User guide to chi analysis

Simon M. Mudd, School of GeoSciences, University of Edinburgh

Table of Contents

Preface by Simon Mudd 1
1. Introduction 2
    1.1. Background 2
2. Getting the software 3
    2.1. Starting up on your own Linux machine 3
    2.2. Starting up on a non-Linux operating system (Windows, MacOS) 3
    2.3. Troubleshooting a vagrant server 4
    2.4. Installing the code and setting up directories 5
3. Preparing your data. 7
4. Running chi analysis 8
5. Visualisation 9

Preface by Simon Mudd

This is the documentation for the chi analysis tool. It is the second generation of the software reported in Mudd et al. 2014. Right now there is not very much documentation but that will change in the near future.

Chapter 1. Introduction

This section explains the theory behind analysis of channels using both slope-area analysis and chi analysis. If you just want to run the software skip ahead to the section Getting the software.

1.1. Background

In the late 1800s, G.K. Gilbert proposed that bedrock channel incision should be proportional to topographic gradients and the amount of water flowing in a channel.

> We have already seen that erosion is favored by declivity. Where the declivity is great the agents of erosion are powerful; where it is small they are weak; where there is no declivity they are powerless. Moreover it has been shown that their power increases with the declivity in more than simple ratio.
>
> — G.K. Gilbert, Geology of the Henry Mountains 1877

Since then, many geomorpholgists have attempted to extract information about erosion rates from channel profiles. Chi analysis is a method of extracting information from channel profiles that attempts to compare channels with different discharges first proposed by Leigh Royden and colleagues at MIT. LSDTopoTools has a number of tools for performing chi analysis.

This document gives instructions on how to use the segment fitting tool for channel profile analysis developed by the Land Surface Dynamics group at the University of Edinburgh. The tool is used to examine the geometry of channels using the integral method of channel profile analysis. For background to the method, and a description of the algorithms, we refer the reader to Mudd et al. (2014). For background into the strengths of the integral method of channel profile analysis, the user should read Perron and Royden (2013, ESPL).

Chapter 2. Getting the software

We have attempted, to the best of our ability, to streamline the process of installing our software on any platform (Windows, Linux, MacOS). You will need to install a bit of software. If you have a decent internect connections this should take under an hour. The main rate limiting step is downloading files: You will need to download ~2Gb of software. This requirement is substantially reduced if you want to install on a a native Linux machine.

2.1. Starting up on your own Linux machine

If you work on a Linux operating system we have a python script for installing everything. You will need to be connected to the internet, and have python installed. You can skip ahead to the section Installing the code and setting up directories.

2.2. Starting up on a non-Linux operating system (Windows, MacOS)

Below are quick instructions. If you have trouble, follow the link above and then go through the complete instructions.

Quick Instructions for using Vagrant for LSDTopoTools

1. Download and install virtualbox.
2. Download and install vagrant. You might have to restart your computer after this.
3. If you are on Windows, download putty.exe. If you are on Linux or MacOS you can skip this (they have built-in equivalents).
4. Make a directory for your vagrant box. We tend to put ours in a directory called `VagrantBoxes`.
5. Inside that directory make two new directories: `LSDTopoTools` and `Ubuntu`. The second directory's name doesn't matter, it is just for holding a vagrant file (see below). However you **MUST** have the `LSDTopoTools` directory. The directory name is **case sensitive**!
6. Download one of our vagrantfiles: https://github.com/LSDtopotools/LSDTT_vagrantfiles into the `Ubuntu` directory (again, the name of the directory doesn't matter).
7. Rename the vagrantfile from the repo (either `Vagrantfile_32bit_FFTW` or `Vagrantfile_64bit_FFTW`) simply `vagrantfile`. Your operating system is almost certainly 64 bit, but on most computers you need to select 32 bit because the default setting is to disable 64 bit guest operating systems. This can be changed but only by expert users.
8. Open a terminal or powershell window and navigate to the directory with the vagrantfile.
9. Run `vagrant up` from the command line.

   WARNING If you are running `vagrant up` for the first time it can take some time to download the base box. They are several hundred Mb each!

10. Run `vagrant provision` after the box has started.
11. If on Windows, you should now be able to use `putty.exe` to ssh into your LSDTopoTools server. The host name is almost always `127.0.0.1` and the port is almost always `2222`.
12. On Windows, you will need to give a username and password after connecting using `putty.exe`. The machine is running locally on your computer so nothing is being sent over the internet. The username is always `vagrant` and the password is also `vagrant`.
13. If you are on MacOS or Linux you do not need `putty.exe`; all you need to do is type `vagrant ssh` into the command line. See the vagrant instructions.

2.3. Troubleshooting a vagrant server

There are a few common problems people have when running a vagrant server.

- If you are on an old computer, sometimes vagrant times out before the virtual machine boots. This most frequently happens the first time you boot a vagrant machine. The most effective way to fix this is with the canonical IT solution: turning it off and on again. To do that run `vagrant halt` and `vagrant up` in succession.
- If vagrant hangs up in the powershell or terminal window and does not give you back the command prompt, turn it off and on again by typing ctrl-c and then running the vagrant command again.
- If your files are not synching across your host and vagrant machine, it is probably because there is some misspelling in your `LSDTopoTools` folder on the host machine. Make sure that folder is in the correct place and is spelled correctly (remember it should be case sensitive!!).

2.4. Installing the code and setting up directories

Easy setup quick start

If you just want to get started with the standard setup without any details, do the following:

1. Go into a Linux terminal. If you are on your own computer and it is not Linux, you will need to start a Vagrant session. You will need python but this is now standard on most Linux distributions so we will assume you have it.
2. If you used our vagrantfile the setup tool will already be on your vagrant box. It is sitting in the LSDTopoTools directory. You can go there with `cd /LSDTopoTools`.
3. If you are on your own linux machine you can get the setuptool with:

```
$ wget
https://raw.githubusercontent.com/LSDtopotools/LSDAutomation/master/LSDTopoToolsS
etup.py
```

4. Run the setup tool:

```
$ python LSDTopoToolsSetup.py -id 0 -MChi True
```

5. You are ready to go!!

NOTE **For native Linux users**: The `-id` flag tells LSDTopoToolsSetup.py where to put the files. In vagrant it always starts in the root directory (this is for file syncing purposes). However, if your native operating system is Linux, then you can either install in your home directory (`-id 0`) or in the directory where you called LSDTopoToolsSetup.py (`-id 1`).

1. The first thing you need to do is get our python program **LSDTopoToolsSetUp.py**. It lives here: https://github.com/LSDtopotools/LSDAutomation/blob/master/LSDTopoToolsSetup.py

   You will need to run this file in your Linux environment, so in a terminal window type:

   ```
   $ wget
   https://raw.githubusercontent.com/LSDtopotools/LSDAutomation/master/LSDTopoToolsSetup.p
   y
   ```

NOTE	If `wget` doesn't work, you can follow the link: https://raw.githubusercontent.com/LSDtopotools/LSDAutomation/master/LSDTopoToolsSetup.py Copy the text, paste it into a text editor and save it as `LSDTopoToolsSetup.py`.

2. Now, in your terminal window run this script. It has some options:

   a. For the most basic setup, type:

   ```
   $ python LSDTopoToolsSetup.py -id 0  -MChi True
   ```

   In our Vagrant setup, this will install everything in the root directory (you can got there with `$ cd /`), which is the default setup generated by our Vagrantfile. If you are not in vagrant the **LSDTopoTools** directories will be in your home directory (you can get there with `$ cd ~`).

   b. If you don't want **LSDTopoTools** in your home directory, you can install it in your current directory with:

   ```
   $ python LSDTopoToolsSetup.py -id 1 -MChi True
   ```

3. It turns out the `LSDTopoToolsSetup.py` tool has a number of options but we will explain these later. If you want a preview of what it does, you can call its help options:

   ```
   $ python LSDTopoToolsSetup.py -h
   ```

Chapter 3. Preparing your data.

Use GDAL. And also DEM_preprocessing. They are both great.

Chapter 4. Running chi analysis

All the magic happens here.

Chapter 5. Visualisation

We use python, since it is much better than the alternatives. Don’t ever use Matlab.