# Making the Next Generation of the 3D Interstellar Medium Dust Temperature Maps

Oluwasemoore Tijani

## Introduction

The aim of this project was to build up on the 3D interstellar dust temperature map produced by Ioana Zelko, by developing a method to shift superpixels of the map in different directions, in order to create various new versions of the map. The project jupyter notebook can be found at `https://github.com/Semooretj/SURP/tree/main/project`

## Methods

The HEALPix system used to create the map works by arranging an array of pixels into ring ordering. From the project jupyter notebook, the first function called ***'turn'*** shifts pixels to the left along their colatitude ring (see Figure 1). It does this by grouping all the indices into lists according to their ring, then moving the last element of each ring to be the first element, and then returns the resulting array with new positions.

Another method was using the *healpy.Rotator* class to create a rotation function that rotates based on euler angles given. As shown in ***'rot'*** function, it converts the given data into colatitudes $\theta$ and longitudes $\phi$, applies the given rotation, then returns the data back into pixels. Like the ***'turn'*** function, the ***'rot'*** function was successfully used to shift pixels left or right along their rings, by changing $\alpha$ component of euler angle in ZYX convention (see Figure 2). However, it was unsuccessful when shifting pixels upwards or downwards to new rings possibly because the colatitudes are not evenly spaced, resulting in some positions having the same value.

The last method used was based on the *healpy.get_all_neighbours function*, which gives an array of all the neighbours around a given pixel. The ***'shift_SW'*** function uses this function to get the index of the SW neighbour for each pixel and replaces the value at each position with this SW value, therby shifting the map north-eastwards. This method was mostly successful (see Figure 3), however, some pixels at the top row were lost while the bottom rows had some repeated pixels. This method would have similar results if SE, NW or NE neighbour is used instead, but would encounter some null values for N, E, W or S direction. At higher resolutions, the shift would be mostly accurate except for the poles of the map.

## Figures

The figures below are example outputs of Nside 1 map for each of the 3 methods described, using the same test data of 12 pixels ranging evenly from 0 to 11.

## Rotation using ’turn’ function

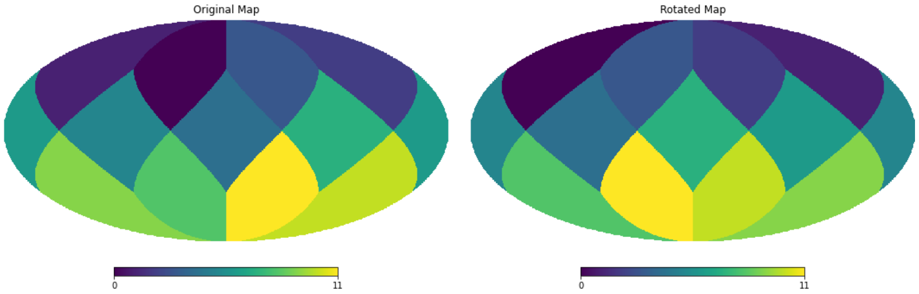

Figure 1: Rotating test Nside 1 map by 1 pixel to the left

## Rotation using ’rot’ function

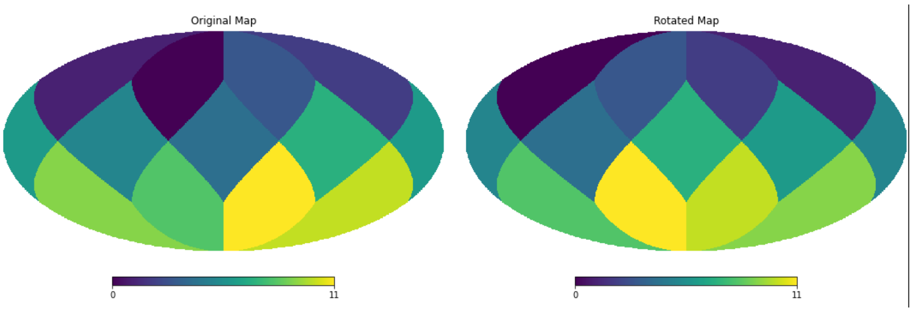

Figure 2: Rotating test Nside 1 map by [90,0,0] euler angle

## Rotation using ’shift_SW’ function

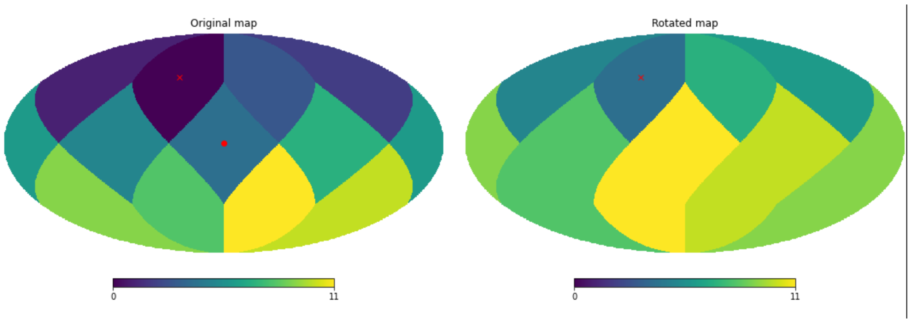

Figure 3: Shifting test Nside 1 map by 1 pixel northeast