Test\_sequence\_matching.py

import unittest

from src.sequence\_matcher import read\_sequences

from Bio import SeqIO

class TestSequenceMatcher(unittest.TestCase):

def test\_read\_sequences\_valid(self):

# Test if the FASTA file is read correctly

sequences = read\_sequences("data/dog\_breeds.fa")

self.assertGreater(len(sequences), 0) # Ensure sequences were loaded

def test\_read\_sequences\_invalid(self):

# Test if the function handles invalid files gracefully

sequences = read\_sequences("data/nonexistent\_file.fa")

self.assertEqual(sequences, []) # No sequences should be read

def test\_find\_best\_match(self):

# Test that the correct best match is found

dog\_breeds\_sequences = list(SeqIO.parse("data/dog\_breeds.fa", "fasta"))

mystery\_sequences = list(SeqIO.parse("data/mystery.fa", "fasta"))

best\_match, best\_score = find\_best\_match(mystery\_sequences[0], dog\_breeds\_sequences)

self.assertIsNotNone(best\_match) # Ensure a match is found

self.assertGreater(best\_score, 0) # Ensure score is positive (indicating a match)

if \_\_name\_\_ == "\_\_main\_\_":

unittest.main()