# Sequence\_matcher.py saved in the 4. Source code folder

‘’’ This function will start reading the input FASTA files (dog\_breeds.fa and mystery.fa) using **BioPython's SeqIO** module. This will allow the processing of the sequences file and find the best match against the mystery sequence.

It includes the following executable steps:

**read\_sequences**: Reads sequences from a FASTA file and returns them as a list.

**save\_results**: Saves the closest match and alignment score to a text file in the results/ folder.

**main**: The main function that coordinates the process of reading files, finding the best match, and saving the results.

‘’’

from Bio import SeqIO

def read\_sequences(file\_path):

"""Reads sequences from a FASTA file."""

try:

sequences = list(SeqIO.parse(file\_path, "fasta"))

return sequences

except FileNotFoundError:

print(f"Error: The file {file\_path} was not found.")

return []

except Exception as e:

print(f"Error while reading the file {file\_path}: {e}")

return []

def main():

# Define file paths for data files

dog\_breeds\_file = "data/dog\_breeds.fa"

mystery\_file = "data/mystery.fa"

# Read sequences from the files

dog\_breeds\_sequences = read\_sequences(dog\_breeds\_file)

mystery\_sequences = read\_sequences(mystery\_file)

if dog\_breeds\_sequences and mystery\_sequences:

print("Files successfully read!")

else:

print("Error reading files.")

return

# Further processing will be done here (alignment and matching)

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

main()

# code continues to store the best match sequence and its alignment score in a **text file**

called classification\_results.txt in the **6. Results** folder.

from align\_sequences import find\_best\_match

from Bio import SeqIO

def save\_results(best\_match, best\_score):

"""Save the best match and score to a results file."""

with open("results/classification\_results.txt", "w") as result\_file:

result\_file.write(f"Best Match: {best\_match.id}\n")

result\_file.write(f"Alignment Score: {best\_score}\n")

def main():

# Define file paths

dog\_breeds\_file = "data/dog\_breeds.fa"

mystery\_file = "data/mystery.fa"

# Read sequences

dog\_breeds\_sequences = read\_sequences(dog\_breeds\_file)

mystery\_sequences = read\_sequences(mystery\_file)

if not dog\_breeds\_sequences or not mystery\_sequences:

print("Error: Missing or empty sequences. Exiting.")

return

# Assuming we have only one mystery sequence

test\_sequence = mystery\_sequences[0]

# Find the best match from the dog breed database

best\_match, best\_score = find\_best\_match(test\_sequence, dog\_breeds\_sequences)

# Save the result to a file

if best\_match:

print(f"Best match: {best\_match.id} with a score of {best\_score}")

save\_results(best\_match, best\_score)

else:

print("No close match found.")

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

main()