

# Irregular2dWavelets

*Joseph Antonelli*

---

**2018-01-26**

---

Here we will present how to use the package Irregular2dWavelets, which performs two-dimensional wavelet decompositions on irregularly spaced grids. The primary purpose of this package is to perform the two-dimensional wavelet decompositions on irregular shapes and grids as seen in Antonelli et. al (2016) (<http://arxiv.org/abs/1609.05186>). It is also possible to implement one-dimensional wavelets on irregular grids using the functions in this package, however, these ideas come from (Wand and Ormerod 2011), and are generally only included in the package as they are needed in order to build the 2d wavelets. This document will walk the user through the steps required to install the package from github, implement the main functions of the package, and perform 2d wavelet decompositions on simulated pollution data for the Boston area.

## Installation

To install the package the user must first download the R package, devtools, which can be achieved with the following line of code

```
library(devtools)
```

Once devtools is installed then the package can be loaded from github using the following

```
install_github(repo = "jantonelli111/Irregular2dWavelets")  
library(Irregular2dWavelets)
```

Now that the package is installed and loaded we can begin to describe how it is used for 2d wavelet decompositions.

## Data example

To motivate and illustrate the ideas seen in the package we will be using simulated data that is loosely based on  $PM_{2.5}$  data from the Boston area. The data is a part of the package and can be loaded as follows

```
data(BostonPMsim)
```

This data is loosely based on pollution measurements found in (Kloog et al. 2012), though it has been changed in a manner such that the resulting values no longer have any scientific interpretation. It will, however, resemble the surface of pollution in the Boston area and will be useful to illustrate the ideas of breaking the surface of pollution into separate spatial scales. Let's first examine the data to get an idea of it's structure.

```
head(BostonPMsim)
```

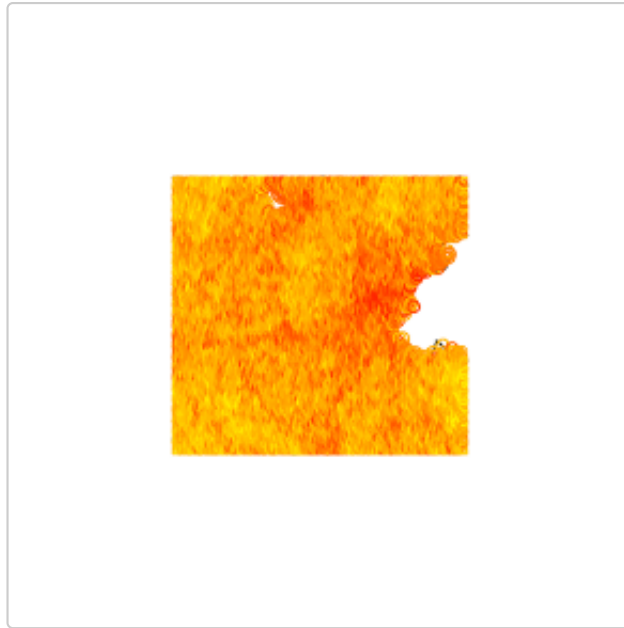
```
##      lat      long      pm
## 1 42.58426 -71.79859 4.051212
## 2 42.60908 -71.79808 4.030715
## 3 42.63568 -71.79732 3.944956
## 4 42.66359 -71.79712 3.999458
## 5 42.47489 -71.79625 3.929886
## 6 42.50148 -71.79829 3.995344
```

So we see that the data has 3 columns: latitude, longitude, and the simulated  $PM_{2.5}$  values. Now we can also visualize the data:

```
library(maps)
```

```
## Warning: package 'maps' was built under R version 3.3.2
```

```
colors = colorRampPalette(c("yellow", "orange", "red"))
BostonPMsim$col = colors(100)[as.numeric(cut(BostonPMsim$pm,breaks = 100))]
map("state","massachusetts", xlim=range(BostonPMsim$long),
    ylim=range(BostonPMsim$lat))
points(BostonPMsim$long, BostonPMsim$lat, col=BostonPMsim$col, cex=.75)
```



## Two-dimensional decomposition

Now we will apply the main function of the package, which performs the two dimensional wavelet decomposition

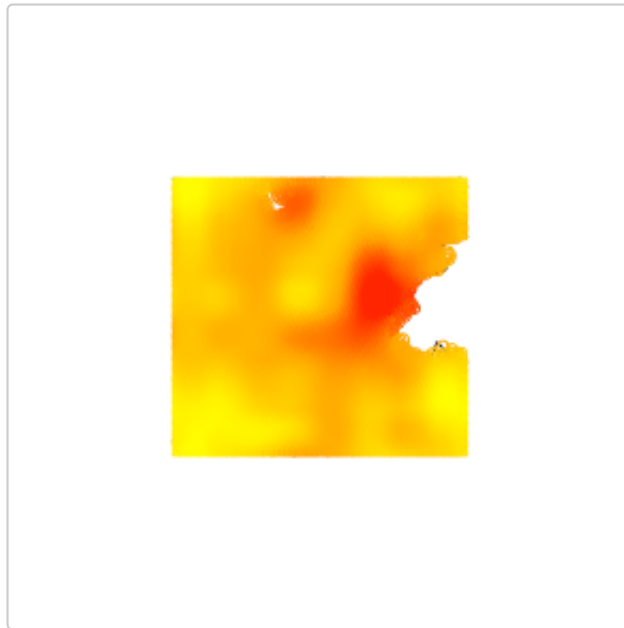
```
numLevels = 4
Decomp = Irregular2dWavelet(x=BostonPMsim$long,
                           y=BostonPMsim$lat,
                           f=BostonPMsim$pm,
                           numLevels=numLevels)
```

So we now have the estimated wavelet coefficients and we can use these to examine different spatial scales of the surface. This is where the `threshold2d` function comes into play. The `threshold2d` function takes in the scales that the user wishes to hard threshold coefficients to zero.

```
threshold.betalow4 <- threshold2d(Decomp$beta, numLevels,
                                   remove.x=4:numLevels, remove.y=4:numLevels)
threshold.fHatlow4 <- Decomp$Zxy%%threshold.betalow4
```

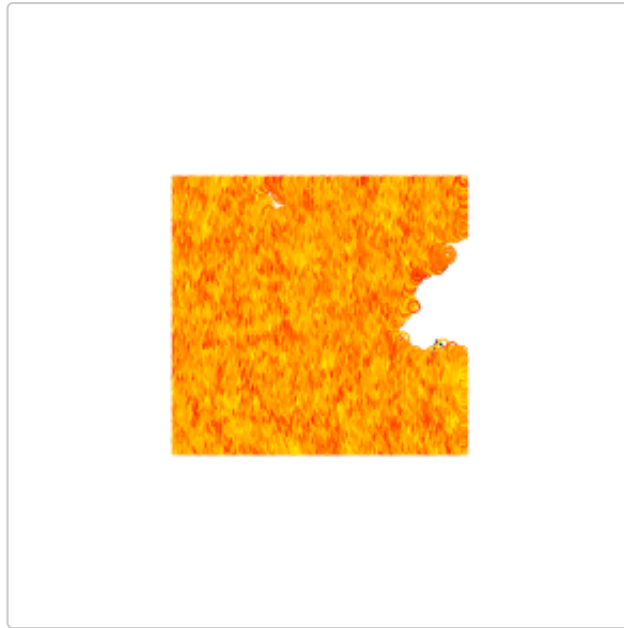
And now we can plot the resulting surface, which should be much smoother as we have removed any signal that was captured by scales 4 through numLevels.

```
threshold.col = colors(100)[as.numeric(cut(threshold.fHatlow4,breaks = 100))]  
map("state","massachusetts", xlim=range(BostonPMsim$long),  
ylim=range(BostonPMsim$lat))  
points(BostonPMsim$long, BostonPMsim$lat, col=threshold.col, cex=.75)
```



We could also plot the high frequency component, which is taken as the difference between the true surface and the smooth surface.

```
threshold.high4 = BostonPMsim$pm - threshold.fHatlow4  
threshold.col.high = colors(100)[as.numeric(cut(threshold.high4,breaks =  
100))]  
map("state","massachusetts", xlim=range(BostonPMsim$long),  
ylim=range(BostonPMsim$lat))  
points(BostonPMsim$long, BostonPMsim$lat, col=threshold.col.high, cex=.75)
```



Kloog, Itai, Steven J Melly, William L Ridgway, Brent A Coull, Joel Schwartz, and others. 2012. "Using New Satellite Based Exposure Methods to Study the Association Between Pregnancy Pm2. 5 Exposure, Premature Birth and Birth Weight in Massachusetts." *Environmental Health* 11 (1). BioMed Central: 1–8.

Wand, MP, and John T Ormerod. 2011. "Penalized Wavelets: Embedding Wavelets into Semiparametric Regression." *Electronic Journal of Statistics* 5. Institute of Mathematical Statistics: 1654–1717.