Analysis Modules

Learn yt workshop 2016

Nathan Goldbaum

(some slides from Britton Smith's
2012 clump finding tutorial)

Outline

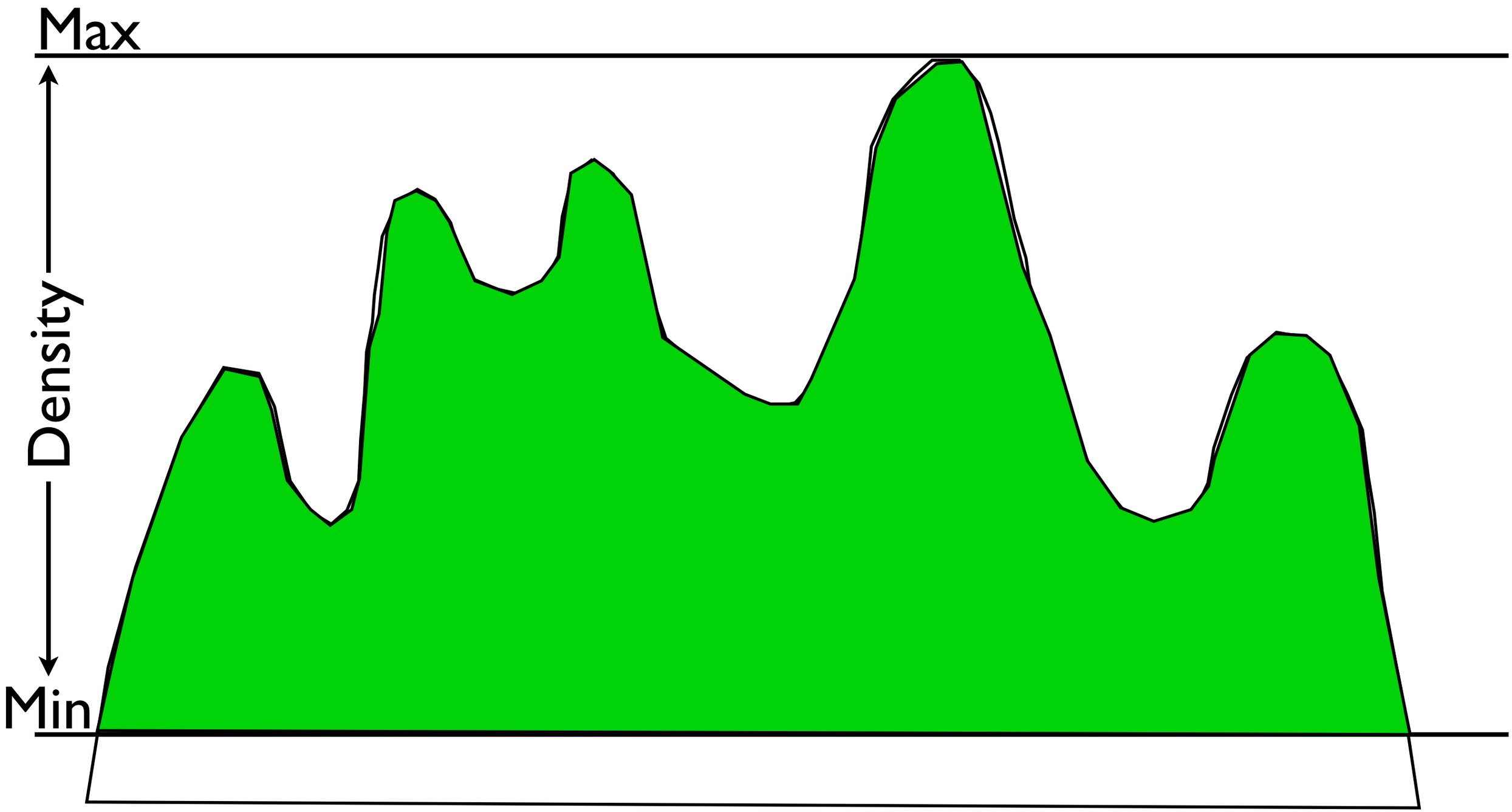
- Clump finding
- Halo finding
  - rockstar
- Synthetic Observations
  - Trident, pyXSIM

Clump Finding

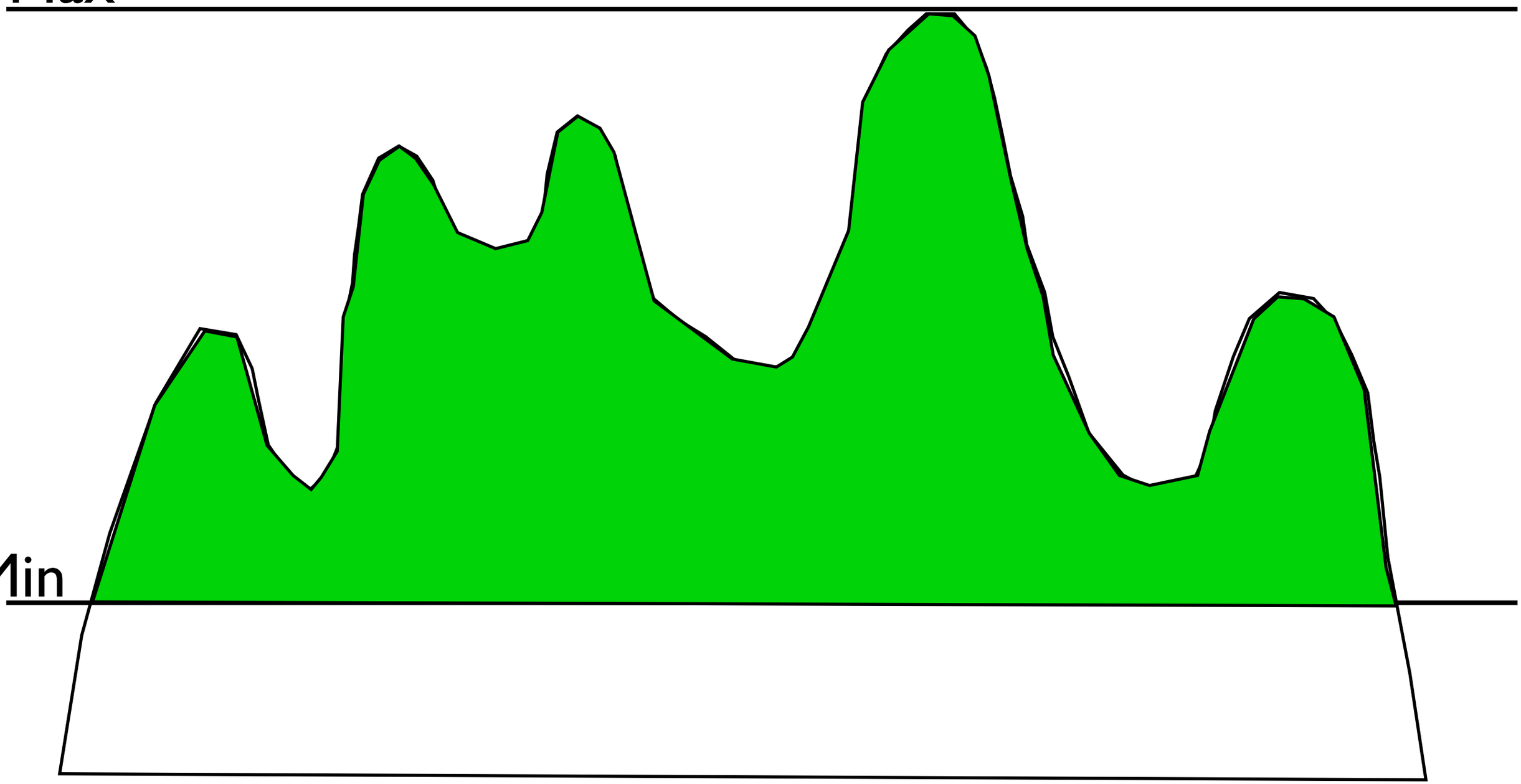
“clump”:

The largest disconnected isocontour satisfying some criteria, such as being gravitationally bound.

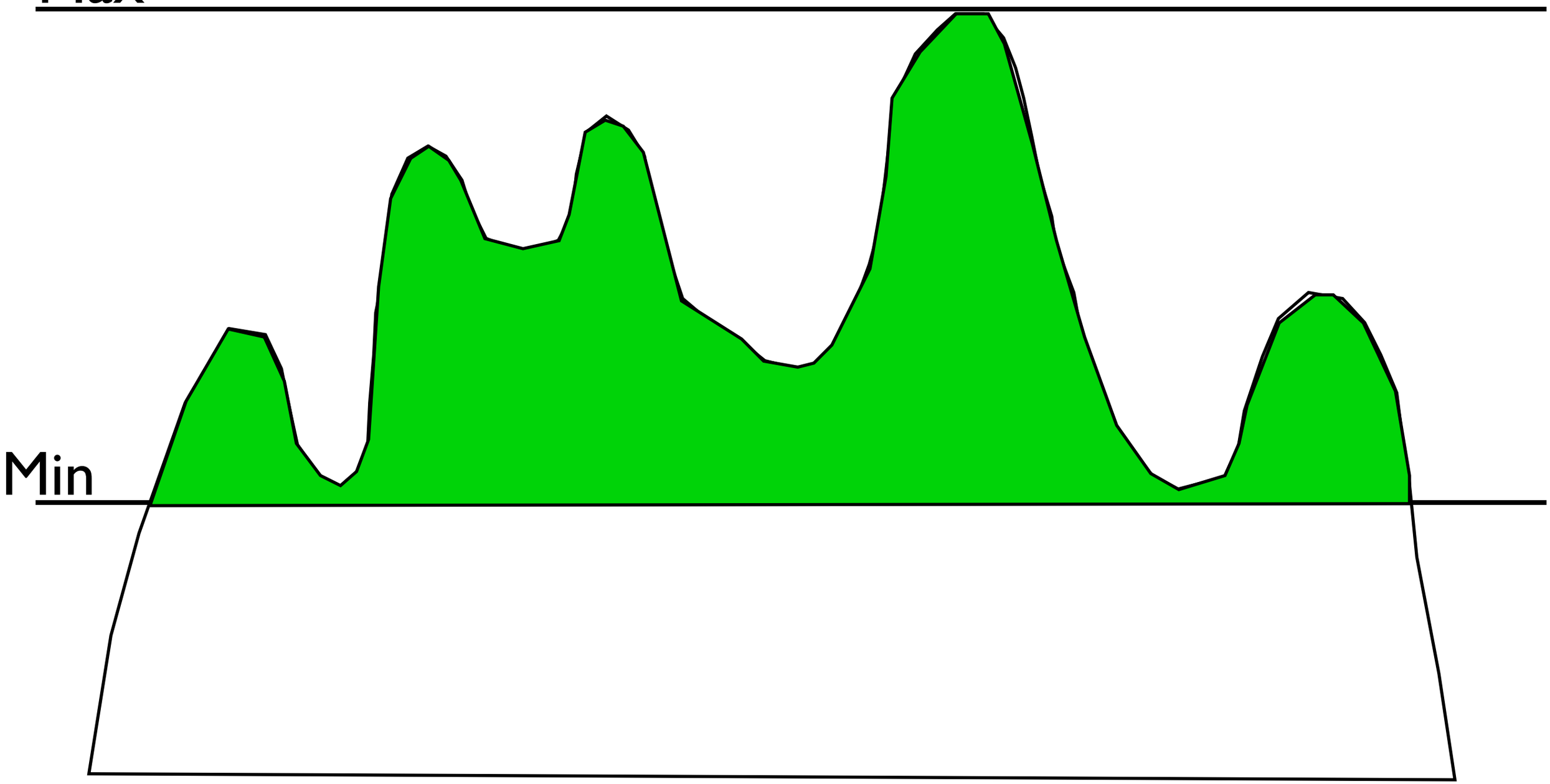
Finding Clumps



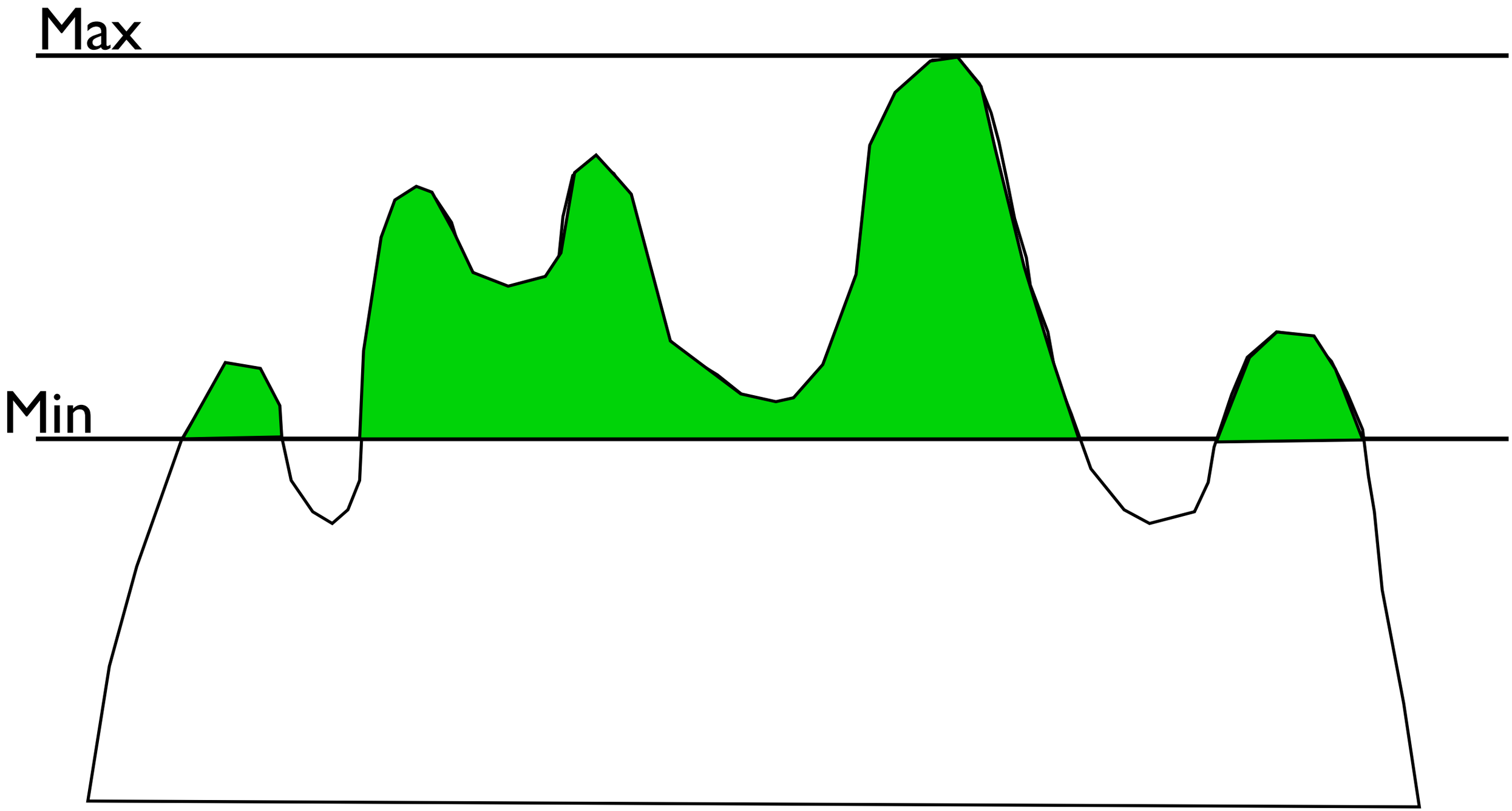
Finding Clumps



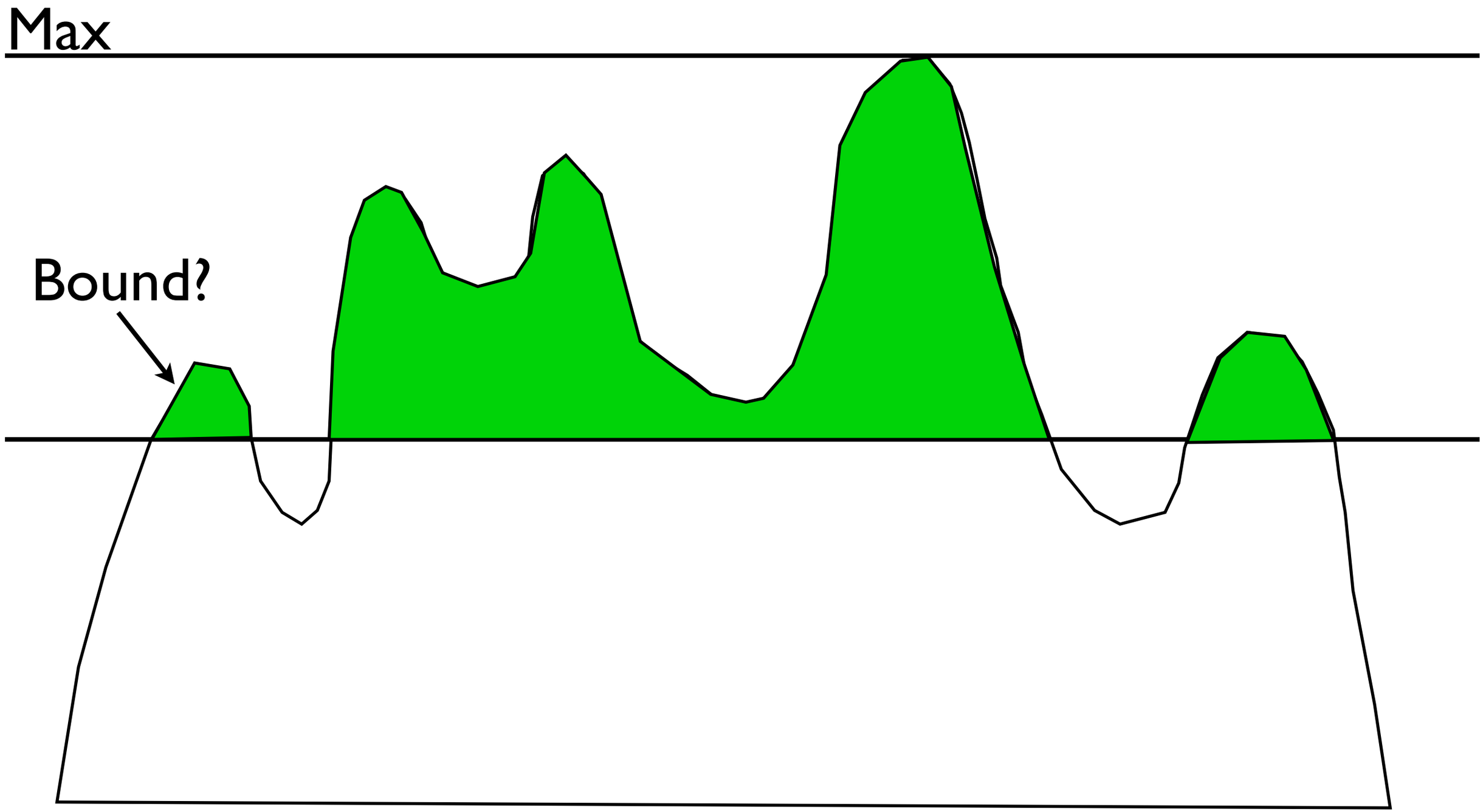
Finding Clumps



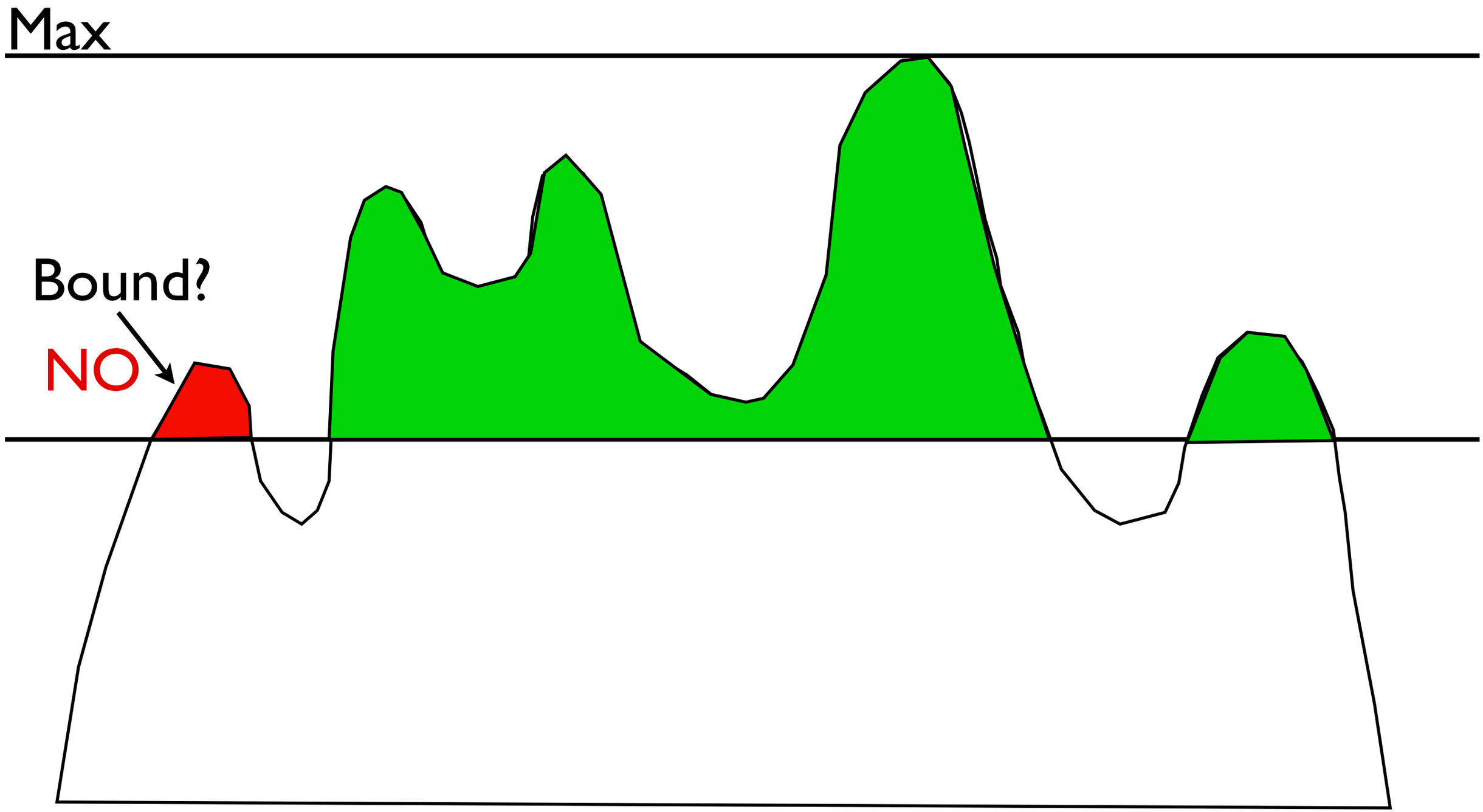
Finding Clumps



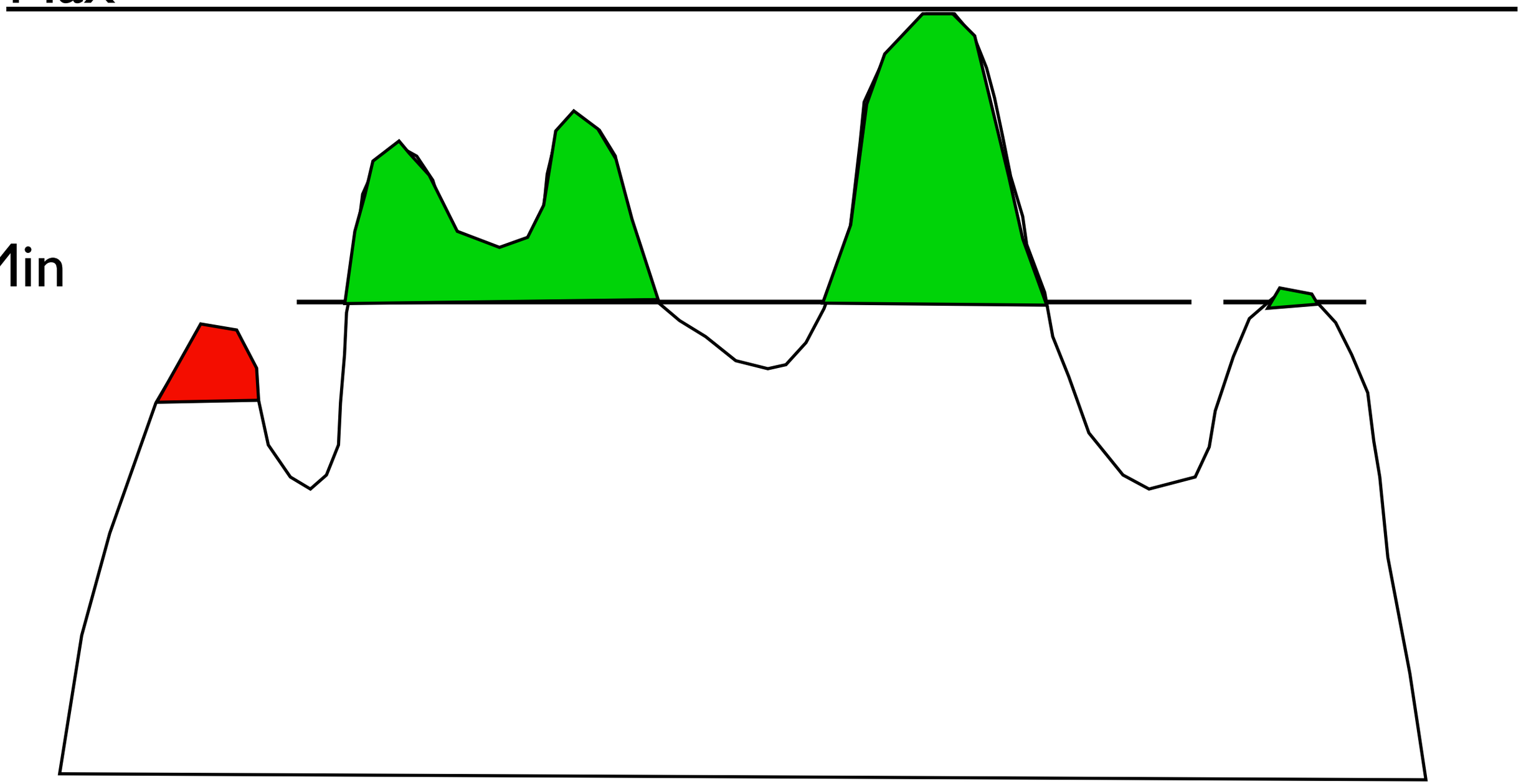
Finding Clumps



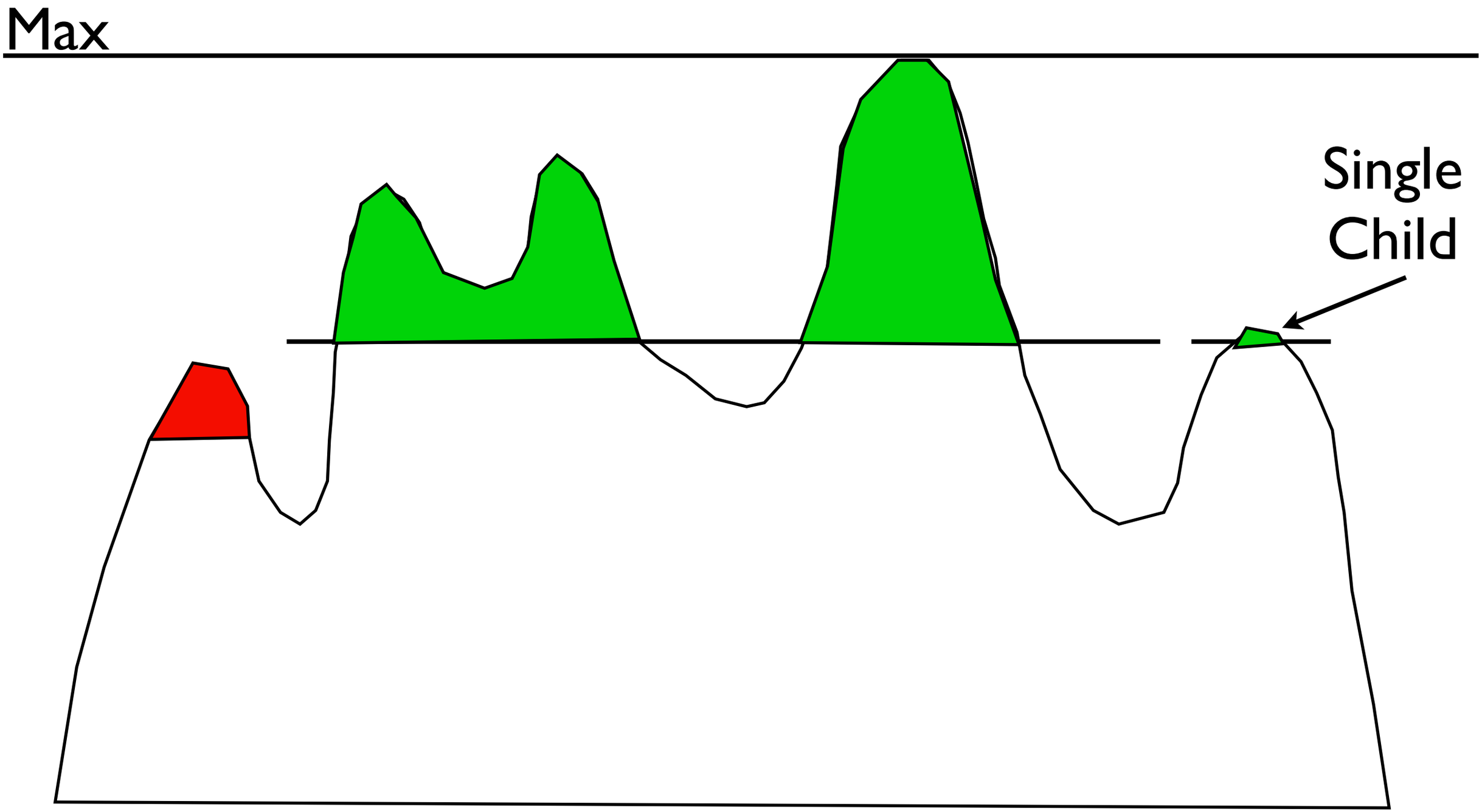
Finding Clumps



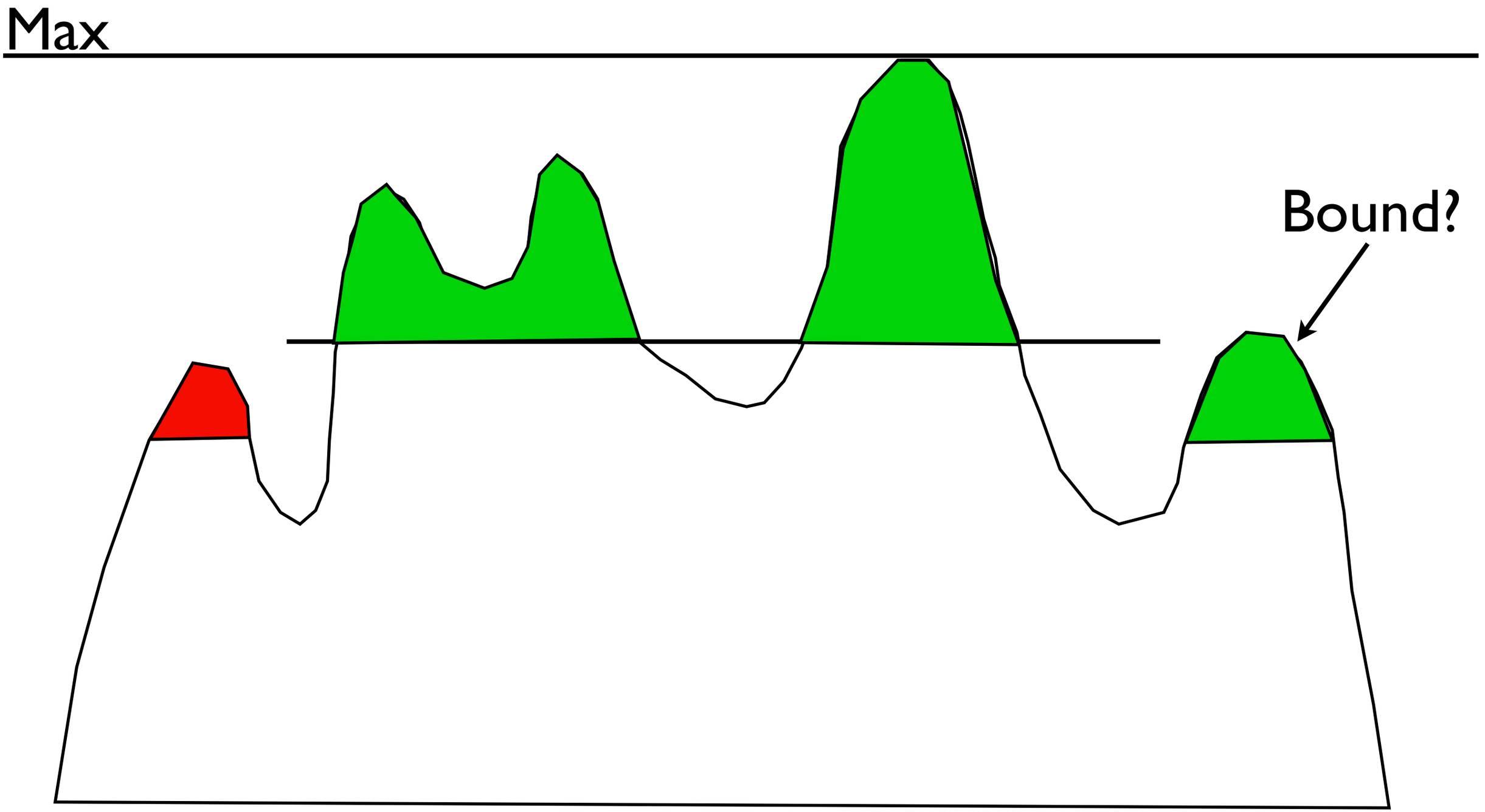
Finding Clumps



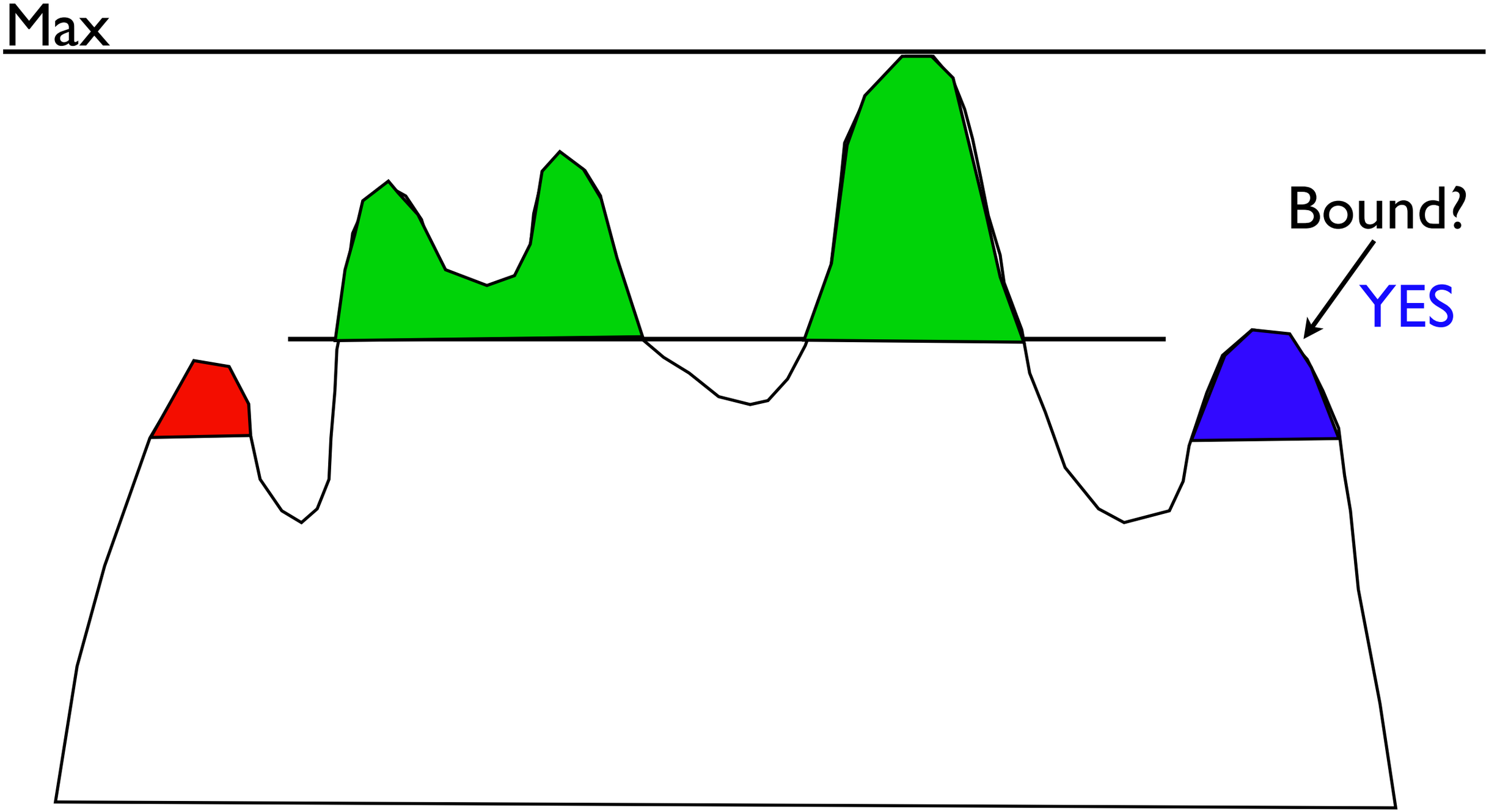
Finding Clumps



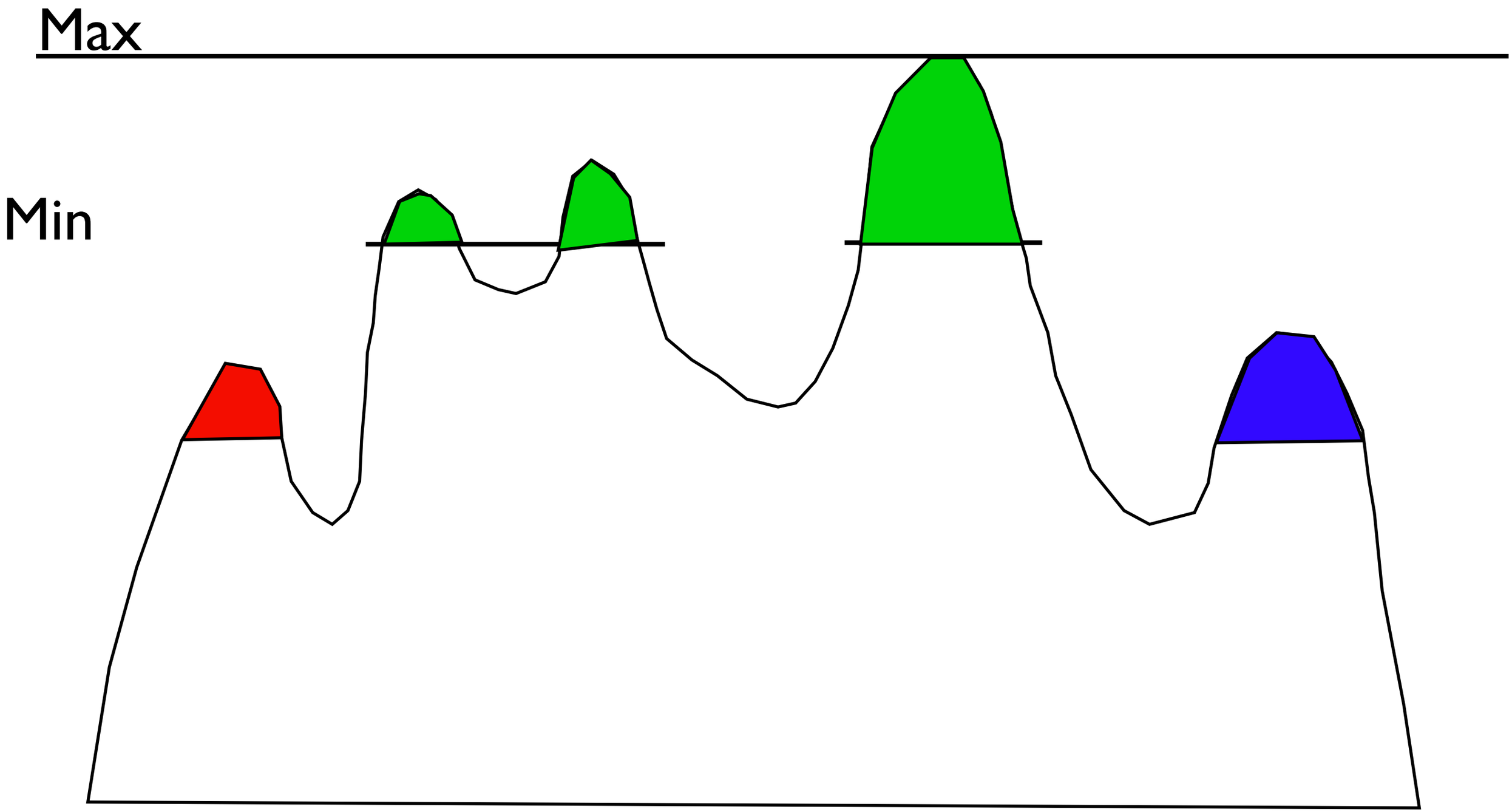
Finding Clumps



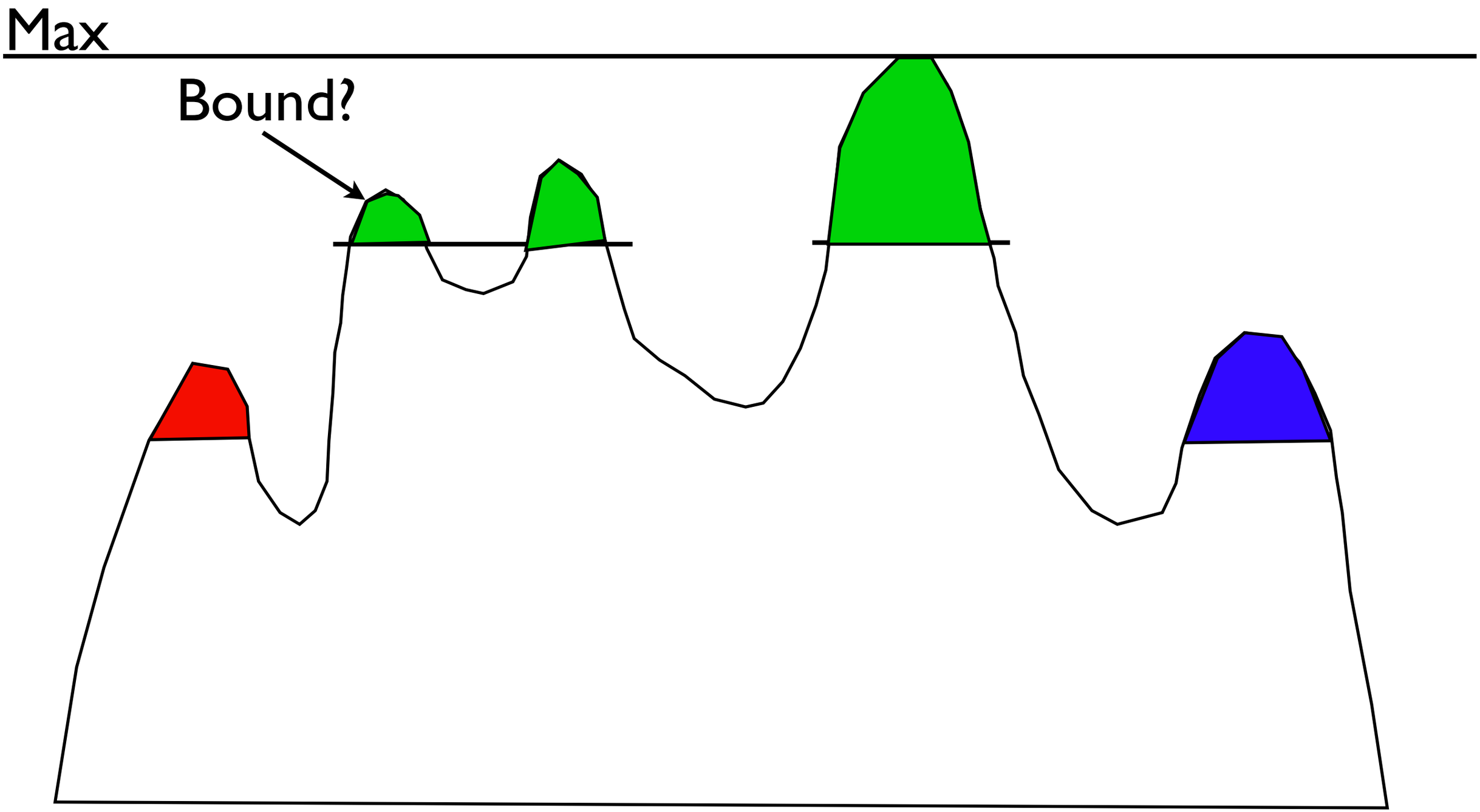
Finding Clumps



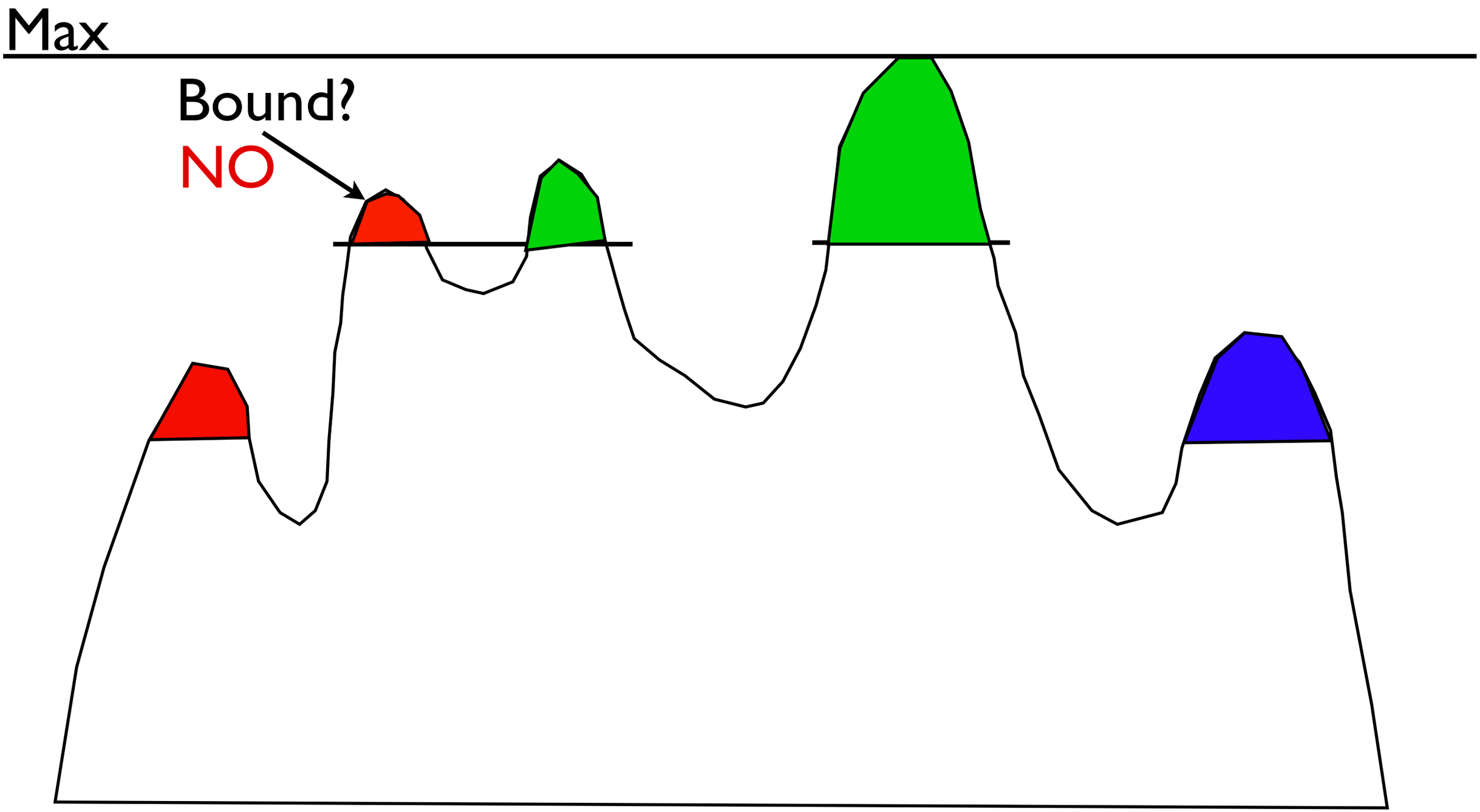
Finding Clumps



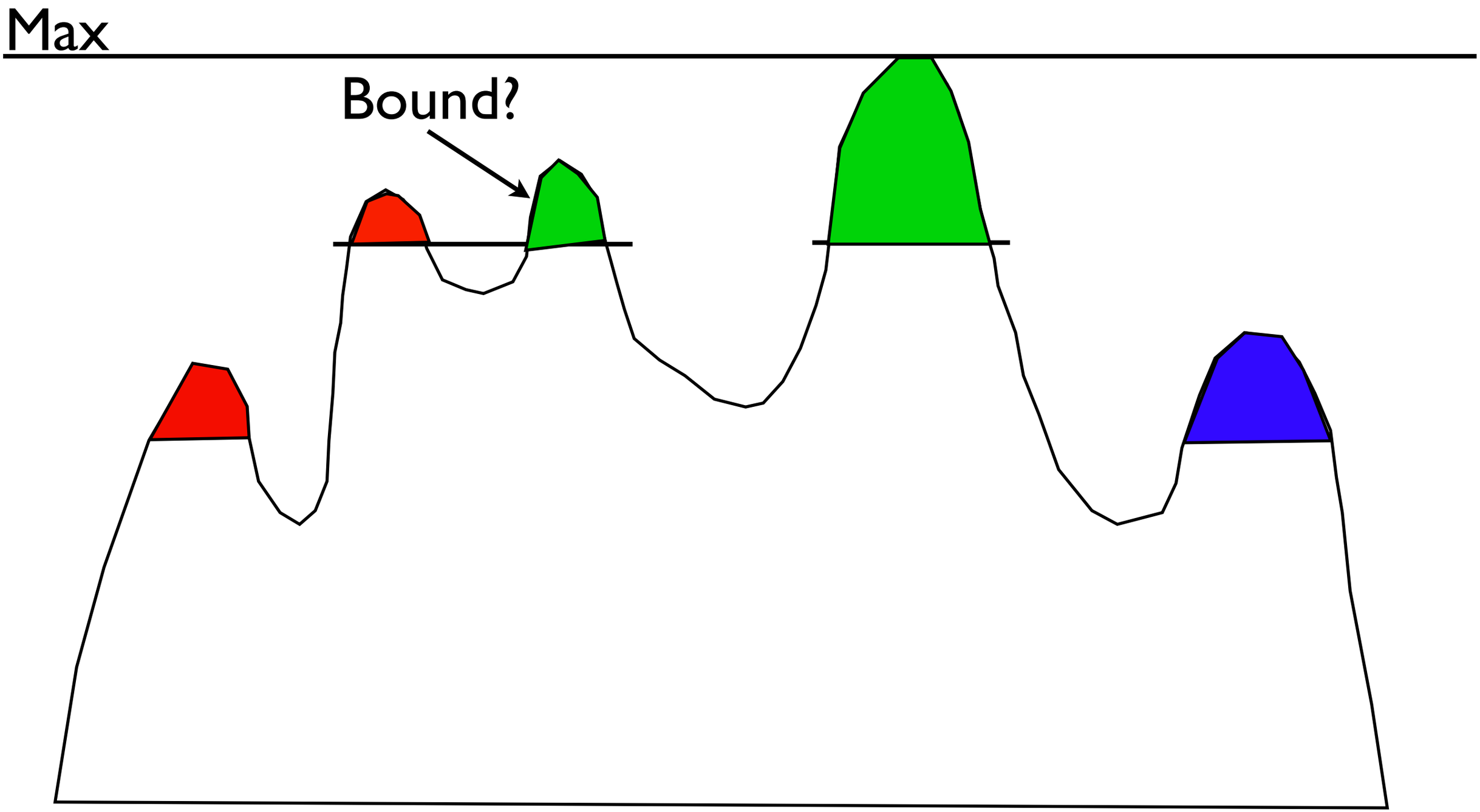
Finding Clumps



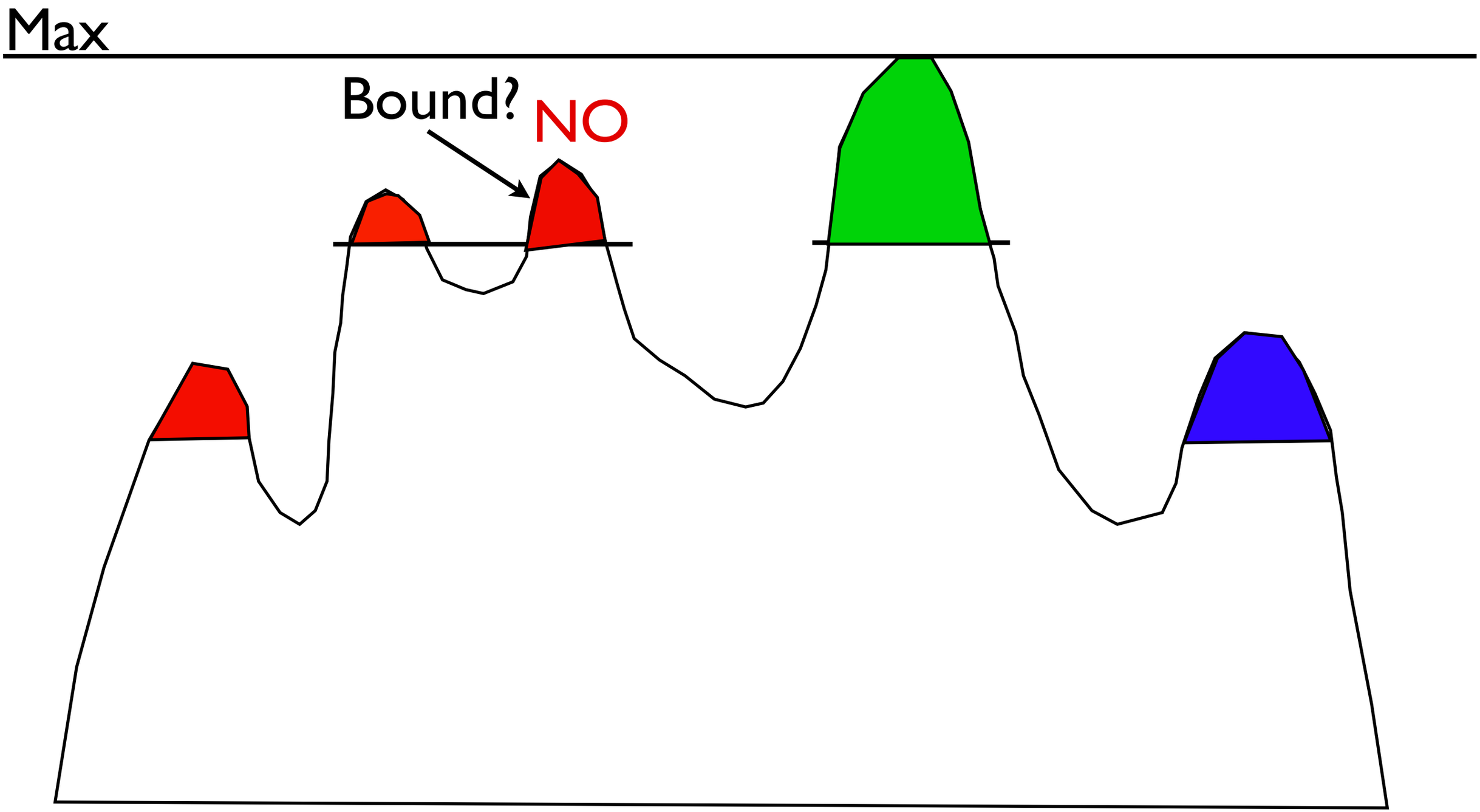
Finding Clumps



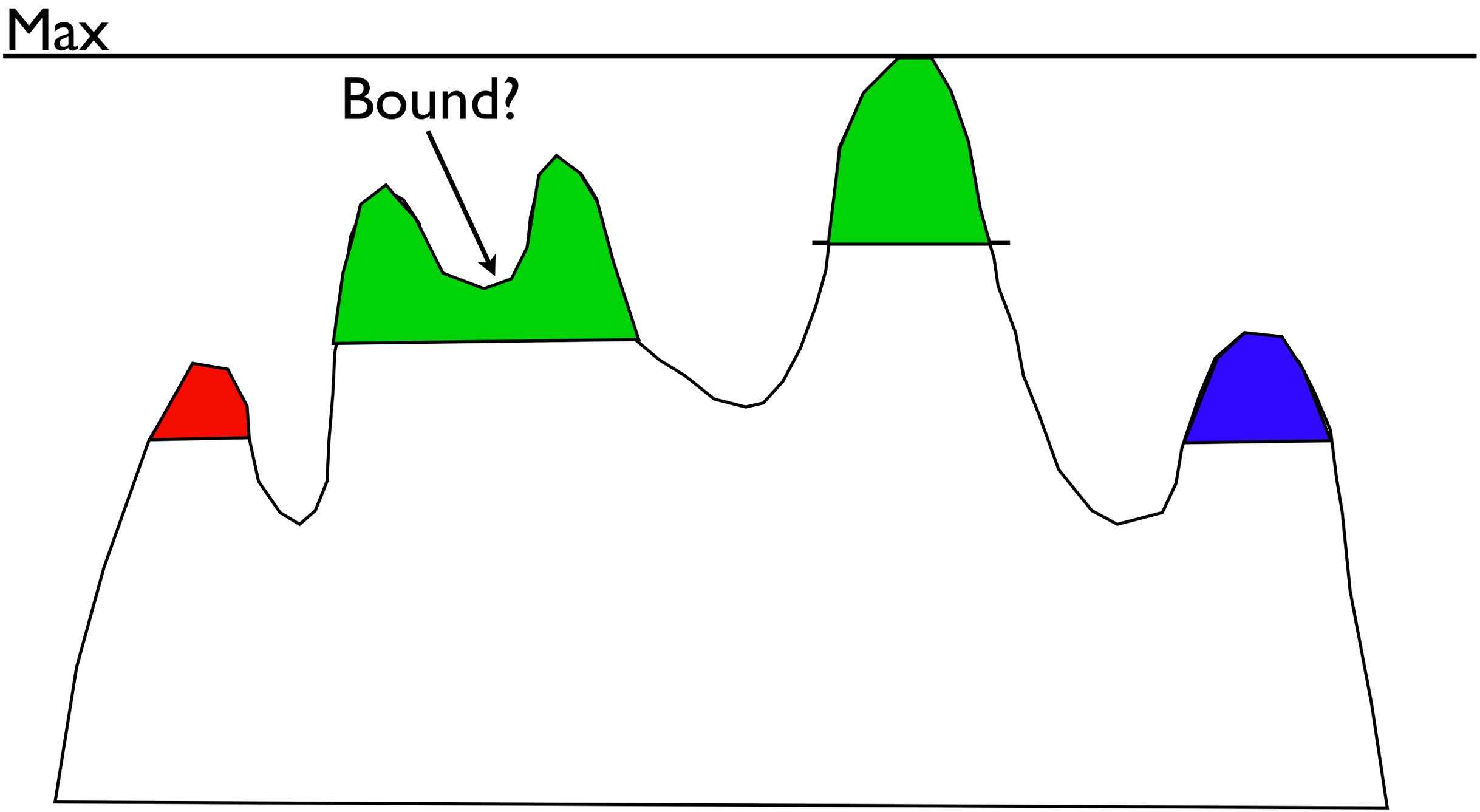
Finding Clumps



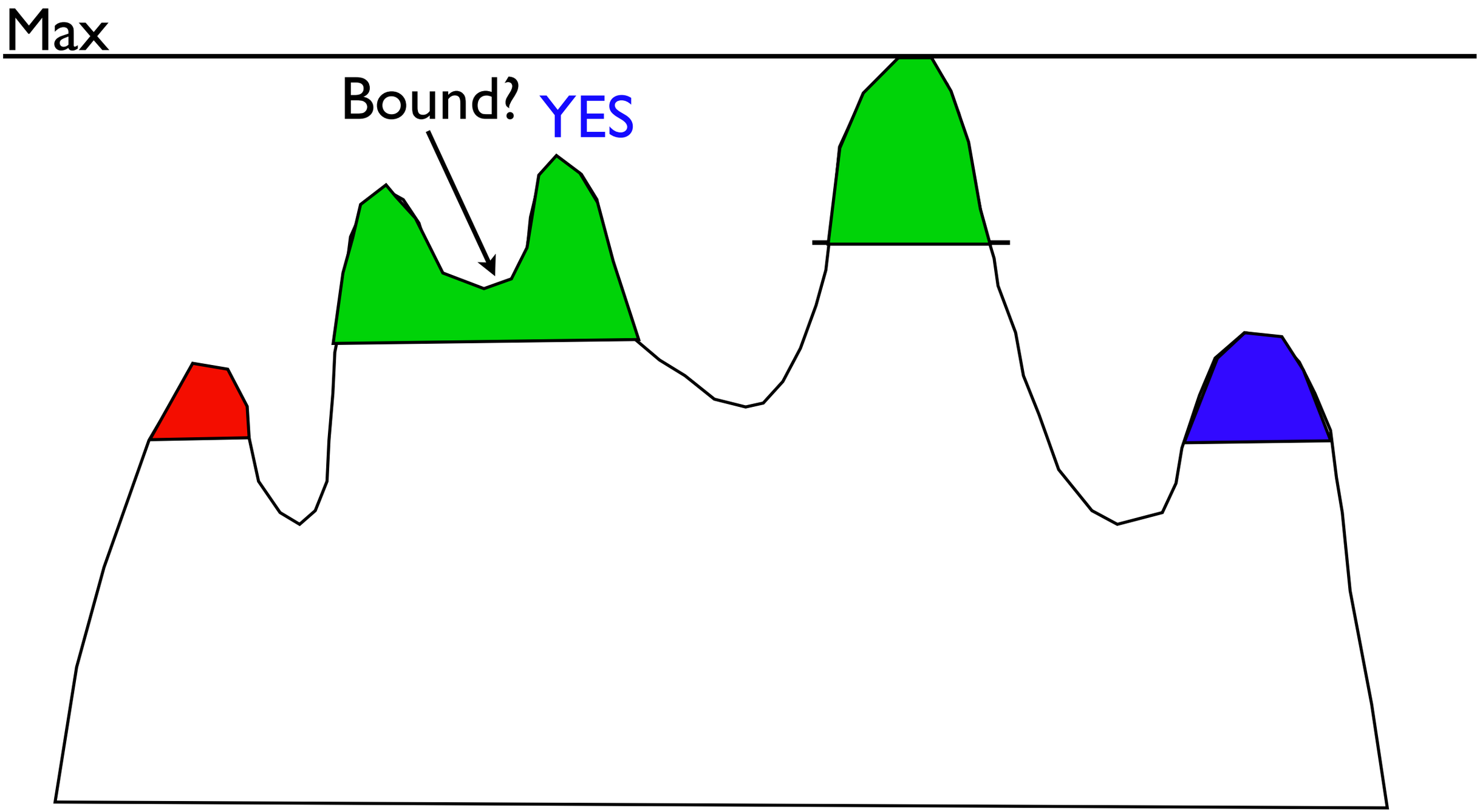
Finding Clumps



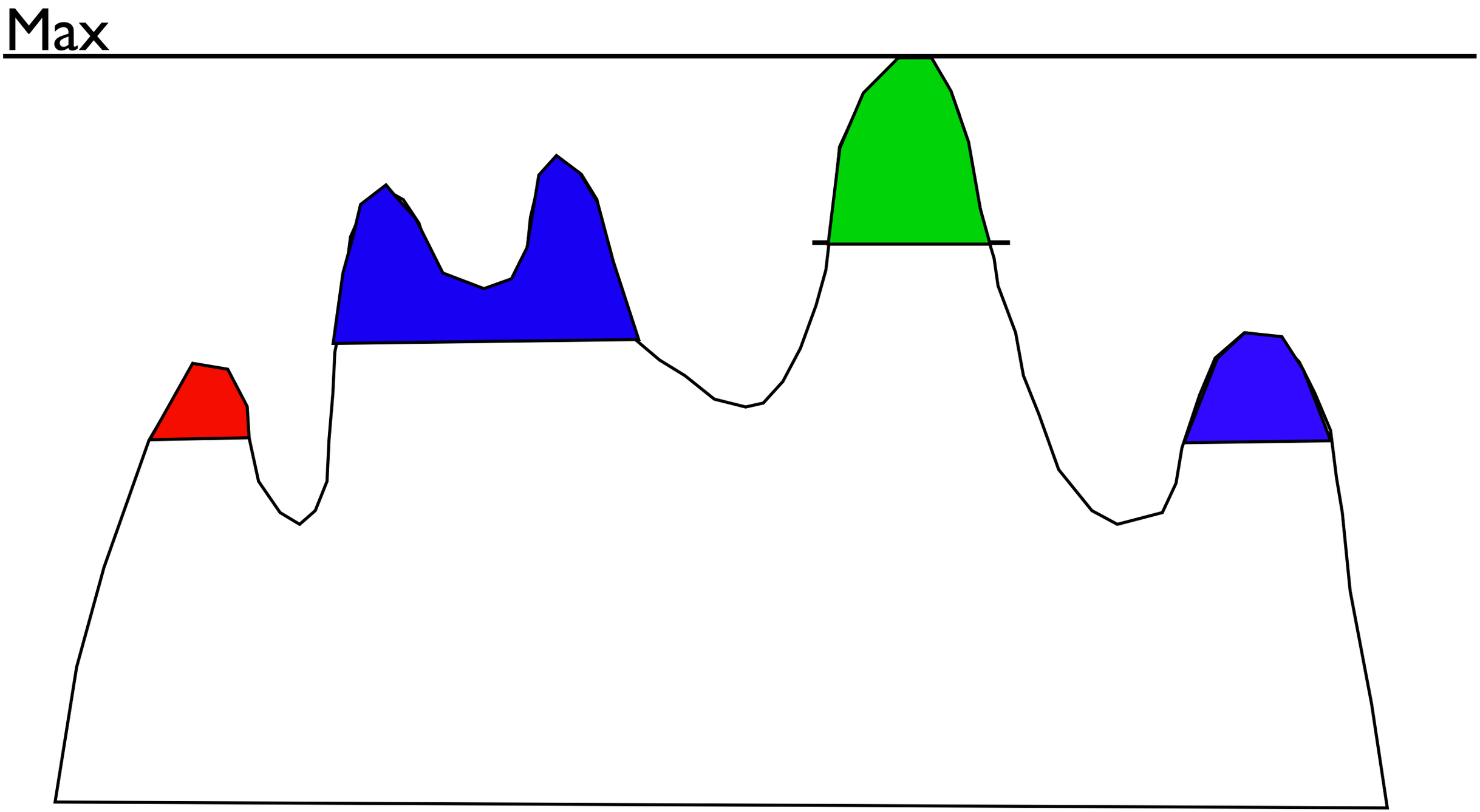
Finding Clumps



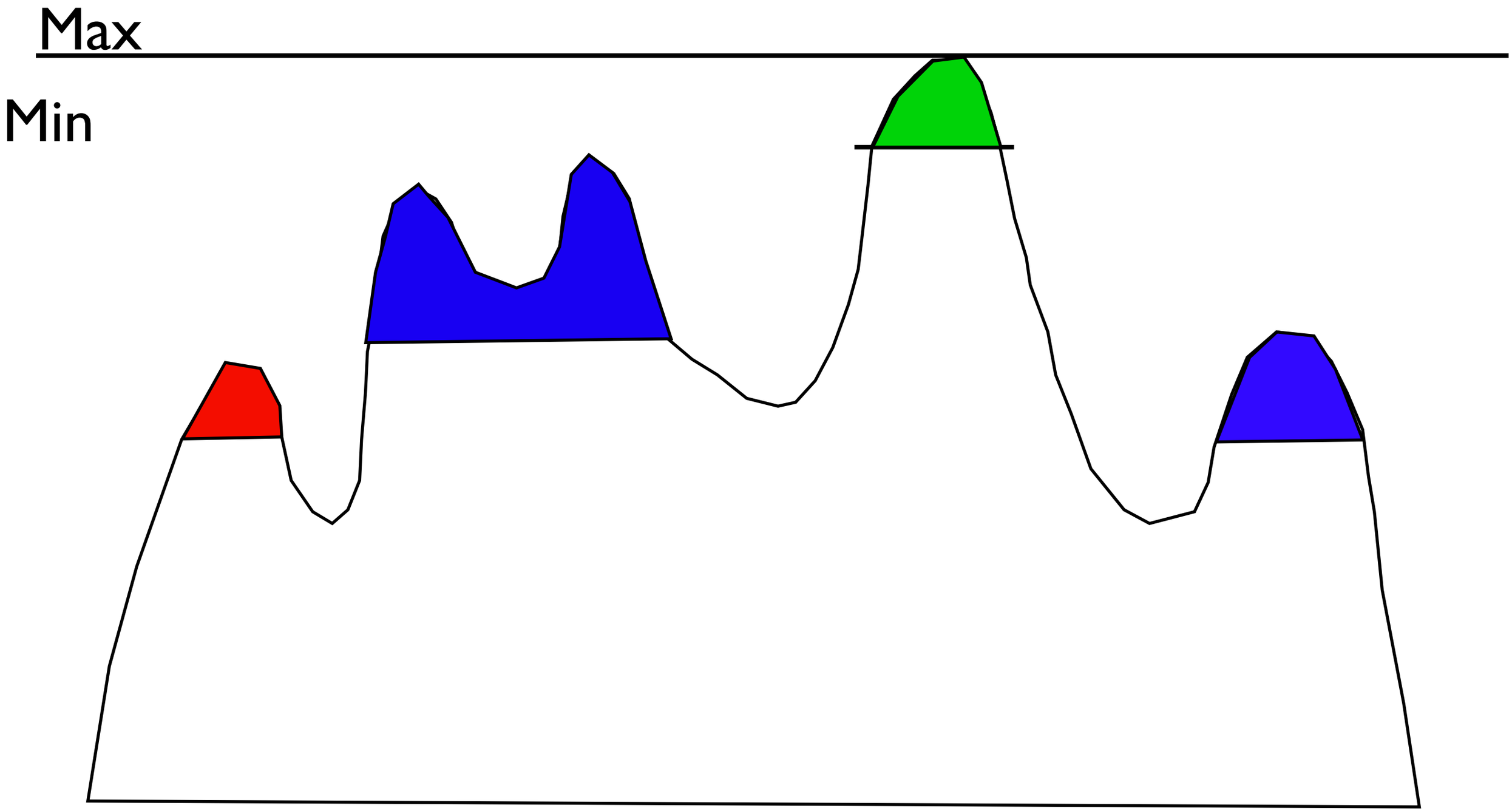
Finding Clumps



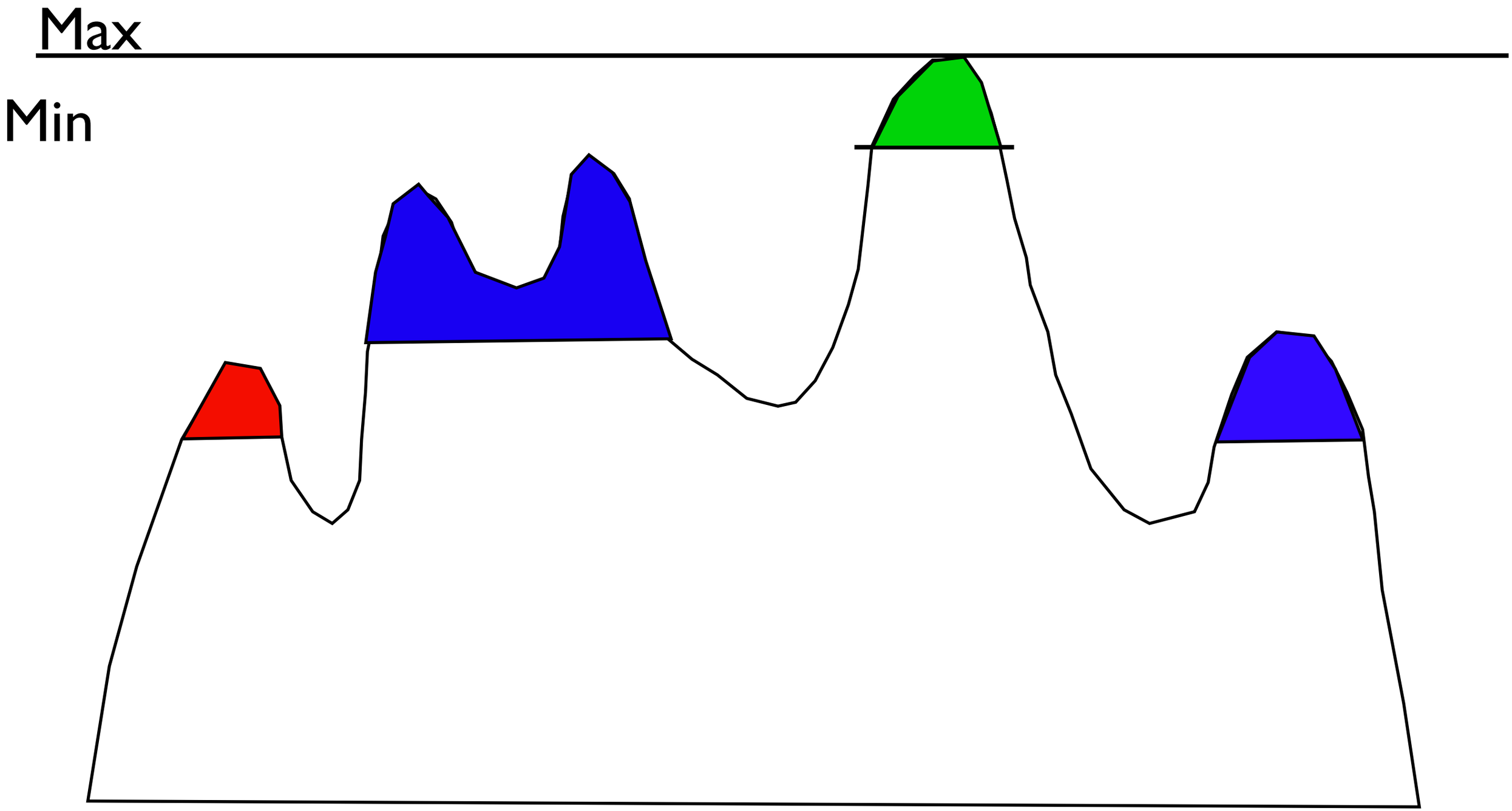
Finding Clumps



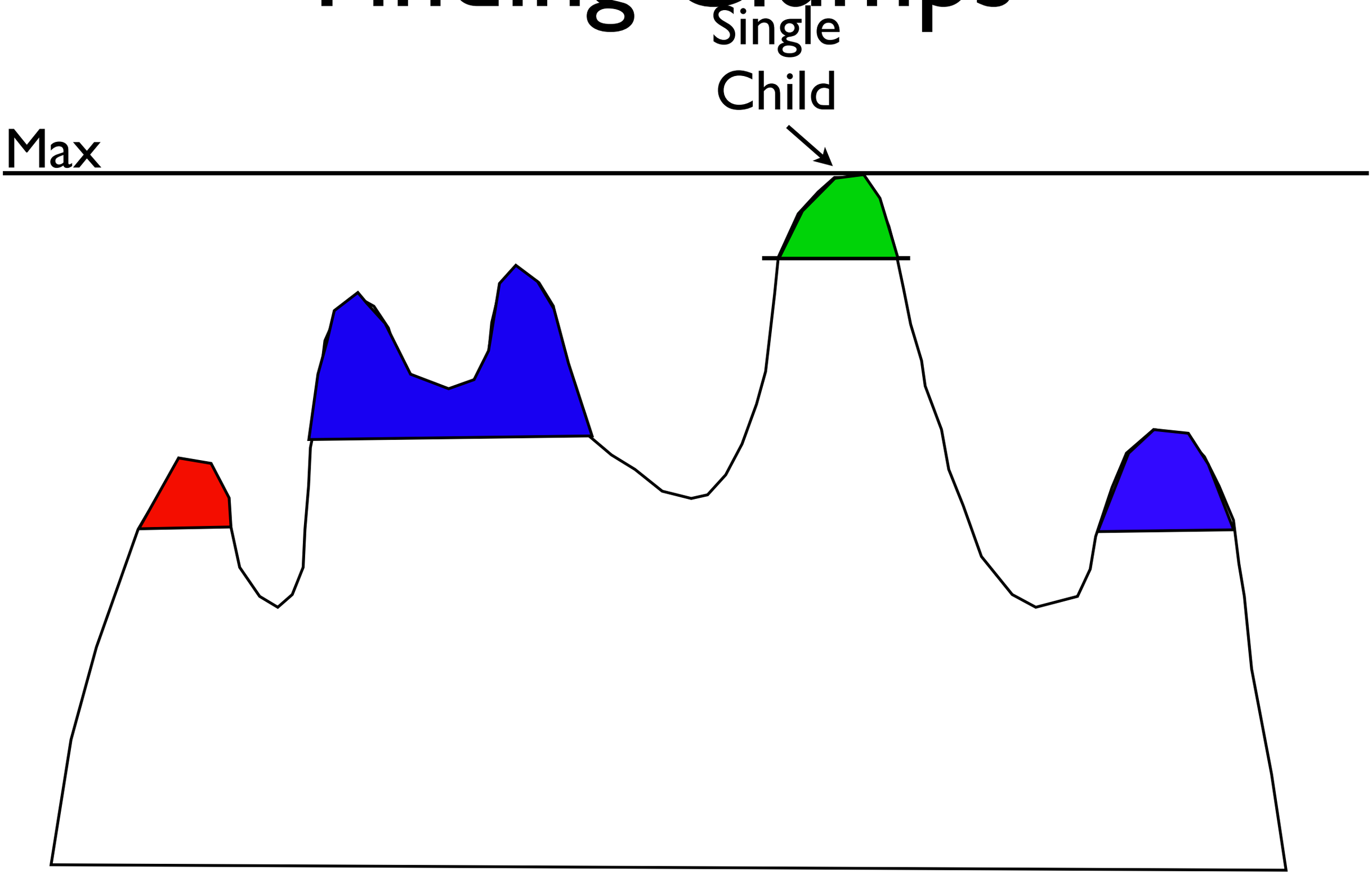
Finding Clumps



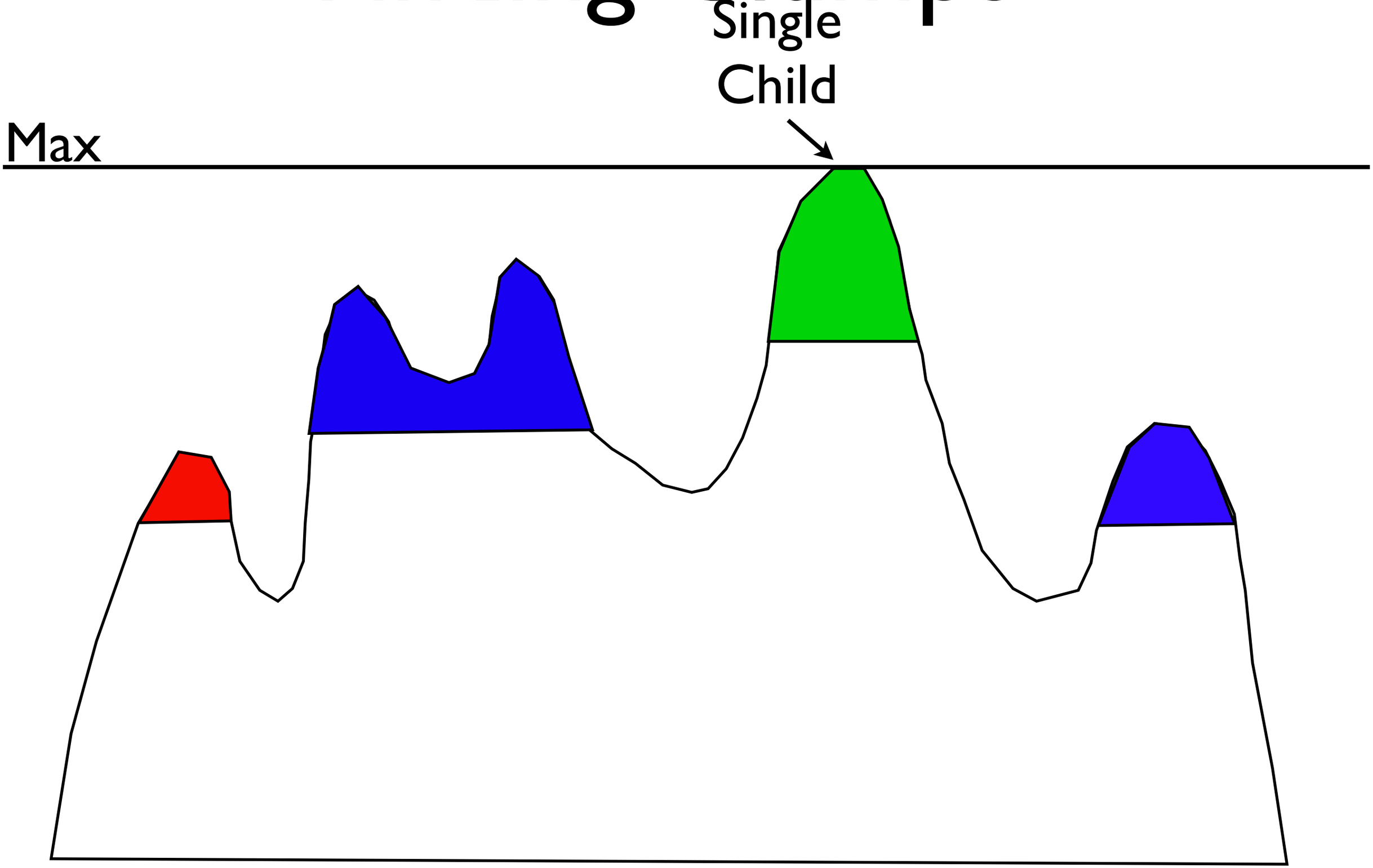
Finding Clumps



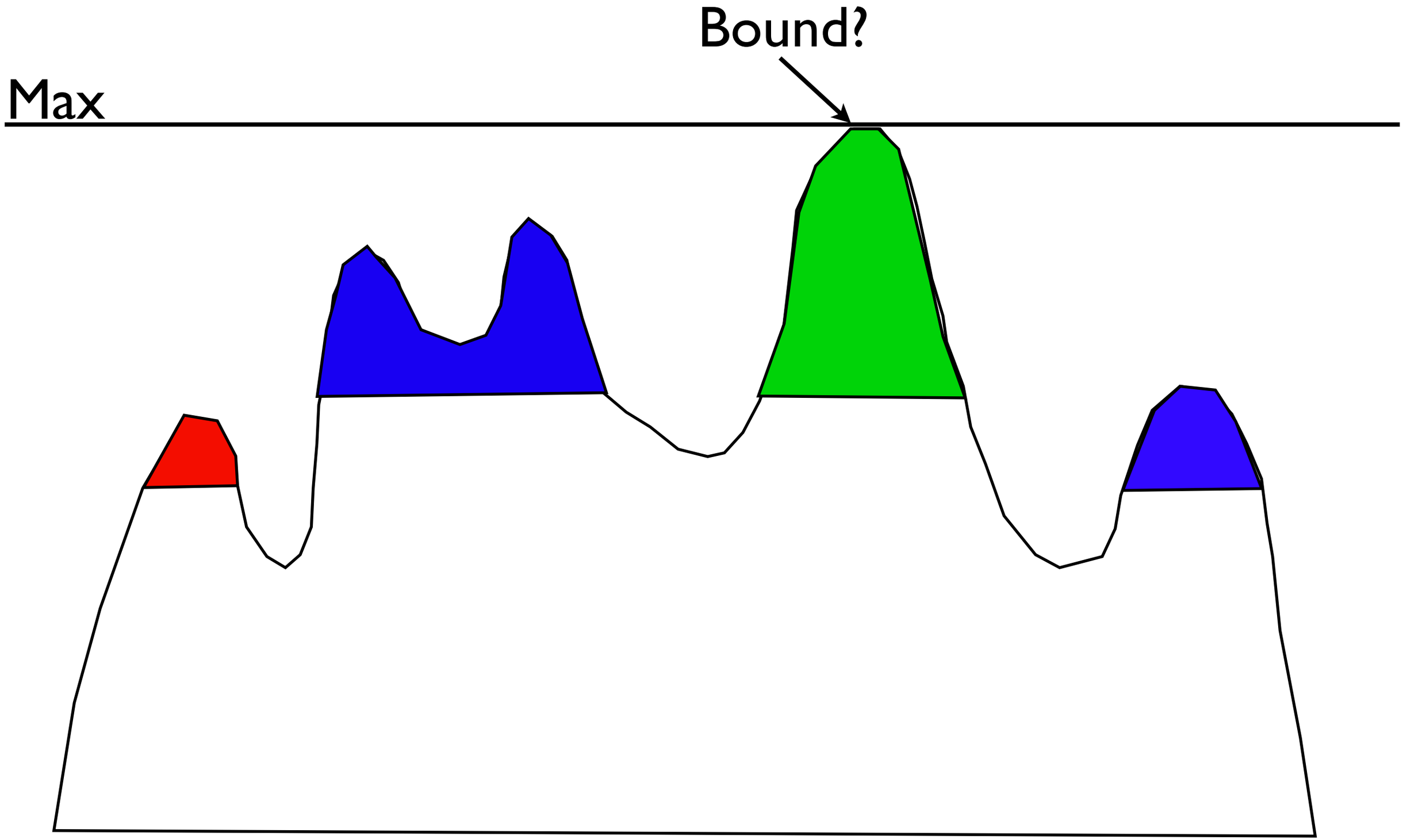
Finding Clumps



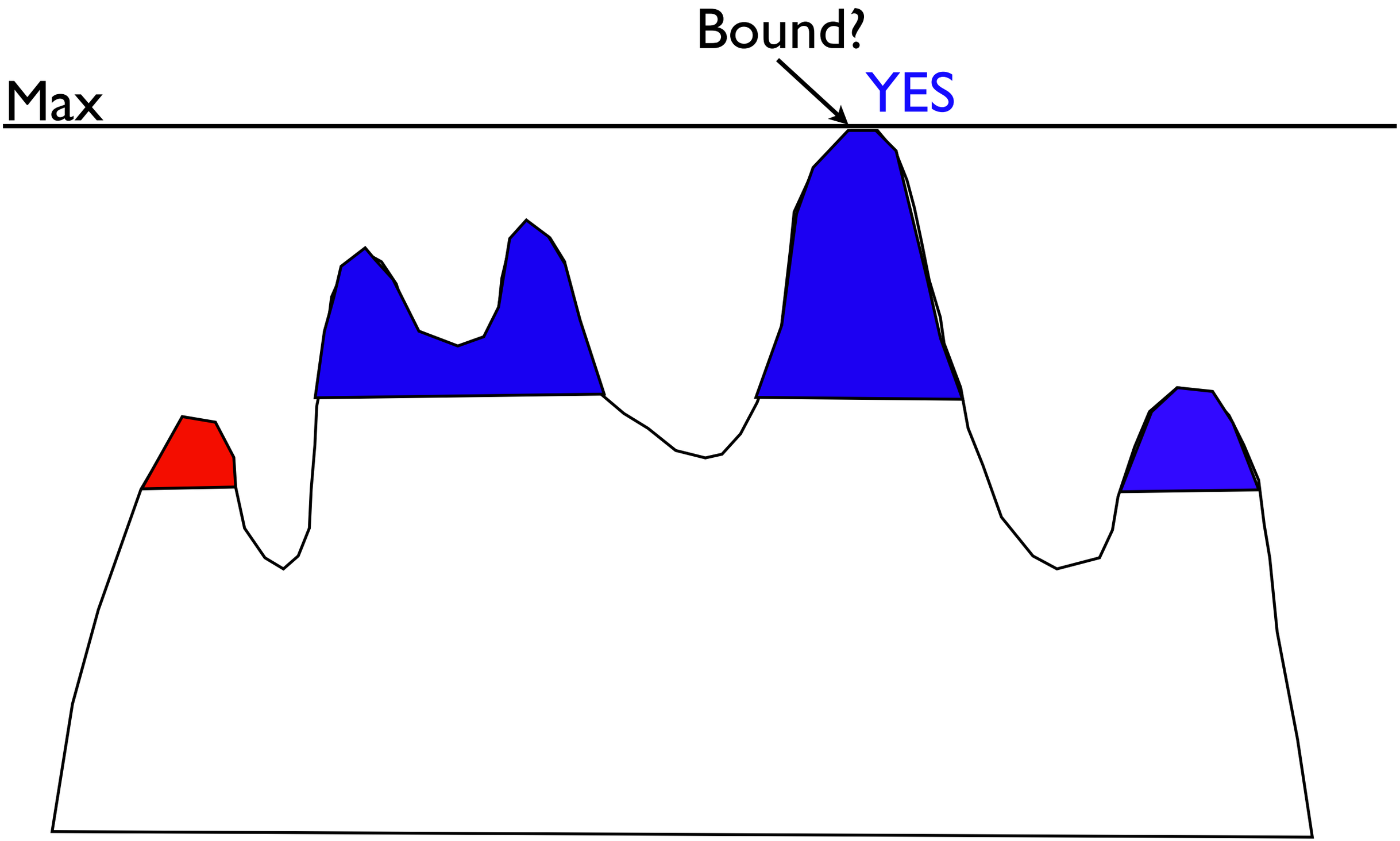
Finding Clumps



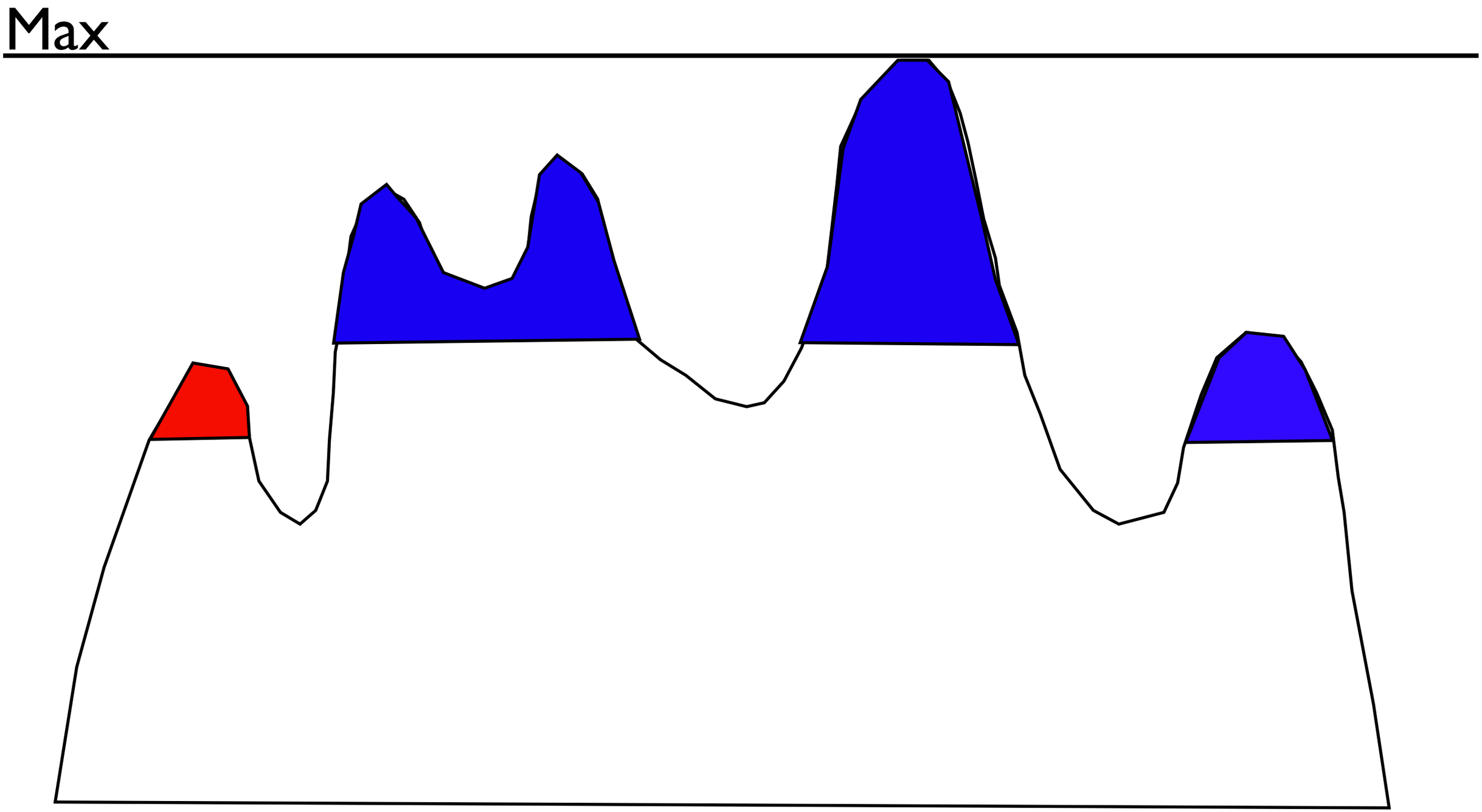
Finding Clumps



Finding Clumps

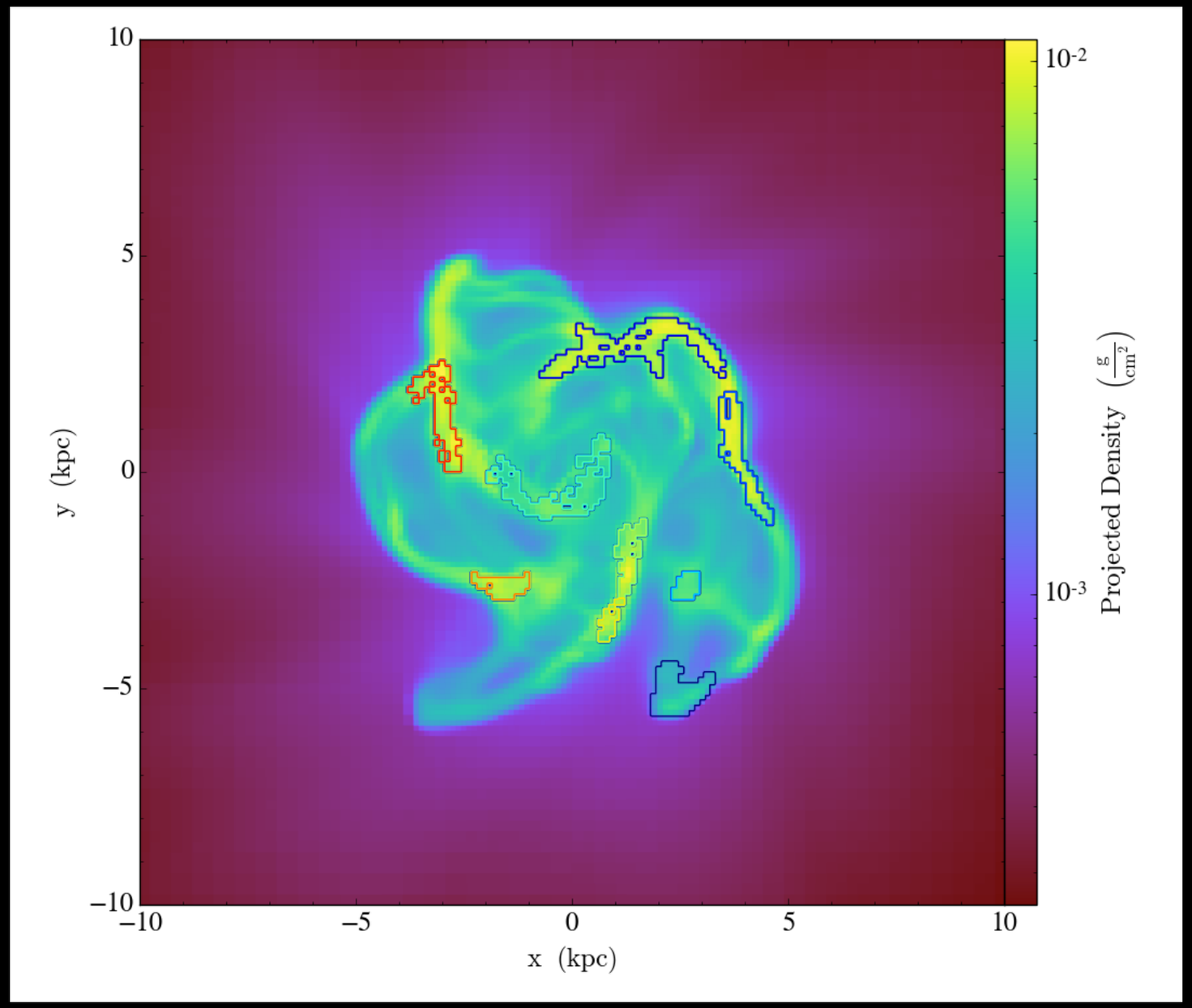


Finding Clumps



Using the clump finder

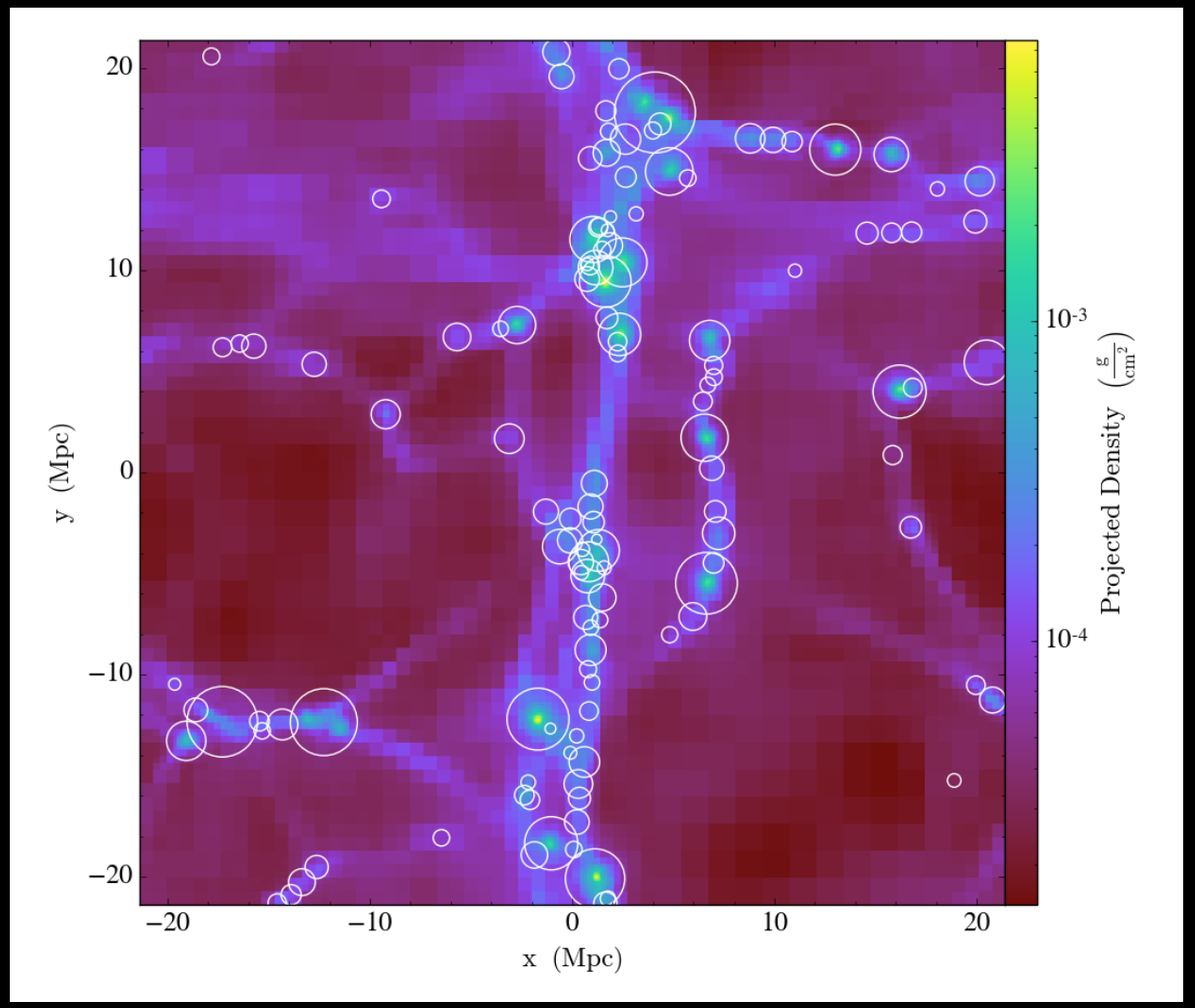
- Create “master” clump
- Add validators
  - Gravitationally bound?
- Find clumps
- Analyze, write to disk, and plot

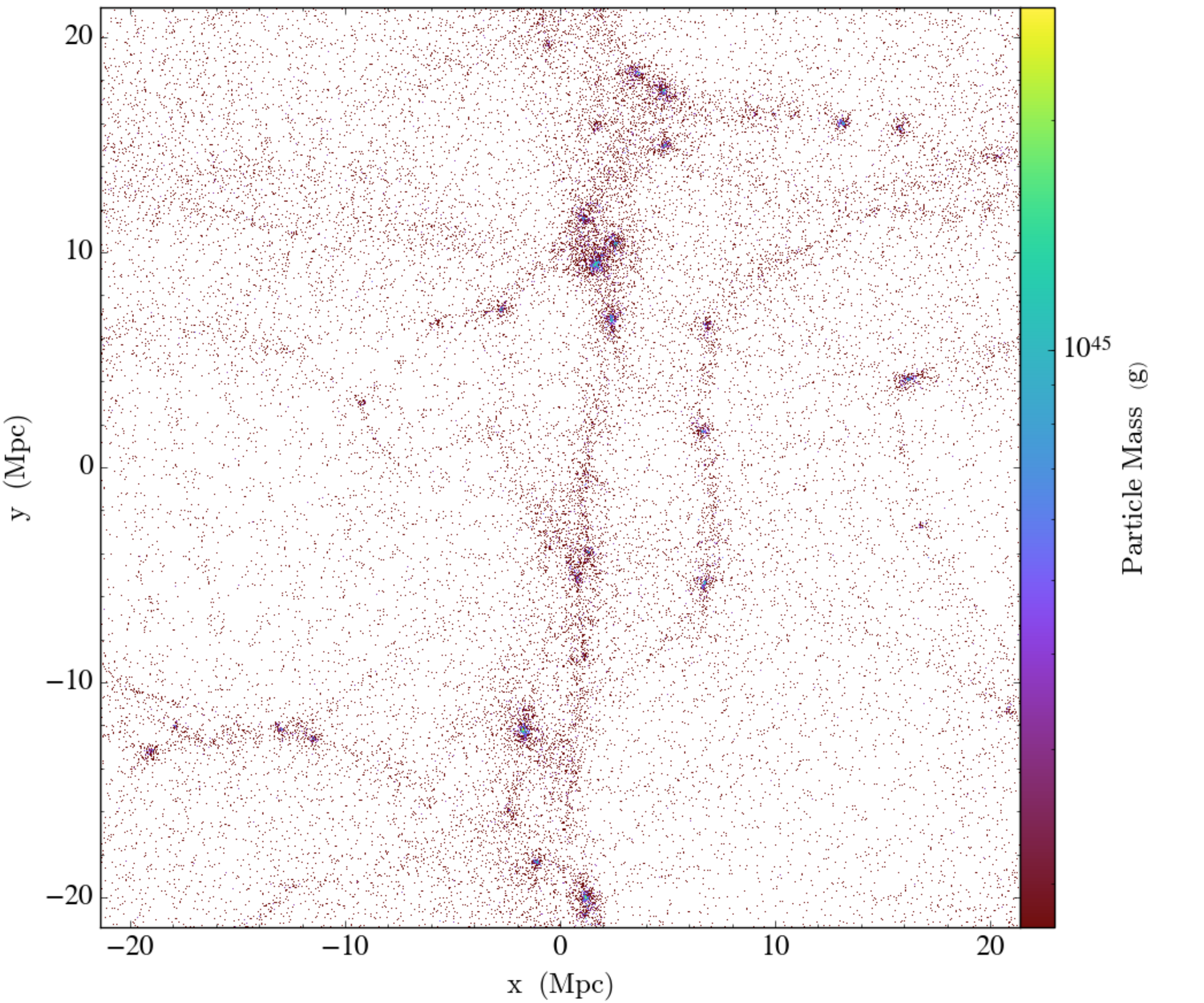


$ yt pastebin_grab 6867 > find_clumps.py

Halo finding and analysis

- Halo Finders
  - FOF, Hop, Rockstar
- Halo analysis
  - Halo catalog
  - Saving and reloading




20
10
0
−10
−20
y (Mpc)
−20
−10
0
10
20
x (Mpc)
10^{45}
Particle Mass (g)

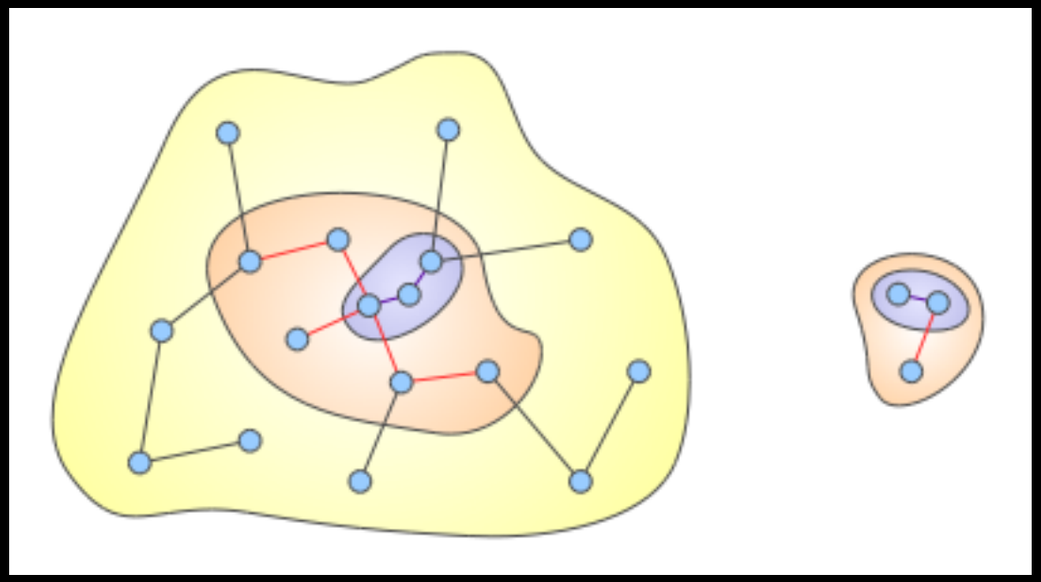
Halo Finders

Identify particle clusters based on linking length (FOF)

Calculate R_vir, mass, position

HOP, Rockstar, similar but add refinements like smoothing and taking into account velocity information

Halo Finders

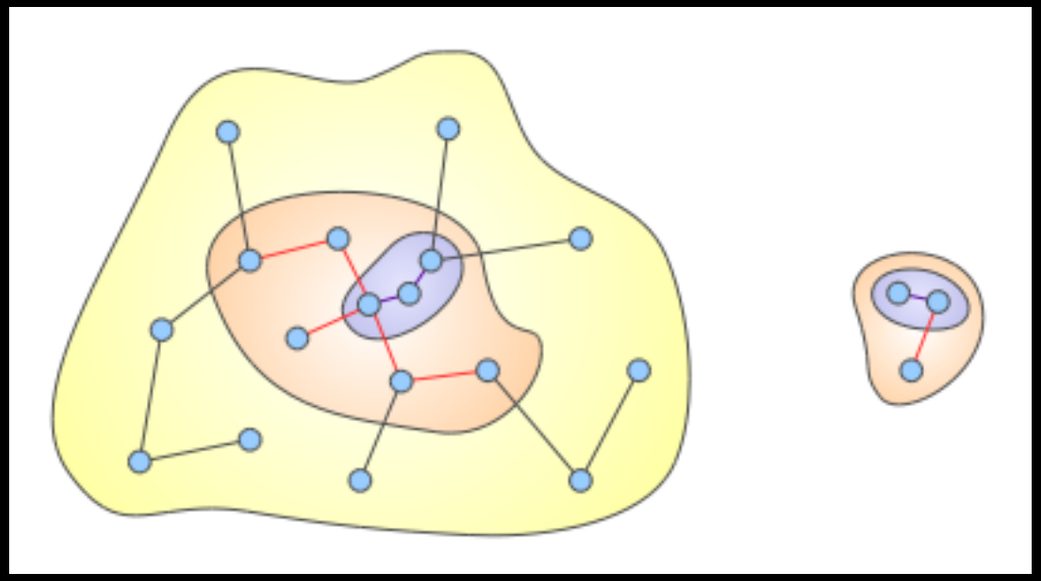


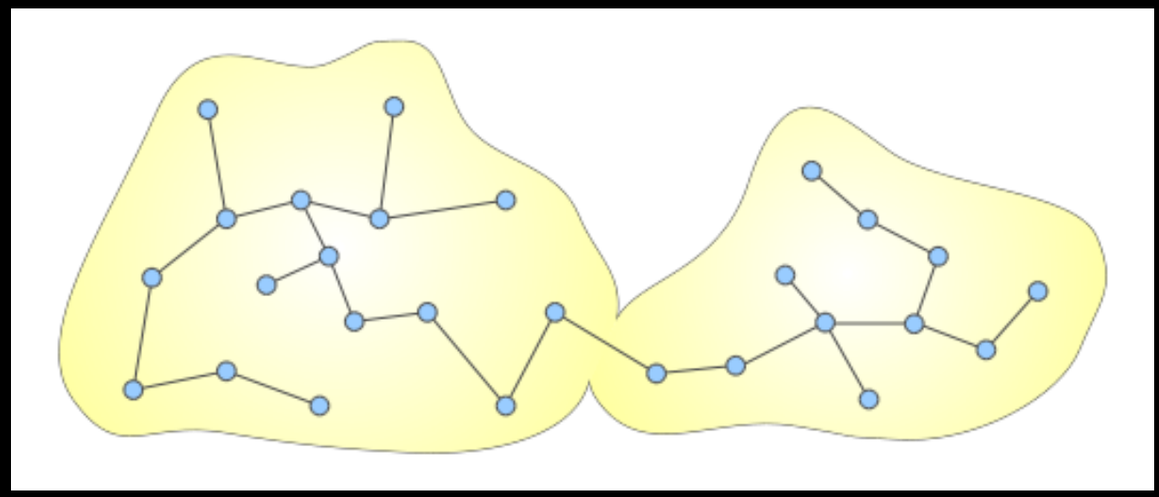
Identify particle clusters based on linking length (FOF)

Calculate R_vir, mass, position

HOP, Rockstar, similar but add refinements like smoothing and taking into account velocity information

Halo Finders





Identify particle clusters based on linking length (FOF)

Calculate R_vir, mass, position

HOP, Rockstar, similar but add refinements like smoothing and taking into account velocity information

Installing Rockstar

- Install script: easiest method
- Build Matt's fork of rockstar: http://bitbucket.org/MatthewTurk/rockstar
- Build yt from source, pointing it at the rockstar build directory by adding a rockstar.cfg file
    - Seems to be an issue on MacOS Sierra?

Running Rockstar

```
import yt
yt.enable_parallelism()
from yt.analysis_modules.halo_finding.rockstar.api import RockstarHaloFinder


# create a particle filter to remove star particles
@yt.particle_filter("dark_matter", requires=["creation_time"])
def _dm_filter(pfilter, data):
    return data["creation_time"] <= 0.0


def setup_ds(ds):
    ds.add_particle_filter("dark_matter")


es = yt.simulation("enzo_cosmology_plus/AMRCosmology.enzo", "Enzo")
es.get_time_series(setup_function=setup_ds, redshift_data=False)
rh = RockstarHaloFinder(es, num_readers=1, num_writers=2,
                        particle_type="dark_matter")
rh.run()
```

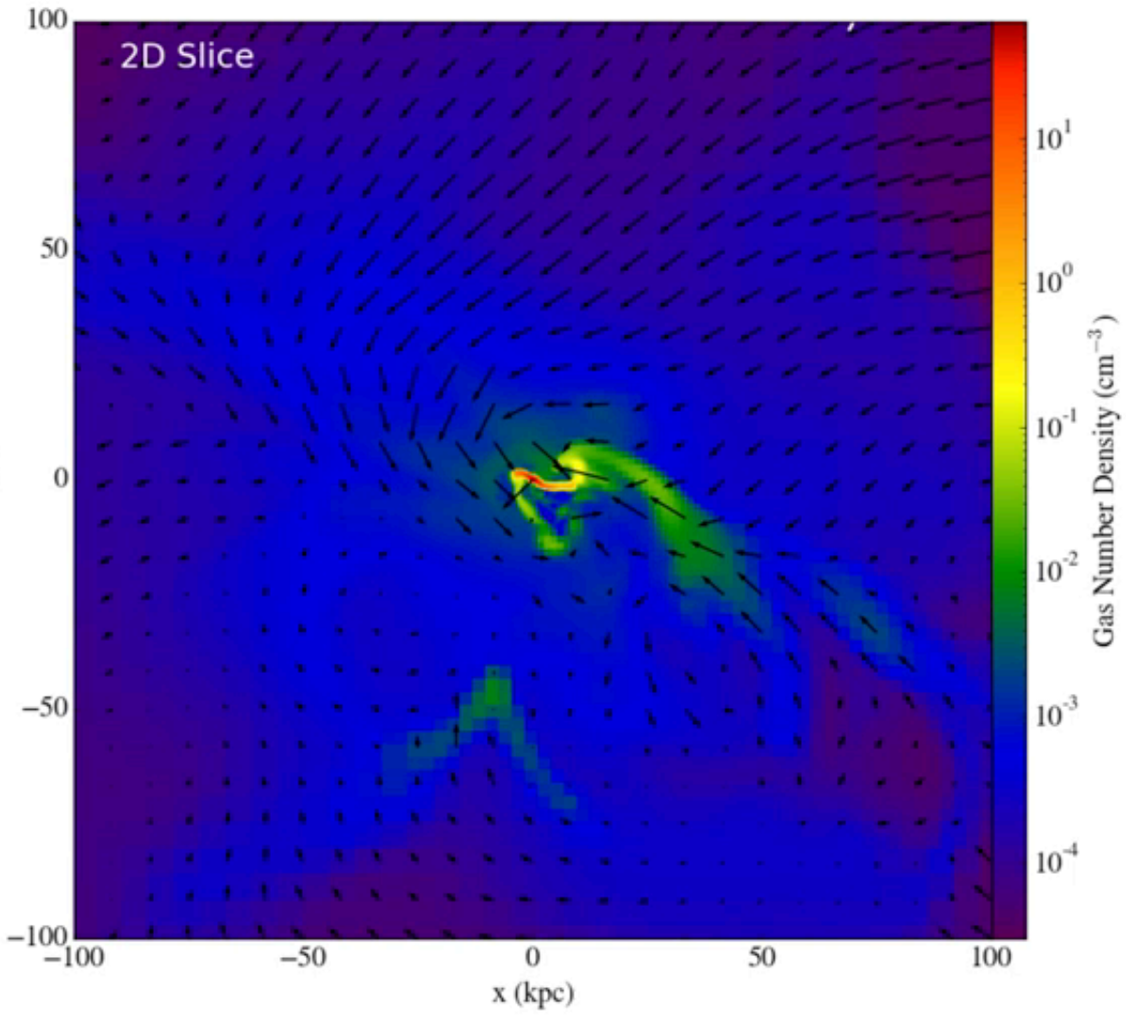
Needs at least 3 MPI tasks

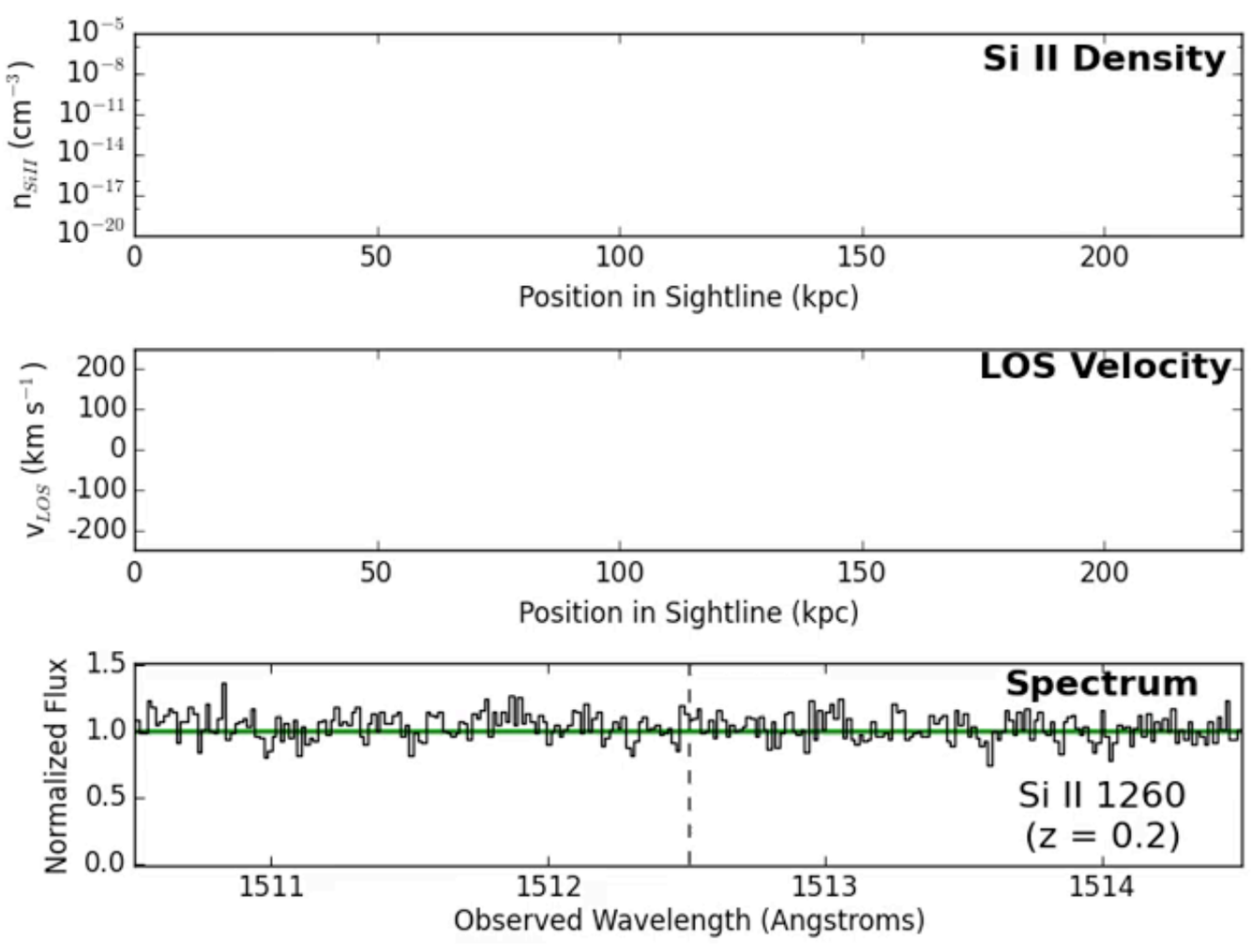
Halo analysis example

http://yt-project.org/docs/dev/cookbook/halo_analysis_example.html

Let’s modify this example in the yt docs to use this dataset we’ve been working with

Trident

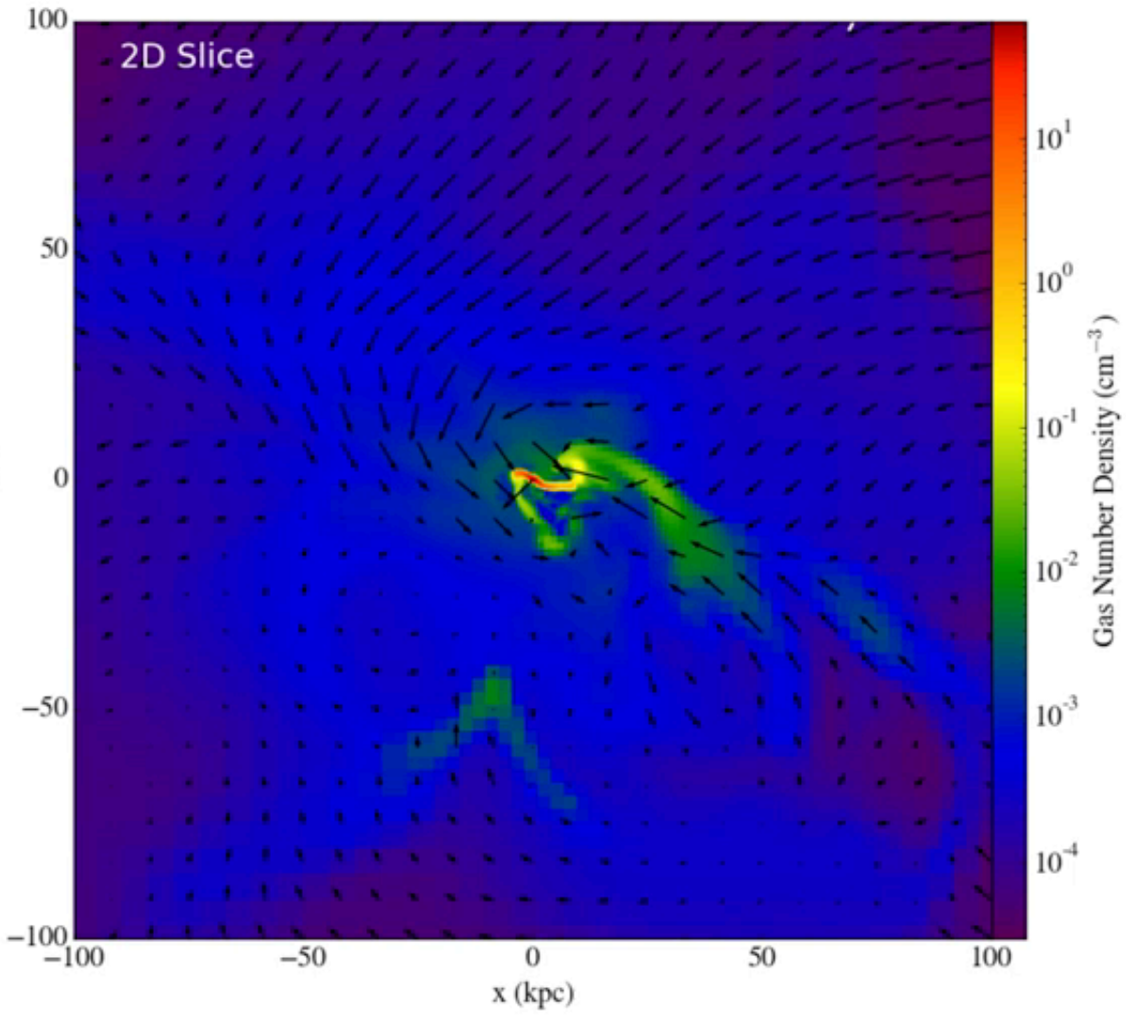


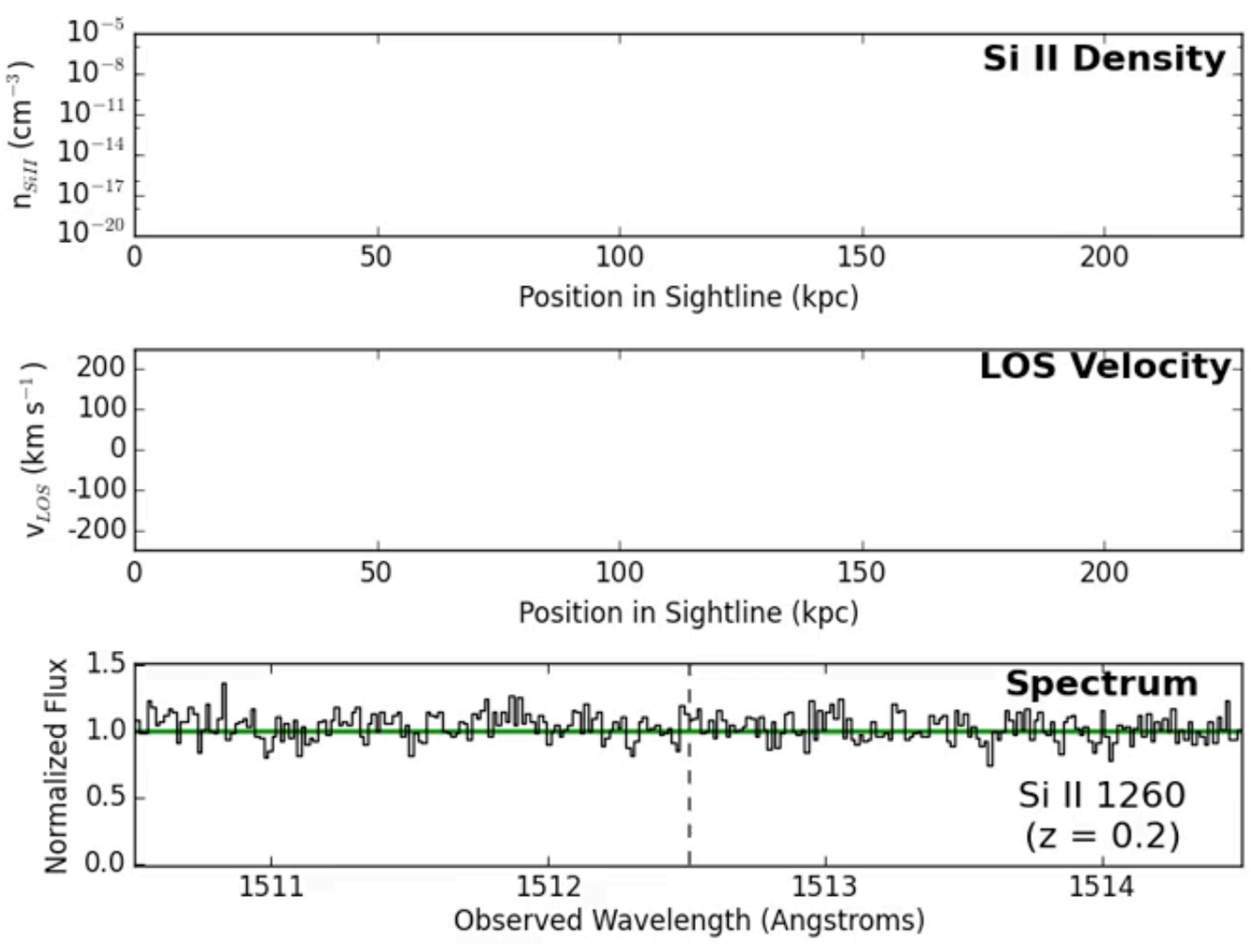


http://trident.readthedocs.io/en/latest

http://trident-project.org

Trident

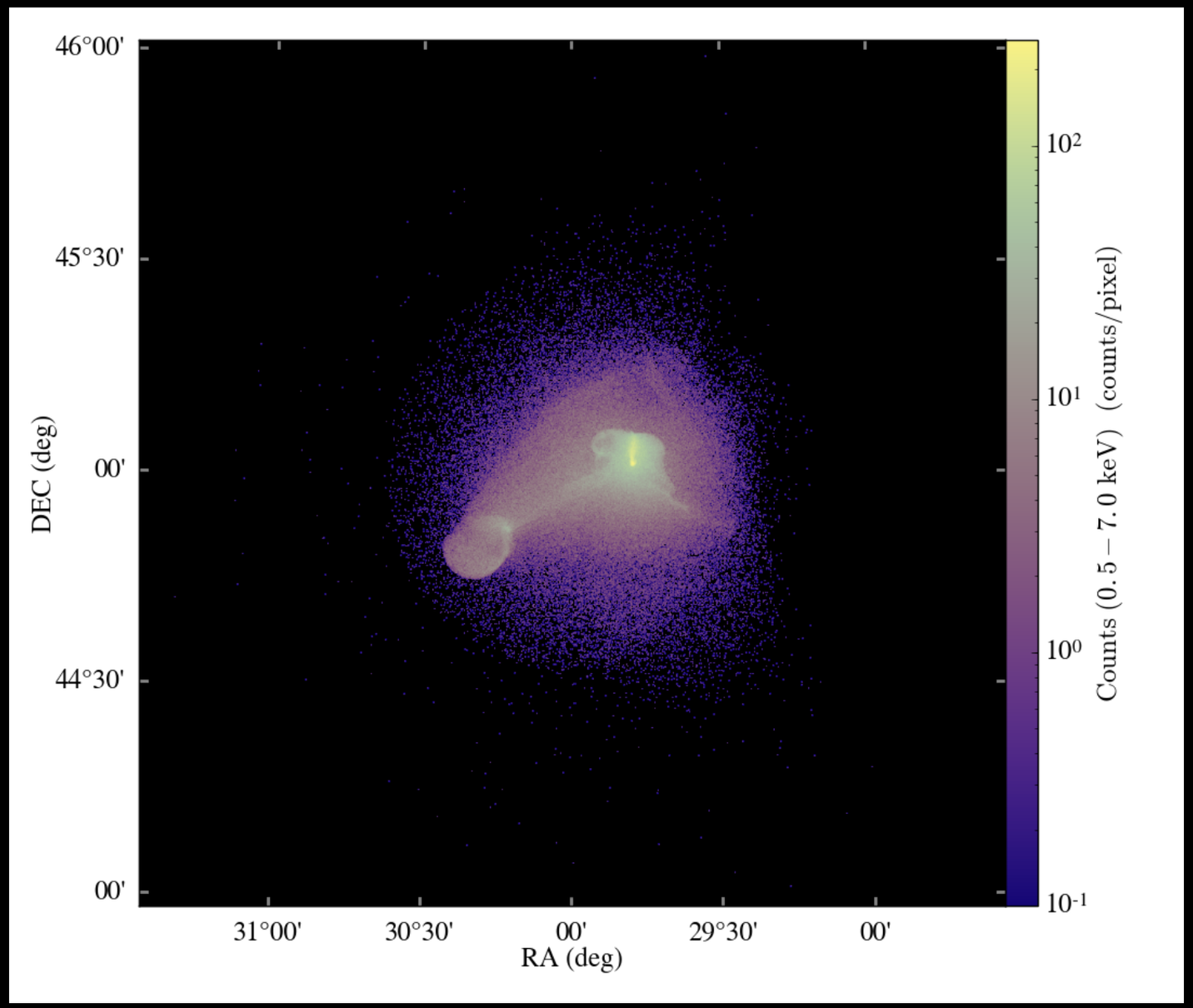




http://trident.readthedocs.io/en/latest

http://trident-project.org

pyXSIM



Simulating xray observations

http://hea-www.cfa.harvard.edu/~jzuhone/pyxsim/index.html