**Matthew McGowan**

**Assignment #15**

**Introduction:**

Data science is rapidly becoming the most important discipline for pushing the limits of empirical inquiry in nearly every field of science. Biology is no exception. Computer science is changing nearly every aspect of biological study. Data is being collected by automated robots and drones. Data is being cleaned, processed, and archived with computer programs. Data is being scrutinized with computational methods of testing for consistency and bias. All three of these; collection, analysis, and reflection are crucial for the generation of reproducible, reliable, and applicable scientific experimentation. Therefore, I would like to discuss a paper for each of these processes.

**DATA GENERATION**

**Topics:** Python code to address research needs, Python visualization

**Citation:** Khramtsova, Ekaterina A., and Barbara E. Stranger. “Assocplots: a Python Package for Static and Interactive Visualization of Multiple-Group GWAS Results.” *Bioinformatics*, 2016, doi:10.1093/bioinformatics/btw641.

**Description:**

The visual display of quantitative information is a topic that is often underappreciated when considering the most important advances in scientific technology. When facing competition such as the DNA sequencer, the Large Hadron Collider, and the electron microscope, emphasis often focuses on the aspect of data collection. Indeed, all three of these examples are primarily involved in this process, which is accessing the new frontiers of data available. However, I believe it is equally as important to emphasize the reimagining of the old and remember to value new technologies that allow scientists to revisit or reimagine the mountains of data already available to us. Therefore, I have decided to start with a paper that demonstrates this. An incredibly important way we access and process data to increase our understanding of underlying processes is through the creation of charts and graphs. In our course, we explored the basic capabilities of using programming to better access and visualize our data through the use of multiple plotting packages available for Python. This paper directly applies to these lessons. In it, the authors use Python and the interactive visualization library bokeh to create more accessible ways of generating high quality manhattan plots and quantile-quantile plots that can be directly use in Jupyter notebooks. Beyond simply generating static plots however, this package also includes a dynamic interactive feature where hovering the mouse over a SNP will reveal further information regarding this SNP. The graph can also be zoomed in and out as well as shared in a form that maintains the underlying data required for interactivity. One direct application I think this tool would be useful for is the publication of ‘living papers’ that can be published online in a way that lets readers interactively interact with the data and figures as well as leverages social networking technologies for readers to interact and document the scholarly debates, observations, and experiences with the paper.

**DATA PROCESSING**

**Topics:** Python code to address research needs, uses Machine learning methods

**Citation:** Gehan, Malia A, et al. “PlantCV v2.0: Image Analysis Software for High-Throughput Plant Phenotyping.” *PeerJ*, 1 Dec. 2017, doi:10.7717/peerj.4088.

**Description:**

Alongside the need for better ways to scientists to visualize data, there is also an increasing need for automated ways of processing raw image data into higher level forms of data. This need is most pronounced in the field of high-throughput plant phenotyping. Because sequencing technologies have caused the price to drastically fall (even to the extent of beating Moore’s law), many of our analytical techniques are running into a high ‘m’, low ‘n’ problem. This relates to the very important matrix used in GWAS and genomic prediction modeling where n is the number samples and therefore the number of phenotypic observations and m is the number of genetic markers available for testing. High-throughput SNP chips, whole genome sequencing, and RNA-seq have allowed the number of markers to increase into the millions of markers. However, the cost to obtain the very important phenotypic trait values which we are testing has remained relatively stable leading to a phenotyping bottleneck. Therefore, it is important that new technologies are leveraged in a way that high-throughput phenotyping can be done in a way that reduces cost and labor while increasing the usability and range of application. This paper satisfies this need by describing a python package that was built specifically for processing plant imagery for scientific applications. Featuring open-source source software built on OpenCV, NumPy and MatPlotLib, this paper closely relates to the skills learned this semester. By integrating with Jupyter, analytical pipelines are easier to document and share. This software highlights multiple important design aspects emphasized in class such as parallel functionality, managing raw data through the use of SQL database consolidation, and GitHub based version control. Lastly, the most recent version uses machine learning (naïve Bayes classifier) for image segmentation.

**DATA VALIDATION**

**Topics:** Data integrity, high-throughput analysis,

**Citation:** Leek, Jeffrey T., et al. “Tackling the Widespread and Critical Impact of Batch Effects in High-Throughput Data.” *Nature Reviews Genetics*, vol. 11, no. 10, 2010, pp. 733–739., doi:10.1038/nrg2825.

**Description:**

Lastly, I feel it fitting to end with a paper that describes experimental and computational approaches to address the influence of batch effects in high-throughput data. This is because most scientific disciplines (especially Biology) are in the midst of a self-reflective crisis where scientific research is losing credibility due the inability to reproduce many findings reported in peer reviewed journals. Because modern scientific disciplines are founded on the ability to explain and predict events in the objective universe, a lack of trust in the reliability and reproducibility of scientific findings erodes the foundations of the discipline. A significant portion of this class focused on how to process data in ways that can improve consistency and integrity. Therefore, it is important that improved methods in the fields of computer science and statistics are continuously adopted in biology to help address the problems of data and analytical integrity. This paper does this by describing the problem of batch effects in large datasets and reviews useful approaches for mitigating these effects. In brief, batch effects are sub-groups of measurements in a dataset that behave differently across experimental conditions and are unrelated to the scientific variables in the study. For example, these could be due to a different technician handling the samples, a different machine for measuring the values, and different reagents. While the presence of these effects are not unique to high-throughput data, they are often impossible to detect in low-dimensional datasets where there is not enough data to test for them. However, in the context of high-throughput data, it is possible to test and include these effects during experimental design or data analysis. The authors successfully describe this issue through a concise and structured argument. First, they demonstrate the ubiquitous existence of batch effects by using examples from multiple different public databases and domains of interest. Second, they describe multiple examples for how failure to address batch effects led researchers to a misinformed conclusion. Third, they describe multiple ways that experiments can be designed to allow easier testing for batch effects as well as statistical methods for analyzing those effects.

**CONCLUSIONS**

The skills we learned in class directly relate to the three aspects of data science research described. The first focuses on data visualization. The authors build a tool in Python that takes the general visualization packages Bokeh and matplotlib and modify their functionality to specifically focus on data generated from GWAS analysis. These tools were taught in class, and should I ever need to use this package, I could easily extend more functionality customized to my needs. The second paper focuses on data generation. While we did not cover image analysis in class, we did review methods of manipulating data and processing high dimensional raw data into lower dimensional outputs. Instead of processing images into masked/segmented pixel objects, we processed raw sequence data into SNPs. The last paper describes a type of data bias often present in high-throughput data and explains several methods that can be used to mitigate this bias. These methods could be implemented to function alongside many of the other statistical tools used in class (such as the Scikit Learn module) to improve data quality like scaling, normalizing, and error testing. Despite all three of these papers dealing with completely different aspects of data science, all of them can be implemented using skills learned in class.