Bioinformatics

CS300

Chap 2

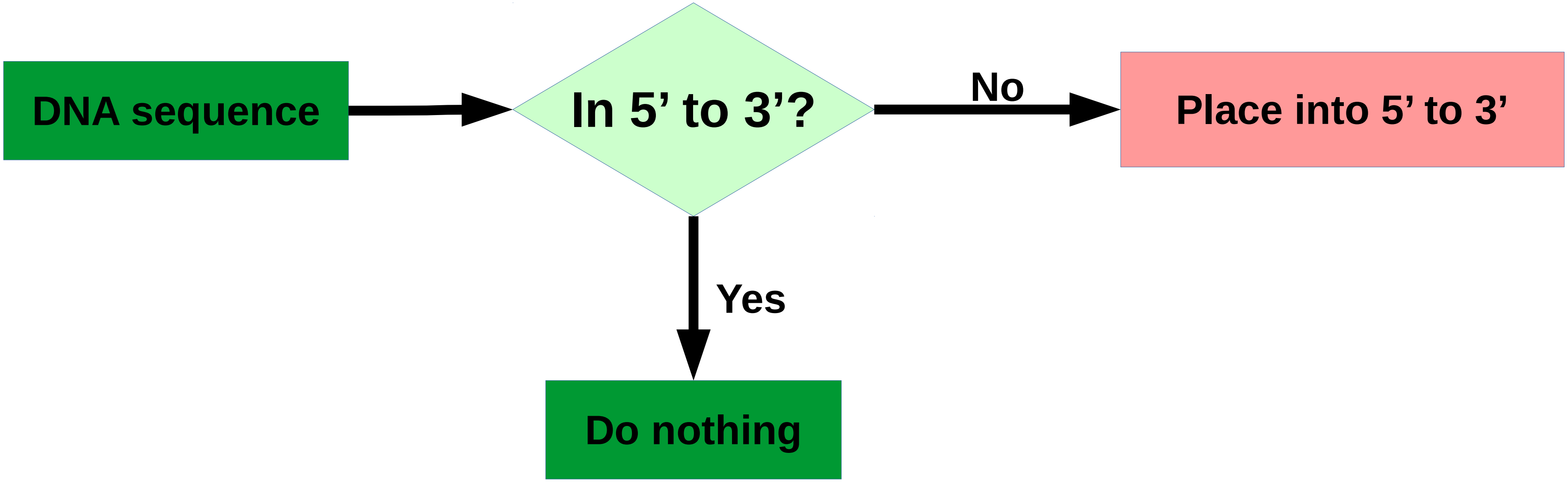
Computational Manipulation of DNA

Fall 2019

Oliver BONHAM-CARTER

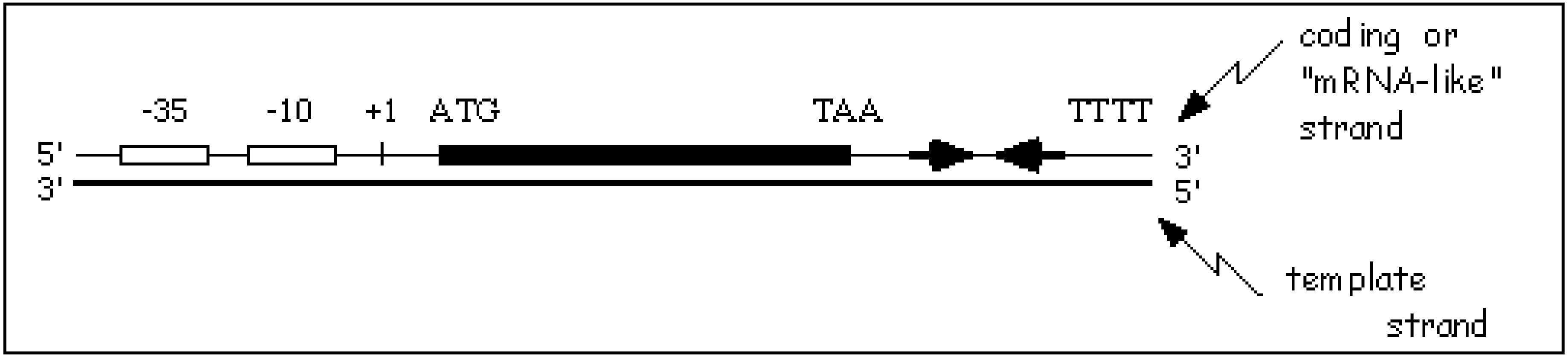
The Unnamed Sequence

- Unlabeled strands of DNA are assumed to be in the 5’ to 3’, (left to right) direction.
- A new sequence is given to us for analysis.
- What are the steps to place this sequence into a format for use with bioinformatics tools?

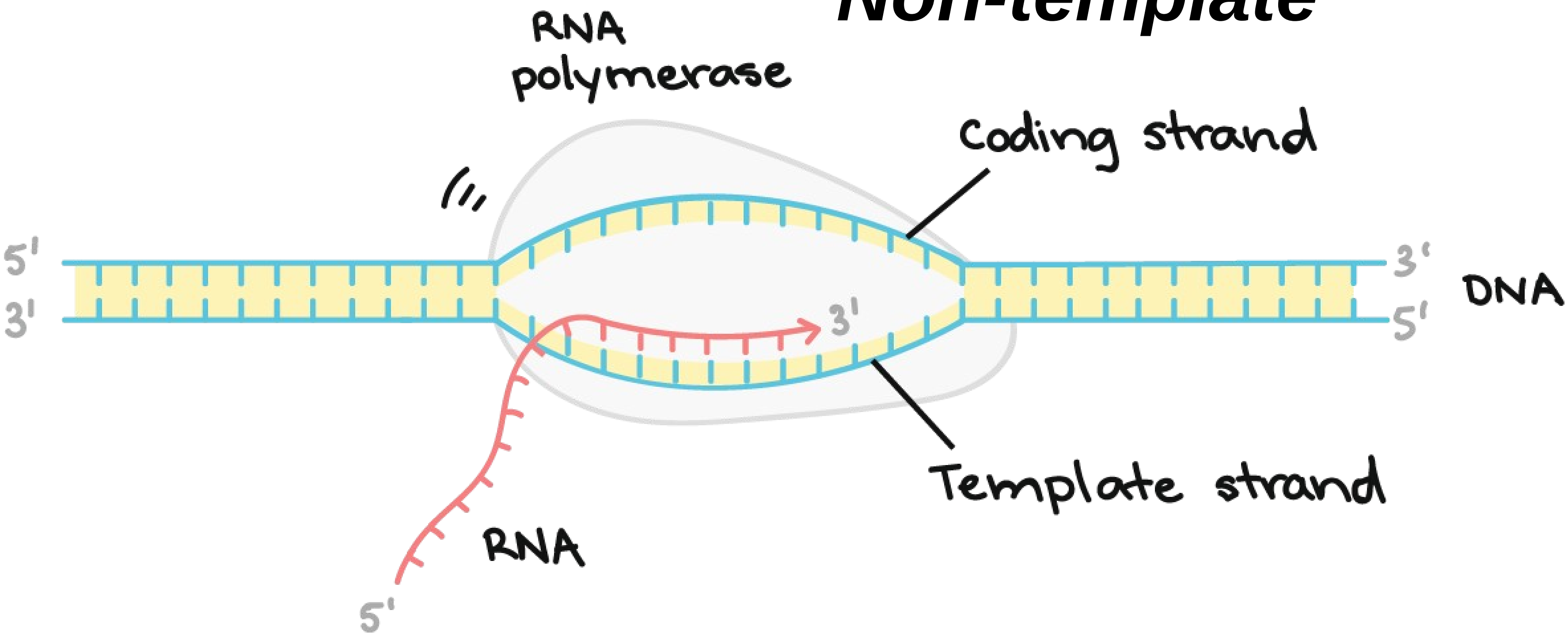


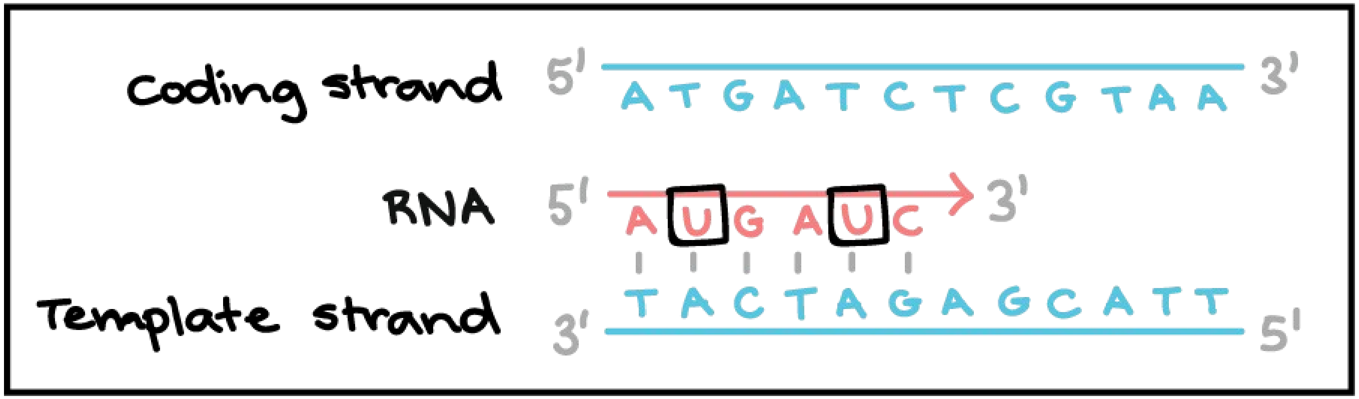
Template vs nonTemplate

- Input:
  - DNA sequence: AGCAT
  - Strand: **template** (used to make mRNA) or **non-template** (the compliment of this strand that looks like mRNA)
  - Orientation: 3’ -> 5’ or 5’ -> 3’
- Output:
  - Template strand in 3’ -> 5’ orientation ready for transcription



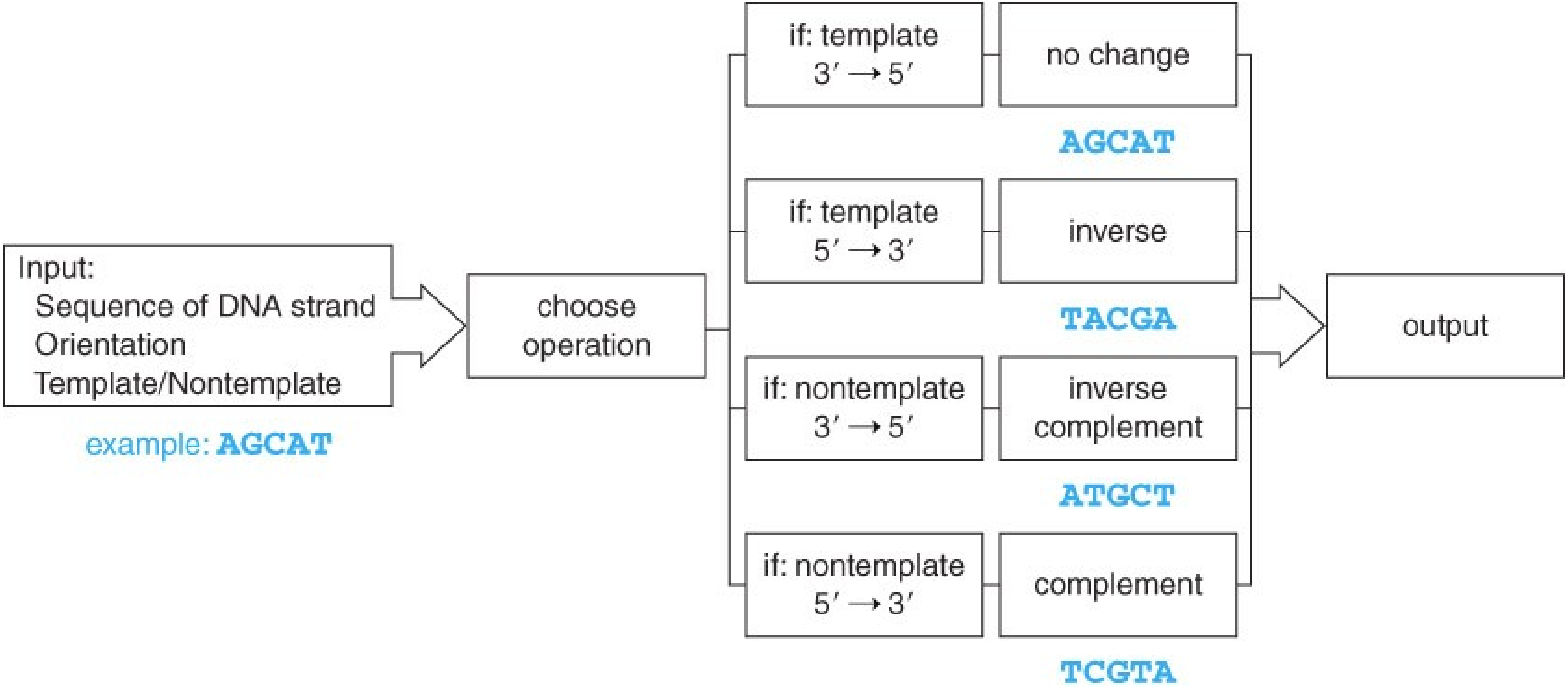
Non-template




Coding strand 5' A T G A T C T C G T A A 3'
RNA 5' A U G A U C 3'
Template strand 3' T A C T A G A G C A T T 5'

DNA Manipulation Algorithm

- A series of steps when handling DNA



Output DNA in 3’ to 5’

The DNA Manipulation Algorithm

1. Input a DNA sequence, including details of being a template or non-template strand as well as its orientation
2. Convert to all uppercase
3. Choose the appropriate operation:
    1. If it is the **template** strand and oriented **3’ -> 5’**, simply output the same sequence
    2. If it is the **template** strand and oriented **5’ -> 3’**, **inverse** the sequence (traverse the string from right to left and add each character to output the string)
    3. If it is the **non-template** strand and oriented **3’ -> 5’**, generate the **inverse complement** sequence ((i.) traverse the string from right to left and (ii) for each character, add the complement to the output string)
    4. If it is the **non-template** strand and oriented **5’ -> 3’**, generate the **complement** ((i.) traverse the string from left to right and (ii) for each character add the complement to the output string)
4. Output the completed sequence, including 5’ and 3’ end labels

Transcription Algorithm

- Input: **template** strand in the **3’** → **5’** orientation
- Output: mRNA strand in the 5’ → 3’ orientation
  - Traverse the string from left to right
    - add complementary base to the output string
    - (note T is now U)

Alternative Transcription Algorithm

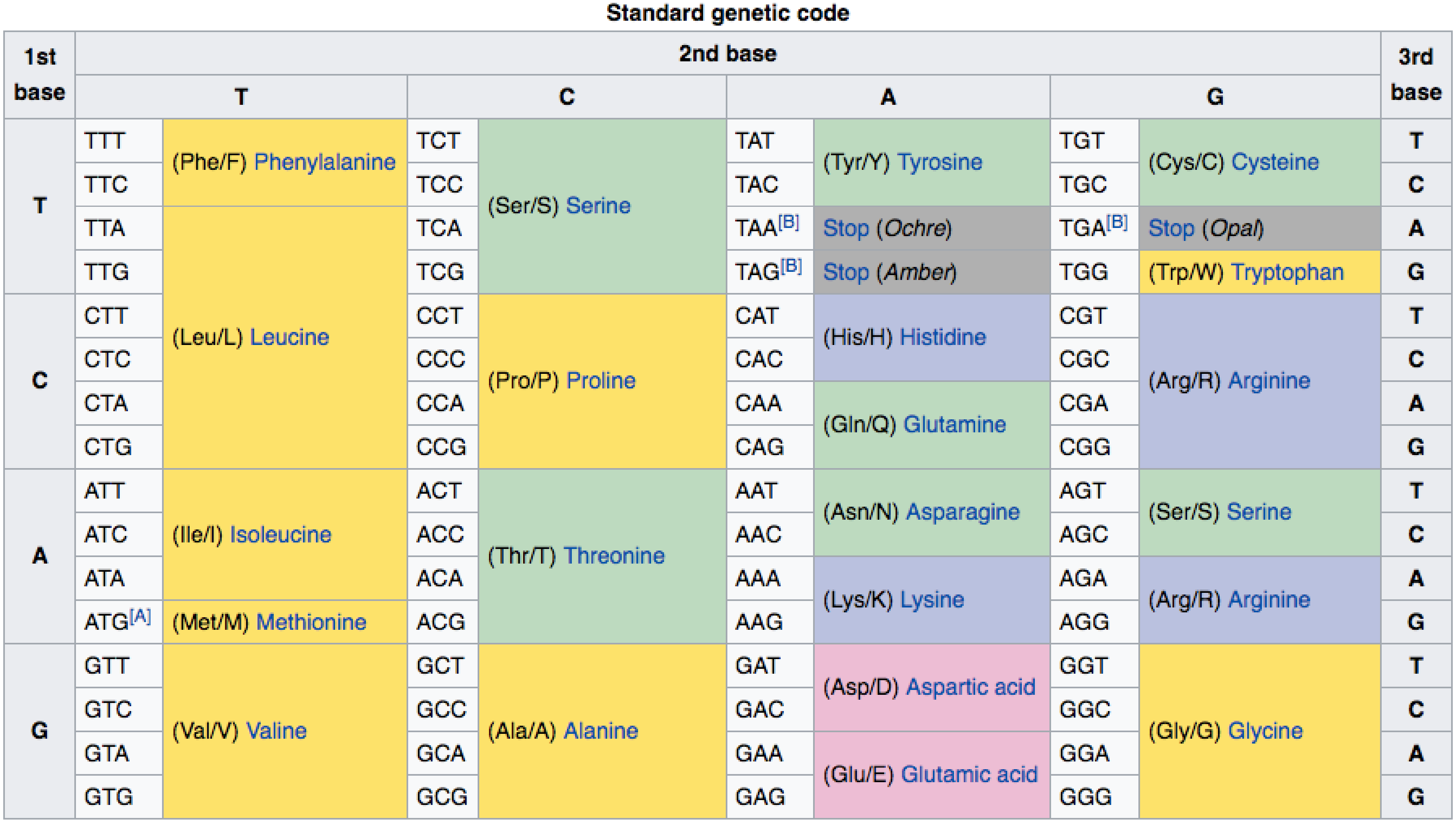
- Input: **non-template** strand in the **5’** → **3’** orientation
- Output: mRNA strand in the **5’** → **3’** orientation
  - Traverse the string from left to right
    - Replace all the T’s with U’s

Translation Algorithm

- Input: mRNA strand in the **5’** → **3’** orientation
- Output: amino acid sequence
  - Traverse the string looking at one codon at a time
  - Add one amino acid corresponding to the protein sequence.

Codon Table to Translate the Protein Product

- DNA triplets read in groups of three called codons and represent an amino acid



Standard genetic code

1st base	2nd base								3rd base
	T		C		A		G		
T	TTT	(Phe/F) Phenylalanine	TCT	(Ser/S) Serine	TAT	(Tyr/Y) Tyrosine	TGT	(Cys/C) Cysteine	T
	TTC		TCC		TAC		TGC		C
	TTA	(Leu/L) Leucine	TCA		TAA[B]	Stop (*Ochre*)	TGA[B]	Stop (*Opal*)	A
	TTG		TCG		TAG[B]	Stop (*Amber*)	TGG	(Trp/W) Tryptophan	G
C	CTT		CCT	(Pro/P) Proline	CAT	(His/H) Histidine	CGT	(Arg/R) Arginine	T
	CTC		CCC		CAC		CGC		C
	CTA		CCA		CAA	(Gln/Q) Glutamine	CGA		A
	CTG		CCG		CAG		CGG		G
A	ATT	(Ile/I) Isoleucine	ACT	(Thr/T) Threonine	AAT	(Asn/N) Asparagine	AGT	(Ser/S) Serine	T
	ATC		ACC		AAC		AGC		C
	ATA		ACA		AAA	(Lys/K) Lysine	AGA	(Arg/R) Arginine	A
	ATG[A]	(Met/M) Methionine	ACG		AAG		AGG		G
G	GTT	(Val/V) Valine	GCT	(Ala/A) Alanine	GAT	(Asp/D) Aspartic acid	GGT	(Gly/G) Glycine	T
	GTC		GCC		GAC		GGC		C
	GTA		GCA		GAA	(Glu/E) Glutamic acid	GGA		A
	GTG		GCG		GAG		GGG		G

Python Programming

- Biopython
- DNA Manipulation Algorithm
- Translation functions
  - DNA → RNA
  - RNA → DNA
  - RNA → Protein

Follow along in class and save your notes in a text file!!