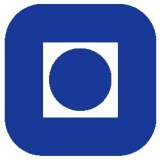


NTNU

Innovation and Creativity

TDT4287
Algorithms for bioinformatics

Autumn 2021

Pål Sætrom

Dept. of Computer Science

Dept. of Clinical and Molecular Medicine

Today

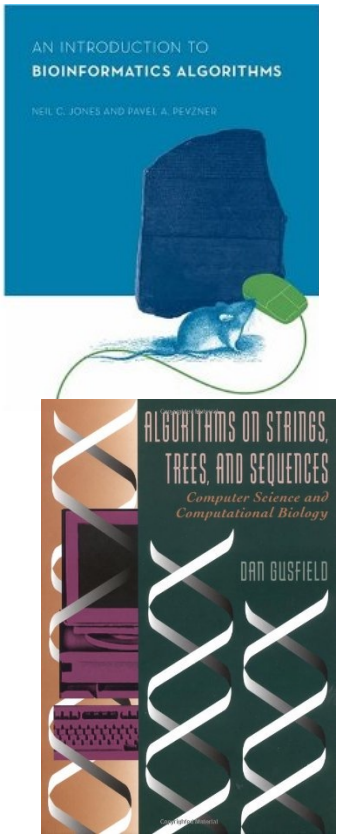
- (Practical matters)
- Bioinformatics?
- Genes and sequences
- Comparing sequences

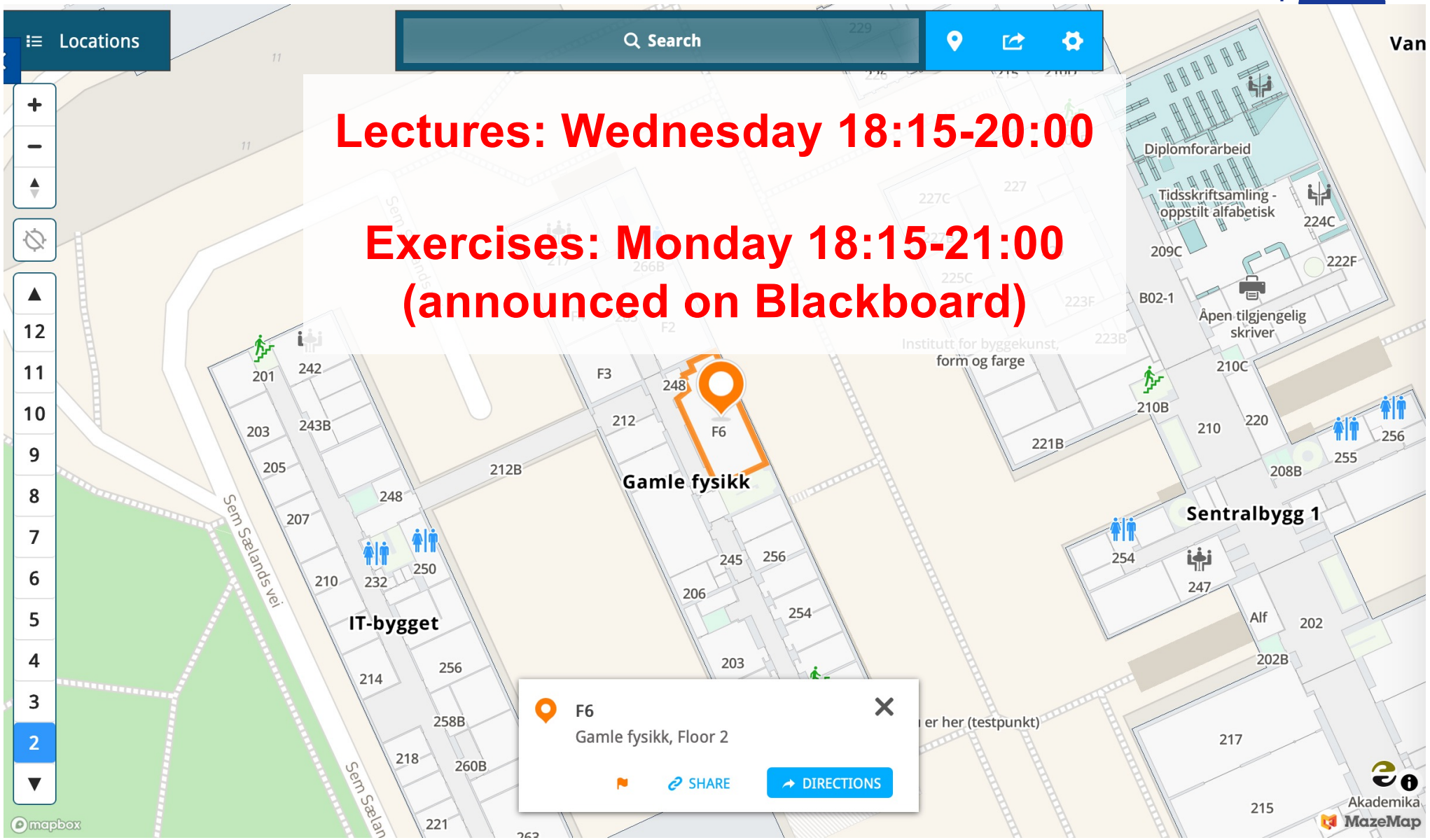
Goals:

- Molecular biology for computer scientists
- Basic sequence comparison (and why)

Course material and contents

- Exam:
  - Written (or home)
  - Code D simple calculator only
  - Date: TBD
- Course material
  - An Introduction to Bioinformatics Algorithms
  - Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology
  - Handouts (blackboard)
- Mandatory project




Locations
Search
Lectures: Wednesday 18:15-20:00
Exercises: Monday 18:15-21:00
(announced on Blackboard)
Gamle fysikk
IT-bygget
Sentralbygg 1
Diplomforarbeid
Tidsskriftsamling - oppstilt alfabetisk
Åpen tilgjengelig skriver
Institutt for byggekunst, form og farge
Sem Sælands vei
F6
Gamle fysikk, Floor 2
SHARE
DIRECTIONS
er her (testpunkt)
Alf
mapbox
Akademika
MazeMap

Bioinformatics?

Webster:

bio·in·for·mat·ics (bī'ō-ĭn'fər-măt'ĭks)
Function: noun plural but singular in construction
: the collection, classification, storage, and analysis of biochemical and **biological information** using **computers** especially as applied in molecular genetics and genomics

- Bioinformatics
  - Tools
  - Data
- Computational biology
  - Biological questions
  - Tools, models, and data are the means

Bioinformatics – this course

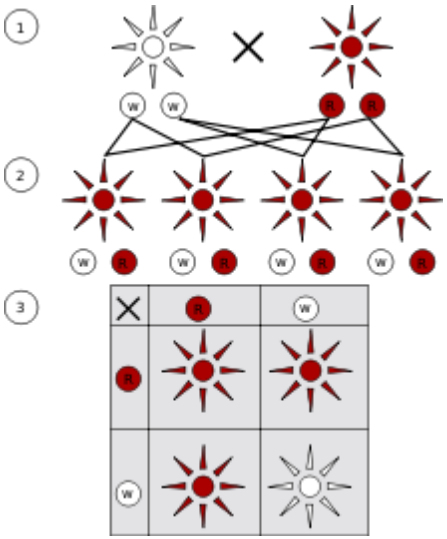
- Sequence (string) analysis
  - Alignment -> finding similar sequences
  - Suffix trees -> index-based searching
  - Motif finding -> similarities between sequences
  - Assembly -> reconstructing sequences from fragments
  - RNA structures -> similarities within a sequence

Week	Topic	Curiculum
34	Introduction	
35	Alignment	Chps. 6.4-6.9 in Jones & Pevzner
36	Heuristic alignment	Chps. 9.3, 9.6-9.8 in Jones & Pevzner
37	BLAST/Statistical analysis	
38	Substring indexes	Chps. 9.4-9.5 in Jones & Pevzner; Handouts (available on itslearning).
39	Suffix tree applications	Chps. 9.4-9.5 in Jones & Pevzner; Handouts.
40	Sequencing and assembly; Project info	Chps. 8.1-8.9 in Jones & Pevzner
41	Markov Chains	Handouts; Ewens and Grant
42	Hidden Markov Models	Ewens and Grant
43	HMMs continued	
44	Alignment 2	Chps. 5.5, 12.2-12.3; 6.10, 7.2-7.4 in Jones & Pevzner
45	Motif discovery	Chps. 4.4-4.9, 12.2-12.3 in Jones & Pevzner
46	Motif discovery - ctd.	
47	RNA secondary structure prediction	Chps. 5.1-5.4; 6.11-6.14 in Jones & Pevzner; Article 1 Article 2
...	Exam	

- *Why sequence analysis?*

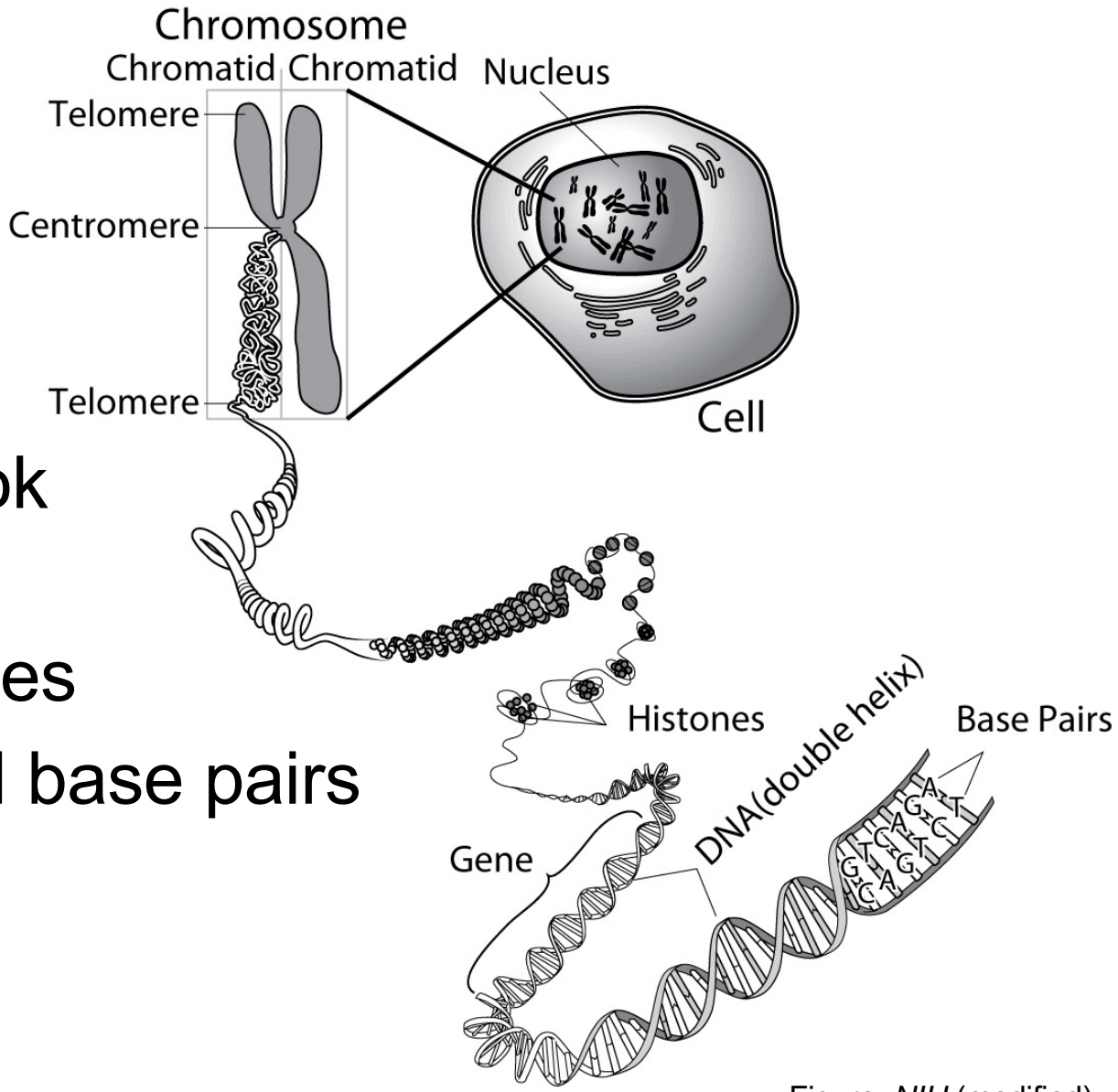
Genes control heritable characteristics

- Gene (Gregor J. Mendel - 1866)
  - Abstract unit of heredity
  - Alternative forms (alleles) -> different (physical) characteristics
  - Genotype + environment -> phenotype



Pictures: *Wikipedia*

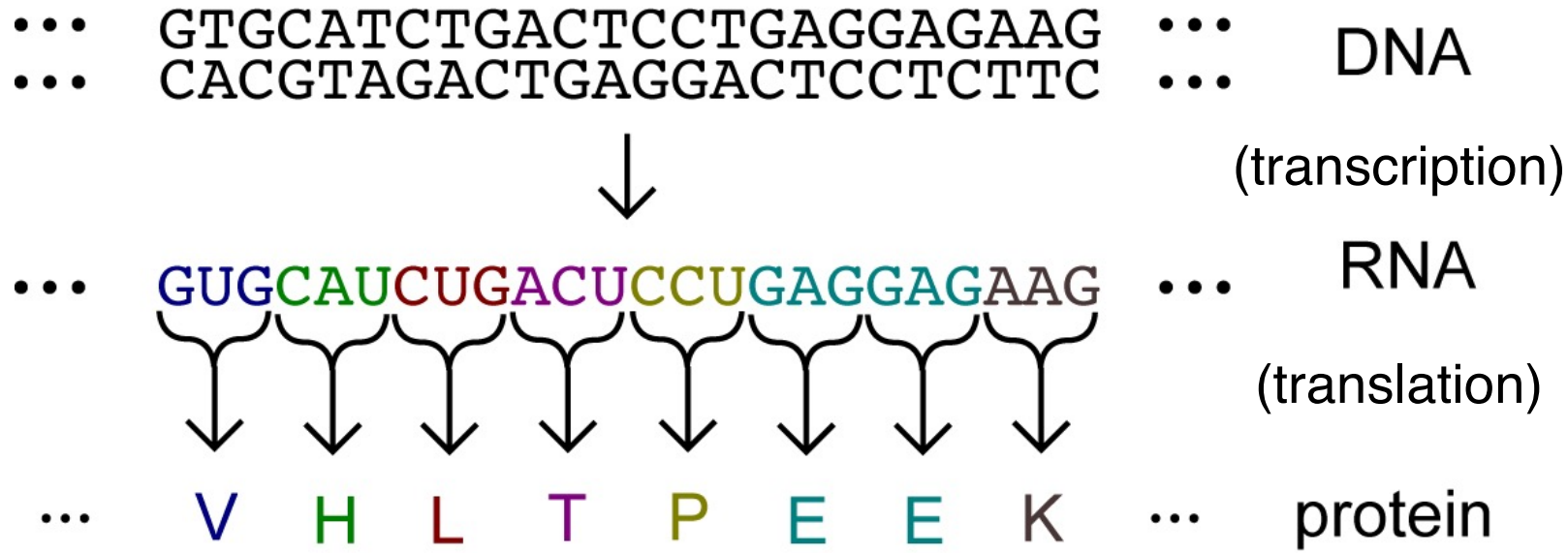
Genes are in DNA

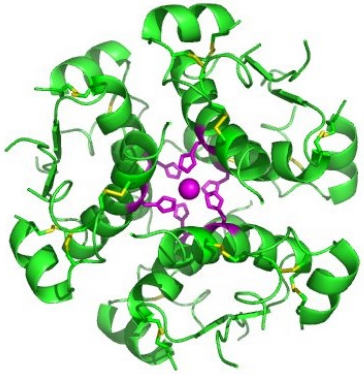


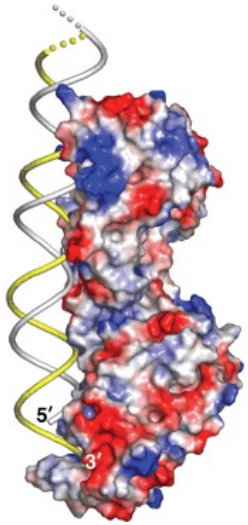
- Cells' cookbook
  - "Book of life"
- 23 chromosomes
- 3 * 10^9 ordered base pairs
- Self-copying
  - *Mechanism?*

Figure: *NIH* (modified)

Genes are recipes for molecules







Figures: *Wikipedia, MacRae*

MacRae, I.J. (2006). *Science* **311**: 195-198.

A cell as seen by a computer scientist

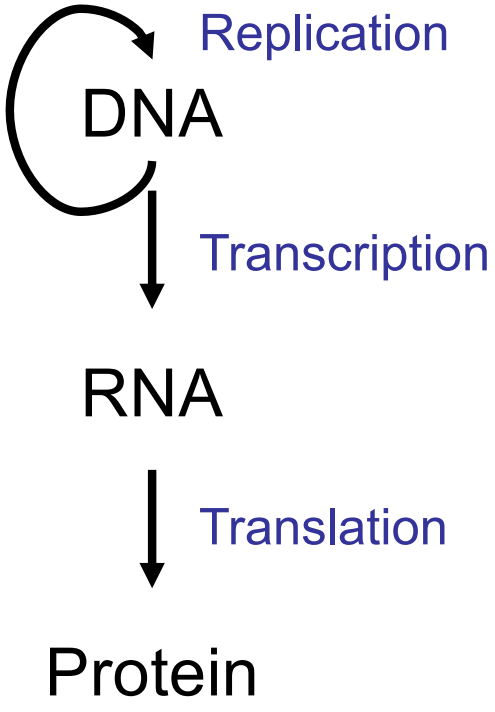
…CCAGCTGCTTCGGGCTGCCGAGGACCTTCTGGGCCCCCA
CATTAATGAGGCAGCCACCTGGCGAGTCTGACATGGCTGT
CAGCGACGCGCTGCTCCCATCTTTCTCCACGTTCGCGTC…

f_{TR}

CAUUAAUGAGGCAGCCACCUGGCGAGUCUGAC**AUGGCUGU**
CAGCGACGCGCUGCUCCCAUCUUUCUCCACGUUCGCGUC…

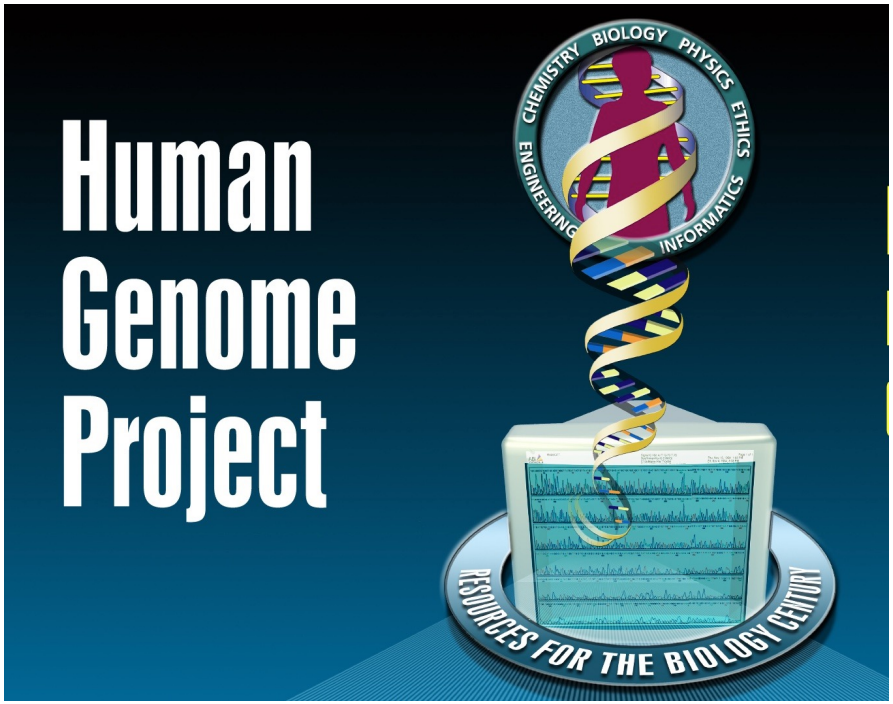
f_{TL}

MRQPPGESDMAVSDA…



If you know the DNA-sequence (the genome),

you will also know the genes!

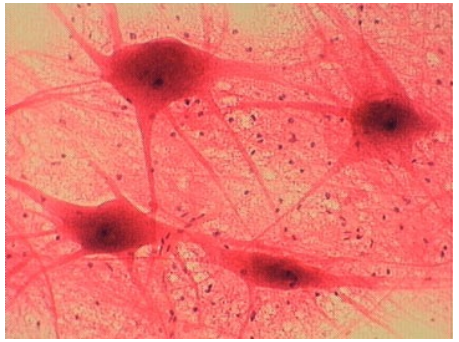


…and explain traits and disease?

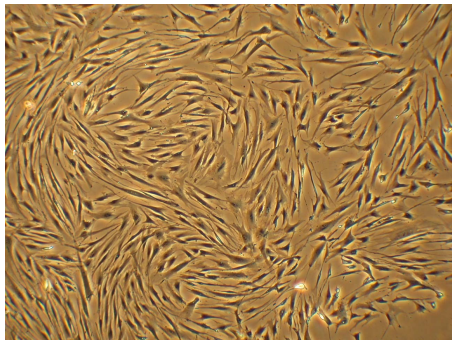
Identical twins have the same genes, but are still different

411 cell types, 10^{14} cells – same genes!

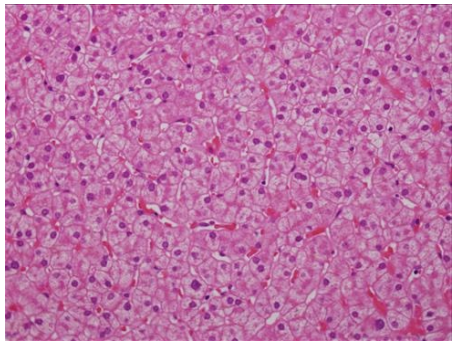
Brain



Connective tissue



Liver



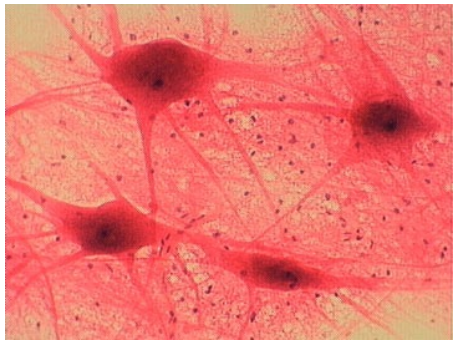
Muscle

Vickaryous & Hall (2006). *Biol Rev Camb Philos Soc* **81**: 425-55.

Pictures:
Photo Researchers, Inc., Iowa State Univ,.
Stephanie Saade, USA Today

Different genes turned on in different cell types

Brain

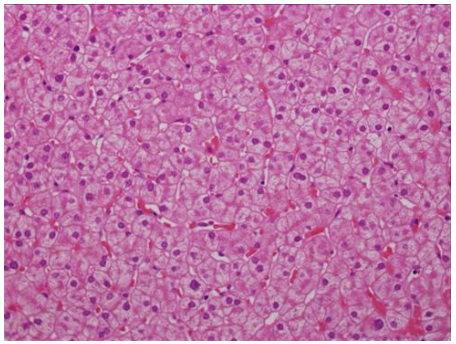


Doublecortin
Myosin
Albumin

Muscle

Doublecortin
Myosin
Albumin

Liver



Doublecortin
Myosin
Albumin

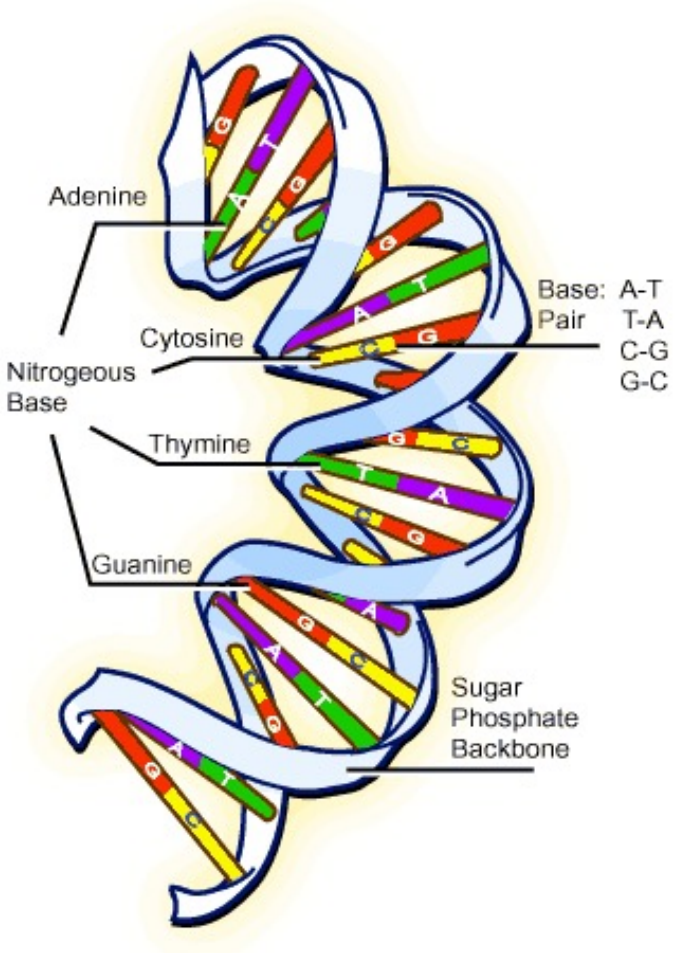
List of biological sequence problems

- How do genes encode proteins?
- What is the function of a protein?
- How do we recognize genes in DNA?
- How are genes controlled so that proteins are produced at the correct place and time?

Facts about genes (1960)

- Genes are stored in DNA
- Genes encode proteins (& RNAs)
- RNA is information carrier

- Proteins?



http://www.scq.ubc.ca/?p=263

Protein structure

- Sequence (string) of amino acids
  - 20 different amino acids
  - Defined direction (H_2N->COOH or N->C)
  - *Primary structure*
  - First sequence: Insulin (Sanger, 1955)
- Amino acid sequence folds into (unique), reproducible 3D structure
  - Tertiary structure
- Tertiary structure essential
  - Binding sites
  - Structure determines function

- How do genes encode proteins?

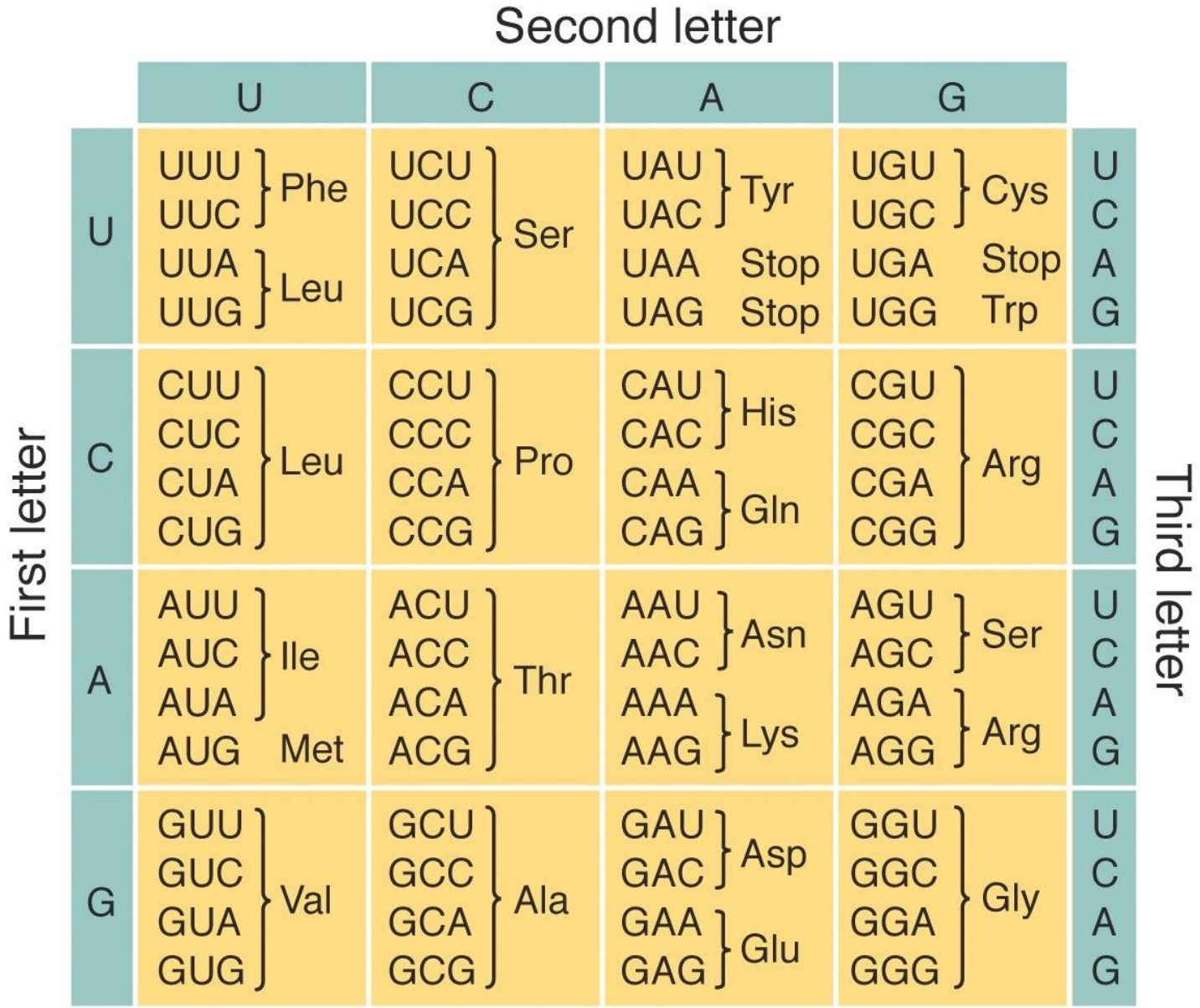
Genes (DNA) and proteins are sequences

- DNA (4 nucleotides)
- Proteins (20 amino acids)
- Genes (DNA) encode proteins:
  - Mapping f : DNA → Protein
- $f = ?$

The genetic code

- …must use at least 3 nucleotides to encode all 20 amino acids ($4^2 = 16$)
- …has some redundancy ($4^3 = 64$)
- …signal for start and stop

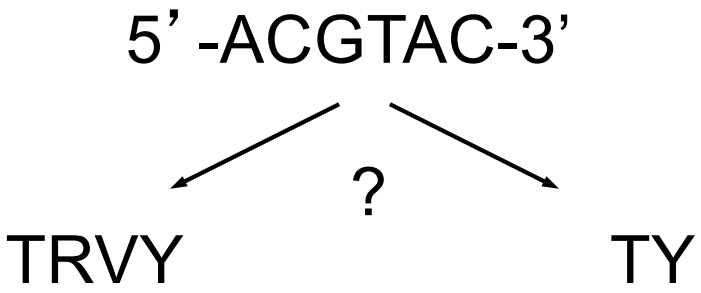
The genetic code (Nirenberg & Khorana, 1965)



Steven M. Carr, Memorial University of Newfoundland

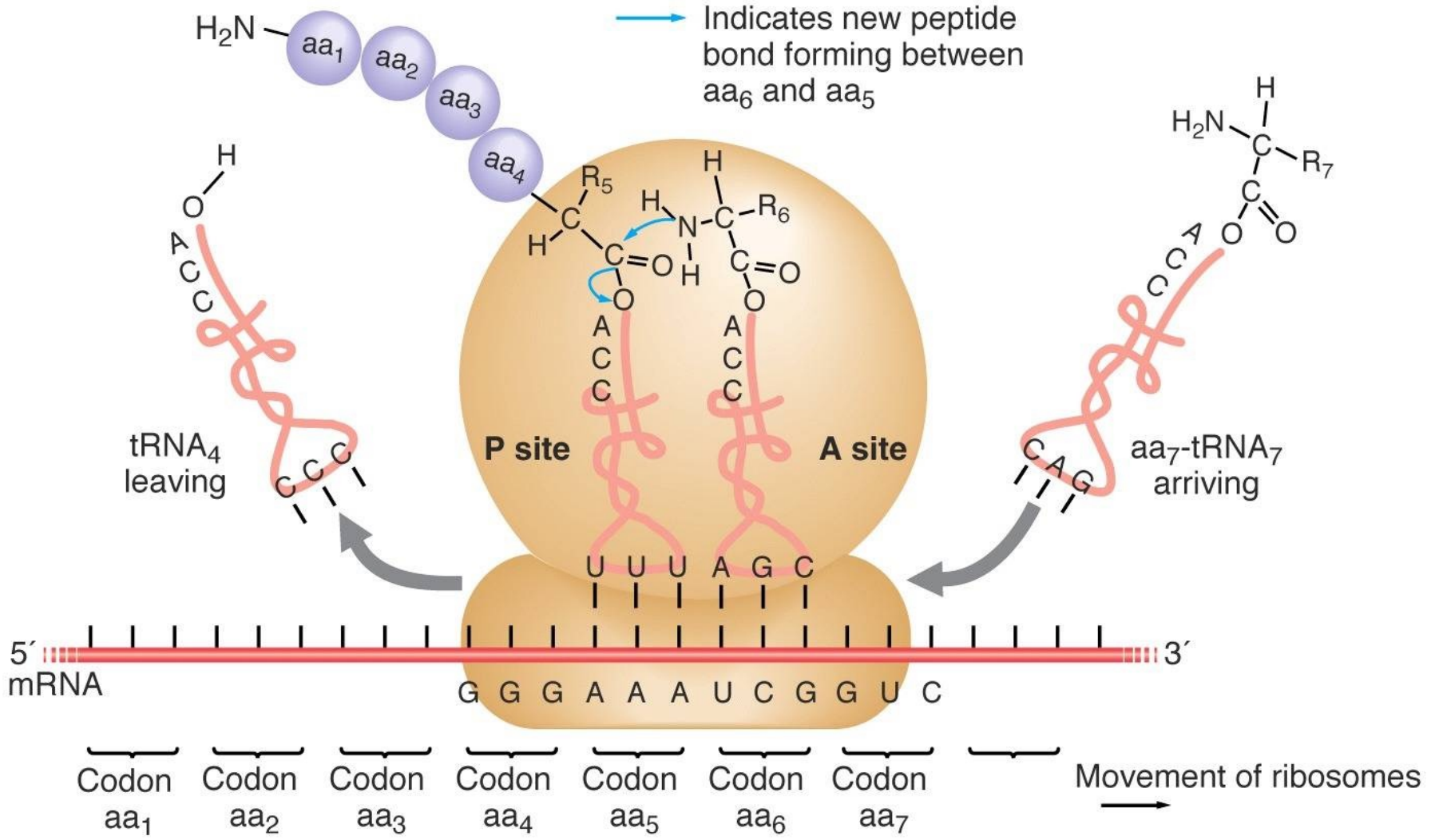
Reading the genetic code?

- Overlapping or exclusive words?



- Hints:
  - Adding or deleting 1 nucleotide destroys gene
  - Adding and deleting 1 nucleotide may be OK

Ribosomes translate mRNA to proteins



Steven M. Carr, Memorial University of Newfoundland

Changes in the recipe can affect function

```
... GTGCATCTGACTCCTGAGGAGAAG ...

... GTGCATCTGACACCTGAGGAGAAG ...

... GTGCATCTGAC-CCTGAGGAGAAG ...

... GTGCATCTGACGTCCTGAGGAGAAG ...
```

Which of the above could have the strongest effect on function?

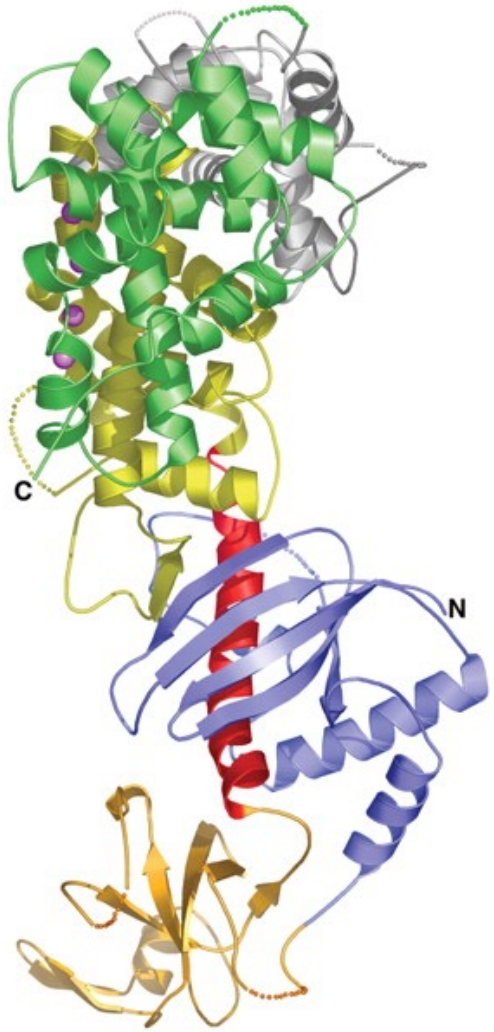
List of biological sequence problems

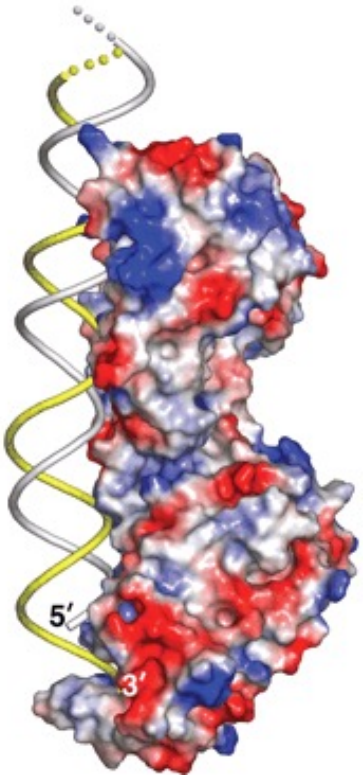
- ✓ How do genes encode proteins?
- ❑ What is the function of a protein?
- ❑ How do we recognize genes in DNA?
- ❑ How are genes controlled so that proteins are produced at the correct times?

Protein sequence determines function

Sequence → Structure → Function

>2FFL *Giardia intestinalis* Dicer
GAMHALGHCCTVVTTRGPSHWLLLLDTHLGTLPGF
KVSAGRGLPAAEVYFEAGPRVSLSRTDATIVAVYQ
SILFQLLGPTFPASWTEIGATMPHNEYTFPRFISN
PPQFATLAFLPLLSPTSPLDLRALMVTAQLMCDAK
RLSDEYTDYSTLSASLHGRMVATPEISWSLYVVLG
IDSTQTSLSYFTRANESITYMRYYATAHNIHLRAA
DLPLVAAVRLDDLKDHQIPAPGSWDDLAPKLRFLP
PELCLLLPDEFDLIRVQALQFLPEIAKHICDIQNT
ICALDKSFPDCGRIGGERYFAITAGLRLDQGRGRG
LAGWRTPFGPFGVSHTDVFQRLELLGDAVLGFIVT
ARLLCLFPDASVGTLVELKMELVRNEALNYLVQTL
GLPQLAEFSNNLVAKSKTWADMYEEIVGSIFTGPN
GIYGCEEFLAKTLMSPEHSKTVGSACPDAVTKASK
RVCMGEAGAHEFRSLVDYACEQGISVFCSSRVSTM
FLERLRDIPAEDMLDWYRLGIQFSHRSGLSGPGGV
VSVIDIMTHLARGLWLGSPGFYVEQQTDKNESACP
PTIPVLYIYHRSVQCPVLYGSLTETPTGPVASKVL
ALYEKILAYESSGGSKHIAAQTVSRSLAVPIPSGT
IPFLIRLLQIALTPHVYQKLELLGDAFLKCSLALH
LHALHPTLTEGALTRMRQSAETNSVLGRLTKRFPS
VVSEVIIESHPKIQPDSKVYGDTFEAILAAILLAC
GEEAAGAFVREHVLPQVVADA





RCSB PDB (http://www.rcsb.org/pdb/)

MacRae, I.J. (2006). *Science* **311**: 195-198.

Search to determine gene function

- Genes evolved from common ancestor – *homologs* – often have similar functions
- *Homologous* genes
  - *Orthologs* -> Genes in separate species
  - *Paralogs* -> Duplicated genes
- Homologous genes often have sequence similarity
  - Sequence similarity indicates homolgy
  - …and similar function

Evaluating sequence similarity?

String terminology

- String
  - $S = s_1 \ldots s_n$, $s_i \in A$, $|S| = n$
  - $S =$ ACGTTAGCT
- Substring of S
  - $S_{i,m} = s_{1+i} \ldots s_{m+i}$, $0 \leq i$, $i + m \leq n$
  - $S_{1,4} =$ CGTT
- Prefix of S
  - $S_i = s_1 \ldots s_i$, $0 \leq i \leq n$
  - $S_3 =$ ACG
- Suffix of S
  - $S_j = s_{1+j} \ldots s_n$, $0 \leq j \leq n$
  - $S_3 =$ TTAGCT

Subsequences

Subsequence: Ordered sequence of characters from string

`AGCTTAGCTG`

`ACTGT`, `CTTA`, `GCG` are subsequences

`GACA`, `TCA`, `ATGA` are not

Set of indexes:

`ACTGT` = {1,3,4,7,9}

Common subsequences

Common subsequence of two strings: subsequence in both strings

Formally:

$S = s_1 \ldots s_n$, $R = r_1 \ldots r_m$

$1 \le i_1 < i_2 < \ldots < i_k \le n$ (subsequence in S)

$1 \le j_1 < j_2 < \ldots < j_k \le m$ (subsequence in R)

$s_{it} = r_{jt}$, for $1 \le t \le k$.

AG**C**TTA**G**C**TG**, T**CG**GA**TG**

CGTG is a common subsequence

Longest common subsequence (LCS)

Find the longest subsequence common to two strings

Input: Two strings, S and R

Output: Longest common subsequence of S and R

Recursive LCS

LCS(AGCTTAGCT**G**, TCGGAT**G**)?

Three possibilities:

1. Last letter from both strings part of LCS if identical; none of them otherwise
2. Last letter from left string is part of LCS
3. Last letter from right string is part of LCS

Recursive LCS (2)

Assume you know

LCS(AGCTTAGCT, TCGGAT) = $LCS_{-1,-1}$

LCS(AGCTTAGCT**G**, TCGGAT) = $LCS_{0,-1}$

LCS(AGCTTAGCT, TCGGAT**G**) = $LCS_{-1,0}$

LCS(AGCTTAGCTG, TCGGATG)

= max{$LCS_{0,-1}$, $LCS_{-1,0}$, $LCS_{-1,-1}$ + f(**G**, **G**)}

$$f(r,s) = \begin{cases} 1 & r = s \\ 0 & r \neq s \end{cases}$$

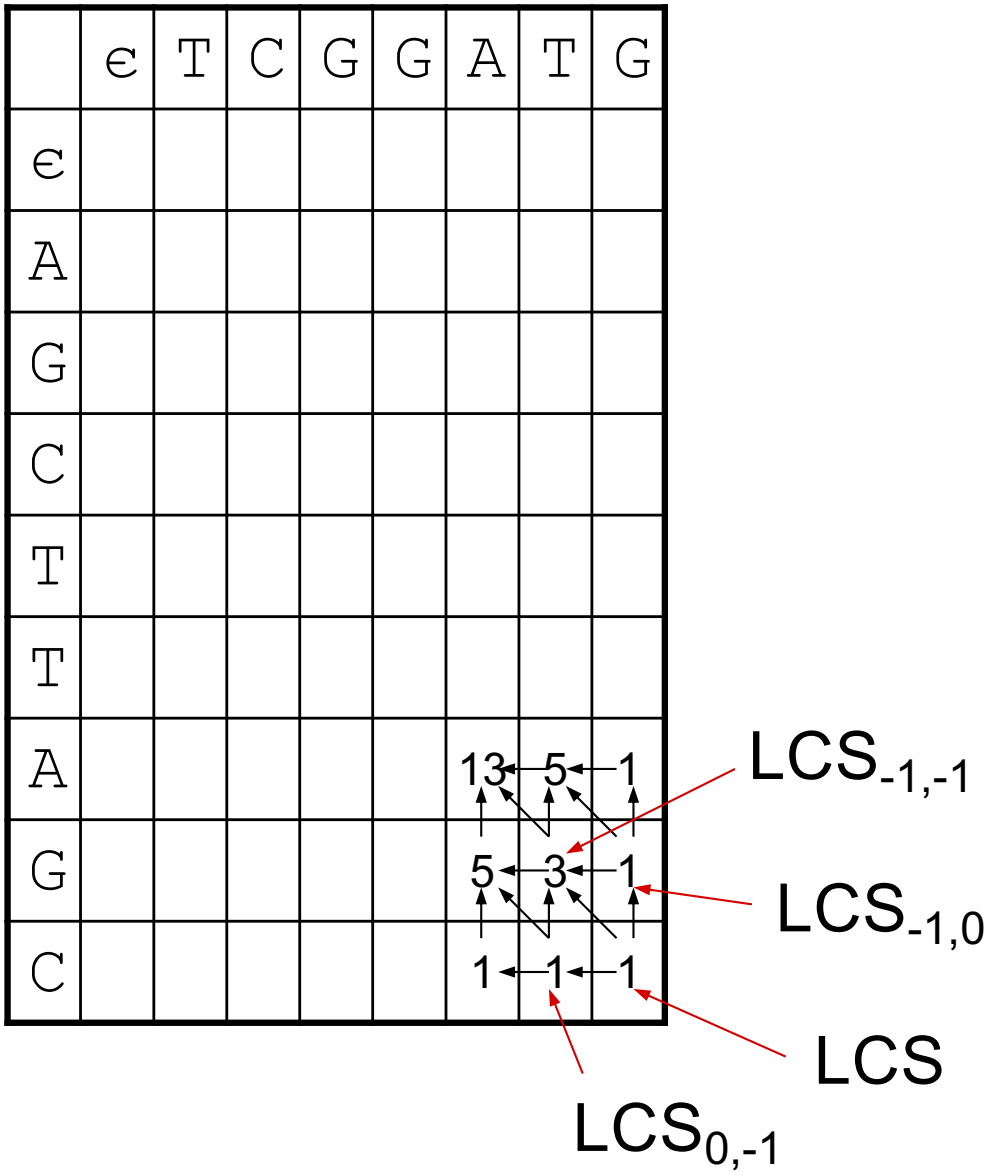
Recursive LCS (3)

- Recursive LCS solves sub-problems multiple times
- Recursion table

LCS(AGCTTAGC, TCGGATG)

= max{$LCS_{0,-1}$, $LCS_{-1,0}$, $LCS_{-1,-1}$ + f(C, G)}

Alternative solution?



Dynamic programming (DP) LCS

- Solve LCS bottom up

$LCS(S_i, R_j)$

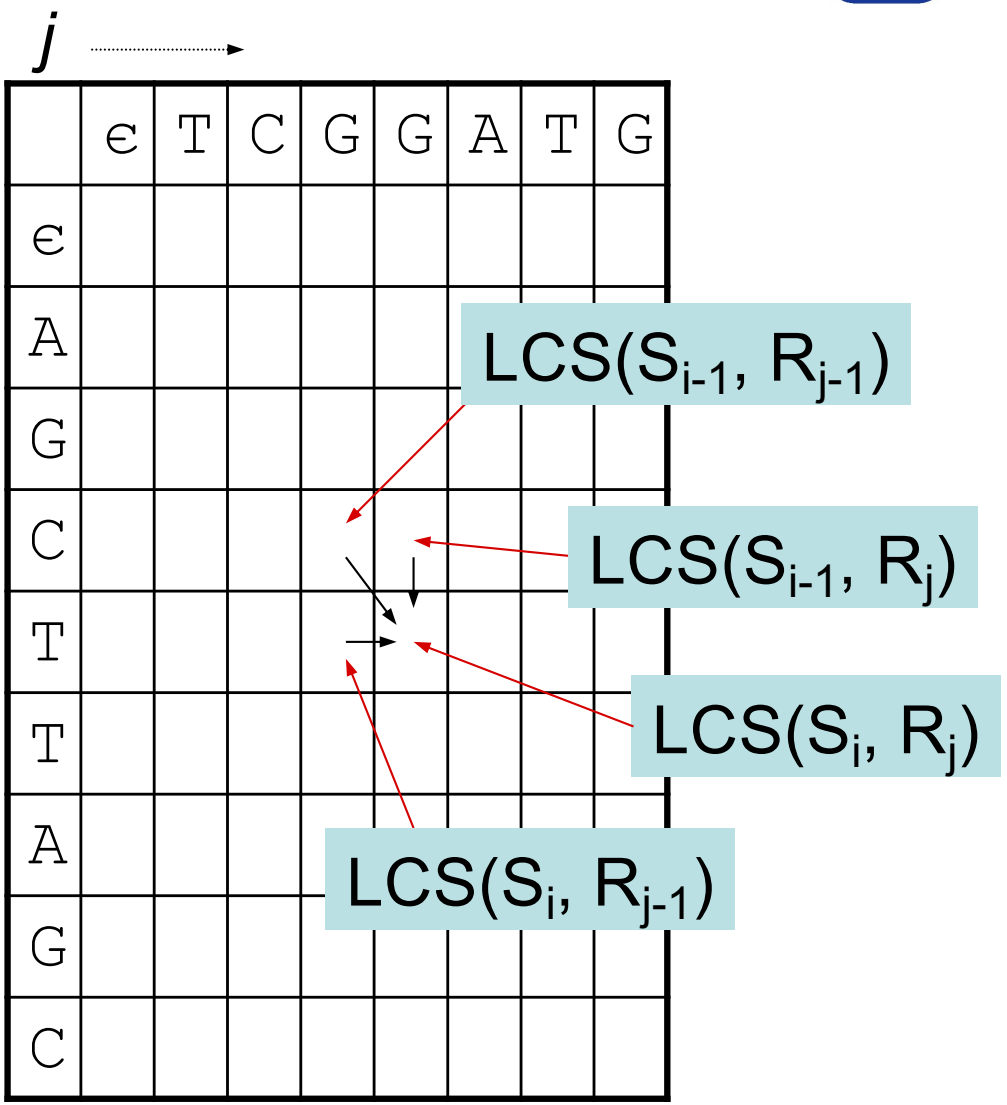
= max{

$LCS(S_{i-1}, R_j)$,

$LCS(S_i, R_{j-1})$,

$LCS(S_{i-1}, R_{j-1}) + f(s_i, r_j)$

}



Dynamic programming LCS (2)

- Solve LCS bottom up

$LCS(S_i, R_j)$

= max{

$LCS(S_{i-1}, R_j)$,

$LCS(S_i, R_{j-1})$,

$LCS(S_{i-1}, R_{j-1}) + f(s_i, r_j)$

}

1. *Initialize matrix?*
2. *Fill matrix using recurrence?*

	ϵ	T	C	G	G	A	T	G
ϵ								
A								
G								
C								
T								
T								
A								
G								
C								

Dynamic programming LCS (2)

- Solve LCS bottom up

$LCS(S_i, R_j)$

= max{

$LCS(S_{i-1}, R_j)$,

$LCS(S_i, R_{j-1})$,

$LCS(S_{i-1}, R_{j-1}) + f(s_i, r_j)$

}

1. Initialize matrix
2. *Fill matrix using recurrence?*

	ϵ	T	C	G	G	A	T	G
ϵ	0	0	0	0	0	0	0	0
A	0							
G	0							
C	0							
T	0							
T	0							
A	0							
G	0							
C	0							

Dynamic programming LCS (3)

- Solve LCS bottom up

$LCS(S_i, R_j)$

= max{

$LCS(S_{i-1}, R_j)$,

$LCS(S_i, R_{j-1})$,

$LCS(S_{i-1}, R_{j-1}) + f(s_i, r_j)$

}

1. Initialize matrix
2. Fill matrix using recurrence

	ϵ	T	C	G	G	A	T	G
ϵ	0	0	0	0	0	0	0	0
A	0	0						
G	0	0						
C	0	0						
T	0	1						
T	0							
A	0							
G	0							
C	0							

Dynamic programming LCS (3)

- Solve LCS bottom up

$LCS(S_i, R_j)$

= max{

$LCS(S_{i-1}, R_j)$,

$LCS(S_i, R_{j-1})$,

$LCS(S_{i-1}, R_{j-1}) + f(s_i, r_j)$

}

1. Initialize matrix
2. Fill matrix using recurrence

	ϵ	T	C	G	G	A	T	G
ϵ	0	0	0	0	0	0	0	0
A	0	0	0					
G	0	0	0					
C	0	0	1					
T	0	1	1					
T	0	1	1					
A	0	1	1					
G	0	1	1					
C	0	1	2					

Dynamic programming LCS (3)

- Solve LCS bottom up

$LCS(S_i, R_j)$

= max{

$LCS(S_{i-1}, R_j)$,

$LCS(S_i, R_{j-1})$,

$LCS(S_{i-1}, R_{j-1}) + f(s_i, r_j)$

}

1. Initialize matrix
2. Fill matrix using recurrence

	ϵ	T	C	G	G	A	T	G
ϵ	0	0	0	0	0	0	0	0
A	0	0	0	0	0	1	1	1
G	0	0	0	1	1	1	1	2
C	0	0	1	1	1	1	1	2
T	0	1	1	1	1	1	2	2
T	0	1	1	1	1	1	2	2
A	0	1	1	1	1	2	2	2
G	0	1	1	2	2	2	2	3
C	0	1	2	2	2	2	2	3

LCS DP algorithm

LCS(S, R):

1. a[0,0] = 0
2. for i in range(1, |S|):
    1. a[i,0] = 0
3. for j in range(1, |R|):
    1. a[0,j] = 0
4. for i in range(1, |S|):
    1. for j in range(1, |R|):
        1. a[i,j] = max{a[i-1,j],a[i,j-1],a[i-1,j-1] + $f(s_i, r_j)$}

But what is the LCS?

$LCS(S_i, R_j)$

= max{

$LCS(S_{i-1}, R_j)$,

$LCS(S_i, R_{j-1})$,

$LCS(S_{i-1}, R_{j-1}) + f(C, G)$

}

- LCS is path through matrix

  TTG

- "Remember where you've been"

	ϵ	T	C	G	G	A	T	G
ϵ	0	0	0	0	0	0	0	0
A	0	0	0	0	0	1	1	1
G	0	0	0	1	1	1	1	2
C	0	0	1	1	1	1	1	2
T	0	1	1	1	1	1	2	2
T	0	1	1	1	1	1	2	2
A	0	1	1	1	1	2	2	2
G	0	1	1	2	2	2	2	3
C	0	1	2	2	2	2	2	3

LCS DP algorithm with backtrack matrix

LCS(S, R):

1. a[0,0] = 0
2. for i in range(1, |S|):
    1. a[i,0] = 0
3. for j in range(1, |R|):
    1. a[0,j] = 0
4. for i in range(1, |S|):
    1. for j in range(1, |R|):
        1. a[i,j] = max{a[i-1,j],a[i,j-1],a[i-1,j-1] + $f(s_i, r_j)$}
        2. b[i,j] = {0 if a[i,j] == a[i-1,j], 1 if a[i,j] == a[i,j-1], 2 if a[i,j] == a[i-1,j-1] + 1}

0 ==
1 ==
2 ==

Does the path fit the algorithm?

4.1.2 b[i,j] = {

0 if a[i,j] == a[i-1,j],

1 if a[i,j] == a[i,j-1],

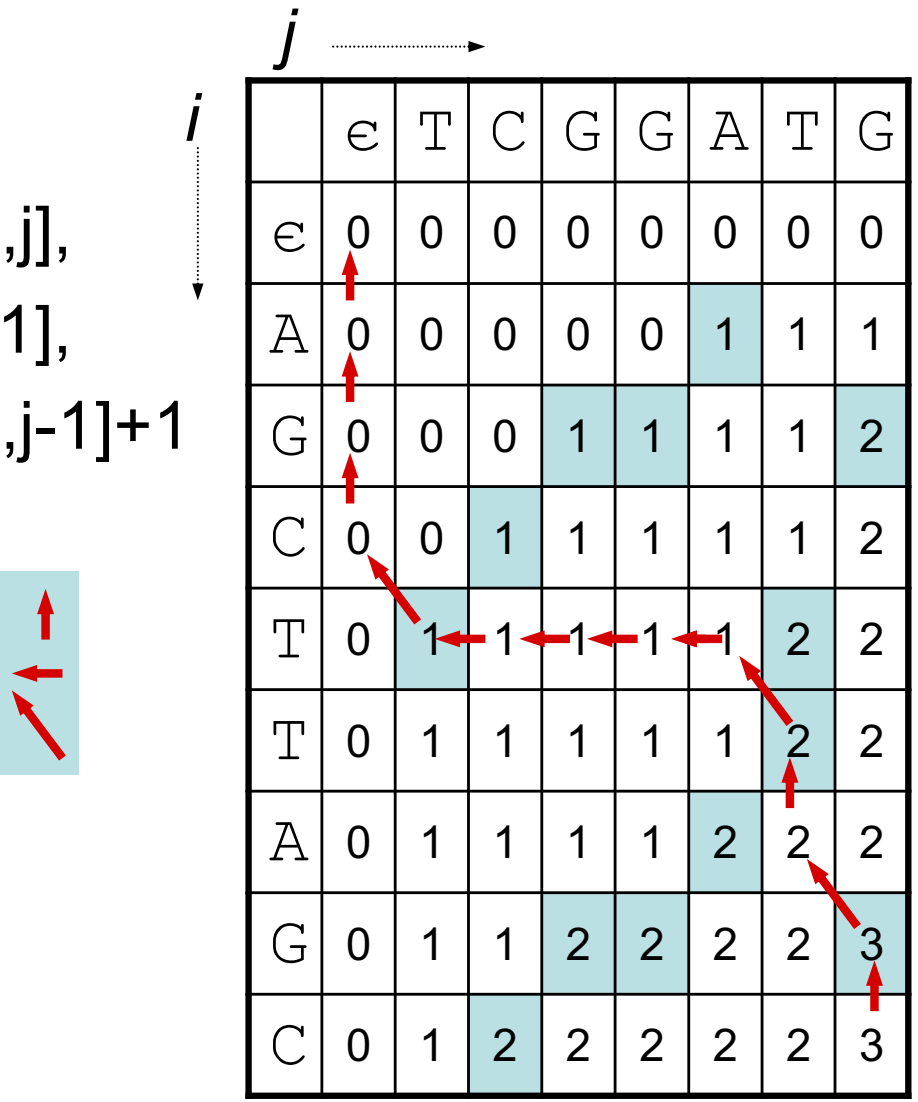
2 if a[i,j] == a[i-1,j-1]+1

}

0 ==

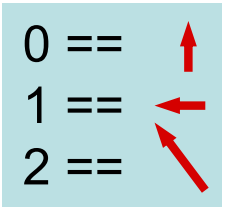
1 ==

2 ==



	ϵ	T	C	G	G	A	T	G
ϵ	0	0	0	0	0	0	0	0
A	0	0	0	0	0	1	1	1
G	0	0	0	1	1	1	1	2
C	0	0	1	1	1	1	1	2
T	0	1	1	1	1	1	2	2
T	0	1	1	1	1	1	2	2
A	0	1	1	1	1	2	2	2
G	0	1	1	2	2	2	2	3
C	0	1	2	2	2	2	2	3

Printing the LCS



PrintLCS(b, S, i, j):

1. if i == 0 or j == 0:
    1. return
2. if b[i,j] == 2:
    1. PrintLCS(b, S, i-1, j-1)
    2. print s_i
3. else if b[i,j] == 0:
    1. PrintLCS(b, S, i-1, j)
4. else:
    1. PrintLCS(b, S, i, j-1)

Homework

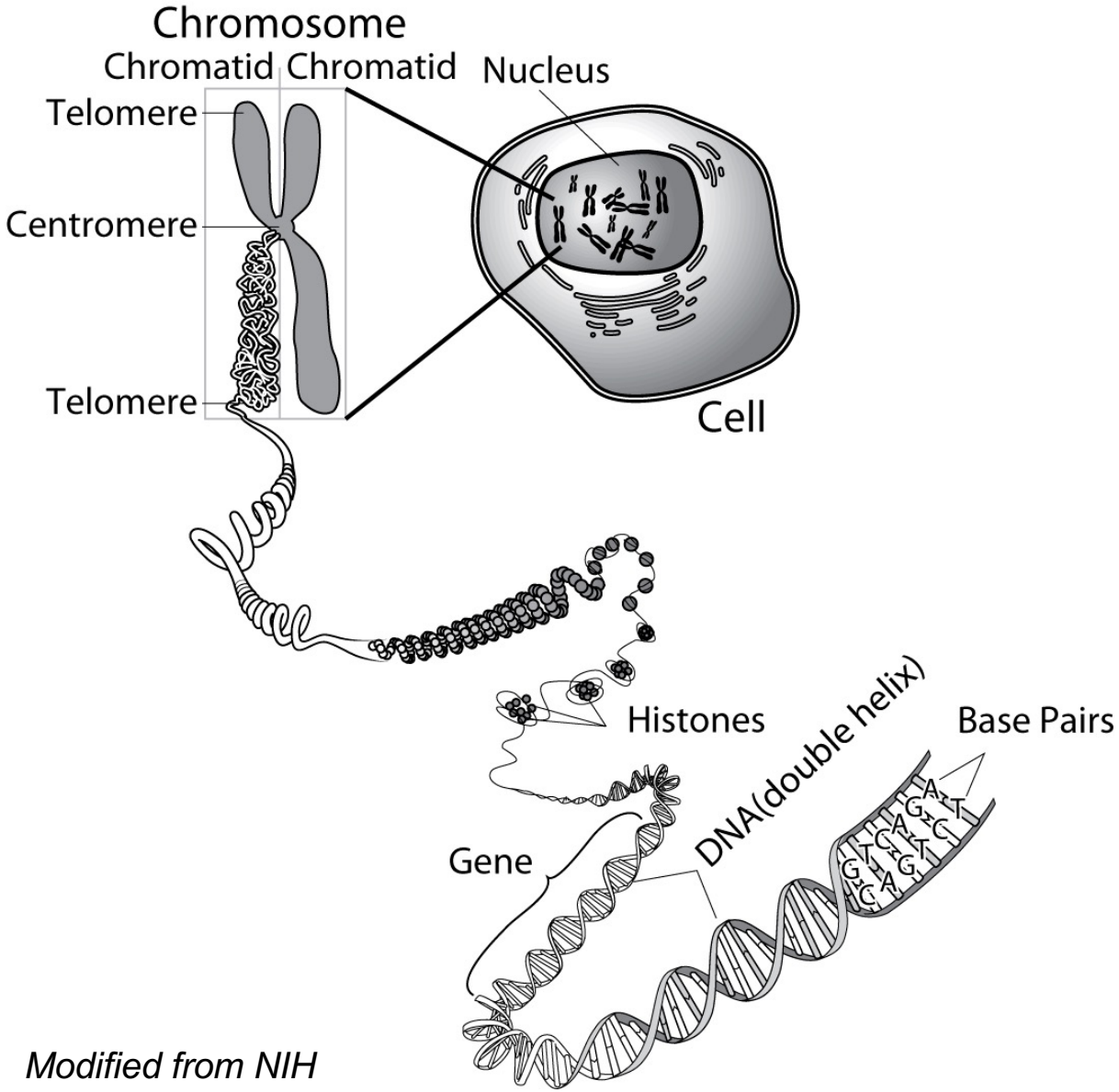
1. Compute the LCS DP matrix for LCS(`ATTCGGTTA`, `TAGTGATG`).

2. Find the LCS without using a backtrack matrix.

3. LCS(`AGCTTAGCTG`, `TCGGATG`) has multiple solutions (for example, `TTG` or `ATG`). Find an algorithm that returns all the LCSs. (Hint: a stack could be useful)

Summary

Sequences – basic data structures in cells



Modified from NIH
http://www.accessexcellence.org/RC/VL/GG/chromosome.php

Sequence data