

# Modelling the effects of domestication in Wheat through novel computer-vision techniques

Nathan Hughes

April 14, 2018

# Outline

- 1 Description of Topic
- 2 Materials and Image Capture Setup
- 3 Completed Work
- 4 Work in Progress
- 5 Additional Aims
- 6 Thanks

# What is the project?

## Description

The project is aiming to use computational methods to answer biologically significant questions on wheat grain morphology and domestication using  $\mu$ CT images.

## How?

To do this, I will be using:

- Computer vision on 3D image sets
- Statistical analysis and data science
- Scientific theory to create reproducible results

# Research Question:

Is it possible to use  $\mu$ CT imaging to answer questions about Wheat domestication?

*I hope so!*

## Null-Hypothesis

- $H_0$  = Domestication has no effect on the morphometric properties of wheat
- $H_0$  = Ploidy has no effect on the morphometric properties of wheat
- $H_0$  = There is no difference in hulled and non-hulled genotypes

*\*Hulling is a trait associated with domestication*

# Extracted Features

## Features List

The features I am collecting are:

- Length
- Width
- Depth
- Volume
- Surface Area
- Crease Depth
- Crease Volume
- X,Y,Z coordinates of grains

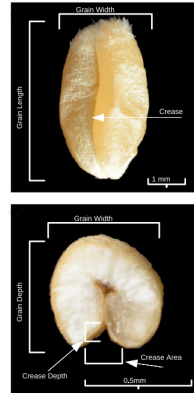


Figure 1: Major features extracted from analysis

## Primary Aims

I am wanting to produce:

- A software library (in Python) which can be used to help analysis of  $\mu$ CT scanned seeds
- A GUI application for researchers to use to auto analyse seeds
- Answers to my hypothesis

# Outline

- 1 Description of Topic
- 2 Materials and Image Capture Setup
- 3 Completed Work
- 4 Work in Progress
- 5 Additional Aims
- 6 Thanks

## Wheat information

I have a wide range of  
Wheat genotypes, these are:

- Ranged between diploid, tetraploid and hexiploid
- 12 total genotypes
- Divided between domestication status



**Figure 2:** Two wheat spikes, showing diversity in Population, Compactum (6N) left, Durum right (4N)



## Capture Methods

In order to generate this data a  $\mu$ CT100 scanner (ScancoMedical, Switzerland) was used Each spike was:

- ~1000 slices (51 slices per stack),
- 125 projections/180° were taken and a
- binning of 6 was used.
- Output images are a 0.2 megapixel (512 x 512) resolution (68.8  $\mu$ /pixel)

# Outline

- 1 Description of Topic
- 2 Materials and Image Capture Setup
- 3 Completed Work**
- 4 Work in Progress
- 5 Additional Aims
- 6 Thanks

## MATLAB Software

Data has been extracted from the  $\mu$ CT images using a method I previously developed at IBERS [Hughes et al. (2017)]. A lot of additions and novel augmentations have been added in order to work with the range of Wheat genotypes.

- This is done via separating non-grain from grain ROIs in the image data
- Measurements are taken on a per grain level
- Measurements are in mm, mm<sup>2</sup> and mm<sup>3</sup> based on type
- Data is exported as CSV files

# Example Data

## Image Analysis Output (Data Truncated for Presentation)

Each 3D image scan results in an individual CSV file, the contents of which is similar to the example below, where each row is an individual grain:

	scanid	length	width	depth	crease_depth	circularity	surface_area	volume
0	C02871	4.47	3.74	3.693	0.65	0.8283	56.1	31.7688
1		5.29	3.80	3.106	0.3	0.923	53.7	31.059
2		4.05	4.49	3.728	0.59	0.848	57.722	32.978
3		4.05	4.03	2.725	0.79	0.779	49.0	25.41
4		3.99	3.84	3.5	0.616	0.785	54.67	30.99
5		4.8	3.86	3.1	0.482	0.913	50.4	28.288
6		3.7	3.82	2.9	0.75	0.816	47.30	24.16
7		3.99	4.46	3.584	0.638	0.85	53.7	30.88

# Improved Watershedding Algorithm

## Algorithm for *Quasi-Euclidean* Distance

$$|x_1 - x_2| + (\sqrt{2} - 1), |x_1 - x_2| > |y_1 - y_2| (\sqrt{2} - 1) |x_1 - x_2|, \text{ otherwise} \quad (1)$$

## Visual Example

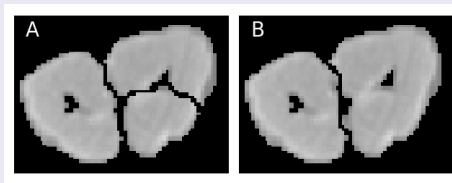


Figure 3: A showing the chessboard method, B improved quasi-euclidean method

# Grain Analyser Library example output - 1

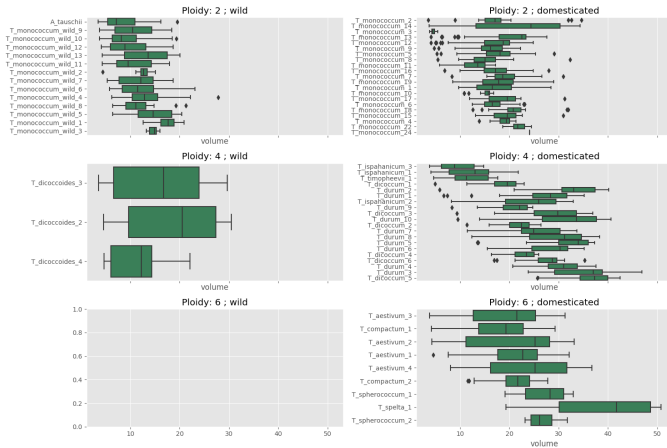


Figure 4: Statistical Python Library Boxplot output

# Grain Analyser Library example output - 2

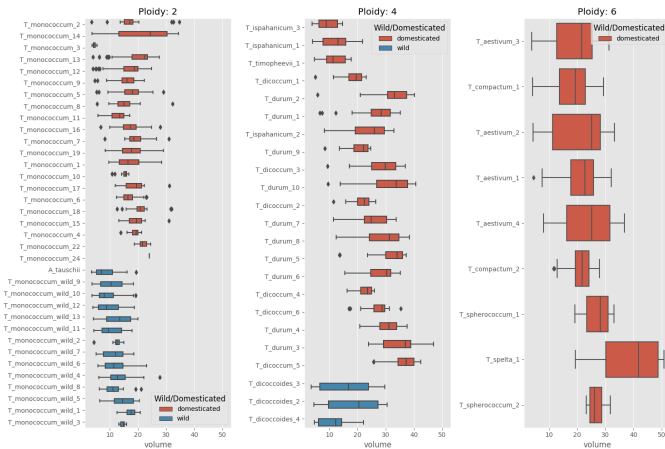


Figure 5: Statistical Python Library Boxplot output style 2

# Grain Analyser Library example output - 3

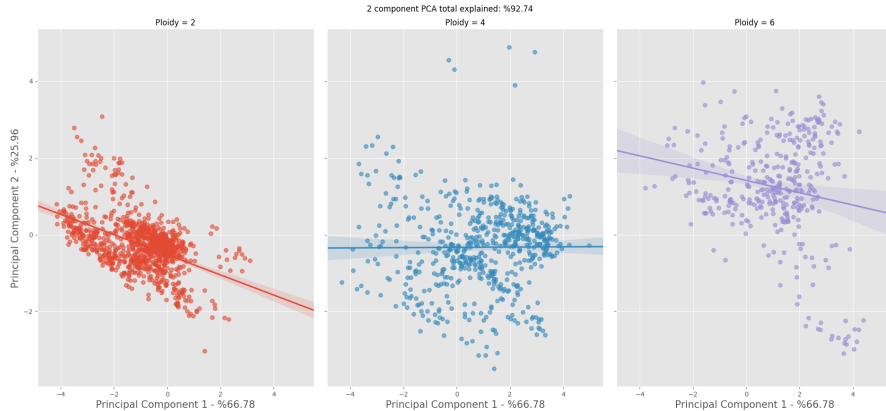


Figure 6: Statistical Python Library PCA output



# Outline

- 1 Description of Topic
- 2 Materials and Image Capture Setup
- 3 Completed Work
- 4 Work in Progress**
- 5 Additional Aims
- 6 Thanks

## Statistical Methods to be implemented

These will be implemented in Python using the library *SciPy* [Jones et al. (2014)]

- ANOVA
- MANOVA
- GLMs
- $\chi^2$  Tests
- PCA
- T/F-Tests

## Graphing Methods to be implemented

All Graphing is being implemented using Matplotlib [Hunter (2007)]

- Box
- Swarm
- Scatter / LM
- Violin
- Correlation Maps
- Frequency/Histograms

# Reproducible Analysis GUI - 1 *WIP*

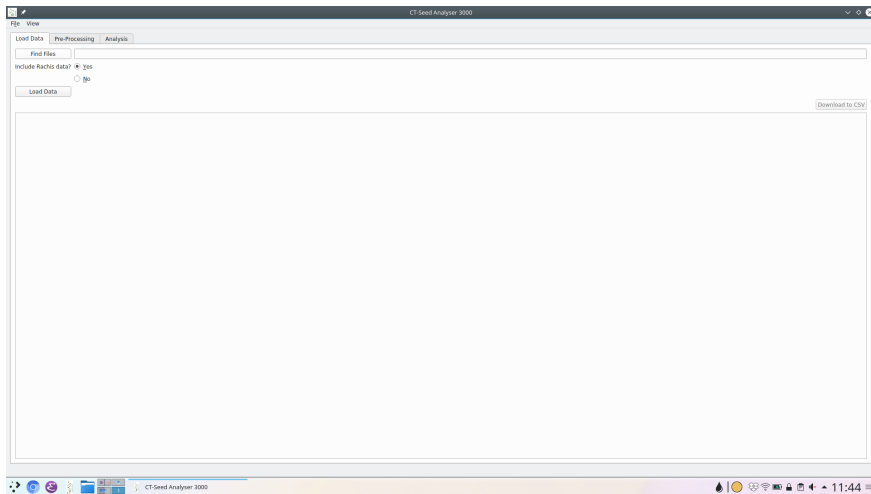


Figure 7: Initial GUI

# Reproducible Analysis GUI - 2 WIP

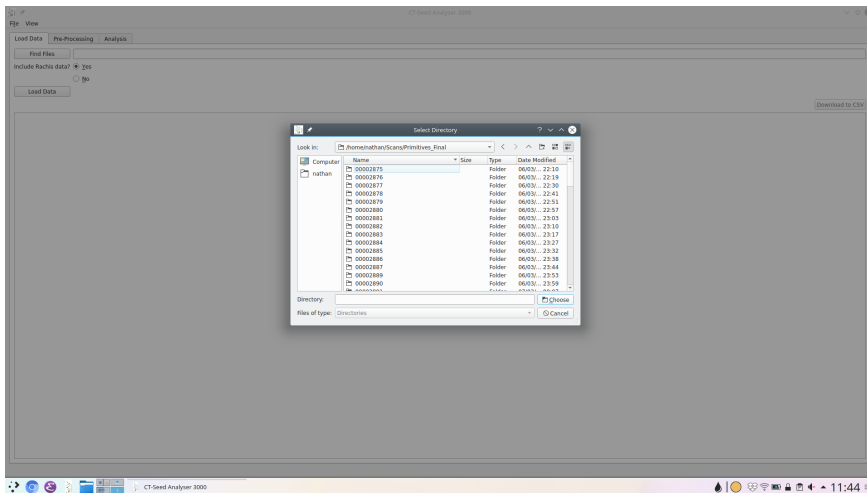


Figure 8: GUI select folders to search for data

# Reproducible Analysis GUI - 3 WIP

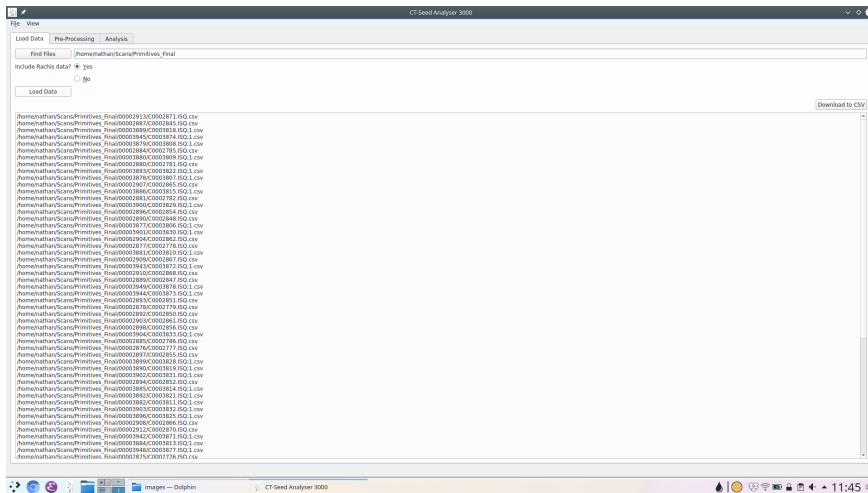


Figure 9: GUI with data loaded successfully

# Reproducible Analysis GUI - 4 *WIP*



Figure 10: GUI dynamically graphing crease\_depth

# Reproducible Analysis GUI - 5 WIP

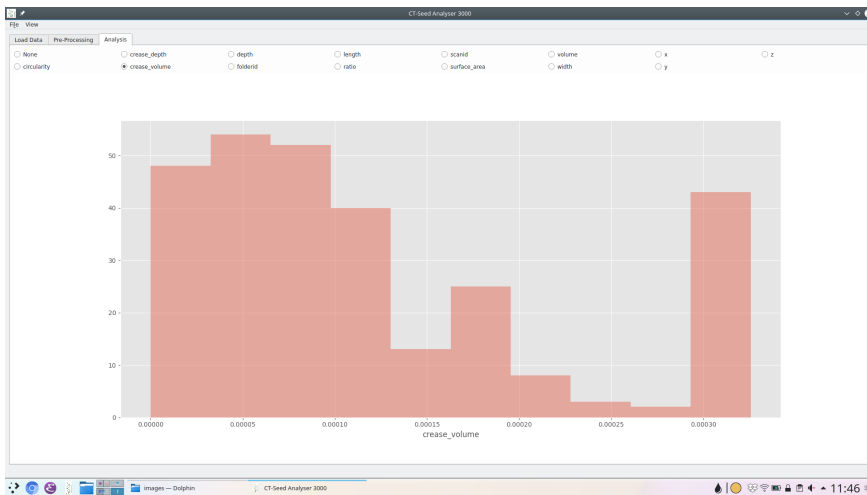


Figure 11: GUI dynamically graphing crease\_volume

# Outline

- 1 Description of Topic
- 2 Materials and Image Capture Setup
- 3 Completed Work
- 4 Work in Progress
- 5 Additional Aims**
- 6 Thanks



# Additional Aims

Not yet started, but would like to implement by end of project

- Automate error finding and identification
- Visually Display Errors in Data
- Write testing suite for software

Ideas for future work

- Test software library on other datasets
- Grow more plants and create more replicates for more robust study

# Outline

- 1 Description of Topic
- 2 Materials and Image Capture Setup
- 3 Completed Work
- 4 Work in Progress
- 5 Additional Aims
- 6 Thanks**

# Thanks to

All these people:

Dr. Wayne Aubrey

Dr. Candida Nibau

Dr. Kim Kenobi

Prof. John Doonan

Dr. Kevin Williams

Everyone at the NPPC

## References

Hughes N, Askew K, Scotson CP, Williams K, Sauze C, Corke F, et al. Non-destructive, high-content analysis of wheat grain traits using X-ray micro computed tomography. *Plant Methods* 2017;13.

Hunter JD. Matplotlib: A 2D graphics environment. *Computing In Science & Engineering* 2007;9(3):90–95.

Jones E, Oliphant T, Peterson P. {SciPy}: open source scientific tools for {Python} 2014;.