Using VQE with CVaR expectation value_COBYLA_optimzer_9_amino_Bradykinin

April 4, 2023

[1]:
```
from qiskit_research.protein_folding.interactions.random_interaction import (
    RandomInteraction,
)
from qiskit_research.protein_folding.interactions.miyazawa_jernigan_interaction␣
 ↪import (
    MiyazawaJerniganInteraction,
)
from qiskit_research.protein_folding.peptide.peptide import Peptide
from qiskit_research.protein_folding.protein_folding_problem import (
    ProteinFoldingProblem,
)

from qiskit_research.protein_folding.penalty_parameters import PenaltyParameters

from qiskit.utils import algorithm_globals, QuantumInstance

algorithm_globals.random_seed = 25
```

[2]:
```
main_chain = "RPPGFSPFR" #Bradykinin Peptide
```

[3]:
```
side_chains = [""] * 9
```

[4]:
```
random_interaction = RandomInteraction()
mj_interaction = MiyazawaJerniganInteraction()
```

[5]:
```
penalty_back = 10
penalty_chiral = 10
penalty_1 = 10

penalty_terms = PenaltyParameters(penalty_chiral, penalty_back, penalty_1)
```

[6]:
```
peptide = Peptide(main_chain, side_chains)
```

[7]:
```
protein_folding_problem = ProteinFoldingProblem(peptide, mj_interaction,␣
 ↪penalty_terms)
qubit_op = protein_folding_problem.qubit_op()
```

[8]:
```
print(qubit_op)
```

```
5764.091 * IIIIIIIIIIIIIIII
+ 1240.0 * IIIIIIIIIIIIIIZII
- 550.0 * IIIIIIIIIIIIIIZZ
+ 550.0 * IIIIIIIIIIIIIIZZZ
- 652.5 * IIIIIIIIIIIIZIZII
- 652.5 * IIIIIIIIIIIIIZIZI
- 652.5 * IIIIIIIIIIIIZZZZI
- 550.0 * IIIIIIIIIIZIZIIII
- 550.0 * IIIIIIIIIIIZIZIII
- 550.0 * IIIIIIIIIIZZZZIII
- 457.5 * IIIIIIIIZIZIIIIII
- 457.5 * IIIIIIIIIZIZIIIII
- 457.5 * IIIIIIIIZZZZIIIII
- 227.5 * IIIIIIIZIZIIIIIII
- 227.5 * IIIIIIIIZIZIIIIII
- 227.5 * IIIIIIIZZZZIIIIII
+ 690.0 * IIIIIIIIIIIIIIZZI
- 1250.0 * IIIIIIIIIIIIIZIIII
- 695.0 * IIIIIIIIIIIIIZZIII
+ 460.0 * IIIIIIIIIIIIIIIIZ
+ 555.0 * IIIIIIIIIIIIIZIIZ
- 555.0 * IIIIIIIIIIIIIZZIIZ
- 904.5565 * IIIIIZIIIIIIIIIII
- 295.0 * IIIIIZIIIIIIIZII
- 197.5 * IIIIIZIIIIIIIZZI
+ 302.5 * IIIIIZIIIIIIZIIII
+ 202.5 * IIIIIZIIIIIIZZIII
+ 100.0 * IIIIIZIIIIIIZIZII
+ 100.0 * IIIIIZIIIIIIZIZI
+ 100.0 * IIIIIZIIIIIIZZZZI
- 200.0 * IIIIIZIIIIIIIIIIZ
+ 97.5 * IIIIIZIIIIIIIIIZZ
- 97.5 * IIIIIZIIIIIIIIZZZ
- 100.0 * IIIIIZIIIIIIIZIIZ
+ 100.0 * IIIIIZIIIIIIIZZIIZ
+ 230.0 * IIIIIIIIIIIIIIIZI
- 230.0 * IIIIIIIIIIIIIZIII
+ 2.5 * IIIIIZIIIIIIIIIZI
- 2.5 * IIIIIZIIIIIIIZIII
+ 945.0 * IIIIIIIIIIZIIIIII
+ 490.0 * IIIIIIIIIIZZIIIII
+ 555.0 * IIIIIIIIIIZIIIZII
+ 555.0 * IIIIIIIIIIZIIIZI
+ 555.0 * IIIIIIIIIIZZIIZZI
- 455.0 * IIIIIIIIIIIZIIIIZ
+ 455.0 * IIIIIIIIIIZZIIIIZ
```

```
+ 7.5 * IIIIIZIIIIZIIIIII
+ 5.0 * IIIIIZIIIIZZIIIII
+ 2.5 * IIIIIZIIIIZIIIZII
+ 2.5 * IIIIIZIIIIIZIIIZI
+ 2.5 * IIIIIZIIIIZZIIZZI
- 2.5 * IIIIIZIIIIZIZIIII
- 2.5 * IIIIIZIIIIIZIZIII
- 2.5 * IIIIIZIIIIZZZZIII
- 2.5 * IIIIIZIIIIIZIIIIZ
+ 2.5 * IIIIIZIIIIZZIIIIZ
- 762.5 * IIIIIIIIZIIIIIIII
- 400.0 * IIIIIIIIZZIIIIIII
- 462.5 * IIIIIIIIZIIIIIZII
- 462.5 * IIIIIIIIIZIIIIIZI
- 462.5 * IIIIIIIIZZIIIIZZI
+ 460.0 * IIIIIIIIZIIIZIIII
+ 460.0 * IIIIIIIIIZIIIZIII
+ 460.0 * IIIIIIIIZZIIZZIII
+ 362.5 * IIIIIIIIIZIIIIIIZ
- 362.5 * IIIIIIIIZZIIIIIIZ
- 1616.9259999999997 * IIIIZIIIIIIIIIIII
- 400.0 * IIIIZIIIIIIIIZII
- 267.5 * IIIIZIIIIIIIIZZI
+ 400.0 * IIIIZIIIIIIIZIIII
+ 267.5 * IIIIZIIIIIIIZZIII
- 400.0 * IIIIZIIIIIZIIIIII
- 267.5 * IIIIZIIIIIZZIIIII
+ 407.5 * IIIIZIIIZIIIIIIII
+ 272.5 * IIIIZIIIZZIIIIIII
+ 132.5 * IIIIZIIIIIIIZIZII
- 132.5 * IIIIZIIIIIZIIIZII
+ 135.0 * IIIIZIIIZIIIIIZII
+ 132.5 * IIIIZIIIIIIIIZIZI
- 132.5 * IIIIZIIIIIIZIIIZI
+ 135.0 * IIIIZIIIIZIIIIIZI
+ 132.5 * IIIIZIIIIIIIZZZZI
- 132.5 * IIIIZIIIIIZZIIZZI
+ 135.0 * IIIIZIIIZZIIIIZZI
+ 132.5 * IIIIZIIIIIZIZIIII
- 135.0 * IIIIZIIIZIIIZIIII
+ 132.5 * IIIIZIIIIIIZIZIII
- 135.0 * IIIIZIIIIZIIIZIII
+ 132.5 * IIIIZIIIIIZZZZIII
- 135.0 * IIIIZIIIZZIIZZIII
+ 135.0 * IIIIZIIIZIZIIIIII
+ 135.0 * IIIIZIIIIZIZIIIII
+ 135.0 * IIIIZIIIZZZZIIIII
- 270.0 * IIIIZIIIIIIIIIIIZ
```

```
+ 132.5 * IIIIZIIIIIIIIIIZZ
- 132.5 * IIIIZIIIIIIIIIIZZZ
- 132.5 * IIIIZIIIIIIIIZIIZ
+ 132.5 * IIIIZIIIIIIIIZZIIZ
+ 132.5 * IIIIZIIIIIIIZIIIIZ
- 132.5 * IIIIZIIIIIIZZIIIIZ
- 135.0 * IIIIZIIIIIZIIIIIIZ
+ 135.0 * IIIIZIIIIZZIIIIIIZ
+ 235.0 * IIIIIIIIIIIIZIIIIII
- 130.0 * IIIIIIIIIIZIIIIIIII
+ 2.5 * IIIIZIIIIIIIIIIZI
- 2.5 * IIIIZIIIIIIIIIZIII
+ 2.5 * IIIIZIIIIIIIZIIIII
- 2.5 * IIIIZIIIIIZIIIIIII
+ 257.5 * IIIIIIIZIIIIIIIIII
+ 130.0 * IIIIIIIZZIIIIIIIII
+ 232.5 * IIIIIIIZIIIIIIIZII
+ 232.5 * IIIIIIIZIIIIIIIIZI
+ 232.5 * IIIIIIIZZIIIIIIIZZI
- 230.0 * IIIIIIIZIIIIIIZIIII
- 230.0 * IIIIIIIZIIIIIIZIIII
- 230.0 * IIIIIIIZZIIIIIZZIII
+ 230.0 * IIIIIIIZIIIIZIIIIII
+ 230.0 * IIIIIIIZIIIIZIIIIII
+ 230.0 * IIIIIIIZZIIZZIIIIII
- 127.5 * IIIIIIIZIIIIIIIIIZ
+ 127.5 * IIIIIIIZZIIIIIIIIZ
+ 7.5 * IIIIZIZIIIIIIIIIII
+ 5.0 * IIIIZIZZIIIIIIIIII
+ 2.5 * IIIIZIZIIIIIIIZII
+ 2.5 * IIIIZIIZIIIIIIIZI
+ 2.5 * IIIIZIZZIIIIIIZZI
- 2.5 * IIIIZIZIIIIIIZIIII
- 2.5 * IIIIZIIZIIIIIIZIII
- 2.5 * IIIIZIZZIIIIIZZIII
+ 2.5 * IIIIZIZIIIIZIIIIII
+ 2.5 * IIIIZIIZIIIIZIIIII
+ 2.5 * IIIIZIZZIIZZIIIIII
- 2.5 * IIIIZIZIZIIIIIIIII
- 2.5 * IIIIZIIZIZIIIIIIII
- 2.5 * IIIIZIZZZIIIIIIII
- 2.5 * IIIIZIIZIIIIIIIIZ
+ 2.5 * IIIIZIZZIIIIIIIIZ
- 686.9490000000001 * IIIZIIIIIIIIIIIII
- 190.0 * IIIZIIIIIIIIIZII
- 100.0 * IIIZIIIIIIIIIIZI
- 95.0 * IIIZIIIIIIIIIIZZI
+ 190.0 * IIIZIIIIIIIZIIII
```

```
+ 100.0 * IIIZIIIIIIIIIZIII
+ 95.0 * IIIZIIIIIIIIZZIII
- 195.0 * IIIZIIIIIIZIIIIII
- 102.5 * IIIZIIIIIIIZIIIII
- 97.5 * IIIZIIIIIIZZIIIII
+ 95.0 * IIIZIIIIIIIIZIZII
- 97.5 * IIIZIIIIIIIZIIIZII
+ 95.0 * IIIZIIIIIIIIIZIZI
- 97.5 * IIIZIIIIIIIZIIIZI
+ 95.0 * IIIZIIIIIIIIZZZZI
- 97.5 * IIIZIIIIIIZZIIZZI
+ 97.5 * IIIZIIIIIIZIZIIII
+ 97.5 * IIIZIIIIIIIZIZIII
+ 97.5 * IIIZIIIIIIZZZZIII
+ 95.0 * IIIZIIIIIIIIIIIZZ
- 95.0 * IIIZIIIIIIIIIIZZZ
- 95.0 * IIIZIIIIIIIIIZIIZ
+ 95.0 * IIIZIIIIIIIIZZIIZ
+ 97.5 * IIIZIIIIIIIZIIIIZ
- 97.5 * IIIZIIIIIIZZIIIIZ
+ 5.0 * IIIZIIIIIIIIIIIIZ
- 5.0 * IIIZIIIIZIIIIIIII
- 2.5 * IIIZIIIIIZIIIIIII
- 2.5 * IIIZIIIIZZIIIIIII
- 2.5 * IIIZIIIIZIIIIIZII
- 2.5 * IIIZIIIIIZIIIIIZI
- 2.5 * IIIZIIIIZZIIIIZZI
+ 2.5 * IIIZIIIIZIIIZIIII
+ 2.5 * IIIZIIIIIZIIIZIII
+ 2.5 * IIIZIIIIZZIIZZIII
- 2.5 * IIIZIIIIZIZIIIIII
- 2.5 * IIIZIIIIIZIZIIIII
- 2.5 * IIIZIIIIZZZZIIIII
+ 2.5 * IIIZIIIIIZIIIIIIZ
- 2.5 * IIIZIIIIZZIIIIIIZ
+ 140.0 * IIIIIIIZIIIIIIIII
- 1344.5399999999997 * IIZIIIIIIIIIIIIII
- 265.0 * IIZIIIIIIIIIIZII
- 137.5 * IIZIIIIIIIIIIIZI
- 132.5 * IIZIIIIIIIIIIZZI
+ 265.0 * IIZIIIIIIIIIZIIII
+ 137.5 * IIZIIIIIIIIIIZIII
+ 132.5 * IIZIIIIIIIIIIZZII
- 265.0 * IIZIIIIIIIZIIIIII
- 137.5 * IIZIIIIIIIIZIIIII
- 132.5 * IIZIIIIIIIIZZIIII
+ 265.0 * IIZIIIIIZIIIIIIII
+ 137.5 * IIZIIIIIIZIIIIIII
```

```
+ 132.5 * IIZIIIIIZZIIIIIII
- 270.0 * IIZIIIZIIIIIIIII
- 140.0 * IIZIIIIZIIIIIIII
- 135.0 * IIZIIIZZIIIIIIII
+ 132.5 * IIZIIIIIIIIIZIZII
- 132.5 * IIZIIIIIIIZIIIZII
+ 132.5 * IIZIIIIIZIIIIIZII
- 135.0 * IIZIIIZIIIIIIIZII
+ 132.5 * IIZIIIIIIIIIIZIZI
- 132.5 * IIZIIIIIIIIZIIIZI
+ 132.5 * IIZIIIIIIZIIIIIZI
- 135.0 * IIZIIIIZIIIIIIIZI
+ 132.5 * IIZIIIIIIIIIIZZZZI
- 132.5 * IIZIIIIIIIIZZIIZZI
+ 132.5 * IIZIIIIIZZIIIIZZI
- 135.0 * IIZIIIZZIIIIIIZZI
+ 132.5 * IIZIIIIIIIZIZIIII
- 132.5 * IIZIIIIIZIIIZIIII
+ 135.0 * IIZIIIZIIIIIZIIII
+ 132.5 * IIZIIIIIIIIIZIZIII
- 132.5 * IIZIIIIIIZIIIZIII
+ 135.0 * IIZIIIIZIIIIIZIII
+ 132.5 * IIZIIIIIIIIZZZZIII
- 132.5 * IIZIIIIIZZIIZZIII
+ 135.0 * IIZIIIZZIIIIZZIII
+ 132.5 * IIZIIIIIZIZIIIIII
- 135.0 * IIZIIIZIIIZIIIIII
+ 132.5 * IIZIIIIIIZIZIIIII
- 135.0 * IIZIIIIZIIIZIIIII
+ 132.5 * IIZIIIIIZZZZIIIII
- 135.0 * IIZIIIZZIIZZIIIII
+ 135.0 * IIZIIIZIZIIIIIIII
+ 135.0 * IIZIIIIZIZIIIIIII
+ 135.0 * IIZIIIZZZZIIIIIII
+ 132.5 * IIZIIIIIIIIIIIIZZ
- 132.5 * IIZIIIIIIIIIIIZZZ
- 132.5 * IIZIIIIIIIIIIZIIZ
+ 132.5 * IIZIIIIIIIIIIZZIIZ
+ 132.5 * IIZIIIIIIIIIZIIIIZ
- 132.5 * IIZIIIIIIIIZZIIIIZ
- 132.5 * IIZIIIIIIIZIIIIIIZ
+ 132.5 * IIZIIIIIIIZZIIIIIZ
+ 135.0 * IIZIIIIZIIIIIIIIIZ
- 135.0 * IIZIIIZZIIIIIIIIIZ
+ 5.0 * IIZIIIIIIIIIIIIIZ
- 591.696 * IZIIIIIIIIIIIIIII
+ 95.0 * IZIIIIIIIIIIZIZII
- 95.0 * IZIIIIIIIIZIIIZII
```

```
+ 97.5 * IZIIIIIIZIIIIIZII
- 95.0 * IZIIIIIIIIIIIZII
+ 95.0 * IZIIIIIIIIIIIZIZI
- 95.0 * IZIIIIIIIIIZIIIZI
+ 97.5 * IZIIIIIIIZIIIIIZI
+ 95.0 * IZIIIIIIIIIIZZZZI
- 95.0 * IZIIIIIIIIZZIIZZI
+ 97.5 * IZIIIIIIZZIIIIZZI
+ 95.0 * IZIIIIIIIIZIZIIII
- 97.5 * IZIIIIIIZIIIZIIII
+ 95.0 * IZIIIIIIIIIIZIIII
+ 95.0 * IZIIIIIIIIIZIZIII
- 97.5 * IZIIIIIIIZIIIZIII
+ 95.0 * IZIIIIIIIIZZZZIII
- 97.5 * IZIIIIIIZZIIZZIII
+ 97.5 * IZIIIIIIZIZIIIIII
- 95.0 * IZIIIIIIIIZIIIIII
+ 97.5 * IZIIIIIIIZIZIIIII
+ 97.5 * IZIIIIIIZZZZIIIII
+ 97.5 * IZIIIIIIZIIIIIIII
+ 97.5 * IZIIIIIIIIIIIIIZZ
- 97.5 * IZIIIIIIIIIIIIZZZ
- 97.5 * IZIIIIIIIIIIIZIIZ
+ 97.5 * IZIIIIIIIIIIZZIIZ
+ 97.5 * IZIIIIIIIIIZIIIIZ
- 97.5 * IZIIIIIIIIZZIIIIZ
- 100.0 * IZIIIIIIIZIIIIIIZ
+ 100.0 * IZIIIIIIZZIIIIIIZ
+ 2.5 * IZIIIIIIIIIIIIZI
+ 2.5 * IZIIIIIIIIIIIIZZI
- 2.5 * IZIIIIIIIIIIIZIII
- 2.5 * IZIIIIIIIIIIZZIII
+ 2.5 * IZIIIIIIIIIZIIIII
+ 2.5 * IZIIIIIIIIZZIIIII
- 2.5 * IZIIIIIIIZIIIIIII
- 2.5 * IZIIIIIIZZIIIIIII
+ 2.5 * IZIIIIIZIIIIIIIII
+ 2.5 * IZIIIIIZIIIIIIIZI
+ 2.5 * IZIIIIIZIIIIIIIZZI
- 2.5 * IZIIIIIZIIIIIZIII
- 2.5 * IZIIIIIZIIIIIZIII
- 2.5 * IZIIIIIZZIIIIZZII
+ 2.5 * IZIIIIIZIIIZIIIII
+ 2.5 * IZIIIIIZIIIZIIIII
+ 2.5 * IZIIIIIZZIIZZIIII
- 2.5 * IZIIIIIZIZIIIIIII
- 2.5 * IZIIIIIZIZIIIIIII
```

```
- 2.5 * IZIIIIZZZZIIIIIII
- 2.5 * IZIIIIIZIIIIIIIIZ
+ 2.5 * IZIIIIIZZIIIIIIIZ
- 606.9235 * ZIIIIIIIIIIIIIII
+ 100.0 * ZIIIIIIIIIIIZIZII
- 100.0 * ZIIIIIIIIIZIIIZII
+ 100.0 * ZIIIIIIIZIIIIIZII
- 102.5 * ZIIIIIZIIIIIIIZII
+ 100.0 * ZIIIIIIIIIIIIZIZI
- 100.0 * ZIIIIIIIIIIZIIIZI
+ 100.0 * ZIIIIIIIIZIIIIIZI
- 102.5 * ZIIIIIIZIIIIIIIZI
+ 100.0 * ZIIIIIIIIIIIZZZZI
- 100.0 * ZIIIIIIIIIZZIIZZI
+ 100.0 * ZIIIIIIIZZIIIIZZI
- 102.5 * ZIIIIIZZIIIIIIZZI
+ 97.5 * ZIIIIIIIIIZIZIIII
- 97.5 * ZIIIIIIIZIIIZIIII
+ 100.0 * ZIIIIIZIIIIIZIIII
+ 97.5 * ZIIIIIIIIIIZIZIII
- 97.5 * ZIIIIIIIIZIIIZIII
+ 100.0 * ZIIIIIIZIIIIIZIII
+ 97.5 * ZIIIIIIIIIZZZZIII
- 97.5 * ZIIIIIIIZZIIZZIII
+ 100.0 * ZIIIIIZZIIIIZZIII
+ 97.5 * ZIIIIIIIZIZIIIIII
- 100.0 * ZIIIIIZIIIZIIIIII
+ 97.5 * ZIIIIIIIIZIZIIIII
- 100.0 * ZIIIIIIZIIIZIIIII
+ 97.5 * ZIIIIIIIZZZZIIIII
- 100.0 * ZIIIIIZZIIZZIIIII
+ 100.0 * ZIIIIIZIZIIIIIIII
+ 100.0 * ZIIIIIIZIZIIIIIII
+ 100.0 * ZIIIIIZZZZIIIIIII
+ 2.5 * ZIIIIIIIIIIIIIZII
- 2.5 * ZIIIIIIIIIIIZIIII
+ 2.5 * ZIIIIIIIIIZIIIIII
- 2.5 * ZIIIIIIIZIIIIIIII
+ 2.5 * ZIIIIIZIIIIIIIIII
- 2.5 * ZIIIIIIIIIIIIIZZ
+ 2.5 * ZIIIIIIIIIIIIZZZ
+ 2.5 * ZIIIIIIIIIIIIZIIZ
- 2.5 * ZIIIIIIIIIIIZZIIZ
- 2.5 * ZIIIIIIIIIIZIIIIZ
+ 2.5 * ZIIIIIIIIIZZIIIIZ
+ 2.5 * ZIIIIIIIIZIIIIIIZ
- 2.5 * ZIIIIIIIZZIIIIIIZ
- 2.5 * ZIIIIIIZIIIIIIIIZ
```

```
+ 2.5 * ZIIIIIZZIIIIIIIIZ
```

[9]:
```python
# Get the qubit operator and count the number of qubits
qubit_op = protein_folding_problem.qubit_op()
num_qubits = qubit_op.num_qubits

print(f"The protein folding problem requires {num_qubits} qubits.")
```

```
The protein folding problem requires 17 qubits.
```

[10]:
```python
from qiskit.circuit.library import RealAmplitudes
from qiskit.algorithms.optimizers import COBYLA, SPSA
from qiskit.algorithms import NumPyMinimumEigensolver
from qiskit.algorithms.minimum_eigensolvers import SamplingVQE
from qiskit import execute, Aer
from qiskit.primitives import Sampler
```

[11]:
```python
# set classical optimizer
optimizer = SPSA(maxiter=150)

# set variational ansatz
ansatz = RealAmplitudes(reps=1)

counts = []
values = []
```

[12]:
```python
def store_intermediate_result(eval_count, parameters, mean, std):
    counts.append(eval_count)
    values.append(mean)


# initialize VQE using CVaR with alpha = 0.1
vqe = SamplingVQE(
    Sampler(),
    ansatz=ansatz,
    optimizer=optimizer,
    aggregation=0.1,
    callback=store_intermediate_result,
)

raw_result = vqe.compute_minimum_eigenvalue(qubit_op)
print(raw_result)
```

```
SamplingMinimumEigensolverResult:
        Eigenvalue: 59.883826428389135
        Best measurement
: {'state': 69827, 'bitstring': '10001000011000011', 'value':
(-2.3010000000000446+0j), 'probability': 1.345470155e-05}
```

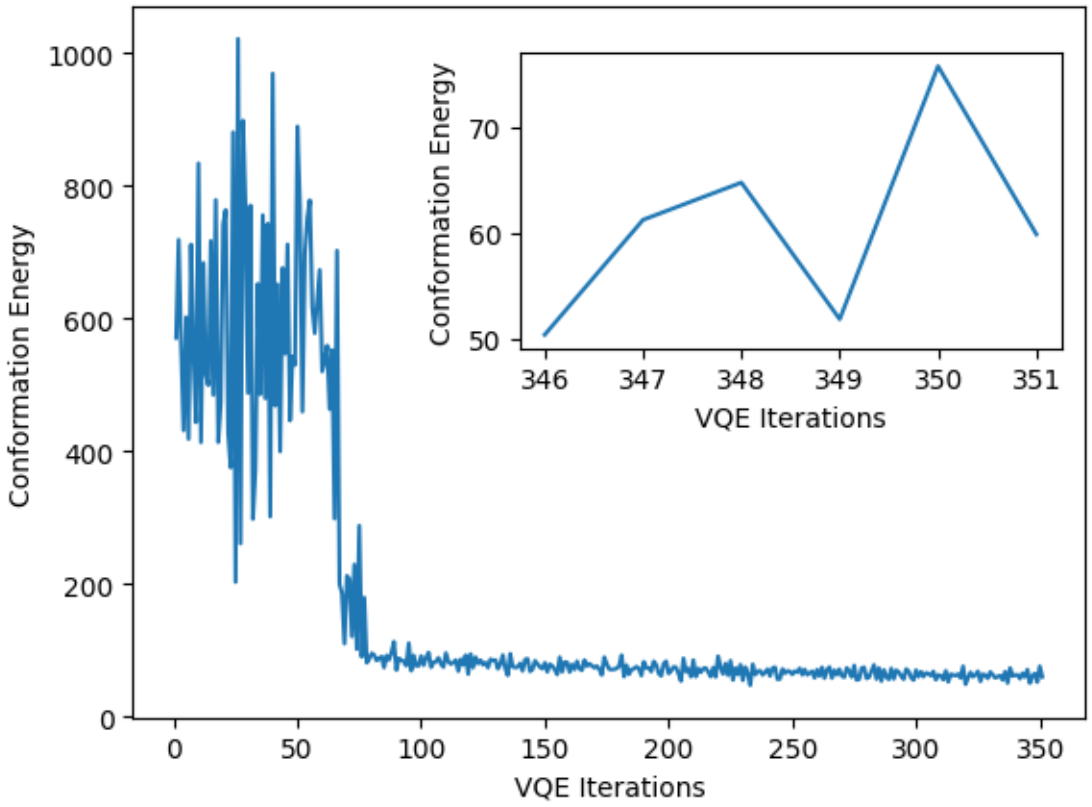
[19]:
```
import matplotlib.pyplot as plt

fig = plt.figure()

plt.plot(counts, values)
plt.ylabel("Conformation Energy")
plt.xlabel("VQE Iterations")

fig.add_axes([0.44, 0.51, 0.44, 0.32])

plt.plot(counts[345:], values[345:])
plt.ylabel("Conformation Energy")
plt.xlabel("VQE Iterations")
plt.show()
```



[14]:
```
result = protein_folding_problem.interpret(raw_result=raw_result)
print(
    "The bitstring representing the shape of the protein during optimization is:
 ↪ ",
    result.turn_sequence,
)
```

```
print("The expanded expression is:", result.get_result_binary_vector())
```

```
The bitstring representing the shape of the protein during optimization is:
01010000010000100
The expanded expression is: 0_______1_____0_1_____0_0_________________________
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
_____________________________0001000010_0____
```

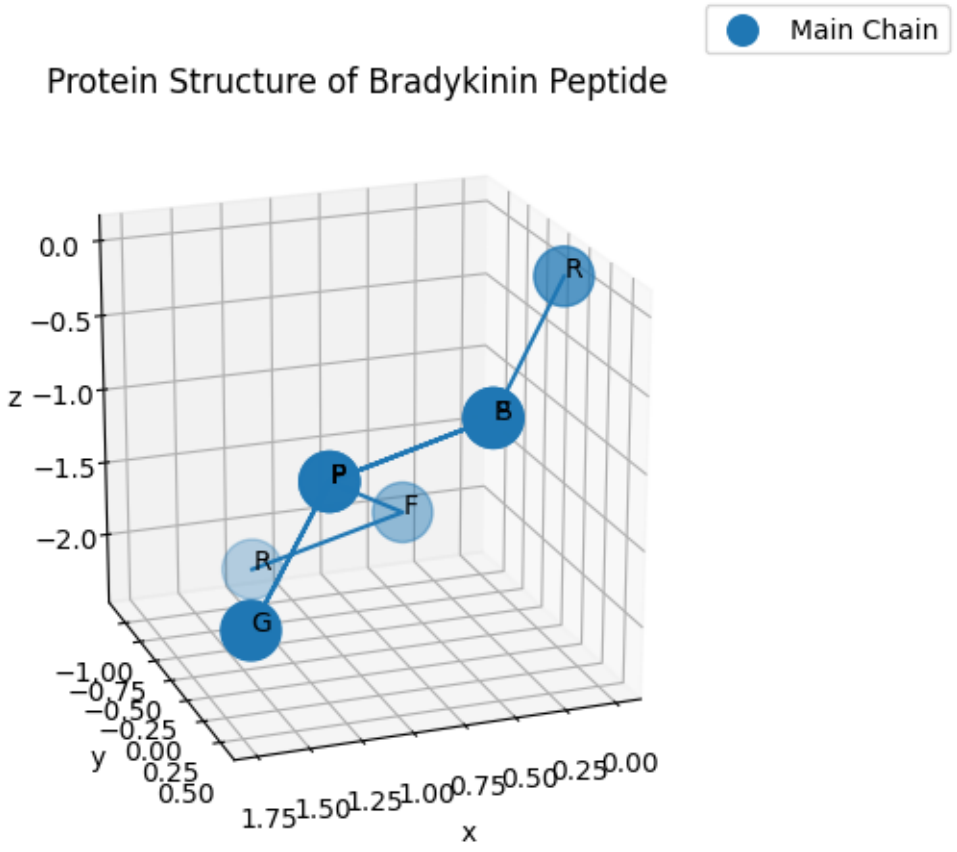
[15]:
```
print(
    f"The folded protein's main sequence of turns is: {result.
 ↪protein_shape_decoder.main_turns}"
)
print(f"and the side turn sequences are: {result.protein_shape_decoder.
 ↪side_turns}")
```

```
The folded protein's main sequence of turns is: [1, 0, 1, 1, 0, 0, 2, 0]
and the side turn sequences are: [None, None, None, None, None, None, None,
None, None]
```

[16]:
```
print(result.protein_shape_file_gen.get_xyz_data())
```

```
[['R' '0.0' '0.0' '0.0']
 ['P' '0.5773502691896258' '0.5773502691896258' '-0.5773502691896258']
 ['P' '1.1547005383792517' '0.0' '-1.1547005383792517']
 ['G' '1.7320508075688776' '0.5773502691896258' '-1.7320508075688776']
 ['F' '1.154700538379252' '0.0' '-1.154700538379252']
 ['S' '0.5773502691896261' '0.5773502691896258' '-0.5773502691896261']
 ['P' '1.154700538379252' '0.0' '-1.154700538379252']
 ['F' '0.5773502691896261' '-0.5773502691896258' '-1.7320508075688776']
 ['R' '1.154700538379252' '-1.1547005383792517' '-2.3094010767585034']]
```

[20]:
```
fig = result.get_figure(title="Protein Structure of Bradykinin Peptide",␣
 ↪ticks=True, grid=True)
fig.get_axes()[0].view_init(20, 70)
```


Main Chain
Protein Structure of Bradykinin Peptide
R
B
P
F
R
G
0.0
−0.5
z −1.0
−1.5
−2.0
−1.00
−0.75
−0.50
−0.25
y 0.00
0.25
0.50
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
x

[ ]: