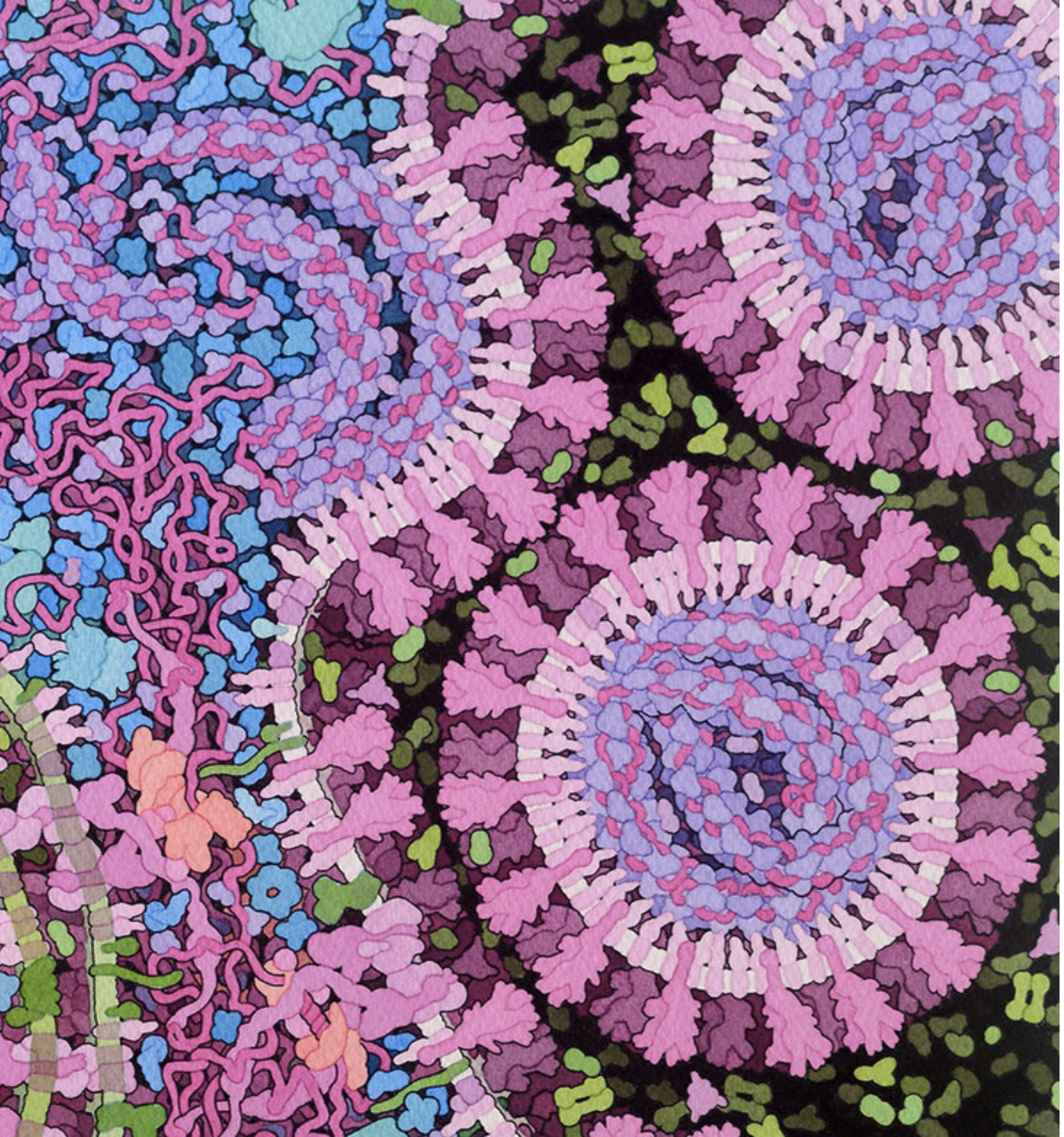


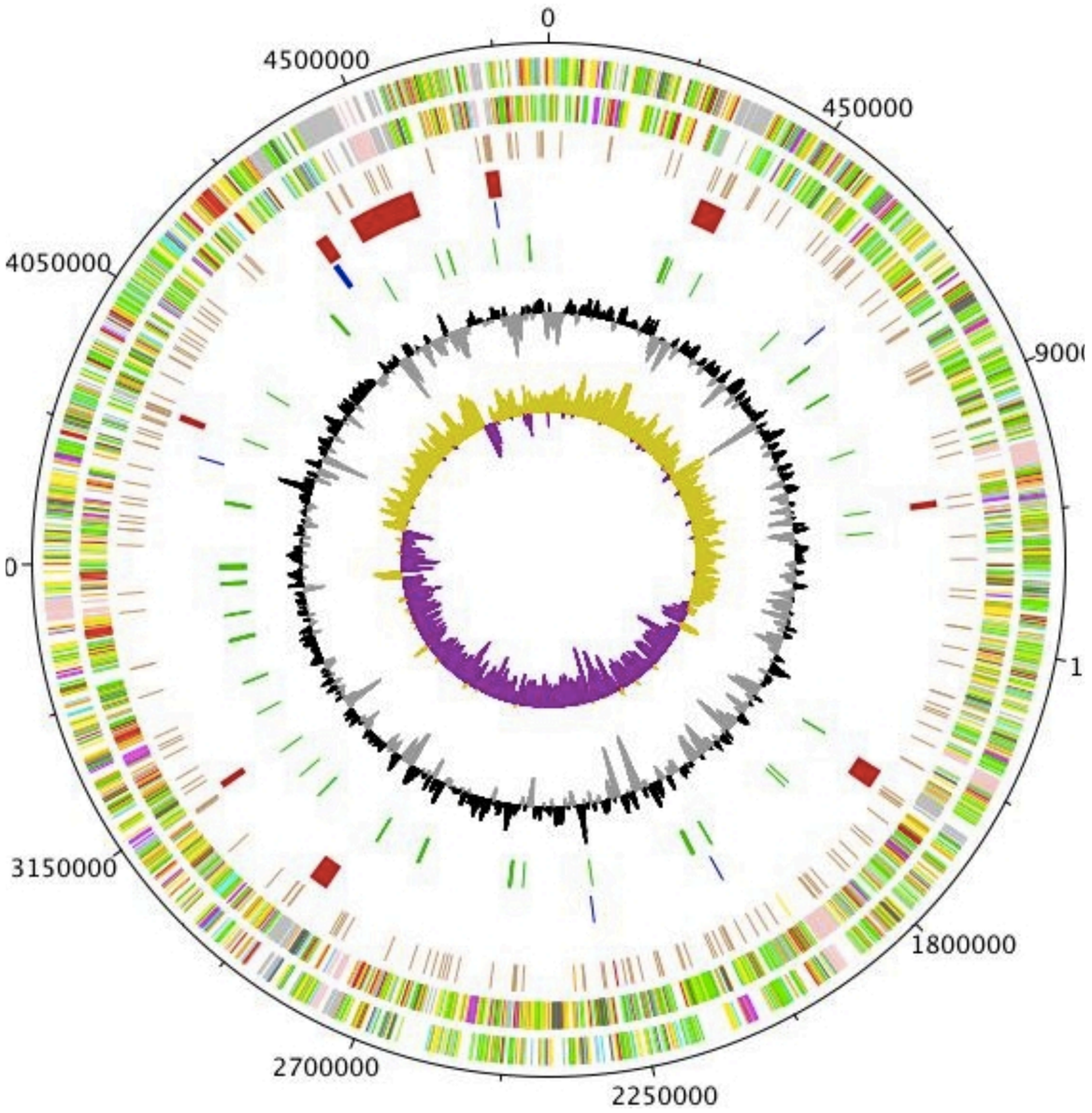
بسم الله الرحمن الرحیم

ژنومیک محاسباتی

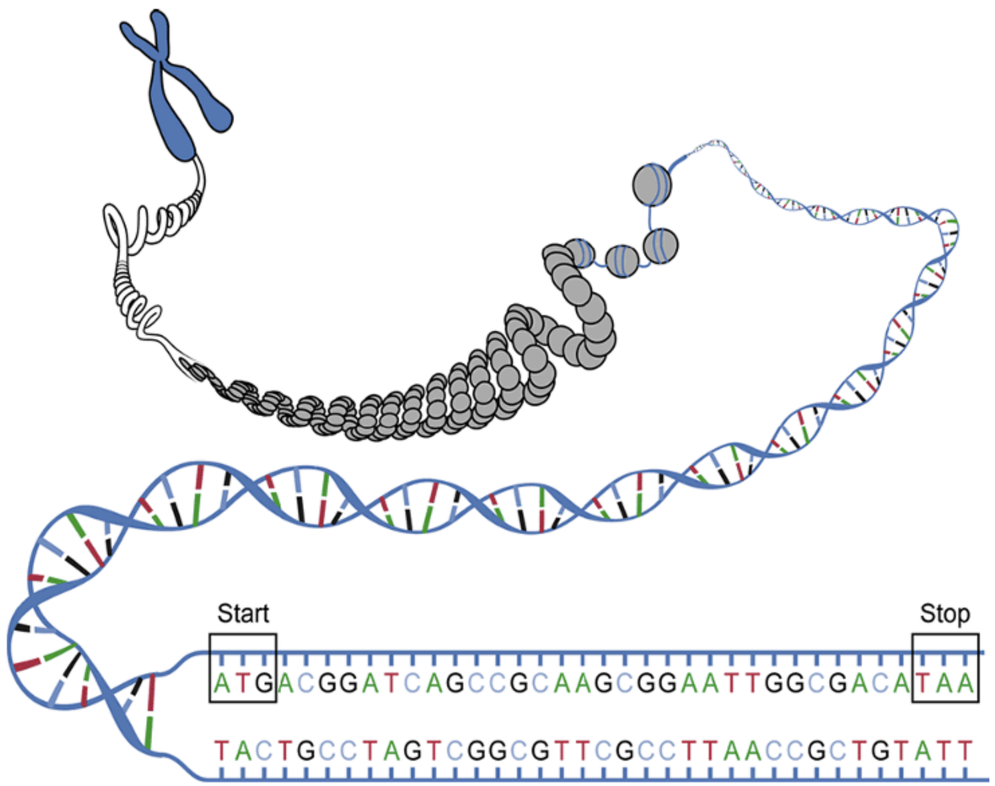
جلسه ۲: هم‌ترازی توالی

ترم پاییز ۱۴۰۰-۱۴۰۱

رشته‌ها

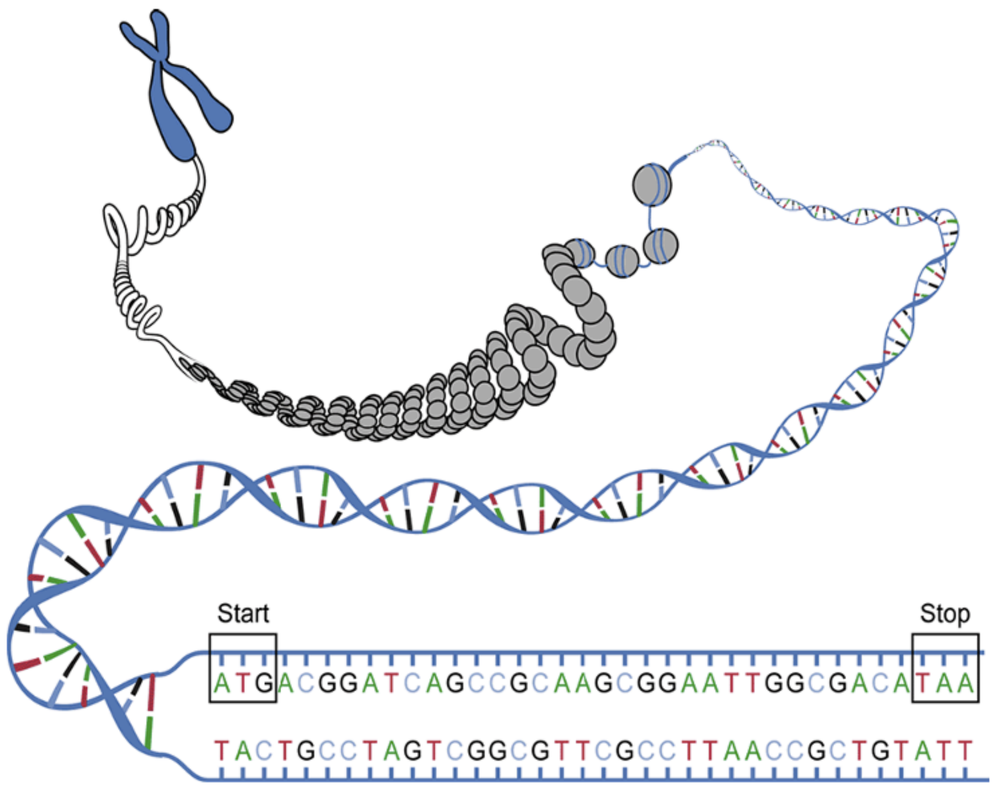

0
450000
900
1
1800000
2250000
2700000
3150000
0
4050000
4500000

رشته‌های DNA


Start
Stop
ATGACGGATCAGCCGCAAGCGGAATTGGCGACATAA
TACTGCCTAGTCGGCGTTCGCCTTAACCGCTGTATT

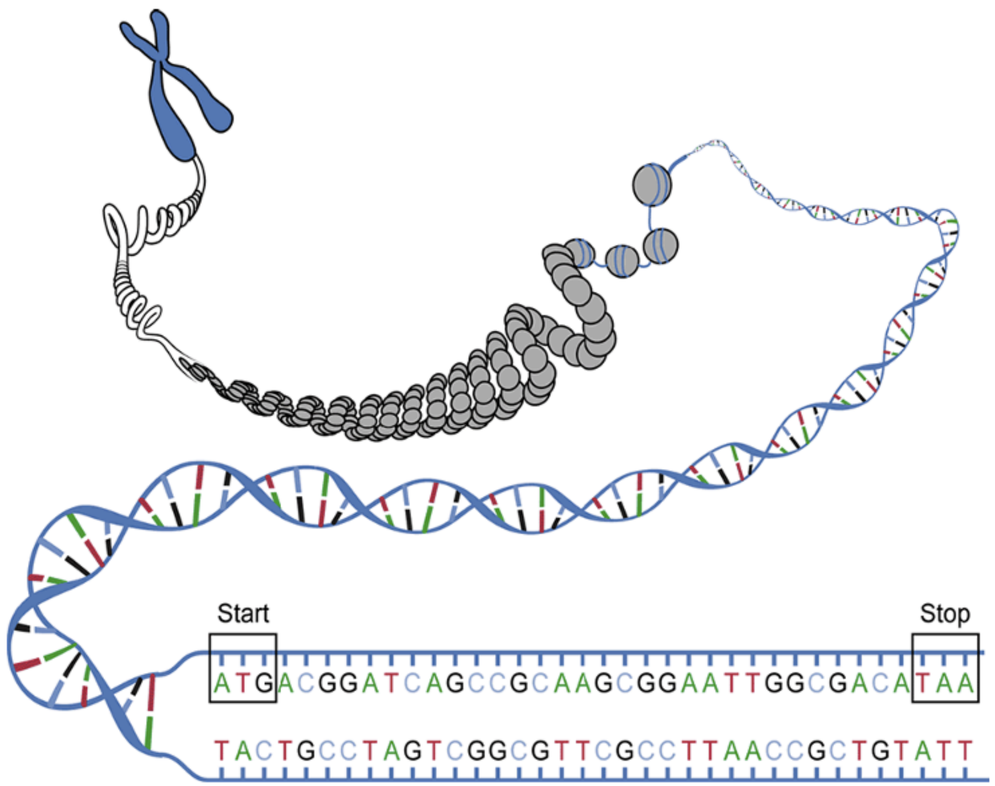
رشته‌های DNA

- توالی‌های ژنومی
- یک رشته از الفبای {A, T, C, G}

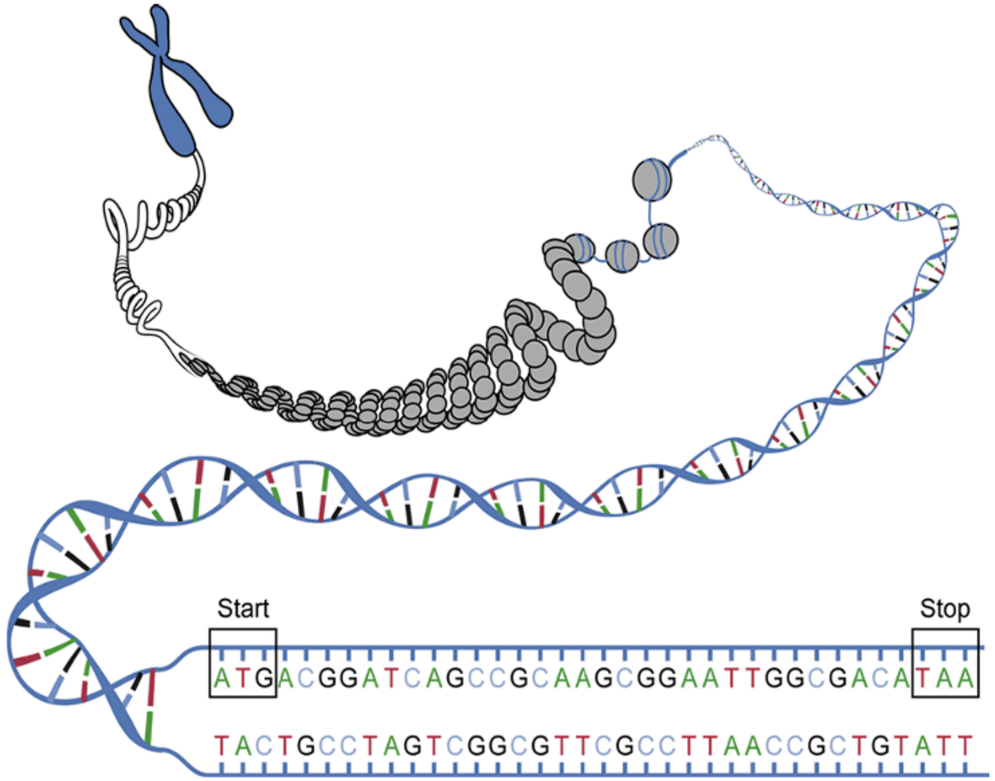


رشته‌های DNA

- توالی‌های ژنومی
- یک رشته از الفبای {A, T, C, G}
- ژنوم انسان = ۳ میلیارد (هر مجموعه کروموزومی)



رشته‌های DNA

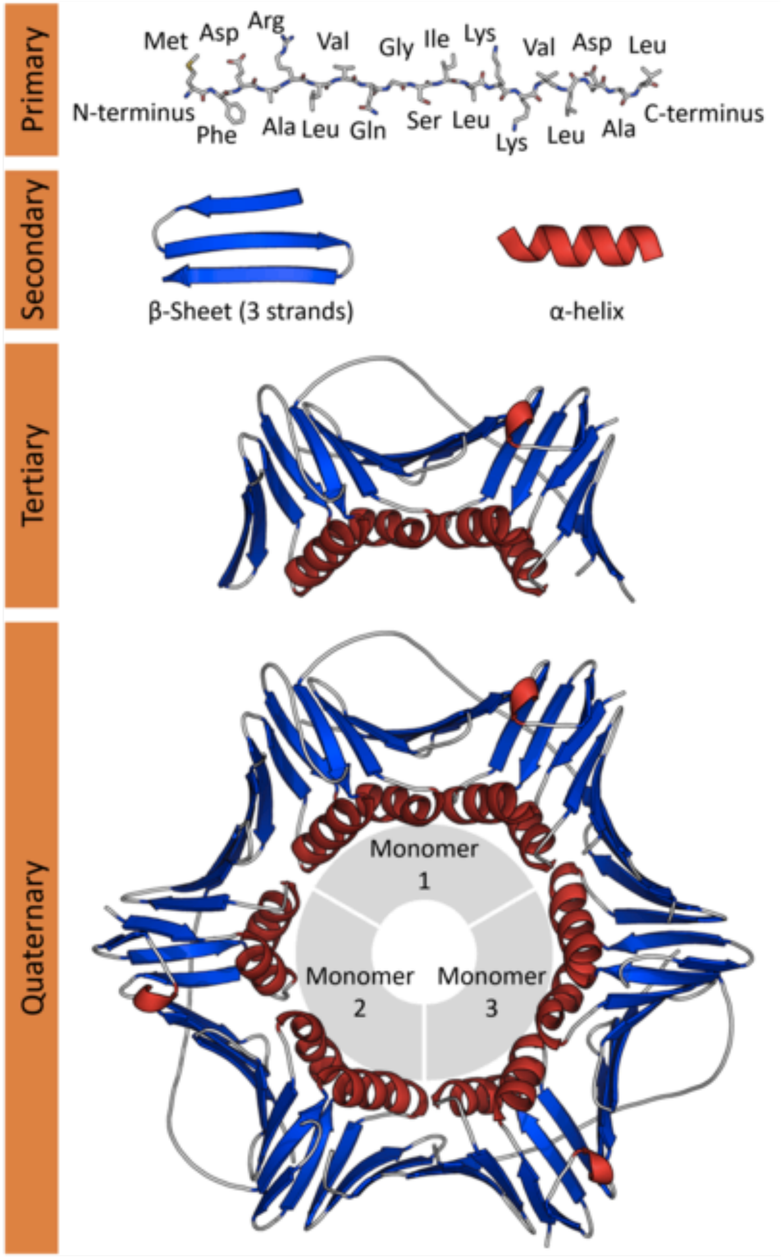


- توالی‌های ژنومی
- یک رشته از الفبای {A, T, C, G}
- ژنوم انسان = ۳ میلیارد (هر مجموعه کروموزومی)

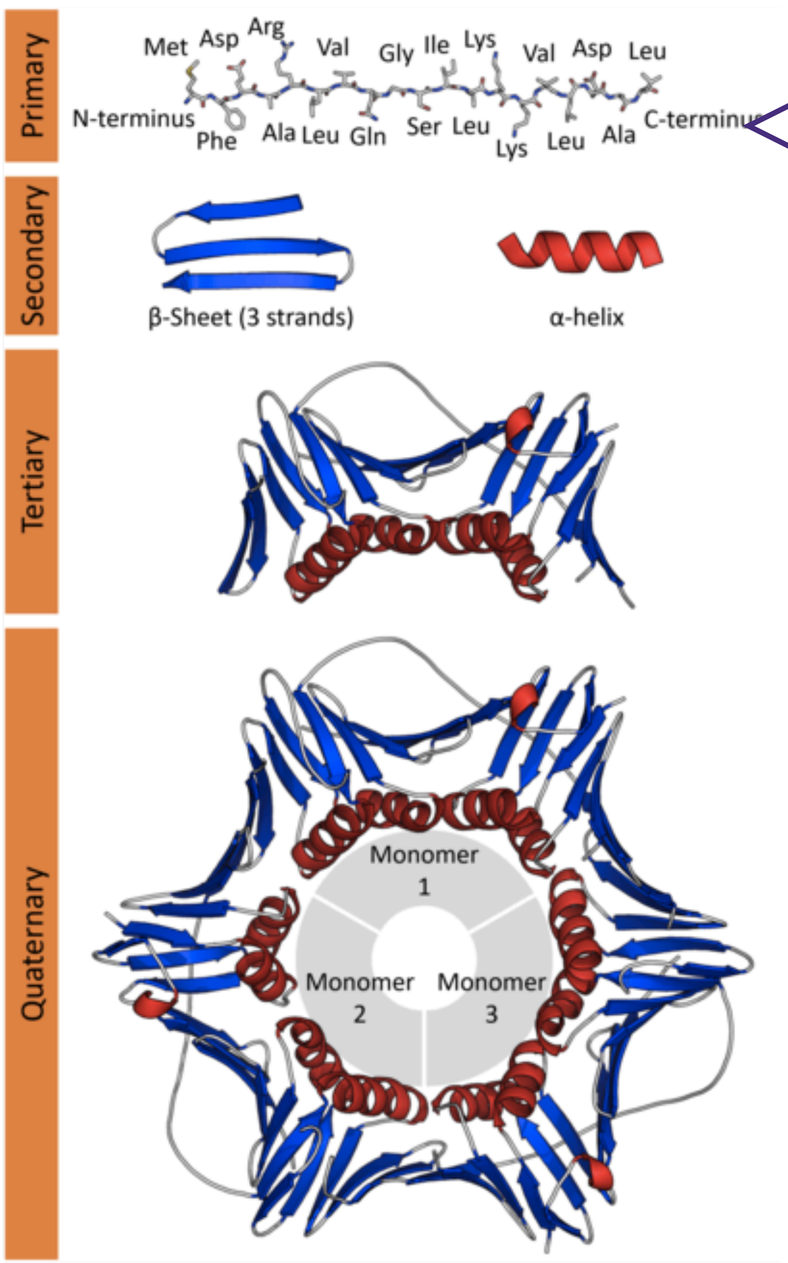
مثال

```
  1 atcttggctc acaggggacg atgtcaagct cttcctggct ccttctcagc cttgttgctg
 61 taactgctgc tcagtccacc attgaggaac aggccaagac atttttggac aagtttaacc
121 acgaagccga agacctgttc tatcaaagtt cacttgcttc ttggaattat aacaccaata
181 ttactgaaga gaatgtccaa aacatgaata atgctgggga caaatggtct gccttttaa
241 aggaacagtc cacacttgcc caaatgtatc cactacaaga aattcagaat ctcacagtca
```

رشته‌های پروتئینی



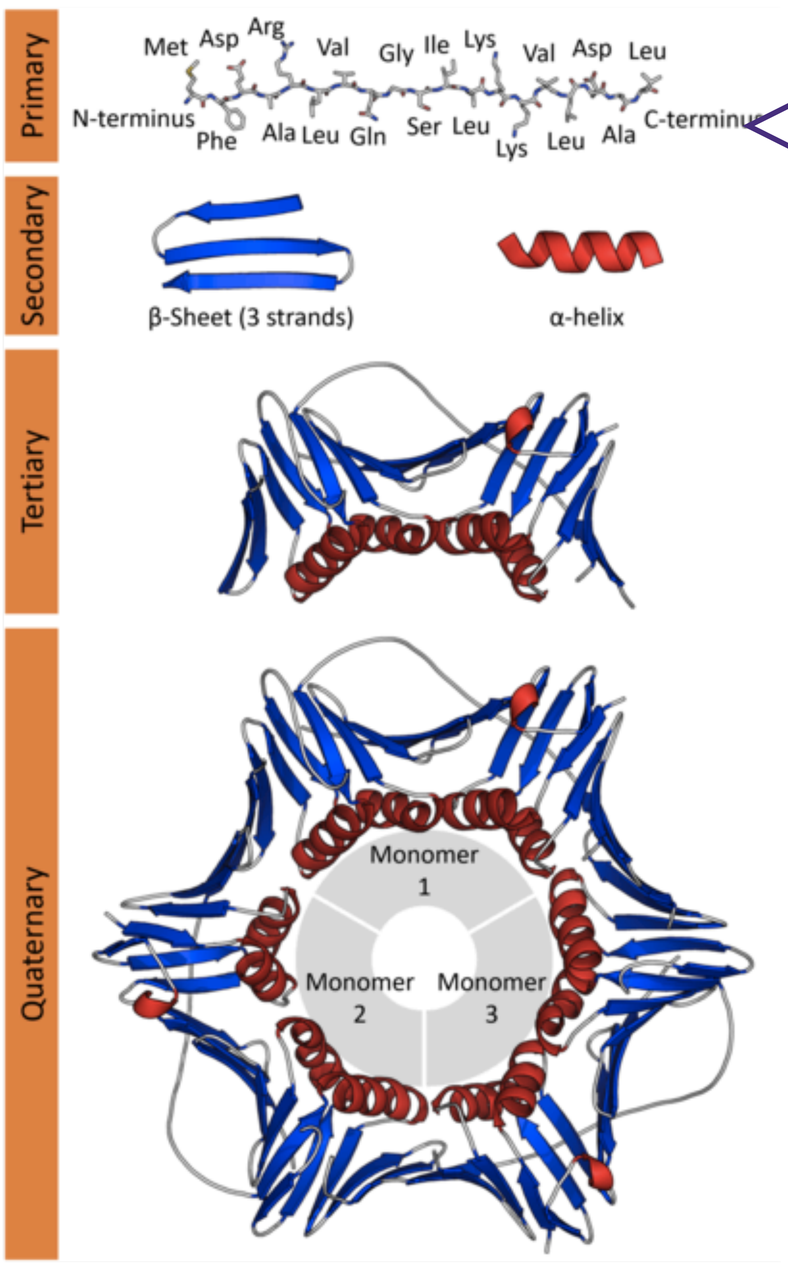
رشته‌های پروتئینی



Amino acid	3- and 1-	
	3	1
Alanine	Ala	A
Arginine	Arg	R
Asparagine	Asn	N
Aspartate	Asp	D
Cysteine	Cys	C
Glutamine	Gln	Q
Glutamate	Glu	E
Glycine	Gly	G
Histidine	His	H
Isoleucine	Ile	I
Leucine	Leu	L
Lysine	Lys	K
Methionine	Met	M
Phenylalanin	Phe	F
Proline	Pro	P
Serine	Ser	S
Threonine	Thr	T
Tryptophan	Trp	W
Tyrosine	Tyr	Y
Valine	Val	V

رشته‌های پروتئینی

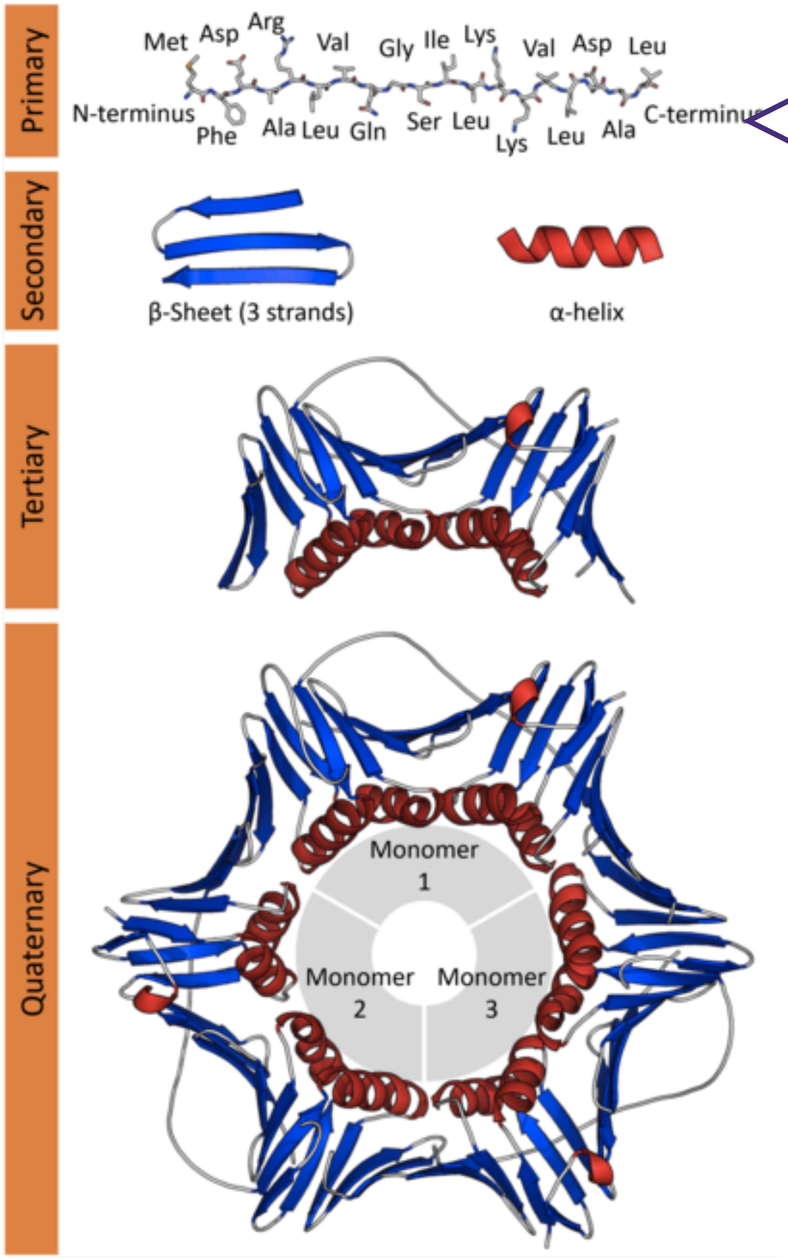
- توالی پروتئینی = یک رشته از الفبای
- A R N D C Q E G H I L K M F P S T W Y V
- ۲۰ حرفی



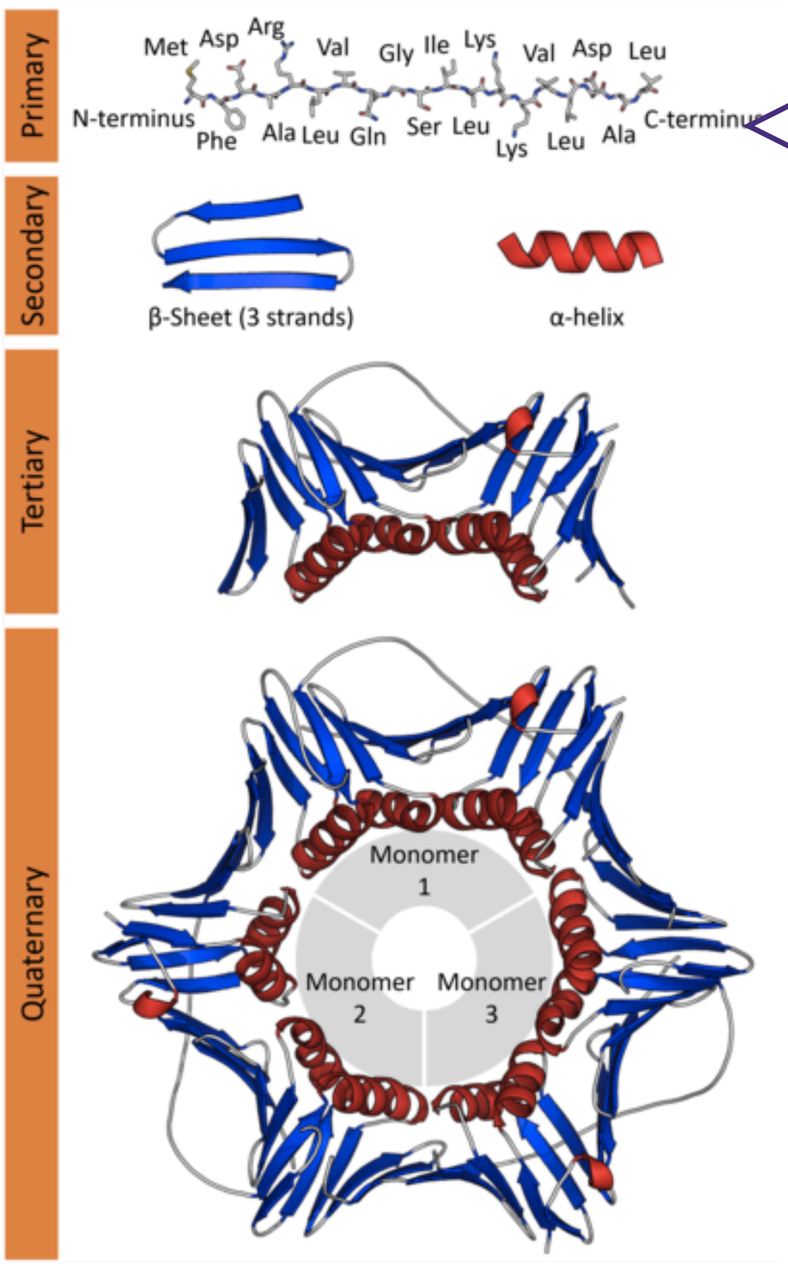
Amino acid	3- and 1-	
	3	1
Alanine	Ala	A
Arginine	Arg	R
Asparagine	Asn	N
Aspartate	Asp	D
Cysteine	Cys	C
Glutamine	Gln	Q
Glutamate	Glu	E
Glycine	Gly	G
Histidine	His	H
Isoleucine	Ile	I
Leucine	Leu	L
Lysine	Lys	K
Methionine	Met	M
Phenylalanin	Phe	F
Proline	Pro	P
Serine	Ser	S
Threonine	Thr	T
Tryptophan	Trp	W
Tyrosine	Tyr	Y
Valine	Val	V

رشته‌های پروتئینی

- توالی پروتئینی = یک رشته از الفبای
- A R N D C Q E G H I L K M F P S T W Y V
- ۲۰ حرفی
- طول < ۲۰۰۰



Amino acid	3- and 1-	
	3	1
Alanine	Ala	A
Arginine	Arg	R
Asparagine	Asn	N
Aspartate	Asp	D
Cysteine	Cys	C
Glutamine	Gln	Q
Glutamate	Glu	E
Glycine	Gly	G
Histidine	His	H
Isoleucine	Ile	I
Leucine	Leu	L
Lysine	Lys	K
Methionine	Met	M
Phenylalanin	Phe	F
Proline	Pro	P
Serine	Ser	S
Threonine	Thr	T
Tryptophan	Trp	W
Tyrosine	Tyr	Y
Valine	Val	V



Amino acid	3- and 1-	
	3	1
Alanine	Ala	A
Arginine	Arg	R
Asparagine	Asn	N
Aspartate	Asp	D
Cysteine	Cys	C
Glutamine	Gln	Q
Glutamate	Glu	E
Glycine	Gly	G
Histidine	His	H
Isoleucine	Ile	I
Leucine	Leu	L
Lysine	Lys	K
Methionine	Met	M
Phenylalanin	Phe	F
Proline	Pro	P
Serine	Ser	S
Threonine	Thr	T
Tryptophan	Trp	W
Tyrosine	Tyr	Y
Valine	Val	V

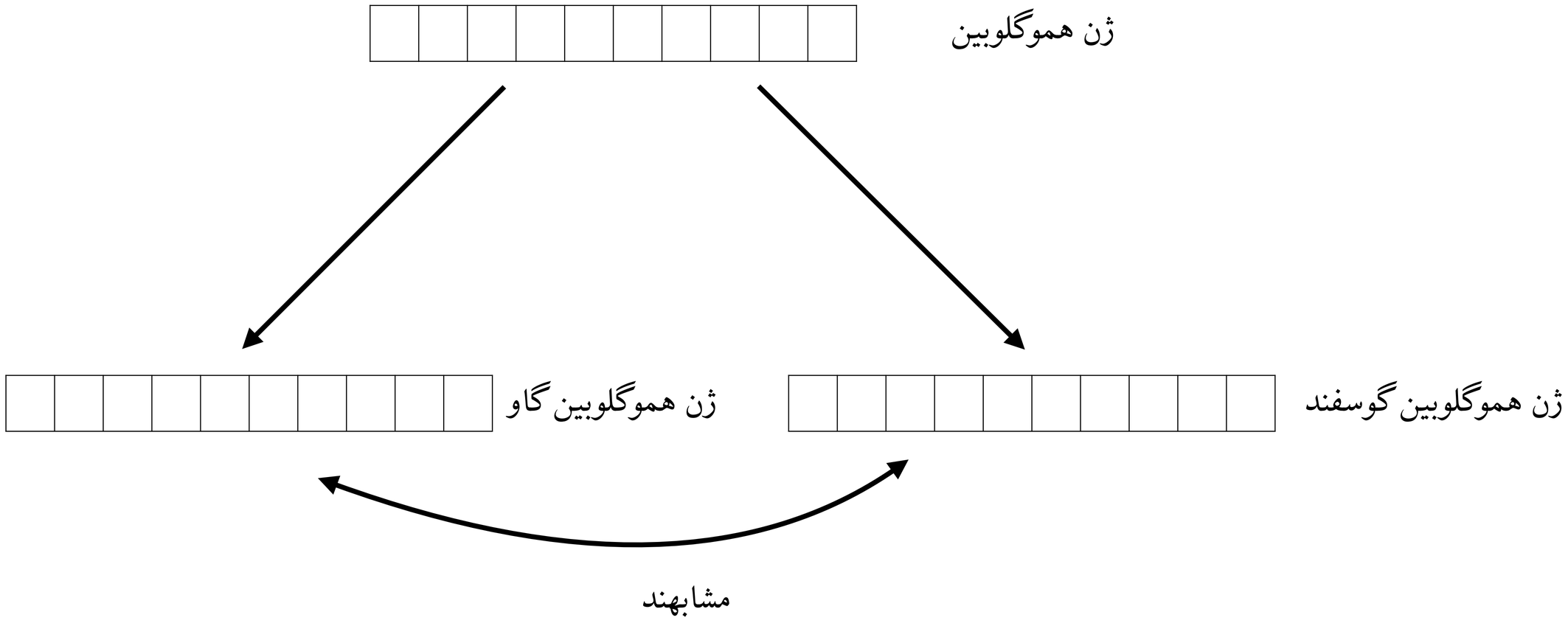
رشته‌های پروتئینی

- توالی پروتئینی = یک رشته از الفبای
- A R N D C Q E G H I L K M F P S T W Y V
- ۲۰ حرفی
- طول > ۲۰۰۰
- مثال:

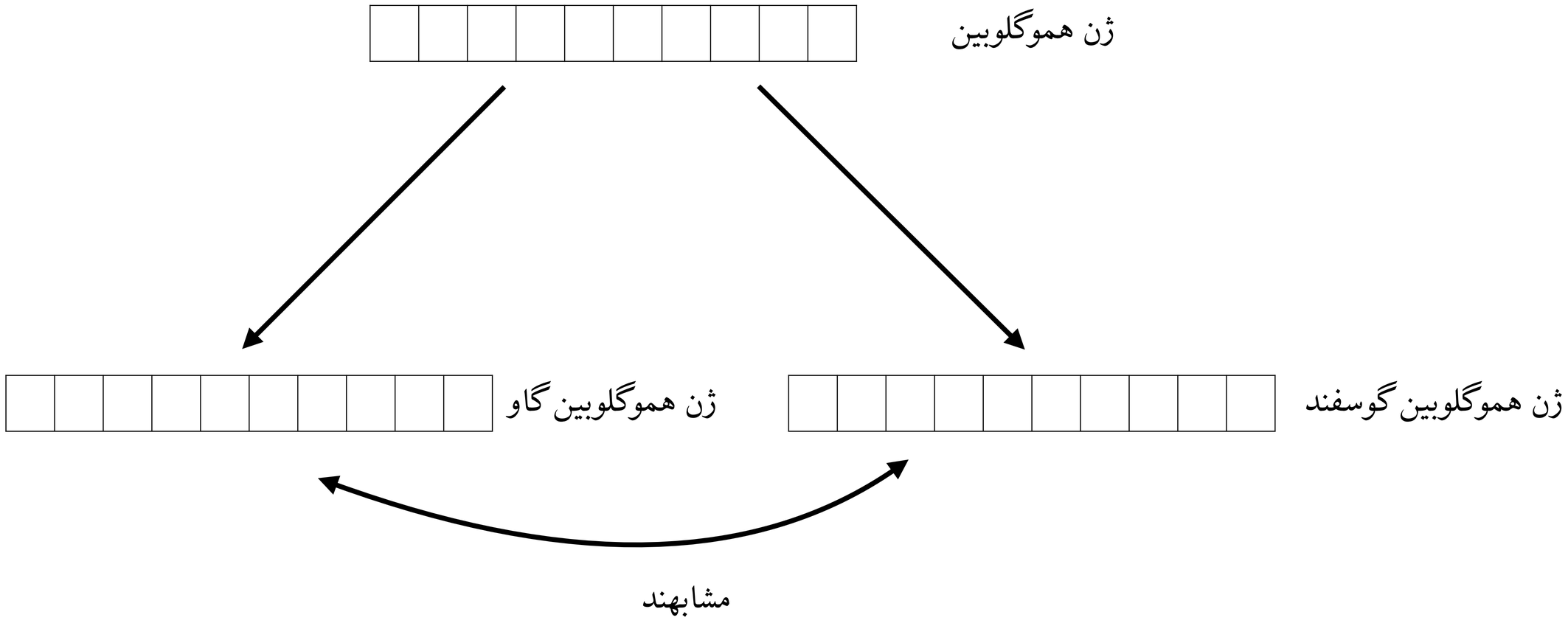
MSSSSWLLLSLVAVTAAQS TIEEQAKTFLDKFNHEAE DLFYQSS

هم‌ترازی

چرا هم‌ترازی مهم است؟



چرا هم‌ترازی مهم است؟



- شباهت => تکامل، شباهت => عملکرد یکسان، کمی شباهت => فاصله تکاملی، ...

مسئله فاصله ویرایشی

- ورودی: دو رشته S و T
- عملیات:
  - الف) حذف یک حرف
  - ب) اضافه کردن یک حرف
  - ج) تغییر یک حرف به حرف دیگر
- خروجی: کم‌ترین عملیات لازم برای تبدیل S به T

مسئله فاصله ویرایشی

- ورودی: دو رشته S و T
- عملیات:
  - الف) حذف یک حرف
  - ب) اضافه کردن یک حرف
  - ج) تغییر یک حرف به حرف دیگر
- خروجی: کم‌ترین عملیات لازم برای تبدیل S به T

- مثال:
- S = ACAAT CC
- T = AGCAT GC

مسئله فاصله ویرایشی

- ورودی: دو رشته S و T
- عملیات:
  - الف) حذف یک حرف
  - ب) اضافه کردن یک حرف
  - ج) تغییر یک حرف به حرف دیگر
- خروجی: کم‌ترین عملیات لازم برای تبدیل S به T

- مثال:
- S = ACAAT CC
- T = AGCAT GC
- روش تبدیل ۱: سه تا تغییر

مسئله فاصله ویرایشی

- ورودی: دو رشته S و T
- عملیات:
  - الف) حذف یک حرف
  - ب) اضافه کردن یک حرف
  - ج) تغییر یک حرف به حرف دیگر
- خروجی: کم‌ترین عملیات لازم برای تبدیل S به T

- مثال:
  - S = ACAAT CC
  - T = AGCAT GC
  - روش تبدیل ۱: سه تا تغییر
  - روش تبدیل ۲: اضافه کردن G، حذف A دوم، تغییر C به G

مسئله هم‌ترازی ساده

- ورودی: دو رشته S و T
- خروجی:
  - یک جدول
    - سطر اول S که تعدادی «ـ» به آن اضافه شده
    - سطر اول T که تعدادی «ـ» به آن اضافه شده

s1	-	s2	s3	s4	-	-	-	s5	-
-	T1	T2	T3	-	-	T4	T5	T6	T7

  - دارای بیشترین ستون‌های مساوی

- مثال:
- S = ACAAT CC
- T = AGCAT GC

مسئله هم‌ترازی ساده

- ورودی: دو رشته S و T
- خروجی:
  - یک جدول
    - سطر اول S که تعدادی «ـ» به آن اضافه شده
    - سطر اول T که تعدادی «ـ» به آن اضافه شده

s1	-	s2	s3	s4	-	-	-	s5	-
-	T1	T2	T3	-	-	T4	T5	T6	T7

- دارای بیشترین ستون‌های مساوی

- مثال:
  - S = ACAAT CC
  - T = AGCAT GC
  - مثال هم‌ترازی:

$$S = \texttt{A-CAATCC}$$
$$T = \texttt{AGCA-TGC}$$

«مسئله فاصله ویرایشی» = «مسئله هم‌ترازی ساده»

$$S = \texttt{A-CAATCC}$$
$$T = \texttt{AGCA-TGC}$$

- قضیه: اگر امتیاز هم‌ترازی بهینه a* باشد و هزینه روش ویرایش بهینه e* باشد داریم: $e^* = |T_a| - a^*$
- ایده اثبات:
  - هر هم‌ترازی ====> روش ویرایش
    - $e = |T_a| - a$
  - هر روش ویرایش ====> هم‌ترازی
    - $a \geq |T_a| - e$

سرتاسری

مسئله هم‌ترازی

- ورودی: دو رشته S و T و ماتریس مشابهت حروف
$\delta : (\Sigma \cup \{-\})^2 \to \mathbb{R}$
- خروجی:
  - یک جدول
    - سطر اول S که تعدادی «_» به آن اضافه شده
    - سطر اول T که تعدادی «_» به آن اضافه شده

s1	-	s2	s3	s4	-	-	-	s5	-
-	T1	T2	T3	-	-	T4	T5	T6	T7

  - دارای بیشترین امتیاز
    - امتیاز = جمع امتیاز ستون‌ها (بر اساس δ)

سرتاسری

مسئله هم‌ترازی

- ورودی: دو رشته S و T و ماتریس مشابهت حروف
$\delta : (\Sigma \cup \{-\})^2 \to \mathbb{R}$
- خروجی:
  - یک جدول
    - سطر اول S که تعدادی «ـ» به آن اضافه شده
    - سطر اول T که تعدادی «ـ» به آن اضافه شده

s1	-	s2	s3	s4	-	-	-	s5	-
-	T1	T2	T3	-	-	T4	T5	T6	T7

  - دارای بیشترین امتیاز
    - امتیاز = جمع امتیاز ستون‌ها (بر اساس δ)

- مثال:
  - S = ACAAT CC
  - T = AGCAT GC

δ

	-	A	C	G	T
-		-1	-1	-1	-1
A	-1	2	-1	-1	-1
C	-1	-1	2	-1	-1
G	-1	-1	-1	2	-1
T	-1	-1	-1	-1	2

سرتاسری

مسئله هم‌ترازی

- ورودی: دو رشته S و T و ماتریس مشابهت حروف
$\delta : (\Sigma \cup \{-\})^2 \to \mathbb{R}$
- خروجی:
  - یک جدول
  - سطر اول S که تعدادی «ـ» به آن اضافه شده
  - سطر اول T که تعدادی «ـ» به آن اضافه شده

s1	-	s2	s3	s4	-	-	-	s5	-
-	T1	T2	T3	-	-	T4	T5	T6	T7

- دارای بیشترین امتیاز
  - امتیاز = جمع امتیاز ستون‌ها (بر اساس δ)

- مثال:
  - S = ACAAT CC
  - T = AGCAT GC
- مثال هم‌ترازی:

$$S = \texttt{A-CAATCC}$$
$$T = \texttt{AGCA-TGC}$$

δ

	-	A	C	G	T
-		-1	-1	-1	-1
A	-1	2	-1	-1	-1
C	-1	-1	2	-1	-1
G	-1	-1	-1	2	-1
T	-1	-1	-1	-1	2

سرتاسری

مسئله هم‌ترازی

- ورودی: دو رشته S و T و ماتریس مشابهت حروف
$\delta : (\Sigma \cup \{-\})^2 \to \mathbb{R}$
- خروجی:
  - یک جدول
  - سطر اول S که تعدادی «ـ» به آن اضافه شده
  - سطر اول T که تعدادی «ـ» به آن اضافه شده

s1	-	s2	s3	s4	-	-	-	s5	-
-	T1	T2	T3	-	-	T4	T5	T6	T7

- دارای بیشترین امتیاز
  - امتیاز = جمع امتیاز ستون‌ها (بر اساس δ)

- مثال:
  - S = ACAAT CC
  - T = AGCAT GC
- مثال هم‌ترازی:

$$S = \texttt{A-CAATCC}$$
$$T = \texttt{AGCA-TGC}$$

- امتیاز: ؟؟؟

δ

	-	A	C	G	T
-		-1	-1	-1	-1
A	-1	2	-1	-1	-1
C	-1	-1	2	-1	-1
G	-1	-1	-1	2	-1
T	-1	-1	-1	-1	2

الگوریتم Needleman-Wunsch

- ورودی: دو رشته S و T و ماتریس مشابهت حروف $\delta : (\Sigma \cup \{-\})^2 \to \mathbb{R}$

الگوریتم Needleman-Wunsch

- ورودی: دو رشته S و T و ماتریس مشابهت حروف $\delta : (\Sigma \cup \{-\})^2 \to \mathbb{R}$
- روش: برنامه‌نویسی پویا

الگوریتم Needleman-Wunsch

- ورودی: دو رشته S و T و ماتریس مشابهت حروف $\delta : (\Sigma \cup \{-\})^2 \to \mathbb{R}$
- روش: برنامه‌نویسی پویا
  - تعریف $V(i,j)$: بیشترین امتیاز هم‌ترازی بین $S[1..i]$ و $T[1..j]$

الگوریتم Needleman-Wunsch

- ورودی: دو رشته S و T و ماتریس مشابهت حروف $\delta : (\Sigma \cup \{-\})^2 \to \mathbb{R}$
- روش: برنامه‌نویسی پویا
  - تعریف $V(i,j)$: بیشترین امتیاز هم‌ترازی بین $S[1..i]$ و $T[1..j]$
  - رابطه بازگشتی:

$$V(i,j) = \max \begin{cases} V(i-1,j-1) + \delta(S[i],T[j]) & \texttt{match}/mismatch \\ V(i-1,j) + \delta(S[i],_) & \texttt{d}elete \\ V(i,j-1) + \delta(_,T[j]) & \texttt{i}nsert \end{cases}$$

الگوریتم Needleman-Wunsch

- ورودی: دو رشته S و T و ماتریس مشابهت حروف $\delta : (\Sigma \cup \{-\})^2 \to \mathbb{R}$
- روش: برنامه‌نویسی پویا
  - تعریف $V(i,j)$: بیشترین امتیاز هم‌ترازی بین $S[1..i]$ و $T[1..j]$
  - رابطه بازگشتی:

$$V(i,j) = \max \begin{cases} V(i-1,j-1) + \delta(S[i],T[j]) & \texttt{match}/mismatch \\ V(i-1,j) + \delta(S[i],_) & \texttt{d}elete \\ V(i,j-1) + \delta(_,T[j]) & \texttt{i}nsert \end{cases}$$

- حالت پایه:

$$V(0,0) = 0$$
$$V(0,j) = V(0,j-1) + \delta(_,T[j]) \text{ Insert } j \text{ times}$$
$$V(i,0) = V(i-1,0) + \delta(S[i],_) \quad \text{Delete } i \text{ times}$$

Needleman-Wunsch مثال الگوریتم

T = AGCATGC

S = AC AAT C C

	_	A	G	C	A	T	G	C
_	0	-1	-2	-3	-4	-5	-6	-7
A	-1	2	1	0	-1	-2	-3	-4
C	-2	1	1	3	2	1	0	-1
A	-3	0	0	2	5	4	3	2
A	-4	-1	-1	1	4	4	3	2
T	-5	-2	-2	0	3	6	5	4
C	-6	-3	-3	0	2	5	5	7
C	-7	-4	-4	-1	1	4	4	7

δ

	-	A	C	G	T
-		-1	-1	-1	-1
A	-1	2	-1	-1	-1
C	-1	-1	2	-1	-1
G	-1	-1	-1	2	-1
T	-1	-1	-1	-1	2

محاسبه خود بهترین هم‌ترازی؟

T = AGCATGC

S = AC AAT C C

	_	A	G	C	A	T	G	C
_	0	-1	-2	-3	-4	-5	-6	-7
A	-1	2	1	0	-1	-2	-3	-4
C	-2	1	1	3	2	1	0	-1
A	-3	0	0	2	5	4	3	2
A	-4	-1	-1	1	4	4	3	2
T	-5	-2	-2	0	3	6	5	4
C	-6	-3	-3	0	2	5	5	7
C	-7	-4	-4	-1	1	4	4	7

الگوریتم Needleman-Wunsch

- ورودی: دو رشته S و T و ماتریس مشابهت حروف $\delta : (\Sigma \cup \{-\})^2 \to \mathbb{R}$
- روش: برنامه‌نویسی پویا
  - تعریف $V(i,j)$: بیشترین امتیاز هم‌ترازی بین $S[1..i]$ و $T[1..j]$
  - رابطه بازگشتی:

$$V(i,j) = \max \begin{cases} V(i-1,j-1) + \delta(S[i],T[j]) & \texttt{match}/mismatch \\ V(i-1,j) + \delta(S[i],_) & \texttt{d}elete \\ V(i,j-1) + \delta(_,T[j]) & \texttt{i}nsert \end{cases}$$

- حالت پایه:

$$V(0,0) = 0$$
$$V(0,j) = V(0,j-1) + \delta(_,T[j]) \text{ Insert } j \text{ times}$$
$$V(i,0) = V(i-1,0) + \delta(S[i],_) \quad \text{Delete } i \text{ times}$$

زمان اجرا:

الگوریتم Needleman-Wunsch

- ورودی: دو رشته S و T و ماتریس مشابهت حروف $\delta : (\Sigma \cup \{-\})^2 \to \mathbb{R}$
- روش: برنامه‌نویسی پویا
  - تعریف $V(i,j)$: بیشترین امتیاز هم‌ترازی بین $S[1..i]$ و $T[1..j]$
  - رابطه بازگشتی:

$$V(i,j) = \max \begin{cases} V(i-1,j-1) + \delta(S[i],T[j]) & \texttt{match}/mismatch \\ V(i-1,j) + \delta(S[i],_) & \texttt{d}elete \\ V(i,j-1) + \delta(_,T[j]) & \texttt{i}nsert \end{cases}$$

- حالت پایه:

$$V(0,0) = 0$$
$$V(0,j) = V(0,j-1) + \delta(_,T[j]) \text{ Insert } j \text{ times}$$
$$V(i,0) = V(i-1,0) + \delta(S[i],_) \text{ Delete } i \text{ times}$$

زمان اجرا:
O(mn)

الگوریتم Needleman-Wunsch

- ورودی: دو رشته S و T و ماتریس مشابهت حروف $\delta : (\Sigma \cup \{-\})^2 \to \mathbb{R}$
- روش: برنامه‌نویسی پویا
  - تعریف $V(i,j)$: بیشترین امتیاز هم‌ترازی بین $S[1..i]$ و $T[1..j]$
  - رابطه بازگشتی:

$$V(i,j) = \max \begin{cases} V(i-1,j-1) + \delta(S[i],T[j]) & \texttt{match}/mismatch \\ V(i-1,j) + \delta(S[i],_) & \texttt{delete} \\ V(i,j-1) + \delta(_,T[j]) & \texttt{insert} \end{cases}$$

- حالت پایه:

$$V(0,0) = 0$$
$$V(0,j) = V(0,j-1) + \delta(_,T[j]) \text{ Insert } j \text{ times}$$
$$V(i,0) = V(i-1,0) + \delta(S[i],_) \text{ Delete } i \text{ times}$$

زمان اجرا:
O(mn)

حافظه:

الگوریتم Needleman-Wunsch

- ورودی: دو رشته S و T و ماتریس مشابهت حروف $\delta : (\Sigma \cup \{-\})^2 \to \mathbb{R}$
- روش: برنامه‌نویسی پویا
  - تعریف $V(i,j)$: بیشترین امتیاز هم‌ترازی بین $S[1..i]$ و $T[1..j]$
  - رابطه بازگشتی:

$$V(i,j) = \max \begin{cases} V(i-1,j-1) + \delta(S[i],T[j]) & \texttt{match}/mismatch \\ V(i-1,j) + \delta(S[i],_) & \texttt{delete} \\ V(i,j-1) + \delta(_,T[j]) & \texttt{insert} \end{cases}$$

- حالت پایه:

$$V(0,0) = 0$$
$$V(0,j) = V(0,j-1) + \delta(_,T[j]) \text{ Insert } j \text{ times}$$
$$V(i,0) = V(i-1,0) + \delta(S[i],_) \text{ Delete } i \text{ times}$$

الگوریتم Needleman-Wunsch

زمان اجرا:
O(mn)

حافظه:
O(mn)

- ورودی: دو رشته S و T و ماتریس مشابهت حروف $\delta : (\Sigma \cup \{-\})^2 \to \mathbb{R}$
- روش: برنامه‌نویسی پویا
  - تعریف $V(i,j)$: بیشترین امتیاز هم‌ترازی بین $S[1..i]$ و $T[1..j]$
  - رابطه بازگشتی:

$$V(i,j) = \max \begin{cases} V(i-1,j-1) + \delta(S[i],T[j]) & \texttt{match}/mismatch \\ V(i-1,j) + \delta(S[i],_) & \texttt{delete} \\ V(i,j-1) + \delta(_,T[j]) & \texttt{insert} \end{cases}$$

- حالت پایه:

$$V(0,0) = 0$$
$$V(0,j) = V(0,j-1) + \delta(_,T[j]) \text{ Insert } j \text{ times}$$
$$V(i,0) = V(i-1,0) + \delta(S[i],_) \text{ Delete } i \text{ times}$$