PARAFAC

Date: NOV 22

① Make a Changelog sheet
- to cut down the data to put into MATLAB.
- in excel

Don't put ABC in excel Changelog file

A	B	C
path to your samples (in template form).	RU ↑ to recognize the tab RU in templates	PARAFAC_1 PARAFAC_2

- Put all samples in a folder called EEM-Scans.
Go to this folder & highlight all your samples
right click
→ Path Copy
→ Copy Long Path
- And paste into column A

File → save as change Log in Folder where you are keeping everything.

② Open Matlab (7.50 (R2007b))

Editor
- where you code is (to read)

Workspace
data in program that you're working with

Current Directory
where you switch b/t file folders.

Command Window
where you type the codes in.

- Click on Editor tab

Date

③ File
Open
→ My Computer → Local Disk C → Program Files
→ MatLAB → R2007b → toolbox → DOMFluor1_7
→ DOMFluor0.
→ EEM Change for MatLAB.

④ Within the code:
A = can remain the same.
B = should be ok. It's saying what part of the B-tab in the change log do you want to use.
want → Ex: 250-500
Em: 300-600

in RU tab.

↓	Ex: 250 → 500 (6:56)
17	300
167	600
Em	

∴ (17:167)

so just ensure this is right in the templates.

⑤ In Folders,
- make new one called 'NovEEM.'
(can change this to whatever.)
& 'NovEEM_Plot'

⑥ Click on NovEEM, open, + copy path
paste into the purple code ② pathname=
at Line 31

Date

⑦ Click on NOV EEM Plot, open, copy path & paste into pathname = ① Line 95

* keep \'; At end of extension.

⑧ At line 115 = pathname
-need to tell where you want FI_new Spreadsheet to put. So put a folder ~~ext~~ pathway infront of '\FI_new.xls'

The FI_new wants EX = 370 (30 or A5)
EM = 470 → 520
(102 127)
so code is: EM:470 EX:370 EM:520 EX:370
B(102,30)/B(127,30) (line 112)

look at RU tab to ensure matching up

⑨ Go to workspace
File → import data → Get Change_Log Spreadsheet
→ open → when Importwizard dialog box Appears, rename "sheet 1" to "test"
→ click Finish.

⑩ Go back to Editor-tab
copy all & paste all into command Window.
There will be a prompt for how many samples you have in change_log folder. (i.e → 5.)

Page

The contour plots will appear.
Watch + make sure nothing seems wrong (ie ⇒ axis doesn't change), the overall pattern doesn't change)

* if it says something is missing, program will stop. Can go see where it stopped + make new excel file keeping only to where it stopped.
- then in command type 'clear'
- reload new file
- copy + paste again
- type in how many samples have now

(11) The FL_new didn't make sample names, but they are in the Change_log.
- make new excel sheet called NOVEEM_sample order.

names PARAFAC # | Removed B4 Big Matrix | Big Matrix order (n=5) →

Cut data 20x20 ↗ Cut data 2 (35x35)

Removed due to contamination

test 1 n = ≤ (if removed should be less than any)

test 2 n = __

Date

(12) In MatLab
- click editor → open → pathway to DOM Fluoro → combine sample file.

(13) in Current Directory
- click through directory & open the NOV_EEM folder (with PARAFAC_1, PARAFAC_2 ..., files).
(.xls)

* make sure files saved as PARAFAC_1.xls or will not import w/o error.

(14) In the Editor, copy statement: line 6
bigMatrix = combineSampleFiles('*.xls');
* paste into command window.
- hit enter. You will see

	n	#Emλ	#Ex	
(not shown)	n			2
(shown)	5	157	51	5

Can verify by opening PARAFAC_1 file & counting rows down (151) & across (51).

(15) Back in Editor,
File → open → pathway to DOM Fluoro → PARAFAC_modeling.

Page

(Prehandling of data software).

Date

(16) In the command window, copy + paste from code:

Step P1

```
X=bigMatrix;
Clear bigMatrix;
```

Step P2

```
Em =(300:2:600)'; From Step 4
Ex= (250:5:500)';
```

Step P3

```
nEM=151;  % # of emmissions
nEX=51;   % # of excitations
```
} in PARAFAC_1.xls files

```
nSample=__; % need to change this to # of samples you have. Like in your change log.xls
XBackup=X; % backup of original data.
```

* All these files will show in workspace tab.
* Never change original file matrix.

Step P4

-copy all of step P4 and paste into command window.

* Now steps follow with the Paper tutorial.

Steadmon + Bro.

Tutorial + Continuation of Steps. Date

-The last thing you should see in command window is Original data:

Ex: [51 x 1 double] list of Ex λ (nm) 250→500nm, measuring every 5 nm.

Em: [151 x 1 double] list of Em λ (nm) 300→600nm, measuring every 2nm

X: [5 x 151 x 51 double] the fluoro. data as a 3D array (5 samples x 151 em λ's x 51 ex λ's)

nEX: 51
nEm: 151
nSample: 5
} # of ex λ's, # of em λ's, # of samples.

XBackup: [5 x 151 x 51 double] backup copy of data.

Step 1: Check EEM's.

PlotEEMby4(1, Original data, 'R.U.')

-red image would indicate a (-) #. We tell them to be 0.
-if they are in areas that are essential to us, we would have to take them out + remodel.
-write down red ones.

Date

Step 2: Cutting the region of the Spectra influenced by Scatter peaks.

[CutData] = EEMCut(OriginalData, 20, 20, NaN, NaN, 'No')

if change no 'Yes', you click space bar to see graph. No just makes them go after a few seconds.

- this deletes data in region where no fluorescence. (em λ's are less than ex λ) & regions influenced by scatter, & replaces them with NaN = (not a #).

*can change the 20x20.
Clayton DecEEM Sample Summary file has 35x35,

*if need to remove data, may want to go to step 4 (removing outliers) first, before step 3 (identify outliers). So can get rid of the ones you visually see first.

Step 3: Initial explorative data analysis & outlier identification

Tuturial step before M.L step 3:
[Test 1] = OutlierTest(CutData, 2, 1, 7, 'No', 'No')

- will run models 2→7 components.

Date

-to assess these models, use loadings & leverages.

PlotLoadings(Test1, 2)

-will show the loadings of of the model (a, b, c from Test1)
- b & c show the Fm & Ex loadings.
- a shows how the concentration of the two components varies between samples.
- loadings should be smooth

PlotLeverage(Test1, 2)
-will show a figure for the leverages
-plots with extreme leverages may indicate outliers. (ie → higher than 0.5)

component.

PlotLL(Test1, 2)
will put the loadings & leverages on one graph.

-now would do PlotLL(Test1, 3)
" " 4)
" " 5)
" " 6)
" " 7)

-would need to constrain model, since some loadings may be negative. This makes loadings "look" wrong, bc a wrong amount of components may be being forced to persue,

even if a certain number of them aren't valid.
- so we rerun with non-negatively constraints.

M.L Step 3:

lowest # of component highest # of component

[Test2] = OutlierTest(CutData, 2, 1, 7, 'Yes', 'No')

skipping by # of components.

'Yes' = make (-) #'s a 0?
'No' = stop everytime it makes a model?

do the PlotLL(Test2, 2), 'T'
" " " 3)
" " " 4)
" " " 5)
" " " 6)
" " " 7)

* if the leverage plots show samples that may be problematic, try plotting them again & seeing how they compare to others.
ie: write down their numbers [6, 9, 47]
↳ do PlotEEMby4(1:5, Test2, 'RU')
↳ watch for the identified samples.
- if they look questionable, remove!

Date

Step 4 Remove Outliers from DataSet

[Test3] = RemoveOutliers(CutData, [5], [], [])

↑ would

* so, if wanted to remove samples 2 & 5, redd... [2 5], [], [])

- (I only removed 5 in practice run NOV 22).

To rerun model on new data:

[Test3] = OutlierTest(Test3, 1, 1, 7, 'No', 'No')

↑ but this would cause the negative issues again. (And MatLab just shows a error...)

so go

[Test3] = OutlierTest(Test3, 1, 1, 7, 'Yes', 'No')

↑ to make 0 values a 0.

Again want to look at the loadings/leverages. to see if any other now look questionable.

PlotLL(Test3, 1)

* if some looked questionable, remove, & repeat steps. Also can compare samples again by PlotEEMby1(1:4, Test3, 'RU')

Date

Now, comparing the spectral sum of squares.
CompareSpecSSE (Test3, 4, 5, 6,)
components you're interested in.

-can then compare other components.
ie → (6,7,8) .. 4,6,8) .. (3,5,7) ..

-once comparing them all, if you see, for example, that the step from the 6→7 model offers little improvement of fit, it suggests that 6 or fewer components are adequate for the data.
So you know the highest component may be 6.

Now, Compare 2 models
Compare2Models(Test3, 3, 4)

this will look at the models 3 & 4. Then can repeat for 4,5 ; & 5,6 , & 6,7

This will somehow? tell you the at least number of components you have. (ie maybe 4)

- can then compare loadings of models by PlotLoadings(Test3, 4)
i)

Step

This

-crec
sum
-The
simi

Now

this

Next

This
mo
& r

The

Date

Step 5: Split Half Analysis + Validation.

This splits the data twice < 1st split / 2nd split.

[AnalysisData] = SplitData(Test+3)

- creates a figure w four plots, showing the sum of squares for each split.
- The curves for each pair of halves should be similar.

Now to split between components

[AnalysisData] = SplitHalfAnalysis(AnalysisData, (3:7), 'Mydata.mat')
or SplitHalfTest1.mat)

this will fit a model with 3-7 components.

Next, use

SplitHalfValidation(AnalysisData, '1-2', 3)

This will look at split 1-2 on the 3 component model. Then do 3-4 on 3 component model, & repeat for model 4-8 (or highest chosen.

[e Split..., '1-2', 4)
'3-4', 4)
'1-2', 5)
'3-4', 5)

The code will tell you if validated or not

Date

Step 6 Analysis using Random initialization.

[AnalysisData] = RandInitAnal(AnalysisData, 4, 10)

This is running 10-four component models using random initialization. (Note: if in Step 5, the Split half validation Analysis found that component 5 or 6, were validated, you would have to run these components too)
ie ...AnalysisData, 5, 10)
...AnalysisData, 6, 10)

- once the model finishes, it shows a plot of the sum of squares. The model with the least sum of squares is highlighted with a green circle.
(Note you can enter in more than one
[AnalysisData] = RandInitAnal(AnalysisData, 5, 10)
" 6, 10)
at once. will show you the component image.

Date

-Now, check & see that the loadings & fit of the model are ok.

```
PLotLL(AnalysisData,4)
SplitHalfValidation(AnalysisData,'1-2',4)
```

can also `EvalModel(AnalysisData,4)`
& do the Tucker Congruence Coefficient

```
TCC(AnalysisData.Model(4),AnalysisData.
Split(1).Fac{4})
```

↑ M.L code says 13 wrong.

*using component 4 in all these because thats what was validated.

Now, create contour plots of components

```
ComponentEEM(AnalysisData,4)
```

Step 7: Save & Export data

[EMax] highlight path of directory where you want it to save. \namespreadsheet.xls

save Path of directory \name.mat