

## Task 1 & 2

Data Sequence 1 was taken from [Click Here](#) or

[https://www.ncbi.nlm.nih.gov/nucore/NM\\_001302688.2?report=fasta](https://www.ncbi.nlm.nih.gov/nucore/NM_001302688.2?report=fasta)

Nucleotide

Advanced

FASTA

Send to: Change region shown

Customize view

Analyze this sequence

Run BLAST

Pick Primers

Show in Genome Data Viewer

Articles about the APOE gene

Interaction analysis of ancestry-enriched variants with APOE-ε4 on MCI in the Stud [Sci Rep. 2023]

Association of APOE (rs429358 and rs7412) and PON1 (Q192R and L55M) [Genes (Basel). 2023]

[Apolipoprotein E enhances migration of endometrial [Nan Fang Yi Ke Da Xue Xue Bao....]

See all...

Reference sequence information

RefSeq alternative splicing

See 5 reference mRNA sequence splice variants for the APOE gene.

Data Sequence 2 was taken from [Click Here](#) or

[https://www.ncbi.nlm.nih.gov/nucore/NM\\_000041.4?report=fasta](https://www.ncbi.nlm.nih.gov/nucore/NM_000041.4?report=fasta)

Nucleotide

Advanced

FASTA

Send to: Change region shown

Customize view

Analyze this sequence

Run BLAST

Pick Primers

Show in Genome Data Viewer

Articles about the APOE gene

Interaction analysis of ancestry-enriched variants with APOE-ε4 on MCI in the Stud [Sci Rep. 2023]

Association of APOE (rs429358 and rs7412) and PON1 (Q192R and L55M) [Genes (Basel). 2023]

[Apolipoprotein E enhances migration of endometrial [Nan Fang Yi Ke Da Xue Xue Bao....]

See all...

Reference sequence information

RefSeq alternative splicing

See 5 reference mRNA sequence splice variants for the APOE gene.

These sequences are the interpretation of Human Apolipoprotein E with different variances. This APOE is involved in the metabolism of fats in human for making a protein called apolipoprotein E which combines with fats in the body to form lipoprotein.

### Task 3

This sequence can be found at: [Click Here](#) or <https://www.ncbi.nlm.nih.gov/gene/6331#gene-expression>

The screenshot shows the NCBI Gene database page for the gene SCN5A (sodium voltage-gated channel alpha subunit 5) in Homo sapiens (human). The page is titled "SCN5A sodium voltage-gated channel alpha subunit 5 [ Homo sapiens (human) ]" and includes a "Download Datasets" button. The "Summary" section provides key information:

- Official Symbol:** SCN5A provided by HGNC
- Official Full Name:** sodium voltage-gated channel alpha subunit 5 provided by HGNC
- Primary source:** HGNC:HGNC:10593
- See related:** Ensembl:ENSG00000183873 MIM:600163; AllianceGenome:HGNC:10593
- Gene type:** protein coding
- RefSeq status:** REVIEWED
- Organism:** Homo sapiens
- Lineage:** Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi; Mammalia; Eutheria; Euarchontoglires; Primates; Haplorhini; Catarrhini; Hominoidea; Homo
- Also known as:** HB1; HB2; HH1; IVF; VF1; HBBD; ICCD; LQT3; SSS1; CDCD2; CMD1E; CMPD2; PFHB1; Nav1.5
- Summary:** The protein encoded by this gene is an integral membrane protein and tetrodotoxin-resistant voltage-gated sodium channel subunit. This protein is found primarily in cardiac muscle and is responsible for the initial upstroke of the action potential in an electrocardiogram. Defects in this gene have been associated with long QT syndrome type 3 (LQT3), atrial fibrillation, cardiomyopathy, and Brugada syndrome 1, all autosomal dominant cardiac diseases. Alternative splicing results in several transcript variants encoding different isoforms. [provided by RefSeq, May 2022]
- Expression:** Restricted expression toward heart (RPKM 17.6) [See more](#)

The right sidebar contains a "Table of contents" with links to various sections: Summary, Genomic context, Genomic regions, transcripts, and products, Expression, Bibliography, Phenotypes, Variation, Pathways from PubChem, Interactions, General gene information, Markers, Homology, Gene Ontology, General protein information, NCBI Reference Sequences (RefSeq), Related sequences, and Additional links.

This protein is found in cardiac muscle and used as the initial upstroke of the action in an electrocardiogram. This gene has defects in its findings which contain such as Atrial Fibrillation, Cardiomyopathy, and Brugada Syndrome 1. I took the Sequence Reference from transcript [variant 2](#), [variant 3](#), [variant 4](#).

## Result

Google Colab Link: [Click Here](https://colab.research.google.com/drive/1bHldt-ufEXHFRC1wfmsZiTgWv4il7FSs?usp=sharing) or <https://colab.research.google.com/drive/1bHldt-ufEXHFRC1wfmsZiTgWv4il7FSs?usp=sharing>

### Task 1

```
# Task 1
from Bio import pairwise2
from Bio.Seq import Seq
from Bio.pairwise2 import format_alignment

# Sequence Initialization
seq1 = Seq('CTACTCAGCCCCAGCGGAGGTGAAGGACGTCCTTCCCAGGAGCCGGTGAGAAGCGCAGTCGGGGGCACGGGGATGAGCTCAGGGGCCCTAGAAAAGAGCTGGGACCTGGGAACCCCTGGCCTCCAGACTGGCCCAATC')
seq2 = Seq('CTACTCAGCCCCAGCGGAGGTGAAGGACGTCCTTCCCAGGAGCCGACTGGCCAATCACAGGCAGGAAGATGAAGTTCTGTGGGCTGCGTTGCTGGTCACATTCTGGCAGGATGCCAGGCCAAGGTGGAGCAAGCGG')

# Global Alignment
alignments = pairwise2.align.globalxx(seq1, seq2)
alignments
```

[1] ✓ 0.9s Python

d:\Iannn\Anaconda\envs\biopy\lib\site-packages\Bio\pairwise2.py:278: BiopythonDeprecationWarning: Bio.pairwise2 has been deprecated, and we intend to remove warnings.warn(

[Alignment(seqA='CTACTCAGCCCCAGCGGAGGTGAAGGACGTCCTTCCCAGGAGCCGGTGAGAAGCGCAGTCGGGGGCACGGGGATGAGCTCAGGGGCCCTAGAAAAGAGCTGGGACCTGGGAACCCCTGGCCTCCAGACTGGCCCAAT', Alignment(seqA='CTACTCAGCCCCAGCGGAGGTGAAGGACGTCCTTCCCAGGAGCCGGTGAGAAGCGCAGTCGGGGGCACGGGGATGAGCTCAGGGGCCCTAGAAAAGAGCTGGGACCTGGGAACCCCTGGCCTCCAGACTGGCCCAAT', Alignment(seqA='CTACTCAGCCCCAGCGGAGGTGAAGGACGTCCTTCCCAGGAGCCGGTGAGAAGCGCAGTCGGGGGCACGGGGATGAGCTCAGGGGCCCTAGAAAAGAGCTGGGACCTGGGAACCCCTGGCCTCCAGACTGGCCCAAT', Alignment(seqA='CTACTCAGCCCCAGCGGAGGTGAAGGACGTCCTTCCCAGGAGCCGGTGAGAAGCGCAGTCGGGGGCACGGGGATGAGCTCAGGGGCCCTAGAAAAGAGCTGGGACCTGGGAACCCCTGGCCTCCAGACTGGCCCAAT

```
# Alignment format
print(format_alignment(*alignments[0]))
```

[2] ✓ 0.1s Python

```
CTACTCAGCCCCAGCGGAGGTGAAGGACGTCCTTCCCAGGAGCCGGTGAGAAGCGCAGTCGGGGGCACGGGGATGAGCTCAGGGGCCCTAGAAAAGAGCTGGGACCTGGGAACCCCTGGCCTCCAGACTGGCCCAATCACAGGCAGGAAGATGA
|||||
CTACTCAGCCCCAGCGGAGGT-----G-----A-----G-G-A--C-GT-----C-C-----T---T-----CC-C-----C---A-----GG-A-----G---CC--GACTGGCCCAATCACAGGCAGGAAGATGA
Score=1166
```

```
# View all possible alignment
for a in alignments:
    print(format_alignment(*a))
```

[3] ✓ 0.1s Python

```
CTACTCAGCCCCAGCGGAGGTGAAGGACGTCCTTCCCAGGAGCCGGTGAGAAGCGCAGTCGGGGGCACGGGGATGAGCTCAGGGGCCCTAGAAAAGAGCTGGGACCTGGGAACCCCTGGCCTCCAGACTGGCCCAATCACAGGCAGGAAGATGA
|||||
CTACTCAGCCCCAGCGGAGGT-----G-----A-----G-G-A--C-GT-----C-C-----T---T-----CC-C-----C---A-----GG-A-----G---CC--GACTGGCCCAATCACAGGCAGGAAGATGA
Score=1166

CTACTCAGCCCCAGCGGAGGTGAAGGACGTCCTTCCCAGGAGCCGGTGAGAAGCGCAGTCGGGGGCACGGGGATGAGCTCAGGGGCCCTAGAAAAGAGCTGGGACCTGGGAACCCCTGGCCTCCAGACTGGCCCAATCACAGGCAGGAAGATGA
|||||
CTACTCAGCCCCAGCGGAGGT-----G-----A-----G-G-A--C-GT-----C-C-----T---T-----CC-C-----C---A-----GG-A-----G---CC--GACTGGCCCAATCACAGGCAGGAAGATGA
Score=1166
```

### Task 2

```
# Task 2
from Bio import pairwise2
from Bio.Seq import Seq

# Sequence Initialization
seq1 = Seq('CTACTCAGCCCCAGCGGAGGTGAAGGACGTCCTTCCCAGGAGCCGGTGAGAAGCGCAGTCGGGGGCACGGGGATGAGCTCAGGGGCCCTAGAAAAGAGCTGGGACCTGGGAACCCCTGGCCTCCAGACTGGCCCAATC')
seq2 = Seq('CTACTCAGCCCCAGCGGAGGTGAAGGACGTCCTTCCCAGGAGCCGACTGGCCAATCACAGGCAGGAAGATGAAGTTCTGTGGGCTGCGTTGCTGGTCACATTCTGGCAGGATGCCAGGCCAAGGTGGAGCAAGCGG')

# Get the alignment by only the score (Global)
glb_alignment2 = pairwise2.align.globalxx(
    seq1, seq2, one_alignment_only=True, score_only=True)
global_alignment = glb_alignment2/len(seq1) * 100

# Get the local alignment by only the score (LOCAL)
loc_alignment2 = pairwise2.align.localxx(
    seq1, seq2, one_alignment_only=True, score_only=True)
local_alignment = loc_alignment2/len(seq1) * 100

print("Global Alignment: ", global_alignment)
print("Local Alignment: ", local_alignment)
```

[19] ✓ 0.1s Python

Global Alignment: 93.42948717948718  
Local Alignment: 93.42948717948718

## Task 3

```

# Task 3
# Sequence initialization
var2 = Seq('AGAGCCGCGGAGCCGAGACGGCGGCGGCCCTAGGATGCAGGGATCGTCCCCGGGGCCGCTGAGCCTGCGCCAGTCCCCGAGCCCCGCGCCGAGCCGAGTCCGCGCCAAGCAGCAGCCGCCACCCCGGGGCC')
var3 = Seq('AGAGCCGCGGAGCCGAGACGGCGGCGGCCCTAGGATGCAGGGATCGTCCCCGGGGCCGCTGAGCCTGCGCCAGTCCCCGAGCCCCGCGCCGAGCCGAGTCCGCGCCAAGCAGCAGCCGCCACCCCGGGGCC')
var4 = Seq('AGAGCCGCGGAGCCGAGACGGCGGCGGCCCTAGGATGCAGGGATCGTCCCCGGGGCCGCTGAGCCTGCGCCAGTCCCCGAGCCCCGCGCCGAGCCGAGTCCGCGCCAAGCAGCAGCCGCCACCCCGGGGCC')

# Perform Local Alignment and print the scores
AvB = pairwise2.align.localxx(
    var2, var3, one_alignment_only=True, score_only=True)
BvC = pairwise2.align.localxx(
    var3, var4, one_alignment_only=True, score_only=True)
AvC = pairwise2.align.localxx(
    var2, var4, one_alignment_only=True, score_only=True)

# Calculate and display the percentage of the alignment score
print("AvB", AvB/len(var3) * 100)
print("BvC", BvC/len(var3) * 100)
print("AvC", AvC/len(var4) * 100)

# Check the concept : Is 100? similarity score means the two DNA sequences are exact match?
print(var2 == var3)

```

[20] ✓ 2.0s Python

```

... AvB 99.65958445826975
    BvC 99.36612278436436
    AvC 99.6574128765505
    False

```