import healpy as hp

import numpy as np

import matplotlib.pyplot as plt

from scipy.stats import pearsonr

# Load SMICA CMB map

cmb\_map = hp.read\_map("COM\_CMB\_IQU-smica\_2048\_R3.00\_oe2.fits")

nside = hp.get\_nside(cmb\_map)

npix = hp.nside2npix(nside)

# Coordinates of a known hot spot (approx): l = 220°, b = -30°

theta\_hot = np.radians(90 + 30)  # colatitude

phi\_hot = np.radians(220)

# Parameters

num\_rings = 10

max\_radius\_deg = 50

num\_trials = 1000

# Ring profile extractor

def ring\_profile(field, theta, phi, nside, npix):

    vec = hp.ang2vec(theta, phi)

    vecs = np.array(hp.pix2vec(nside, np.arange(npix)))

    dots = vecs[0]\*vec[0] + vecs[1]\*vec[1] + vecs[2]\*vec[2]

    dots = np.clip(dots, -1.0, 1.0)

    angs\_deg = np.degrees(np.arccos(dots))

    edges = np.linspace(0, max\_radius\_deg, num\_rings + 1)

    profile = []

    for i in range(num\_rings):

        mask = (angs\_deg >= edges[i]) & (angs\_deg < edges[i+1])

        profile.append(np.mean(field[mask]))

    return profile

# Extract CMB texture profile around hot spot

cmb\_profile = ring\_profile(cmb\_map, theta\_hot, phi\_hot, nside, npix)

# MBT curvature shell rooted at hot spot

vec\_hot = hp.ang2vec(theta\_hot, phi\_hot)

mbt\_field\_hot = np.zeros(npix)

mbt\_field\_hot[hp.query\_disc(nside, vec\_hot, np.radians(10.0))] = 1.0

mbt\_profile\_hot = ring\_profile(mbt\_field\_hot, theta\_hot, phi\_hot, nside, npix)

# Reference correlation

ref\_corr, \_ = pearsonr(cmb\_profile, mbt\_profile\_hot)

# Monte Carlo comparison

correlations = []

for \_ in range(num\_trials):

    rand\_theta = np.radians(np.random.uniform(0, 180))

    rand\_phi = np.radians(np.random.uniform(0, 360))

    vec\_rand = hp.ang2vec(rand\_theta, rand\_phi)

    mbt\_field\_rand = np.zeros(npix)

    mbt\_field\_rand[hp.query\_disc(nside, vec\_rand, np.radians(10.0))] = 1.0

    mbt\_profile\_rand = ring\_profile(mbt\_field\_rand, theta\_hot, phi\_hot, nside, npix)

    r, \_ = pearsonr(cmb\_profile, mbt\_profile\_rand)

    correlations.append(r)

# Histogram

plt.figure(figsize=(8, 5))

plt.hist(correlations, bins=50, color='salmon', edgecolor='black')

plt.axvline(ref\_corr, color='blue', linestyle='--', label=f'MBT Alignment (r = {ref\_corr:.2f})')

plt.xlabel("Pearson Correlation")

plt.ylabel("Number of Random MBT Fields")

plt.title("MBT–Hot Spot Texture Correlation")

plt.legend()

plt.grid(True, alpha=0.3)

plt.tight\_layout()

plt.show()

# Significance output

extreme\_hits = np.sum(np.array(correlations) >= ref\_corr)

p\_value = extreme\_hits / num\_trials

print(f"MBT–Hot Spot correlation: {ref\_corr:.4f}")

print(f"Fraction of rotations with r ≥ {ref\_corr:.2f}: {p\_value:.4f} ({extreme\_hits} out of {num\_trials})")