

Go

Protein module

```
needcoffee.protein.get_lookup_dict(aap_data)
    build dict out of aminoacid properties dataframe :param aap_data: :return:
    lookup_dict: holdes mapping dicts for diff properties
class needcoffee.protein.protein(protein_id, lookup_dict)
    The class protein allows for analysing and ... .. attribute:: protein_id

    identifying code used by uniprot to assign a specific pos to the protein that is
    being looked up

    type: str
lookup_dict
    dictionary providing the aminoacid to value references of the chosen properties

    Type: dict
sq
    protein sequence

    Type: str
def get_data():
    pulls sequence form uniprot based on the protein id
def map(self, prop):
    builds a list of values using the sequence and mapping dict to link sequence and
    values
def map_sld_wdw(self, prop, wd):
    builds a list of property values following the sequence using a sliding window
def plot(self, prop, sld=False, wd=None):
    plots a selected properties values against the sq of the protein that was created
    as instance the plot will be shown in a browser setting
get_data()
    [A] pulls sq form uniprot given an id name inspired by paul for prettyness
    :return: said proteins sq
map(prop)
    [B] builds a list of values using the sq and mapping dict to link sq and values
    :param prop: property to be analysed :return: prop_val_sq: property value sq
map_sld_wdw(prop, wd)
    [D] builds list of property values following sq using a sliding window :param
    prop: property to be analysed :param wd: width of analysing window :return:
plot(prop, sld=False, wd=None)
    plots a selected properties values against the sq of the protein that was created
    as instance the plot will be shown in a browser setting :param prop: name of the
    property of the aminoacid properties that we want to plot :param sld: do you
    want to use a sliding window approach please set this to True :param wd: width
    of the sliding window should you decide to go with said option :return:
```