

pyTopoComplexity: A Python package for topographic complexity analysis

18 August 2024

Summary

pyTopoComplexity is a Python package that provides a computationally efficient and customizable implementation of three methods for quantifying topographic complexity. These methods include two-dimensional continuous wavelet transform (2D-CWT) analysis, fractal dimension estimation, and rugosity index calculation across various spatial scales. This package addresses the scarcity of open-source software for these sophisticated methods, which are crucial in modern terrain analysis, and facilitates data comparison and reproducibility. In the [software repository](#), we also include a Jupyter Notebook that integrates components from the Python-based surface-process modeling platform [Landlab](#) (Hobley et al. 2017). This allows researchers to simulate the smoothing of topography over time through terrestrial nonlinear hillslope diffusion processes. By combining these features, **pyTopoComplexity** advances the toolset available to researchers for measuring and simulating the time-dependent persistence of topographic complexity signatures against environmental forces on terrain surfaces.

Statement of need

Topographic complexity, often referred to as topographic roughness or surface roughness, provides critical insights into surface processes and the interactions among the geosphere, biosphere, and hydrosphere. With the increasing availability, utility, and popularity of digital terrain model (DTM) data, quantifying topographic complexity has become an essential measure in terrain analysis across various research fields. This necessity spans applications such as terrain classification and mapping at various spatial scales (Weiss 2001; Robbins 2018; John B. Lindsay, Newman, and Francioni 2019; Pardo-Igúzquiza and Dowd 2022a), evaluating the depositional age of event sedimentation and subsequent erosion processes (Hetz et al. 2016; Johnston et al. 2018; Booth et al. 2017; LaHusen et al. 2020; Herzog et al. 2023), and identifying habitats to assess ecological diversity on land and seafloor (Frost et al. 2005; Hetz et al. 2016; Wilson et al. 2007).

In recent years, several advanced methods for quantifying topographic complexity have been developed, including two-dimensional continuous wavelet transform (2D-CWT) analysis (Booth, Roering, and Perron 2009; Berti, Corsini, and Daehne 2013), fractal dimension estimation (Taud and Parrot 2005; Robbins 2018; Pardo-Igúzquiza and Dowd 2020), and rugosity index calculation (Jenness 2004; Du Preez 2015). These methods are considered more effective for terrain analysis tasks compared to conventional morphological metrics such as variations in local slope and relief. Despite their importance, comprehensive publicly available tools that incorporate these advanced methods for studying topographic complexity are lacking. Common open-source geospatial analysis software, such as QGIS (QGIS Development Team 2023), GRASS GIS (GRASS Development Team 2023), and WhiteboxTools (J. B. Lindsay 2016), only implement basic, conventional methods, limiting the reproducibility and comparability of these newer approaches. Although some specialized programs for calculating the rugosity index exist (Walbridge et al. 2018; Benham 2022), they have been limited to marine bathymetric studies and involve various mathematical choices and designs.

To address this gap, we have developed an open-source Python toolkit called **pyTopoComplexity**, which provides computationally efficient and easily customizable implementations of three modules for performing and visualizing the results of 2D-CWT, fractal dimension, and rugosity calculations (Table 1). This toolkit can detect the grid spacing and unit of the projected coordinate system (acceptable in meters, U.S. survey feet, and international feet) from the input raster DTM file (GeoTIFF format) and automatically conduct unit conversions in necessary calculation steps to ensure data consistency and reproducibility. Results at nodes affected by edge effects due to no-data values outside the input raster will be removed by default. Users can define the suitable spatial scale to match their research purposes and choose computational approaches (e.g., chunk processing, faster mathematical approximations) to optimize performance (see details in the [Methods and Features overview](#) section).

Table 1 : Modules contained in the **pyTopoComplexity** package.

Modules	Classes	Method Descriptions	References
pycwtmexhat.py	CWTMexHat	Quantifies the wavelet-based curvatures of the land surface using two-dimensional continuous wavelet transform (2D-CWT) with a Mexican Hat wavelet	Booth, Roering, and Perron (2009); Booth et al. (2017)
pyfracd.py	FracD	Conducts fractal-dimension analysis on the land surface using variogram procedures	Wen and Sinding-Larsen (1997); Pardo-Igúzquiza and Dowd (2020)
pyrugosity.py	RugosityIndex	Calculates the rugosity index of the land surface	Jenness (2004); Du Preez (2015)

Each module of the **pyTopoComplexity** is provided with a corresponding [example Jupyter Notebook](#) for usage instructions, using the Light Detection and Ranging (LiDAR) DTM data of a deep-seated landslide that occurred in 2014 in the Oso area of the North Fork Stillaguamish River Valley, Washington State, USA (Washington Geological Survey 2023). In the software repository, we also include an additional Jupyter Notebook file [nonlineardiff_Landlab.ipynb](#), which allows researchers to simulate the smoothing of topography over time via terrestrial nonlinear hillslope diffusion processes (Roering, Kirchner, and Dietrich 1999). This is achieved by employing the [TaylorNonLinearDiffuser](#) module from the [terrainBento](#) Python package (Barnhart et al. 2019) and running the simulation in the [Landlab](#) environment (Hobley et al. 2017).

By bridging the gap between different advanced terrain analytical approaches and incorporating functionality for landscape evolution modeling, **pyTopoComplexity** serves as a valuable resource for topographic complexity research and has the potential to foster new insights and interdisciplinary collaborations in the fields of geology, geomorphology, geography, ecology, and oceanography.

Methods and features overview

Two-dimensional continuous wavelet transform analysis

The [pycwtmexhat.py](#) module in **pyTopoComplexity** implements the 2D-CWT method for terrain analysis, providing detailed information on how amplitude is distributed across spatial frequencies at each position in the data by transforming spatial data into position-frequency space. This method is particularly effective for depicting the Laplacian of topography (Torrence and Compo 1998; Lashermes, Foufoula-Georgiou, and Dietrich 2007), revealing concave and convex regions of topography at various scales (Malamud and Turcotte 2001; Strubel et al. 2021), identifying deep-seated landslides (Booth, Roering, and Perron 2009; Berti, Corsini, and Daehne 2013), and estimating the depositional ages of landslide deposits (Booth et al. 2017; LaHusen et al. 2020; Underwood 2022; Herzog et al. 2023).

The 2D-CWT is computed by convolving the elevation data z with a wavelet family ψ , using a wavelet scale parameter s at every location (x, y) :

$$C(s, x, y) = \Delta^3 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} z(x, y)\psi(x, y) dx dy$$

, where the resultant wavelet coefficient $C(s, x, y)$ provides a measure of how well the wavelet ψ matches the data z at each grid (Torrence and Compo 1998). When s is large, ψ is spread out, capturing long-wavelength features of z ; when s is small, ψ becomes more localized, making it sensitive to fine-scale features of z . In this implementation, we use the 2D Mexican Hat wavelet (i.e., Ricker wavelet) function to define ψ :

$$\psi = -\frac{1}{\pi(s\Delta)^4}(1 - \frac{x^2 + y^2}{2s^2}) e^{-\frac{x^2+y^2}{2s^2}} \quad \lambda = \frac{2\pi s}{\sqrt{5/2}}$$

The Mexican hat wavelet is proportional to the second derivative of a Gaussian envelope, with its Fourier wavelength (λ) dependent on the grid spacing (Δ) of the input DTM raster. The wavelet function ψ is scaled according to the wavelet scale parameter s and the grid spacing Δ , ensuring that the resultant wavelet coefficient C signifies concave and convex landforms corresponding to the wavelet scale s . Users can define the value of λ in meters as the targeted spatial scale for landform roughness analysis, and the [pycwtmexhat.py](#) module will automatically compute the wavelet scale s based on the grid spacing (Δ) of the input raster file (Figure 1).

We note that the C and ψ equations presented here are the mathematical approaches adapted in later publications (LaHusen et al. 2020; Underwood 2022; Herzog et al. 2023) and ongoing works (Booth and Petruson 2023; Lai et al. 2023; Ozikoko, Booth, and Duvall 2023) of landslide mapping and age dating studies, with minor differences from the earlier similar research in Booth, Roering, and Perron (2009) and Booth et al. (2017) (original MATLAB codes available from [Booth's personal website](#)). These differences in mathematical approach can result C values in different units and order of magnitude (e.g., 10^{-3} to $10^{-4} [m^{-2}]$ in Booth et al. (2017) and prior studies; 10^{-2} to $10^{-3} [m^{-1}]$ in LaHusen et al. (2020) thereafter). Despite this discrepancy, the C values yielded from these two approaches are linearly scaled and interconvertible, and they both reflect identical spatiotemporal patterns of topographic complexity (i.e., surface roughness).

Figure 1. Hillshade map of the 2014 Oso Landslide region (upper-left), and the results of the two-dimensional continuous wavelet transform (2D-CWT) analysis using the [pycwtmexhat.py](#) module, with designated Fourier wavelengths (λ) of the Mexican Hat wavelet at 15 m, 30 m, 45 m, 60 m, and 75 m.

Fractal dimension analysis

The [pyfracd.py](#) module in **pyTopoComplexity** calculates the fractal dimension, which measures the fractal characteristics of natural features (Mandelbrot and Wheeler 1983). This method provides insights into the self-similarity of landscapes, helping quantify their irregularity and fragmentation, which is crucial for studying the surface processes that shape Earth and planetary surfaces (Xu, Moore, and Gallant 1993).

In this module, we adapt the variogram method to estimate the local fractal dimension within a moving window centered at each cell of the DTM (Taud and Parrot 2005; Pardo-Igúzquiza and Dowd 2020, 2022a). This approach simplifies the problem to estimating the fractal dimension of one-dimensional topographic profiles (Dubuc et al. 1989) within the two-dimensional moving window. For a one-dimensional profile of length L , the variogram $\gamma_1(k)$ can be estimated at the K lag distances (K lag distances ($k = 1, \dots, K$)) by:

$$\gamma_1(k) = \frac{1}{2(L-k)} \sum_{i=1}^{L-k} [Z(i) - Z(i+k)]^2$$

, where $Z(i)$ is the elevation at location i along the profile. The local fractal dimension (FD) is estimated from one-dimensional profiles in principal directions (i.e., horizontal, vertical, and diagonal) within a square moving window. Assuming that fractional Brownian motion is an appropriate stochastic model for natural surfaces, its variogram follows a power-law model with respect to k (Wen and Sinding-Larsen 1997):

$$\gamma_1(k) = \alpha k^\beta, \quad \alpha \geq 0; \quad 0 \leq \beta < 2$$

, and its exponent β is related to the local fractal dimension (FD) by:

$$FD = T D + 1 - \frac{\beta}{2}$$

, where $T D$ is the topological dimension in the Euclidean space of the fractional Brownian motion. For one-dimensional fractional Brownian motion, $T D = 1$. The fractal dimension of the two-dimensional surface (FD) can be estimated as the average fractal dimension of the one-dimensional profiles (FD):

$$(FD)_2 = 1 + (FD)_1$$

Users can specify the size (number of grids along each edge) of the moving window to study fractal characteristics at desired spatial scales. In addition to calculating the fractal dimension, the [pyfracd.py](#) module also computes reliability parameters such as standard error and the coefficient of determination (R^2) to assess the robustness of the analysis (Figure 2).

This module adapts the Triangulated Irregular Networks (TIN) method from Jenness (2004) to approximate the contoured area as the sum of eight truncated-triangle areas. These triangles connect the central grids, four corner grids, and four grids at the middle points of the surrounding edges within the moving window. If no local slope correction is applied, the planimetric area is considered to be the horizontal planar area of the moving window, as described in Jenness (2004). Another approach considers slope correction where the planimetric area is projected onto a plane of the local gradient (Du Preez 2015).

By definition, the rugosity index is as a minimum value of one (completely flat surface). Typical values range from the conventional rugosity index (without slope correction) from one to three although larger values are possible in very steep terrains. The slope-corrected rugosity index, also called arc-chord ratio rugosity index, could provide a better representation of local surface complexity (Figure 3).

The [pyrugosity.py](#) module in **pyTopoComplexity** implements the Arc-Chord Ratio (ACR) rugosity index (Booth, Roering, and Perron 2009; Berti, Corsini, and Daehne 2013) and the conventional rugosity index (Mandelbrot and Wheeler 1983). This method provides insights into the self-similarity of landscapes, helping quantify their irregularity and fragmentation, which is crucial for studying the surface processes that shape Earth and planetary surfaces (Xu, Moore, and Gallant 1993).

In this module, we adapt the variogram method to estimate the local fractal dimension within a moving window centered at each cell of the DTM (Taud and Parrot 2005; Pardo-Igúzquiza and Dowd 2020, 2022a). This approach simplifies the problem to estimating the fractal dimension of one-dimensional topographic profiles (Dubuc et al. 1989) within the two-dimensional moving window. For a one-dimensional profile of length L , the variogram $\gamma_1(k)$ can be estimated at the K lag distances (K lag distances ($k = 1, \dots, K$)) by:

$$\gamma_1(k) = \frac{1}{2(L-k)} \sum_{i=1}^{L-k} [Z(i) - Z(i+k)]^2$$

, where $Z(i)$ is the elevation at location i along the profile. The local fractal dimension (FD) is estimated from one-dimensional profiles in principal directions (i.e., horizontal, vertical, and diagonal) within a square moving window. Assuming that fractional Brownian motion is an appropriate stochastic model for natural surfaces, its variogram follows a power-law model with respect to k (Wen and Sinding-Larsen 1997):

$$\gamma_1(k) = \alpha k^\beta, \quad \alpha \geq 0; \quad 0 \leq \beta < 2$$

, and its exponent β is related to the local fractal dimension (FD) by:

$$FD = T D + 1 - \frac{\beta}{2}$$

, where $T D$ is the topological dimension in the Euclidean space of the fractional Brownian motion. For one-dimensional fractional Brownian motion, $T D = 1$. The fractal dimension of the two-dimensional surface (FD) can be estimated as the average fractal dimension of the one-dimensional profiles (FD):

$$(FD)_2 = 1 + (FD)_1$$

Users can specify the size (number of grids along each edge) of the moving window to study fractal characteristics at desired spatial scales. In addition to calculating the fractal dimension, the [pyfracd.py](#) module also computes reliability parameters such as standard error and the coefficient of determination (R^2) to assess the robustness of the analysis (Figure 2).

This module adapts the Triangulated Irregular Networks (TIN) method from Jenness (2004) to approximate the contoured area as the sum of eight truncated-triangle areas. These triangles connect the central grids, four corner grids, and four grids at the middle points of the surrounding edges within the moving window. If no local slope correction is applied, the planimetric area is considered to be the horizontal planar area of the moving window, as described in Jenness (2004). Another approach considers slope correction where the planimetric area is projected onto a plane of the local gradient (Du Preez 2015).

By definition, the rugosity index is as a minimum value of one (completely flat surface). Typical values range from the conventional rugosity index (without slope correction) from one to three although larger values are possible in very steep terrains. The slope-corrected rugosity index, also called arc-chord ratio rugosity index, could provide a better representation of local surface complexity (Figure 3).

The [pyrugosity.py](#) module in **pyTopoComplexity** implements the Arc-Chord Ratio (ACR) rugosity index (Booth, Roering, and Perron 2009; Berti, Corsini, and Daehne 2013) and the conventional rugosity index (Mandelbrot and Wheeler 1983). This method provides insights into the self-similarity of landscapes, helping quantify their irregularity and fragmentation, which is crucial for studying the surface processes that shape Earth and planetary surfaces (Xu, Moore, and Gallant 1993).

In this module, we adapt the variogram method to estimate the local fractal dimension within a moving window centered at each cell of the DTM (Taud and Parrot 2005; Pardo-Igúzquiza and Dowd 2020, 2022a). This approach simplifies the problem to estimating the fractal dimension of one-dimensional topographic profiles (Dubuc et al. 1989) within the two-dimensional moving window. For a one-dimensional profile of length L , the variogram $\gamma_1(k)$ can be estimated at the K lag distances (K lag distances ($k = 1, \dots, K$)) by:

$$\gamma_1(k) = \frac{1}{2(L-k)} \sum_{i=1}^{L-k} [Z(i) - Z(i+k)]^2$$

, where $Z(i)$ is the elevation at location i along the profile. The local fractal dimension (FD) is estimated from one-dimensional profiles in principal directions (i.e., horizontal, vertical, and diagonal) within a square moving window. Assuming that fractional Brownian motion is an appropriate stochastic model for natural surfaces, its variogram follows a power-law model with respect to k (Wen and Sinding-Larsen 1997):

$$\gamma_1(k) = \alpha k^\beta, \quad \alpha \geq 0; \quad 0 \leq \beta < 2$$

, and its exponent β is related to the local fractal dimension (FD) by:

$$FD = T D + 1 - \frac{\beta}{2}$$

, where $T D$ is the topological dimension in the Euclidean space of the fractional Brownian motion. For one-dimensional fractional Brownian motion, $T D = 1$. The fractal dimension of the two-dimensional surface (FD) can be estimated as the average fractal dimension of the one-dimensional profiles (FD):

$$(FD)_2 = 1 + (FD)_1$$

Users can specify the size (number of grids along each edge) of the moving window to study fractal characteristics at desired spatial scales. In addition to calculating the fractal dimension, the [pyfracd.py](#) module also computes reliability parameters such as standard error and the coefficient of determination (R^2) to assess the robustness of the analysis (Figure 2).

This module adapts the Triangulated Irregular Networks (TIN) method from Jenness (2004) to approximate the contoured area as the sum of eight truncated-triangle areas. These triangles connect the central grids, four corner grids, and four grids at the middle points of the surrounding edges within the moving window. If no local slope correction is applied, the planimetric area is considered to be the horizontal planar area of the moving window, as described in Jenness (2004). Another approach considers slope correction where the planimetric area is projected onto a plane of the local gradient (Du Preez 2015).

By definition, the rugosity index is as a minimum value of one (completely flat surface). Typical values range from the conventional rugosity index (without slope correction) from one to three although larger values are possible in very steep terrains. The slope-corrected rugosity index, also called arc-chord ratio rugosity index, could provide a better representation of local surface complexity (Figure 3).