■,■,■ This is an info\_gain matrices for individual parameters

■■■ This is a sum of the individual parameter info\_gain matrices

■,■,■ This an info\_gain matrix for an individual parameter under different conditions

■■■ This is a sum of the individual parameter info\_gain matrices under different conditions

Below, this is a list: [ ■,■,■ ] and this is an array: [ ■ ■ ■ ]

Inside doMetropolisHastings():

self.info\_gain\_each\_parameter = self.info\_gain\_KL\_each\_parameter =[■,■,■]

self.info\_gain = self.info\_gain\_KL = [■■■]

Inside doeGetInfoGainMatrix(): it calls doMetropolisHastings a bunch of times. There is an array which is the conditions permutation, which I’ll call “c”.

conditionsPermutationAndInfoGain [Sum] = [cN,■■■]

self.info\_gain\_matrix = info\_gain\_matrix = **[** [c1,■■■] **|** [c2,■■■] **]**

self.info\_gain\_matrices\_each\_parameter = [ **[** [c1,■] [c2,■] **]** , **[** [c1,■] [c2,■] **]** , **[** [c1,■] [c2,■] **]** ]

Inside doeParameterModulationCombinationsScanner(): info\_gain\_matricies for different parameter chocies.

info\_gains\_matrices\_list = [ [ [c1,■■■] [c2,■■■] ] ,[ [c1,■■■] [c2,■■■] ] , [ [c1,■■■] [c2,■■■] ] ]

self.info\_gains\_matrices\_array = [ [ [c1,■■■] [c2,■■■] ] [ [c1,■■■] [c2,■■■] ] [ [c1,■■■] [c2,■■■] ] ]

self.info\_gains\_matrices\_arrays\_one\_for\_each\_parameter = [

[ **[** [c1,■] [c2,■] **]** **[** [c1,■] [c2,■] **]**  **[** [c1,■] [c2,■] **]** ] ,

[ **[** [c1,■] [c2,■] **]** **[** [c1,■] [c2,■] **]**  **[** [c1,■] [c2,■] **]** ],

[ **[** [c1,■] [c2,■] **]** **[** [c1,■] [c2,■] **]**  **[** [c1,■] [c2,■] **]** ] ]

Note: in the above structure, the colored brackets are modulations, the gray brackets are for individual parameters \*within\* each modulation.

Inside createInfoGainModulationPlots():

For the sums case:

local\_info\_gains\_matrices\_array=[ [ [c1,■■■] [c2,■■■] ] [ [c1,■■■] [c2,■■■] ] [ [c1,■■■] [c2,■■■] ]]

Then a loop goes across each modulation index (e.g., pink, cyan, blue) one at a time.

So for the each parameter case, we need to do this:

local\_info\_gains\_matrices\_array = np.array(info\_gains\_matrices\_arrays\_one\_for\_each\_parameter)[:][parameterIndex]

This pulls out a parameter’s info gain arrays for each modulation, and then the loop can go across each modulation index (e.g., pink, cyan, blue) one at a time, as before. Note that [:][parameterIndex] is correct and [:,parameterIndex] is not correct.