Considering Interface Concavity in Spongy Mesophyll Segmentation

Evan Cook

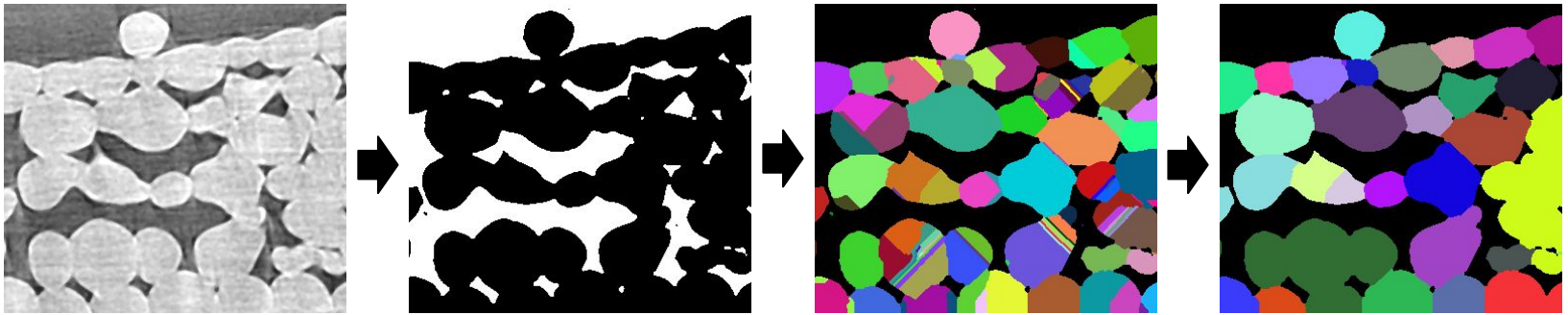
About Me

- Undergrad, rising senior at the University of Chicago
- Double major in Math and CS
- Grew up in Cupertino, California

Cupertino Library

Project Motivation I

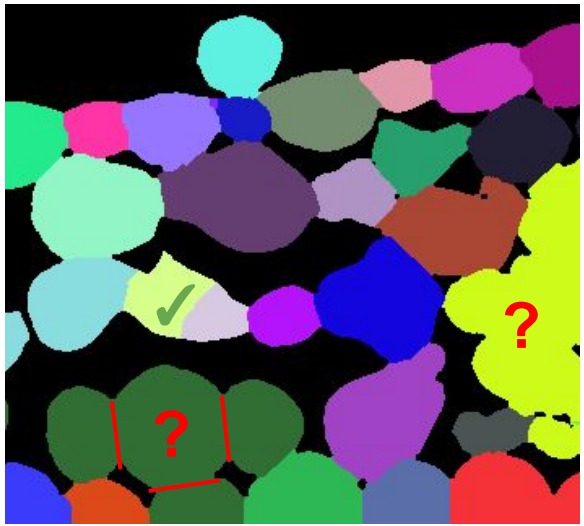
- Recall: goal is “spongy mesophyll segmentation”
- Correct segmentation ⟶ data collection for future studies
- Current pipeline is below:
  - (1) **Threshold** pixel values + remove artifact holes
  - (2) Perform **watershed** transform ⟶ ‘naive’ result
  - (3) **Prune** (merge) false watershed borders

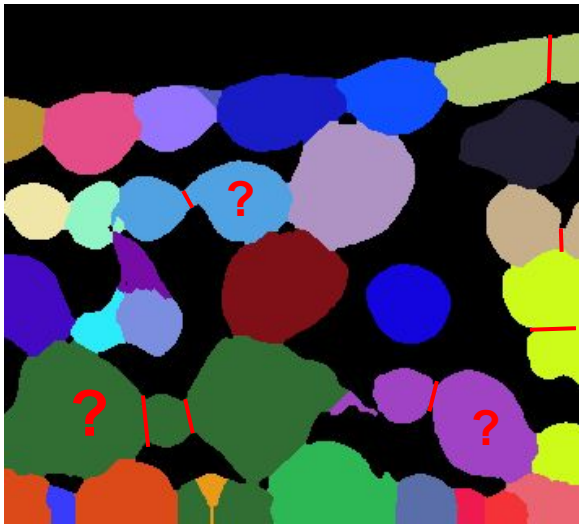


z-slice of original grayscale **binary threshold** **naive watershed** **pruned watershed**

Project Motivation II

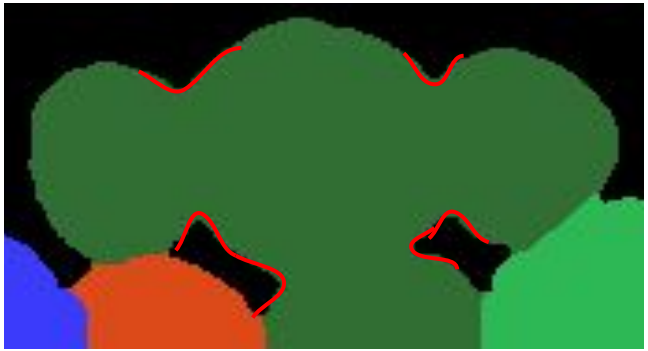
- Results are generally good, but includes some ‘blobs’
- **Undersegmentation** problem vs. **oversegmentation** preference
- Intuition for ‘correct’ cell borders is there – how to translate to algorithm?

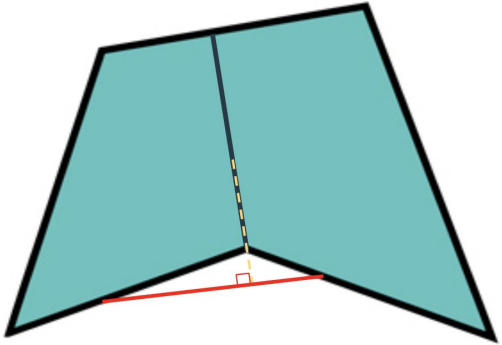




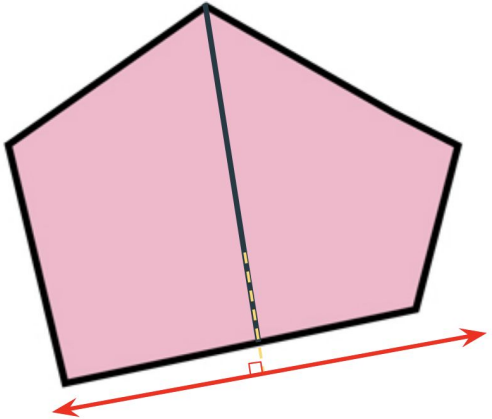
2D Concavity Intuition I

- Intuition has many names – ‘necking,’ ‘pinching,’ ‘concavity’, etc.
- Want to make intuition quantitative, and merge borders using concavity ‘score’
- **Parallel** to cell border, then **perpendicular** to cell matter ⟶ resulting distance measures concavity



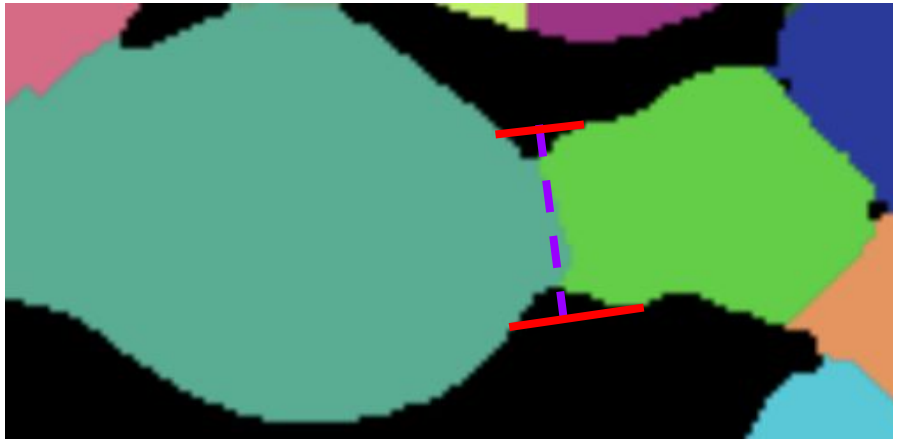


concave border ⟶ lower distance

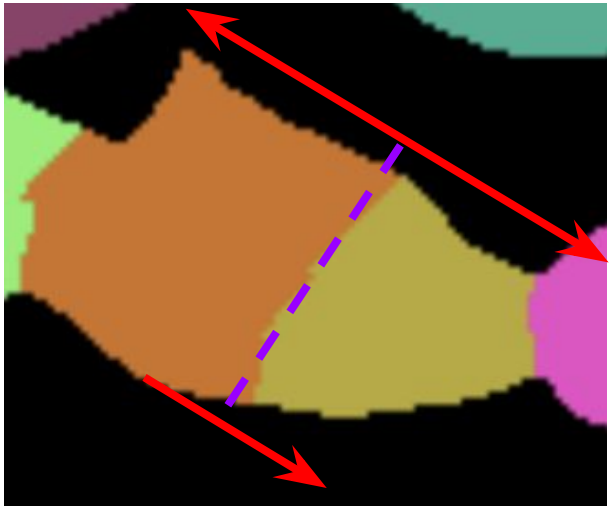


convex border ⟶ higher distance

2D Concavity Intuition II



lower distances → 2 cells



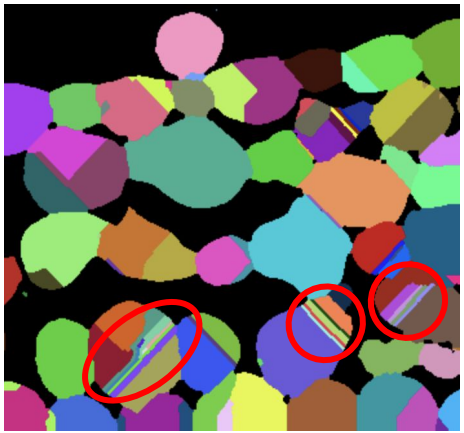
~~**higher distances → 1 cell**~~

2 cells in 3D

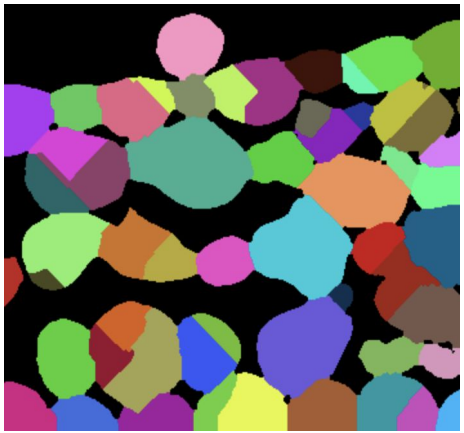
3D Algorithm I

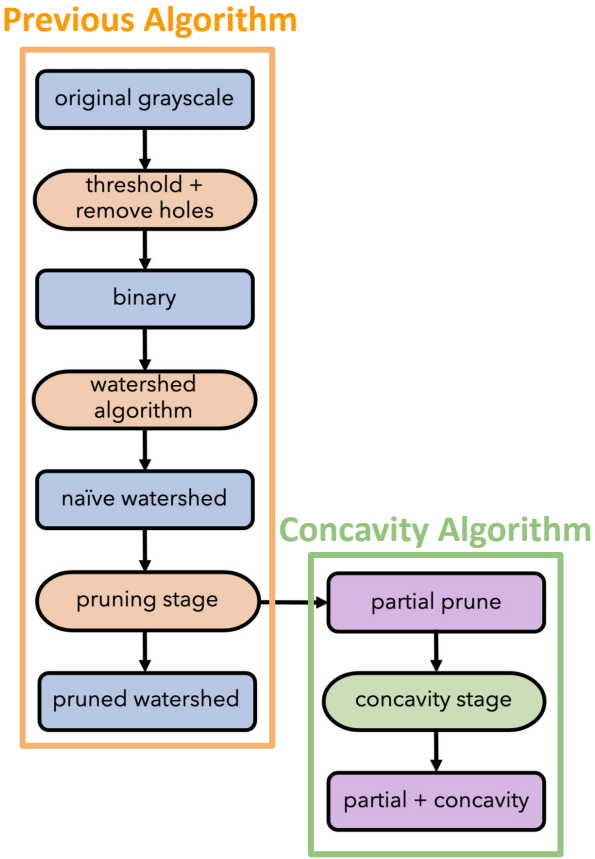
- Most of previous pipeline is preserved
- **partial prune**: like the original pruning stage, but with lower merge strength
- Remove thin strips, let concavity handle edge cases

naive watershed



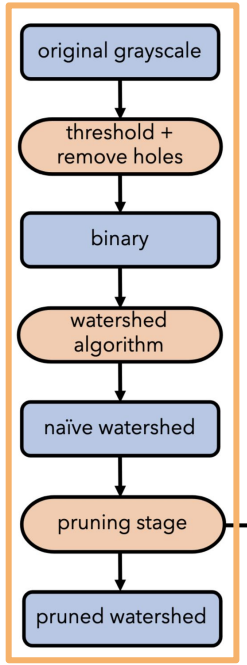
partial prune



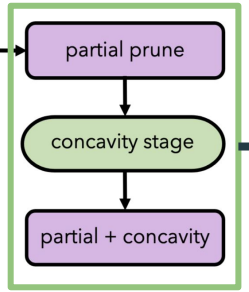


3D Algorithm II

Previous Algorithm



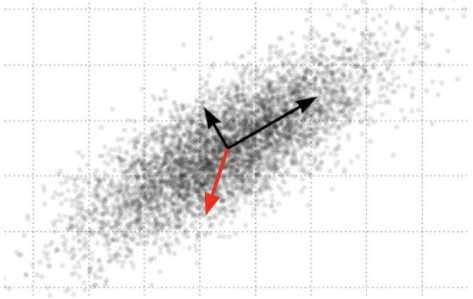
Concavity Algorithm



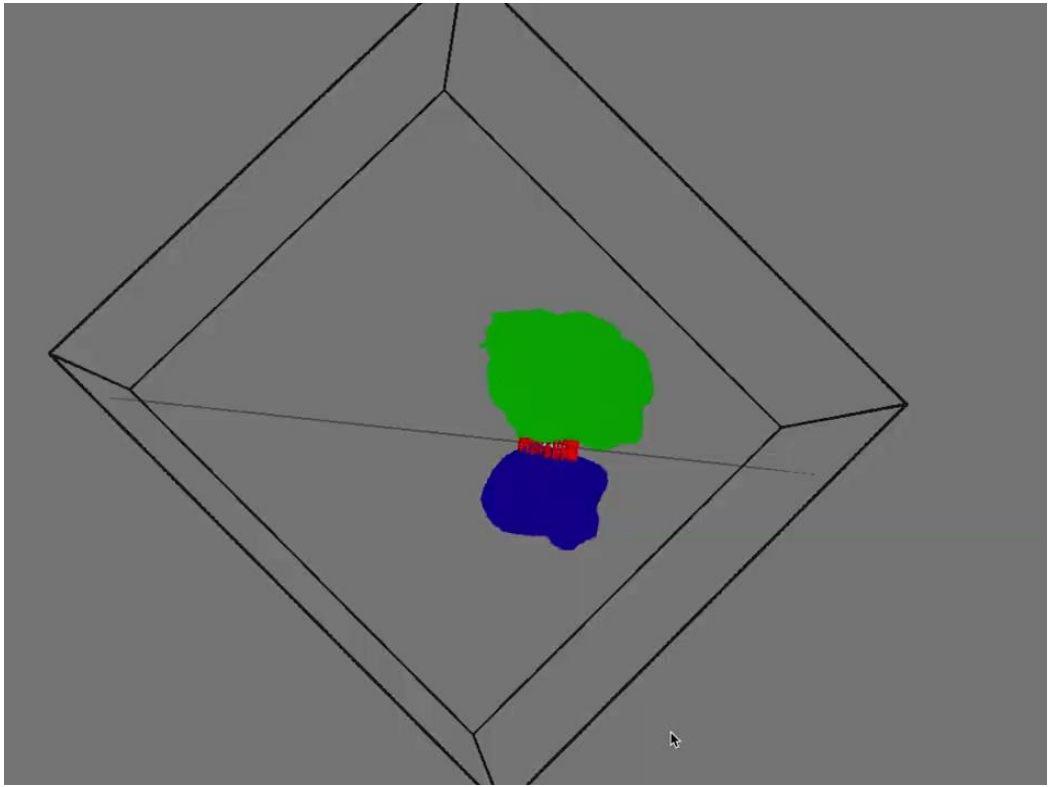
Concavity Algorithm

1. Sample points from cell-cell border
2. Fit a plane to border sample using **PCA**
3. Shift plane to border center, and decompose into unit vectors of various rotations
4. Extend **parallel** to cell border along vectors
5. Find **perpendicular** distance from each extension, and process distances into single score
6. Merge borders with scores over set **threshold**

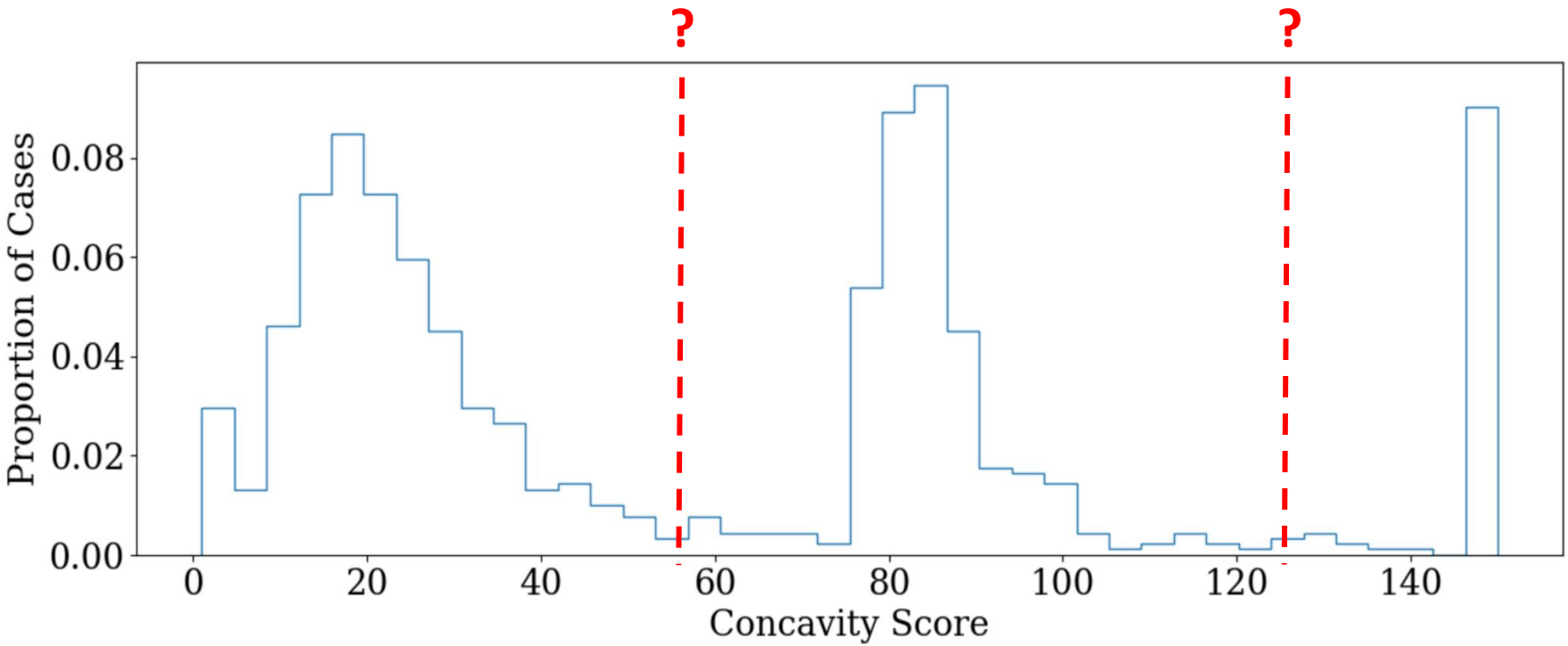
Principal **C**omponent **A**nalysis



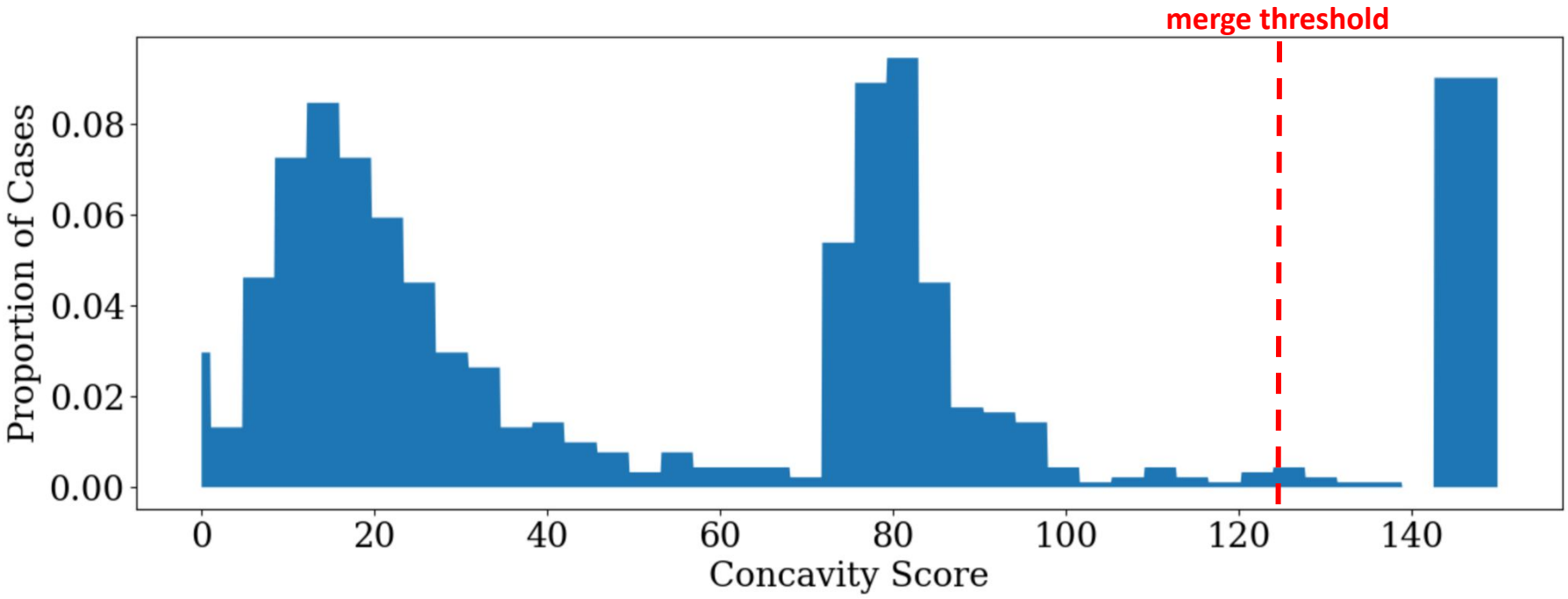
Algorithm Demo



Algorithm Behavior I

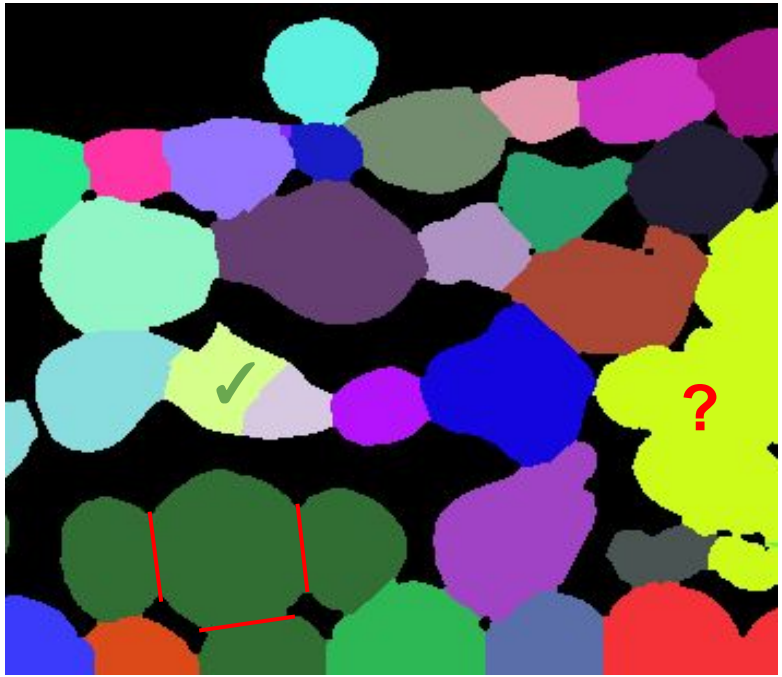


Algorithm Behavior II

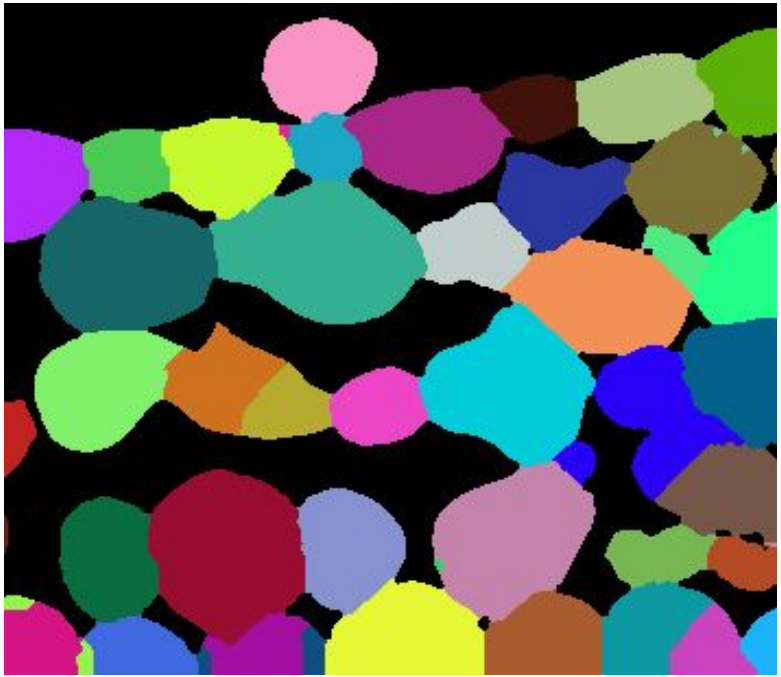


Sample Output (*Camellia yunnanensis*)

previous algorithm

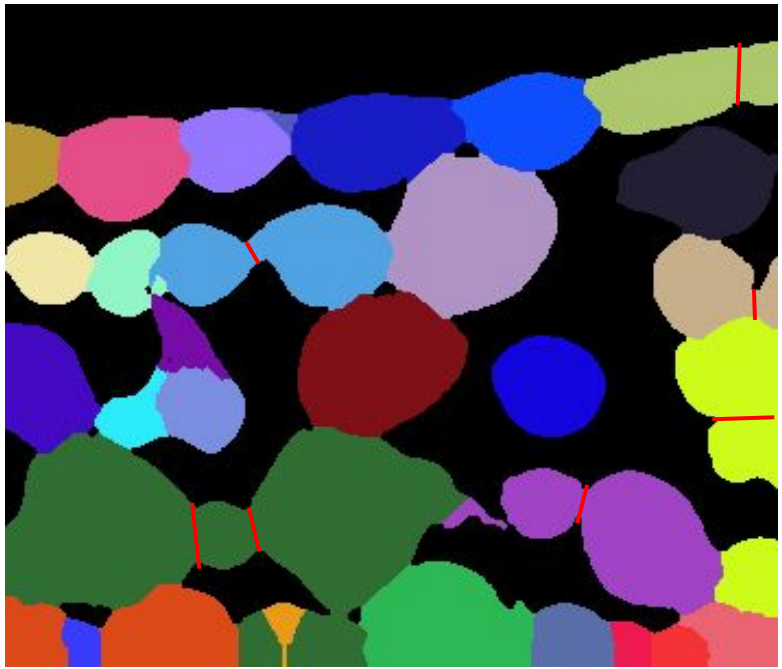


concavity algorithm

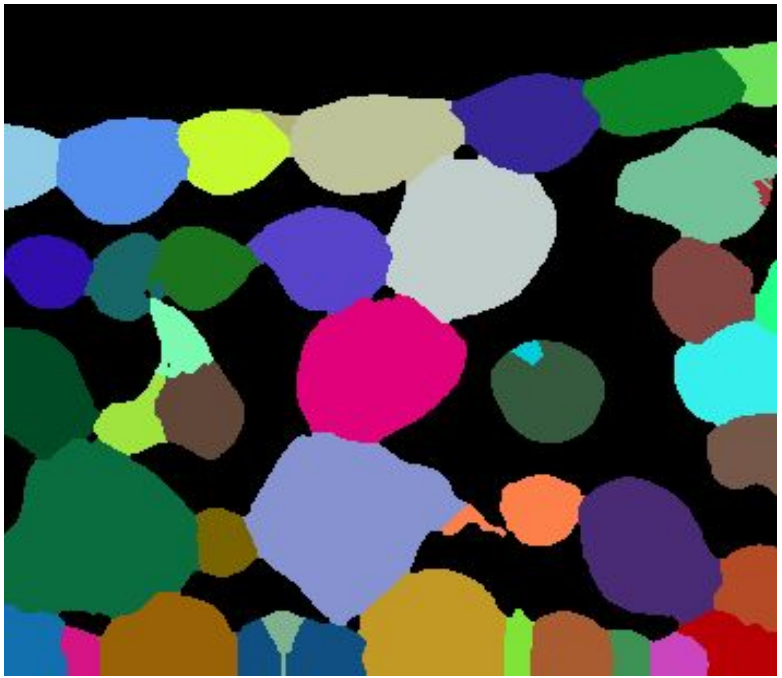


Sample Output (*Camellia yunnanensis*)

previous algorithm

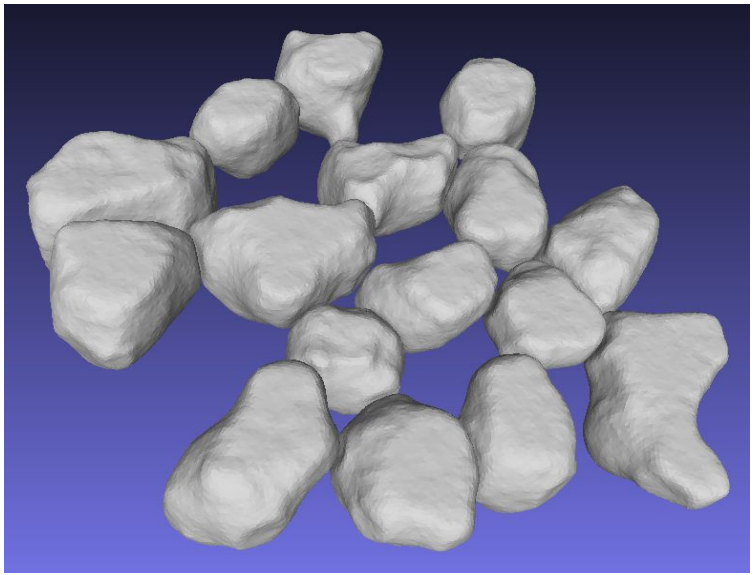


concavity algorithm

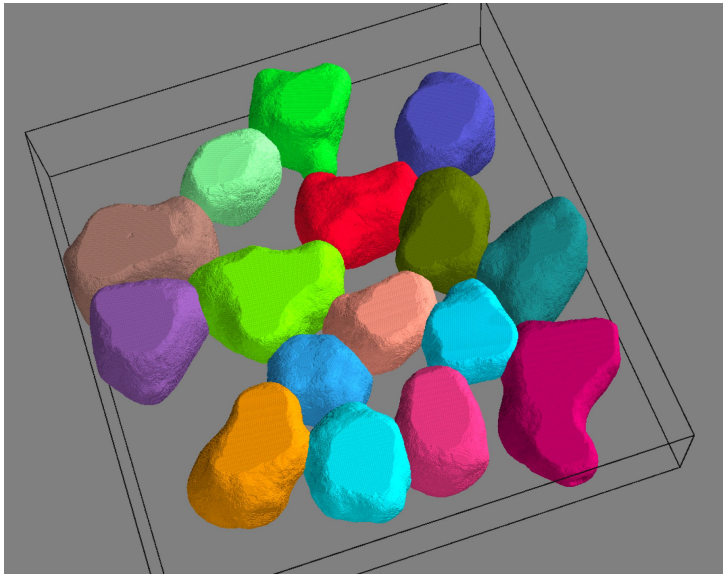


Sample Output (*Arabidopsis thaliana*)

unsegmented binary

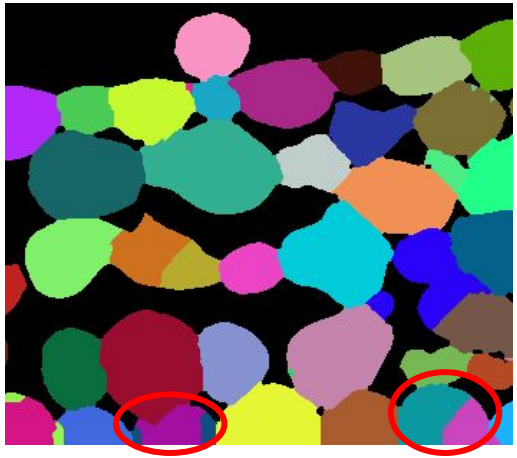


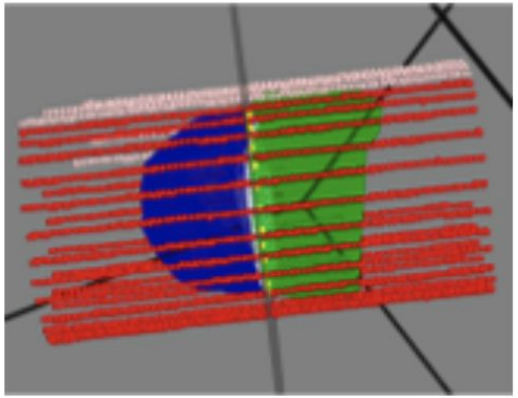
segmented via algorithm



Improvements + Next Steps

- Promising results, but some potential improvements
  - Automating tuning parameters
  - Undersegmentations around border (survivor bias)
  - Viability towards other species/cell shapes





Thanks for Listening