Questions to explore NPac data set

Q1: Which type of transporter proteins are the most abundant?

Possible Approach/s:

* Once data is loaded into python, parse the file into the fields. Then parse target name by “.” Which would allow us to get V, w, x, y, & z alone. To get the transporter class abundance we can set 9 int counter (1 for each transpoerter class) and then iterate through the array of the V values (would have to convert from string to int) and append the corresponding int counter. From here we can make the int counters into an array and plot them. We can also parse on W and do the same int array and iteration process and plot to see the subclasses absence we have and then do the same process again for X but not sure it’s worth graphing X.

Expected Results:

A picture containing text, handwriting, diagram, font

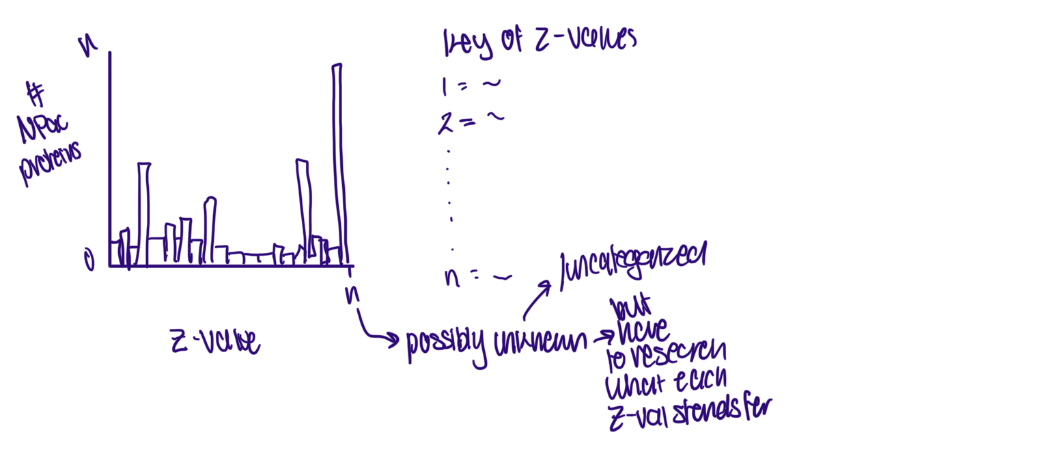
Description automatically generatedExpected Problems:

This part I feel like the problems will just be figuring out python packages and learning to graph bar graphs in python

Q2: Which substrate has the most transporters specified for it?

Possible approach: Parse the Target name on “.” And then make an int array and iterate through it and append to the corresponding int as we count each z-value. Then graph this array (col 1 is the Z value, col2 is amt of proteins in NPac)

Expected Results:



Expected Problems: Not enough research on transporters, so many will have uncategorized/unknown results. Distinction of substrate group vs substrate might have to be made. Also, different proteins might have different confidence levels for substrate specificity (eg some are experimentally verified while others are speculates)

Q3: How does transporter abundance change with depth &/or location?

Possible approach: need location or depth stamps for each sample; Then iterate through that file and match sequences that were matched (after parsing and storing query ID and depth/region fields in an array col 1 would be query ID and col 2 would be corresponding data). Then we need to iterate through matched HMMSearch results and add matched targets to a separate array consisting of 3 columns: 1- query ID, 2- loc/depth data, 3- transporter class. we can then plot the transporter classes for a range of locations/depths or plot

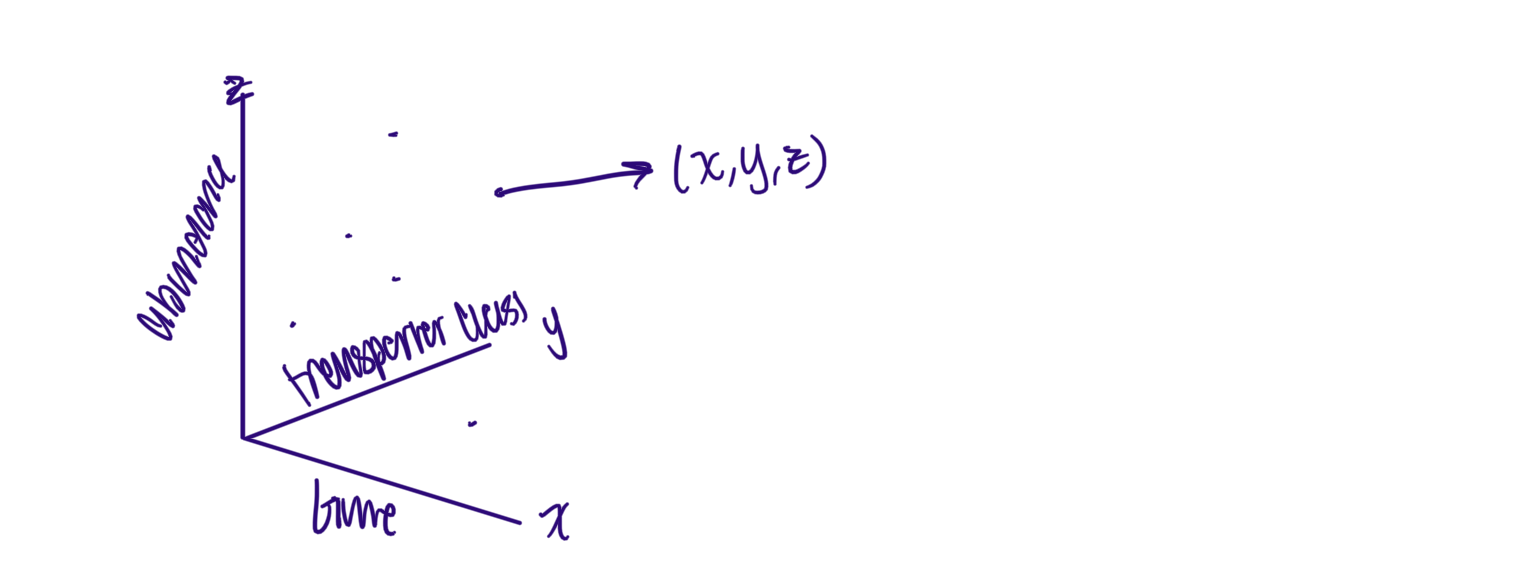
A picture containing text, handwriting, font, calligraphy

Description automatically generatedExpected Results:

Expected problems: getting the dataset from collection; matching a query ID to HMMSearch🡪 while HHMSearch output does have a query name this might not match dataset of collection’s query ID. Have to learn how to graph in 3D in python or see if another software does it better

Q4: How does time of day affect which transporters are activated? 🡪 not feasible but if it were we could potentially cross check if we had metabolite and substrate data from the same regions at different times

Possible approach: same approach as Q3 but now we would have time of day on X axis instead of location/depth

Expected Results:

Expected problems: might not be able to partition data on time of day if collection dataset does not have that; also, the collection is the sequence of ALL proteins so even if they are inactive their DNA will still be there so cannot determine this I believe.

Q5: what is the size distribution of the transporters? 🡪 Maybe not feasible if im understanding data output correctly

Possible approach: parse on the domain from and to fields and then create a resulting array where col 1 is domain size and col 2 is protein abundance.

Expected results:

A picture containing sketch, drawing, line art, child art

Description automatically generated

Expected problem: domain size is not always also the size of the protein; in fact I think its mostly not.

Notes on formatting of results

* HMMSearch has target name first and then query name
* HMMER doc recommends parsing based on space-delimited fields
* See pg 70/227 on HMMER user guide for more information
* Dombtblout: each line is a domain 🡪 NOT A PROTEIN... rather a functional/structural unit within a protein; can be proteins but usually are not 🡪 does this mean we have domain sequences that correspond to a certain protein but not necessarily proteins? Feel like that would affect abundance…
* Target name is in TC format
  + As per tcDoms site TC# is normally in V.W.X.Y.Z. format
    - V: # corresponding to transporter class (eg channel, carrier, etc)
    - W: letter corresponding to transporter subclass 🡪 more specific than transporter class (Eg: 1: Channel/pore 🡪 1.I: membrane-bounded channel)
    - X: # corresponding to transporter family (sometimes superfamily) (eg 1.I: membrane bounded 🡪 1.I.1 : Nuclear pore complex family)
    - Y: # corresponding to subfamily of transporter
    - Z: # corresponding to substate or substrate range
* Tlen: length of target profile residues
* Qlen: length of query sequence residues
* #: domains number (1…n dom)
* Of: total num of domains reported in sequence (ndom)