Authors’ responses and revisions to requested corrections by Associate Editor Simon Mudd for the manuscript titled “Short Communication: Multiscale topographic complexity analysis with pyTopoComplexity” by Lai et al.

Line 23: Consider explaining DTM is "bare Earth". Do you mean to specify this? That is, are you indirectly telling users not to use DSMs?

[Authors’ response]: The tool pyTopoComplexity can analyze the surface complexity of any type of DTM data. We leave it unspecified here because some available DTM data don’t necessarily represent “bare Earth.” Studying surface complexity with surface objects like trees or buildings might reveal other utilities we’re not aware of. However, we’ve clarified that we’ll use “bare Earth” DTM from the North Fork Stillaguamish River valley as the example data for our case study (line 50).

Line 35 ‘significances’: This sounds a bit weird. Consider rewording.

[Authors’ response]: We have clarified it as “geomorphic features.”

Line 99 ‘roughness’: In the response to reviewers you make an argument that you take "roughness" to have a specific meaning. You haven't defined that yet. I would specify what you mean here.

[Authors’ response]: We’ve clarified it as “land surface complexity” to align with other sections of the manuscript.

Line 109 ‘Booth et al. (2009)’: typo

[Authors’ response]: The typo of citation is fixed.

Line 214 ‘typically assumed to have a ration m/n ~ 0.5’: Most of the literature on this says the value falls between 0.4-0.6 (Kirby & Whipple, 2012; Tucker & Whipple, 2002; Whipple, 2004). I would say that.

[Authors’ response]: We have corrected the m/n ratio range and provided the suggested references.

Line 231 figure 2: This is a bit of a weird axis label. If you are going to do this I would say "cells" or "pixels" instead of "grids". It would be a lot better to plot this in metres!

You don't have to do that (the code is a bit tedious) but you are welcome to copy the code from https://github.com/LSDtopotools/lsdviztools/blob/master/lsdviztools/lsdmapfigure/plottinghelpers.py under the GetTicksForUTMNoInversion for the axis styling

[Authors' response]: We've changed the axis label to "grid cells" based on this suggestion and ensured consistency with our notion table appendix B.

In the current version, we plotted the x and y axes in grid cells for an accurate representation of the input data. DTM data with a reprojected coordinate system may not have the same grid spacing along the x and y axes. This inaccuracy could be more pronounced when plotting DTM with a large area. Since the grid spacing is used in some calculations of the pyTopoComplexity modules, it may not be straightforward to directly implement the codes from *LSDtopotools*. However, we agree that it would be great to allow users to choose the axis label. Adding this function is on our list for the next update of pyTopoComplexity.

Line 261 'dated to ~11693 years before present': reference needed

[Authors' response]: Reference Booth et al. (2017) has been added.

Line 308-309: refer to figure

[Authors' response]: We have referred the text here to Fig. 3c and **S2.**

Line 398 figure 4 & line 426 figure 5: Needs units. You can put (in m^/yr^-1) at the top of this legend. Same for erodibility. I know it is in the caption but it is easier for readers if it is in the legend.

Can I suggest for this figure 4 and figure 5 to put the hillshades of end member simulations either as a sub-panel or a separate figure? I think readers will want to see how different the landscapes look (not just stylistically) when these parameters change.

[Authors' response]: We've implemented the suggestion to include units in the legend and a sample hillshade panel for figures 4 and 5.

I have been in contact with the lead author about some minor software changes that have already been implemented, which means I have been able to confirm the code runs smoothly and right out of the box with a !pip install command on google colab. I have a few further suggestions:

In setup.py make sure to bump the version number

In /pytopocomplexity/__init__.py add the line __version__ = '1.1.1' (or whatever version it is) so pytopocomplexity.__version__ works

In the notebooks I suggest adjusting the code so if you are working in, say, colab, the notebook will look for the data and download it if it is not there:

```
import os

import requests

# Define the directory and file paths

base_dir = os.path.join(os.getcwd(), 'ExampleDEM') # input file base directory

input_file = 'Ososlid2014_f_3ftgrid.tif' # input file name

input_dir = os.path.join(base_dir, input_file) # input file directory

# Define the download URL

download_url = "https://github.com/GeoLarryLai/pyTopoComplexity/raw/main/ExampleDEM/Ososlid2014_f_3ftgrid.tif"

# Check if directory exists, if not, create it

if not os.path.exists(base_dir):

os.makedirs(base_dir)

# Check if file exists, if not, download it

if not os.path.exists(input_dir):

response = requests.get(download_url)

with open(input_dir, 'wb') as f:
```

```
f.write(response.content)

print(f"File downloaded to {input_dir}")

else:

print("Good news, I've already got the input file")
```

Finally, the authors might consider adding a colab link to each of the notebooks (apart from the landlab one, I haven't tested but I suspect installation would fail).

https://colab.research.google.com/github/googlecolab/colabtools/blob/master/notebooks/colab-github-demo.ipynb

[Authors' response]: We've been in touch with Associate Editor Simon Mudd. In the latest version, v.1.1.1 of pyTopoComplexity, we've incorporated all of Editor Mudd's suggested changes and enhancements into the codes. All notebooks can now be executed on Google Colab.