

Sent by you:

What are the 20 common amino acids?

The 20 common amino acids, which are the building blocks of proteins, include:

1. **Alanine (Ala)**
2. **Arginine (Arg)**
3. **Asparagine (Asn)**
4. **Aspartic acid (Asp)**
5. **Cysteine (Cys)**
6. **Glutamic acid (Glu)**
7. **Glutamine (Gln)**
8. **Glycine (Gly)**
9. **Histidine (His)**
10. **Isoleucine (Ile)**
11. **Leucine (Leu)**
12. **Lysine (Lys)**
13. **Methionine (Met)**
14. **Phenylalanine (Phe)**
15. **Proline (Pro)**
16. **Serine (Ser)**
17. **Threonine (Thr)**
18. **Tryptophan (Trp)**
19. **Tyrosine (Tyr)**
20. **Valine (Val)**

These amino acids play crucial roles in various biological processes, including protein synthesis, enzyme function, and cell signaling. Do you have a specific interest in amino acids, like their functions or how they interact in the body?

Sent by you:

I am interested in protein translation. what are codons and their relationship to the 20 common amino acids?

Great question! Protein translation is a fascinating process. Here's a breakdown:

Codons

- **Codons** are sequences of three nucleotides (triplets) in messenger RNA (mRNA) that correspond to specific amino acids or stop signals during protein synthesis.
- Each codon is made up of three of the four nucleotides: adenine (A), cytosine (C), guanine (G), and uracil (U).

Relationship to Amino Acids

- There are 64 possible codons (4^3 combinations), but only 20 amino acids. This means that multiple codons can code for the same amino acid, a feature known as **redundancy** or **degeneracy** of the genetic code.
- For example:
 - The codons **UUU** and **UUC** both code for **Phenylalanine (Phe)**.
 - The codons **AUG** codes for **Methionine (Met)** and also serves as the **start codon** to initiate translation.
 - **UAA**, **UAG**, and **UGA** are **stop codons** that signal the end of protein synthesis.

Translation Process

1. **Initiation:** The ribosome assembles around the start codon (AUG) on the mRNA.
2. **Elongation:** Transfer RNA (tRNA) molecules bring amino acids to the ribosome, matching their anticodon with the mRNA codon.
3. **Termination:** When a stop codon is reached, the ribosome releases the newly synthesized protein.

Would you like to dive deeper into any specific part of this process?

Sent by you:

Can you list all 64 codons and their corresponding amino acids?

Sure! Here is a table listing all 64 codons and their corresponding amino acids:

Codon	Amino Acid	Codon	Amino Acid
UUU	Phenylalanine (Phe)	CUU	Leucine (Leu)
UUC	Phenylalanine (Phe)	CUC	Leucine (Leu)
UUA	Leucine (Leu)	CUA	Leucine (Leu)
UUG	Leucine (Leu)	CUG	Leucine (Leu)
UCU	Serine (Ser)	CCU	Proline (Pro)
UCC	Serine (Ser)	CCC	Proline (Pro)
UCA	Serine (Ser)	CCA	Proline (Pro)
UCG	Serine (Ser)	CCG	Proline (Pro)
UAU	Tyrosine (Tyr)	CAU	Histidine (His)
UAC	Tyrosine (Tyr)	CAC	Histidine (His)
UAA	Stop	CAA	Glutamine (Gln)
UAG	Stop	CAG	Glutamine (Gln)
UGU	Cysteine (Cys)	CGU	Arginine (Arg)
UGC	Cysteine (Cys)	CGC	Arginine (Arg)

Codon	Amino Acid	Codon	Amino Acid
UGA	Stop	CGA	Arginine (Arg)
UGG	Tryptophan (Trp)	CGG	Arginine (Arg)
CUU	Leucine (Leu)	AUU	Isoleucine (Ile)
CUC	Leucine (Leu)	AUC	Isoleucine (Ile)
CUA	Leucine (Leu)	AUA	Isoleucine (Ile)
CUG	Leucine (Leu)	AUG	Methionine (Met)
CCU	Proline (Pro)	ACU	Threonine (Thr)
CCC	Proline (Pro)	ACC	Threonine (Thr)
CCA	Proline (Pro)	ACA	Threonine (Thr)
CCG	Proline (Pro)	ACG	Threonine (Thr)
CAU	Histidine (His)	AAU	Asparagine (Asn)
CAC	Histidine (His)	AAC	Asparagine (Asn)
CAA	Glutamine (Gln)	AAA	Lysine (Lys)
CAG	Glutamine (Gln)	AAG	Lysine (Lys)
CGU	Arginine (Arg)	AGU	Serine (Ser)
CGC	Arginine (Arg)	AGC	Serine (Ser)
CGA	Arginine (Arg)	AGA	Arginine (Arg)

Codon	Amino Acid	Codon	Amino Acid
CGG	Arginine (Arg)	AGG	Arginine (Arg)
AUU	Isoleucine (Ile)	GUU	Valine (Val)
AUC	Isoleucine (Ile)	GUC	Valine (Val)
AUA	Isoleucine (Ile)	GUA	Valine (Val)
AUG	Methionine (Met)	GUG	Valine (Val)
ACU	Threonine (Thr)	GCU	Alanine (Ala)
ACC	Threonine (Thr)	GCC	Alanine (Ala)
ACA	Threonine (Thr)	GCA	Alanine (Ala)
ACG	Threonine (Thr)	GCG	Alanine (Ala)
AAU	Asparagine (Asn)	GAU	Aspartic acid (Asp)
AAC	Asparagine (Asn)	GAC	Aspartic acid (Asp)
AAA	Lysine (Lys)	GAA	Glutamic acid (Glu)
AAG	Lysine (Lys)	GAG	Glutamic acid (Glu)
AGU	Serine (Ser)	GGU	Glycine (Gly)
AGC	Serine (Ser)	GGC	Glycine (Gly)
AGA	Arginine (Arg)	GGA	Glycine (Gly)
AGG	Arginine (Arg)	GGG	Glycine (Gly)

This table shows how each codon specifies a particular amino acid or a stop signal during protein synthesis. If you have any more questions about protein translation or anything else, feel free to ask!

Sent by you:

This is great, thanks. Can you simplify the table to only include the codons and the matching one-letter code for each amino acid?

Absolutely! Here's a simplified table with the codons and their corresponding one-letter amino acid codes:

Codon	Amino Acid
UUU	F
UUC	F
UUA	L
UUG	L
UCU	S
UCC	S
UCA	S
UCG	S
UAU	Y
UAC	Y
UAA	Stop
UAG	Stop
UGU	C
UGC	C
UGA	Stop
UGG	W

Codon	Amino Acid
CUU	L
CUC	L
CUA	L
CUG	L
CCU	P
CCC	P
CCA	P
CCG	P
CAU	H
CAC	H
CAA	Q
CAG	Q
CGU	R
CGC	R
CGA	R
CGG	R
AUU	I
AUC	I
AUA	I

Codon	Amino Acid
AUG	M
ACU	T
ACC	T
ACA	T
ACG	T
AAU	N
AAC	N
AAA	K
AAG	K
AGU	S
AGC	S
AGA	R
AGG	R
GUU	V
GUC	V
GUA	V
GUG	V
GCU	A
GCC	A

Codon	Amino Acid
GCA	A
GCG	A
GAU	D
GAC	D
GAA	E
GAG	E
GGU	G
GGC	G
GGA	G
GGG	G

This should make it easier to see the relationship between codons and their corresponding amino acids. If you have any more questions or need further clarification, feel free to ask!

Sent by you:

Please change "Stop" to periods

Sure! Here’s the updated table with “Stop” codons replaced by periods:

Codon	Amino Acid
UUU	F
UUC	F
UUA	L
UUG	L
UCU	S
UCC	S
UCA	S
UCG	S
UAU	Y
UAC	Y
UAA	.
UAG	.
UGU	C
UGC	C
UGA	.
UGG	W

Codon	Amino Acid
CUU	L
CUC	L
CUA	L
CUG	L
CCU	P
CCC	P
CCA	P
CCG	P
CAU	H
CAC	H
CAA	Q
CAG	Q
CGU	R
CGC	R
CGA	R
CGG	R
AUU	I
AUC	I
AUA	I

Codon	Amino Acid
AUG	M
ACU	T
ACC	T
ACA	T
ACG	T
AAU	N
AAC	N
AAA	K
AAG	K
AGU	S
AGC	S
AGA	R
AGG	R
GUU	V
GUC	V
GUA	V
GUG	V
GCU	A
GCC	A

Codon	Amino Acid
GCA	A
GCG	A
GAU	D
GAC	D
GAA	E
GAG	E
GGU	G
GGC	G
GGA	G
GGG	G

I hope this helps! If you have any more questions or need further assistance, feel free to ask.

Sent by you:

Awesome. In the codons, can you also replace the U with T?

Sure! Here's the updated table with "U" replaced by "T" in the codons:

Codon	Amino Acid
TTT	F
TTC	F
TTA	L
TTG	L
TCT	S
TCC	S
TCA	S
TCG	S
TAT	Y
TAC	Y
TAA	.
TAG	.
TGT	C
TGC	C
TGA	.

Codon	Amino Acid
TGG	W
CTT	L
CTC	L
CTA	L
CTG	L
CCT	P
CCC	P
CCA	P
CCG	P
CAT	H
CAC	H
CAA	Q
CAG	Q
CGT	R
CGC	R
CGA	R
CGG	R
ATT	I
ATC	I

Codon	Amino Acid
ATA	I
ATG	M
ACT	T
ACC	T
ACA	T
ACG	T
AAT	N
AAC	N
AAA	K
AAG	K
AGT	S
AGC	S
AGA	R
AGG	R
GTT	V
GTC	V
GTA	V
GTG	V
GCT	A

Codon	Amino Acid
GCC	A
GCA	A
GCG	A
GAT	D
GAC	D
GAA	E
GAG	E
GGT	G
GGC	G
GGA	G
GGG	G

I hope this helps! If you have any more questions or need further assistance, feel free to ask.

Sent by you:

Now I'd like to turn this table into a Python dictionary. Please re-write the table as a dictionary named `codon_to_aa` where each key is a codon and each value is an amino acid (or period if its a stop codon)

Sure! Here's the table rewritten as a Python dictionary named `codon_to_aa`:

```
codon_to_aa = {
    "TTT": "F", "TTC": "F",
    "TTA": "L", "TTG": "L",
    "TCT": "S", "TCC": "S",
    "TCA": "S", "TCG": "S",
    "TAT": "Y", "TAC": "Y",
    "TAA": ".", "TAG": ".",
    "TGT": "C", "TGC": "C",
    "TGA": ".", "TGG": "W",
    "CTT": "L", "CTC": "L",
    "CTA": "L", "CTG": "L",
    "CCT": "P", "CCC": "P",
    "CCA": "P", "CCG": "P",
    "CAT": "H", "CAC": "H",
    "CAA": "Q", "CAG": "Q",
    "CGT": "R", "CGC": "R",
    "CGA": "R", "CGG": "R",
    "ATT": "I", "ATC": "I",
    "ATA": "I", "ATG": "M",
    "ACT": "T", "ACC": "T",
    "ACA": "T", "ACG": "T",
    "AAT": "N", "AAC": "N",
    "AAA": "K", "AAG": "K",
    "AGT": "S", "AGC": "S",
    "AGA": "R", "AGG": "R",
    "GTT": "V", "GTC": "V",
    "GTA": "V", "GTG": "V",
    "GCT": "A", "GCC": "A",
    "GCA": "A", "GCG": "A",
    "GAT": "D", "GAC": "D",
    "GAA": "E", "GAG": "E",
    "GGT": "G", "GGC": "G",
    "GGA": "G", "GGG": "G"
}
```

You can use this dictionary to look up the amino acid corresponding to any given codon. If you need any further assistance or have more questions, feel free to ask!

Sent by you:

Cool. Now I want to roll this data around a bit. Please create a new dictionary, where each amino acid is a key and the value is a list of the codons that can be used to represent it (so the values should have different lengths - for example, 'M' should have value ['ATG'] while 'R' will have 6 values)

Sure! Here's a new dictionary where each amino acid is a key and the value is a list of the codons that can represent it:

```
aa_to_codons = {
    "F": ["TTT", "TTC"],
    "L": ["TTA", "TTG", "CTT", "CTC", "CTA", "CTG"],
    "S": ["TCT", "TCC", "TCA", "TCG", "AGT", "AGC"],
    "Y": ["TAT", "TAC"],
    ".": ["TAA", "TAG", "TGA"],
    "C": ["TGT", "TGC"],
    "W": ["TGG"],
    "P": ["CCT", "CCC", "CCA", "CCG"],
    "H": ["CAT", "CAC"],
    "Q": ["CAA", "CAG"],
    "R": ["CGT", "CGC", "CGA", "CGG", "AGA", "AGG"],
    "I": ["ATT", "ATC", "ATA"],
    "M": ["ATG"],
    "T": ["ACT", "ACC", "ACA", "ACG"],
    "N": ["AAT", "AAC"],
    "K": ["AAA", "AAG"],
    "V": ["GTT", "GTC", "GTA", "GTG"],
    "A": ["GCT", "GCC", "GCA", "GCG"],
    "D": ["GAT", "GAC"],
    "E": ["GAA", "GAG"],
    "G": ["GGT", "GGC", "GGA", "GGG"]
}
```

This dictionary allows you to see all the codons that correspond to each amino acid. If you have any more questions or need further assistance, feel free to ask!

Sent by you:

Great, thanks. Please write a function that, provided a string of amino acids, will return a DNA sequence that would code for said protein, but each codon is randomly drawn from the list returned by the aa_to_codons dictionary

Sure! Here's a Python function that takes a string of amino acids and returns a DNA sequence where each codon is randomly selected from the possible codons for each amino acid:

```
import random

# Dictionary mapping amino acids to their possible codons
aa_to_codons = {
    "F": ["TTT", "TTC"],
    "L": ["TTA", "TTG", "CTT", "CTC", "CTA", "CTG"],
    "S": ["TCT", "TCC", "TCA", "TCG", "AGT", "AGC"],
    "Y": ["TAT", "TAC"],
    ".": ["TAA", "TAG", "TGA"],
    "C": ["TGT", "TGC"],
    "W": ["TGG"],
    "P": ["CCT", "CCC", "CCA", "CCG"],
    "H": ["CAT", "CAC"],
    "Q": ["CAA", "CAG"],
    "R": ["CGT", "CGC", "CGA", "CGG", "AGA", "AGG"],
    "I": ["ATT", "ATC", "ATA"],
    "M": ["ATG"],
    "T": ["ACT", "ACC", "ACA", "ACG"],
    "N": ["AAT", "AAC"],
    "K": ["AAA", "AAG"],
    "V": ["GTT", "GTC", "GTA", "GTG"],
    "A": ["GCT", "GCC", "GCA", "GCG"],
    "D": ["GAT", "GAC"],
    "E": ["GAA", "GAG"],
    "G": ["GGT", "GGC", "GGA", "GGG"]
}

def generate_dna_sequence(amino_acids):
    dna_sequence = ""
    for aa in amino_acids:
        if aa in aa_to_codons:
            codon = random.choice(aa_to_codons[aa])
            dna_sequence += codon
        else:
            raise ValueError(f"Invalid amino acid: {aa}")
    return dna_sequence

# Example usage
protein = "MST"
dna_sequence = generate_dna_sequence(protein)
print(dna_sequence)
```

This function iterates over each amino acid in the input string, randomly selects a corresponding codon from the aa_to_codons dictionary, and concatenates it to form the DNA sequence. If an invalid amino acid is encountered, it raises a ValueError.

You can test this function with different protein sequences to see how it generates the corresponding DNA sequences. If you have any more questions or need further assistance, feel free to ask!

Demonstration of the code in a Jupyter notebook available at

[https://github.com/dbwilburn/BIOC5721/blob/main/Reverse Translation Example AU24.ipynb](https://github.com/dbwilburn/BIOC5721/blob/main/Reverse%20Translation%20Example%20AU24.ipynb)



Reverse Translation Example AU24 ☆

File Edit View Insert Runtime Tools Help

+ Code + Text

Reverse Translation Example

by DBW 8/28/24

This notebook includes test code that was generated using Microsoft CoPilot to randomly generate a DNA sequence that codes for a provided protein sequence (only works for the 20 common amino acids).

```
[1] import random

# Dictionary mapping amino acids to their possible codons
aa_to_codons = {
    "F": ["TTT", "TTC"],
    "L": ["TTA", "TTG", "CTT", "CTC", "CTA", "CTG"],
    "S": ["TCT", "TCC", "TCA", "TCG", "AGT", "AGC"],
    "Y": ["TAT", "TAC"],
    ".": ["TAA", "TAG", "TGA"],
    "C": ["TGT", "TGC"],
    "W": ["TGG"],
    "P": ["CCT", "CCC", "CCA", "CCG"],
    "H": ["CAT", "CAC"],
    "Q": ["CAA", "CAG"],
    "R": ["CGT", "CGC", "CGA", "CGG", "AGA", "AGG"],
    "I": ["ATT", "ATC", "ATA"],
    "M": ["ATG"],
    "T": ["ACT", "ACC", "ACA", "ACG"],
    "N": ["AAT", "AAC"],
    "K": ["AAA", "AAG"],
    "V": ["GTT", "GTC", "GTA", "GTG"],
    "A": ["GCT", "GCC", "GCA", "GCG"],
    "D": ["GAT", "GAC"],
    "E": ["GAA", "GAG"],
    "G": ["GGT", "GGC", "GGA", "GGG"]
}

def generate_dna_sequence(amino_acids):
    dna_sequence = ""
    for aa in amino_acids:
        if aa in aa_to_codons:
            codon = random.choice(aa_to_codons[aa])
            dna_sequence += codon
        else:
            raise ValueError(f"Invalid amino acid: {aa}")
    return dna_sequence

# Example usage
protein = "MST"
dna_sequence = generate_dna_sequence(protein)
print(dna_sequence)
```

ATGTCCACA

```
[2] new_protein = 'MY.NAME.IS.DAMIEN.'
new_dna = generate_dna_sequence(new_protein)
print(new_dna)
```

ATGTACTAAAACGCTATGGAGTAAATTAGCTAAGATGCGATGATTGAAAATTAA

Verify that the DNA sequence translates to the expected protein using a verified tool such as ExPASy (<https://web.expasy.org/translate/>):

ExPASy

Translate

Home

Programmatic Access

Contact

Translate tool

Translate is a tool which allows the translation of a nucleotide (DNA/RNA) sequence to a protein sequence.

DNA or RNA sequence

```
atgtactaaaacgctatggagtaaattagctaagatgcgatgattgaaa  
attaa
```

Output format
☐ Verbose: Met, Stop, spaces between residues
☒ Compact: M, -, no spaces
☐ Includes nucleotide sequence
☐ Includes nucleotide sequence, no spaces

DNA strands
☒ forward ☒ reverse

Genetic codes - [See NCBI's genetic codes](#)

Standard

reset **TRANSLATE!**

Results of translation

- Open reading frames are highlighted in red
- Select your initiator on one of the following frames to retrieve your amino acid sequence

Download all the translated frames

5'3' Frame 1

MY-NAME-IS-DAMIEN-

Looks good :D

Application Used: Microsoft CoPilot

Intended Purpose: Develop a reverse translation tool to generate random DNA sequences that code for specific proteins (or fun words/sentences).

Quality of Initial GenAI Output: I intentionally asked the AI to generally describe what it understood about amino acids, proteins, codons, etc. before asking any programming-specific questions. The initial outputs describing these biophysical principles were quite good.

Iteration and Refinement: Further questioning focused on guiding the AI to produce a function and dictionary structure in Python to produce a function that would behave as intended

Incorporation in Completed Assignment: The results of the conversation were included, and testing the function in CoLab with validation of results using a separate translation tool (ExPASy) confirm the generated code is working as intended.