

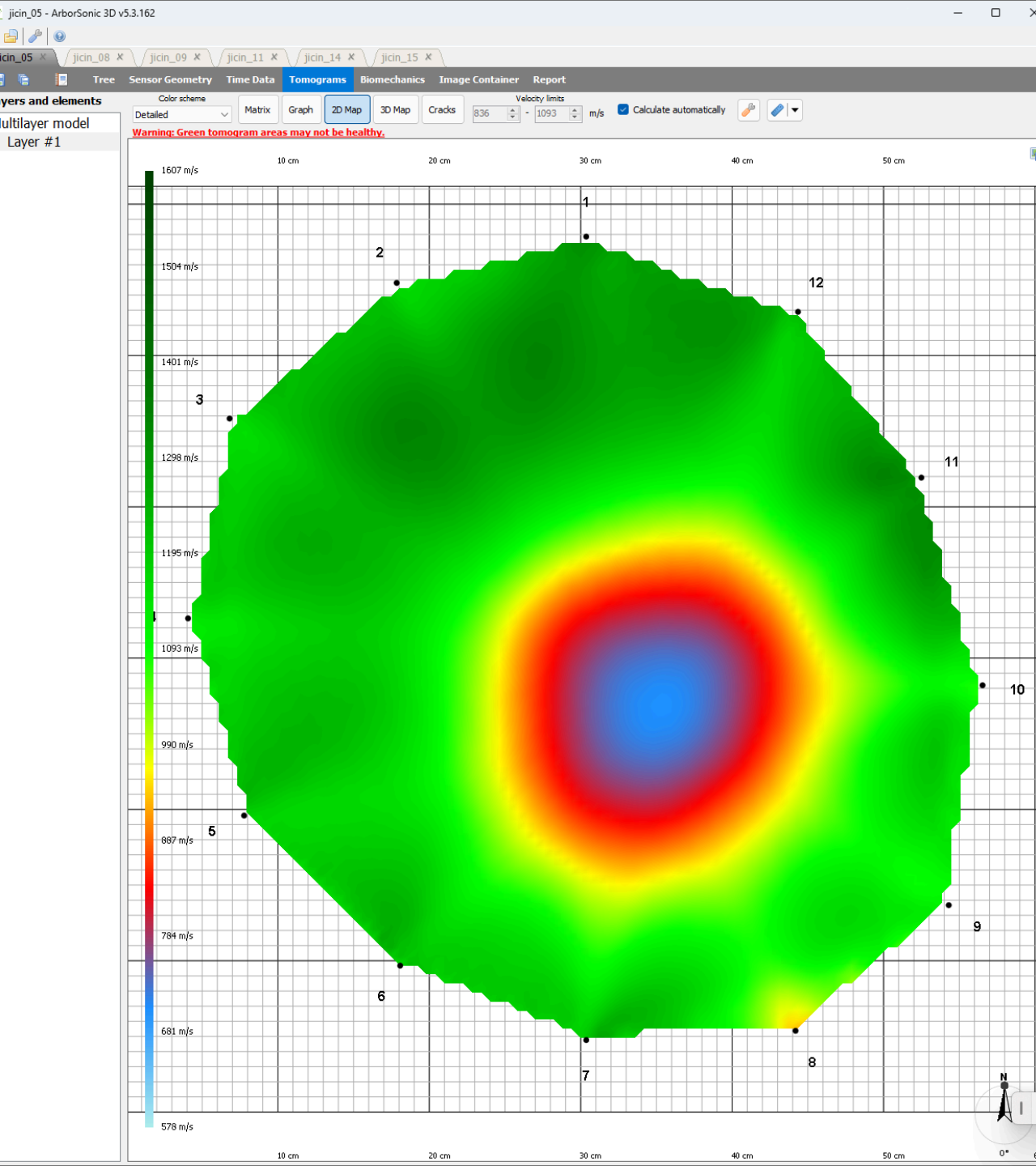
Resistograph meets tomograph

Robert Mařík & Valentino Cristini

Mendel University in Brno

Content of the talk

- Resistograph and tomograph: strengths and limitations
- Combined approach: a Python library to merge data from both devices
- Technical note 1: vibe coding in 2025 (GUI via ChatGPT)
- Technical note 2: code sharing in 2025 (Docker)

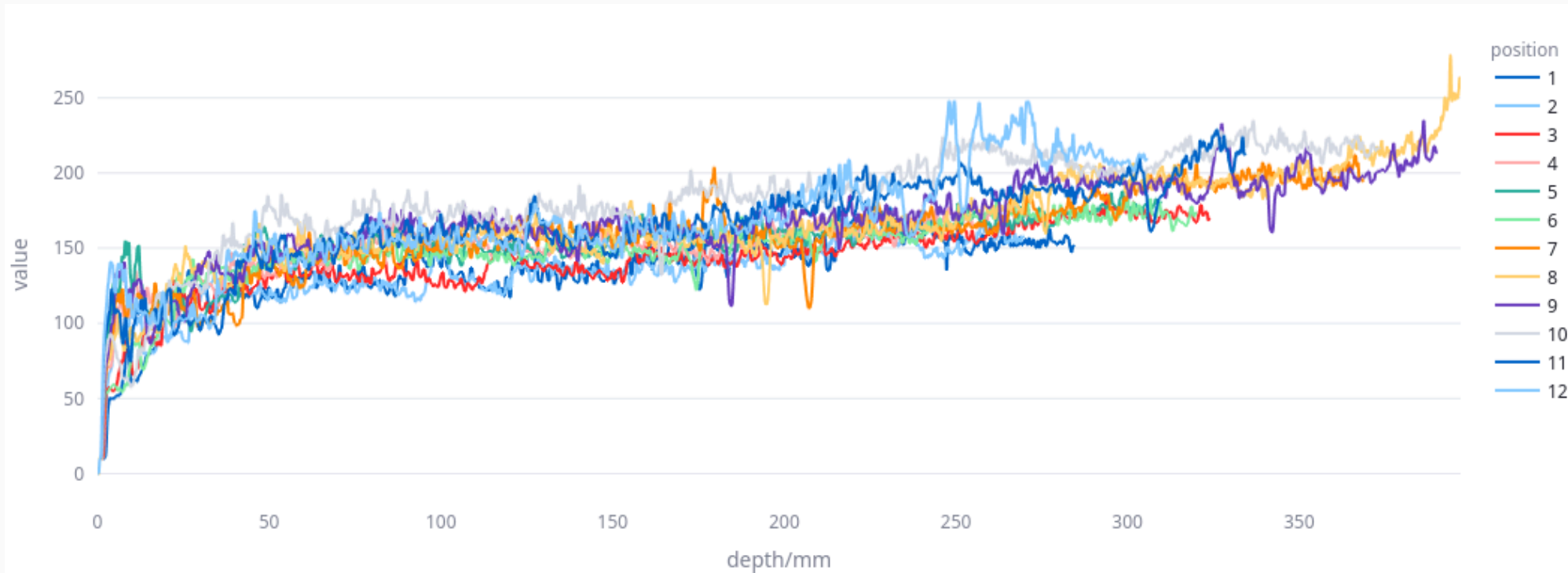


Tomograph

- fast and reliable tool for stem inspection
- global information from the whole cross section
- shows the size and shape of the internal defects
- cracks are reported as cavities

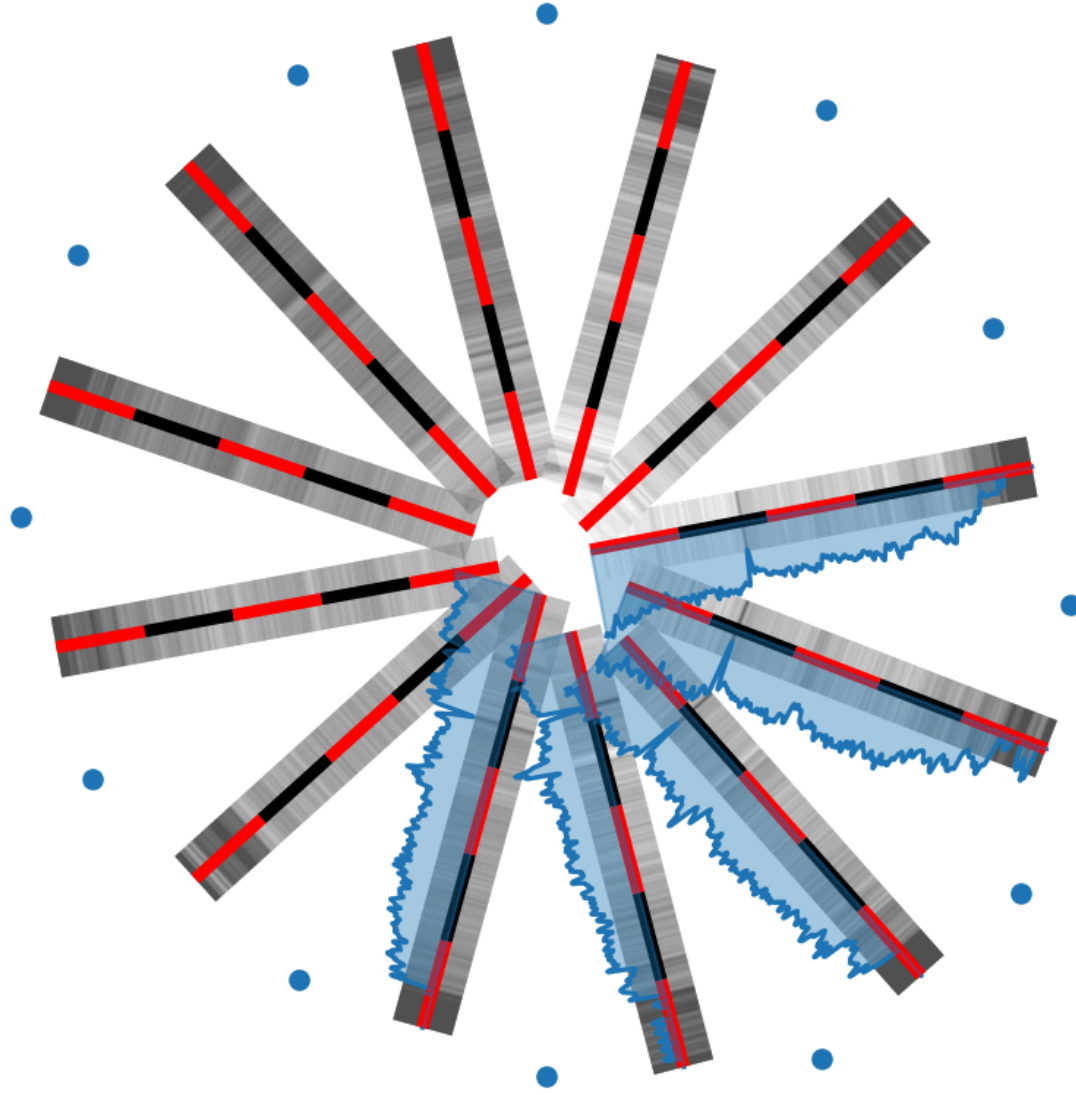
Resistograph

- scans the power required to microdrilling at given speed
- measures mechanical properties of the material
- local information



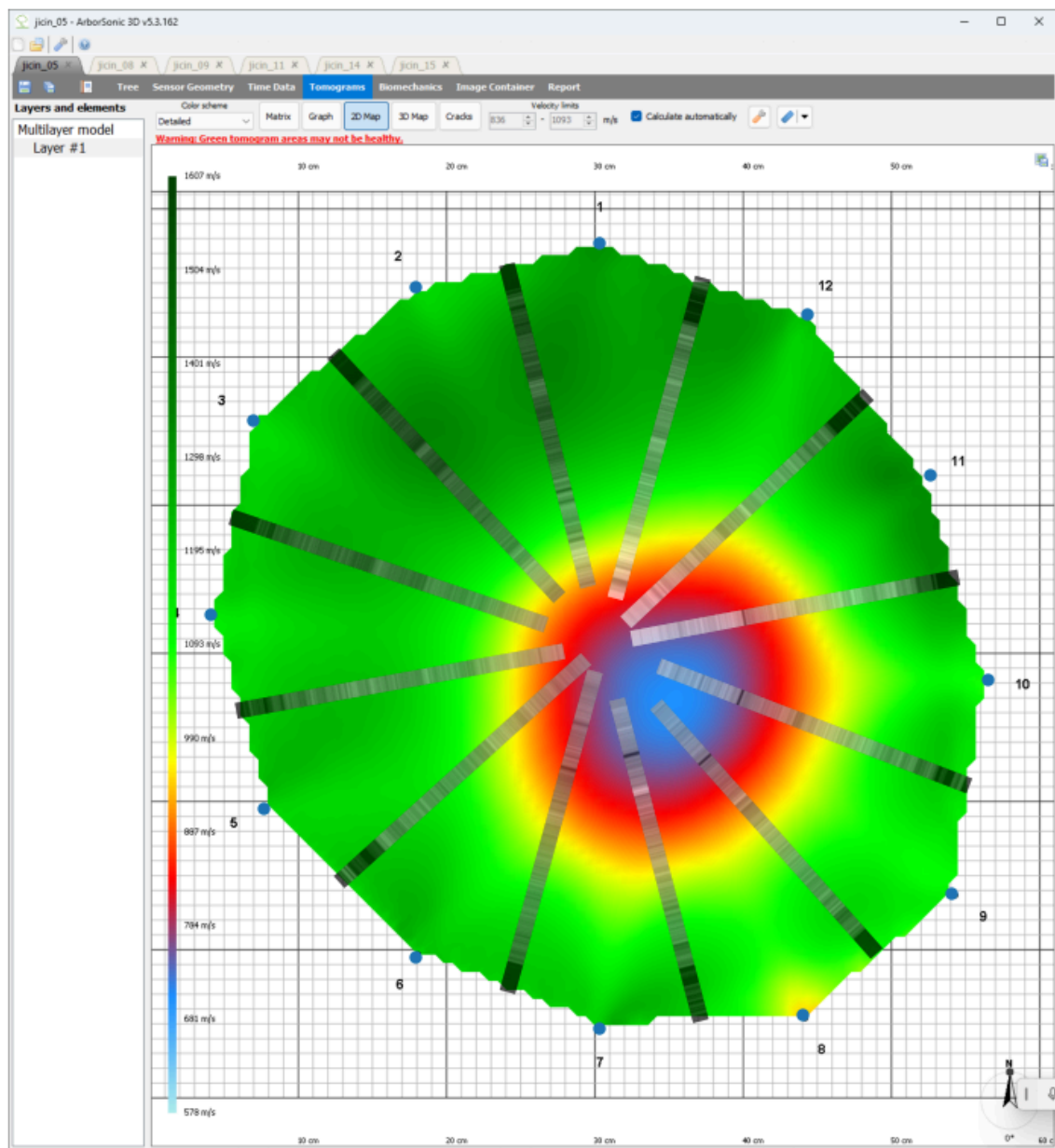
Resistograph Data Visualization in 2D plane

5cm 10cm



Merge data I

- Transform resistograph data to 2D geometry of the cross section
- Visualize the data in the new geometry



Merge data II

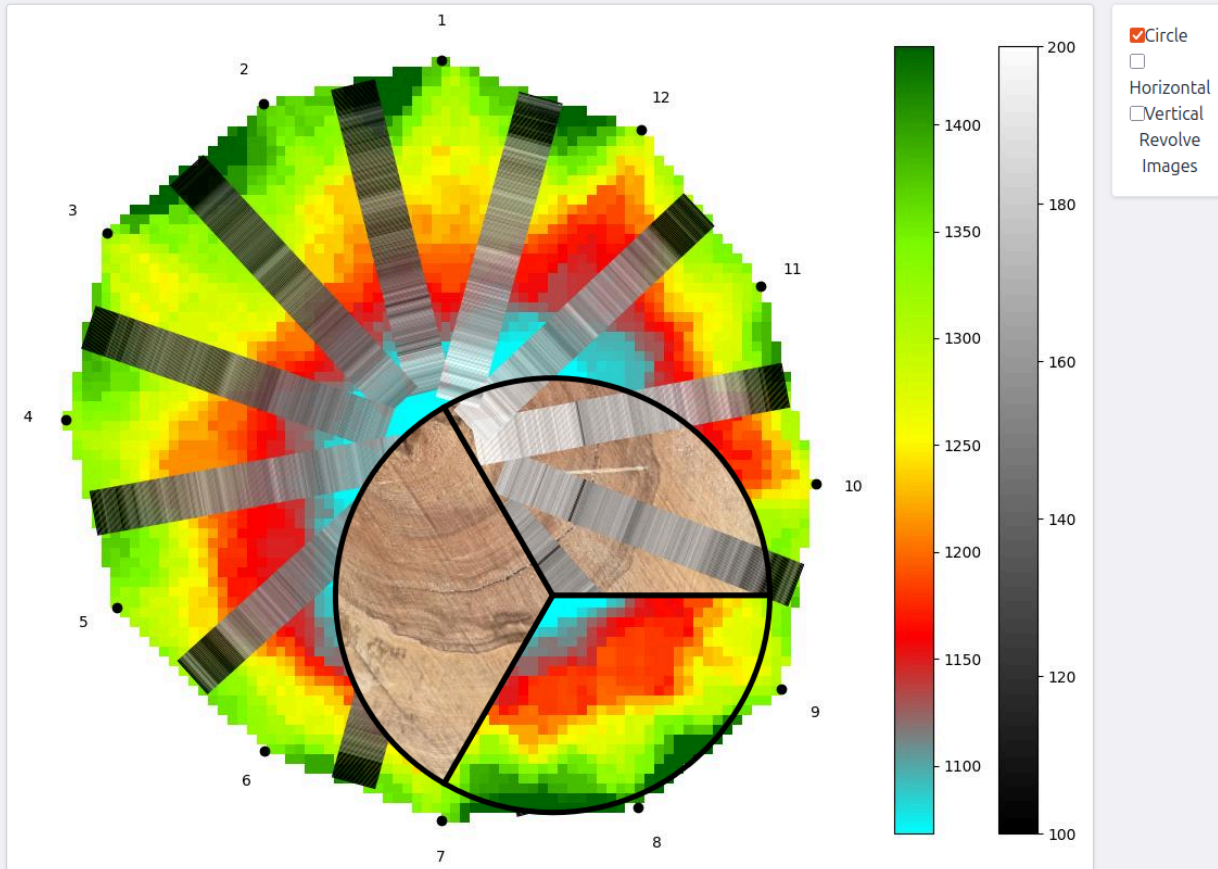
- Merge resistograph data with tomograph data.
- Visualize the merged data.
- Look for short or long decreases in resistograph data. This indicates cracks and cavities, respectively.

When resistograph meets tomograph

The demo of overlays of four images. See [the repository](#) for the code.

- Tomogram
- Tomogram with resistograph data
- Section photo
- Section photo with resistograph data

You can move the mouse over the image to reveal the other layers or click the image to switch layers.



Python library

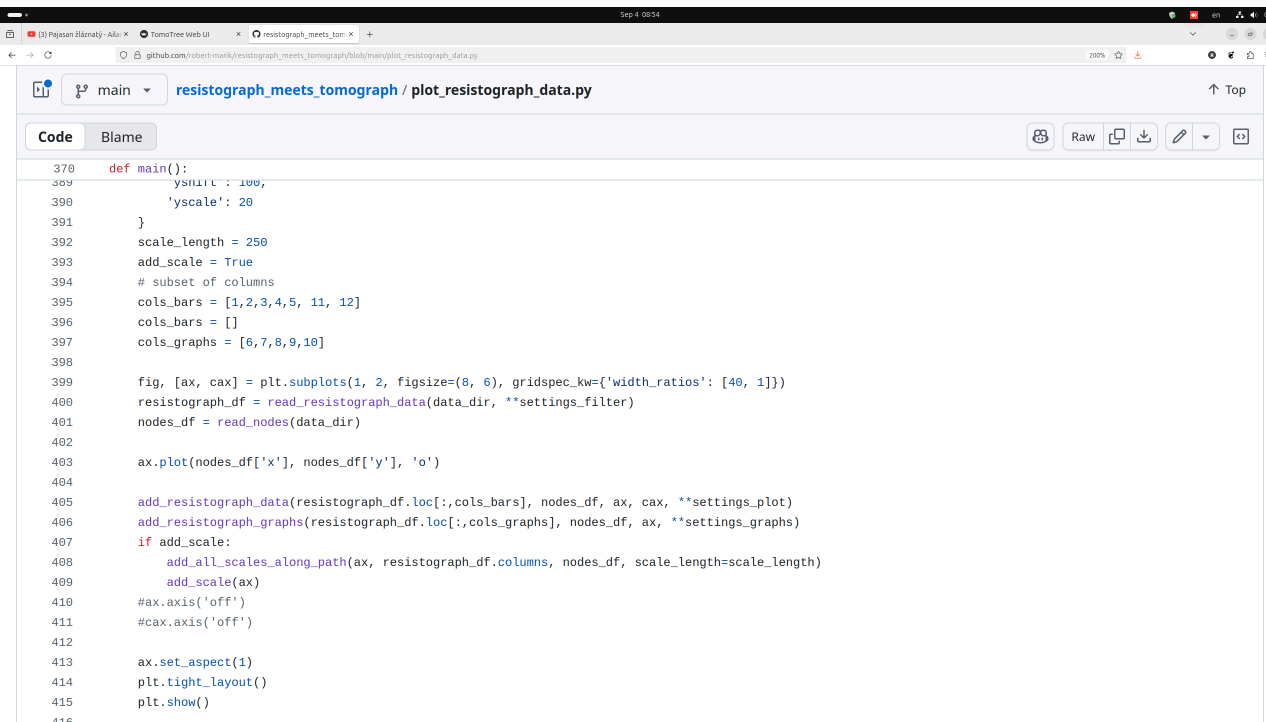
Advantages

- language widely used in scientific data processing
- many libraries for data processing and visualization
- easy to automate, scale, modify, share and reuse
- easy to integrate with other tools

Python library

Limitations

- requires programming skills
- requires installation of Python, Python IDE and libraries
- no GUI



```
370 def main():
371     yscale = 100,
372     'yscale': 20
373 }
374 scale_length = 250
375 add_scale = True
376 # subset of columns
377 cols_bars = [1,2,3,4,5, 11, 12]
378 cols_bars = []
379 cols_graphs = [6,7,8,9,10]
380
381 fig, [ax, cax] = plt.subplots(1, 2, figsize=(8, 6), gridspec_kw={'width_ratios': [40, 1]})
382 resistograph_df = read_resistograph_data(data_dir, **settings_filter)
383 nodes_df = read_nodes(data_dir)
384
385 ax.plot(nodes_df['x'], nodes_df['y'], 'o')
386
387 add_resistograph_data(resistograph_df.loc[:, cols_bars], nodes_df, ax, cax, **settings_plot)
388 add_resistograph_graphs(resistograph_df.loc[:, cols_graphs], nodes_df, ax, **settings_graphs)
389 if add_scale:
390     add_all_scales_along_path(ax, resistograph_df.columns, nodes_df, scale_length=scale_length)
391     add_scale(ax)
392 #ax.axis('off')
393 #cax.axis('off')
394
395 ax.set_aspect(1)
396 plt.tight_layout()
397 plt.show()
398
```

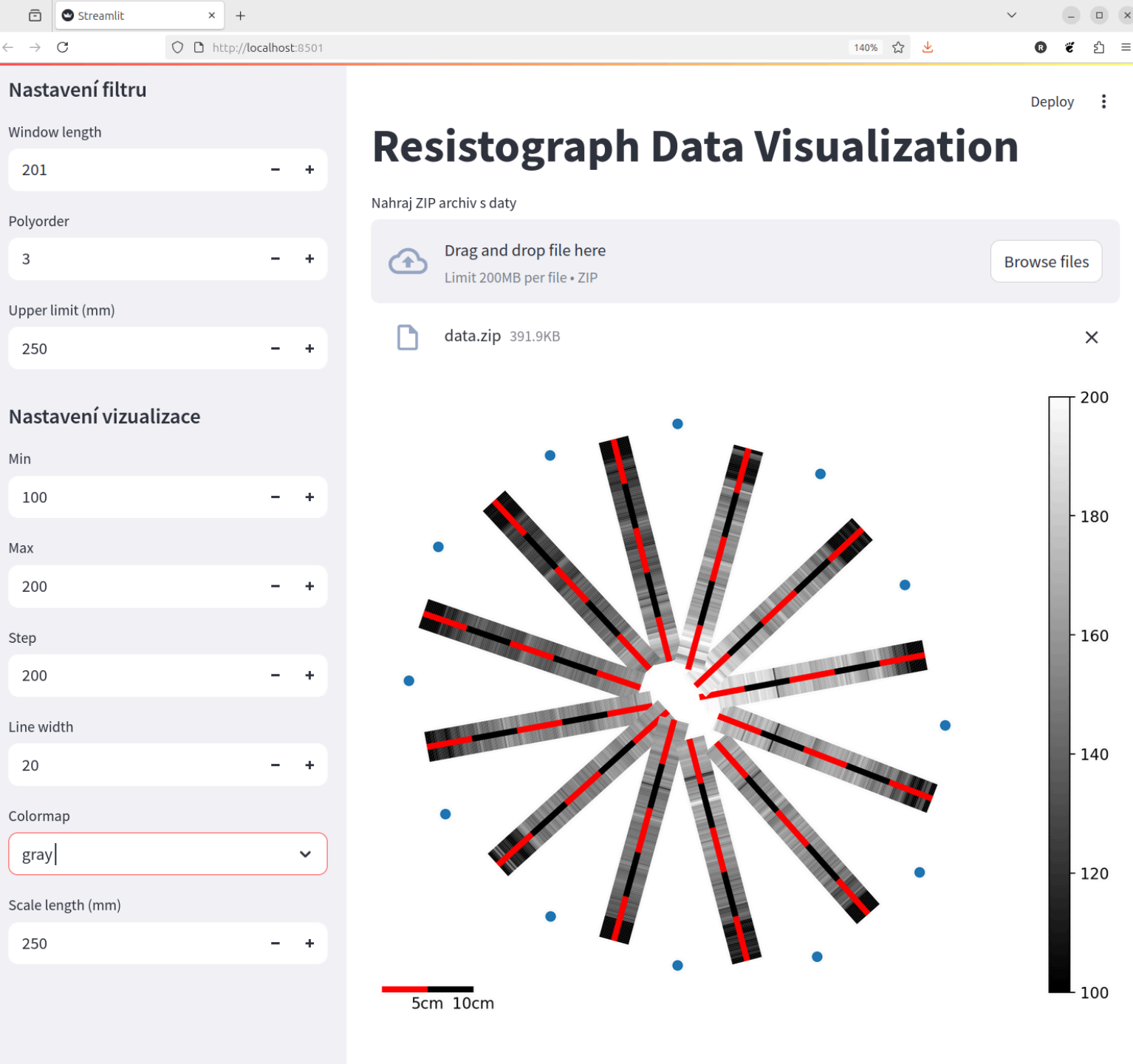

User complaints

- Want to use a GUI. No coding.
- Need simple installation.
- Need easy sharing.

Streamlit

- Python library for building web applications
- easy to use, requires minimal code
- interactive widgets for user input
- real-time updates
- free hosting on Streamlit Cloud
- widely used in data science and machine learning communities
- widely used in industry and academia

Btw: what exactly does it mean "easy to use"?



Vibe coding with ChatGPT

- ChatGPT 5 on August 2025
- web app in 2 prompts

Mam nasledujici knihovnu. Napis streamlit program, který umožni nahrát zazipovany adresar s daty a spusti na nem prikazy odpovidajici main funkci. Vystup se zobrazi.

OK. V levem panelu chci mit moznost menit prednastavene volby.

Docker






A containerization platform

- packages application and its dependencies into a container
- ensures consistency across different environments
- easy to share and deploy
- widely used in industry and academia
- starts a container with a single command in milliseconds

Running dockerized app

```
docker compose up
```

Advantages

-  No Python install
-  No dependency issues
-  Works on Win / Mac / Linux
-  Just clone repo with `Dockerfile` and `docker-compose.yml`
-  First run = minutes, later = ms

Summary

- Resistograph and tomograph are complementary tools for tree stem inspection
- Merging data from both devices provides a more comprehensive understanding of stem condition
- A Python library was developed to facilitate data merging and visualization
- Streamlit can be used to create a user-friendly web application for non-programmers
- Docker ensures easy installation across different systems