việc tìm các motif tương tự nhau trong nhiều vùng điều chỉnh gen cho ra một quan hệ điều chỉnh trong số những gen này.

Sự phức tạp trong việc định danh motif

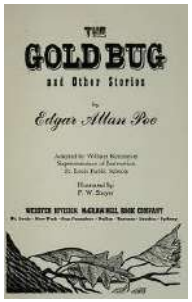
- Không biết được trình tự motif
- Không biết nó nằm ở đâu so với gen bắt đầu
- Các motif có thể khác nhau đôi chút từ gen này đến gen kế tiếp, vì vậy làm sao có thể phân biệt được nó từ các motif ngẫu nhiên



- Bài toán tìm motif (Motif Finding Problem) tương tự như bài toán được đặt ra bởi Edgar Allan Poe (1809 – 1849) trong quyển truyện *Gold Bug*



- Edgar Allan Poe là một tác giả người Mỹ, được xem như người đã tạo ra thể loại văn học trinh thám.
  - Bản thảo tìm thấy trong chai (Ms. Found in a Bottle)
  - Bọ rầy vàng óng ánh (The Gold Bug)



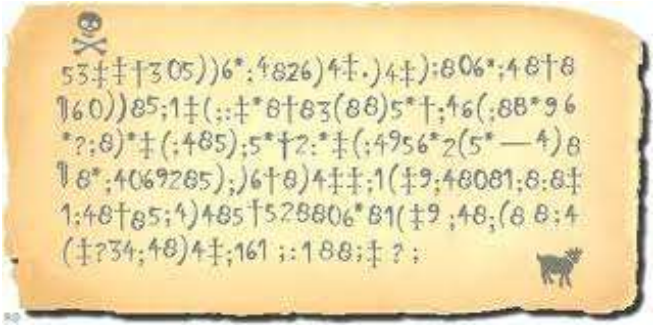
Truyện Gold Bug

- "Gold Bug" là câu chuyện về một người đàn ông tên là William Legrand, người dường như bị điên sau khi bị cắn bởi một con bọ được làm bằng vàng nguyên chất.
- Legrand nói người bạn thân nhất của mình, đến thăm tại nhà của ông trên đảo Sullivan ở South Carolina.

- Hai người bắt tay vào tìm kiếm kho báu bị mất cùng với một người hầu tên là Jupiter.
- Người bạn nghi ngờ tính đúng đắn trong câu chuyện của Legrand.

- Tuy nhiên, sau khi làm theo vài manh mối, họ tìm thấy một kho báu bị chôn vùi bởi cướp biển khét tiếng tên là Captain Kidd
- Kho báo ước tính trị giá khoảng 14.000.000 USD.

- Trong số các manh mối đó, có thông điệp bí mật như sau:



The Gold Bug Problem

- Thông điệp bí mật

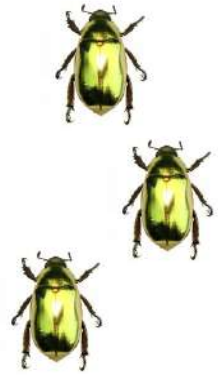
```
53++!305))6*;4826)4+.)4+);806*;48!8`60))85;]8*:+*8!8
  3(88)5*!;
46(;88*96*?;8)*+(;485);5*!2:*+(;4956*2(5*-4)8`8*;
  4069285);)6
!8)4++;1(+9;48081;8:8+1;48!85;4)485!528806*81(+9;48;
  (88;4(+?3
4;48)4+;161;:188;+?;
```

- Hãy giải mã thông điệp được mã hóa này

- Các gợi ý như sau:
  - Thông điệp được mã hóa bằng tiếng Anh
  - Mỗi ký hiệu tương ứng với một chữ cái trong bảng chữ cái tiếng Anh
  - Không có dấu chấm câu được mã hóa

Cách giải quyết

- Đếm tần số xuất hiện của mỗi ký hiệu trong thông điệp được mã hóa
- Tìm tần số của mỗi ký tự trong bảng chữ cái của văn bản tiếng Anh



- So sánh các tần số của các bước trước đó, cố gắng tìm một mối tương quan và ánh xạ các ký hiệu với một ký tự trong bảng chữ cái

- Tần số theo thông điệp của Gold Bug

Symbol	8	;	4	)	+	*	5	6	(	!	1	0	2	9	3	:	?	`	-	]	.
Frequency	34	25	19	16	15	14	12	11	9	8	7	6	5	5	4	4	3	2	1	1	1

- Tần số theo bảng chữ cái tiếng Anh

e t a o i n s r h l d c u m f p g w y b v k x j q z

Tần số cao → tần số thấp

- Bằng cách ánh xạ đơn giản các ký hiệu có tần số cao nhất đến các ký tự có tần số cao nhất tương ứng trong bảng chữ cái.
- Giải mã 4 mãnh trong thông điệp bí mật

```
sfiilfcsoorntaeuroaikoaiotecrntaeleyr
cooestvenpinelefheeosnlt
```



```
arhteenmrnwteonihtaesotsnlupnihtamsrn
uhsnbaoeyentacrmuesotorl
eoaiitdhimtaecedtepeidtaelestaoaeslsu
eecrnedhimtaetheetahiwfa
taeoaitdrdtpdeetiwt
```

- Kết quả không có ý nghĩa gì cả

Cách tiếp cận tốt hơn

- Đánh giá tần số của *l*-tuples như tổ hợp của 2 ký hiệu, 3 ký hiệu, v.v… Chẳng hạn,
  - "The" là 3-tupe có tần số cao nhất trong tiếng Anh; ";48" là 3-tuple có tần số cao nhất trong thông điệp mã hóa

- Suy ra tương tự cho các ký hiệu chưa biết trong văn bản mã hóa dựa trên tần số của các *l*-tuple.

- Ánh xạ "the" đến ";48" và thay thế tất cả các ký hiệu xuất hiện:

```
53++!305))6*the26)h+.)h+)te06*the!e`60))e5t]
e*:+*e!e3(ee)5*!t
h6(tee*96*?te)*+(the5)t5*!2:*+(th956*2(5*h)e
`e*th0692e5)t)6!e
)h++t1(+9the0e1te:e+1the!e5th)he5!52ee06*e1(
+9thet(eeth(+?3ht
he)h+t161t:1eet+?t
```


- Suy đoán

```
53++!305))6*the26)h+.)h+)te06*the!e`60))e5t]
e*:+*e!e3(ee)5*!t
h6(tee*96*?te)*+(the5)t5*!2:*+(th956*2(5*h)e
`e*th0692e5)t)6!e
)h++t1(+9the0e1te:e+1the!e5th)he5!52ee06*e1(
+9thet(eeth(+?3ht
he)h+t161t:1eet+?t
```

- "thet(ee" most likely means "the tree"
  - Suy ra Infer "(" = "r"

- Khi đó, “th(+?3h” trở thành “thr+?3h”
  - Sau đó có thể đề xuất “+”, “?” được mã hóa ra sao.

- Sau khi tìm ra tất cả các ánh xạ, thông điệp có thể giải mã như sau:

```
AGOODGLASSINTHEBISHOPSHOSTELINTHEDEVILSSEATWEN
YONEDEGREESANDTHIRTEENMINUTESNORTHEASTANDBYNOR
THMAINBRANCHSEVENTHLIMBEASTSIDESHOOTFROMTHELEF
TEYEOFTHEDEATHSHEADABEELINEFROMTHETREETHROUGHT
HESHOTFIFTYFEETOUT
```


- Sử dụng dấu chấm câu, thông điệp có thể là:

```
A GOOD GLASS IN THE BISHOP'S HOSTEL IN THE
DEVIL'S SEA,
TWENY ONE DEGREES AND THIRTEEN MINUTES
NORTHEAST AND BY NORTH,
MAIN BRANCH SEVENTH LIMB, EAST SIDE, SHOOT
FROM THE LEFT EYE OF
THE DEATH'S HEAD A BEE LINE FROM THE TREE
THROUGH THE SHOT,
FIFTY FEET OUT.
```


Giải bài toán Gold Bug

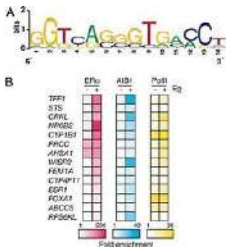
- Những điều kiện tiên quyết để giải bài toán:
  - Cần phải biết tần số tương đối của các chữ cái, và sự kết hợp của hai và ba chữ cái trong tiếng Anh
  - Kiến thức về tất cả các từ trong từ điển tiếng Anh là mong muốn cao để có những kết luận chính xác

Sự tương tự giữa 2 bài toán

- Những nucleotide trong một motif mã hóa là ngôn ngữ của di truyền, tương tự như ký hiệu mã hóa trong "The Gold Bug" của một thông điệp (message) bằng tiếng Anh

- Để giải mã, cần phân tích tần số của các mẫu thông điệp DNA/Gold Bug
- Kiến thức của các motif điều chỉnh được thiết lập làm cơ sở cho việc tìm motif; cũng như kiến thức về các từ trong từ điển Tiếng Anh làm cơ sở cho việc giải bài táon Gold Bug

- **Bài toán Motif Finding**:
  - Phân tích tần suất xuất hiện các mẫu (pattern) trong những trình tự nucleotide
- **Bài toán Gold Bug Problem**
  - Phân tích tần suất xuất hiện các mẫu trong văn bản được viết bằng Tiếng Anh



- **Motif Finding**:
  - Kiến thức về các motif được thiết lập làm giảm độ phức tạp của bài toán
- **Gold Bug Problem**:
  - Kiến thức về các từ trong từ điển Tiếng Anh là hết sức mong đợi

- **Bài toán Motif Finding** khó hơn bài toán **Gold Bug**:
  - Không có từ điển đầy đủ về motif
  - Ngôn ngữ di truyền học không có văn phạm chuẩn
  - Chỉ một phần nhỏ trình tự nucleotide mã hóa cho motif, trong khi đó kích thước dữ liệu lại rất lớn

Minh họa bài toán Motif Finding

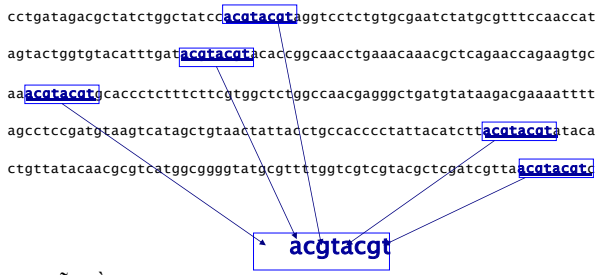
- Cho một mẫu ngẫu nhiên các trình tự DNA

```
cctgatagacgctatctggctatccacgtacgtaggtcctctgtgcgaatctatgcgtttccaaccat
agtactggtgtacatttgatacgtacgtacaccggcaacctgaaacaaacgctcagaaccagaagtgc
aaacgtacgtgcaccctctttcttcgtggctctggccaacgagggctgatgtataagacgaaaatttt
agcctccgatgtaagtcatagctgtaactattacctgccacccctattacatcttacgtacgtataca
ctgttatacaacgcgtcatggcggggtatgcgttttggtcgtcgtacgctcgatcgttaacgtacgtc
```

- Tìm mẫu được ghép vào mỗi trình tự riêng, gọi là motif

- Thông tin thêm:
  - Mỗi trình tự che dấu có chiều dài 8
  - Các mẫu không hoàn toàn giống nhau bởi điểm đột biến là ngẫu nhiên xảy ra trong các trình tự

- Các mẫu cho thấy không có đột biến



Chuỗi đồng thuận hay liên ứng (Consensus Sequence)

- Mẫu với 2 đột biến:

```
cctgatagacgctatctggctatccaGgtacTtaggtcctctgtgcgaatctatgcgtttccaaccat
agtactggtgtacatttgatCcAtacgtacaccggcaacctgaaacaaacgctcagaaccagaagtgc
aaacgtTAgtgcaccctctttcttcgtggctctggccaacgagggctgatgtataagacgaaaatttt
agcctccgatgtaagtcatagctgtaactattacctgccacccctattacatcttacgtCcAtataca
ctgttatacaacgcgtcatggcggggtatgcgttttggtcgtcgtacgctcgatcgttaCcgtacgGc
```

- Liệu có thể tìm được motif với 2 đột biến

aGgtacTt
CcAtacgt
acgtTAgt
acgtCcAt
CcgtacgG

Mẫu với 2 đột biến

acgtacgt

Phân loại bài toán tìm motif

- Có 2 dạng bài toán tìm motif:
  - **Không đột biến**: Cho trước t trình tự, hãy xác định các đoạn trình tự có chiều dài *l* (*l*-mer) trên mỗi trình tự sao cho đoạn này bắt cặp giống nhau.

- **Dạng đột biến**: Cho trước *t* trình tự, hãy xác định các đoạn trình tự có chiều dài *l* sao cho các đoạn này gần giống nhau trên các trình tự cho phép đột biến (sai lệch) *d* vị trí

aGgtacTt
CcAtacgt
acgtTAgt
acgtCcAt
CcgtacgG
acgtacgt

Định nghĩa Motif

- Để định nghĩa một motif, cần biết vị trí bắt đầu của motif trong trình tự.
- Vị trí này có thể biểu diễn bởi $\boldsymbol{s}$ = $(s_1, s_2, s_3, \ldots, s_t)$

Profiles và Consensus

Alignment

```
a G g t a c T t
C c A t a c g t
a c g t T A g t
a c g t C c A t
C c g t a c g G
```

Profile

A	3	0	1	0	3	1	1	0
C	2	4	0	0	1	4	0	0
G	0	1	4	0	0	0	3	1
T	0	0	0	5	1	0	1	4

Consensus A C G T A C G T

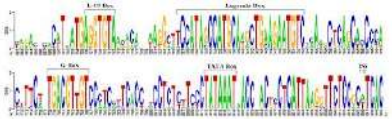
- Sắp hàng các mẫu theo vị trí bắt đầu của nó

  $\boldsymbol{s} = (s_1, s_2, \ldots, s_t)$

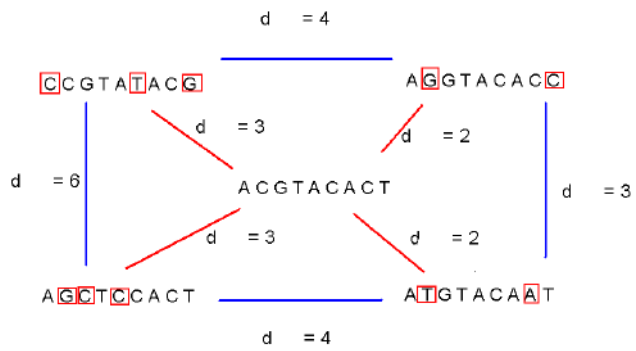
- Xây dựng ma trận profile với tần suất xuất hiện của mỗi nucleotide theo cột
- Consensus nucleotide là nucleotide có điểm cao nhất trong cột

Consensus

- Consensus (trình tự đồng thuận) ở đây được hiểu như là một motif tổ tiên mà từ đó các motif đột biến xuất hiện



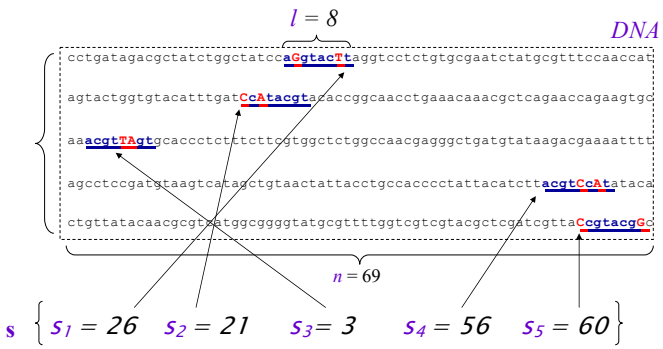
Khoảng cách giữa các trình tự



Đánh giá motif

- Trước hết, ta có các tham số
  - ***t*** - số mẫu trình tự DNA
  - ***n*** - chiều dài mỗi trình tự DNA
  - ***DNA*** –mẫu DNA (mảng *t* x *n*)
  - ***l*** - chiều dài của motif (*l*-mer)
  - s_i - vị trí bắt đầu của motif trong trình tự *i*
  - **s**=(s_1, s_2,… s_t) - mảng chứa các vị trí bắt đầu của motif

Ví dụ về các tham số



Tính điểm để đánh giá

- Cho **s** = (s_1, … s_t) và ***DNA***:

$$\text{score}(s, DNA) = \sum_{i=1}^{l} \max_{k \in \{A,T,C,G\}} \text{count}(k,i)$$

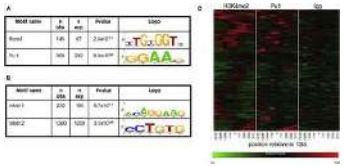
- Với count(k,i) là số nucleotide thứ k ở vị trí thứ i của *l*-motif

	a	G	g	t	a	c	T	t
	C	c	A	t	a	c	g	t
	a	c	g	t	T	A	g	t
	a	c	g	t	C	c	A	t
	C	c	g	t	a	c	g	G
A	3	0	1	0	3	1	1	0
C	2	4	0	0	1	4	0	0
G	0	1	4	0	0	0	3	1
T	0	0	0	5	1	0	1	4
Consensus	a	c	g	t	a	c	g	t

Score 3+4+4+5+3+4+3+4=30

- Nếu các vị trí bắt đầu **s**=(s_1, s_2,… s_t) cho trước, việc tìm consensus dễ dàng ngay cả khi có đột biến trong các trình tự.
  - Bởi khi đó ta có thể xây dựng ma trận profile, từ đó tìm motif (consensus)

- Nhưng, khi **s** không cho trước, làm thế nào để tìm ma trận profile tốt nhất.
  - Bài toán đặt ra:



Bài toán

- Mục tiêu: Cho mẫu DNA, tìm tập *l*-mers từ các trình tự sao cho điểm consensus là cực đại
- Nhập: A ***t*** x ***n*** mảng các mẫu ***DNA***, và chiều lài *l* của pattern muốn tìm
- Xuất: Mảng t vị trí **s** = (s_1, s_2, … s_t) mà Score(**s**,***DNA***) đạt cực đại

Thuật toán Brute Force

- Tính score của một tổ hợp với vị trí bắt đầu **s**
- Điểm tốt nhất được xác định bởi profile tốt nhất.
- Tìm *Score*(**s**,***DNA***) lớn nhất bằng cách thay đổi vị trí bắt đầu s_i, với i từ 1 đến n-l+1

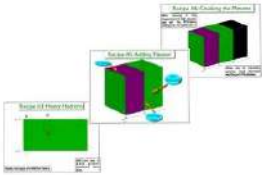

```
BruteForceMotifSearch(DNA, t, n, l)
bestScoe ← 0
for s=(s1,s2 , . . ., st) from (1,1 . . . 1) to
(n-l+1, . . ., n-l+1)
     if (Score(s,DNA) > bestScore)
          bestScore ← score(s, DNA)
          bestMotif ← (s1,s2 , . . . , st)
return bestMotif
```

Nhận xét

- Thay đổi $(n - l + 1)$ vị trí trong t trình tự, cần $(n - l + 1)^t$ tập hợp các vị trí bắt đầu
- Đối với mỗi tập hợp vị trí bắt đầu, score được tính dựa trên l phép toán, vì vậy độ phức tạp tính toán là $l \times (n - l + 1)^t = O(ln^t)$
- Với $t = 8$, $n = 1000$, $l = 10$ phải thực hiện xấp xỉ 10^{32} tính toán.

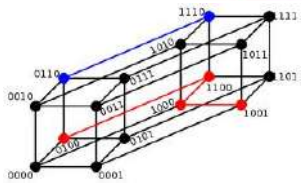
Bài toán Median String

- Với lý do trên, nên vấn đề đặt ra là tìm một thuật toán nhanh hơn để giải quyết.
- Bài toán Motif Finding được đưa về bài toán Median String (chuỗi trung bình)



- Bài toán Median String:
  - Cho mẫu t trình tự DNA tìm pattern xuất hiện trong tất cả t trình tự với số đột biến ít nhất
  - Pattern này chính là motif

Khoảng cách Hamming

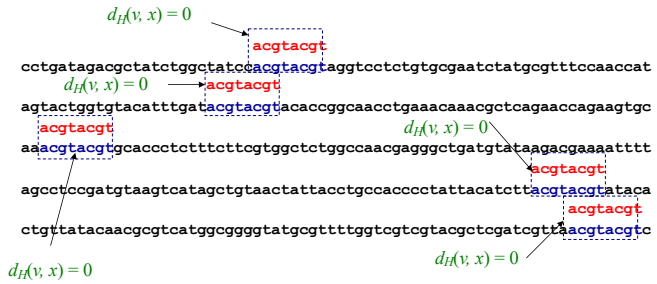


- Khoảng cách Hamming:
  - $d_H(\boldsymbol{v},\boldsymbol{w})$ là số cặp nucleotide mismatch (do not match) khi sắp hàng $\boldsymbol{v}$ và $\boldsymbol{w}$. Chẳng hạn

$$d_H(\texttt{AAAAAA},\texttt{ACAAAC}) = 2$$

Ví dụ

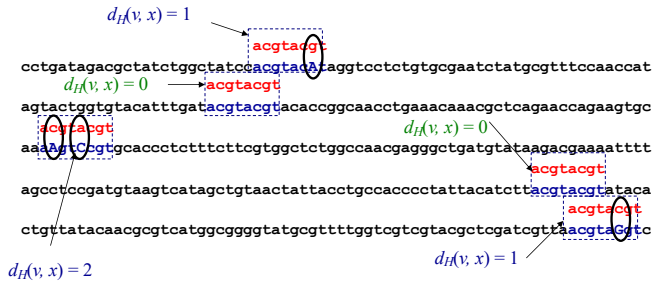
- Cho ***v*** = "acgtacgt" và **mẫu DNA**



*TotalDistance(**v**,**DNA**)* = 0

Ví dụ

- Cho ***v*** = "acgtacgt" và **mẫu DNA**



*TotalDistance(**v**,**DNA**)* = 1 + 0 + 2 + 0 + 1 = 4

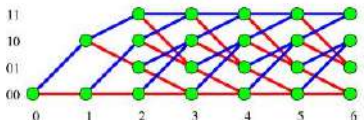
- Với trình tự DNA thứ ***i***, tính tất cả $d_H(\boldsymbol{v}, \boldsymbol{x})$, ở đó ***x*** là *l*mer với vị trí bắt đầu s_i

  $(1 \le s_i \le \boldsymbol{n} - l + 1)$
- Tìm cực tiểu $d_H(\boldsymbol{v}, \boldsymbol{x})$ của tất cả các *l*mers trong trình tự ***i***
- *TotalDistance(**v**,**DNA**)* tổng của các khoảng cách Hamming tối thiểu cho trình tự DNA thứ i
- *TotalDistance(**v**,**DNA**)* = $\min_{\boldsymbol{s}} d_H(\boldsymbol{v}, \boldsymbol{s})$, ở đó ***s*** là tập hợp các vị trí bắt đầu $s_1, s_2, \ldots s_t$

Thuật toán

- Mục tiêu: cho mẫu các trình tự DNA, tìm chuỗi trung bình
- Nhập: Ma trận DNA ***t*** x ***n***, chiều dài *l* của mẫu cần tìm.
- Xuất: chuỗi ***v*** gồm *l* nucleotides mà *TotalDistance(**v**,**DNA**)* đạt cực tiểu đối với tất cả các chuỗi có cùng chiều dài.

```
MedianStringSearch (DNA, t, n, l)
bestWord ← AAA…A
bestDistance ← ∞
for each l-mer s from AAA…A to TTT…T
     if TotalDistance(s,DNA) < bestDistance
          bestDistance←TotalDistance(s,DNA)
          bestWord ← s
 return bestWord
```

- Motif Finding Problem == Median String Problem
  - *Motif Finding* là bài toán cực đại, trong khi *Median String* là bài toán cực tiểu

- Tuy nhiên, đây là 2 bài toán tương đương
  - *TotalDistance* đạt cực tiểu tương đương *Score* đạt cực đại

Sự giống nhau

```
Alignment     a G g t a c T t
              C c A t a c g t
              a c g t T A g t
              a c g t C C A t
              C c g t a c g G
              ----------------

Profile     A 3 0 1 0 3 1 1 0
            C 2 4 0 0 1 4 0 0
            G 0 1 4 0 0 0 3 1
            T 0 0 0 5 1 0 1 4
              ----------------

Consensus     a c g t a c g t

Score         3+4+4+5+3+4+3+4

TotalDistance 2+2+2+2+2 = 10 =
  2+1+1+0+2+1+2+1

Sum           5 5 5 5 5 5 5 5
```

Ta có:

$Score_i + TotalDistance_i = t$ với các cột

Suy ra:

$l \times (Score_j + TotalDistance_j) = l \times t$

Hay $Score = l \times t - TotalDistance$

Mà $l \times t$ là hằng, nên vế phải đạt cực tiểu tương đương vế trái đạt cực đại

- Tại sao lại quan tâm đến chuyện thay bài toán Motif Finding bằng Median String
  - Motif Finding Problem cần tính toán với tất cả các tổ hợp của **s**. Đó là $(\boldsymbol{n} - \ell + 1)^t$ tổ hợp.
  - Median String Problem cần tính toán 4^ℓ tổ hợp của ***v***. Con số này tương đối nhỏ hơn.

Các bước tìm motif

- Cho một trình tự v có chiều dài *l* (gọi là *l*-mer)
- Và cho *t* trình tự có chiều dài *n*
- Tính các khoảng cách Hamming $d_H(v,x)$, trong đó x là *l*-mer có vị trí bắt đầu lần lượt từ 1 đến *n*-*l*+1 trong trình tự thứ *i*

- Từ đây suy ra $d_H(v,x_i)$ là khoảng cách cực tiểu trong các khoảng cách này của trình tự *i*.
- Tính TotalDistance là tổng các $d_H(v,x_i)$ với *i* từ 1 đến *t*.
- Khi đó các x_i là các motif tìm được trên cơ sở trình tự v cho trước.

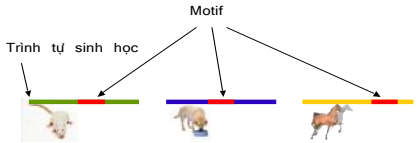
- Nhận xét:
  - Trong trường hợp v chưa biết trước, số lượng motif x_i cần tìm là quá ít so với tập hợp tìm kiếm.

KÝ HIỆU PROTEIN MOTIF

Ký hiệu protein motif

- x: được dùng để chỉ vị trí mà bất cứ amino acide nào cũng được chấp nhận
- []: tại vị trí này có thể là một trong các amino acide được liệt kê
- {}: tại vị trí này có thể bất kỳ amino acide nào ngoại trừ phân tử được liệt kê

- Ví dụ:
  - [AC]xVx(2){GV}
  - Là motif gồm: Alanine hoặc Cysteine - amino acide – Valine - amino acide -amino acide - Ngoại trừ Glutamate và Valine
- x(2): có 2 amino acide bất kỳ
- x(0,3): có từ 0 đến 3 amino acide bất kỳ

- <: cho biết motif nằm ở đầu trình tự protein
- >: cho biết motif nằm ở cuối trình tự protein
- Ví dụ: < Ax[ST](2)x(0,1)V
  - Motif nằm ở đầu trình tự gồm: Alanine – amino acide - Serine hoặc Threonine - Serine hoặc Threonine – có amino acide hoặc không – Valine

- Ví dụ:
  - Với l = 10, n = 1000, t = 8
  - Số mẫu *l*-mer cần tìm trong mỗi trình tự là 1000-10+1 = 991
  - Trong t trình tự có 8 x 991 = 7928 mẫu
  - Như vậy: chỉ tìm 8 mẫu (8 motif) trong 7928 mẫu