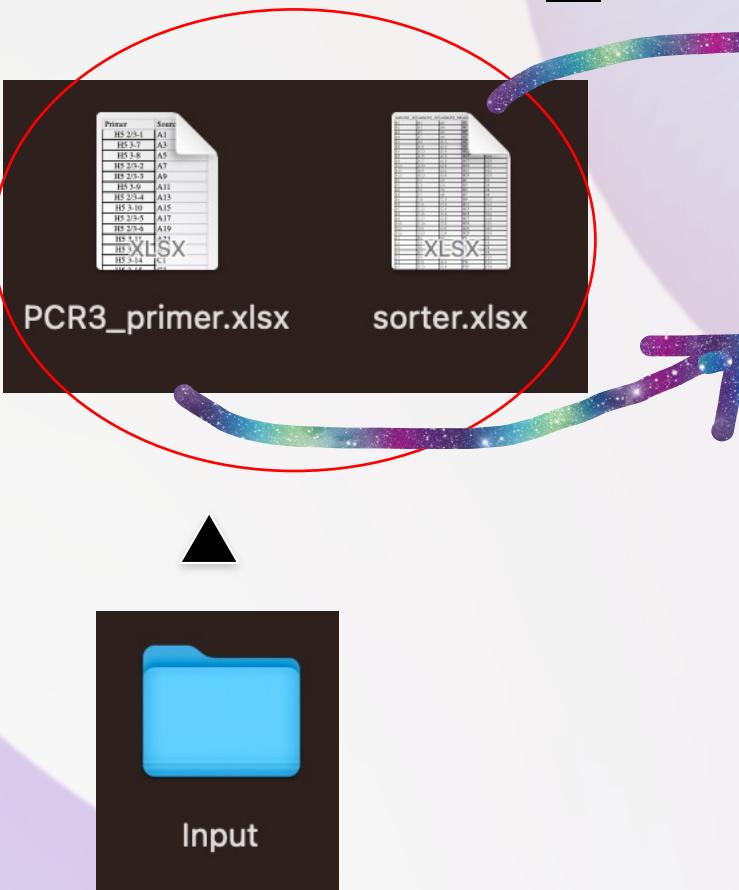




05_PCR3.py: input files



	A	B
1	Primer	Source Well
2	H5 2/3-1	A1
3	H5 3-7	A3
4	H5 3-8	A5
5	H5 2/3-2	A7
6	H5 2/3-3	A9
7	H5 3-9	A11
8	H5 2/3-4	A13
9	H5 3-10	A15
10	H5 2/3-5	A17
11	H5 2/3-6	A19
12	H5 3-11	A21

	A	B	C	D	E
	sorterA1_A2	sorterA1_A3	sorterA2_A4	sorterB1_B3	sorterB2_B4
A1	A1	A2		B1	B2
A2	A3	A4		B3	B4
A3	A5	A6		B5	B6
A4	A7	A8		B7	B8
A5	A9	A10		B9	B10
A6	A11	A12		B11	B12
A7	A13	A14		B13	B14
A8	A15	A16		B15	B16
A9	A17	A18		B17	B18
A10	A19	A20		B19	B20
A11	A21	A22		B21	B22
A12	A23	A24		B23	B24
B1	C1	C2		D1	D2
B2	C3	C4		D3	D4





05_PCR3.py: input files



LATEST PCR3_StoreList.xlsx (modification date)



File provided by Knime Script

A	B	C	D	E	F	G	H	I
cDNA_Barcode	cDNA_Well	chain	PCR1_copy_Barcode	PCR1_copy_Well	Seq1_Name	Primer 5'	Primer 3'	_2_PCR_to_CELE_S::PCR_to_CELE_ID
BAOscDNAp00090	B3	L	BAOsP1COp00104	C6	Mock_Barcod1_D12	L5 2/3-6	L3 2/3	4A50EBA3-1183-4944-85C7-BC90BBA3E5A8
BAOscDNAp00090	B4	H	BAOsP1COp00104	C7	Mock_Barcod1_E12	H5 3-10	H3 3-1	55B3B142-66D0-374C-A0C5-9D51318C2887
BAOscDNAp00090	B4	K	BAOsP1COp00104	D8	Mock_Barcod1_F12	K5 3-1	K3 3-1	55B3B142-66D0-374C-A0C5-9D51318C2887
BAOscDNAp00090	B5	H	BAOsP1COp00104	C9	Mock_Barcod1_G12	H5 2/3-4	H3 3-2	2D827BE9-7FF2-1646-814F-B4B56D718511
BAOscDNAp00090	B5	I	BAOsP1COp00104	C10	Mock_Barcod1_H12	I5 2/2-2	I3 2/2	2D827BE9-7FF2-1646-814F-B4B56D718511



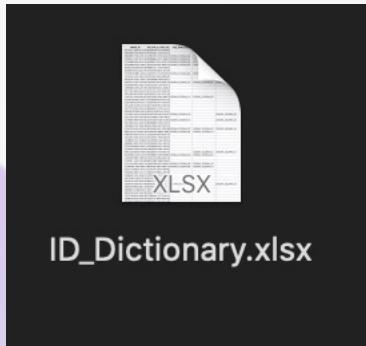


05 _PCR3.py: input files

File provided by Knime Script



FK_PCR_to_CELE_ID	aBASE_ID	Seq1_Name_heavy	Seq1_Name_kappa	Seq1_Name_Lambda
D6D680D0-0234-4D9F-B3AB-1967B11E12EA	9261F02A-F4B6-4CA5-8D8A-56A6AA95FEB3			
FA3E48DA-7275-46AA-B864-E310CE7435B4	F9FA4B3F-C828-4E5A-93EC-E810BC4E28B5			
A7B913E7-3CC5-49AD-B3C3-C3565985EEC9	1EB26BF1-372D-475C-855E-EE18C8E82589	Mock_Barcod55_D1	Mock_Barcod55_E1	
E78BBBF0-AC71-45B2-84F4-B71077DAC8EF	D24D2941-3282-4473-AE69-8238B67BE187	Mock_Barcod55_F1	Mock_Barcod55_G1	
26EBB550-F019-422E-AF69-1E4BE8F89C4E	57C60BB2-9086-487A-866D-A9CABAFCDD96	Mock_Barcod55_H1	Mock_Barcod55_A2	
5FE44C00-0043-4BB5-915D-5E9E97FF0CAF	0D705DD0-D1EC-4880-9392-1792D23A33C3	Mock_Barcod55_B2	Mock_Barcod55_C2	Mock_Barcod55_D2
006BCC00-77A0-4A06-B1FC-9E8C6BBEC197	D83D517E-E24E-43E7-8B85-2CF75CEC1F61			
3357E000-8EA4-4203-8B8E-4B6BC2DBA36C	6E6DB064-02D2-40C4-90AB-3F6CDE2EB753			





05_PCR3.py.py: run

usage: 05_PCR3.py.py **-s** SPECPCRPLATE **-e** CELE_PLATE **-c** SPECPCR_COPY_PLATE

arguments:

-s SPECPCRPLATE, Latest specPCR plate number
-e CELE_PLATE, Latest cele plate number
-c SPECPCR_COPY_PLATE, Latest specPCR copy plate number

example run:

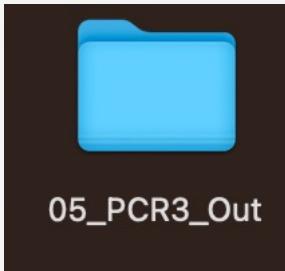
in terminal, from script folder, type:

python 05_PCR3.py.py -s 40 -e 20 -c 60





05_PCR3.py: output



PCR3_Intermed_date-time.xlsx

aBASE_ID	chain	SpecPCR_Barcod	SpecPCR_Well	SpecPCR_copy_Barcod	SpecPCR_copy_Well	Cele_SpecPCR_Barcod	SpecPCR_Well
65747025-5AB6-B541-9F34-4A58AA21B157	H	BAOsPCR3p00059	A1	BAOsP3COp00028	A1	BAOsCELEp00380	A1
65747025-5AB6-B541-9F34-4A58AA21B157	K	BAOsPCR3p00059	A2	BAOsP3COp00028	A3	BAOsCELEp00380	A2
65747025-5AB6-B541-9F34-4A58AA21B157	L	BAOsPCR3p00059	A3	BAOsP3COp00028	A5	BAOsCELEp00380	A3
9A8D6FD2-AE43-3547-89F7-B6E05CBAF36C	H	BAOsPCR3p00059	A4	BAOsP3COp00028	A7	BAOsCELEp00380	A4
9A8D6FD2-AE43-3547-89F7-B6E05CBAF36C	K	BAOsPCR3p00059	A5	BAOsP3COp00028	A9	BAOsCELEp00380	A5

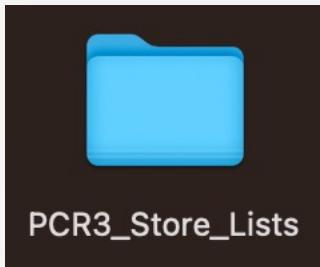
BAOsPCR3p000XX_PrimerPickList

BAOsPCR3p000XX_TemplatePickList





05_PCR3.py: output



NEW PCR3_StoreList_Date-time.xlsx

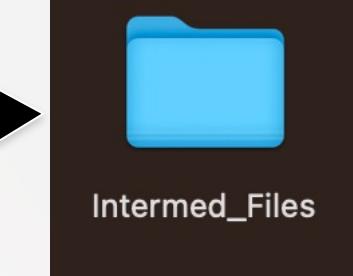
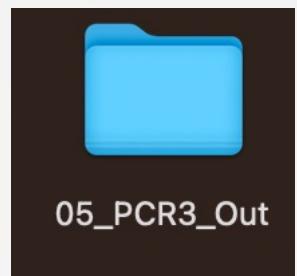


cDNA_Barcode	cDNA_Well	chain	PCR1_copy_Barcode	PCR1_copy_Well	Seq1_Name	Primer 5'	Primer 3'	_2_PCR_to_CELE_S::PCR_to_CELE_ID
BAOscDNAp00090	B3	L	BAOsP1COp00104	C6	Mock_Barcode1_D12	L5 2/3-6	L3 2/3	4A50EBA3-1183-4944-85C7-BC90BBA3E5A8
BAOscDNAp00090	B4	H	BAOsP1COp00104	C7	Mock_Barcode1_E12	H5 3-10	H3 3-1	55B3B142-66D0-374C-A0C5-9D51318C2887
BAOscDNAp00090	B4	K	BAOsP1COp00104	D8	Mock_Barcode1_F12	K5 3-1	K3 3-1	55B3B142-66D0-374C-A0C5-9D51318C2887
BAOscDNAp00090	B5	H	BAOsP1COp00104	C9	Mock_Barcode1_G12	H5 2/3-4	H3 3-2	2D827BE9-7FF2-1646-814F-B4B56D718511
BAOscDNAp00090	B5	L	BAOsP1COp00104	C10	Mock_Barcode1_H12	L5 2/3-3	L3 2/3	2D827BE9-7FF2-1646-814F-B4B56D718511
BAOscDNAp00090	B6	H	BAOsP1COp00104	C11	Mock_Barcode2_A1	H5 2/3-5	H3 3-1	1022FA0B-EFC6-D945-A638-3A65763CF22A

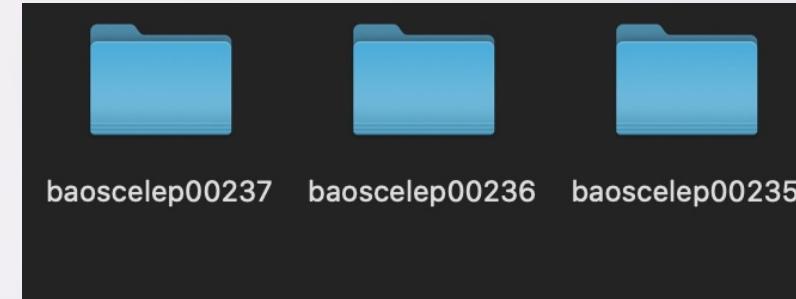
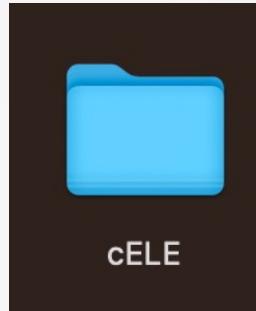
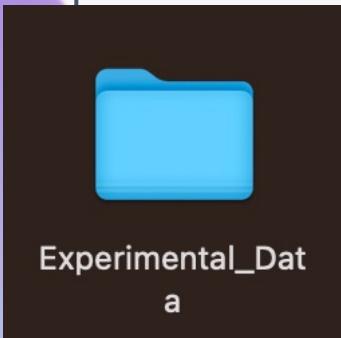
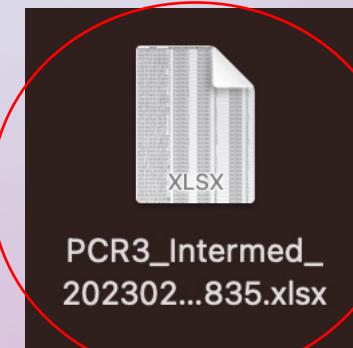




06_cELE2.py: input files



LATEST specPCR_partial_import_date-time.xlsx



Must contain
peak tables,
name=barcode





06_cELE2.py: run



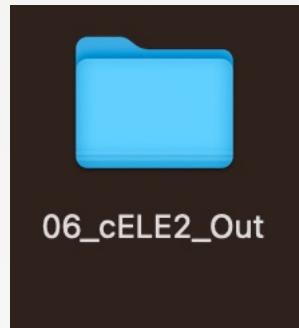
example run:

in terminal, from script folder, type:

python 06_cELE2.py



06_cELE2.py: output



aBASE_ID	chain	SpecPCR_Barcode	SpecPCR_Well
8B1069DB-CEF3-4345-ADCF-8A44D1DAB5FC	L	BAOsPCR3p00035	A1
C4C7234C-B8FF-47A4-9694-1C40213D0A6D	H	BAOsPCR3p00035	A2
C4C7234C-B8FF-47A4-9694-1C40213D0A6D	K	BAOsPCR3p00035	A3
88B52BB4-738B-4D5F-985F-E0E6567E04DB	H	BAOsPCR3p00035	A4
88B52BB4-738B-4D5F-985F-E0E6567E04DB	K	BAOsPCR3p00035	A5
42563430-99F1-4AC1-9D76-6AD05F4BA336	H	BAOsPCR3p00035	A6
42563430-99F1-4AC1-9D76-6AD05F4BA336	K	BAOsPCR3p00035	A7
42563430-99F1-4AC1-9D76-6AD05F4BA336	L	BAOsPCR3p00035	A8
D9E48CA1-75A1-4A2C-AB22-1107ADE034AB	H	BAOsPCR3p00035	A9
D9E48CA1-75A1-4A2C-AB22-1107ADE034AB	L	BAOsPCR3p00035	A10
5E9FC4C6-E306-4CC8-9783-1D02564601BC	H	BAOsPCR3p00035	A11
5E9FC4C6-E306-4CC8-9783-1D02564601BC	L	BAOsPCR3p00035	A12
E24B9B33-58D3-46B1-930A-A60644E3B0D2	H	BAOsPCR3p00035	B1
E24B9B33-58D3-46B1-930A-A60644E3B0D2	L	BAOsPCR3p00035	B2
A2BE54A9-B45A-40BB-BE1F-C6FBEB12CF5A	H	BAOsPCR3p00035	B3
A2BE54A9-B45A-40BB-BE1F-C6FBEB12CF5A	K	BAOsPCR3p00035	B4

1

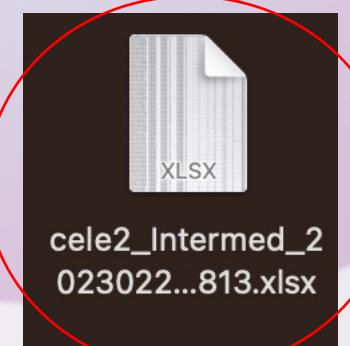




07_GiAS.py: input files



LATEST cele2_Intermed_date-time.xlsx





07_GiAS.py: run



usage: 07_GiAS.py -g GIBSON_PLT

arguments:

-g GIBSON_PLT,

Latest gibson plate number

example run:

in terminal, from script folder, type:

python 02_Gibson.py -g 45





07 _GiAS.py: user input



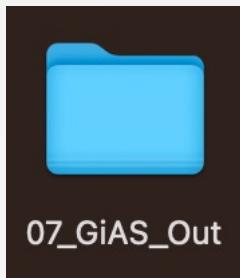
Example user input

```
Number of Gibson Plates: 3
Enter heavy, kappa, lambda plasmid wells, separated by comma for plate 1...
E2,F2,G2
Enter MM wells, separated by comma for plate 1...
A2,B2,C2
Enter heavy, kappa, lambda plasmid wells, separated by comma for plate 2...
E4,F4,G4
Enter MM wells, separated by comma for plate 2...
A4,B4,C4
Enter heavy, kappa, lambda plasmid wells, separated by comma for plate 3...
E6,F6,G6
Enter MM wells, separated by comma for plate 3...
A6,B6,C6
(base) BN-M-3016768dzne:Wrk_Flowpy_Scripts kotowiczm$
```

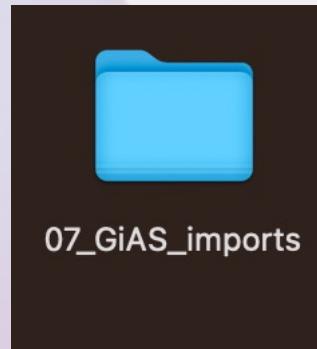




07_GiAS.py: output



07_GiAS_Out



07_GiAS_imports



GiAS_import_date-time.xlsx

BAOsGiAsp000XX_Plating_Spotting.xlsx
BAOsGiAsp000XX_MM_Spotting.xlsx
BAOsGiAsp000XX_PCR_Spotting.xlsx

aBASE_ID	chain	SpecPCR_Barcode	SpecPCR_Well
8B1069DB-CEF3-4345-ADCF-8A44D1DAB5FC	L	BAOsPCR3p00035	A1
C4C7234C-B8FF-47A4-9694-1C40213D0A6D	H	BAOsPCR3p00035	A2
C4C7234C-B8FF-47A4-9694-1C40213D0A6D	K	BAOsPCR3p00035	A3
88B52BB4-738B-4D5F-985F-E0E6567E04DB	H	BAOsPCR3p00035	A4
88B52BB4-738B-4D5F-985F-E0E6567E04DB	K	BAOsPCR3p00035	A5
42563430-99F1-4AC1-9D76-6AD05F4BA336	H	BAOsPCR3p00035	A6
42563430-99F1-4AC1-9D76-6AD05F4BA336	K	BAOsPCR3p00035	A7
42563430-99F1-4AC1-9D76-6AD05F4BA336	L	BAOsPCR3p00035	A8

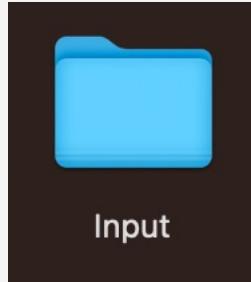
...

GiAS_Barcode	GiAS_Well
BAOsGiAsp00078	A1
BAOsGiAsp00078	A2
BAOsGiAsp00078	A3
BAOsGiAsp00078	A4
BAOsGiAsp00078	A5
BAOsGiAsp00078	A6
BAOsGiAsp00078	A7
BAOsGiAsp00078	A8

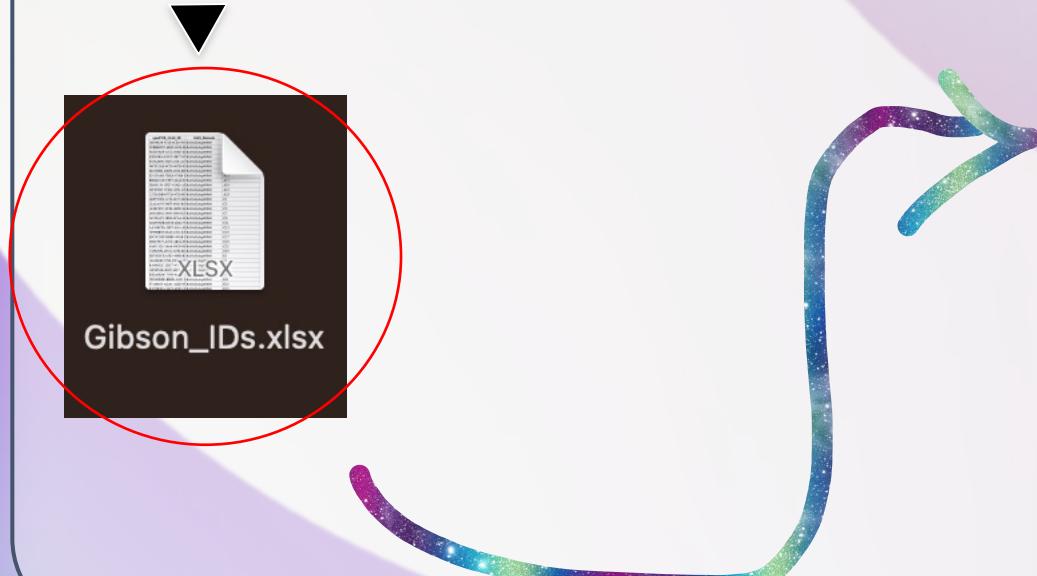




08_PLA.py: input files



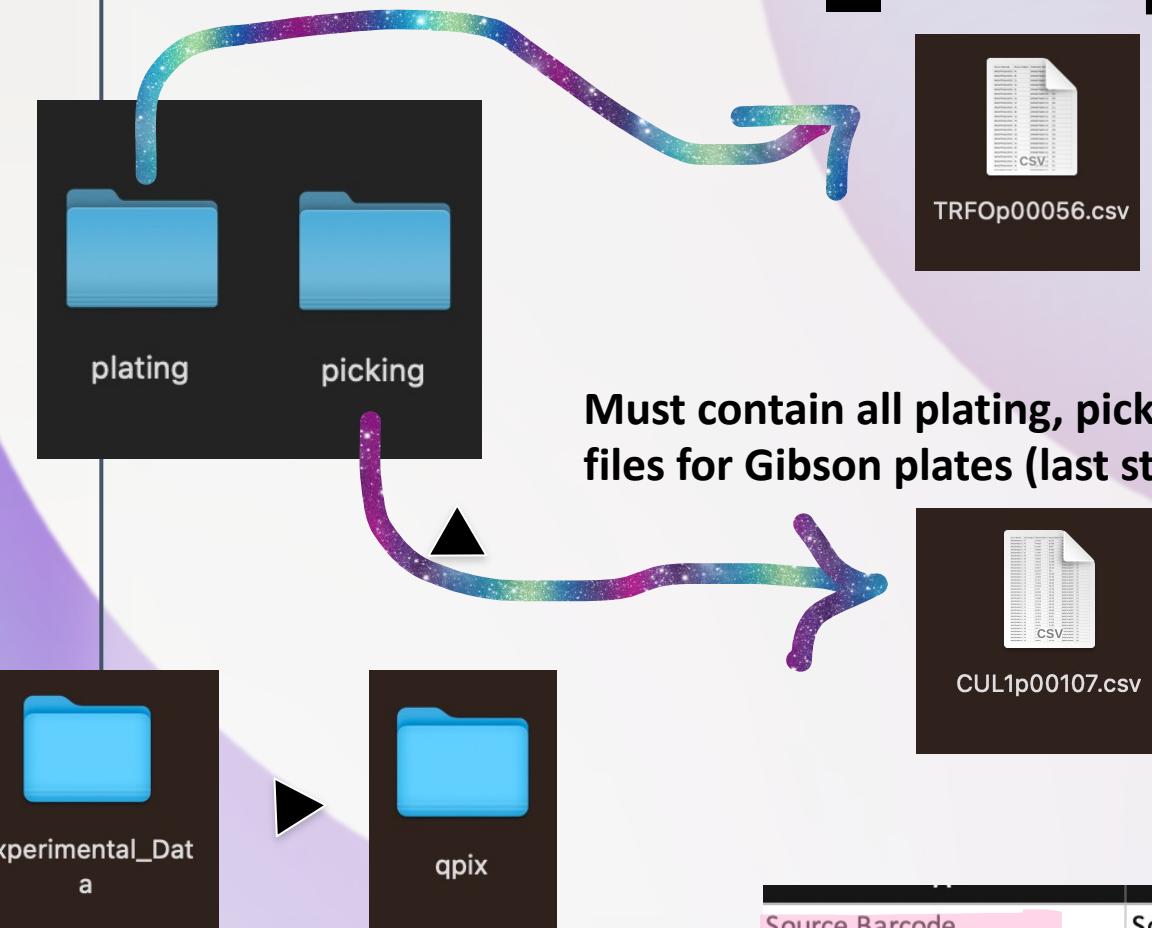
File provided by FileMaker (DB)
script



specPCR_GiAS_ID	GiAS_Barcod	GiAS_Well
5DA2B3FD-530A-4486-868	BAOsGiAsp00026	A1
2343E1BE-029C-475F-B195	BAOsGiAsp00026	A3
D77E74BC-76F1-4F14-8683	BAOsGiAsp00026	A5
4BE3B516-F0DA-4DFB-B73	BAOsGiAsp00026	A7
BE2D7E34-C9C9-4A29-A2E	BAOsGiAsp00026	A9
A3858841-8DB9-4D88-996	BAOsGiAsp00026	A11
FCE44CD8-88D2-40BE-B181	BAOsGiAsp00026	A13
98E64650-D572-4E7C-A081	BAOsGiAsp00026	A15
2C036C85-44AC-4369-99B1	BAOsGiAsp00026	A17
70C81E52-71CF-48B0-A0C1	BAOsGiAsp00026	A19
5AF54659-F611-42F8-B331	BAOsGiAsp00026	A21



08_PLA.py: input files



Source Barcode	Source Region	Destination Barcode	Destination Region
Auto6378667015064533003	A1	Auto6378675383150341000	A1
Auto6378667015064533003	B1	Auto6378675383150341000	A2
Auto6378667015064533003	C1	Auto6378675383150341000	A3
Auto6378667015064533003	D1	Auto6378675383150341000	A4
Auto6378667015064533003	E1	Auto6378675383150341000	B1
Auto6378667015064533003	F1	Auto6378675383150341000	B2
Auto6378667015064533003	G1	Auto6378675383150341000	B3

Destination Barcode in trfo.csv must equal Source Barcode in pick.csv!

Source Barcode	Source Region	Feature Position X	Feature Position Y	Destination Barcode	Destination Well
Auto6378675383150341000	A1	17.3168	32.4568	Auto6378675443816326512	A1
Auto6378675383150341000	A2	36.8496	33.9639	Auto6378675443816326512	B1
Auto6378675383150341000	A3	74.6901	33.26	Auto6378675443816326512	C1
Auto6378675383150341000	A4	99.4108	30.8381	Auto6378675443816326512	D1
Auto6378675383150341000	B1	15.8533	72.3121	Auto6378675443816326512	E1
Auto6378675383150341000	B2	46.8628	72.2123	Auto6378675443816326512	F1



08_PLA.py: run



example run:

in terminal, from script folder, type:

python 08_PLA.py

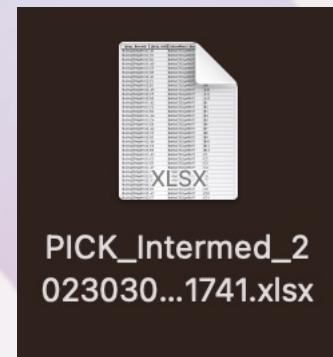
example user input:

```
Gibson barcodes in this transformation round... ['BAOsGiAsp00026', 'BAOsGiAsp00027']
Enter trafo filenames, in Gibson order, separated by commas:
TRFOp00056.csv, TRFOp00057.csv
Enter picking filenames, in Gibson order, separated by commas:
CUL1p00107.csv,CUL1p00109.csv
```





08_PLA.py: output



Pick_Intermed_date-time.xlsx



Qtray_Barcode	Qtray_well	CulturePlate1_Barcode	CulturePlate1_Well
Auto6378675383150341000	A1	Auto6378675443816326512	A1
Auto6378675383150341000	C1	Auto6378675443816326512	A2
Auto6378675383150341000	E1	Auto6378675443816326512	A3
Auto6378675383150341000	A5	Auto6378675443816326512	A4

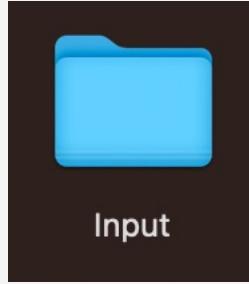
PLA_import_date-time.xlsx

specPCR_GiAS_ID	Trafo_Barcode	Trafo_Well	Qtray_Barcode	Qtray_well	Colonies
5DA2B3FD-530A-4486-8681-98B0FC807CAD	Auto6378667015064533003	A1	Auto6378675383150341000	A1	Y
2343E1BE-029C-475F-B195-1660DD8D977C	Auto6378667015064533003	A2	Auto6378675383150341000	C1	Y
D77E74BC-76F1-4F14-8683-194ABBD323C1	Auto6378667015064533003	A3	Auto6378675383150341000	E1	Y
4BE3B516-F0DA-4DFB-B73A-67B443D8A474	Auto6378667015064533003	A4	Auto6378675383150341000	A5	Y





09_MINI.py: input files



Input



Trfo_IDs.xlsx

Plating_ID	Qtray_Barcod	Qtray_well	Colonies
8AD4903D-4B01-4066-B97	Auto6378675383150341000	A1	Y
81EA71FA-A035-4AC9-87A	Auto6378675383150341000	C1	Y
8BB24D0A-F1DF-440B-9AE	Auto6378675383150341000	E1	Y
DAB43F32-2716-479B-8A2	Auto6378675383150341000	A5	Y
299ED759-A9B7-40F9-83E	Auto6378675383150341000	C5	Y
E2466DDC-CF8D-4728-BB1	Auto6378675383150341000	E5	Y

File provided by FileMaker (DB)
script

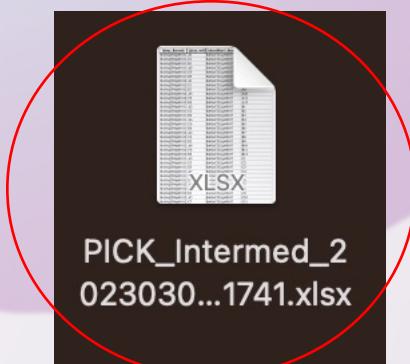




09_MINI.py: input files



LATEST PICK_Intermed_date-time.xlsx





04_picking_to_minis.py: run



example run:

in terminal, from script folder, type:

python 09_MINI.py

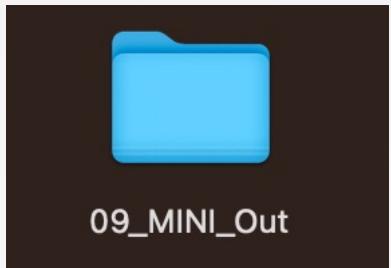
example user input:

```
Enter GLYCEROL stock barcode for culture plate Auto6378675529556306895...
gly2
Enter PLASMID plate barcode for culture plate Auto6378675529556306895...
pla2
Enter WELLS with NO GROWHT for culture plate Auto6378675529556306895, separated by commas...
h9,h12
Enter MINIPREPS date for culture plate Auto6378675529556306895 as YYYYMMDD...tomorrow
Was culture plate Auto6378675529556306895 a REPICK? N if no, Y+repick num if yes e.g., Y1...
y1
Enter WELLS + CONC for culture plate Auto6378675529556306895, separated by commas, format: well,conc...
h9,77,h10,101.7
```





09_MINI.py: output



MINI_import_date-time.xlsx

A	B	C	D
Qtray_Barcode	Qtray_well	CulturePlate1_Barcode	Cultur
Auto6378675383150341000	A1	Auto6378675443816326512	A1
Auto6378675383150341000	C1	Auto6378675443816326512	A2
Auto6378675383150341000	E1	Auto6378675443816326512	A3
Auto6378675383150341000	A5	Auto6378675443816326512	A4
Auto6378675383150341000	C5	Auto6378675443816326512	A5
Auto6378675383150341000	E5	Auto6378675443816326512	A6

...

H	I	J	K	L	M	N
GlycerolStock_Barcode	GlycerolStock_Well	PLA1_Barcode	PLA1_Well	Repick	MiniPrepDate	NANODROP_cc
gly1	A1	pla1	A1	N	today	234
gly1	A2	pla1	A2	N	today	345.6
gly1	A3	pla1	A3	N	today	
gly1	A4	pla1	A4	N	today	
gly1	A5	pla1	A5	N	today	
gly1	A6	pla1	A6	N	today	

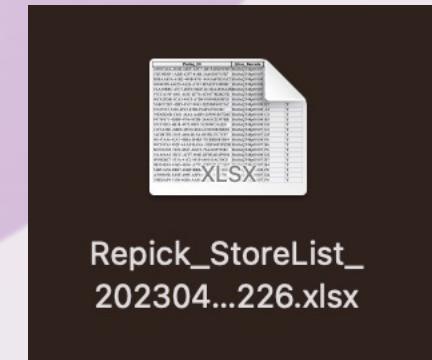
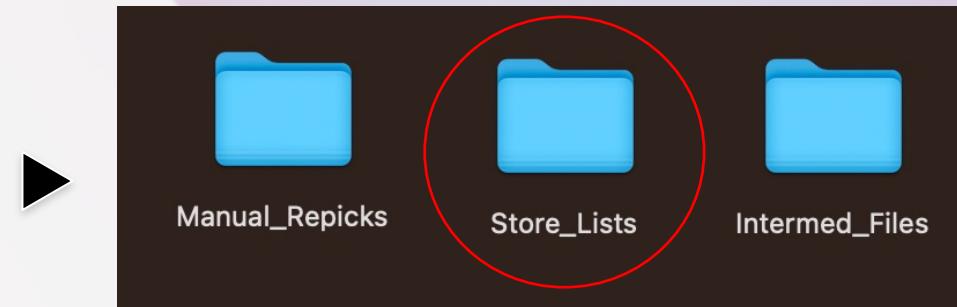
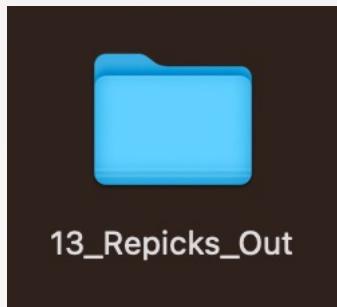




13_Repick.py: input files



LATEST Repick_StoreList.xlsx
(modification date)



A	B	C	D
Plating_ID	Qtray_Barcode	Qtray_well	Colonies
751225D5-471F-44E4-8AF2-FAC2AC27F6E1	BAOsQT48p00095	C6	Y
40F0708C-EB54-4CA1-AA37-E6E43A78910C	BAOsQT48p00096	A2	Y
333B20FB-9F46-45F0-A070-89D161881E07	BAOsQT48p00095	A3	Y
17D06E9D-747D-415C-800D-0026CB772E62	BAOsQT48p00095	C7	Y

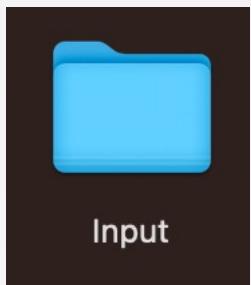


Provided by Knime Script

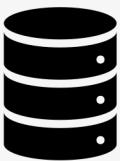




13_Repick.py : input files



File provided by FileMaker (DB)
script



A	B	C
FK_Plating_ID	CulturePlate1_Barcode	CulturePlate1_Well
95CFE459-EF9D-4897-9139	BAOsCUL1p00006	G4
DDA5A604-D929-44C6-BCA	BAOsCUL1p00006	G12
74B158E6-03AF-44AA-A59	BAOsCUL1p00006	H8
9325C462-A7A2-4899-AED	BAOsCUL1p00006	G2
10675A47-73E2-408C-8203	BAOsCUL1p00006	G3
C1E587BD-512C-4585-BD3	BAOsCUL1p00006	H7
843C6E0A-E0A5-4A7E-9CB	BAOsCUL1p00006	A1

...

I	J	K	L
no._	Growth_CulturePlate2	MiniPrepDate	Comment
Y		20201109	





13_Repick.py : run

usage: 13_Repick.py -c CULTURE_PLATE -b GLYCEROL_STOCK_PLATE -p PLASMID_PLATE

arguments:

- h, --help show this help message and exit
- c CULTURE_PLATE, Latest culture plate number
- b GLYCEROL_STOCK_PLATE, Latest glycerol stock plate number
- p PLASMID_PLATE, Latest plasmid plate number

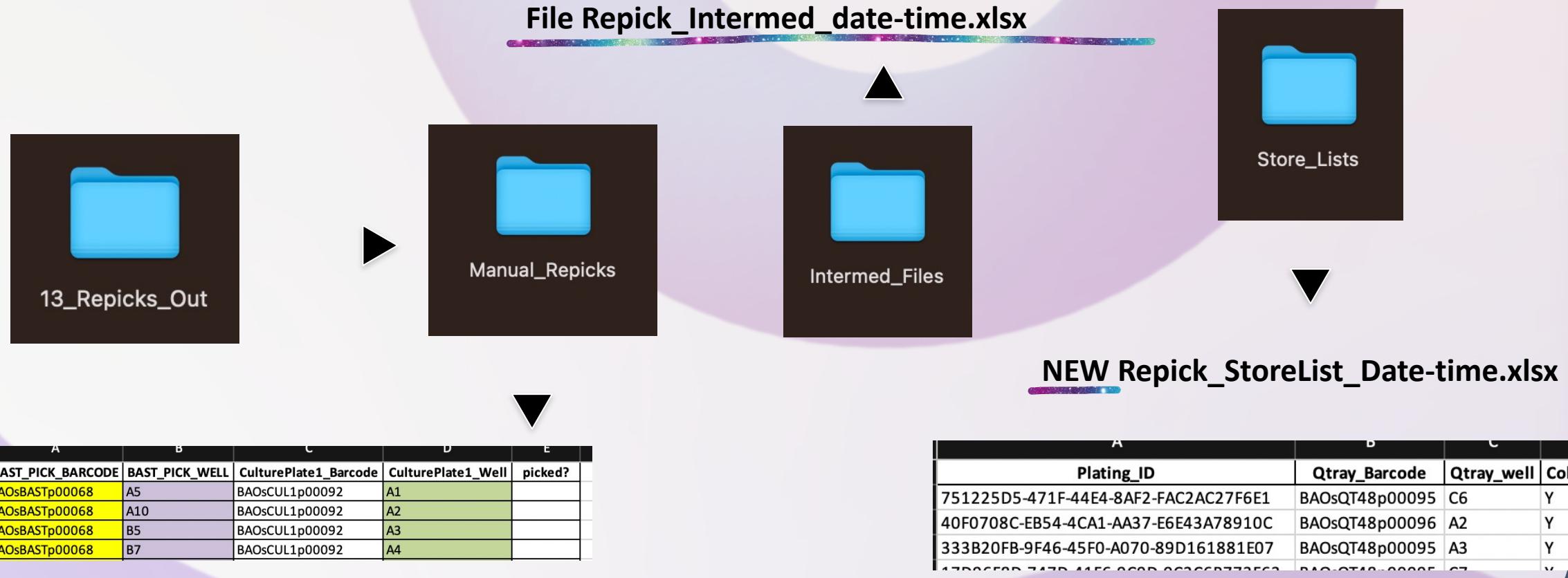
example run:

in terminal, from script folder, type:

python 13_Repick.py -c 22 -b 55 -p 88



13_Repick.py: output



List for wet lab manual handling



13_Repick.py: output



A	B	C	D
FK_Plating_ID	CulturePlate1_Barcode	CulturePlate1_Well	
0D8804CA-82B9-419A-AC00-8F40C1EFFCDE	BAOsCUL1p00093	A1	
1D9DB4D3-E6A5-4584-9BC7-398604F89ED0	BAOsCUL1p00093	A2	
90FF7C6A-A94B-4E3A-8EFA-07253ED9F4AD	BAOsCUL1p00093	A3	
57B5D592-2F94-4D97-BCA9-075109352DC5	BAOsCUL1p00093	A4	

K	L	M
BAST_PICK_BARCODE	BAST_PICK_WELL	Comment
BAOsBASTp00076	H9	inoculated from/identical with BAOsBASTp00076, H9
BAOsBASTp00078	A2	inoculated from/identical with BAOsBASTp00078, A2
BAOsBASTp00078	A5	inoculated from/identical with BAOsBASTp00078, A5
BAOsBASTp00078	A8	inoculated from/identical with BAOsBASTp00078, A8

Current database import file Repick_Import_date-time.xlsx file(s)



Ab Parts

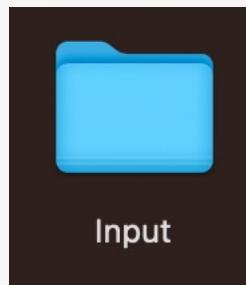




00_Ab_Traf.py: input files



A	B	C	D	E
PlateBarcode_WorkingStock	PlateWell_WorkingStock	A_SingleChainList_C::Plasmid Name	Working_Stock_ID	A_SingleChainList_C::Antibody_ID
FE00450359	A1	HP004_H000713	3823BFC4-9AD7-FC43-BAEC-2F5C7B3E17A9	HP004_HK000713
FE00450360	A1	HP004_K000713	8F8F7375-F971-C44F-9796-0009D6E059B4	HP004_HK000713
FE00450359	A10	HP004_H000722	F2B217C3-AC79-7D44-BD15-E740624F7555	HP004_HL000722
FE00450360	A10	HP004_L000722	5D6A6C40-DE16-7844-969A-48B46BD797BC	HP004_HL000722
FE00450359	A11	HP004_H000723	BA3FB2A9-CC64-9144-973D-2AEC3EE7A050	HP004_HK000723



File provided by FileMaker (DB)
script





00_Ab_Traf.py: run



usage: 00_Ab_Traf.py -d TRANS_DATE -t TRANS_PLT

arguments:

- h, --help show this help message and exit
- d TRANS_DATE, Transfection date
- t TRANS_PLT, Transfection plate barcode

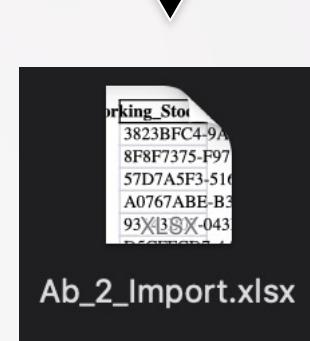
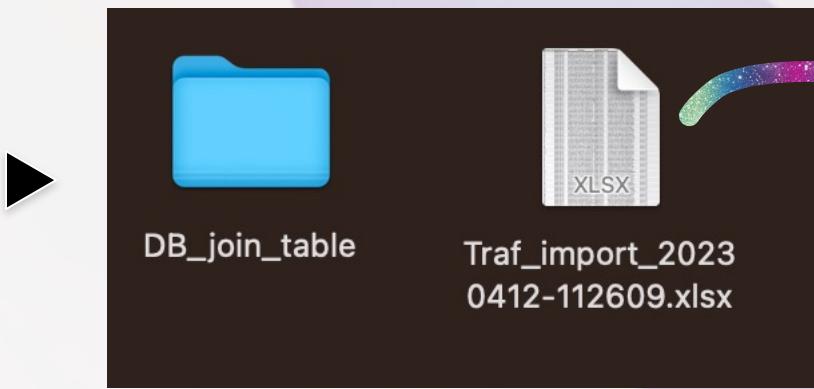
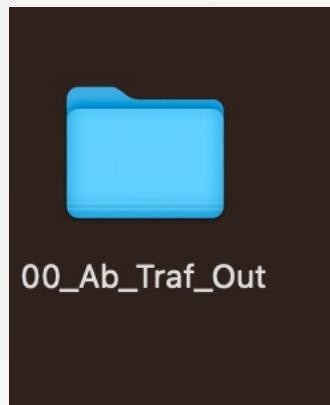
example run:

in terminal, from script folder, type:

python 00_Ab_Traf.py -d 20221021 -t p1_



00_Ab_Traf.py: output



Traf_import_date-time.xlsx:

A database import file

Transfection_Well	PlasmidName_pDNA1
A1	HP004_H000713
A2	HP004_H000714
A3	HP004_H000715
A4	HP004_H000716
A5	HP004_H000717

Transfection_date	Transfection_Barcode
20221021	p1_

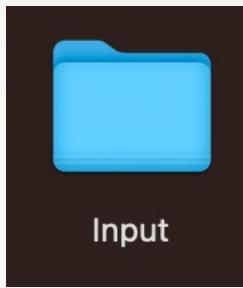
Ab2_import.xlsx: a database import file (needed for join table)

! Running the script updates this file

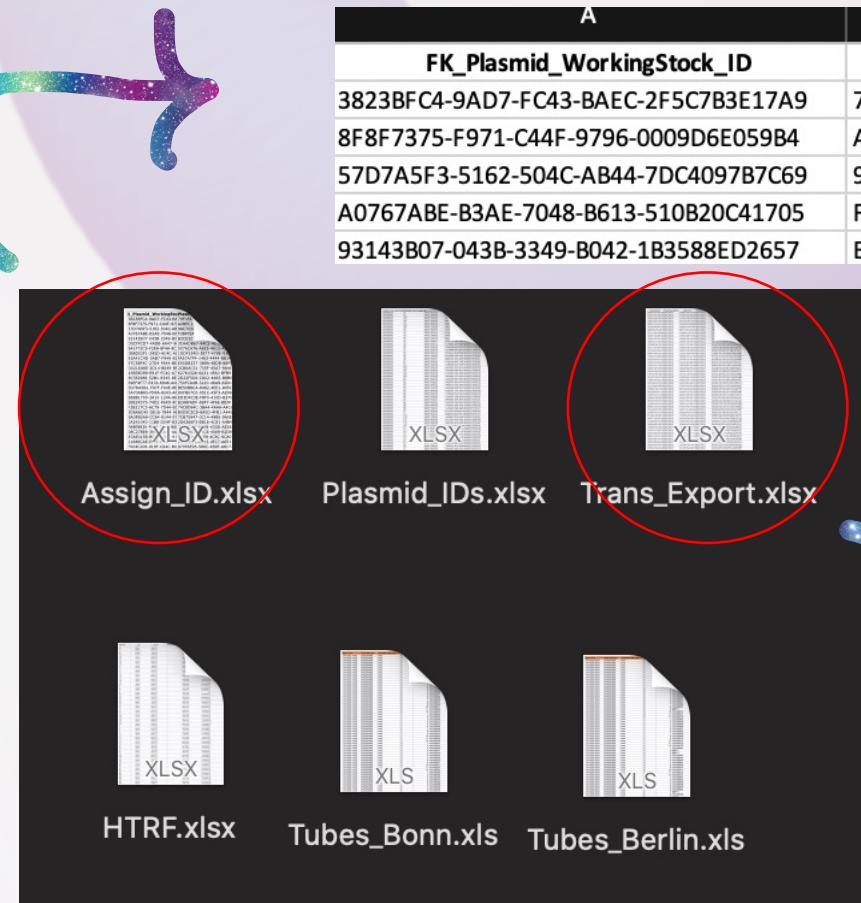
Working_Stock_ID
3823BFC4-9AD7-FC43-BAEC-2F5C7B3E17A9
8F8F7375-F971-C44F-9796-0009D6E059B4
57D7A5F3-5162-504C-AB44-7DC4097B7C69



01_Ab_Quant: input files



Files provided by FileMaker
(DB) script

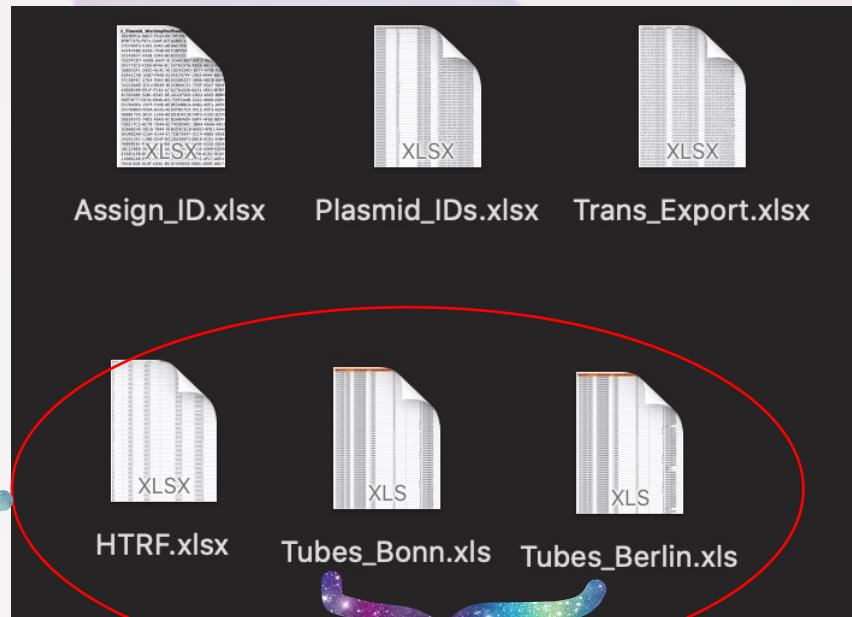
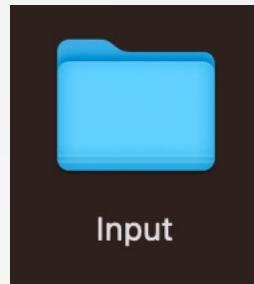


A	B
FK_Plasmid_WorkingStock_ID	Plasmid_Assignment_ID
3823BFC4-9AD7-FC43-BAEC-2F5C7B3E17A9	79F05882-3ECE-4644-B252-D81DF1E110A2
8F8F7375-F971-C44F-9796-0009D6E059B4	A2BEC144-05CD-46CC-A6DE-EC4167E3A0F1
57D7A5F3-5162-504C-AB44-7DC4097B7C69	9AC70504-DB0A-41AD-9C0E-FA39F2059707
A0767ABE-B3AE-7048-B613-510B20C41705	F3BFF5A6-D3AC-4F29-88AF-07F441AD6EF7
93143B07-043B-3349-B042-1B3588ED2657	B2CE321B-7D9D-4802-A34C-09FD2ABA65B1

A	B	C	D	E	F
Transfection_ID	PlasmidName_pDNA1	pDNA1_ID	PlasmidName_pDNA2	pDNA2_ID	Transfection_Well
0B4F5C49-7283-4AA7-B0B	HP004_H000713	3823BFC4-9AD7-FC43-BAE	HP004_K000713	8F8F7375-F971-C44F-9796-0009D6E059B4	A1
075E6932-51A6-4365-8C9	HP004_H000714	57D7A5F3-5162-504C-AB4	HP004_L000714	A0767ABE-B3AE-7048-B613-510B20C41705	A2
822FE17B-1944-439B-9DB	HP004_H000715	93143B07-043B-3349-B042	HP004_L000715	D5CFCD7-4ADB-6A47-9696-B5146A78906F	A3
30BE8835-E8FA-42EF-81D	HP004_H000716	B41F72C9-F2E8-8F4A-8C5	HP004_L000716	0B8D02F1-295D-AC4C-A92A-901A38C3119B	A4



01_Ab_Quant.py: input files



Script needs HTRF data, raw tube scan (Bonn, Berlin) for current plate (experimental readouts files, tube barcodes for harvested mABs)

A	B	C	D	E	F	G
HTRF plate	620	665	plate	RATIO 665/620*10000	µg/ml	96well
A1	2663865	3033940	p1	11389.24082	5.558483003	A1
A3	4554307	846408	p1	1858.478139	0.222480964	A2
A5	2995105	4249577	p1	14188.40742	8.513462887	A3
A7	2474888	4410326	p1	17820.3054	16.9718241	A4
A9	4376196	939471	p1	2146.775419	0.334850928	A5
A11	4114234	1679668	p1	4082.577705	1.142276243	A6

Scan Time	Rack Barcode	Orientation Barcode	Tube Row	Tube Column	Tube Barcode
17/1/2022 14:22	FE00263692	none	A		1 1047488544
17/1/2022 14:22	FE00263692	none	A		2 1047488552
17/1/2022 14:22	FE00263692	none	A		3 1047488560
17/1/2022 14:22	FE00263692	none	A		4 1047488568
			A		5 1047488576





01_Ab_Quant.py: run



usage: 01_Ab_Quant.py

example run:

in terminal, from script folder, type:

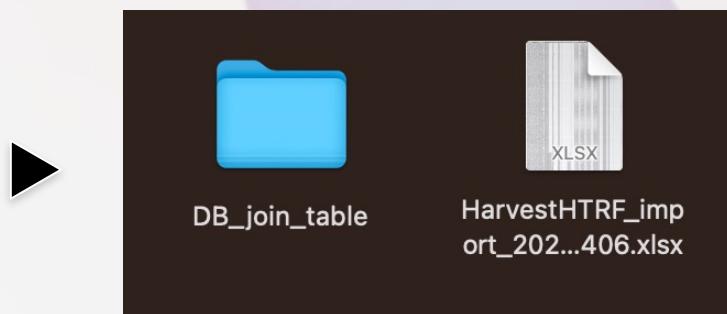
python 01_Ab_Quant.py

example input:

```
Enter harvest date YYYYMMDD...
20221025
Enter harvest day (e.g. D4)...
D4
Enter the volume for LAT aliquot [ul]...
400
Enter the volume for Berlin aliquot [ul]...
300
Enter date of sending to Berlin...
20221010
```



01_Ab_Quant.py: output



HarvestHTRF_import_date-time.xlsx:

A database import file

A	B	C	D	
Transfection_ID	Aliquot_Barcode	Aliquot_well	Location	Sample
0B4F5C49-7283-4AA7-B0B0-5FC68455A60D	FE00263692	A1	LAT	
075E6932-51A6-4365-8C92-FE74F8A568F8	FE00263692	A2	LAT	
822FE17B-1944-439B-9DBB-D405B8080890	FE00263692	A3	LAT	
30BE8835-E8FA-42EF-81D2-162772D84869	FE00263692	A4	LAT	
570B247B-F9A9-472D-8114-59C6DF7BADD6	FE00263692	A5	LAT	
869BEDA2-33C8-450B-A557-DA56B6A4DE66	FE00263692	A6	LAT	
86F92CE1-F71E-461D-A3C6-974B747AF0BD	FE00263692	A7	LAT	
0252027B-FED9-446E-A608-8E7AAEFA0642	FE00263692	A8	LAT	

K	L	M	
I	Harvest_Date	Harvest_Day	Transfection_Method
1	20221025	D4	96
2	20221025	D4	96
3	20221025	D4	96
4	20221025	D4	96
5	20221025	D4	96
6	20221025	D4	96
7	20221025	D4	96
8	20221025	D4	96

A database import file (needed for join table)

A	B	C
Plasmid Assignment ID	Transfection ID	PlasmidName_pDNA
79F05882-3ECE-4644-B252-D81DF1E110A2	0B4F5C49-7283-4AA7-B0B0-5FC68455A60D	HP004_H000713
A2BEC144-05CD-46CC-A6DE-EC4167E3A0F1	0B4F5C49-7283-4AA7-B0B0-5FC68455A60D	HP004_K000713
9AC70504-DB0A-41AD-9C0E-FA39F2059707	075E6932-51A6-4365-8C92-FE74F8A568F8	HP004_H000714
F3BFF5A6-D3AC-4F29-88AF-07F441AD6EF7	075E6932-51A6-4365-8C92-FE74F8A568F8	HP004_L000714
B2CE321B_7D9D_4802_A34C_095D2ABA65B1	822EE17B_1944_439B_9DBB_D405B8080890	HP004_H000715