

**Changes of dissolved organic matter following salinity invasion in different seasons in a nitrogen rich
tidal reach**

Rongrong Xie ^{1,2,3,*}, Jiabin Qi^{1#}, Chengchun Shi⁴, Peng Zhang⁵, Rulin Wu¹, Jiabing Li ^{1,2,3}, Joanna J. Waniek ^{6*}

1. College of Environmental Science and Engineering, Fujian Normal University, Fuzhou 350007, China

2. Key Laboratory of Pollution Control and Resource Recycling of Fujian Province, Fujian Normal University, Fuzhou 350007, China

3. Digital Fujian environmental monitoring Internet of things Laboratory, Fujian Normal University, Fuzhou 350007, China

4. Fujian Research Academy of Environmental Sciences, Fuzhou 350013, China

5. School of Environmental and Municipal Engineering, North China University of Water Resources and Electric Power, Zhengzhou 450046, China

6. Leibniz Institute for Baltic Sea Research, Warnemünde, Rostock 18119, Germany

The authors contributed equally to this work.

*Correspondence to :

Dr. Rongrong Xie:

Address: 8 Shangsang Rd, Fuzhou, 350007, China

Email: xierr1118@163.com

Tel: +86 15205006921

Prof. Dr. Joanna J. Waniek:

Address: 15 Seestraße, Rostock, 18119, Germany

Email: joanna.waniek@io-warnemuende.de

Tel: 049 381 5197300

Contents

Supplementary text 1

Supplementary Table S1

Supplementary Figure S1

Supplementary text 1 Python script for the cosine-histogram similarity

```
# -*- coding: utf-8 -*-

import os

from PIL import Image

from numpy import average, linalg, dot

def get_thumbnail(image, size=(256, 256), greyscale=False):

    image = image.resize(size, Image.ANTIALIAS)

    if greyscale:

        image = image.convert('L')

    return image

def image_similarity_vectors_via_numpy(image1, image2):

    image1 = get_thumbnail(image1)

    image2 = get_thumbnail(image2)

    images = [image1, image2]

    vectors = []

    norms = []

    for image in images:

        vector = []

        for pixel_tuple in image.getdata():

            vector.append(average(pixel_tuple))

        vectors.append(vector)

        norms.append(linalg.norm(vector, 2))

    a, b = vectors

    a_norm, b_norm = norms

    res = dot(a/a_norm, b/b_norm)

    return res

def make_regalur_image(img, size = (256, 256)):

    return img.resize(size).convert('RGB')

def hist_similar(lh, rh):

    assert len(lh) == len(rh)
```

```

        return sum(1 - (0 if l == r else float(abs(l - r))/max(l, r)) for l, r in zip(lh, rh))/len(lh)

def calc_similar(li, ri):

    return hist_similar(li.histogram(), ri.histogram())

def compare(path1, path2):

    img1 = Image.open(path1)

    img2 = Image.open(path2)

    cosin_similarity = image_similarity_vectors_via_numpy(img1, img2)

    hist_img1 = make_regalur_image(img1)

    hist_img2 = make_regalur_image(img2)

    hist_similarity = calc_similar(img1, img2)

    similarity = (cosin_similarity + hist_similarity)/2

    return similarity

path1 = '/Users/apple/Desktop/02.jpg'

path2 = '/Users/apple/Desktop/01.jpg'

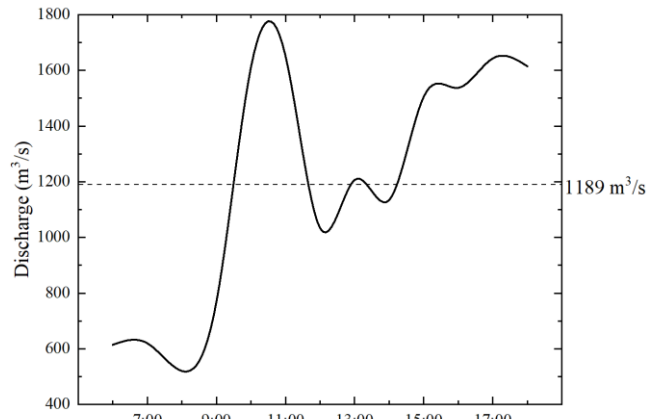
cosin = compare(path1, path2)

print(cosin)

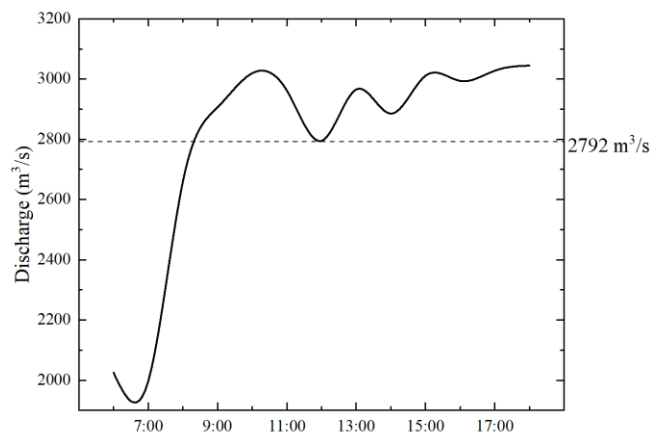
```

Supplementary Table S1 Pairwise test evaluating the significance of possible differences between the sampling stations.

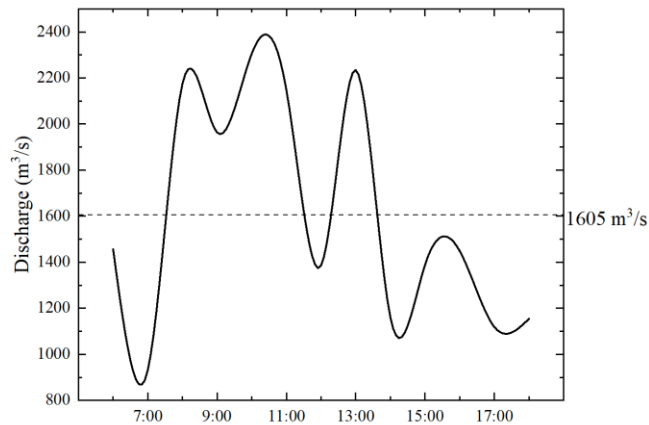
Station Groups	R-value	P-value
S1/ S1	1	1
S1/ S2	-0.407	1
S1/ S3	-0.296	0.8
S1/ S4	-0.407	1
S1/ S5	-0.37	0.8
S1/ S6	-0.074	0.5
S1/ S7	-0.259	0.8
S1/ S8	-0.222	0.8
S1/ S8-F	-0.407	1
S1/ S9	-0.222	0.8
S1/ S9-F	0.185	0.3



November 2018



April 2019



August 2019

Supplementary Figure S1 Variations of the freshwater discharge regulated by the Shuikou dam during the day time (6:00-18:00) of sampling days in November 2018, April 2019 and August 2019.