Supplementary Material

Modifiers of Somatic Repeat Instability in Mouse Models of Friedreich Ataxia and the Fragile X-Related Disorders: Implications for the Mechanism of Somatic Expansion in Huntington's Disease

Mathematical modelling of the differential effect of MutLα/PMS2 on repeat expansion in different diseases, disease models and cell types. A Python script was used to generate a pool of MutL complexes containing MutLα, MutLβ and MutLγ in various ratios that roughly reflect the relative proportions of PMS2, PMS1 and MLH3 in cells [1]. The number of expansion substrates available was then defined and each MutL complex in the pool was then randomly assigned to successive expansion substrates until either three MutLs were complexed on each substrate or the pool was exhausted. At that point bound MutL trimers containing MutLγ were scored as expansions. This process was repeated 1000 times and the average number of expansions calculated. The process was then repeated for increasing numbers of theoretical expansion substrates. The python script is shown in (A) and examples of the results obtained with different ratios of MutLα, MutL β and MutLγ in (B)

A) Python3 script:

```
# Assumptions
# Two MutS molecules must bind to a lesion but are not limiting here
# Three MutL molecules are required to bind to a lesion to resolve it, at least one must be Mlh3 for an expansion event, otherwise no expansion occurs
# The MutL proteins do not load onto a lesion simultaneously but sequentially, ie if there are four substrates a single MutL will load onto each
# prior to a second MutL loading onto the substrates.
# If insufficient MutS or MutL bind the lesion disappears without expanding
#
# Variables
# trialNum, each lesion value is tested trialNum times to smooth out the effects of random choice
# numPms2, numPms1, numMlh3 indicate how many molecules of each there are in the pot
```

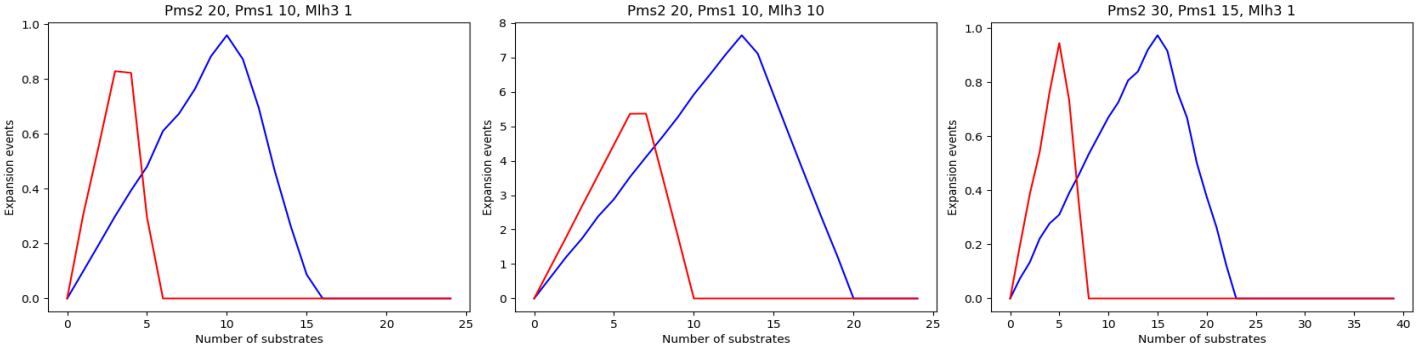
```
# numLes, incLes, finLes indicate the starting number of lesions, how the number of lesions is incremented and what the final number of lesions tested will be
# Note that large values of lesions will take a very long time to process.
#
# Running the script
# In the Terminal cd to the directory where this file is located
# Enter "python expansion_modeller_MutS.py"
# Each lesion value is tested trialNum times and the number of expansion events reported is cumulative not the value per trial
# Output is displayed in the Terminal and comprises Number of Lesions, Expansions in WT and Expansions in Pms2-null
# At completion a new text file is created containing the output data.

import random
import copy
#
# Variables
trialNum = 1000
numPms2 = 10
numPms1 = 5
numMlh3 = 2
numLes = 0
incLes = 1
finLes = 11
#
# Make the MutS molecules
MutS = ["Msh6" for i in range(10000)]
# MutS = []  # uncomment this for Msh6 null
numMsh3 = 1000
Msh3 = ["Msh3" for i in range(numMsh3)]
MutS.extend(Msh3)
#
# Make the MutL molecules
MutL = ["Pms2" for i in range(numPms2)]
Pms1 = ["Pms1" for i in range(numPms1)]
MutL.extend(Pms1)
Mlh3 = ["Mlh3" for i in range(numMlh3)]
MutL.extend(Mlh3)
# make the Pms2 null
noPms2 = []
noPms2.extend(Pms1)
noPms2.extend(Mlh3)
#
# Final output
expnOutput = {}
# Begin
```

```
while numLes < finLes:
    # populate the output counter
    expnOutput[numLes] = [0, 0]
    # Start trialNum trials of each lesion value
    for trial in range(trialNum):
        # reset the available MutS and MutL complexes
        muts = copy.deepcopy(MutS)
        mutl = copy.deepcopy(MutL)
        nop2 = copy.deepcopy(noPms2)
        # holders for assembled MutS and MutL complexes, one WT the other with no Pms2
        mutDict = {}
        mutDictnoP = {}
        for lesion in range(numLes):
            mutDict[lesion] = []
            mutDictnoP[lesion] = []
        # put MutS on each lesion such that all lesions get the first MutS and then get the second
MutS
        for protS in range(2):
            for lesion in range(numLes):
                if len(muts) > 0:
                    chosIdx = random.randint(0, (len(muts) -1))
                    addMutS = muts.pop(chosIdx)
                    mutDict[lesion].append(addMutS)
                    mutDictnoP[lesion].append(addMutS)
                else: pass
        # put MutL on each lesion that has two MutS present.
        for protL in range(3):
            for lesion in range(numLes):
                # first the WT
                if len(mutDict[lesion]) >= 2 and len(mutl) > 0:
                    chosIdx = random.randint(0, (len(mutl) -1))
                    mutDict[lesion].append(mutl.pop(chosIdx))
                else: pass
                # then the Pms2 null
                if len(mutDictnoP[lesion]) >= 2 and len(nop2) > 0:
                    chosIdx = random.randint(0, (len(nop2) -1))
                    mutDictnoP[lesion].append(nop2.pop(chosIdx))
                else: pass
        # count succesful expansions
        for lesion in range(numLes):
            if len(mutDict[lesion]) == 5 and "Mlh3" in mutDict[lesion]:
                expnOutput[numLes][0] +=1
            else: pass
            if len(mutDictnoP[lesion]) == 5 and "Mlh3" in mutDictnoP[lesion]:
                expnOutput[numLes][1] +=1
            else: pass
```

```
    print(numLes, expnOutput[numLes])
    # increment the number of lesions
    numLes += incLes
outfile = open("Expansion_results_WT_and_Pms2KO_MutS.txt", 'w')
print("NumLesions\tWTExpns\tPms2Expns", file=outfile)
for lesion in sorted(expnOutput):
    print(str(lesion) + "\t" + str(expnOutput[lesion][0]) + "\t" + str(expnOutput[lesion][1]),
file=outfile)
outfile.close()
```

B) Examples of script output obtained using the indicated ratios of MutL□/PMS1, MutLα/PMS2 and MutLγ/MLH3.



REFERENCES

[1] Cannavo E, Marra G, Sabates-Bellver J, Menigatti M, Lipkin SM, Fischer F, et al. Expression of the MutL homologue hMLH3 in human cells and its role in DNA mismatch repair. Cancer Res. 2005;65(23):10759-66.