* Nucleotide sequence held in memory
  + Init it with something (maybe nothing)
  + Add a sequence – pass a name and a sequence
  + Add the data structure/dictionary that you init
  + Set up functions to be called individually later
* Translation table in the class already, decode the sequence
* Aacomp could pass the key (the name of the sample)
  + Return a dict with counts for one of the sequences (maybe 5 all separate)
  + Give you sequence with counts
* Substantiated sequence

Extra thoughts to make the process later in the code easier:

Init:

* Make a dictionary of dictionaries
  + Self.sequences – value being the sequence
  + Key is name of the sequence, value is the amino acid (count is value)
  + Add aacomp nuccomp and nuccount (all empty dictionaries) as values
    - Add sequence and add the data structure for the composition sections
    - Add a count to th