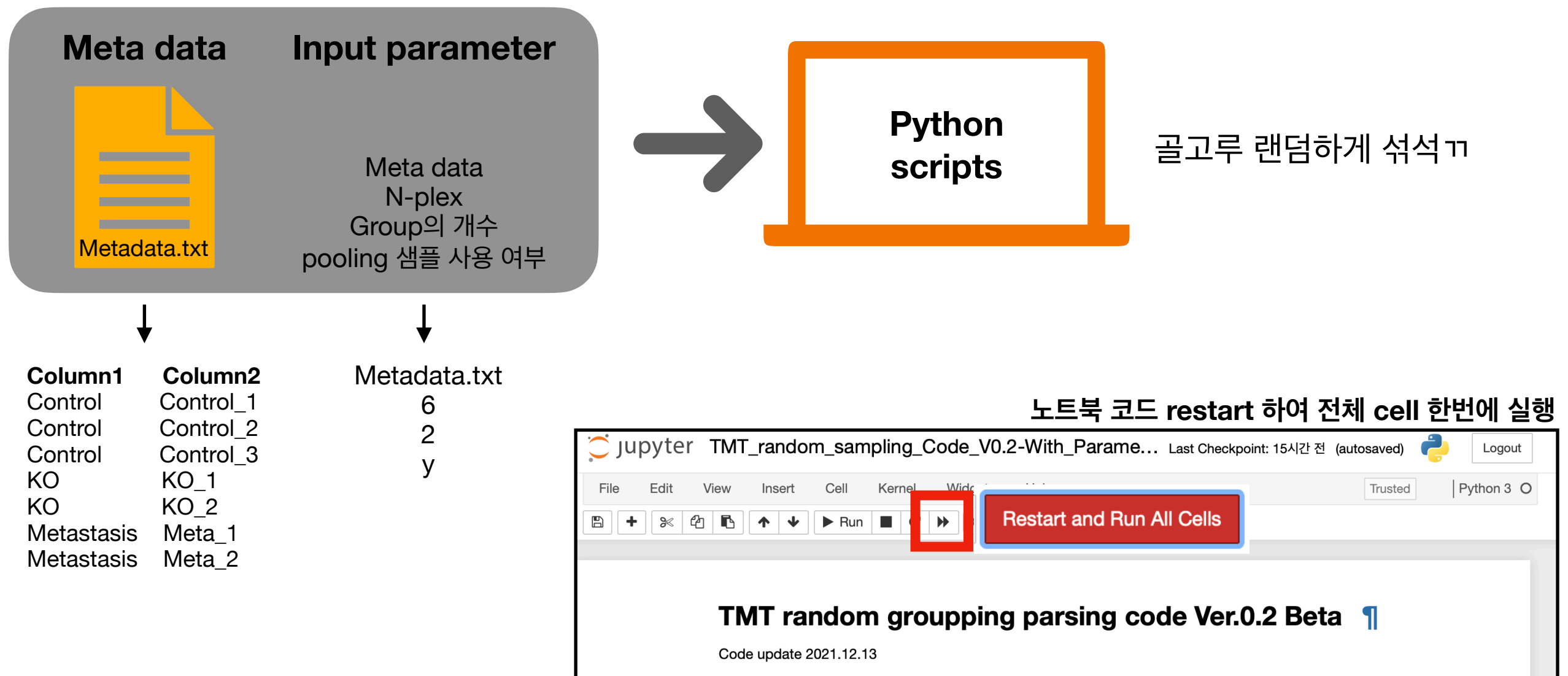


TMT labeling random sampling - Scheme

http://snuhcpl.asuscomm.com:2072/notebooks/Desktop/CSM/9.Parsing_Codes/TMT_RandomSampling_Code/TMT_random_sampling_Code_V0.2-With_Parameters.ipynb



TMT labeling random sampling - MetaData 준비

Conditions	Sample_label	Peptide_label	peptide_conc
MT	MT_1	MT2	0.22
MT	MT_2	MT3	0.26
MT	MT_3	MT4	0.24
MT	MT_4	MT7	0.26
MT	MT_5	MT9	0.29
MT	MT_6	MT10	0.31
MT	MT_7	MT11	0.31
MT	MT_8	MT12	0.33
MT	MT_9	MT14	0.28
MT	MT_10	MT16	0.27
MT	MT_11		
MT	MT_12		
MT	MT_13		
MT	MT_14		
MT	MT_15		
PM	PM_1		
PM	PM_2		
PM	PM_3		
PM	PM_4	PM6	0.26
PM	PM_5	PM7	0.27
PM	PM_6	PM8	0.27
PM	PM_7	PM9	0.26
PM	PM_8	PM10	0.27
PM	PM_9	PM12	0.27
PM	PM_10	PM13	0.26
PM	PM_11	PM14	0.29
PM	PM_12	PM16	0.3
PM	PM_13	PM17	0.28
NL	NL_1	NL2	0.28
NL	NL_2	NL3	0.29
NL	NL_3	NL5	0.28

... ..

Meta_data_TMTlabel.txt 파일 내용

MetaData 작성 시 주의 사항

1. 컬럼의 이름이 반드시 있어야 함, 컬럼 이름 안에는 공백이 없어야 함.
2. 1번 컬럼은 반드시 샘플 컨디션에 대한 정보여야 함.
3. 2번 컬럼은 반드시 샘플 고유 번호여야 함
4. 3번 컬럼부터 나머지는 자유롭게 사용해도 되며, 필요에 따라 다른 내용을 컬럼에 추가해도 됨

메타데이터에 pooling 샘플 넣을거면 이렇게 추가!

```
pool pool_1
pool pool_2
pool pool_3
```

...
...
...

TMT labeling random sampling - 파라미터 인풋 창

TMT random groupping parsing code Ver.0.2 Beta

Code update 2021.12.13

-- input parameters --

In [1]: MetaData = input("\nMetaData 파일 경로와 이름을 입력해 주세요: \n예제파일 이름: Meta_data_TMTlabel.txt")

MetaData 파일 경로와 이름을 입력해 주세요:
예제파일 이름: Meta_data_TMTlabel.txt

Meta_data_TMTlabel.txt

In [2]: NumberOfMultiplex = int(input("\nTMT labeling개수를 입력해 주세요, ex) 6 or 10 or 11: "))

TMT labeling개수를 입력해 주세요, ex) 6 or 10 or 11: 11

In [3]: NumberOfGroup = int(input("\n총 그룹의 개수는 몇개인가요? 숫자를 입력해주세요: "))

총 그룹의 개수는 몇개인가요? 숫자를 입력해주세요: 5

In [4]: pooling = input("\nPooling sample을 사용하시나요? \ny 를 선택하면 pooling 샘플을 제외한 상태로 그룹핑이 됩니")

Pooling sample을 사용하시나요?
y 를 선택하면 pooling 샘플을 제외한 상태로 그룹핑이 됩니다.
n를 선택하면 multiplex 개수를 꼭 채워서 그룹핑이 됩니다

만약 MetaData에 pooling 샘플 정보가 포함된 경우라면 n 를 선택해 주세요

Select y or n :y

TMT labeling random sampling - Script logic

Part 1.

```
import random, sys
Meta = open(MetaData, "r").read().splitlines() # Data input 하여 한줄당 하나씩 리스트로 반환

plex = NumberOfMultiplex #pooling 샘플 y면 그룹당 1개씩 풀링 샘플이 들어갈거니깐 plex 개수에서 한자리 비워두기
if pooling == "y":
    plex = plex - 1

def get_groups(): #컬럼 1번 정보를 빼와서 groups에 넣고, 중복 제거하여 group에 집어넣어 conditions 개수 확인
    groups=[]
    for line in Meta[1:len(Meta)]:
        groups.append(line.split()[0])
    group = list(set(groups))

    return([group, groups])
group = get_groups()[0]
groups = get_groups()[1]
#중복 제거한 그룹 목록 = 0번 (set(column1)), 중복 포함된 그룹 목록 = 1번 (list(column1))

def get_dict(): #Column2번을 Key로, Column 1번을 Value로 하는 딕셔너리 구현
    gorup_dic = {}
    for line in Meta[1:len(Meta)]:
        gorup_dic[str(line.split()[1])] = str(line.split()[0])
    return gorup_dic
dic = get_dict()
# sample label : group label 로 정리된 딕셔너리 구현

def get_Meta(): #Column 2번을 키로, 해당 샘플의 모든 정보를 value로 하는 딕셔너리 구현 - 최종 출력에 활용
    dic = {}
    for line in Meta[1:len(Meta)]:
        dic[str(line.split()[1])] = line
    return dic
dic_Meta = get_Meta()
#sample label : line 으로 정리된 딕셔너리 구현
```

Parameter 값 기반으로
input file 정보 받아 입력 제어

필요한 함수 구현

if pooling parameter :y

한개의 그룹에 pooling 샘플 최소 1개는
들어 가야 하니깐 한자리 미리 비워둠

TMT labeling random sampling - Script logic

Part 2.

```
samplelist_by_groups = []
for i in range(0, len(group)):
    sample_set = []
    for key, value in dic.items():
        if value == group[i]:
            sample_set.append(key)
    samplelist_by_groups.append(sample_set)
for i in range(0, len(samplelist_by_groups)):
    a = random.shuffle(samplelist_by_groups[i])
# 각 그룹별로 리스트 되어있고, 순서가 random으로 섞인 이중 array 구현.

num_of_TMT_set = NumberOfGroup # input option
random_samples = []
for i in range(0, num_of_TMT_set):
    sample_set = []
    for i in range(0, len(group)):
        try:
            sample_set.append(samplelist_by_groups[i].pop())
        except:
            pass

    random_samples.append(sample_set)
#TMT group 개수에 따라 각각 하나씩 집어넣은 목록 정리. 한 샘플이 한쪽에 쏠리지 않도록 구현

other_samples = sum(samplelist_by_groups, [])

for i in range(0, 10):
    random.shuffle(other_samples) # 10 time shuffling

# 남은것들 한개의 리스트로 묶어주고, 랜덤으로 셔플링 하기
for i in range(0, len(random_samples)):
    while True:
        try:
            if plex > len(random_samples[i]):
                random_samples[i].append(other_samples.pop())
            else:
                break
        except: break
```

Metadata 정보를 컨디션 별로 쪼개기



이래서 Column1번 정보 반드시 샘플의 컨디션 정보여야 함



각 컨디션 안에서 샘플간 순서 섞어줌

앞에서 받은 그룹 정보로 총 몇개의 그룹으로 쪼갤지 정보를 받아서
한 그룹당 각 컨디션이 최소 1개씩 들어가도록 샘플 어레이인지
그룹은 5개인데, 그중 한 컨디션에 샘플이 4개면 4개까지만 들어감

단. 각 컨디션 내 샘플 개수가 전부, 나눌 그룹 개수보다 적으면 문제 생김

그룹당 한개씩 뽑아내고 남은것들 모아서
랜덤으로 10번 섞어준 뒤에
각 그룹에 하나씩 채워줌

TMT labeling random sampling - Script logic

Part 3.

만들어둔 그룹 당 N-plex 개수까지 채워 넣었는데도 샘플이 남으면

코드 중지 & N 개의 샘플이 남았는데, 총 샘플 개수와 TMT-Xplex, 그룹 개수를 다시 보고, 몇개로 설정할지 파라미터를 다시 설정해야 함

```
if len(other_samples) != 0:  
    sys.exit("포함되지 않는 샘플이 " + str(len(other_samples)) + "개 있습니다. 총 샘플 개수와 TMT-Xplex, 그룹 개수를 다시 확인해 주세요")
```

Metadata에 48개의 샘플이 있는데, 6-plex, 2 group, pooling=n 옵션 선택했을때, 36개가 남는 경우 아래의 에러 출력 & 코드 멈춤

An exception has occurred, use %tb to see the full traceback.

SystemExit: 포함되지 않는 샘플이 36개 있습니다. 총 샘플 개수와 TMT-Xplex, 그룹 개수를 다시 확인해 주세요

```
C:\ProgramData\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3445: UserWarning: To exit: use 'exit',  
'quit', or Ctrl-D.  
    warn("To exit: use 'exit', 'quit', or Ctrl-D.", stacklevel=1)
```

TMT labeling random sampling - Script logic

Part 4.

```
#--- pooling y or n 에 따라 결과 값에 pooling 샘플 추가 ---#

#pooling == y 이면, NumOfPlex * NumOfGroup - 총 샘플 개수 = pooling 샘플 개수
if pooling == "y":
    numof_pooling_sample = (NumberOfMultiplex*NumberOfGroup) - len(groups)
    pool = list(range(1,numof_pooling_sample+1))
    pool.reverse()
    for i in range(0, len(random_samples)):
        print("TMT group_",i+1,sep="")
        print(Meta[0])
        for j in range(len(random_samples[i])):
            print(dic_Meta[random_samples[i][j]])
        for p in range(NumberOfMultiplex - len(random_samples[i])):
            pool_num = pool.pop()
            print(f"Pool_{pool_num}\tpooling\tpool_{pool_num}")
        print('\n')
else:
    for i in range(0, len(random_samples)):
        print("TMT group ",i+1,sep="")
        print(Meta[0])
        for j in range(len(random_samples[i])):
            print(dic_Meta[random_samples[i][j]])
        print('\n')
```

48샘플
11-plex
5개 그룹

출력 제어

TMT group 1			
Group	Sample_label	Peptide_label	peptide_conc
PM	PM_5	PM7	0.27
NL	NL_1	NL2	0.28
MT	MT_7	MT11	0.31
RM	RM_3	RM8	0.33
PT	PT_2	PT3	0.28
PM	PM_11	PM14	0.29
MT	MT_14	MT20	0.29
PT	PT_3	PT4	0.28
PM	PM_9	PM12	0.27
PT	PT_1	PT1	0.31
QC	pooling	pool_1	

11sample
per groups

TMT group 5			
Group	Sample_label	Peptide_label	peptide_conc
PM	PM_1	PM3	0.27
NL	NL_4	NL6	0.29
MT	MT_4	MT7	0.26
PT	PT_9	PT12	0.25
PT	PT_11	PT18	0.26
MT	MT_3	MT4	0.24
PM	PM_12	PM16	0.30
PM	PM_3	PM5	0.26
QC	pooling	pool_5	
QC	pooling	pool_6	
QC	pooling	pool_7	

pooling = y 했을때

TMT group 4			
Group	Sample_label	Peptide_label	peptide_conc
PM	PM_10	PM13	0.26
NL	NL_2	NL3	0.29
MT	MT_7	MT11	0.31
RM	RM_2	RM4	0.29
PT	PT_4	PT5	0.26
MT	MT_13	MT19	0.24
PT	PT_2	PT3	0.28
MT	MT_11	MT17	0.27
MT	MT_10	MT16	0.27
PM	PM_1	PM3	0.27
PM	PM_2	PM4	0.24

TMT group 5			
Group	Sample_label	Peptide_label	peptide_conc
PM	PM_8	PM10	0.27
NL	NL_3	NL5	0.28
MT	MT_15	MT21	0.27
PT	PT_3	PT4	0.28

pooling = n

TMT labeling random sampling - Excel processing

Option 입력창	숫자 꼭 써주세요
Amount of Peptide(ug) =	20
Final peptide volume (ul, include OV)	120
Original OV conc.	0.26 ug /ul
Table 설명표 실험과정 헛갈리지 말라고 step별로 색상 구분	
Copy from python scripts	
Adding your tube - step 1	
Adding your tube - step 2	
Final volume (Af. sample Pooling)	

Step 2.

사용할 peptide 양 및 최종 volume 정보만 정해주면

샘플당 peptide volume 계산 및 Adding buffer 양 계산해줌

TMT_group_1																
Group	Sample_label	Peptide_label	peptide_conc	Mass Tag	Plex_number	Peptide_Volume (ul)	OV+Buffer (1:9 mix) ul	Final Volume (ul)	Add buffer (ul)	ACN (ul)	TMT (ul)	Final Total volume	ACN % 25 ~ 30%	Quanching solution (ul)		Total mixed volume (ul)
PM	PM_8	PM10	0.27	TMT10-126	1	74.1	10	120	35.9	30	20	170	29.41	10.2	180.2	1982.2
MT	MT_2	MT3	0.26	TMT10-127N	2	76.9	10	120	33.1	30	20	170	29.41	10.2	180.2	
NL	NL_5	NL7	0.3	TMT10-127C	3	66.7	10	120	43.3	30	20	170	29.41	10.2	180.2	
PT	PT_7	PT10	0.19	TMT10-128N	4	105.3	10	120	4.7	30	20	170	29.41	10.2	180.2	
RM	RM_1	RM2	0.27	TMT10-128C	5	74.1	10	120	35.9	30	20	170	29.41	10.2	180.2	
PT	PT_9	PT12	0.25	TMT10-129N	6	80.0	10	120	30.0	30	20	170	29.41	10.2	180.2	
PT	PT_5	PT6	0.28	TMT10-129C	7	71.4	10	120	38.6	30	20	170	29.41	10.2	180.2	
MT	MT_8	MT12	0.33	TMT10-130N	8	60.6	10	120	49.4	30	20	170	29.41	10.2	180.2	
PM	PM_3	PM5	0.26	TMT10-130C	9	76.9	10	120	33.1	30	20	170	29.41	10.2	180.2	
PM	PM_5	PM7	0.27	TMT10-131	10	74.1	10	120	35.9	30	20	170	29.41	10.2	180.2	
Pooling				TMT10-131C	11	#DIV/0!	10	120	#DIV/0!	30	20	170	29.41	10.2	180.2	

파이썬 결과에서 그대로 복붙

Step 1.

4번째 컬럼만 peptide conc. 정보면 됨

TMT labeling random sampling - Excel processing

http://snuhcpl.asuscomm.com:2072/notebooks/Desktop/CSM/9.Parsing_Codes/TMT_RandomSampling_Code

Option 입력창	숫자 꼭 써주세요
Amount of Peptide(ug) =	20
Final peptide volume (ul, include OV)	100

만약 peptide 농도가 낮아서 최종 볼륨을 초과해 버리면?

Red color labeling

TMT group_1																
Group	Sample_label	Peptide_label	peptide_conc	Mass Tag	Plex_number	Peptide_Volume (ul)	OV+Buffer (1:9 mix) ul	Final Volume (ul)	Add buffer (ul)	ACN (ul)	TMT (ul)	Final Total volume	ACN % 25 ~ 30%	Quanching solution (ul)		Total mixed volume (ul)
PM	PM_8	PM10	0.27	TMT10-126	1	74.1	10	100	15.9	30	20	150	33.33	9	159	1749
MT	MT_2	MT3	0.26	TMT10-127N	2	76.9	10	100	13.1	30	20	150	33.33	9	159	
NL	NL_5	NL7	0.3	TMT10-127C	3	66.7	10	100	23.3	30	20	150	33.33	9	159	
PT	PT_7	PT10	0.19	TMT10-128N	4	105.3	10	100	-15.3	30	20	150	33.33	9	159	
RM	RM_1	RM2	0.27	TMT10-128C	5	74.1	10	100	15.9	30	20	150	33.33	9	159	
PT	PT_9	PT12	0.25	TMT10-129N	6	80.0	10	100	10.0	30	20	150	33.33	9	159	
PT	PT_5	PT6	0.28	TMT10-129C	7	71.4	10	100	18.6	30	20	150	33.33	9	159	
MT	MT_8	MT12	0.33	TMT10-130N	8	60.6	10	100	29.4	30	20	150	33.33	9	159	
PM	PM_3	PM5	0.26	TMT10-130C	9	76.9	10	100	13.1	30	20	150	33.33	9	159	
PM	PM_5	PM7	0.27	TMT10-131	10	74.1	10	100	15.9	30	20	150	33.33	9	159	
Pooling				TMT10-131C	11	#DIV/0!	10	100	#DIV/0!	30	20	150	33.33	9	159	