

```

126 from collections import deque
import pandas as pd
import plotly.graph_objs as go
import matplotlib.pyplot as plt

aa_df = pd.read_csv("../data/amino_acid_properties.csv")

def extract_hydropathy_list(dictionary):
    one_letter_code = dictionary['1-letter code'].values
    hydropathy_index = dictionary['hydropathy index (Kyte-Doolittle method)'].values
    hydropathy_list = dict(zip(one_letter_code, hydropathy_index))
    return hydropathy_list

hydropathy_list= extract_hydropathy_list(aa_df)

def extract_sequence(data):
    with open(data, 'r') as file:
        read_data = file.read()
        read_data_split = read_data.split('\n')
        sequences = ""
        for line in read_data_split:
            if not line.startswith('>'):
                sequences += line
    return sequences

sequence = list(extract_sequence('gpcr.fasta'))

def sequence_hydropathy(seq, dictionary):#
    hydropathy = []
    for aa in seq:
        for key in dictionary.keys():
            if aa == key:
                hydropathy.append(dictionary[key])
    return hydropathy

def window_hydropathy(seq, dictionary, len_window=None):
    hydropathy = []
    for aa in sequence:
        for key in dictionary.keys():
            if aa == key:
                hydropathy.append(dictionary[key])

    if len_window is not None:
        hydropathy_mean = []
        window = deque([], maxlen=len_window)
        for pos, aa in enumerate(hydropathy):
            window.append(aa)
            window_mean = sum(window) / len(window)
            hydropathy_mean.append(window_mean)
        hydropathy = hydropathy_mean
    return hydropathy

window_hydropathy_one = window_hydropathy(sequence, hydropathy_list, 1)
window_hydropathy_five = window_hydropathy(sequence, hydropathy_list, 5)
window_hydropathy_ten = window_hydropathy(sequence, hydropathy_list, 10)
window_hydropathy_fifteen = window_hydropathy(sequence, hydropathy_list, 15)

sequence_position = list(range(361))
aa_hydropathy_one = pd.DataFrame({'sequence position': sequence_position, 'hydropathy': window_hydropathy_one})
aa_hydropathy_five = pd.DataFrame({'sequence position': sequence_position, 'hydropathy': window_hydropathy_five})
aa_hydropathy_ten = pd.DataFrame({'sequence position': sequence_position, 'hydropathy': window_hydropathy_ten})
aa_hydropathy_fifteen = pd.DataFrame({'sequence position': sequence_position, 'hydropathy': window_hydropathy_fifteen})

data_one = [
    go.Bar(

```

```

        x=aa_hydrophathy_one["sequence position"],
        y=aa_hydrophathy_one['hydrophathy'],
    )
]

fig_one = go.Figure(data=data_one)
fig_one.update_layout(xaxis = dict(title = 'Sequence position'), yaxis = dict(title = 'Hydrophathy'),
fig_one.show()

data_five = [
    go.Bar(
        x=aa_hydrophathy_five["sequence position"],
        y=aa_hydrophathy_five['hydrophathy'],
    )
]

fig_five = go.Figure(data=data_five)
fig_five.update_layout(xaxis = dict(title = 'Sequence position'), yaxis = dict(title = 'Hydrophathy'),
fig_five.show()

data_ten = [
    go.Bar(
        x=aa_hydrophathy_ten["sequence position"],
        y=aa_hydrophathy_ten['hydrophathy'],
    )
]

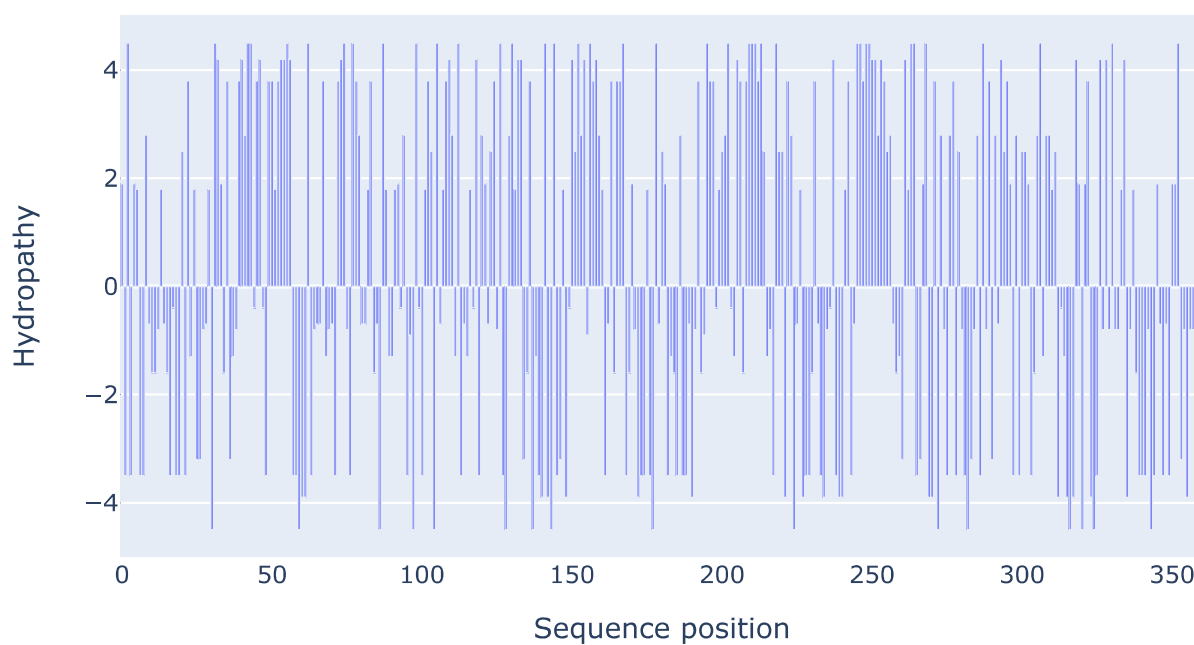
fig_ten = go.Figure(data=data_ten)
fig_ten.update_layout(xaxis = dict(title = 'Sequence position'), yaxis = dict(title = 'Hydrophathy'),
fig_ten.show()

data_fifteen = [
    go.Bar(
        x=aa_hydrophathy_fifteen["sequence position"],
        y=aa_hydrophathy_fifteen['hydrophathy'],
    )
]

fig_fifteen = go.Figure(data=data_fifteen)
fig_fifteen.update_layout(xaxis = dict(title = 'Sequence position'), yaxis = dict(title = 'Hydrophathy')
fig_fifteen.show()

```

Hydropathy of AAs in GPCR 183 with a window of 1



Hydropathy of AAs in GPCR 183 with a window of 5



Hydropathy of AAs in GPCR 183 with a window of 10



Hydropathy of AAs in GPCR 183 with a window of 15



