USGSHydroOpt : Tools for Optical Analysis of Water

Samuel Christel[1] and Steve Corsi[1]

[1]*United States Geological Survey*

August 4, 2014

Contents

1 Introduction to USGSHydroOpt

The USGSHydroOpt package was created to streamline the process of creating explanatory optical variables, and excitation-emission (EEMs) plots from absorbance and fluorescence data. Examples of optical summary variables that can be produced with this package include various absorbance peaks, spectral slopes,

fluorescence and humic index variables, and others. The motivation for producing this package is the power of optical variables for explaining biological and chemical phenomena in water. Optical summary variables produced using USSHydroOpt are extremely useful in statistical modeling efforts.

The functions in this package were designed to operate on dataframes with standardized formats. This package is not amenable to dataframes or arrays that do not fit the defined formats. The example dataframes and array in this package illustrate exactly how dataframes and arrays should be formatted, and the examples illustrate how to use the functions. USGSHydroOpt contains a very useful function for creating contour plots of EEMs spectra. Depicted below is an example of an EEMs plot produced with this package. The user is encouraged to step through each piece of R code as it is introduced in this document. In many instances a sample of the function output is provided via plots and tables, but not in all cases.
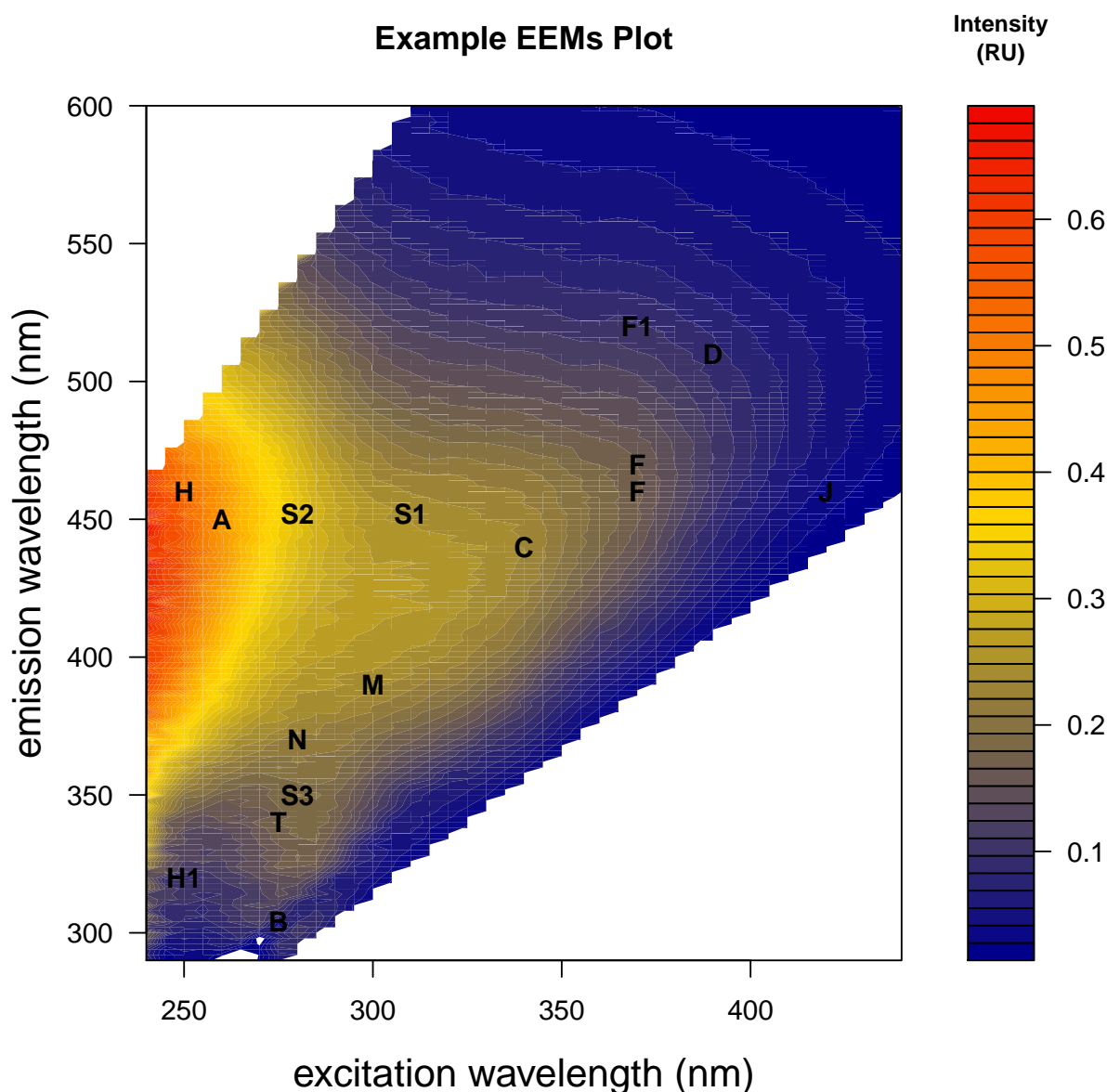
2   Required Data Formats

The functions contained in USGSHydroOpt operate on dataframes with standardized schemas. Users interested in using USGSHydroOpt should take care to ensure that dataframes and arrays are formatted according to the schemas described in this section.

2.1   Absorbance Data

Absorbance dataframes used by functions in USGSHydroOpt should be formatted such that each sample occupies a column, and one column contains the wavelength (nm) for which the absorbance measurement was made (*See example Table 2.1 below*). The example absorbance dataframe included in this package is called **dfabs**. The column with the wavelengths in Table 2.1 does not need to be called "wavelengths," as it is in Table 2.1. Since this package was developed primarily for USGS activities, the default for naming samples is "gr" followed by the sample number. This convention was started by the USGS California Water Sciences Center (CA WSC), and the USGS Wisconsin Water Science Center (WI WSC) follows the same naming convention to minimize heterogenity. While the sample names do not need to start with "gr," they MUST begin with some character string." Sample names can also be listed as all characters. Finally, note that wavelengths measured do not have to match the wavelengths presented in dfabs.

```
  wavelengths      gr13307     gr13351     gr13353
1          750   0.0001296  -0.0003505  -0.0003480
2          749  -0.0002367   0.0000305  -0.0000915
3          748  -0.0001582  -0.0004900  -0.0006325
4          747  -0.0004642  -0.0000105  -0.0000932
5          746  -0.0002551  -0.0000653   0.0000841
6          745  -0.0001842  -0.0002135   0.0002082
```

2.2   Fluorescence Data

Fluorescence dataframes used by functions in USGSHydroOpt should also be formatted such that each sample occupies a column, and one column contains the excitation-emission wavelength pairs (nm) for which the fluorescence measurment was made (*See example Table 2.2 below*). The example fluorescence dataframe included in this package is called **dfFluor**. The column with the excitation-emission pairs in Table 2.2 does not need to be called "Wavelength.Pairs," as it is in Table 2.2. Again since this package was developed for USGS activities, the default sample naming convention is "gr" followed by the sample number. However, the user can use fluorescence data that does not follow this naming convention. Note that the excitation and emission wavelength pairs do not need to match those presented in dfFluor.

```
  Wavelength.Pairs gr13307   gr13351 gr13353 gr13357
1          240/290 0.09457  -0.07268  0.3318  0.2664
2          240/292 0.07386   0.45433  0.5045  0.2986
3          240/294 0.08116   0.49135  0.2218  0.2547
4          240/296 0.12013  -0.12248 -0.3194  0.4069
5          240/298 0.15264   0.54694  0.4558  0.3127
6          240/300 0.16467   0.43319  0.0604  0.2691
```

2.3 Spectral Slopes Data

Information on the upper and lower wavelength (nm) for which a desired spectral slope needs to be stored in a dataframe. The example spectral slopes dataframe included in this package is called **dfsags**. The dataframe should contain exactly three columns. The first column should contain the upper wavelength (nm), the second column should contain the lower wavelength (nm), and the third column should contain the name of the spectral slope being calculated (*See example Table 2.3 below*). The columns in Table 2.3 need to be in this exact order, although the names of the columns may be different. The data types for each column are integer, integer, and character, respectively. The user can add aditional spectral slopes to Table 2.3.

```
  wvlngth1 wvlngth2       Name
1      275      290  Sag275_295
2      290      350  Sag290_350
3      350      400  Sag350_400
4      412      676  Sag412_676
```

2.4 Optical Summary Data

This dataframe contains many of the summary optical variables that can be produced using functions in USGSHydroOpt (*See example Table 2.4a below*). The example summary optical dataframe included in this package is called **dfsummary**. The functions in USGSHydroOpt calculate summary optical variables and add to a dataframe formatted according to Table 2.4a. Table 2.4a is exemplary of how the WI WSC stores optical summary variables. Note that this dataframe can contain other columns with metadata, for instance, the sample date and time, the sample ID, or whether or not the sample went through QA/QC.

```
  GRnumber         B      T      A       J FI_2005     A254
1  gr13307  0.050997 0.1826 0.4553 0.03705   1.639  0.05228
2  gr13351 −0.245602 0.4085 3.0783 0.19110   1.456  0.43531
3  gr13353 −0.175220 0.7794 6.8624 0.56309   1.523  0.68274
4  gr13357  0.111561 0.2691 0.9451 0.06969   1.572  0.08698
5  gr13360 −0.001569 0.4593 2.8254 0.37231   1.563  0.28605
6  gr13363  0.052137 0.4892 1.9518 0.19098   1.583  0.19283
```

However, also note that the naming of the summary optical variables in Table 2.4a must be identical to those specified in Table 2.4b below. For example, the user cannot name the variable Optical Brightener 1 ("OB1") in Table 2.4a as "OptBright1."

```
 [1] "OB1"       "OB2"       "OB3"       "S1.50"
 [5] "S2.50"     "S3.50"     "S1.25"     "S2.25"
 [9] "S3.25"     "Mrange.25" "Mrange.50" "B"
[13] "T"         "M"         "A"         "C"
[17] "N"         "D"         "F"         "J"
[21] "S1"        "S2"        "S3"        "H1"
[25] "H2"        "F1"        "F2"        "W"
[29] "LT1"       "LT2"       "LT3"       "LA"
[33] "HIX_2002"  "FI_2005"   "FI_2001"   "FreshI"
[37] "A254"      "A275"      "A280"      "A290"
[41] "A295"      "A350"      "A370"      "A400"
```

```
[45] "A412"        "A440"        "A488"        "A510"
[49] "A532"        "A555"        "A650"        "A676"
[53] "A715"        "Sag275_290" "Sag290_350" "Sag350_400"
[57] "Sag412_676" "Aresids"
```

2.5   Excitation-Emission (EEMs) Peak Data

The EEMs peak data contains three columns listing the name of a characterized EEM peak along with the corresponding excitation-emission wavelengths (nm) (*See example Table 2.5 below*). The example EEMs peak dataframe included in this package is called **ex_ems**. The first column, in Table 2.5, "Peak," contains the name of the characterized EEM peak. The next two columns in Table 2.5 contain the excitation and emission wavelengths (nm) at which a given peak occurs. The column names must be identical to those displayed in the example below, although the order of the columns can be different.

```
   Peak ExCA EmCA
1     B  275  304
2     T  275  340
3     M  300  390
4     A  260  450
5     C  340  440
6     N  280  370
7     D  390  510
8     F  370  460
9     J  420  460
10   S1  310  452
11   S2  280  452
12   S3  280  350
13    H  250  460
14   H1  250  320
15    F  370  470
16   F1  370  520
```

2.6   Optical Ratio and Signals Data

This dataframe contains one column called "ratioSignals," which contains all of the summary optical variables currently identified by the WI WSC (*See example Table 2.6 below*). The example of Table 2.6 included in USGSHydroOpt is called **ratioSignals**. Note that these are the same variables as those listed in Table 2.4b. The first column must contain the various "ratioSignals" that the user desires, although the column name need not be "ratioSignals."

```
  ratioSignals keep
1          OB1   NA
2          OB2   NA
3          OB3   NA
4        S1.50   NA
5        S2.50   NA
6        S3.50   NA
```

```
7        S1.25    NA
8        S2.25    NA
9        S3.25    NA
10    Mrange.25    NA
11    Mrange.50    NA
12           B    NA
13           T    NA
14           M    NA
15           A    NA
16           C    NA
17           N    NA
18           D    NA
19           F    NA
20           J    NA
21          S1    NA
22          S2    NA
23          S3    NA
24          H1    NA
25          H2    NA
26          F1    NA
27          F2    NA
28           W    NA
29         LT1    NA
30         LT2    NA
31         LT3    NA
32          LA    NA
33    HIX_2002    NA
34     FI_2005    NA
35     FI_2001    NA
36      FreshI    NA
37        A254     1
38        A275     1
39        A280     1
40        A290     1
41        A295     1
42        A350     1
43        A370     1
44        A400     1
45        A412    NA
46        A440    NA
47        A488    NA
48        A510    NA
49        A532    NA
50        A555    NA
51        A650    NA
52        A676    NA
53        A715    NA
54  Sag275_290     1
```

```
55   Sag290_350    1
56   Sag350_400    1
57   Sag412_676    1
58      Aresids   NA
```

2.7   Optical Signals Data

This optical signals dataframe is similar to "ratioSignals" except it provides more metadata about important peaks that have been characterized for EEMs plots. The example optical signals dataframe included in this package is called **signals**. The dataframe should contain six columns (*See example Table 2.7 below*). The first column in Table 2.7, "Peak," contains the name of the characterized EEM peak. The next two columns, "Ex1" and "Ex2," contain the excitation wavelength range (nm) for a given peak. "Ex1" is the lower wavelength and "Ex2" is the upper wavelength for the excitation wavelength range (nm) of a given peak. Similarily, "Em1" and "Em2" contain the emission wavelength range (nm) for a given peak. The final column, "Source," lists the source that characterized the peak. The last column, "Source," is recommended, but not required. The user must exactly replicate the column names in Table 2.7 before using functions in USGSHydroOpt.

	Peak	Ex1	Ex2	Em1	Em2	Source
1	OB1	360	NA	410	598	Hartel Turner
2	OB2	360	NA	436	436	Hartel Turner
3	OB3	365	NA	400	550	Hagedorn Turner
4	S1.50	310	NA	402	502	Sniffer
5	S2.50	280	NA	402	502	Sniffer
6	S3.50	280	NA	310	390	Sniffer
7	S1.25	310	NA	427	477	Sniffer
8	S2.25	280	NA	427	477	Sniffer
9	S3.25	280	NA	330	370	Sniffer
10	Mrange.25	300	NA	365	415	test
11	Mrange.50	300	NA	340	440	test
12	B	275	NA	304	NA	CA
13	T	275	NA	340	NA	CA
14	M	300	NA	390	NA	CA
15	A	260	NA	450	NA	CA
16	C	340	NA	440	NA	CA
17	N	280	NA	370	NA	CA
18	D	390	NA	510	NA	CA
19	F	370	NA	460	NA	CA
20	J	420	NA	460	NA	CA
21	S1	310	NA	452	NA	CA
22	S2	280	NA	452	NA	CA
23	S3	280	NA	350	NA	CA
24	H1	250	NA	460	NA	Ohno2002
25	H2	250	NA	320	NA	Ohno2002
26	F1	370	NA	470	NA	Cory and McKnight, 2005
27	F2	370	NA	520	NA	Cory and McKnight, 2005
28	W	255	290	302	350	
29	LT1	250	NA	340	NA	

```
30        LT2 260  NA 340   NA
31        LT3 240  NA 340   NA
32         LA 240  NA 440   NA
```

2.8   3-Dimensional Excitation-Emission Array

A 3-D array with fluorescence data is used by many of the functions in USGSHydroOpt. The first dimension contains the excitation wavelengths (nm) as data type **character**, for which a given fluorescence measurement was made. The second dimension contains the emission wavelengths (nm) as data type **character**, for which a given fluorescence measurement was made. The third dimension contains the sample names as data type **character** for a given observation. The user must ensure that the third dimension of the array contains the sample numbers or names. Again, in this example the default "gr" followed by the sample number is used as a naming convention for samples. The user can use sample names that do not match this naming convention. The example 3-D array with EEMs data included in this package is called **a**.

To view the categories for each dimension, use the following commands along with the example 3-D EEM array included in USGS HydroOpt:

```
# this command shows the excitation wavelengths (nm)
colnames(a)

  [1] "290" "292" "294" "296" "298" "300" "302" "304" "306"
 [10] "308" "310" "312" "314" "316" "318" "320" "322" "324"
 [19] "326" "328" "330" "332" "334" "336" "338" "340" "342"
 [28] "344" "346" "348" "350" "352" "354" "356" "358" "360"
 [37] "362" "364" "366" "368" "370" "372" "374" "376" "378"
 [46] "380" "382" "384" "386" "388" "390" "392" "394" "396"
 [55] "398" "400" "402" "404" "406" "408" "410" "412" "414"
 [64] "416" "418" "420" "422" "424" "426" "428" "430" "432"
 [73] "434" "436" "438" "440" "442" "444" "446" "448" "450"
 [82] "452" "454" "456" "458" "460" "462" "464" "466" "468"
 [91] "470" "472" "474" "476" "478" "480" "482" "484" "486"
[100] "488" "490" "492" "494" "496" "498" "500" "502" "504"
[109] "506" "508" "510" "512" "514" "516" "518" "520" "522"
[118] "524" "526" "528" "530" "532" "534" "536" "538" "540"
[127] "542" "544" "546" "548" "550" "552" "554" "556" "558"
[136] "560" "562" "564" "566" "568" "570" "572" "574" "576"
[145] "578" "580" "582" "584" "586" "588" "590" "592" "594"
[154] "596" "598" "600"

# this command shows the emission wavelengths (nm)
rownames(a)

 [1] "240" "245" "250" "255" "260" "265" "270" "275" "280"
[10] "285" "290" "295" "300" "305" "310" "315" "320" "325"
[19] "330" "335" "340" "345" "350" "355" "360" "365" "370"
[28] "375" "380" "385" "390" "395" "400" "405" "410" "415"
[37] "420" "425" "430" "435" "440"
```

```
# this command shows the emission wavelengths (nm), only the
# first 20 shown for simplicity
names(a[1, 1, ])[1:20]

 [1] "gr13307" "gr13351" "gr13353" "gr13357" "gr13360"
 [6] "gr13363" "gr13364" "gr13365" "gr13374" "gr13375"
[11] "gr13433" "gr13434" "gr13435" "gr13439" "gr13440"
[16] "gr13442" "gr13465" "gr13467" "gr13478" "gr13505"
```

The user should be aware of **two important caveats**: (1) There should rarely be emission wavelengths (nm) below the excitation wavelength for a given fluorescence reading. Where this occurs an NA will be found in the 3-D EEMs array. (2) Intensities at an emission wavelength that is two times the excitation wavelength will be influenced by second order Rayleigh scatter. The user should keep this in mind during any optical analysis of water.

2.9 Creating a 3-Dimensional Excitation-Emission Array

In Section 2.8 the format of 3-D arrays of fluorescence data that can be used with USGSHydroOpt is discussed. USGSHydroOpt can also produce such 3-D EEMs arrays given the appropriate fluorescence dataframe. Below is an example of how USGSHydroOpt is used to accomplish this task:

```
# set an arbitrary data frame (df) as dfFluor (the example
# fluorescence dataframe in USGSHydroOpt)
df <- dfFluor

# define the column in dfFluor
ExEm <- "Wavelength.Pairs"

# assign a variable grnum to the column in dataSummary with
# sample names
grnum <- "GRnumber"

# run the VectorizedTo3DArray function from USGSHydroOpt that
# creates a 3-D EEMs array given a vectorized fluorescence
# dataframe
aTest <- VectorizedTo3DArray(df, ExEm, grnum)
```

In the example above, dfFluor is formatted according to the fluorescence dataframe discussed in Section 2.2.

3   Creating Summary Optical Variables

Most of the functions included in USGSHydroOpt are for creating variables that summarize optical data for water samples. Such variables can then be used in statistical modeling efforts aimed at describing various phenomena in aquatic ecosystems. This sections steps through the six functions that USGSHydroOpt offers for creating summary optical variables.

3.1 Creating Absorbance Coefficients

The function getAbs opperates on a dataframe formatted according to Section 2.1. Specifically, it picks out absorbance coefficients for a defined set of wavelengths (nm),'wavs,' and adds those coefficients to an optical summary dataframe formatted according to Section 2.4 (for each sample). If the wavelengths in the user specified absorbance dataframe do not match those specified in wavs, the function finds the nearest wavelength to those specified in wavs. The name of the absorbance coefficient is then changed to match the nearest wavelength specified in wavs. Shown below is an example of how the function is used.

```
# assign a variable dataAbs to the absorbance dataframe
# included in USGSHydroOpt
dataAbs <- dfabs

# define the column with the wavelengths in dataAbs
waveCol <- "wavelengths"

# define the wavelengths for which absorbance coefficients
# should be defined
wavs <- c(430, 530, 630, 730)

# define which columns contain the samples
colSubsetString <- "gr"

# assign a variable dataSummary to the dfsummary dataframe
# included in USGSHydroOpt
dataSummary <- dfsummary

# assign a variable grnum to the column in dataSummary with
# sample names
grnum <- "GRnumber"

# use getAbs to produce absorbance coefficients
testAbs <- getAbs(dataAbs, waveCol, wavs, colSubsetString, dataSummary,
    grnum)

# note that the absorbance coefficients as defined by wavs
# have been added to dataSummary
colnames(testAbs)[69:72]

[1] "A430" "A530" "A630" "A730"
```

3.2 Spectral Slopes by Linear Regression

The function getSag also computes spectral slopes by a first order decay function determined by linear regression(Helms et al. 2008). However, only the spectral slopes (for each sample) are added to the absorbance summary dataframe formatted according to Section 2.4. Displayed below is an example of how getSag is used to compute spectral slopes. Note that a dataframe formatted like dfsags (See Section 2.3) is required to define the desired spectral slopes. In some cases, the user defined absorbance dataframe may not match the

wavelengths (wvlngth1,wvlngth2) specified in the example spectral slopes dataframe (dfsags). If this is true, the function finds the nearest (wvlngth1,wvlngth2) and uses that to compute the spectral slope. The name of the spectral slope is then changed to match the wvlngth1 and wvlngth 2 that was used to calculate the spectral slope. That is, the 'Name' column in the sag dataframe is updated to reflect the change in wvlngth1 and wvlngth2 used by the function. The updated name will be added to the optical summary dataframe.

```r
# assign a variable dataAbs to a shortened version of the
# absorbance dataframe included in USGSHydroOpt
dataAbs <- dfabs

# define the column with the wavelengths in dataAbs
waveCol <- "wavelengths"

# define the dataframe with the upper and lower wavelengths
# and name of spectral slope the example dataframe in
# USGSHydroOpt with this info is called 'dfsags'
sag <- dfsags

# define which columns contain the samples
colSubsetString <- "gr"

# assign a variable dataSummary to the dfsummary dataframe
# included in USGSHydroOpt
dataSummary <- dfsummary
sags <- grep("Sag", colnames(dataSummary))

# remove columns with spectral slopes and re-compute with
# getSag
dataSummary <- dataSummary[, -c(sags)]

# assign a variable grnum to the column in dataSummary with
# sample names
grnum <- "GRnumber"

# use getSag to compute spectral slopes and add slopes to
# optical summary dataframe
testSag <- getSag(dataAbs, waveCol, sag, colSubsetString, dataSummary,
    grnum)

# note that the spectral slopes defined in sag have been
# added to dataSummary
colnames(testSag)[65:68]
```

3.3   Residuals of Spectral Slopes by Linear Regression

The function getExpResid computes spectral slopes by a first order decay function determined by linear regression (Helms et al. 2008).The residual at a specified wavelength (nm) is then calculated based on the spectral slope. The residual at a given wavelength and spectral slope is then added to a summary dataframe

11

formatted according to Section 2.4.

Displayed below is an example of how the function can be used. The function produces a plot with the absorbance data for each sample. On the plot the red indicates the spectral slope fit by linear regression, the blue is the absorbance data in rangeGap, and the black is the data in rangeReg but not in rangeGap. The dataframe dataAbs is formatted according to Section 2.1, but is shortened for simplicity. If the user calls "pdf(genericName.pdf)," *See* ?pdf, then the function plots will be printed to a Portable Document Format in the working directory (one plot or sample per page).

```r
# absorbance wavelength (nm) for which residual is calculated
wavelength <- 267

# the absorbance wavelength range (nm) to be considered as a
# numeric string
rangeReg <- c(240, 340)

# the absorbance wavelength range (nm) to be considered as a
# numeric string
rangeGap <- c(255, 300)

# assign a variable dataAbs to a shortened version of the
# absorbance dataframe included in USGSHydroOpt
dataAbs <- dfabs

# define the column with the wavelengths in dataAbs
waveCol <- "wavelengths"

# define which columns contain the samples
colSubsetString <- "gr"

# assign a variable dataSummary to the dfsummary dataframe
# included in USGSHydroOpt column 68 or 'Aresids' is removed
# because we are computing this summary optical variable with
# this function and then adding it to dataSummary
dataSummary <- dfsummary[, -c(68)]

# assign a variable grnum to the column in dataSummary with
# sample names
grnum <- "GRnumber"

# use getExpResid to calculate the residual at a given
# wavelength given a spectral slope calculated per Helms et
# al. 2008.
testdfOpt <- getExpResid(wavelength, rangeReg, rangeGap, dataAbs,
    waveCol, colSubsetString, dataSummary, grnum)

# notice that the variable 'Aresids' has been added to
# dataSummary
colnames(testdfOpt)
```

3.4 Humification and Fluorescence Indices

Humification and fluorescence index summary variables can be computed using the function getIndexes. The humification index summary variable called "HIX_2002" is calculated according to Ohno (2002). The first fluorescence index summary variable "FI_2005" is computed according to Cory and McKnight (2005). The second fluorescence index summary variable, "FI_2001" is computed according to McKnight et al. (2001). The final summary variable produced by this function is the freshness index, "FreshI," is computed according to Parlanti et al. (2000). Each of these four summary variables are computed and added to the optical summary dataframe, formatted according to Section 2.4.

These humification and fluorescence index summary variables are computed using specific excitation and emission wavelengths. The user does not need to have match the exact excitation and emission wavelengths in the defined 3-D excitation emission array. The function will find the nearest excitation or emission wavelength closest to that which is required, and use those wavelengths instead.

In the example below, getIndexes is used to compute these four optical variables, which are subsequently added to an optical summary dataframe. A 3-D excitation-emission dataframe with fluorescence data formatted according to Sections 2.8-2.9 is used in computing these summary variables.

```
# set a variable called a as the example 3-D excitation
# emission array included with the package
a <- a

# assign a variable dataSummary to the dfsummary dataframe
# included in USGSHydroOpt
dataSummary <- dfsummary

# remove those columns with the fluorescence and humic
# indices that we are computing here
dataSummary <- dataSummary[, -c(43:46)]

# assign a variable grnum to the column in dataSummary with
# sample names
grnum <- "GRnumber"

# use getIndexes to compute the four humification and
# fluorescence indices
testIndexes <- getIndexes(a, dataSummary, grnum)

# note that the four indices have been added to dataSummary
colnames(testIndexes)[65:68]

[1] "HIX_2002" "FI_2005"  "FI_2001"  "FreshI"
```

3.5 Optical Ratios

The function getRatios uses absorbance peaks and spectral slopes in an existing summary optical data frame (e.g., dfsummary) and creates ratios between the different peaks and ratios. These ratios can be useful as predictor variables in statistical modeling efforts. In the example below, 65 different ratios are computed

using an optical summary dataframe formatted per Section 2.4, and a ratio signals dataframe formatted per Section 2.6. The names of the calculated ratios added to the optical summary dataframe begin with "r" to signify "ratio" followed by the absorbance peak and/or spectral slope used to calculate the ratio. For example, the ratio of the absorbance peak at 254nm (A254) and the spectral slope between 350 and 400 nm (Sag350_400) is called "rA254_Sag350_400." Note that the sigs do not need to be defined as they are in this example. However, the user should ensure that the names defined in sigs correspond to column names in dataSummary.

```r
# assign a variable dataSummary to the dfsummary dataframe
# included in USGSHydroOpt
dataSummary <- dfsummary

# note the number of variables in dataSummary
length(colnames(dataSummary))

[1] 68

# pick out the absorbance peaks and spectral slopes to be
# used for calculating ratios these correspond to those with
# a 1 in the 'keep' column.
sigs <- ratioSignals[which(ratioSignals[2] > 0), 1]

# assign a variable grnum to the column in dataSummary with
# sample names
grnum <- "GRnumber"

# use getRatios to calculate 65 different ratios of
# absorbance peaks and spectral slopes
test <- getRatios(dataSummary, sigs, grnum)

# notice that 65 ratios have been added to dataSummary
length(colnames(test))

[1] 134

# example of some ratios added
colnames(test)[69:75]

[1] "rA254_A275" "rA254_A280" "rA254_A290" "rA254_A295"
[5] "rA254_A350" "rA254_A370" "rA254_A400"
```

3.6 EEM Peaks Computed by Average Wavelength

Various excitation-emission (EEMs) peaks can be used to predict the presence of different chemical constituents in water. An EEMs peak is always characterized at a specific excitation and emission wavelength (nm), or within specific excitation and emission wavelength ranges. It has become standard practice to extract these peaks from fluorescence data, and identify them on EEMs plots. When given a fluorescence dataframe formatted according to section 2.2, the function getMeanFl computes the various peaks based on a dataframe of signals formatted according to section 2.7.

Many of these peaks can occur within an excitation wavelength (nm) range, and also an emission wavelength (nm) range. For peaks where this is true, the function computes the average of the excitation and/or emission wavelength (nm) range. The resultant mean excitation and/or emission wavelength (nm) is then used to compute the EEMs peak from the fluorescence data. NOte that the user defined excitation emission array does not need to have the exact excitation or emission wavelengths required for computing a given peak. The function will find the nearest wavelength to that which is required for calculating the peak and use that instead. An example of how this function can be used is illustrated below.

```
# set a variable a as the example 3-D excitation emission
# array included with the package
a <- a

# set a variable signals as the example signals dataframe
signals <- signals

# note the length of the EEMs peaks to be computed
length(rownames(signals))

[1] 32

# identify the name of the column with the EEMs Peak names
Peak <- "Peak"

# identify column with the lower excitation wavelength in the
# excitation wavelength range
Ex1 <- "Ex1"

# identify column with the upper excitation wavelength in the
# excitation wavelength range
Ex2 <- "Ex2"

# identify column with the lower emission wavelength in the
# emission wavelength range
Em1 <- "Em1"

# identify column with the upper emission wavelength in the
# emission wavelength range
Em2 <- "Em2"

# assign a variable dataSummary to the dfsummary dataframe
# included in USGSHydroOpt
dataSummary <- dfsummary

# remove the variables in dataSummary that are going to be
# computed with testMeanFl
rm <- which(colnames(dataSummary) %in% signals$Peak)
dataSummary <- dataSummary[, -c(rm)]

# note the number of variables in dataSummary
length(colnames(dataSummary))
```

```
[1] 36

# assign a variable grnum to the column in dataSummary with
# sample names
grnum <- "GRnumber"

# use getMeanFl to compute the different EEMs signals and add
# them to the optical summary data frame
testMeanFl <- getMeanFl(a, signals, Peak, Ex1, Ex2, Em1, Em2,
    dataSummary, grnum)

# notice that signals have been added to dataSummary
length(colnames(testMeanFl))

[1] 68

length(rownames(signals)) + length(colnames(dataSummary))

[1] 68
```

3.7 Log transformation of Summary Optical Variables

Often times it is useful to log 10 tranform summary optical variables to help stabilize variance. This is especially true when using optical variables as predictors for biological response variables such as bacteria and viruses. The function getLog10 log10 transforms user defined optical summary variables to a log10 scale. The function will return NA for values in optical summary variables that are less than 0.

```
# select those variables from ratioSignals which should be
# log 10 transformmed in dataSummary
signals <- ratioSignals[which(ratioSignals[2] > 0), 1]

# note the number of variables to be log 10 transformed
length(signals)

[1] 12

# assign a variable dataSummary to the dfsummary dataframe
# included in USGSHydroOpt
dataSummary <- dfsummary

# note the number of variables in dataSummary
length(colnames(dataSummary))

[1] 68

# assign a variable grnum to the column in dataSummary with
# sample names
grnum <- "GRnumber"

# use getLog10 to log 10 transform signals in dataSummary
```

```
testLog10 <- getLog10(dataSummary, signals, grnum)

# note that the log 10 transformed signals have been added to
# dataSummary
length(colnames(testLog10))

[1] 80

length(signals) + length(colnames(dataSummary))

[1] 80
```

4 Plotting Excitation-Emission (EEMs) Spectra

One of the most useful visualizations of fluorescence data is creating a contour plot that includes the intensities of all excitation-emission wavelength (nm) pairs for a given sample. The function plotEEMs2 creates contour plots of EEMs spectra and also labels important EEM peaks. Displayed below is an example of how plotEEMs2 can be used to plot EEMs spectra. The 3-D Excitation-Emission array discussed in Sections 2.8-2.9 is used to produce the EEMs plot.

```
# choose a sample of interest from the 3-D EEMs array
GRnum <- c("gr13307")

# create a matrix from the 3-D EEMs array with only data for
# sample GRnum
mat <- a[, , GRnum]

# define the excitation wavelengths (nm) as Ex data type
# numeric
Ex <- as.numeric(names(a[, 1, 1]))

# define the emission wavelengths (nm) as Em data type
# numeric
Em <- as.numeric(names(a[1, , 1]))

# define the number of numeric color levels for contour plot
nlevels <- 50

# set a variable Peaks as the example dataframe ex_ems
Peaks <- ex_ems

# define the column in Peaks which contains the name of EEMs
# peaks
peakCol <- "Peak"

# define the column in Peaks which contains the excitation
# wavelength for a given Peak
peakEx <- "ExCA"
```
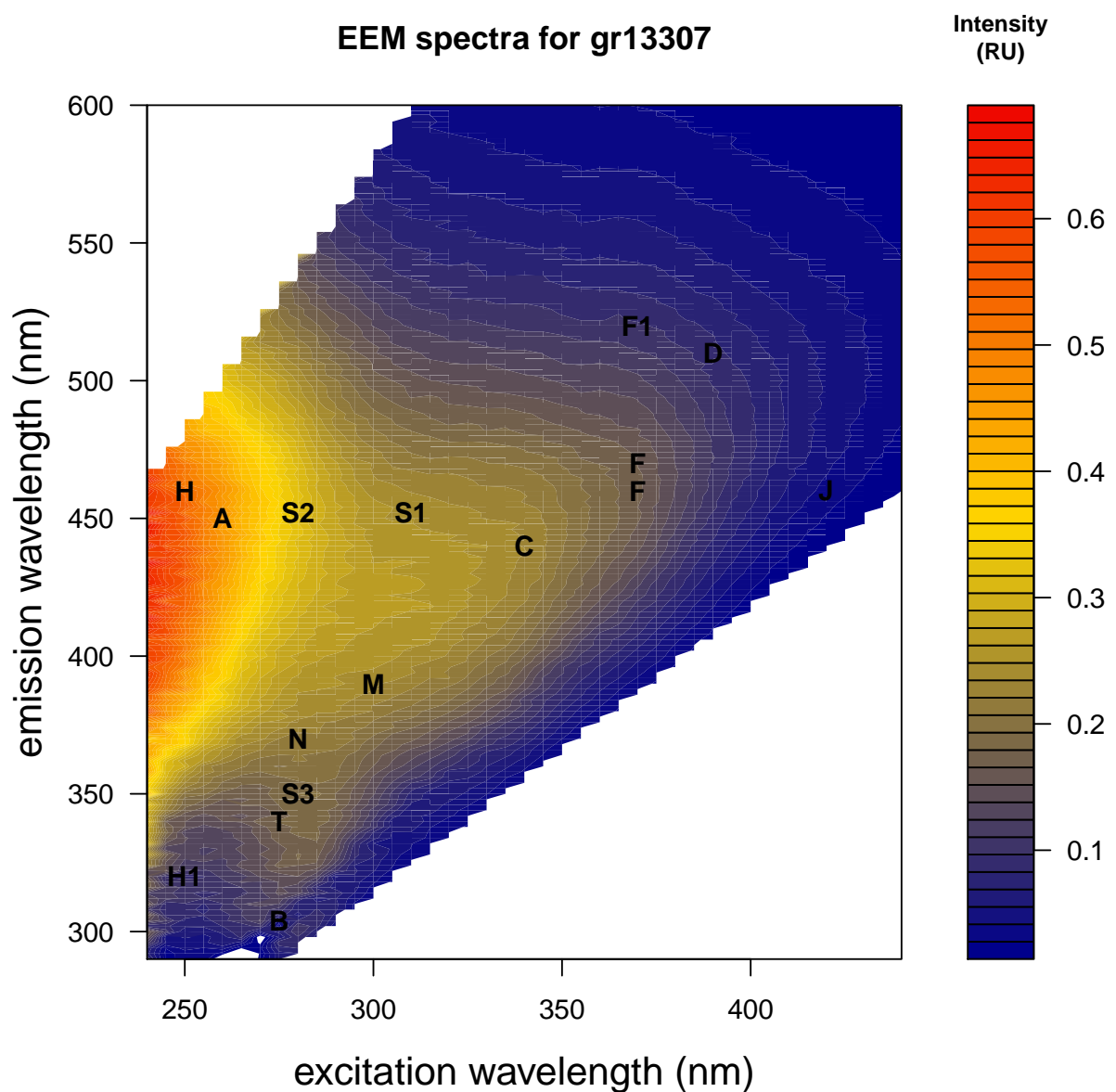
```
# define the column in Peaks whcih contains the emission
# wavelength for a given Peak
peakEm <- "EmCA"

# assign the main plot title
mainTitle <- paste("EEM spectra for", GRnum)

# use plotEEMs2 to create a contour plot of EEM spectra
exampleEEMs2 <- plotEEMs2(mat = mat, Ex = Ex, Em = Em, nlevels = nlevels,
    Peaks = Peaks, peakCol = peakCol, peakEx = peakEx, peakEm = peakEm,
    mainTitle = mainTitle)
```

5 Conclusion

USGSHydroOpt is a useful R package for creating optical summary variables for absorbance and fluorescence spectra. These optical predictor variables are useful for explaining biological and chemical phenomena in water samples. This package also contains a useful function for visualizing fluorescence spectra by creating EEMs contour plots. A downfall of the package is that it is designed for dataframes and arrays that adhere to standardized formats, and therefore it is not ammenable to schemas that deviate from the prescribed formats.

6 References

Cory RM, McKnight DM. 2005. Fluorescence spectroscopy reveals ubiquitous presence of oxidized and reduced quinones in dissolved organic matter. Environmental Science & Technology 39: 8142-8149.

Helms JR, Stubbins A, Ritchie JD, Minor EC, Kieber DJ, Mopper K. 2008. Absorption spectral slopes and slope ratios as indicators of molecular weight, source, and photobleaching of chromophoric dissolved organic matter. Limnology and Oceanography 53: 955-969.

McKnight DM, Boyer EW, Westerhoff PK, Doran PT, Kulbe T, Andersen DT. 2001. Spectrofluorometric characterization of dissolved organic matter for indication of precursor organic material and aromaticity. Limnology and Oceanography 46: 38-48.

Ohno T. 2002. Fluorescence inner-filtering correction for determining the humification index of dissolved organic matter. Environmental Science & Technology 36: 742-746.

Parlanti E, Worz K, Geoffroy L, Lamotte M. 2000. Dissolved organic matter fluorescence spectroscopy as a tool to estimate biological activity in a coastal zone submitted to anthropogenic inputs. Organic Geochemistry 31: 1765-1781.