352SemProj-Final-Zhonghao Zhan

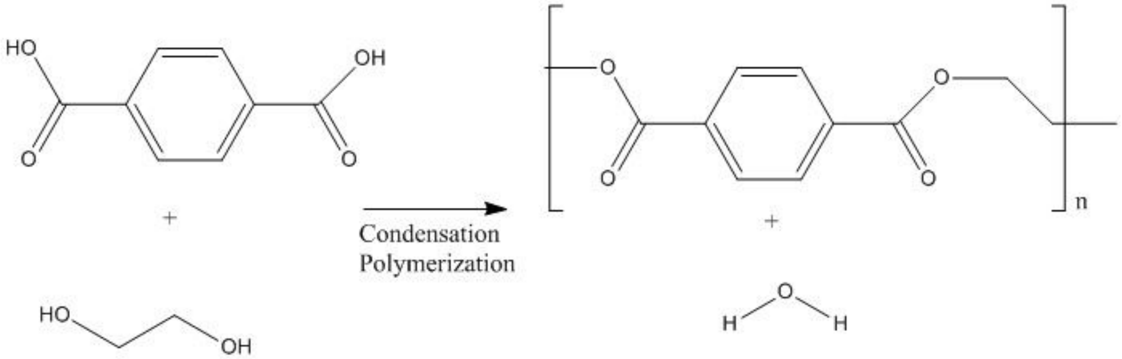
Zhonghao Zhan

July 27, 2017

Table of Contents

Project Background

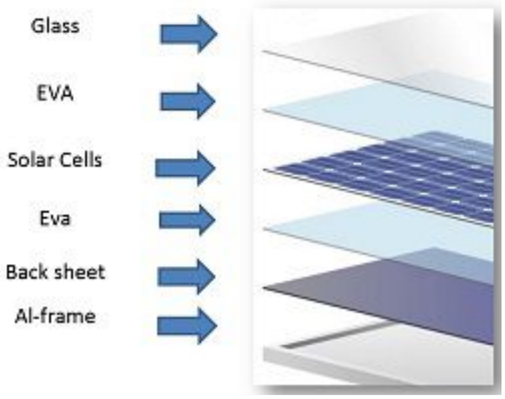
This project works on weathering study data from indoor exposure of clear poly(ethylene terephthalate) films. These data are taken step-wise, where the samples are exposed to accelerated conditions of temperature, humidity and UV irradiance relative to real-world, outdoor conditions in order to more quickly understand the degradation behavior of the materials.



The condensation reaction of terephthalic acid and ethylene glycol to form PET

PET...

- Is a condensation polymer of terephthalic acid/dimethyl terephthalate and ethylene glycol
- Is commonly found in the form of plastic bottles
- Has a specialty application in photovoltaic (PV) backsheets



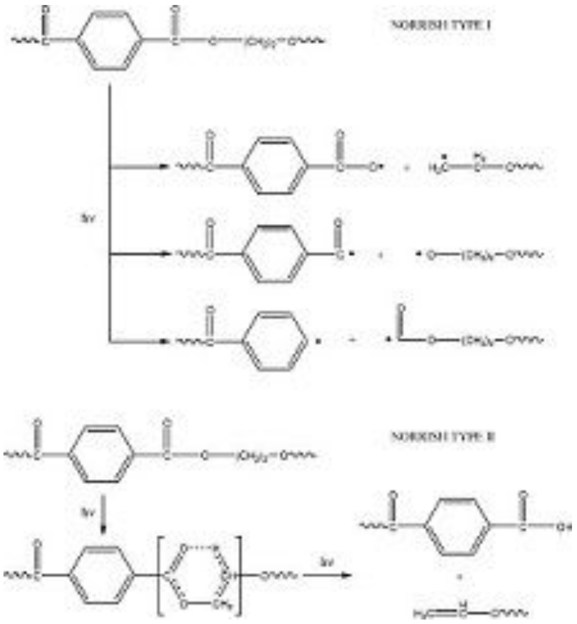
Layout of a PV module

- Backsheets are electrical and environmental barriers for PV modules.

- PET is used as a core layer in backsheets for its high dielectric breakdown strength and low cost.

- PET is susceptible to degradation by environmental stressors such as heat, humidity,and ultraviolet (UV) light.

- Degradation of PET causes changes in its chemical structure that lead to loss of dielectric breakdown strength and mechanical properties.



Norrish degradation mechanisms for PET

- UV and humidity exposure cause degradation through chain scission mechanisms.

- Increases in temperature cause acceleration of degradation processes.

- UV stabilizers function to scavenge radicals, absorb and screen damaging radiation to temporarily

- protect the polymer, or quench electronically excited states.

- Hydrolytic stabilizers act as water and acid scavengers and convert both into neutral urea structures.

- The degradation of PET can be monitored via fluorescence/luminescence spectroscopy. Luminescence/Fluorescence Spectroscopy: https://en.wikipedia.org/wiki/Fluorescence_spectroscopy

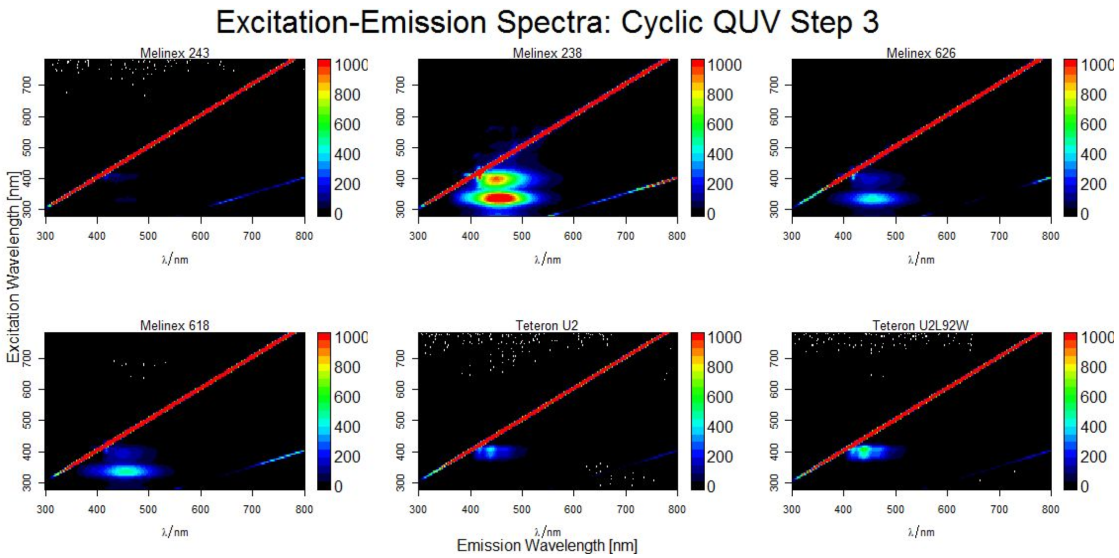
- Fluorescence involves using light to electronically excite a material and subsequent detection and analysis of the light emitted by the material upon relaxation.

- Material structure determines the range of wavelength absorbed by the material and features of the emitted light.

- Enables the characterization of polymers by electronic structure

- Allows observation of changes in polymer structure due to degradation.

- 3-Dimensional excitation emission matrix plots were generated for each exposure step to observe changes in the fluorescence behavior of the material after time in exposure



Example of 3D Excitation-Emission Spectra for PET Films after QUV Exposure

Exposure and Measurement

Luminescence Spectroscopy

- 3D Excitation Emission spectra were measured with a Agilent Cary Eclipse Spectrophotometer following each exposure step

- Measurements were conducted on the retained sample from the exposure protocol

One sample was retained (removed from study) following each exposure step, so the number of samples for a given material under a given exposure will decrease by 1 from step to step.

Hot QUV

168 hr (one week) time steps ○ Continuous exposure to UVA-340 light at 1.55 W/m 2 /nm and 70 °C

Cyclic QUV

168 hr (one week) time steps ○ 8 hrs of exposure to UVA-340 light at 1.55 W/m 2 /nm and 70 °C followed by 4 hrs of darkness and condensing humidity at 50 °C

Damp heat

304 hr (two week) time steps ○ Continuous exposure at 85 °C and at 85 °C relative humidity

Databook

Metadata (Also predictors)

Sample : each sample is assigned a unique identifier, first 5 digits identify original film sheet, second 5 digits describe which subsample of that sheet Material : defines the type of PET film, the films vary in composition Exposure : defines the accelerated exposure that the film underwent

Predictors

Time : defines the amount of time cumulative spent under exposure at the time of measurement

Responses

Fluorescence Spectra : spectra which shows intensity of light emitted by a sample at a given wavelength (energy) versus the wavelength used to cause the excitation phenomena material; changes in the peaks in the spectra correspond to the formation and degradation of chemical species within the materials with increasing exposure time. ● Data saved as .spc files

Project Aim

1. Load excitation-emission spectra into R using the eemR package.
2. Plot excitation-emission spectra using the eemR package.
3. Remove negative values and Raleigh/Raman scatter from the excitation-emission spectra.
4. General EDA and plotting of corrected spectra. Provide insights about the data gain from EDA and analyze the trends with increasing exposure time/steps.
5. Study and explain the concept of parallel factor analysis (PARAFAC)
6. Perform parallel factor analysis on spectra grouped by exposure
7. Provide visualization for PARAFAC results.
8. Attempt to interpret PARAFAC results

Package

● eemR Provides various tools for preprocessing Emission-Excitation Matrix (EEM) for Parallel Factor Analysis (PARAFAC). Different methods are also provided to calculate common metrics such as humification index and fluorescence index.

●shiny Makes it incredibly easy to build interactive web applications with R. Automatic "reactive"" binding between inputs and outputs and extensive prebuilt widgets make it possible to build beautiful, responsive, and powerful applications with minimal effort.

● Multiway (search for parafac function) Fits multi-way component models via alternating least squares algorithms with optional constraints: orthogonal, non-negative, unimodal, monotonic, periodic, smooth, or structure. Fit models include Individual Differences Scaling, Parallel Factor Analysis (1 and 2), Simultaneous Component Analysis, and Tucker Factor Analysis.

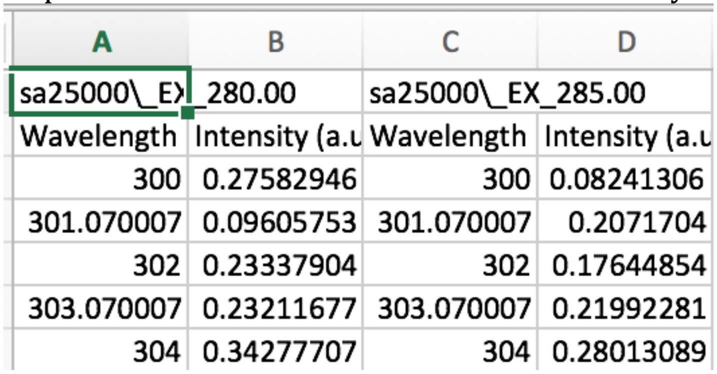
● PLS_toolbox (Matlab) It takes its name from the Partial Least Squares (PLS) regression method, which has become the standard calibration method in many calibration and modelling applications. It contains all the software tools chemical engineers, analytical chemists and other analysis-driven scientists require to fully utilize their data and build predictive models.

● drEEM toolbox (Matlab) The drEEM toolbox combines two toolboxes: the DOMFluor toolbox which introduced several algorithms to define and test the right number of components of a PARAFAC model) and the FDOMcorr toolbox which conducts the neccessary pre-processing steps to make the fluorescence data comparable between instruments and studies). The procedures of these two toolboxes are furthermore extended by additional visualisation functions and other functions to support and simplify modelling of Excitation-Emission-Matrices (EEMs) of natural CDOM with PARAFAC .

Data Cleaning and Correction

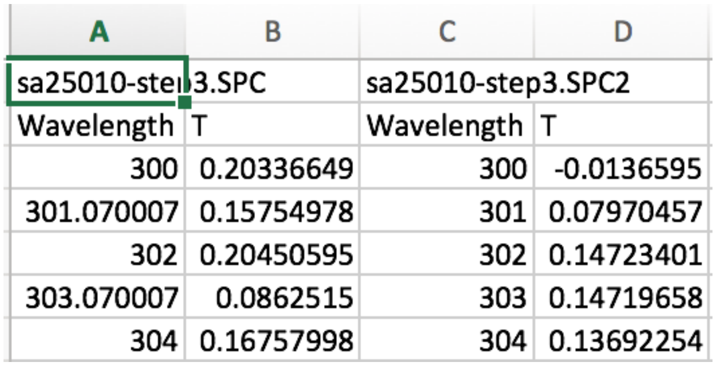
.csv header issue

First, the issue is in the header of the .csv files. If you notice, the header of the CyclicQUV-step1 is different from that of the others. And only this file is in correct form.



A	B	C	D
sa25000_EX	_280.00	sa25000_EX_285.00	
Wavelength	Intensity (a.u	Wavelength	Intensity (a.u
300	0.27582946	300	0.08241306
301.070007	0.09605753	301.070007	0.2071704
302	0.23337904	302	0.17644854
303.070007	0.23211677	303.070007	0.21992281
304	0.34277707	304	0.28013089

The rest of other files don't have the header which show the intensity data, as a result, they can't be process directly.

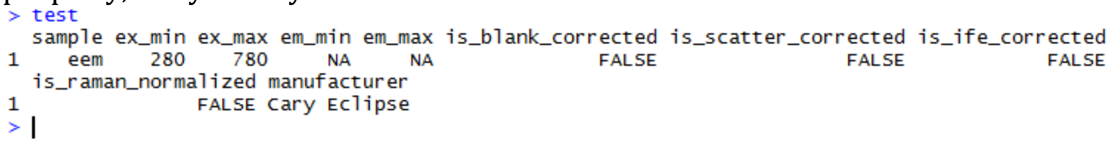


A	B	C	D
sa25010-step3.SPC		sa25010-step3.SPC2	
Wavelength	T	Wavelength	T
300	0.20336649	300	-0.0136595
301.070007	0.15754978	301	0.07970457
302	0.20450595	302	0.14723401
303.070007	0.0862515	303	0.14719658
304	0.16757998	304	0.13692254

To address this issue, several corrections need to be made.

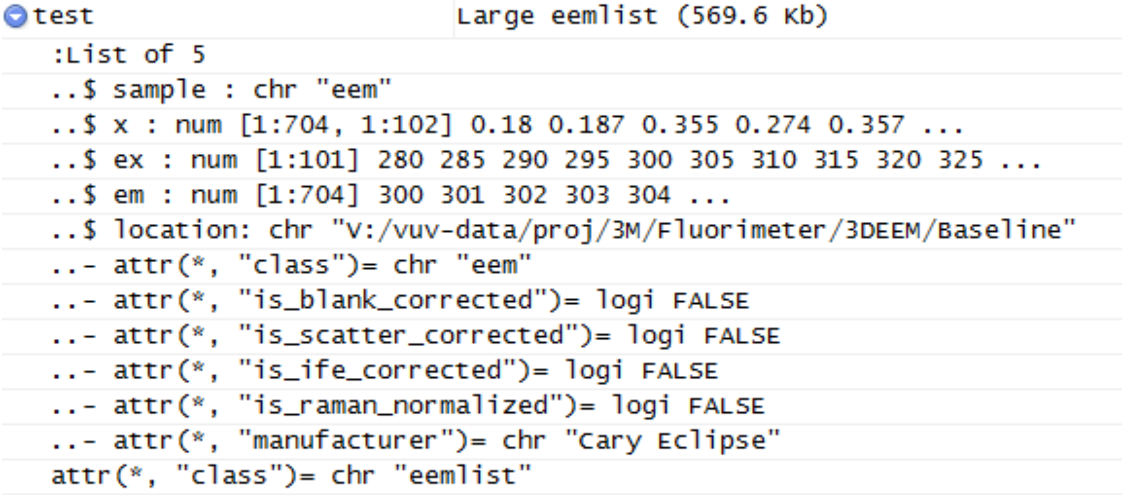
Header correction script

I write a correction script that takes the format of the first two header rows and makes the remainder of the .csv files look the same way. After the correction, the files can be read properly, but you may run into other errors.



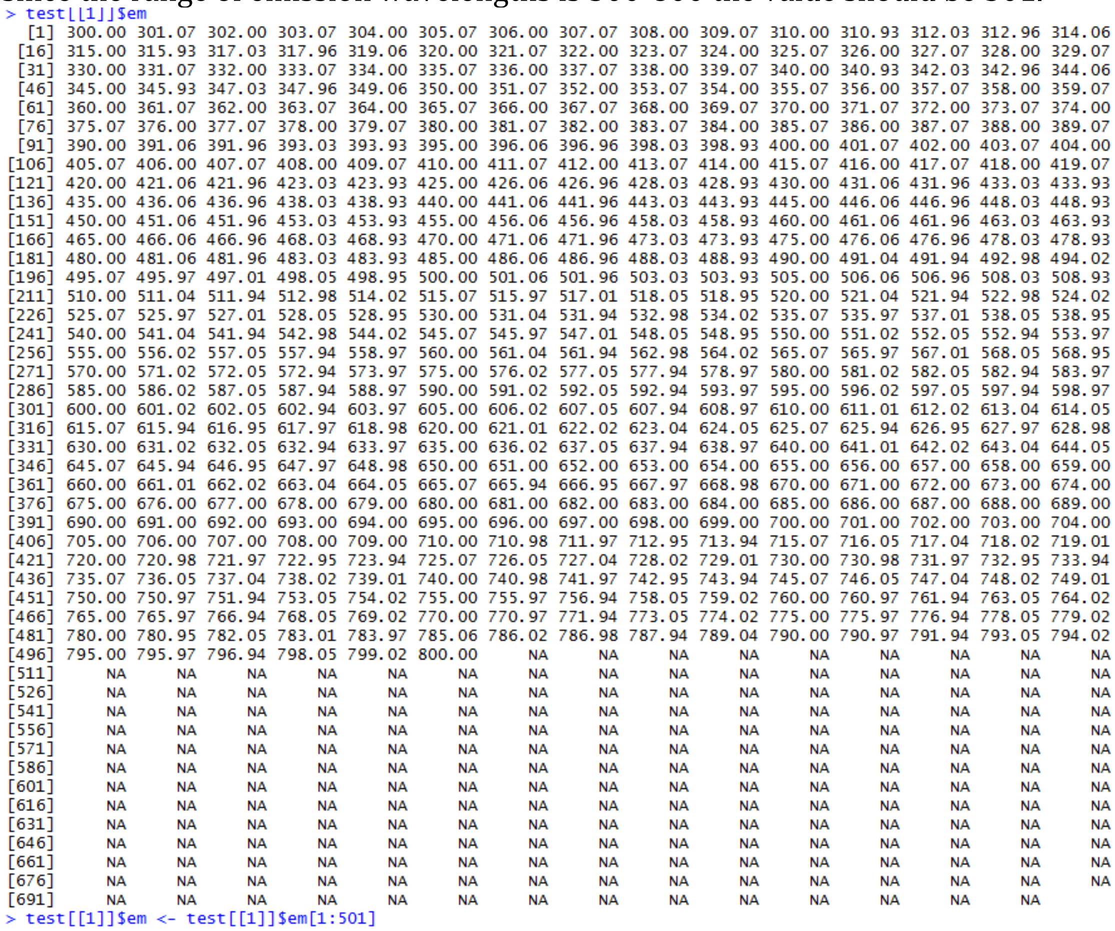
```
> test
  sample ex_min ex_max em_min em_max is_blank_corrected is_scatter_corrected is_ife_corrected
1    eem    280    780     NA     NA              FALSE                FALSE            FALSE
  is_raman_normalized manufacturer
1               FALSE Cary Eclipse
> |
```

This time, although the file successfully imported, the emission min and max are missing. So I look into the lists inside the eemlist to find the errors.



```
test                          Large eemlist (569.6 Kb)
    :List of 5
    ..$ sample : chr "eem"
    ..$ x : num [1:704, 1:102] 0.18 0.187 0.355 0.274 0.357 ...
    ..$ ex : num [1:101] 280 285 290 295 300 305 310 315 320 325 ...
    ..$ em : num [1:704] 300 301 302 303 304 ...
    ..$ location: chr "V:/vuv-data/proj/3M/Fluorimeter/3DEEM/Baseline"
    ..- attr(*, "class")= chr "eem"
    ..- attr(*, "is_blank_corrected")= logi FALSE
    ..- attr(*, "is_scatter_corrected")= logi FALSE
    ..- attr(*, "is_ife_corrected")= logi FALSE
    ..- attr(*, "is_raman_normalized")= logi FALSE
    ..- attr(*, "manufacturer")= chr "Cary Eclipse"
    attr(*, "class")= chr "eemlist"
```

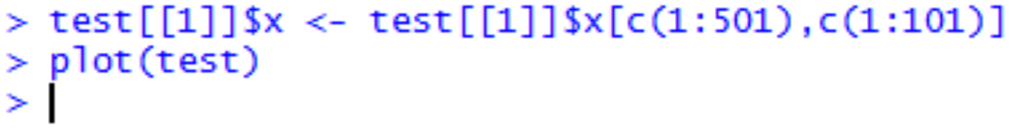
Note that the $em, which holds the emission wavelengths, has 704 values. This is incorrect, since the range of emission wavelengths is 300-800 the value should be 501.



```
> test[[1]]$em
  [1] 300.00 301.07 302.00 303.07 304.00 305.07 306.00 307.07 308.00 309.07 310.00 310.93 312.03 312.96 314.06
 [16] 315.00 315.93 317.03 317.96 319.06 320.00 321.07 322.00 323.07 324.00 325.07 326.00 327.07 328.00 329.07
 [31] 330.00 331.07 332.00 333.07 334.00 335.07 336.00 337.07 338.00 339.07 340.00 340.93 342.03 342.96 344.06
 [46] 345.00 345.93 347.03 347.96 349.06 350.00 351.07 352.00 353.07 354.00 355.07 356.00 357.07 358.00 359.07
 [61] 360.00 361.07 362.00 363.07 364.00 365.07 366.00 367.07 368.00 369.07 370.00 371.07 372.00 373.07 374.00
 [76] 375.07 376.00 377.07 378.00 379.07 380.00 381.07 382.00 383.07 384.00 385.07 386.00 387.07 388.00 389.07
 [91] 390.00 391.06 391.96 393.03 393.93 395.00 396.06 396.96 398.03 398.93 400.00 401.07 402.00 403.07 404.00
[106] 405.07 406.00 407.07 408.00 409.07 410.00 411.07 412.00 413.07 414.00 415.07 416.00 417.07 418.00 419.07
[121] 420.00 421.06 421.96 423.03 423.93 425.00 426.06 426.96 428.03 428.93 430.00 431.06 431.96 433.03 433.93
[136] 435.00 436.06 436.96 438.03 438.93 440.00 441.06 441.96 443.03 443.93 445.00 446.06 446.96 448.03 448.93
[151] 450.00 451.06 451.96 453.03 453.93 455.00 456.06 456.96 458.03 458.93 460.00 461.06 461.96 463.03 463.93
[166] 465.00 466.06 466.96 468.03 468.93 470.00 471.06 471.96 473.03 473.93 475.00 476.06 476.96 478.03 478.93
[181] 480.00 481.06 481.96 483.03 483.93 485.00 486.06 486.96 488.03 488.93 490.00 491.04 491.94 492.98 494.02
[196] 495.07 495.97 497.01 498.05 498.95 500.00 501.06 501.96 503.03 503.93 505.00 506.06 506.96 508.03 508.93
[211] 510.00 511.04 511.94 512.98 514.02 515.07 515.97 517.01 518.05 518.95 520.00 521.04 521.94 522.98 524.02
[226] 525.07 525.97 527.01 528.05 528.95 530.00 531.04 531.94 532.98 534.02 535.07 535.97 537.01 538.05 538.95
[241] 540.00 541.04 541.94 542.98 544.02 545.07 545.97 547.01 548.05 548.95 550.00 551.02 552.05 552.94 553.97
[256] 555.00 556.02 557.05 557.94 558.97 560.00 561.04 561.94 562.98 564.02 565.07 565.97 567.01 568.05 568.95
[271] 570.00 571.02 572.05 572.94 573.97 575.00 576.02 577.05 577.94 578.97 580.00 581.02 582.05 582.94 583.97
[286] 585.00 586.02 587.05 587.94 588.97 590.00 591.02 592.05 592.94 593.97 595.00 596.02 597.05 597.94 598.97
[301] 600.00 601.02 602.05 602.94 603.97 605.00 606.02 607.05 607.94 608.97 610.00 611.01 612.02 613.04 614.05
[316] 615.07 615.94 616.95 617.97 618.98 620.00 621.01 622.02 623.04 624.05 625.07 625.94 626.95 627.97 628.98
[331] 630.00 631.02 632.05 632.94 633.97 635.00 636.02 637.05 637.94 638.97 640.00 641.01 642.02 643.04 644.05
[346] 645.07 645.94 646.95 647.97 648.98 650.00 651.00 652.00 653.00 654.00 655.00 656.00 657.00 658.00 659.00
[361] 660.00 661.01 662.02 663.04 664.05 665.07 665.94 666.95 667.97 668.98 670.00 671.00 672.00 673.00 674.00
[376] 675.00 676.00 677.00 678.00 679.00 680.00 681.00 682.00 683.00 684.00 685.00 686.00 687.00 688.00 689.00
[391] 690.00 691.00 692.00 693.00 694.00 695.00 696.00 697.00 698.00 699.00 700.00 701.00 702.00 703.00 704.00
[406] 705.00 706.00 707.00 708.00 709.00 710.00 710.98 711.97 712.95 713.94 715.07 716.05 717.04 718.02 719.01
[421] 720.00 720.98 721.97 722.95 723.94 725.07 726.05 727.04 728.02 729.01 730.00 730.98 731.97 732.95 733.94
[436] 735.07 736.05 737.04 738.02 739.01 740.00 740.98 741.97 742.95 743.94 745.07 746.05 747.04 748.02 749.01
[451] 750.00 750.97 751.94 753.05 754.02 755.00 755.97 756.94 758.05 759.02 760.00 760.97 761.94 763.05 764.02
[466] 765.00 765.97 766.94 768.05 769.02 770.00 770.97 771.94 773.05 774.02 775.00 775.97 776.94 778.05 779.02
[481] 780.00 780.95 782.05 783.01 783.97 785.06 786.02 786.98 787.94 789.04 790.00 790.97 791.94 793.05 794.02
[496] 795.00 795.97 796.94 798.05 799.02 800.00     NA     NA     NA     NA     NA     NA     NA     NA     NA
[511]     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
[526]     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
[541]     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
[556]     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
[571]     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
[586]     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
[601]     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
[616]     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
[631]     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
[646]     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
[661]     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
[676]     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
[691]     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
> test[[1]]$em <- test[[1]]$em[1:501]
```

I correct this by addressing inside the eemlist and replacing that vector with only the first 501 values, and removing the NAs.

Now note in the $x matrix, where the fluorescence intensity values are stored, the number of values are 704 and 102. We already know the issue with the 704, we have to take only the first 501 values. And the 102 should be 101 because there were a total of 101 fluorescence spectra measured if you look inside the .csv; one spectrum was measured every 5nm from 300 to 800 in excitation, so 101 in total.



```
> test[[1]]$x <- test[[1]]$x[c(1:501),c(1:101)]
> plot(test)
> 
```

I correct this by addressing inside the eemlist and taking the proper dimensions and leaving out the NA data (you can check for the NAs yourself to see is they are present)

The plotting then works.

Data Visualization

CyclicQUV

Step 1

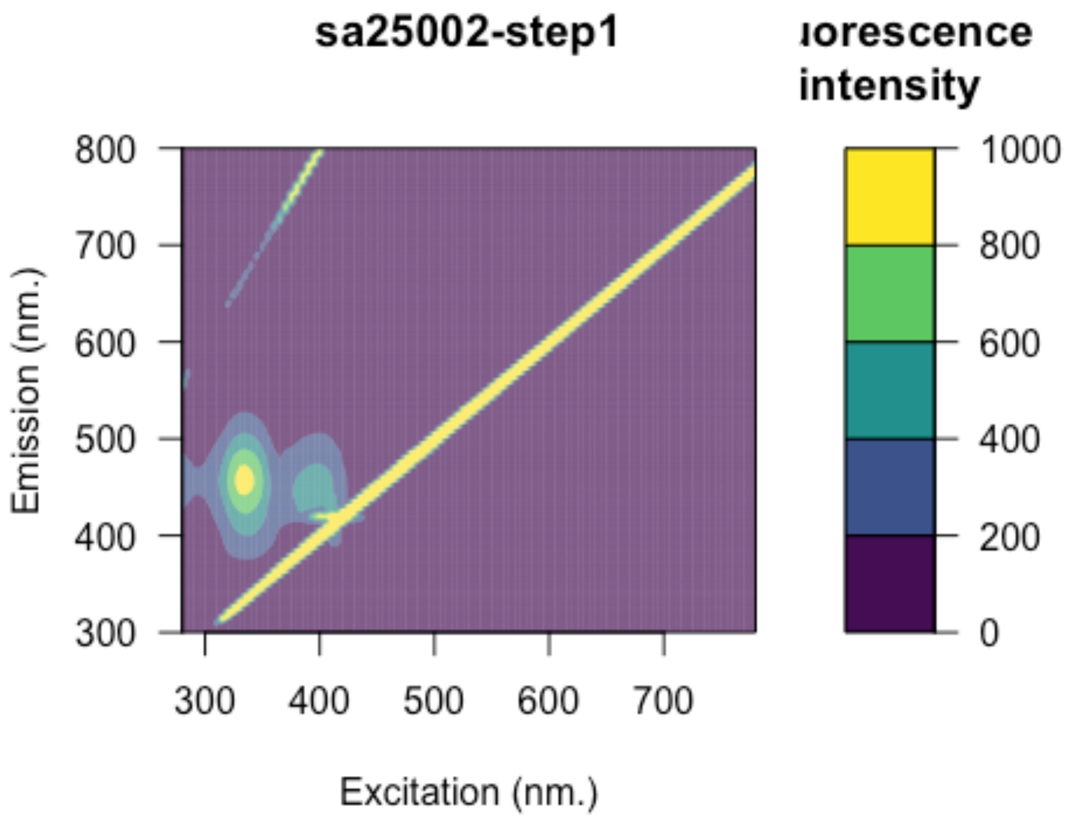
```
library(eemR)
ls("package:eemR")
```

```
##  [1] "absorbance"                "eem_bind"
##  [3] "eem_biological_index"      "eem_coble_peaks"
##  [5] "eem_cut"                   "eem_export_matlab"
##  [7] "eem_extract"               "eem_fluorescence_index"
##  [9] "eem_humification_index"    "eem_inner_filter_effect"
## [11] "eem_names"                 "eem_names<-"
## [13] "eem_raman_normalisation" "eem_read"
## [15] "eem_remove_blank"          "eem_remove_scattering"
## [17] "eem_set_wavelengths"
```

```
cstep1 <- system.file("extdata/cary/CyclicQUV/step1/3DEEM/", package = "eemR"
)
eemsc1 <- eem_read(cstep1)

summary(eemsc1)
```

```
##           sample ex_min ex_max em_min em_max is_blank_corrected
## 1 sa25000-step1    280    780    300    800              FALSE
## 2 sa25002-step1    280    780    300    800              FALSE
## 3 sa25004-step1    280    780    300    800              FALSE
## 4 sa25006-step1    280    780    300    800              FALSE
## 5 sa25008-step1    280    780    300    800              FALSE
## 6 sa25010-step1    280    780    300    800              FALSE
##   is_scatter_corrected is_ife_corrected is_raman_normalized manufacturer
## 1                FALSE            FALSE               FALSE Cary Eclipse
## 2                FALSE            FALSE               FALSE Cary Eclipse
## 3                FALSE            FALSE               FALSE Cary Eclipse
## 4                FALSE            FALSE               FALSE Cary Eclipse
## 5                FALSE            FALSE               FALSE Cary Eclipse
## 6                FALSE            FALSE               FALSE Cary Eclipse
```

```
plot(eemsc1, which=2)
```

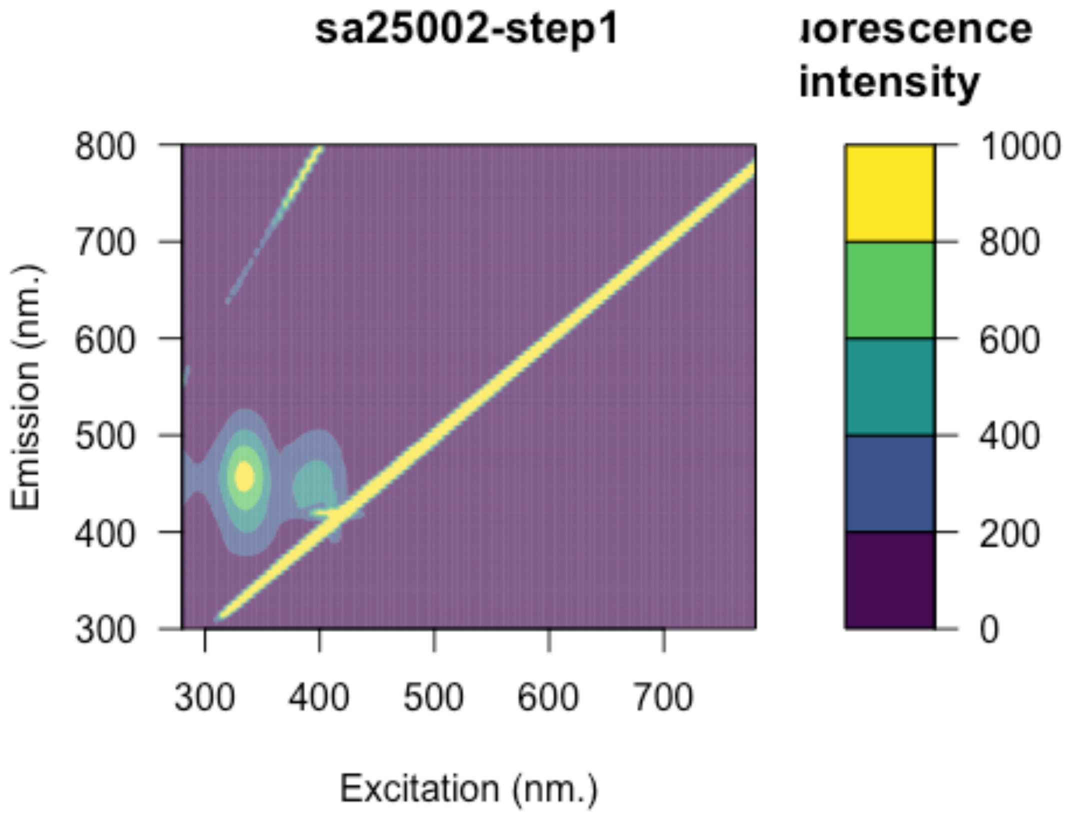


Step 2

```
cstep2 <- system.file("extdata/cary/CyclicQUV/step2/3DEEM/", package = "eemR"
)
eemsc2 <- eem_read(cstep2)

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion
```

```
eemsc2[[1]]$em<-eemsc2[[1]]$em[1:501]
eemsc2[[1]]$x<-eemsc2[[1]]$x[c(1:501),c(1:101)]
eemsc2[[2]]$em<-eemsc2[[2]]$em[1:501]
eemsc2[[2]]$x<-eemsc2[[2]]$x[c(1:501),c(1:101)]
eemsc2[[3]]$em<-eemsc2[[3]]$em[1:501]
eemsc2[[3]]$x<-eemsc2[[3]]$x[c(1:501),c(1:101)]
eemsc2[[4]]$em<-eemsc2[[4]]$em[1:501]
eemsc2[[4]]$x<-eemsc2[[4]]$x[c(1:501),c(1:101)]
eemsc2[[5]]$em<-eemsc2[[5]]$em[1:501]
eemsc2[[5]]$x<-eemsc2[[5]]$x[c(1:501),c(1:101)]
eemsc2[[6]]$em<-eemsc2[[6]]$em[1:501]
eemsc2[[6]]$x<-eemsc2[[6]]$x[c(1:501),c(1:101)]

plot(eemsc2)
```

```
plot(eemsc2, which=2)
```



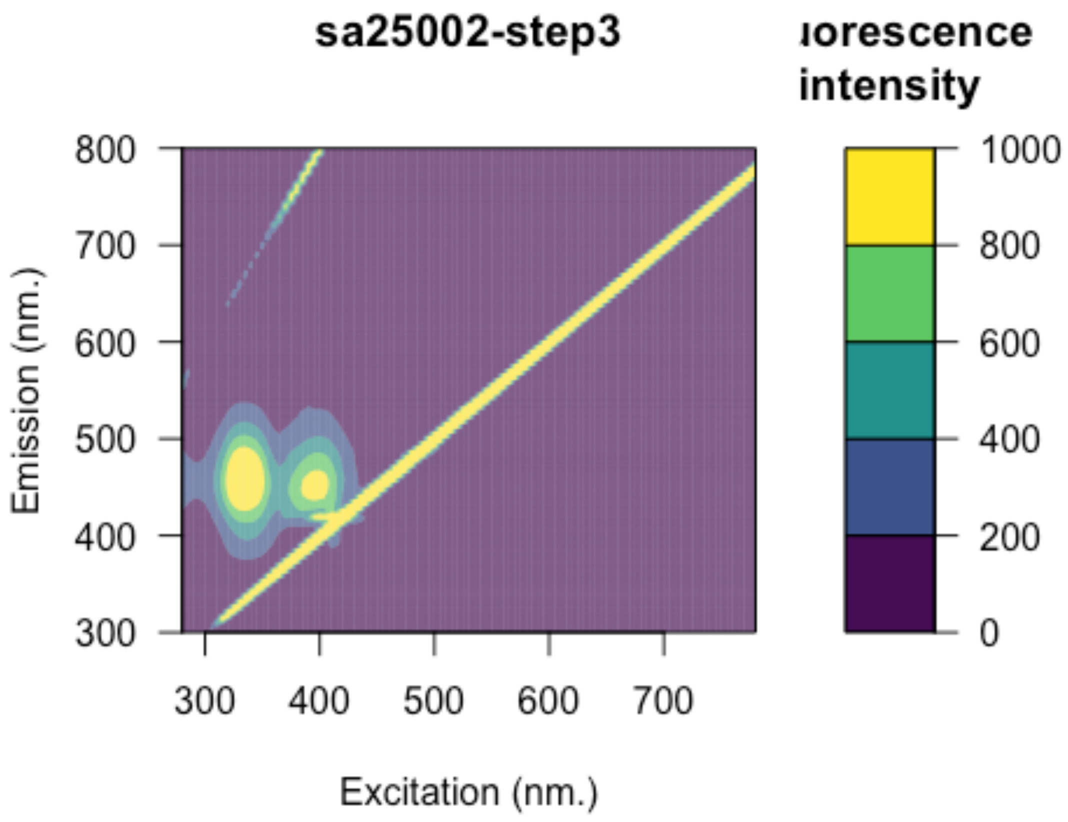
Step 3

```
cstep3 <- system.file("extdata/cary/CyclicQUV/step3/3DEEM/", package = "eemR"
)
eemsc3 <- eem_read(cstep3)
```

```
## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion
```

```
eemsc3[[1]]$em<-eemsc3[[1]]$em[1:501]
eemsc3[[1]]$x<-eemsc3[[1]]$x[c(1:501),c(1:101)]
eemsc3[[2]]$em<-eemsc3[[2]]$em[1:501]
eemsc3[[2]]$x<-eemsc3[[2]]$x[c(1:501),c(1:101)]
eemsc3[[3]]$em<-eemsc3[[3]]$em[1:501]
eemsc3[[3]]$x<-eemsc3[[3]]$x[c(1:501),c(1:101)]
eemsc3[[4]]$em<-eemsc3[[4]]$em[1:501]
eemsc3[[4]]$x<-eemsc3[[4]]$x[c(1:501),c(1:101)]
eemsc3[[5]]$em<-eemsc3[[5]]$em[1:501]
eemsc3[[5]]$x<-eemsc3[[5]]$x[c(1:501),c(1:101)]
eemsc3[[6]]$em<-eemsc3[[6]]$em[1:501]
eemsc3[[6]]$x<-eemsc3[[6]]$x[c(1:501),c(1:101)]

plot(eemsc3)
```

```
plot(eemsc3, which=2)
```

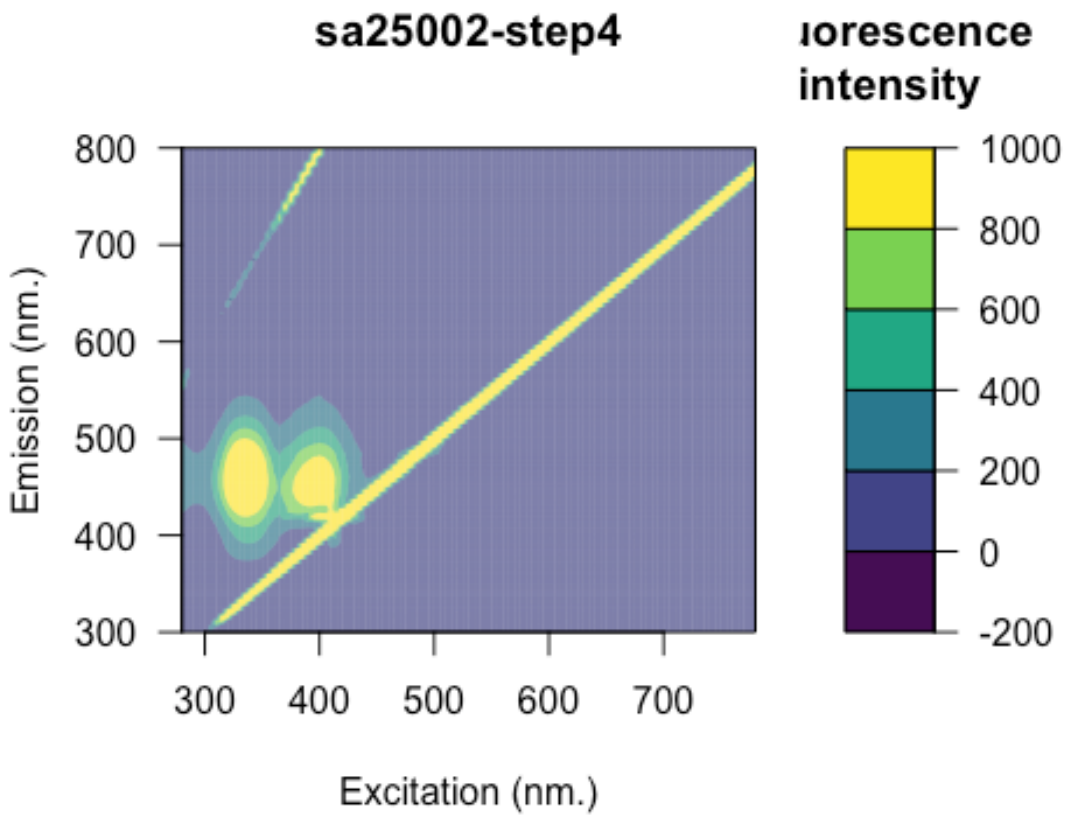


Step 4

```
cstep4 <- system.file("extdata/cary/CyclicQUV/step4/3DEEM/", package = "eemR"
)
eemsc4 <- eem_read(cstep4)

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion
```

```
eemsc4[[1]]$em<-eemsc4[[1]]$em[1:501]
eemsc4[[1]]$x<-eemsc4[[1]]$x[c(1:501),c(1:101)]
eemsc4[[2]]$em<-eemsc4[[2]]$em[1:501]
eemsc4[[2]]$x<-eemsc4[[2]]$x[c(1:501),c(1:101)]
eemsc4[[3]]$em<-eemsc4[[3]]$em[1:501]
eemsc4[[3]]$x<-eemsc4[[3]]$x[c(1:501),c(1:101)]
eemsc4[[4]]$em<-eemsc4[[4]]$em[1:501]
eemsc4[[4]]$x<-eemsc4[[4]]$x[c(1:501),c(1:101)]
eemsc4[[5]]$em<-eemsc4[[5]]$em[1:501]
eemsc4[[5]]$x<-eemsc4[[5]]$x[c(1:501),c(1:101)]
eemsc4[[6]]$em<-eemsc4[[6]]$em[1:501]
eemsc4[[6]]$x<-eemsc4[[6]]$x[c(1:501),c(1:101)]

plot(eemsc4)
```

```
plot(eemsc4, which=2)
```

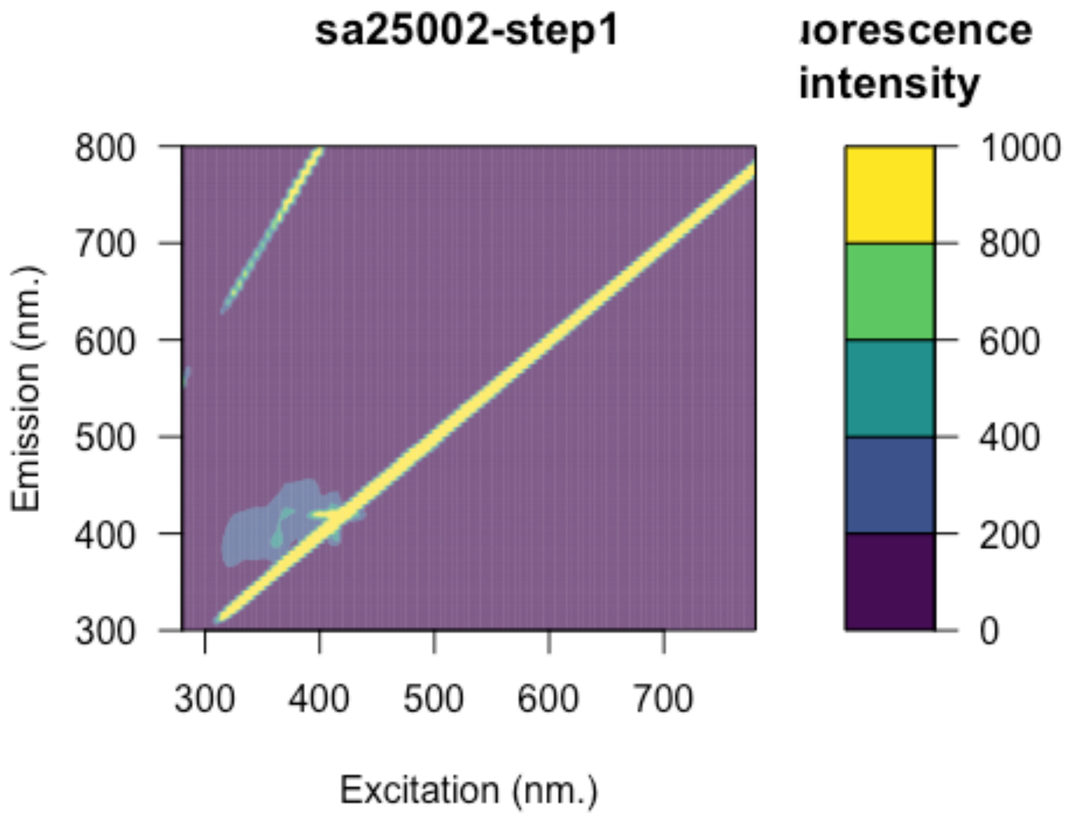


DampHeat

Step 1

```
library(eemR)
ls("package:eemR")

##  [1] "absorbance"                "eem_bind"
##  [3] "eem_biological_index"      "eem_coble_peaks"
##  [5] "eem_cut"                   "eem_export_matlab"
##  [7] "eem_extract"               "eem_fluorescence_index"
##  [9] "eem_humification_index"    "eem_inner_filter_effect"
## [11] "eem_names"                 "eem_names<-"
## [13] "eem_raman_normalisation"  "eem_read"
## [15] "eem_remove_blank"          "eem_remove_scattering"
## [17] "eem_set_wavelengths"

dstep1 <- system.file("extdata/cary/DampHeat/step1/3DEEM/", package = "eemR")
eemsd1 <- eem_read(dstep1)

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

eemsd1[[1]]$em<-eemsd1[[1]]$em[1:501]
eemsd1[[1]]$x<-eemsd1[[1]]$x[c(1:501),c(1:101)]
eemsd1[[2]]$em<-eemsd1[[2]]$em[1:501]
eemsd1[[2]]$x<-eemsd1[[2]]$x[c(1:501),c(1:101)]
eemsd1[[3]]$em<-eemsd1[[3]]$em[1:501]
eemsd1[[3]]$x<-eemsd1[[3]]$x[c(1:501),c(1:101)]
eemsd1[[4]]$em<-eemsd1[[4]]$em[1:501]
eemsd1[[4]]$x<-eemsd1[[4]]$x[c(1:501),c(1:101)]
eemsd1[[5]]$em<-eemsd1[[5]]$em[1:501]
eemsd1[[5]]$x<-eemsd1[[5]]$x[c(1:501),c(1:101)]
```

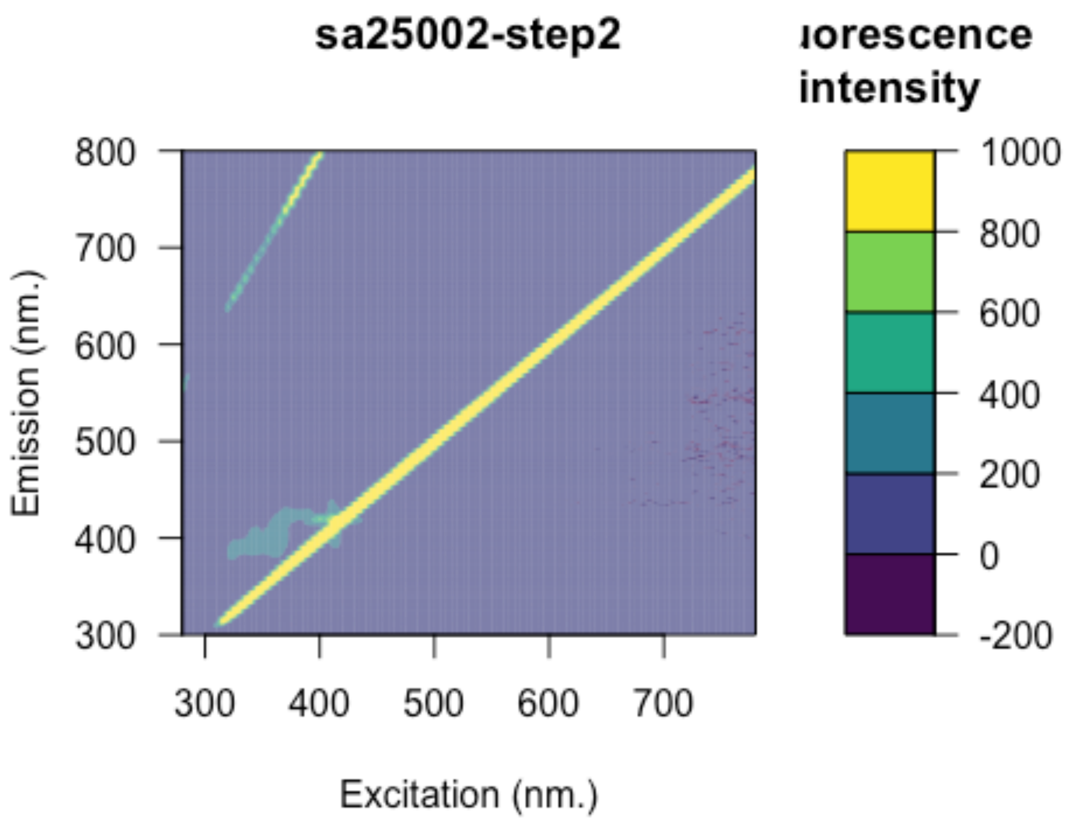
```
eemsd1[[6]]$em<-eemsd1[[6]]$em[1:501]
eemsd1[[6]]$x<-eemsd1[[6]]$x[c(1:501),c(1:101)]

plot(eemsd1, which=2)
```



Step 2

```
dstep2 <- system.file("extdata/cary/DampHeat/step2/3DEEM/", package = "eemR")
eemsd2 <- eem_read(dstep2)

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion
```

```
eemsd2[[1]]$em<-eemsd2[[1]]$em[1:501]
eemsd2[[1]]$x<-eemsd2[[1]]$x[c(1:501),c(1:101)]
eemsd2[[2]]$em<-eemsd2[[2]]$em[1:501]
eemsd2[[2]]$x<-eemsd2[[2]]$x[c(1:501),c(1:101)]
eemsd2[[3]]$em<-eemsd2[[3]]$em[1:501]
eemsd2[[3]]$x<-eemsd2[[3]]$x[c(1:501),c(1:101)]
eemsd2[[4]]$em<-eemsd2[[4]]$em[1:501]
eemsd2[[4]]$x<-eemsd2[[4]]$x[c(1:501),c(1:101)]
eemsd2[[5]]$em<-eemsd2[[5]]$em[1:501]
eemsd2[[5]]$x<-eemsd2[[5]]$x[c(1:501),c(1:101)]
eemsd2[[6]]$em<-eemsd2[[6]]$em[1:501]
eemsd2[[6]]$x<-eemsd2[[6]]$x[c(1:501),c(1:101)]
plot(eemsd2, which=2)
```

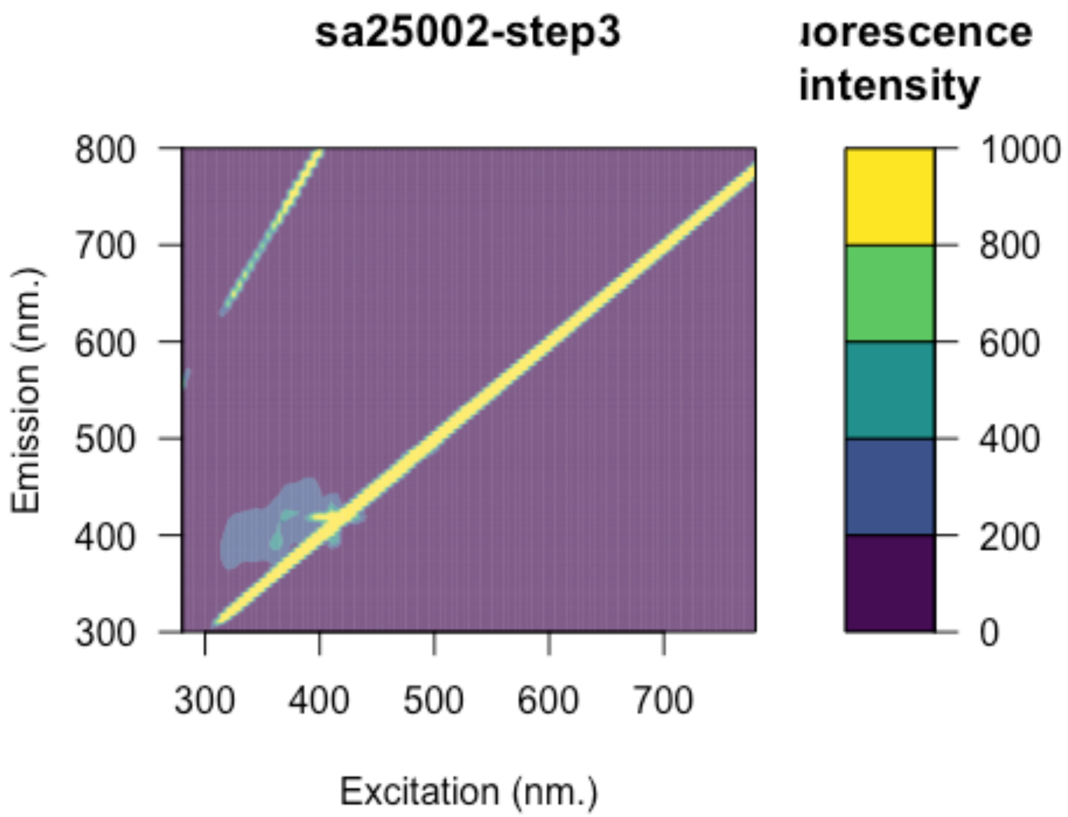


Step 3

```
dstep3 <- system.file("extdata/cary/DampHeat/step3/3DEEM/", package = "eemR")
eemsd3 <- eem_read(dstep3)
```

```
## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion
```

```
eemsd3[[1]]$em<-eemsd3[[1]]$em[1:501]
eemsd3[[1]]$x<-eemsd3[[1]]$x[c(1:501),c(1:101)]
eemsd3[[2]]$em<-eemsd3[[2]]$em[1:501]
eemsd3[[2]]$x<-eemsd3[[2]]$x[c(1:501),c(1:101)]
eemsd3[[3]]$em<-eemsd3[[3]]$em[1:501]
eemsd3[[3]]$x<-eemsd3[[3]]$x[c(1:501),c(1:101)]
eemsd3[[4]]$em<-eemsd3[[4]]$em[1:501]
eemsd3[[4]]$x<-eemsd3[[4]]$x[c(1:501),c(1:101)]
eemsd3[[5]]$em<-eemsd3[[5]]$em[1:501]
eemsd3[[5]]$x<-eemsd3[[5]]$x[c(1:501),c(1:101)]
eemsd3[[6]]$em<-eemsd3[[6]]$em[1:501]
eemsd3[[6]]$x<-eemsd3[[6]]$x[c(1:501),c(1:101)]
plot(eemsd3, which=2)
```

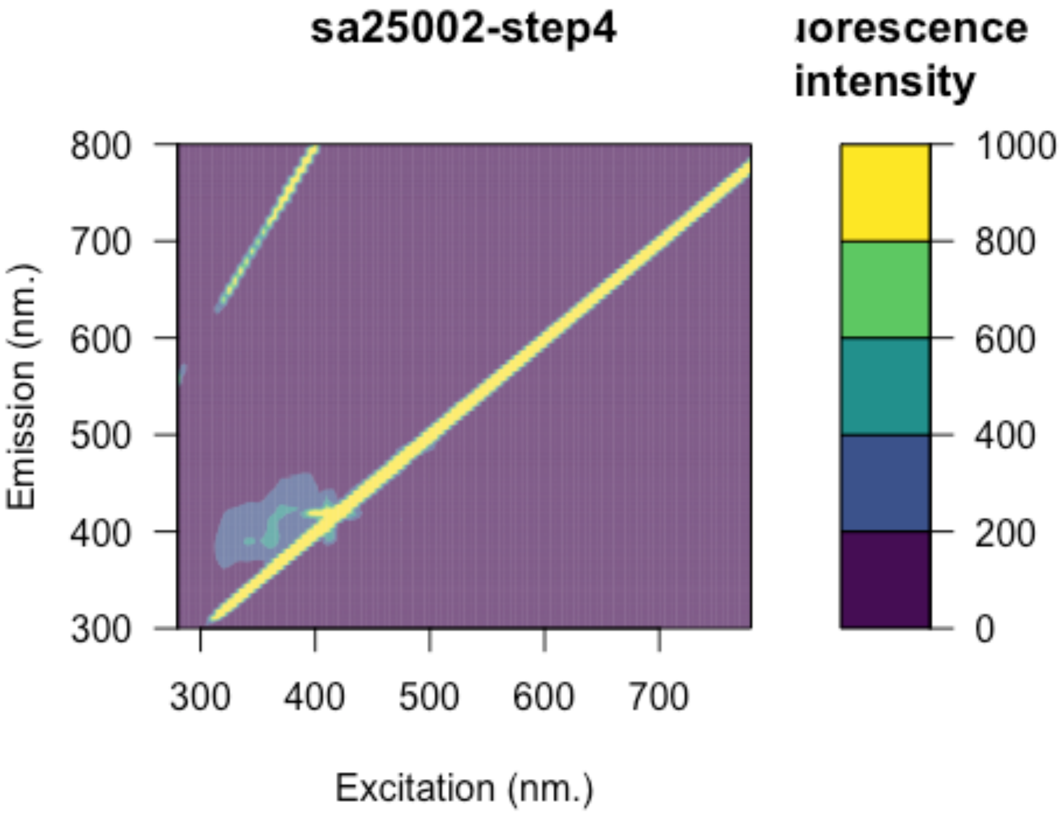


Step 4

```
dstep4 <- system.file("extdata/cary/DampHeat/step4/3DEEM/", package = "eemR")
eemsd4 <- eem_read(dstep4)
```

```
## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion
```

```
eemsd4[[1]]$em<-eemsd4[[1]]$em[1:501]
eemsd4[[1]]$x<-eemsd4[[1]]$x[c(1:501),c(1:101)]
eemsd4[[2]]$em<-eemsd4[[2]]$em[1:501]
eemsd4[[2]]$x<-eemsd4[[2]]$x[c(1:501),c(1:101)]
eemsd4[[3]]$em<-eemsd4[[3]]$em[1:501]
eemsd4[[3]]$x<-eemsd4[[3]]$x[c(1:501),c(1:101)]
eemsd4[[4]]$em<-eemsd4[[4]]$em[1:501]
eemsd4[[4]]$x<-eemsd4[[4]]$x[c(1:501),c(1:101)]
eemsd4[[5]]$em<-eemsd4[[5]]$em[1:501]
eemsd4[[5]]$x<-eemsd4[[5]]$x[c(1:501),c(1:101)]
eemsd4[[6]]$em<-eemsd4[[6]]$em[1:501]
eemsd4[[6]]$x<-eemsd4[[6]]$x[c(1:501),c(1:101)]
plot(eemsd4, which=2)
```



HotQUV

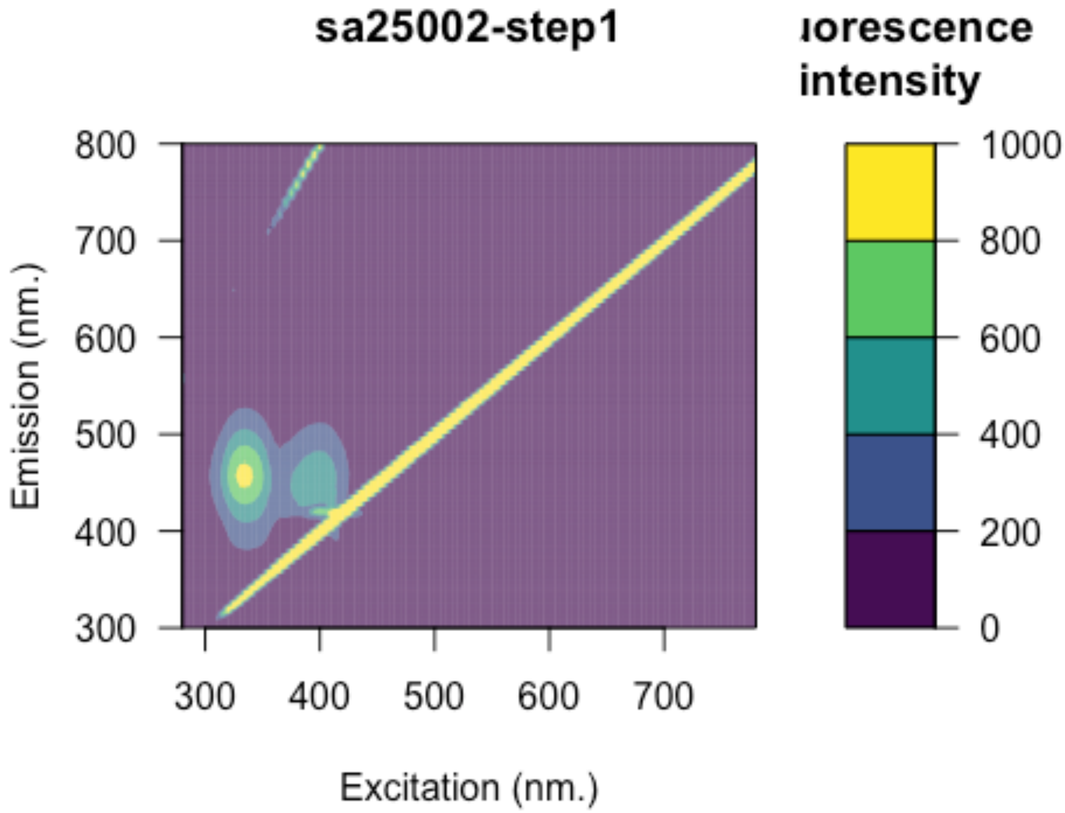
Step 1

```
library(eemR)
ls("package:eemR")

##  [1] "absorbance"              "eem_bind"
##  [3] "eem_biological_index"    "eem_coble_peaks"
##  [5] "eem_cut"                  "eem_export_matlab"
##  [7] "eem_extract"              "eem_fluorescence_index"
##  [9] "eem_humification_index"   "eem_inner_filter_effect"
## [11] "eem_names"                "eem_names<-"
## [13] "eem_raman_normalisation"  "eem_read"
## [15] "eem_remove_blank"         "eem_remove_scattering"
## [17] "eem_set_wavelengths"
```

```
hstep1 <- system.file("extdata/cary/HotQUV/step1/3DEEM/", package = "eemR")
eemsh1 <- eem_read(hstep1)
```

```
## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion
```

```
eemsh1[[1]]$em<-eemsh1[[1]]$em[1:501]
eemsh1[[1]]$x<-eemsh1[[1]]$x[c(1:501),c(1:101)]
eemsh1[[2]]$em<-eemsh1[[2]]$em[1:501]
eemsh1[[2]]$x<-eemsh1[[2]]$x[c(1:501),c(1:101)]
eemsh1[[3]]$em<-eemsh1[[3]]$em[1:501]
eemsh1[[3]]$x<-eemsh1[[3]]$x[c(1:501),c(1:101)]
eemsh1[[4]]$em<-eemsh1[[4]]$em[1:501]
eemsh1[[4]]$x<-eemsh1[[4]]$x[c(1:501),c(1:101)]
eemsh1[[5]]$em<-eemsh1[[5]]$em[1:501]
eemsh1[[5]]$x<-eemsh1[[5]]$x[c(1:501),c(1:101)]
eemsh1[[6]]$em<-eemsh1[[6]]$em[1:501]
eemsh1[[6]]$x<-eemsh1[[6]]$x[c(1:501),c(1:101)]

plot(eemsh1, which=2)
```

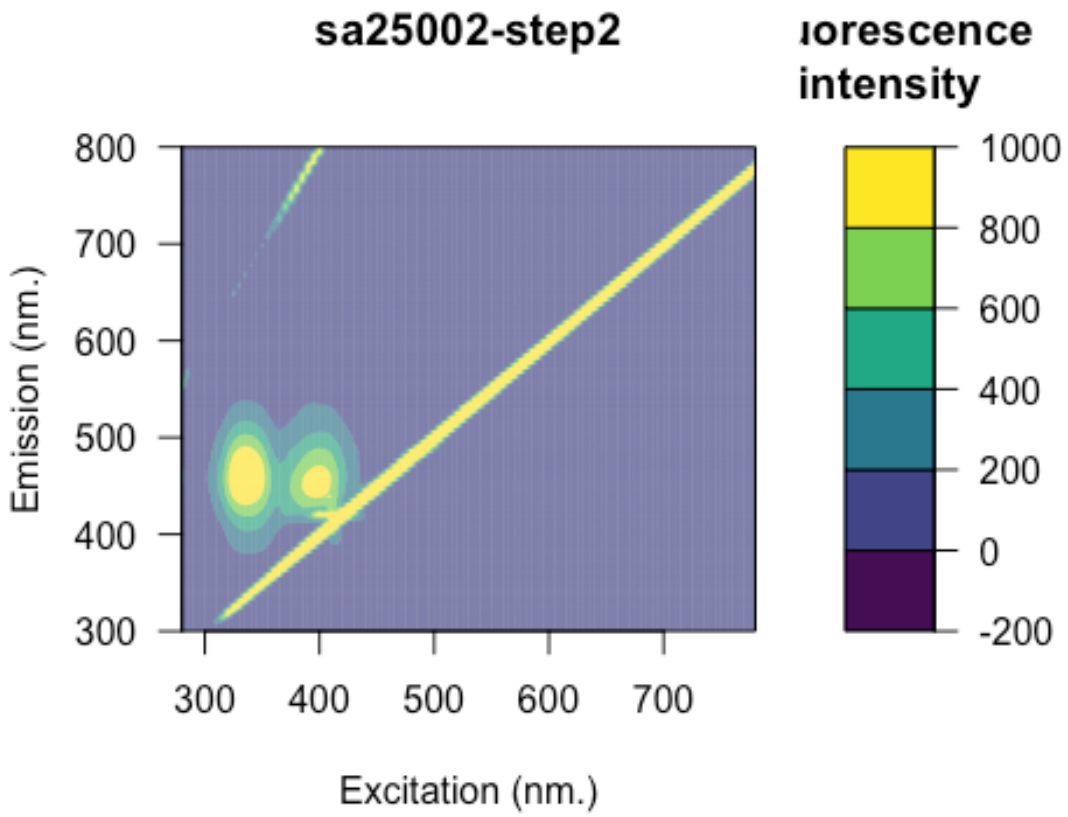


Step 2

```
hstep2 <- system.file("extdata/cary/HotQUV/step2/3DEEM/", package = "eemR")
eemsh2 <- eem_read(hstep2)
```

```
## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion
```

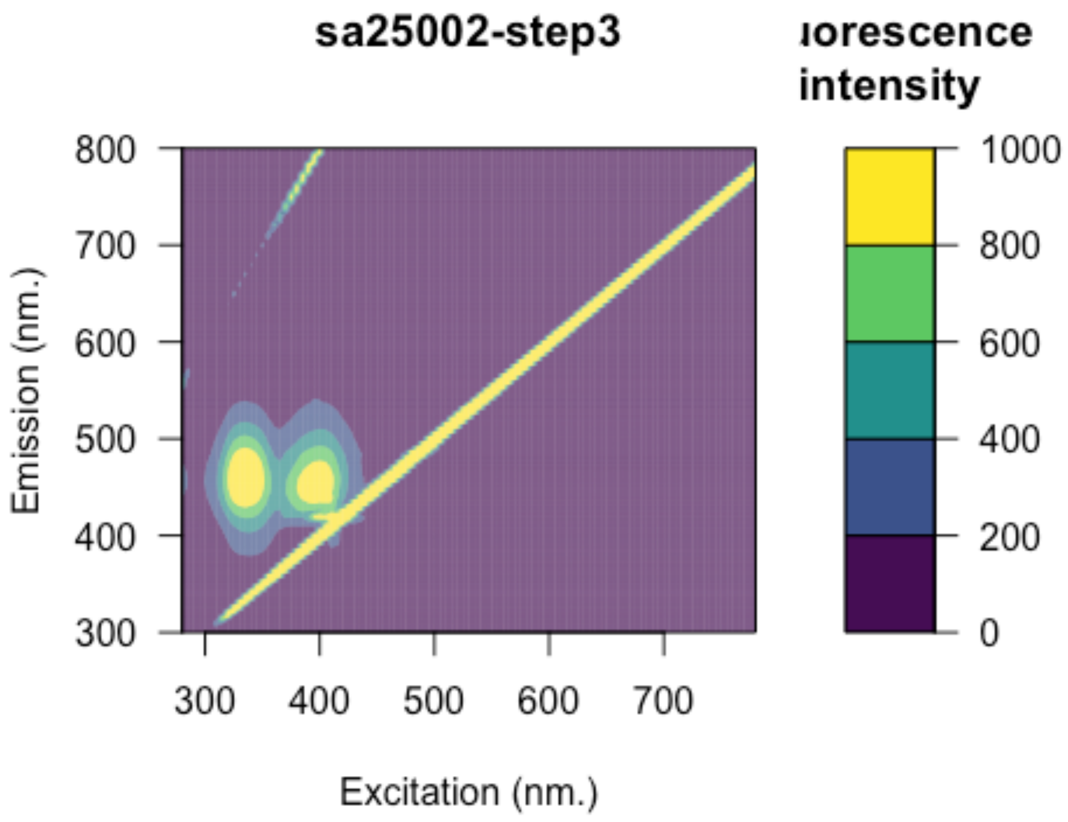
```
eemsh2[[1]]$em<-eemsh2[[1]]$em[1:501]
eemsh2[[1]]$x<-eemsh2[[1]]$x[c(1:501),c(1:101)]
eemsh2[[2]]$em<-eemsh2[[2]]$em[1:501]
eemsh2[[2]]$x<-eemsh2[[2]]$x[c(1:501),c(1:101)]
eemsh2[[3]]$em<-eemsh2[[3]]$em[1:501]
eemsh2[[3]]$x<-eemsh2[[3]]$x[c(1:501),c(1:101)]
eemsh2[[4]]$em<-eemsh2[[4]]$em[1:501]
eemsh2[[4]]$x<-eemsh2[[4]]$x[c(1:501),c(1:101)]
eemsh2[[5]]$em<-eemsh2[[5]]$em[1:501]
eemsh2[[5]]$x<-eemsh2[[5]]$x[c(1:501),c(1:101)]
eemsh2[[6]]$em<-eemsh2[[6]]$em[1:501]
eemsh2[[6]]$x<-eemsh2[[6]]$x[c(1:501),c(1:101)]
plot(eemsh2, which=2)
```



Step 3

```
hstep3 <- system.file("extdata/cary/HotQUV/step3/3DEEM/", package = "eemR")
eemsh3 <- eem_read(hstep3)

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion
## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion
```

```
eemsh3[[1]]$em<-eemsh3[[1]]$em[1:501]
eemsh3[[1]]$x<-eemsh3[[1]]$x[c(1:501),c(1:101)]
eemsh3[[2]]$em<-eemsh3[[2]]$em[1:501]
eemsh3[[2]]$x<-eemsh3[[2]]$x[c(1:501),c(1:101)]
eemsh3[[3]]$em<-eemsh3[[3]]$em[1:501]
eemsh3[[3]]$x<-eemsh3[[3]]$x[c(1:501),c(1:101)]
eemsh3[[4]]$em<-eemsh3[[4]]$em[1:501]
eemsh3[[4]]$x<-eemsh3[[4]]$x[c(1:501),c(1:101)]
eemsh3[[5]]$em<-eemsh3[[5]]$em[1:501]
eemsh3[[5]]$x<-eemsh3[[5]]$x[c(1:501),c(1:101)]
eemsh3[[6]]$em<-eemsh3[[6]]$em[1:501]
eemsh3[[6]]$x<-eemsh3[[6]]$x[c(1:501),c(1:101)]
plot(eemsh3, which=2)
```

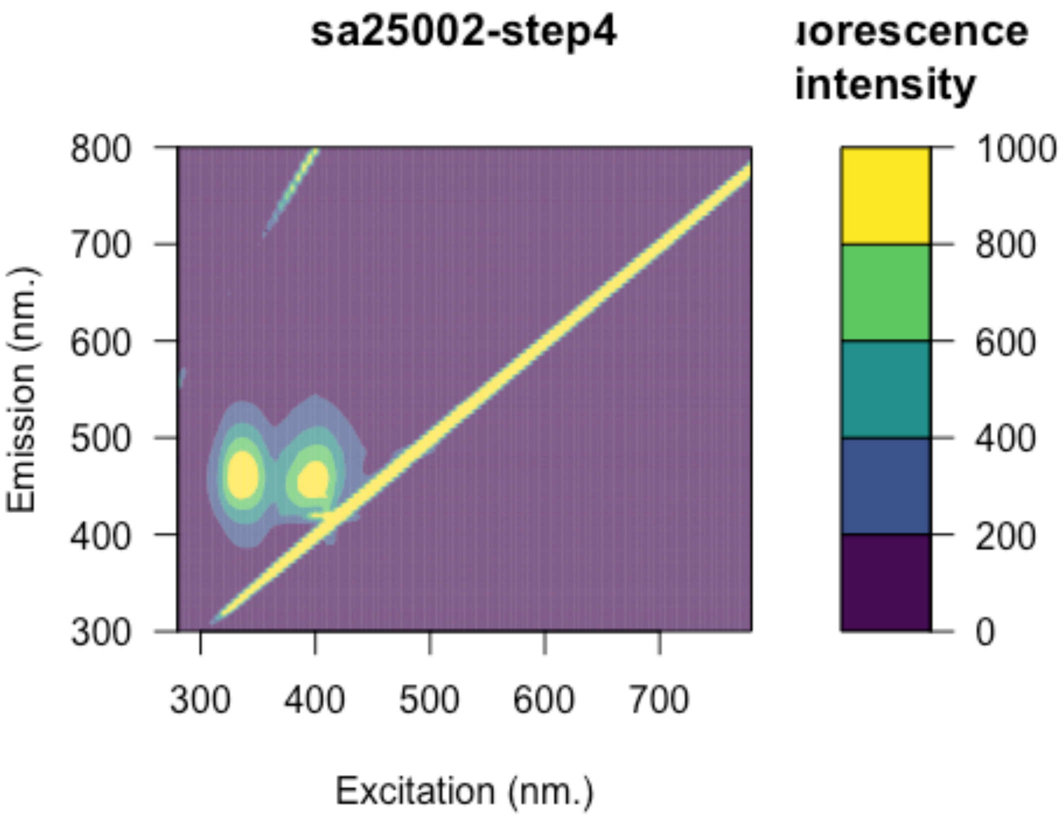


Step 4

```
hstep4 <- system.file("extdata/cary/HotQUV/step4/3DEEM/", package = "eemR")
eemsh4 <- eem_read(hstep4)
```

```
## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion
```

```
eemsh4[[1]]$em<-eemsh4[[1]]$em[1:501]
eemsh4[[1]]$x<-eemsh4[[1]]$x[c(1:501),c(1:101)]
eemsh4[[2]]$em<-eemsh4[[2]]$em[1:501]
eemsh4[[2]]$x<-eemsh4[[2]]$x[c(1:501),c(1:101)]
eemsh4[[3]]$em<-eemsh4[[3]]$em[1:501]
eemsh4[[3]]$x<-eemsh4[[3]]$x[c(1:501),c(1:101)]
eemsh4[[4]]$em<-eemsh4[[4]]$em[1:501]
eemsh4[[4]]$x<-eemsh4[[4]]$x[c(1:501),c(1:101)]
eemsh4[[5]]$em<-eemsh4[[5]]$em[1:501]
eemsh4[[5]]$x<-eemsh4[[5]]$x[c(1:501),c(1:101)]
eemsh4[[6]]$em<-eemsh4[[6]]$em[1:501]
eemsh4[[6]]$x<-eemsh4[[6]]$x[c(1:501),c(1:101)]
plot(eemsh4, which=2)
```



#Study of Second-order emission spectrum

"Another source of unwanted light is higher-order light diffraction by the monochromator. Light diffraction at the grating can occur as a first-, second-, or higher-order process. These diffraction orders frequently overlap. Suppose the excitation monochromator is set at 300 nm. The xenon light source contains output of both 300 and 600 nm. When the monochromator is set at 600 nm, some 300-nm light can be present at the exit slit due to second-order diffraction. Hence, when recording an emission spectrum from a turbid solution, one will often observe a peak at twice the excitation wavelength due to second-order transmission through the excitation monochromator. For this reason, we often use bandpass excitation filters to remove unwanted wavelengths from the excitation beam. Transmission of the second-order diffraction can also result in extraneous light passing through the emission monochromators. Suppose the excitation is at 300 nm, and the emission monochromator is scanned through 600 nm. If the sample is strongly scattering, then some of the scattered light at 300 nm can appear as second-order diffraction when the emission monochromator is set to 600 nm." (Lakowicz, Joseph.Principles of Fluorescence Spectroscopy, 1983)

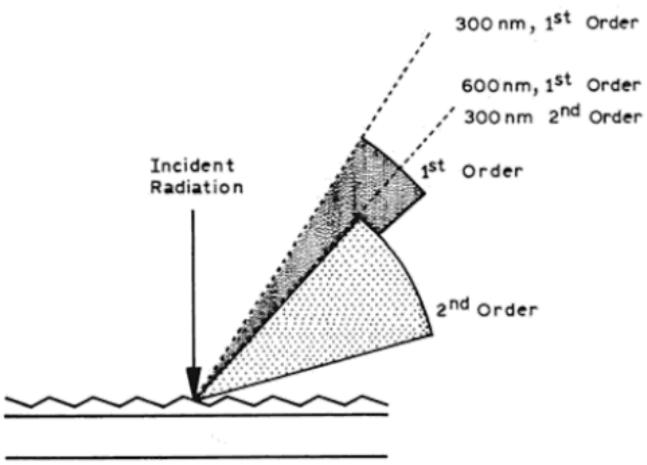


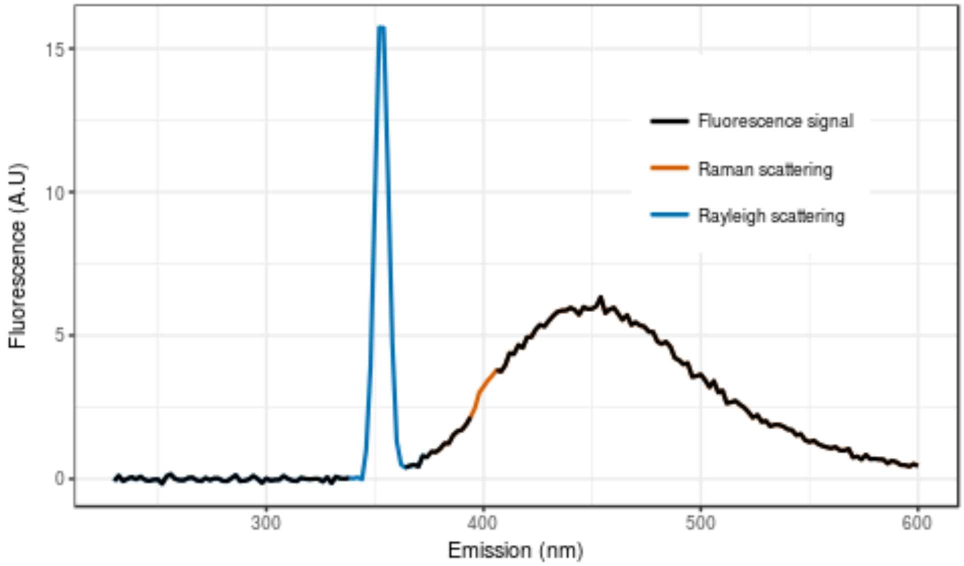
Figure 2.14. First- and second-order diffraction off a diffraction grating. The region where there is no overlap of the first- and second-order diffraction is called the free spectral range.

Second Order

Definition of Raleigh/Raman scatter

Scattering removal：Remove the the so-called scattering bands caused by first and second order of Raman and Rayleigh scattering.

Rayleigh and Raman scattering are optical processes by which some of the incident energy can be absorbed and converted into vibrational and rotational energy (Lakowicz 2006). The resulting scattered energy produce the so-called scattering bands which are visually easily identifiable. Given that both types of scattering are repeated across EEMs, it is important to remove such artifacts prior to analysis (Bahram et al. 2006; Zepp, Sheldon, and Moran 2004).



Emission fluorescence emitted at excitation ex=350. First order of Rayleigh and Raman scattering regions are identified in blue and red.

First order of Rayleigh scattering is defined as the region where emission is equal to excitation (em=ex) causing a diagonal band in the EEM (excitation-emission fluorescence matrix) whereas the second order of Rayleigh scattering occurs at two times the emission wavelength of the primary peak (em=2ex). Mathematically, first order Raman scattering is defined as follow:

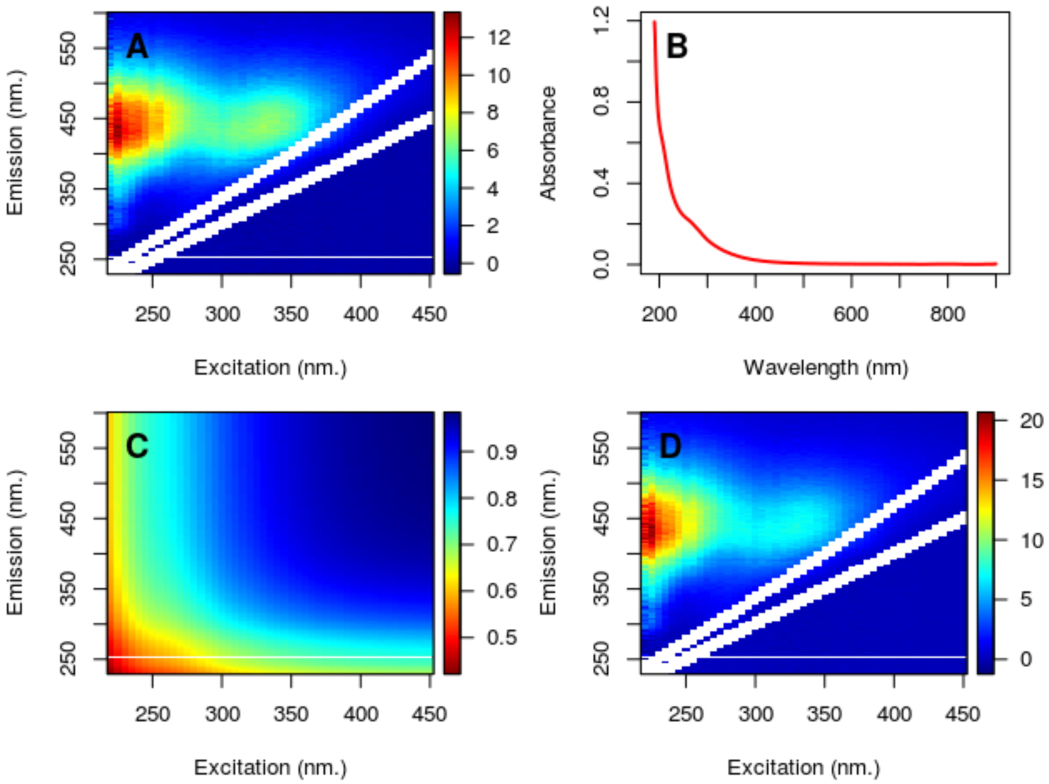
$$\text{Raman}_{\text{1st}} = -\frac{ex}{0.00036ex - 1}$$

where ex is the incident excitation wavelength (nm). Second order Raman scattering is then simply defined as:

$$\text{Raman}_{\text{2nd}} = -\frac{2ex}{0.00036ex - 1}$$

Different interpolation techniques have been proposed to eliminate scattering (Zepp, Sheldon, and Moran 2004; Bahram et al. 2006). However, it is a common practice to simply remove the scattering-bands by inserting missing values at the corresponding positions

(Murphy et al. 2013; C. A. Stedmon and Bro 2008).



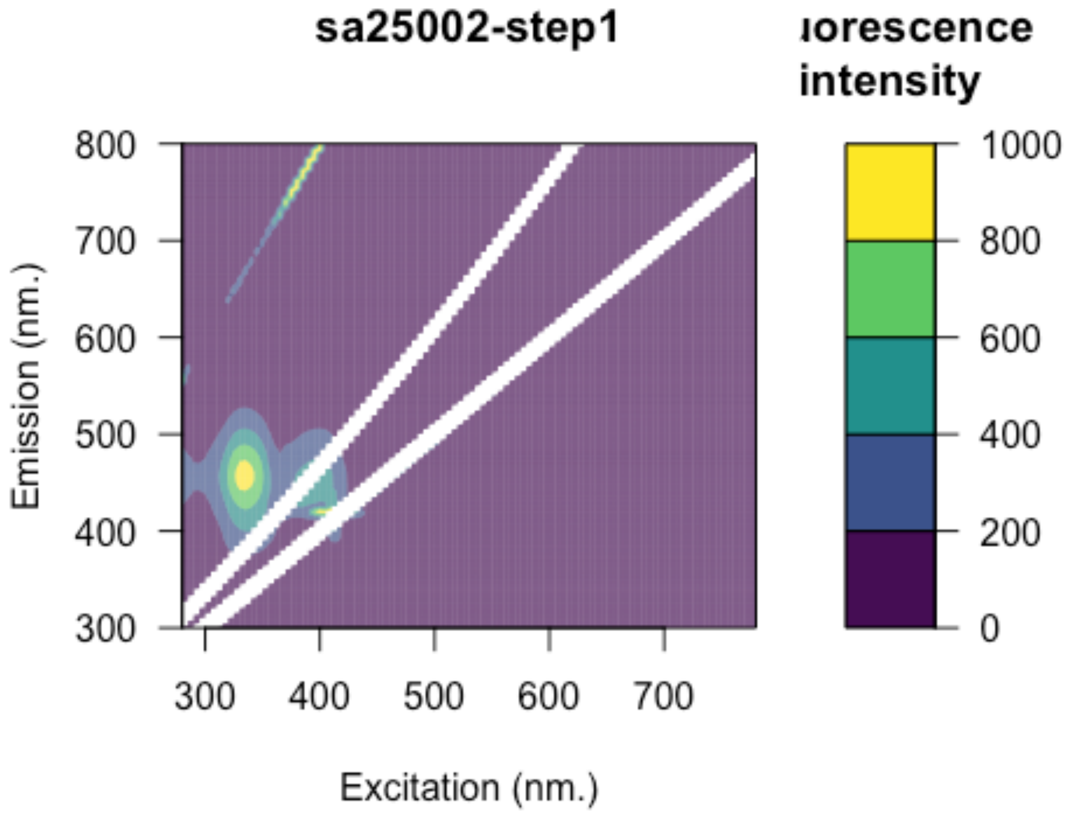
Different interpolation techniques have been proposed to eliminate scattering (Zepp, Sheldon, and Moran 2004; Bahram et al. 2006). However, it is a common practice to simply remove the scattering-bands by inserting missing values (Fig. 3) at the corresponding positions (Murphy et al. 2013; C. A. Stedmon and Bro 2008).

Removing Raleigh/Raman scatter

Scatter removal procedure

At the first step, I follow the suggestion of the eemR package, remove both the first order Raman scatter and Rayleigh scatter. Here, I use Cyclic QUV Step 1 as example.

```
eemsc1 <- eem_remove_scattering(eemsc1, "rayleigh", 1, 10)
eemsc1 <- eem_remove_scattering(eemsc1, "raman", 1, 10)

plot(eemsc1, which = 2)
```



Apparently, the second order emission on top left corner has not been removed. And the removal of Raman scatter overlaps the signal reflecting area of the material. Since the Raman scatter is not affacting the spectra here, I decide to cancel this step. Instead, I remove the second order Rayleigh scatter of the spectra.

The other problem here is the residual scatters. Although this 2D plot shows most of the residuals have been removed,it has been proved later in 3D model that the residual scatters are still evident in the datasets. This fact shows that the scattering are not completely removed using a bandwidth of 10 nm. In addition, since the eemR doesn't have functions to remove negative values, these outliers have not been removed as well, and they will impact the accurancy of the spectra.

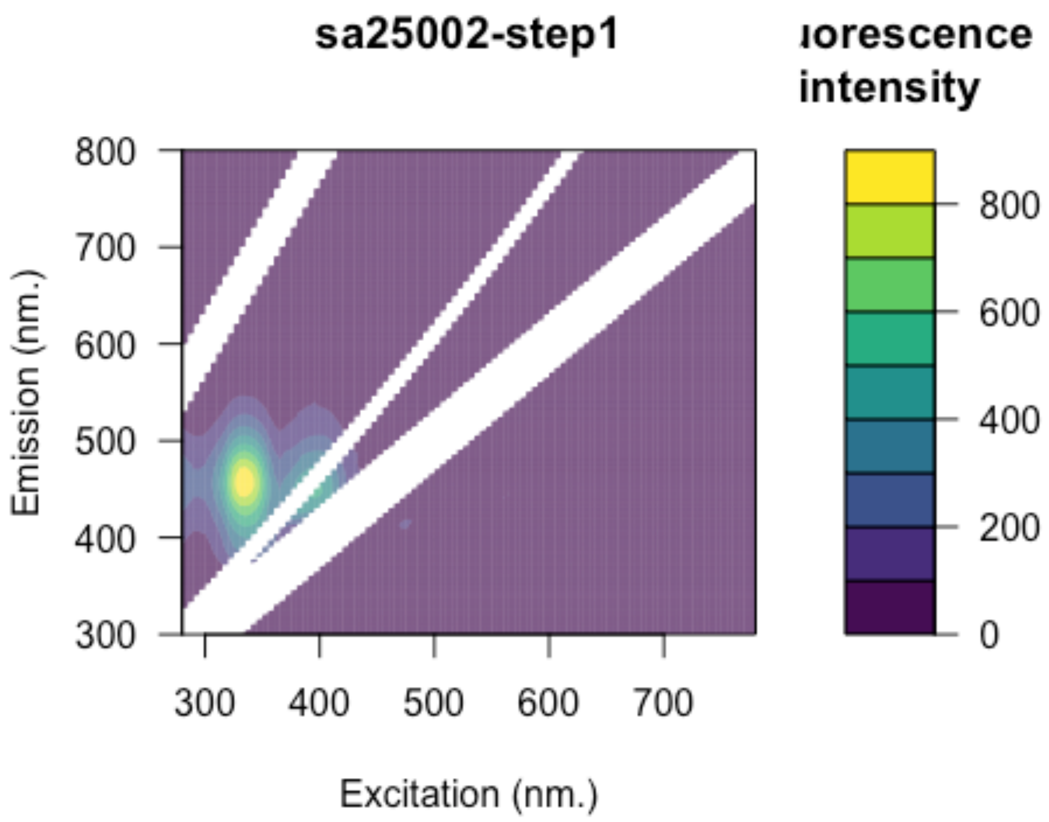
As a result, I choose to increase the bandwidth of the missing values insertions to see if I can resolve this problem. We I increase the bandwidth to 30nm, the missing values overlap most of the negative values.

Cyclic QUV

Step 1

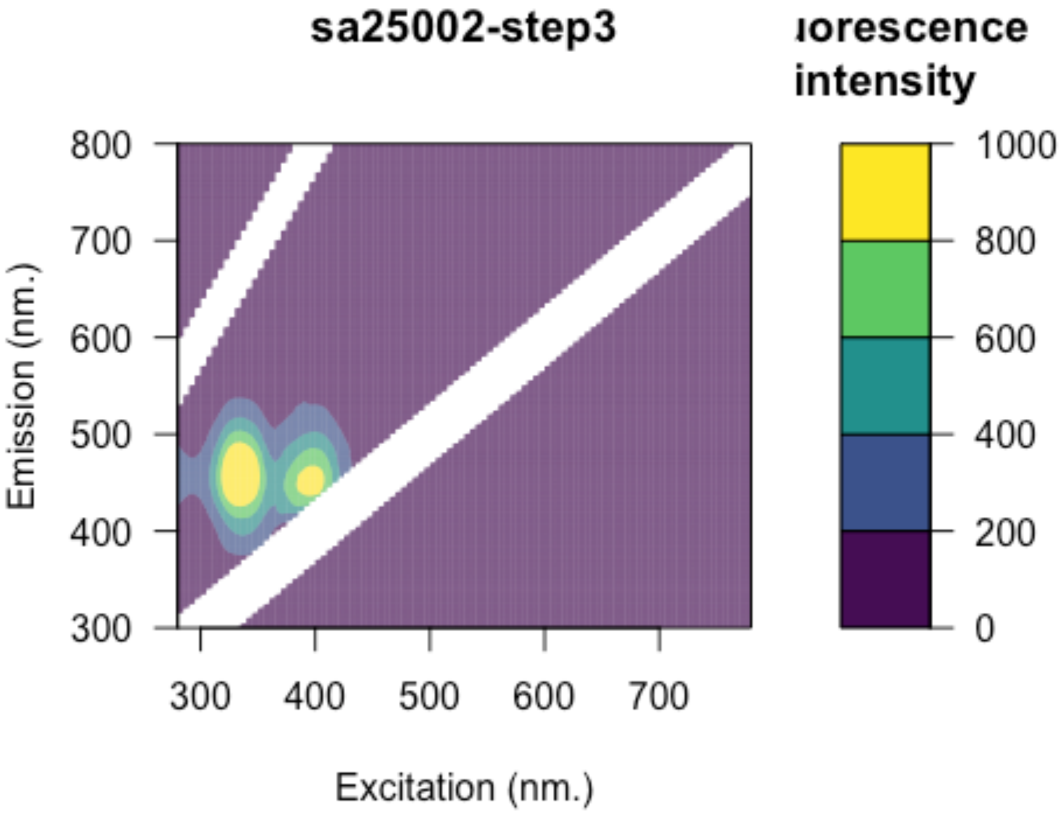
```
eemsc1 <- eem_remove_scattering(eemsc1, "rayleigh", 1, 30)
eemsc1 <- eem_remove_scattering(eemsc1, "rayleigh", 2, 30)
```

```
plot(eemsc1, which = 2)
```



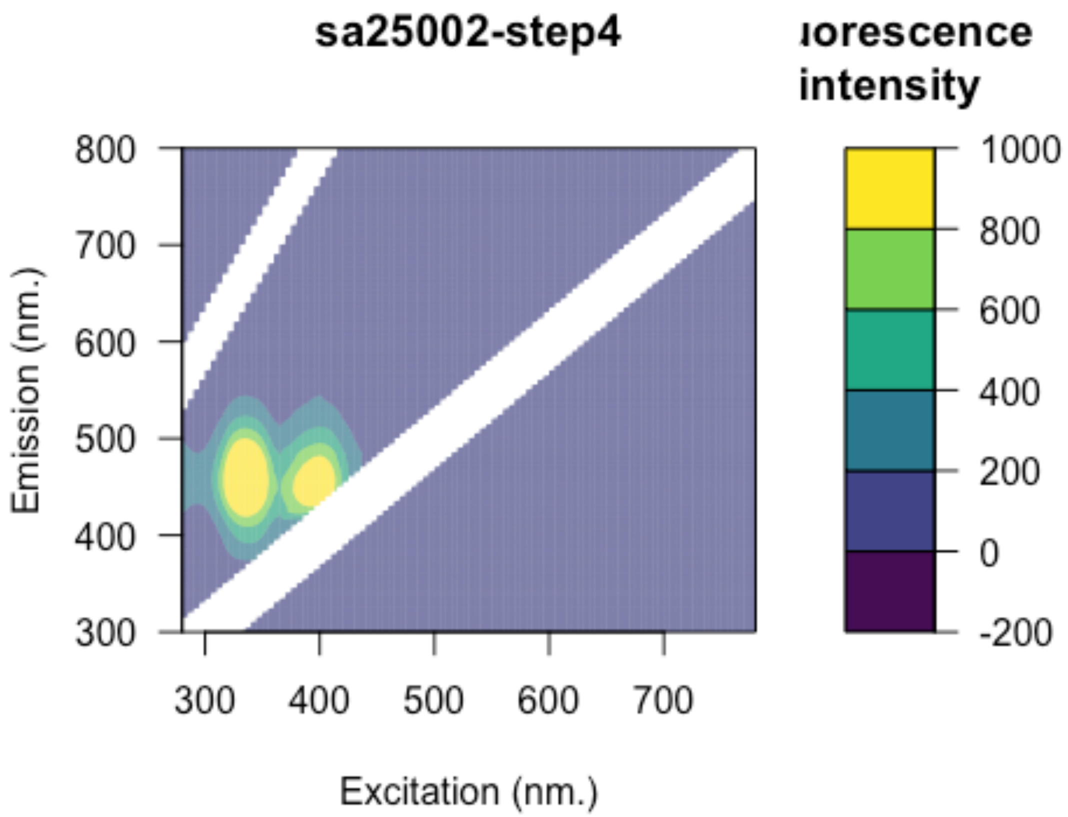
Step 3

```
eemsc3 <- eem_remove_scattering(eemsc3, "rayleigh", 1, 30)
eemsc3 <- eem_remove_scattering(eemsc3, "rayleigh", 2, 30)

plot(eemsc3, which = 2)
```



Step 4

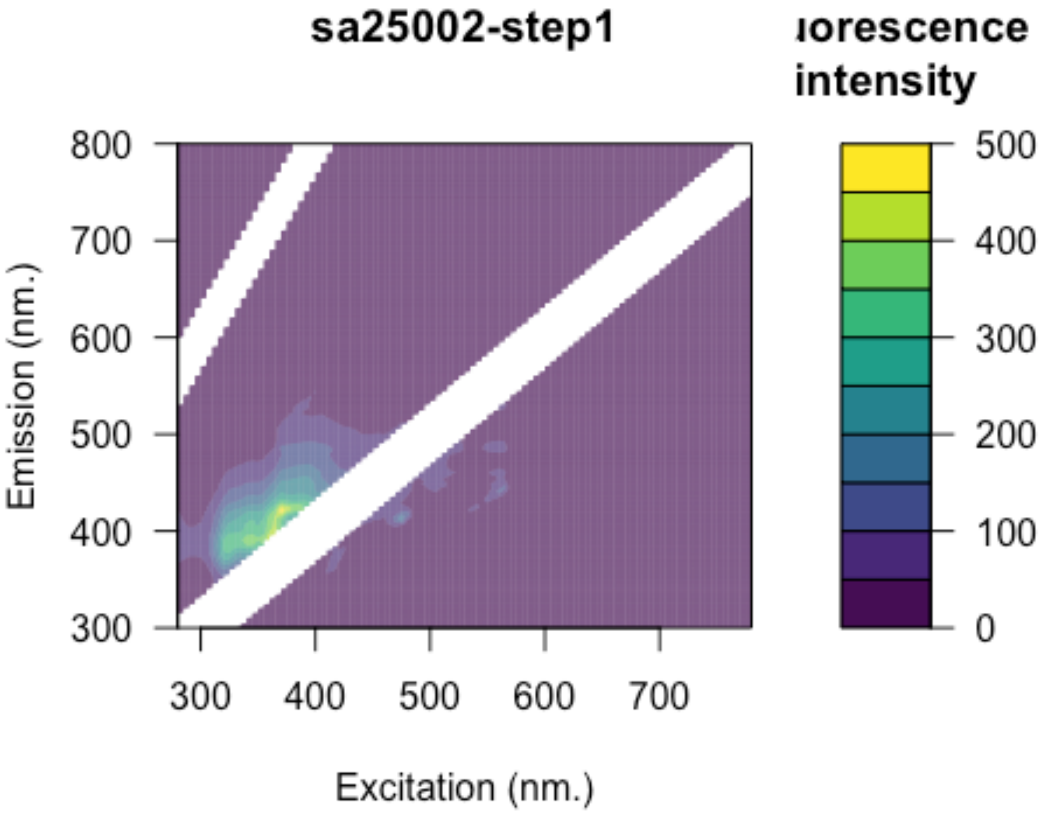
```
eemsc4 <- eem_remove_scattering(eemsc4, "rayleigh", 1, 30)
eemsc4 <- eem_remove_scattering(eemsc4, "rayleigh", 2, 30)

plot(eemsc4, which = 2)
```



DampHeat

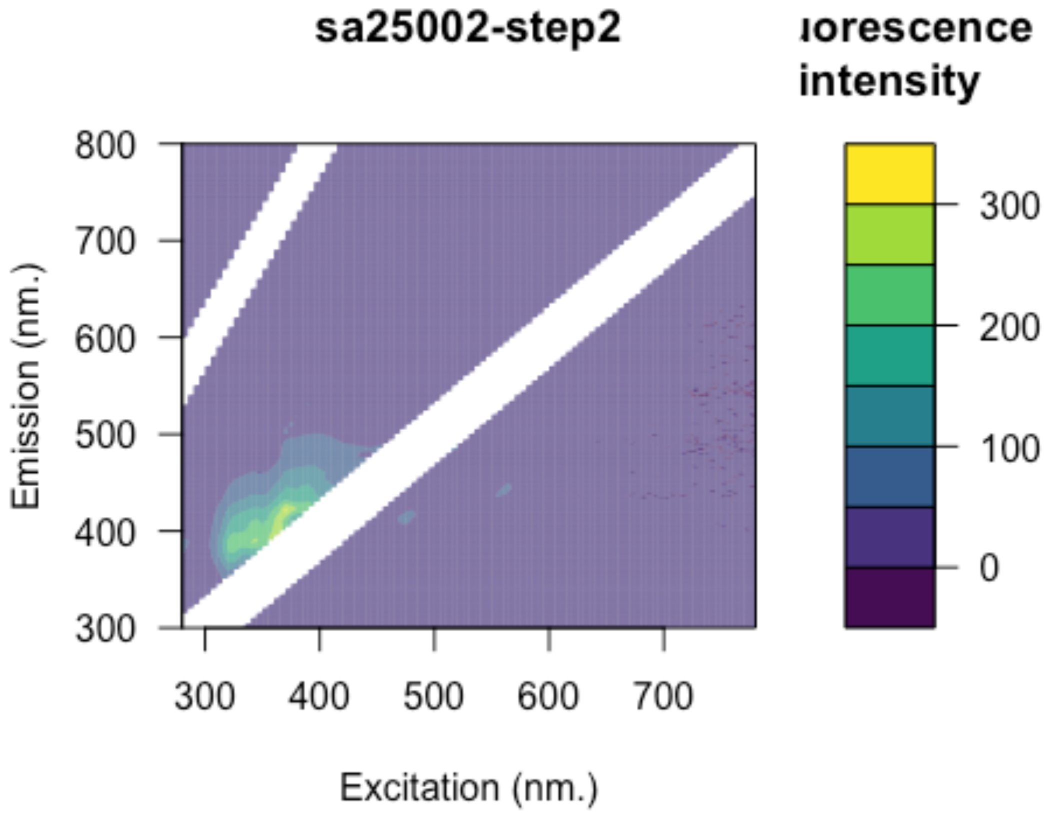
Step 1

```
eemsd1 <- eem_remove_scattering(eemsd1, "rayleigh", 1, 30)
eemsd1 <- eem_remove_scattering(eemsd1, "rayleigh", 2, 30)


plot(eemsd1, which = 2)
```



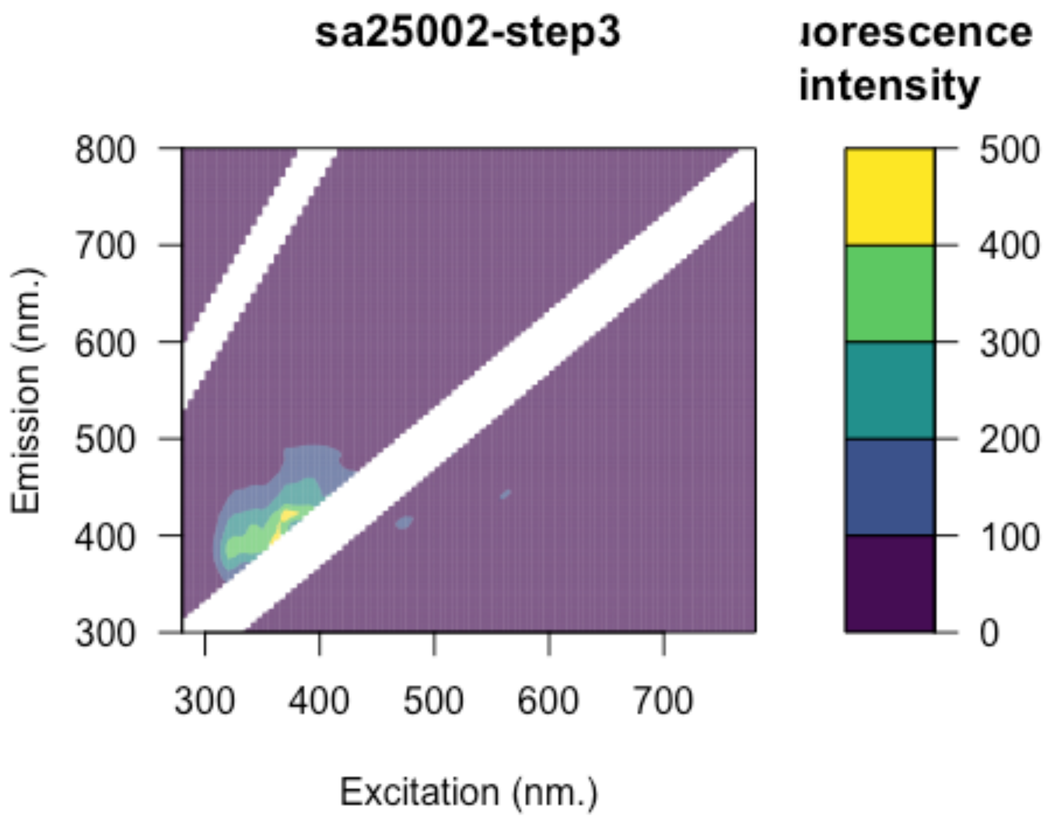
Step 2

```
eemsd2 <- eem_remove_scattering(eemsd2, "rayleigh", 1, 30)
eemsd2 <- eem_remove_scattering(eemsd2, "rayleigh", 2, 30)



plot(eemsd2, which = 2)
```



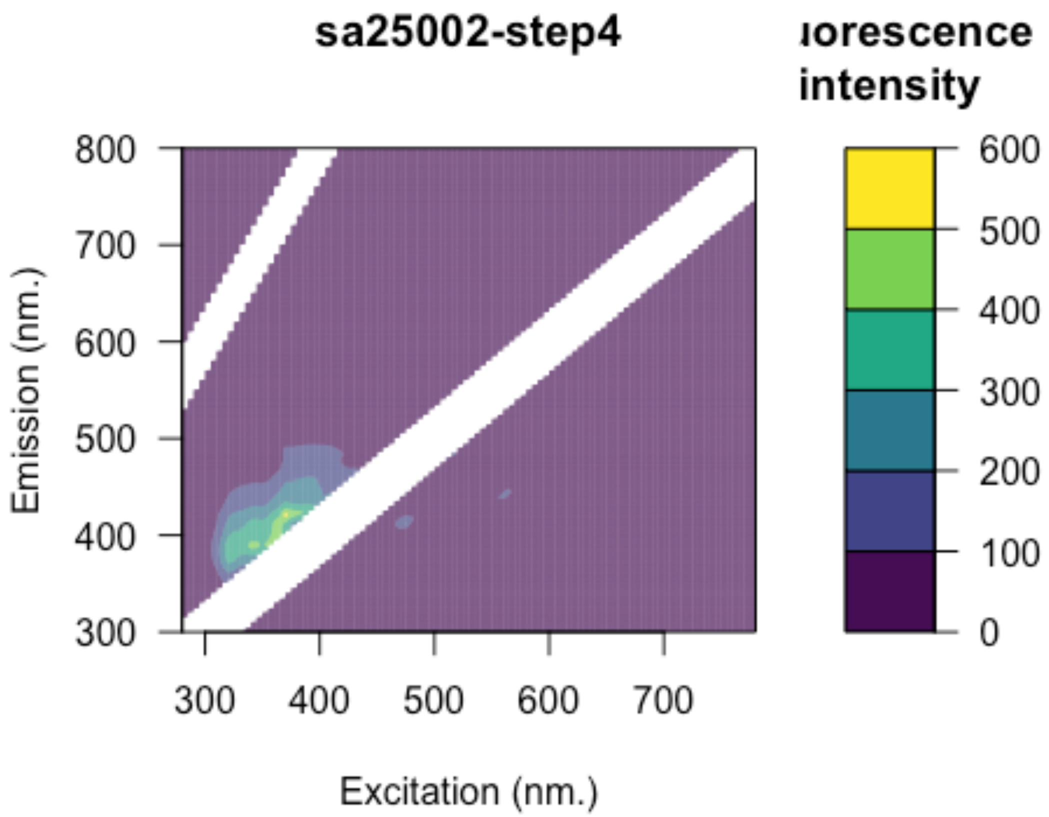
Step 3

```
eemsd3 <- eem_remove_scattering(eemsd3, "rayleigh", 1, 30)
eemsd3 <- eem_remove_scattering(eemsd3, "rayleigh", 2, 30)

plot(eemsd3, which = 2)
```



Step 4

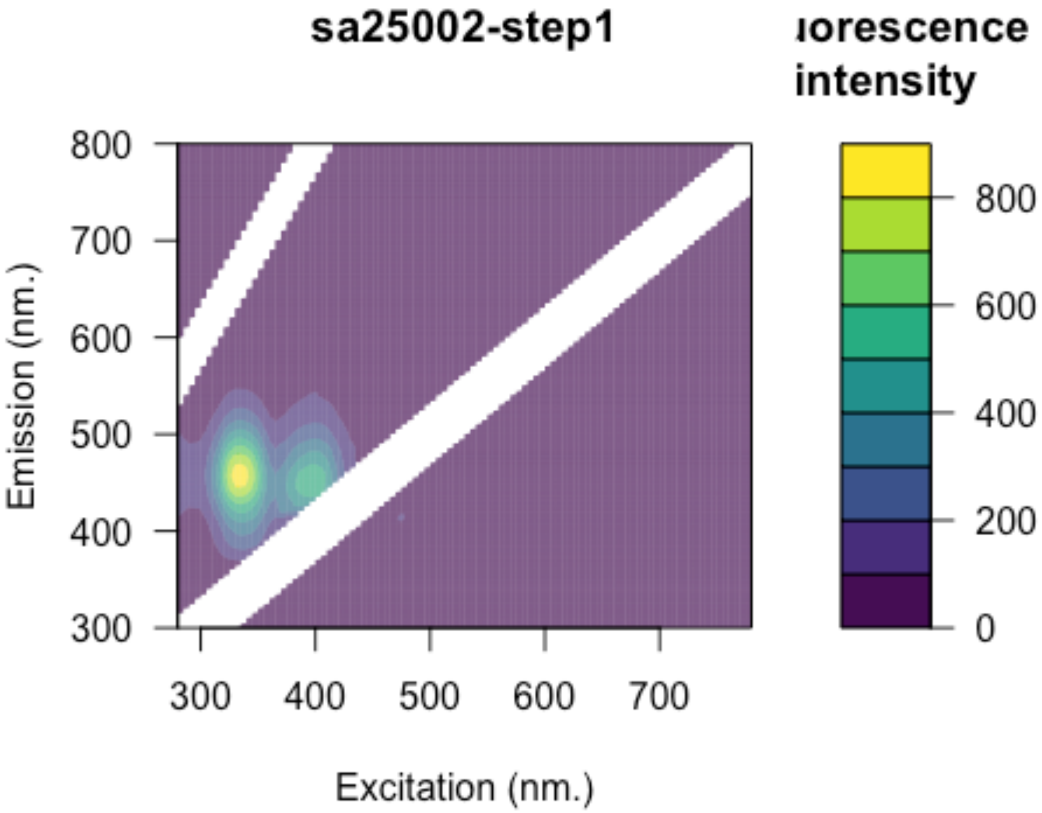
```
eemsd4 <- eem_remove_scattering(eemsd4, "rayleigh", 1, 30)
eemsd4 <- eem_remove_scattering(eemsd4, "rayleigh", 2, 30)

plot(eemsd4, which = 2)
```



HotQUV

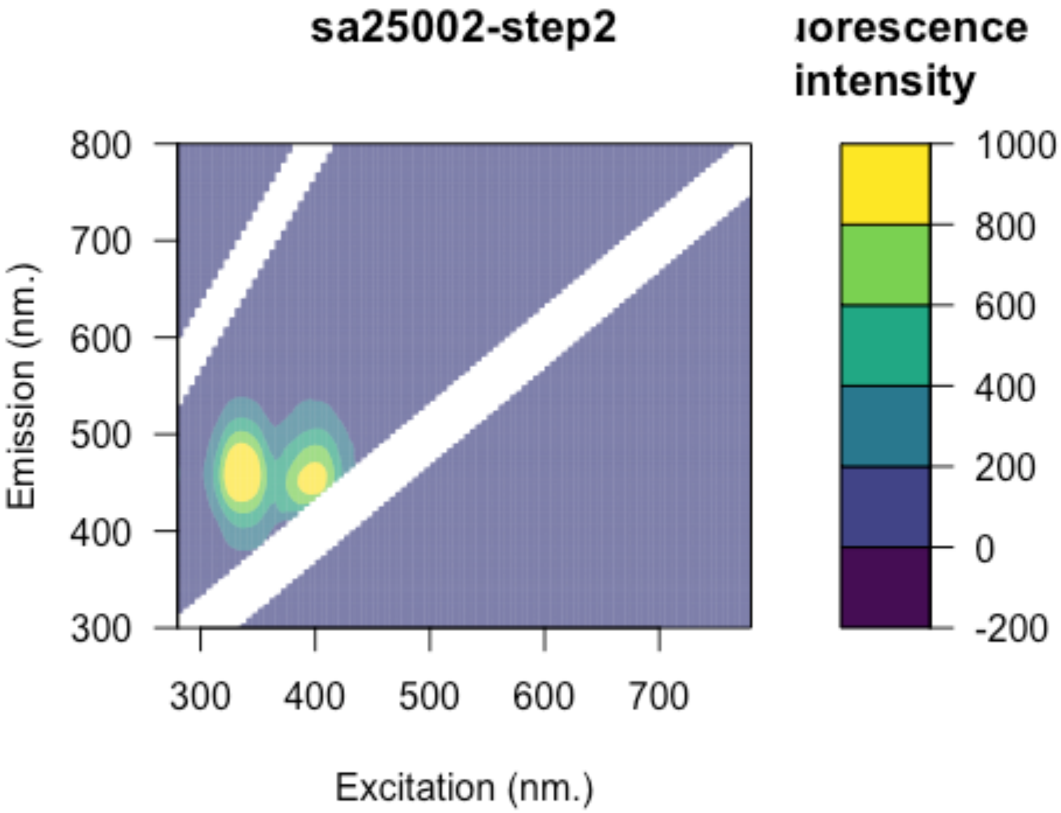
Step 1

```
eemsh1 <- eem_remove_scattering(eemsh1, "rayleigh", 1, 30)
eemsh1 <- eem_remove_scattering(eemsh1, "rayleigh", 2, 30)



plot(eemsh1, which = 2)
```



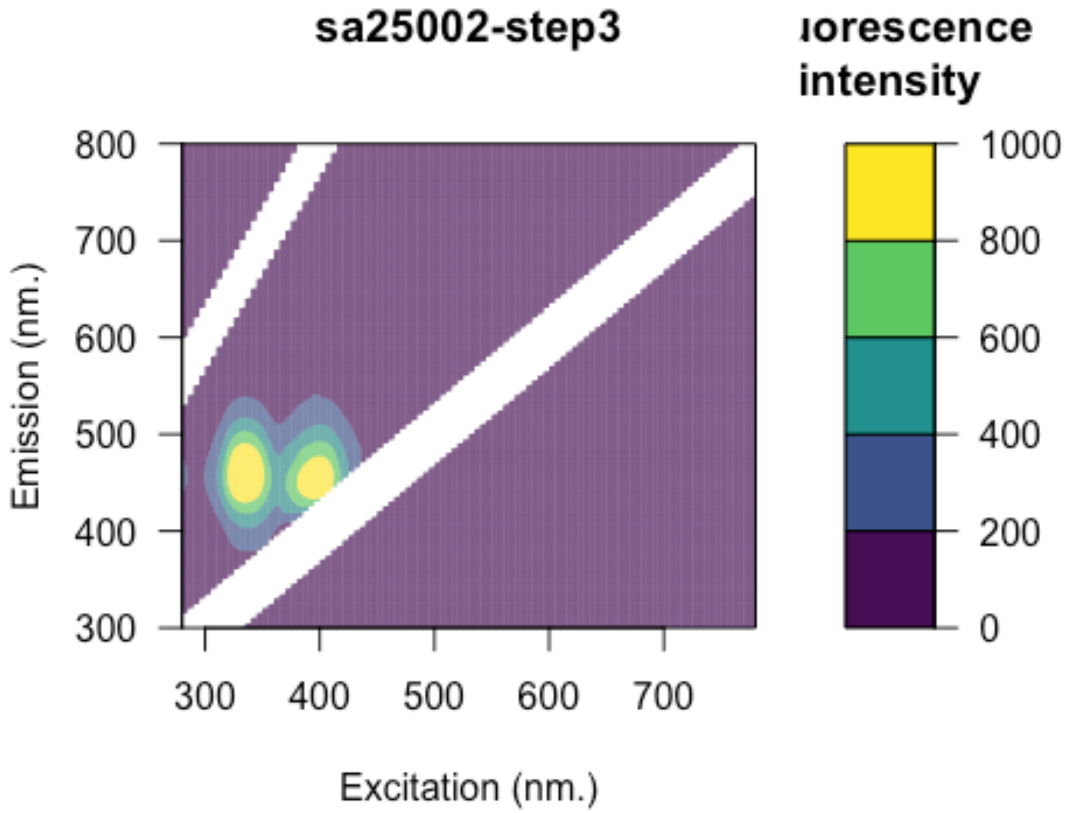
Step 2

```
eemsh2 <- eem_remove_scattering(eemsh2, "rayleigh", 1, 30)
eemsh2 <- eem_remove_scattering(eemsh2, "rayleigh", 2, 30)



plot(eemsh2, which = 2)
```



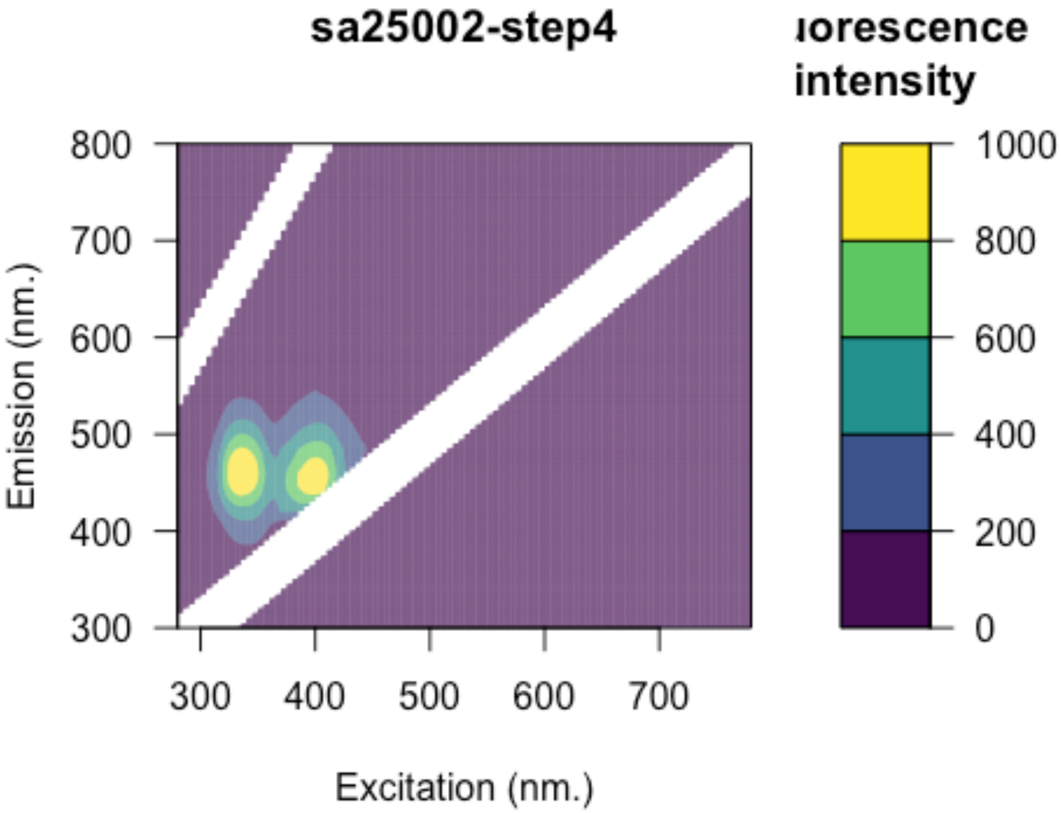
Step 3

```
eemsh3 <- eem_remove_scattering(eemsh3, "rayleigh", 1, 30)
eemsh3 <- eem_remove_scattering(eemsh3, "rayleigh", 2, 30)


plot(eemsh3, which = 2)
```



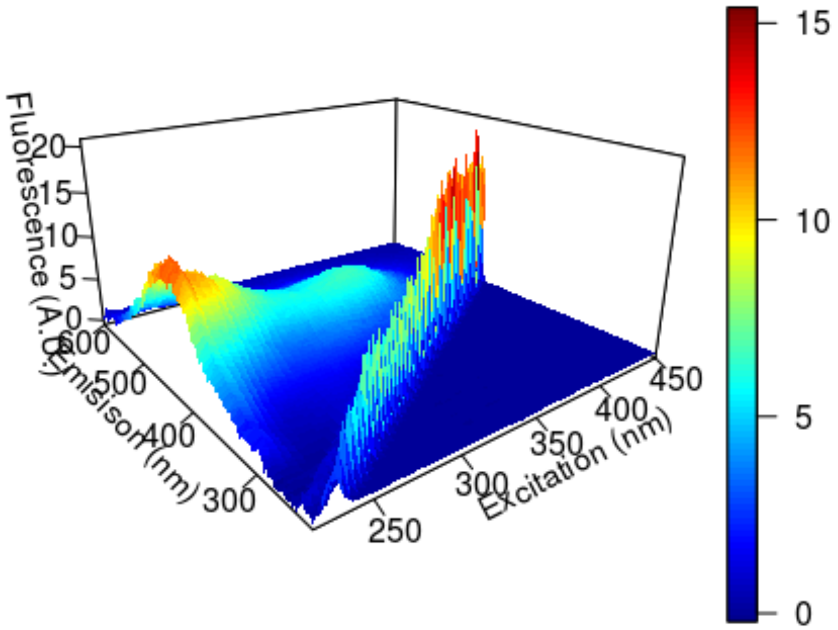
Step 4

```
eemsh4 <- eem_remove_scattering(eemsh4, "rayleigh", 1, 30)
eemsh4 <- eem_remove_scattering(eemsh4, "rayleigh", 2, 30)



plot(eemsh4, which = 2)
```

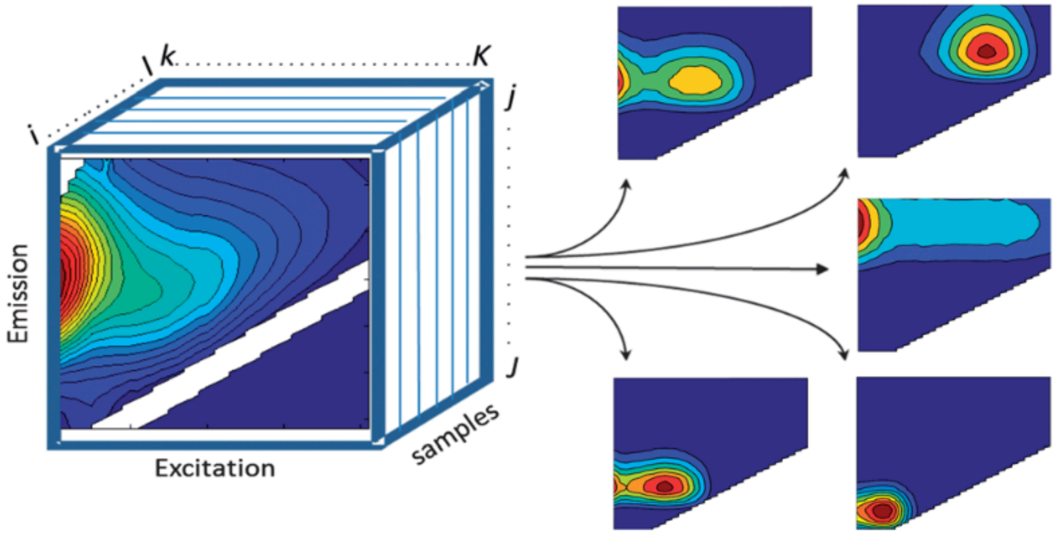


Introduction to PARAFAC

The seminal paper of (C. A. Stedmon, Markager, and Bro 2003) put at the forefront the use of parallel factor analysis (PARAFAC) to aid the characterization of fluorescent DOM. Briefly, this three-way technique allows the decomposition of complex DOM fluorescence signals contained in the excitation-emission matrix (EEM, Fig. 1) into a set of individual chemical components and provides estimations of their relative contribution to the total fluorescence (Bro 1997; Fellman, Hood, and Spencer 2010; C. A. Stedmon, Markager, and Bro 2003).



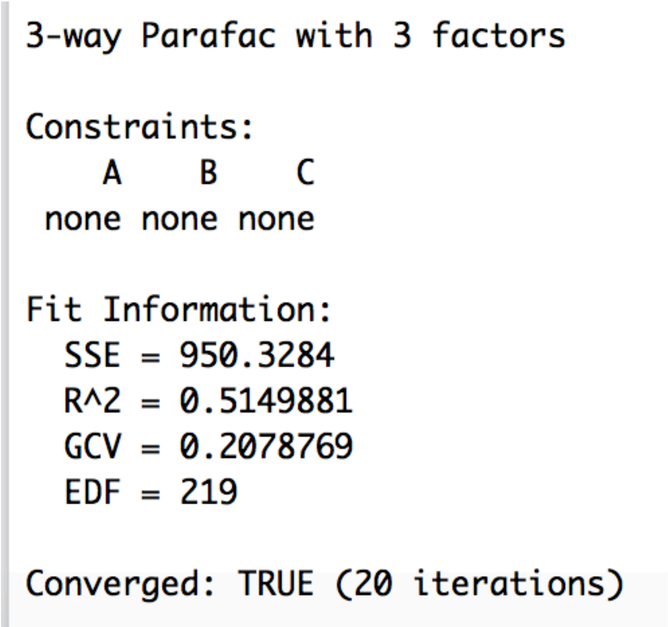
PARAFAC belongs to a family of so-called multi-way methods applicable to data that are arranged in three- or higher-order arrays. Examples of threeway arrays that can be analysed with PARAFAC include fluorescence EEMs (sample excitation wavelength emission wavelength), chromatographic data (GC-MS: sample elution time m/z structure), sensory data (sample attribute judge) and electroencephalography (space time frequency).



# Tools and corrections

multiway package in R

Firstly, to maintain the consisitancy, I tried "multiway" package in R. However, since it is a relatively new package, some functions are still not implemented. The Description of this package is: Fits multi-way component models via alternating least squares algorithms with optional constraints: orthogonal, non-negative, unimodal, monotonic, periodic, smooth, or structure. Fit models include Individual Differences Scaling, Parallel Factor Analysis (1 and 2), Si- multaneous Component Analysis, and Tucker Factor Analysis. However, it only fits the data, and prints constraint, fit, and convergence details of the model. It has no functions to plot the 3D EEMs.

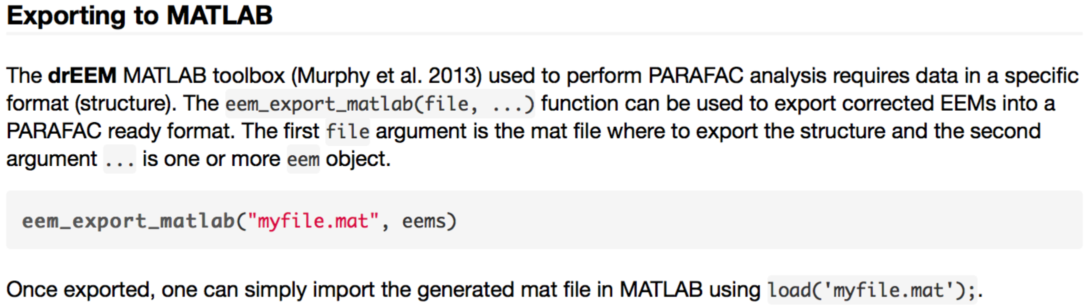


```
3-way Parafac with 3 factors

Constraints:
    A    B    C
 none none none

Fit Information:
  SSE = 950.3284
  R^2 = 0.5149881
  GCV = 0.2078769
  EDF = 219

Converged: TRUE (20 iterations)
```

During this week's test, I make sure that the multiway package in R can't visualize the PARAFAC data. Therefore I give up doing this procedure solely on R and switch to the backup plans.

drEEM toolbox in matlab

Experiencing setbacks in R, I try to export the data to matlab to do the PARAFAC analysis. According to the seminal paper of PARAFAC, I install the drEEM package. It can plot PARAFAC digrams with corrected EEM data. Since I finished the data correction in R with eemR package, I use the export method of eemR to create a PARAFAC ready format.



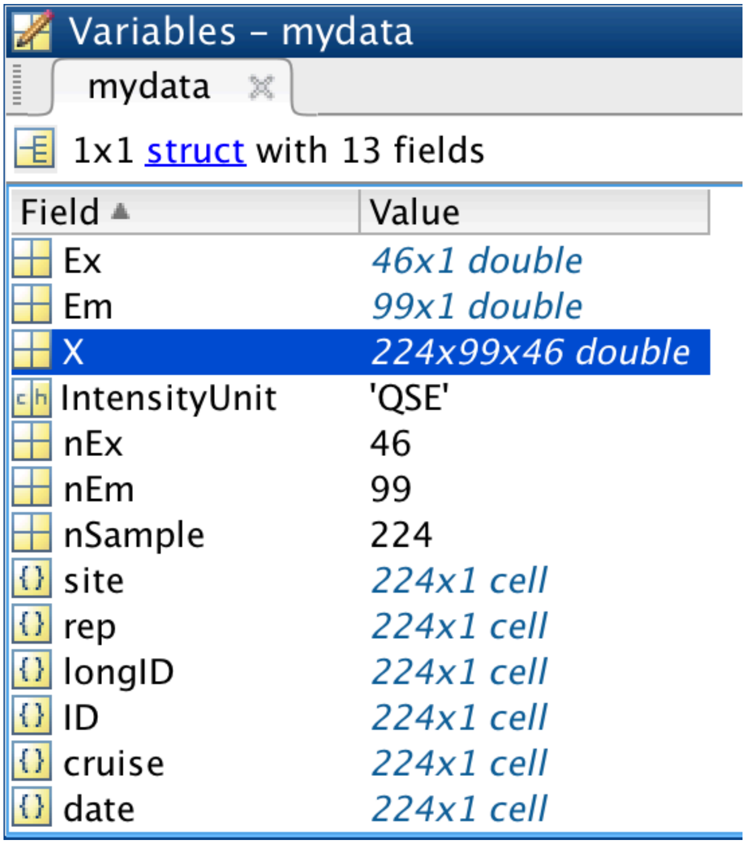
Exporting to MATLAB

The **drEEM** MATLAB toolbox (Murphy et al. 2013) used to perform PARAFAC analysis requires data in a specific format (structure). The `eem_export_matlab(file, ...)` function can be used to export corrected EEMs into a PARAFAC ready format. The first `file` argument is the mat file where to export the structure and the second argument `...` is one or more `eem` object.

```
eem_export_matlab("myfile.mat", eems)
```

Once exported, one can simply import the generated mat file in MATLAB using `load('myfile.mat');`.

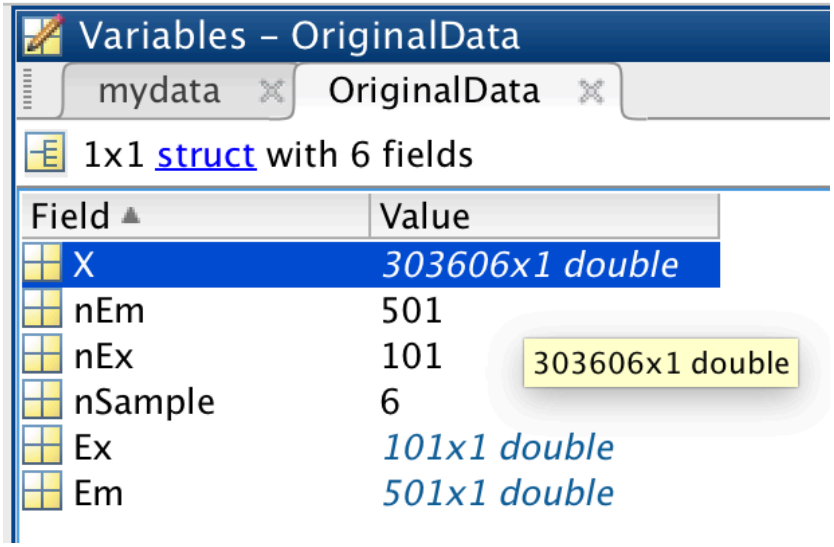
However, the exported data is actually not in correct form. The good data of EEM here is a 3-dimensional array.



1x1 struct with 13 fields

Field	Value
Ex	46x1 double
Em	99x1 double
X	224x99x46 double
IntensityUnit	'QSE'
nEx	46
nEm	99
nSample	224
site	224x1 cell
rep	224x1 cell
longID	224x1 cell
ID	224x1 cell
cruise	224x1 cell
date	224x1 cell

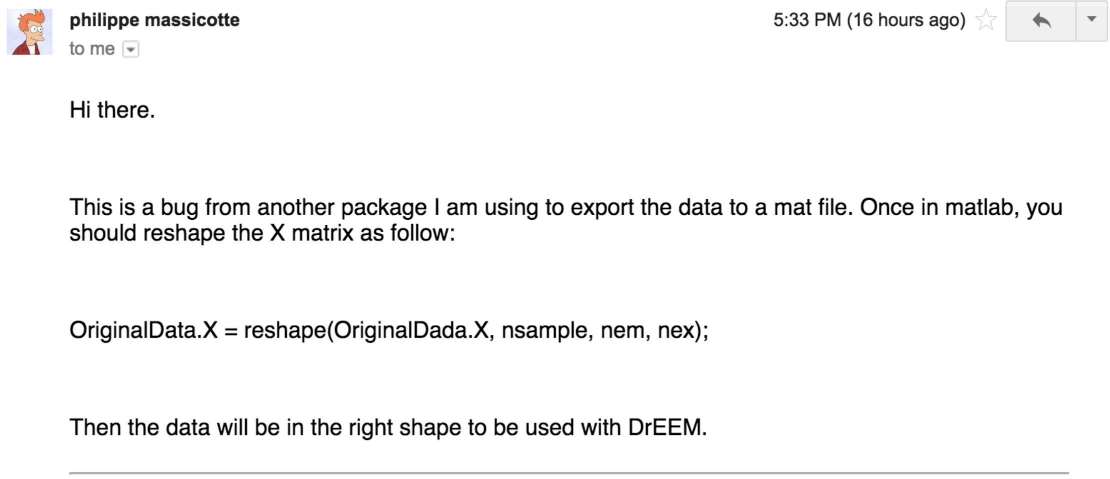
However, the data exported by eemR package only has one dimension.



1x1 struct with 6 fields

Field	Value
X	303606x1 double
nEm	501
nEx	101
nSample	6
Ex	101x1 double
Em	501x1 double

To solve this problem, I directly contact the author of the eemR package.



philippe massicotte

to me

5:33 PM (16 hours ago)

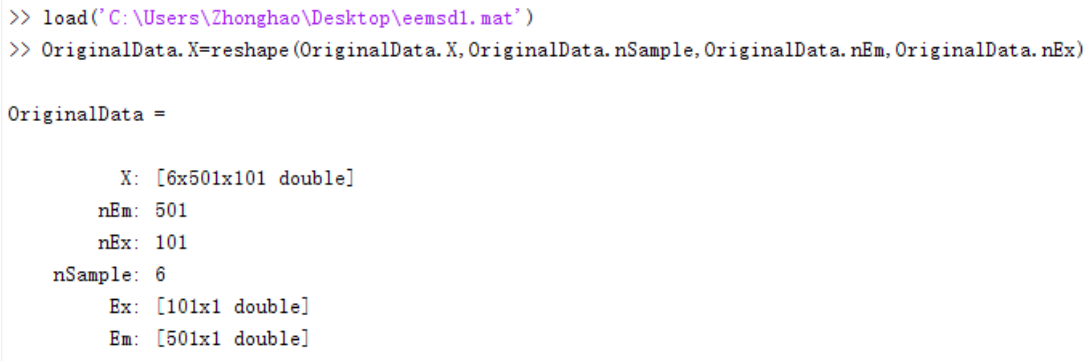
Hi there.

This is a bug from another package I am using to export the data to a mat file. Once in matlab, you should reshape the X matrix as follow:

OriginalData.X = reshape(OriginalDada.X, nsample, nem, nex);

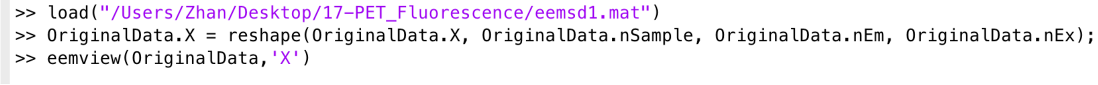
Then the data will be in the right shape to be used with DrEEM.

Although there are several typos in his mail, this reshape function works in the matlab, and the data which I imported to matlab is now in correct form.



```
>> load('C:\Users\Zhonghao\Desktop\eemsd1.mat')
>> OriginalData.X=reshape(OriginalData.X,OriginalData.nSample,OriginalData.nEm,OriginalData.nEx)

OriginalData =

          X: [6x501x101 double]
        nEm: 501
        nEx: 101
    nSample: 6
         Ex: [101x1 double]
         Em: [501x1 double]
```

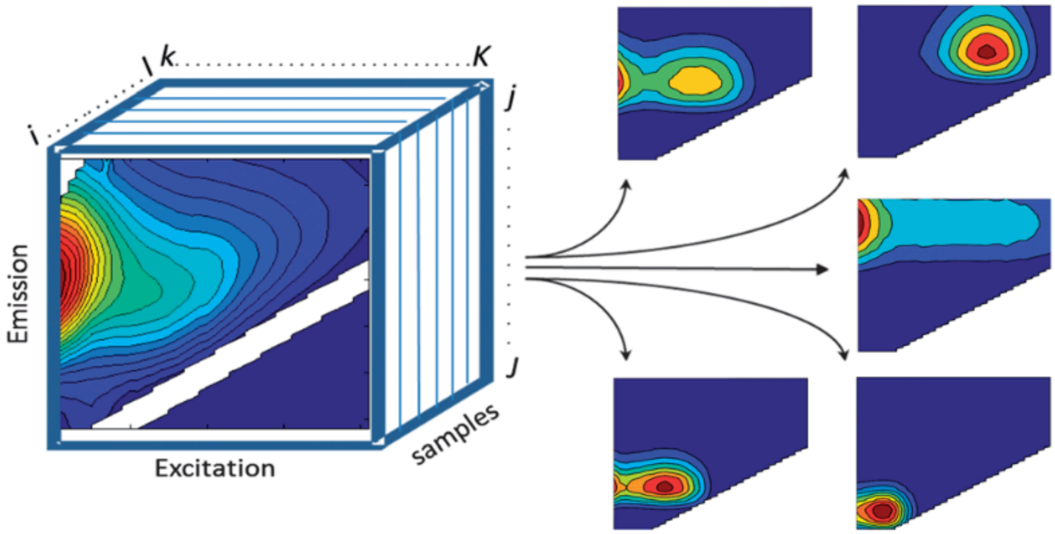
After the correction, I firstly try the free drEEM package.



```
>> load("/Users/Zhan/Desktop/17-PET_Fluorescence/eemsd1.mat")
>> OriginalData.X = reshape(OriginalData.X, OriginalData.nSample, OriginalData.nEm, OriginalData.nEx);
>> eemview(OriginalData,'X')
```

I will show this procedure in matlab.

One problem still remaining is that the drEEM can only plot 2D figures.According to the seminal paper, the three ways of data in drEEM are actually Em, Ex and Samples. This is not very satisfying, since the intensity is an important factor and it also needs to be visualized.



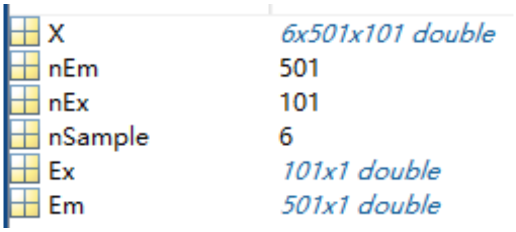
EEM

As a result, I move to the PLS toolbox.

PLS toolbox in matlab

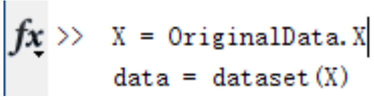
Different with the previous two free packages, PLS is commerical, which costs 495$. The advantage of this package is it can generate 3D plots and presents them in a PARAFAC fashion.

The difference between drEEM and PLS is that PLS needs a data frame. The struct which established in the previous step can't be directly processed by PLS. Therefore I made some further corrections.



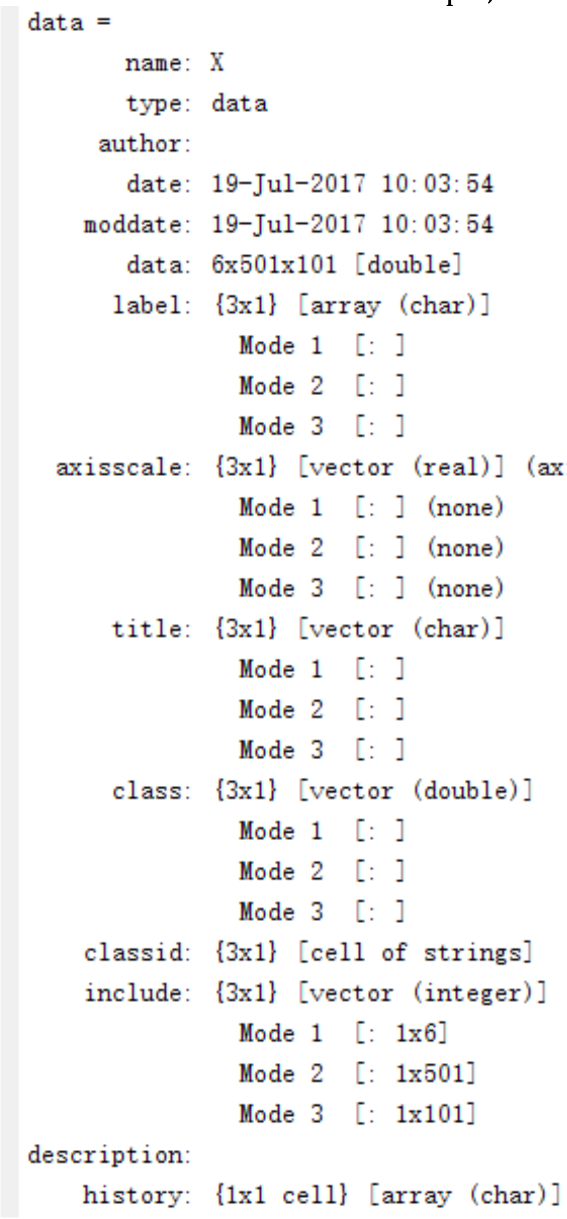
```
X          6x501x101 double
nEm        501
nEx        101
nSample    6
Ex         101x1 double
Em         501x1 double
```

Firstly, I pull out the double array from the struct and save it into a dataset object.



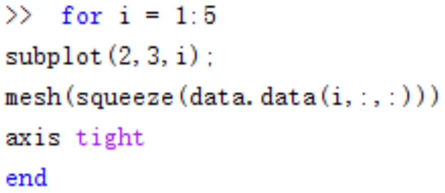
```
>>  X = OriginalData.X
    data = dataset(X)
```

Simple, but effective.



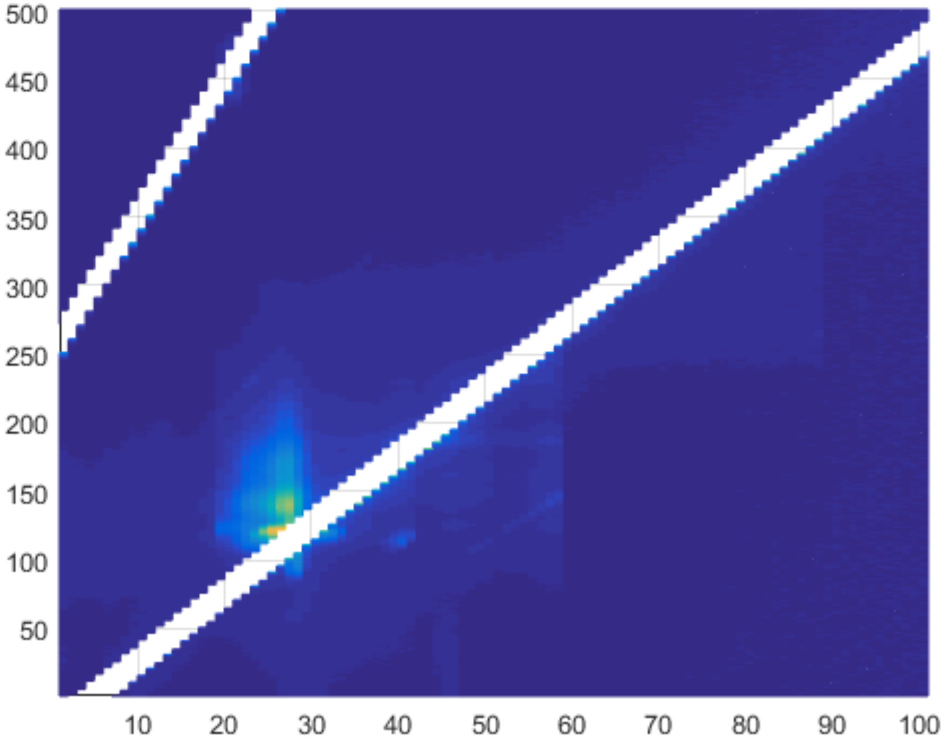
```
data =
       name: X
       type: data
     author:
       date: 19-Jul-2017 10:03:54
    moddate: 19-Jul-2017 10:03:54
       data: 6x501x101 [double]
      label: {3x1} [array (char)]
               Mode 1  [: ]
               Mode 2  [: ]
               Mode 3  [: ]
  axisscale: {3x1} [vector (real)] (axistype)
               Mode 1  [: ] (none)
               Mode 2  [: ] (none)
               Mode 3  [: ] (none)
      title: {3x1} [vector (char)]
               Mode 1  [: ]
               Mode 2  [: ]
               Mode 3  [: ]
      class: {3x1} [vector (double)]
               Mode 1  [: ]
               Mode 2  [: ]
               Mode 3  [: ]
    classid: {3x1} [cell of strings]
    include: {3x1} [vector (integer)]
               Mode 1  [: 1x6]
               Mode 2  [: 1x501]
               Mode 3  [: 1x101]
description:
    history: {1x1 cell} [array (char)]
```

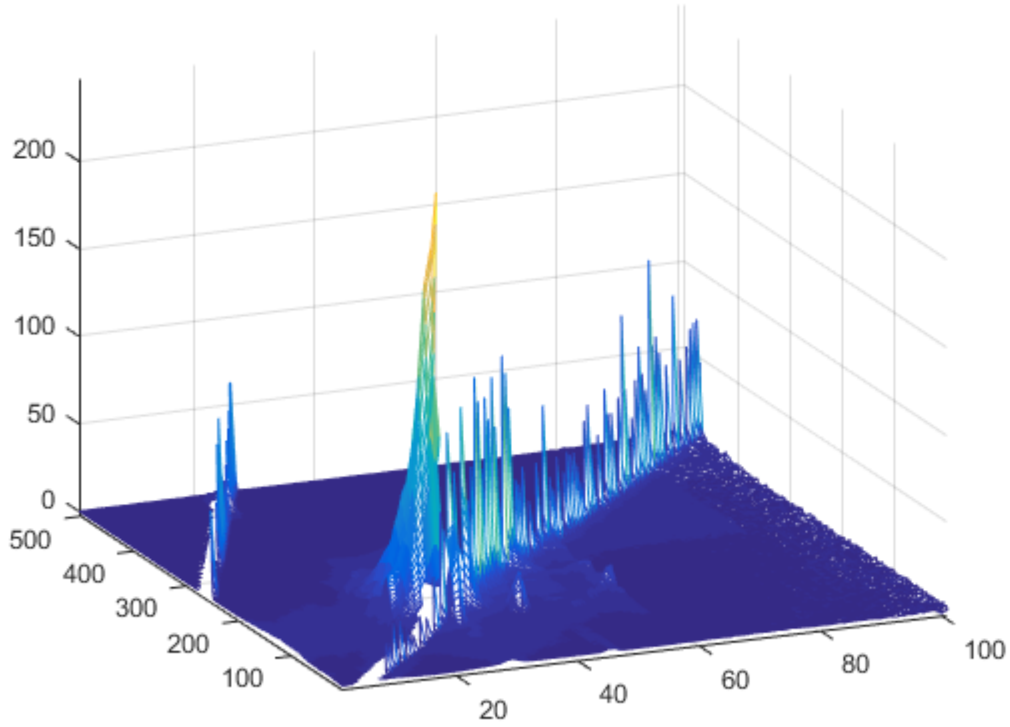
After this step I use the function below to plot the PARAFAC figure.

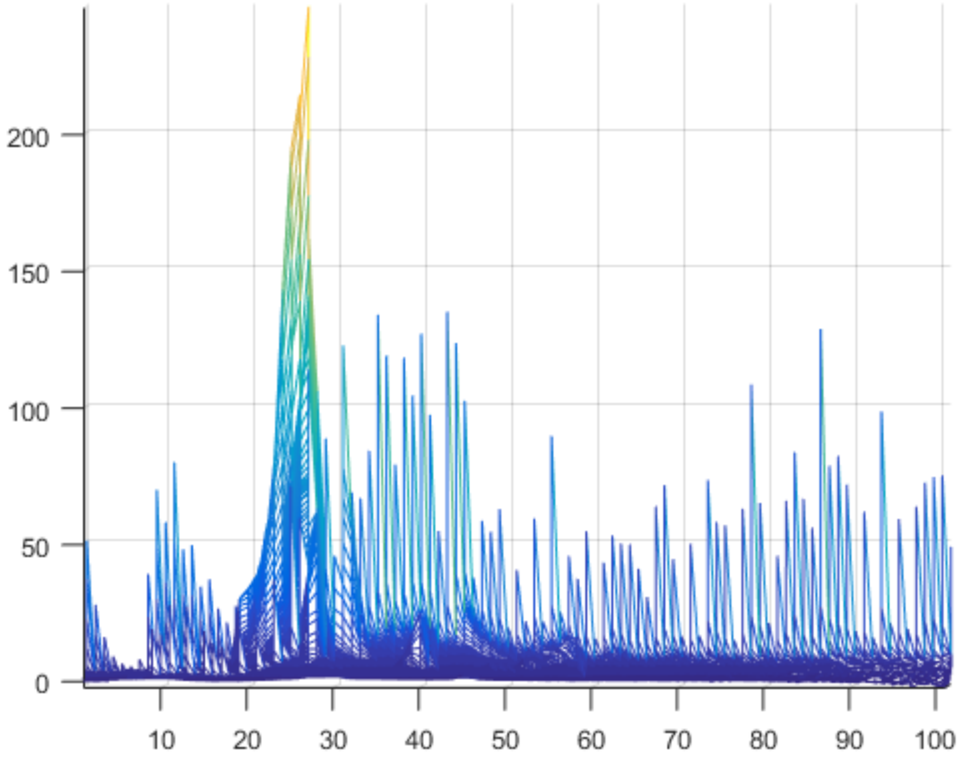


```
>>  for i = 1:5
subplot(2,3,i);
mesh(squeeze(data.data(i,:,:)))
axis tight
end
```

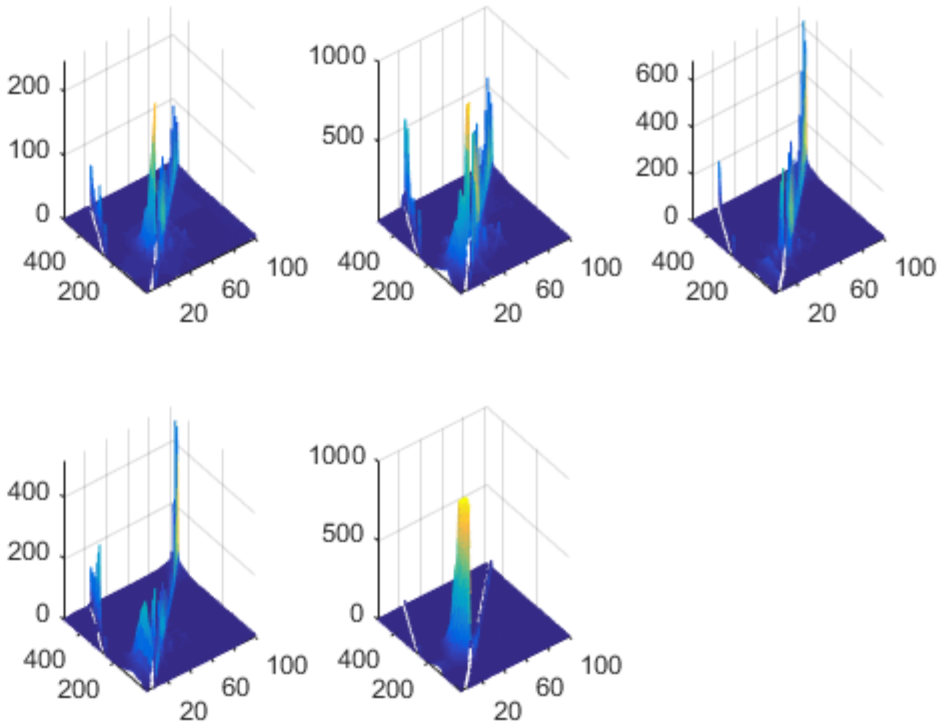
This is the one sample plot.The three ways are Ex Em and Intensity.




200
150
100
50
0
500
400
300
200
100
20
40
60
80
100


200
150
100
50
0
10
20
30
40
50
60
70
80
90
100

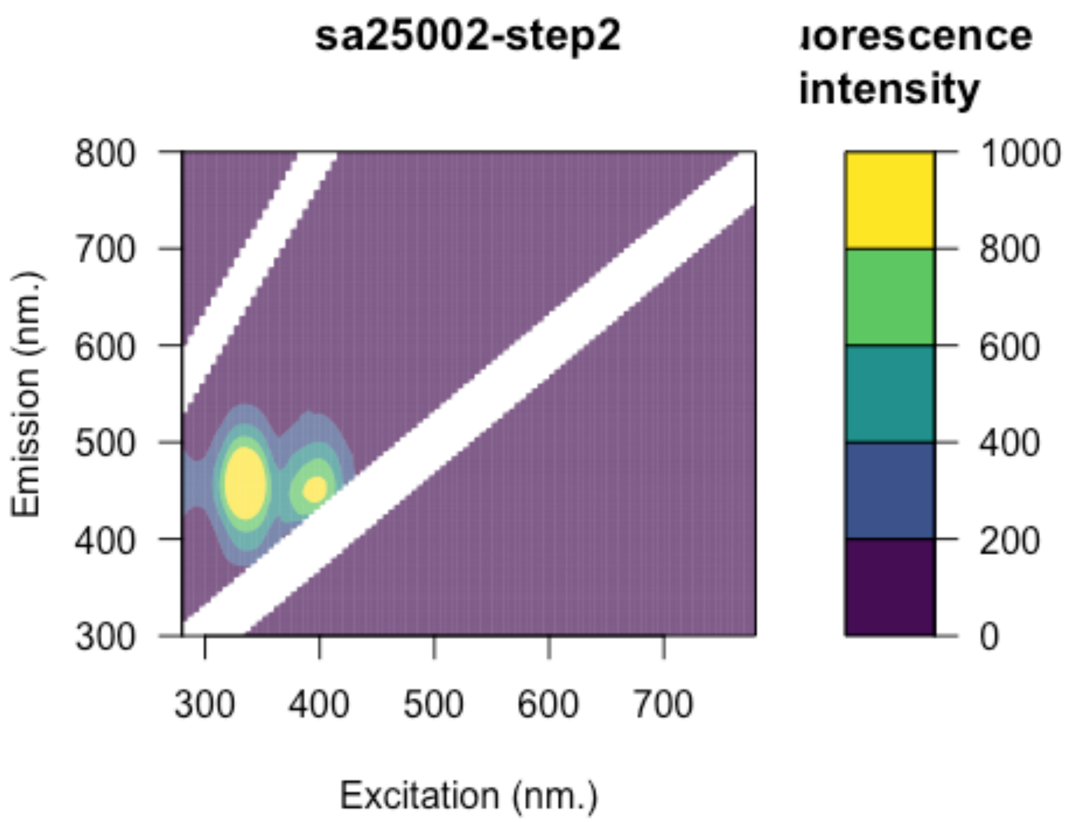
This is the multi sample plot. The four ways are Ex, Em, Intensity, and Samples.

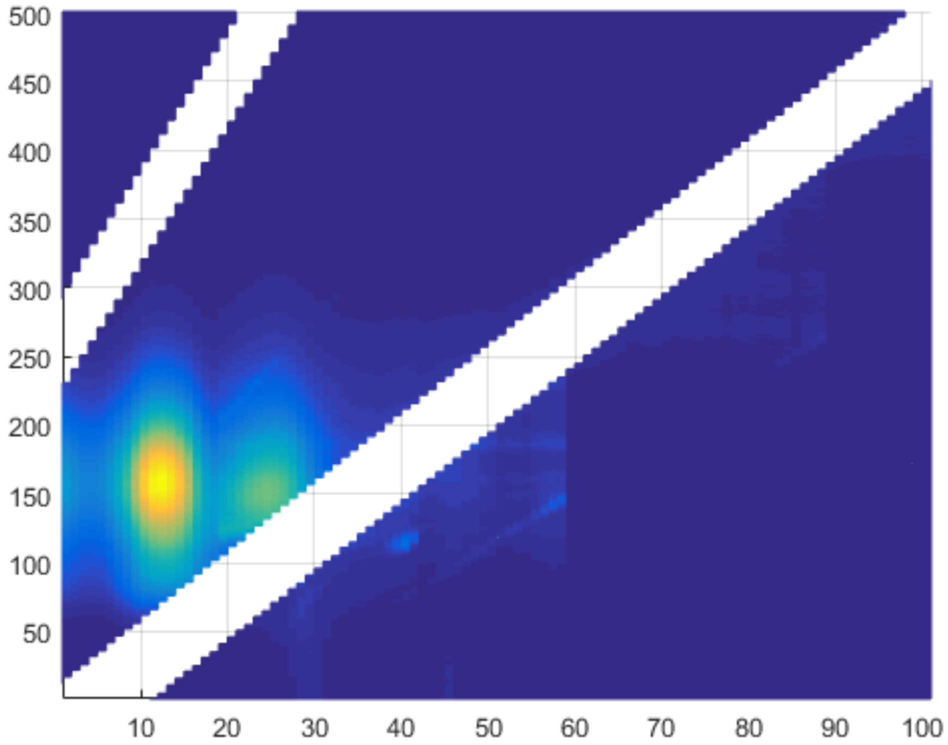


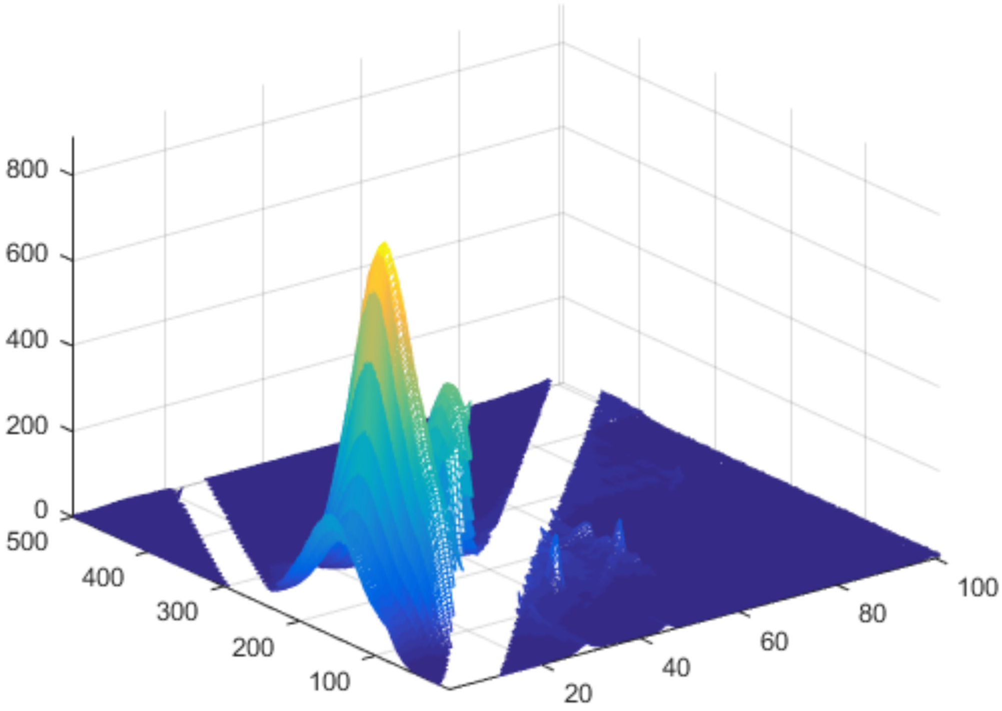
As you can see, with the removal bandwidth of 10nm, the residuals are not properly eliminatied. Therefore, I increase the bandwidth to 30nm. On the other hand, I create a new dataset which includes the spectras of one single PET film in four steps. Here, I will show the plot of sample 2 (sa25002) in Cyclic QUV procedure.

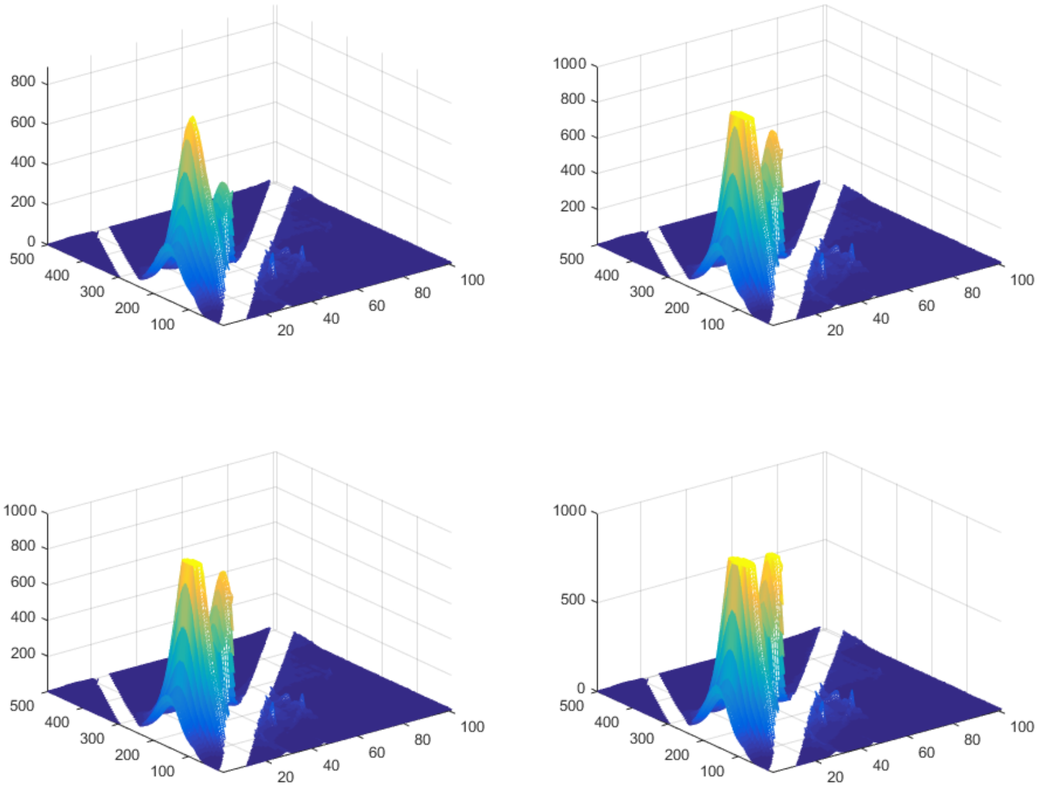
```
sample2 <- system.file("extdata/cary/sample2/", package = "eemR")
eems <- eem_read(sample2)

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

## Warning in matrix(as.numeric(unlist(data, use.names = FALSE)), ncol =
## expected_col, : NAs introduced by coercion

eems[[1]]$em<-eems[[1]]$em[1:501]
eems[[1]]$x<-eems[[1]]$x[c(1:501),c(1:101)]
eems[[2]]$em<-eems[[2]]$em[1:501]
eems[[2]]$x<-eems[[2]]$x[c(1:501),c(1:101)]
eems[[3]]$em<-eems[[3]]$em[1:501]
eems[[3]]$x<-eems[[3]]$x[c(1:501),c(1:101)]
eems[[4]]$em<-eems[[4]]$em[1:501]
```

```
eems[[4]]$x<-eems[[4]]$x[c(1:501),c(1:101)]
eems <- eem_remove_scattering(eems, "rayleigh", 1, 30)
eems <- eem_remove_scattering(eems, "rayleigh", 2, 30)

plot(eems, which=2)
```




500
450
400
350
300
250
200
150
100
50
10
20
30
40
50
60
70
80
90
100


800
600
400
200
0
500
400
300
200
100
20
40
60
80
100



At this point, the PARAFAC analysis has been successfully performed. Most of the goals of this project have been achieved. One drawback of this procedure is that there are no proper function for removing all of the negative value. And the 3D models of the all other samples except Cyclic QUV sample 25002 should be built. #Purposed future work

● Build the 3D model for other samples. ● Find a proper way to remove the residual negative values. ● Write guidelines for processing fluorescences and reproducing the research procedures. ● Provide insights about the trends with increasing exposure time/steps.