

COAL: Coal and Open-pit surface mining impacts on American Lands

Taylor Alexander Brown, Heidi Ann Clayton, and Xiaomei Wang

Group 18

CS 461: Senior Capstone Fall 2016

Oregon State University

Abstract

Mining is known to cause environmental degradation, but software tools to identify mining impacts are lacking. Researchers studying this problem possess large imaging spectroscopy and environmental quality data sets as well as high-performance cloud-computing resources. This project provides a suite of algorithms using these data and resources to identify signatures of mining and correlate them with environmental impacts over time.



CONTENTS

1	Introduction	4
2	Technologies	4
2.1	Feature extraction in AVIRIS data	4
2.1.1	Options	4
2.1.2	Goals	4
2.1.3	Criteria	4
2.1.4	Comparison	4
2.1.5	Discussion	5
2.1.6	Selection	5
2.2	Classification of AVIRIS data	5
2.2.1	Options	5
2.2.2	Goals	5
2.2.3	Criteria	5
2.2.4	Comparison	5
2.2.5	Discussion	5
2.2.6	Selection	6
2.3	Identify Mining	6
2.3.1	Options	6
2.3.2	Goals	6
2.3.3	Criteria	6
2.3.4	Comparison	6
2.3.5	Discussion	6
2.3.6	Selection	7
2.4	Preprocessing Data to Correlate Mining with Environmental Impact	7
2.4.1	Options	7
2.4.2	Goals	7
2.4.3	Criteria	7
2.4.4	Comparison	8
2.4.5	Discussion	8
2.4.6	Selection	8
2.5	Feature Extraction to Correlate Mining with Environmental Impact	8
2.5.1	Options	8
2.5.2	Goals	8
2.5.3	Criteria	9
2.5.4	Comparison	9
2.5.5	Discussion	9
2.5.6	Selection	9
2.6	Rank and Document Changes Over Time	9

2.6.1	Options	9
2.6.2	Goals	10
2.6.3	Criteria	10
2.6.4	Comparison	10
2.6.5	Discussion	10
2.6.6	Selection	10
2.7	API Documentation	10
2.7.1	Options 1, 2, and 3	10
2.7.2	Goals	11
2.7.3	Criteria	11
2.7.4	Comparison	11
2.7.5	Discussion	11
2.7.6	Selection	11
2.8	Static site generator	11
2.8.1	Options	11
2.8.2	Goals	12
2.8.3	Criteria	12
2.8.4	Comparison	12
2.8.5	Discussion	12
2.8.6	Selection	12
2.9	Front-end framework	12
2.9.1	Options	12
2.9.2	Goals	12
2.9.3	Criteria	12
2.9.4	Comparison	13
2.9.5	Discussion	13
2.9.6	Selection	13
3	Conclusion	13
4	Revisions	13
4.1	Feature extraction in AVIRIS data	13
4.2	Classification of AVIRIS data	13
4.3	Identify Mining	13
4.4	Preprocessing Data to Correlate Mining with Environmental Impact	14
4.5	Feature Extraction to Correlate Mining with Environmental Impact	14
4.6	Rank and Document Changes Over Time	14
4.7	API Documentation	14
4.8	Static site generator	14
4.9	Front-end framework	14
	References	14

1 INTRODUCTION

This document is a review of technology and literature relevant to the COAL project. The technologies discussed in this paper correspond to tasks or subtasks identified in our requirements document [1]:

- Develop software to process imagery
- Provide API documentation for developers
- Provide a website that serves as the project homepage

In the past weeks our team has been working with each other and the client to become familiar with the data and the libraries we have selected so we have a more advanced understanding than when we started. However, some material is subject to change as our understanding improves.

Sections 2.1 and 2.2 were authored by Heidi who will take responsibility for the algorithms to identify and classify minerals. Section 2.3 was authored by Taylor on behalf of Xiaomei, who will take responsibility for the algorithms to identify regions of mining activity. Sections 2.4, 2.5, and 2.6 were authored by Taylor who will take responsibility for the algorithms to correlate mining with environmental impact and to rank and document changes over time. Section 2.7 was written by Heidi who will take responsibility for the API documentation. Sections 2.8 and 2.9 were written by Xiaomei who will take responsibility for implementing the backend and the frontend of the project homepage.

2 TECHNOLOGIES

2.1 Feature extraction in AVIRIS data

2.1.1 Options

- 1) Spectral Python [2]
- 2) HyperSpy [3]
- 3) MATLAB Hyperspectral Toolbox [4]

2.1.2 Goals

In order to properly analyze the AVIRIS images, COAL needs a way to identify regions in an image that indicate a particular mineral or other pattern. We need to have a tool that can process hyperspectral imagery in order to recognize the spectral signatures of different minerals and land surfaces so that AVIRIS images can be classified.

2.1.3 Criteria

The main criteria being evaluated are if the software library is currently maintained and developed, if the software is compatible with neural networks, and what language is used in the software.

2.1.4 Comparison

	Spectral Python	HyperSpy	MATLAB Hyperspectral Toolbox
Currently maintained	✓	✓	×
Has neural network functionality	✓	×	×
Language	Python	Python	MATLAB

This table shows an overview of the three choices of software libraries against the criteria of being currently maintained, having neural network functionality, and the language that is used in the implementation.

2.1.5 Discussion

At first glance, it is fairly apparent that the MATLAB Hyperspectral Toolbox does not meet the criteria well at all. It is not currently maintained or developed, with the last update to its GitHub repository being from 2015. There is also a note in the README file that it has been abandoned. MATLAB Hyperspectral Toolbox also does not have neural network functionality, which we would like to use to process the spectral signatures as it has been shown to work well with AVIRIS data [5]. As the name implies, it is also implemented in MATLAB. This is not something that would absolutely stop the team from using a software library, but as the team is more familiar with Python, it is preferred. HyperSpy is a better fit, as it is currently maintained, however it does not have neural network functionality built into the library. Spectral Python is the only option that satisfies all criteria.

2.1.6 Selection

We will be using Spectral Python.

2.2 Classification of AVIRIS data

2.2.1 Options

- 1) Spectral Python
- 2) PyLearn2 [6]
- 3) TensorFlow [7]

2.2.2 Goals

Before any further data analysis can take place, classifying the AVIRIS data into classes based on mineral concentrations and categories such as vegetation and bodies of water will be needed in order to have variables to correlate (or not) with the environmental effects of mining.

2.2.3 Criteria

The main crite

2.2.4 Comparison

	Spectral Python	PyLearn2	TensorFlow
Currently maintained	✓	×	✓
Compatible with feature extraction technology	✓	×	×

This table shows an overview of the three choices of software to classify the data from the trained neural network generated in the feature extraction stage. The only software that meets all of the requirements is Spectral Python.

2.2.5 Discussion

Feature extraction and classification go hand-in-hand, so unsurprisingly, Spectral Python already has the ability to classify hyperspectral imagery based on the feature extraction stage. It is convenient and, as mentioned above, it is currently maintained so that any bugs or new features can be handled in a timely manner. PyLearn2 is not currently maintained. PyLearn2 and TensorFlow are also not designed to be used with Spectral Python, so it would take a lot of time to build that functionality if we decided on either technology. Overall, Spectral Python is the most logical and convenient option to use for mineral classification.

2.2.6 Selection

We will be using Spectral Python for the classification of AVIRIS data.

2.3 Identify Mining

To identify regions of mining activity, the mineral identification and classification data is further processed and possibly combined with geographic information about geology and mines.

2.3.1 Options

Three approaches to identify mining are:

- 1) mineral classification, using patterns of mineral deposits [8], [9];
- 2) GIS comparison, using geographic information such as mining permit boundaries [10] or geologic maps [9]; and
- 3) a combination of mineral classification and GIS comparison, using both sources of data.

2.3.2 Goals

The goal of the mining identification step is to accurately locate regions in which mining is taking place in a form that is suitable for both human inspection and further processing. It uses the output of the mineral identification step which located regions containing minerals associated with mining. Because these minerals may be derived from both natural and artificial processes [9], additional logic or data may be necessary to distinguish mines from preexisting geology. Because existing geographic data may be unreliable or incomplete, some combination of mineral classification with geographic information may produce more accurate results.

2.3.3 Criteria

Criteria being evaluated include the time needed to research and develop each method, the computational cost of processing the data, and the accuracy of the results.

2.3.4 Comparison

	Research and Development Time	Computational Cost	Accuracy
Mineral Classification	Low to Medium	Low to Medium	Medium
GIS Comparison	High	Medium	Medium
Combination	High	Medium to High	High

This table compares each option based on the criteria. Because the options are general approaches to algorithm design and not specific libraries, it was necessary to estimate the costs and benefits.

2.3.5 Discussion

Mineral classification by itself can be a sophisticated method of identifying traces of mining activity. Minerals with low natural concentrations can be used to identify mining activity, as jarosite was used to map active mining operations in the Ray Mine in Arizona [9]. In this case, simply mapping the presence of a mineral may be sufficient to locate mines. In other cases, minerals associated with mining cannot be detected by imaging spectroscopy, such as pyrite in the California Gulch Superfund Site [8]. The solution in this case was to search for patterns of secondary minerals weathered from pyrite that are detectable by the sensors. Therefore the mineral classification method can be simple or complex depending on the chemistry of the minerals of interest and the logic required to find them.

GIS comparison can also provide useful insights for identifying mines. One approach that has been used to successfully visualize mining activity is to overlay permit boundaries on top of classified spectroscopy data [10]. This provided a visual comparison of known mines and their spectroscopic signatures. Another approach is to compare geologic maps with the observed minerals and look for discrepancies [9]. This allowed naturally-occurring deposits to be distinguished from artificial ones. The comparisons between processed data and known geographic information could be made programmatically or simply displayed for human inspection.

A combination of mineral classification with GIS comparison would provide the most extensive analysis but would require the most research and development time and computational cost. Using both methods would require design choices such as whether to automate or display each result. For example, GIS comparison of geologic maps with observed minerals could be used to automatically eliminate deposits that are thought to be naturally-occurring; alternatively, both the geologic maps and the observations could be combined so that more data is displayed and less is lost. Similar choices would be necessary for mineral pattern detection or permit boundary mapping. Combining multiple data sources would be challenging and could make the results more or less accurate depending on the circumstances.

2.3.6 Selection

Based on these criteria, it is recommended to pursue the mineral classification methods first in order to locate regions with mineral deposits that indicate mining. If this can be completed successfully, then evolving a combination of mineral classification and GIS comparison methods would provide an opportunity to improve the analysis.

2.4 Preprocessing Data to Correlate Mining with Environmental Impact

To correlate the regions of mining activity with the regions of environmental degradation, the data sources must be preprocessed to be compatible.

2.4.1 Options

Three approaches to make the mining and environmental data compatible are:

- 1) conversion of mining activity data to GIS,
- 2) conversion of environmental impact data from GIS, and
- 3) conversion of both data sources to a third format.

2.4.2 Goals

The goal of the preprocessing step is to transform the input data, assumed here to be spectroscopic data and geographic information, such that the following feature extraction step can find meaningful relationships. Because this is an intermediate step, no data visualization is generated so the formats may be only machine-readable.

2.4.3 Criteria

Criteria being evaluated include the time needed research and develop each method, the compatibility of the format with existing standards, and the retention or loss of precision produced by each.

2.4.4 Comparison

	Research and Development Time	Standard	Precision
To GIS	High	Yes	Medium
From GIS	Medium	No	Medium to High
Third Format	Medium to High	Maybe	Medium to High

This table compares each of the possible data formats with estimates for each of the criteria.

2.4.5 Discussion

Transforming the spectroscopic data to GIS for combination with the environmental data would produce a standard intermediate format. This format would not necessarily be best suited for the following feature extraction step, and it is speculated that data would inevitably be lost. Furthermore, it would require significant research and development time to transform the processed imagery into a geographical format.

Converting the environmental impact data from GIS so that it is compatible with the processed imagery would generate a nonstandard format, but one that can arguably be better tailored to the application. Research and development time is estimated to be lower for this approach.

Converting both the processed imagery and the geographic information to a third format would provide a choice about compatibility and suitability for further processing. The research and development time for this approach is estimated to be higher.

2.4.6 Selection

As we began to design and implement this step, it was concluded that the most straightforward means of combining geographic information and hyperspectral imagery is to convert the GIS data to the same raster format as the imagery. This will facilitate user-defined correlation functions which operate on pixels or regions of each image. The Geospatial Data Abstraction Library (GDAL) handles many standard georeferenced vector and raster file formats, so the output should be broadly compatible.

2.5 Feature Extraction to Correlate Mining with Environmental Impact

To correlate the regions of mining activity with the regions of environmental degradation, relationships must be extracted from the preprocessed data.

2.5.1 Options

Having combined the mining and environmental data in the previous step, options for finding meaningful relationships include:

- 1) simple logic,
- 2) statistical analysis, and
- 3) machine learning approaches.

2.5.2 Goals

The goal of this stage is to accurately locate regions where mining coincides with environmental degradation. The result should be suitable for both human analysis as well as further processing.

2.5.3 Criteria

Criteria for evaluating these approaches include time needed to research and develop each method, the computational complexity of each, and the accuracy of the results.

2.5.4 Comparison

	Research and Development Time	Computational Cost	Accuracy
Logic	Low	Low	Low
Statistics	Medium	Medium	Medium
Machine Learning	High	High	Medium to High

This table compares each algorithmic approach with estimates for each of the criteria.

2.5.5 Discussion

The simplest way to correlate mining and environmental impact is to identify areas where both are present. Like a Venn diagram, this approach would provide a basic boolean description of regions where both zones intersect. Although it would require very little research and development time to apply logical AND to each pixel, the accuracy is estimated to be low. Furthermore, this method would not be well suited for analyzing results that are probabilistic in nature.

Another way to correlate the data is to use statistical methods to identify zones that are likely to correspond to both mining and environmental data within some level of uncertainty. It would require more research and development to find and implement methods to process the data this way, but the accuracy is estimated to be better. The computational cost of this approach is estimated to be higher.

The most complex way to correlate the data is to use machine learning approaches such as the neural networks which were used to classify the spectroscopic imagery. This would require more research and development time and much more computational complexity than either of the other approaches. It is unknown whether the accuracy of this approach for correlation would exceed that of the statistical method, but it estimated to be as good or better.

2.5.6 Selection

Because we are interested in programming the library but not necessarily conducting a detailed environmental analysis, we have decided to pursue the logical approach to environmental correlation as a case study. We plan to identify pixels where waterway pollution corresponds to mining regions. Although our correlation function will be simplistic, our model will enable more sophisticated functions to be developed for environmental research.

2.6 Rank and Document Changes Over Time

To rank and document changes over time, data from a range of dates should be compared using correlations derived from the previous step.

2.6.1 Options

Several approaches to correlating imagery and historical data include:

- 1) processing separately and comparing manually,
- 2) processing separately and comparing automatically, and
- 3) processing simultaneously and comparing automatically.

Each option describes a high-level approach to designing the algorithm. Implementation details to be determined based on choice.

2.6.2 Goals

The goal of this stage is to add a temporal dimension to the spatial correlation between mining and environmental impact. It will require processing multiple historical data sets. The end user will be expected to select environmental data sets and imagery for a given geographical region. These may be processed manually for maximum control or automatically for batch operation. Because this module relies on all previous stages of the data processing pipeline, implementation details remain to be finalized.

2.6.3 Criteria

Criteria being evaluated include the time needed to research and develop each method, the computational complexity of each approach, and the quality of the results.

2.6.4 Comparison

	Research and Development Time	Computational Cost	Quality
Process Separately, Compare Manually	Low	High	Medium
Process Separately, Compare Automatically	Medium	High	Medium to High
Process Simultaneously, Compare Automatically	High	Very High	Medium to High

This table compares each approach to algorithm design and the estimated costs and benefits of each.

2.6.5 Discussion

The simplest approach is to process the data separately and compare the results manually. Because this simply runs the previous stages over multiple data sets, the research and development time is low. However, the quality of this approach is not high because the end user shoulders responsibility for comparing the results.

A more sophisticated design is to devise a means of combining multiple correlations and producing a unified result. This approach would reuse most of the original data flow and provide a comparison of the results. If variables such as geographic coordinates and atmospheric conditions could be successfully postprocessed by this method, the quality of the results is estimated to be better than the previous approach.

The most complicated approach would be to process multiple historical data sets simultaneously to mediate their transformations. This would require more research and development and likely some degree of refactoring. All of these approaches require a high computational cost, but this one is estimated to have the highest. However, if successfully implemented, this method is estimated to have the highest quality because each of the data sets is available for synchronization from beginning to end.

2.6.6 Selection

We now expect to implement a combination of the options discussed above. The user will be required to select the data for analysis. Then the data could be manually processed for maximum control or batch processed for convenience. Thus we expect to facilitate automated comparisons with both separate and simultaneous processing.

2.7 API Documentation

2.7.1 Options 1, 2, and 3

- 1) Sphinx

- 2) Doxygen
- 3) Epydoc

2.7.2 Goals

Auto-generated documentation is desirable in the design on COAL because there will be code used to initialize data sets from large data files, process said data, and visualize such data. These are complex tasks that will not have the most straight-forward and simple methods of implementation, so clear, comprehensive, and organized documentation would be helpful to anyone who potentially wants to use COAL on a cloud-based platform in the future.

2.7.3 Criteria

The main criteria being evaluated when comparing documentation generators is compatibility with Python 2 and Python 3, search engine integration, and use of reStructuredText.

2.7.4 Comparison

	Sphinx	Doxygen	Epydoc
Python 2 and 3 support	✓	✓	×
Search engine	✓	✓	×
reStructuredText	✓	×	×

This table shows an overview of Sphinx, Doxygen, and Epydoc against the criteria. Sphinx meets all three criteria, Doxygen misses the mark on reStructuredText, and Epydoc does meet any of the criteria.

2.7.5 Discussion

Epydoc is a widely used documentation generator for Python code. However, it was abandoned in 2009 and therefore does not have Python 3 support. It also does not offer a search engine feature, which is an incredibly handy feature to have when trying to find a method or class that meets a certain criteria. It also uses Markdown instead of reStructuredText, making it fall behind in the ease of use and elegance department. Doxygen fits more of the criteria, but it is a bit tricky since Doxygen by itself does not support Python documentation. It only does so through the use of a third-party generator like Epydoc. Sphinx fits all of the criteria and is the only choice that uses reStructuredText. reStructuredText provides mechanisms known as roles and directives that will render parts of the documentation to HTML automatically, making the documentation cleaner by not having to embed HTML, which is a common experience in Markdown [11].

2.7.6 Selection

We will be using Sphinx for the auto-generated documentation.

2.8 Static site generator

2.8.1 Options

- 1) Middleman
- 2) Jekyll
- 3) Roots

2.8.2 Goals

A static site generator will provide a fast and simple back-end for COAL's homepage. The homepage will not be dependent on real-time content, so we deemed a static site generator appropriate for the back-end.

2.8.3 Criteria

- 1) Cost to build.
- 2) Availability: easy to attain.
- 3) Compatibility

2.8.4 Comparison

	Middleman	Jekyll	Roots
Easily extendable	×	✓	✓
Built-in Github support	×	✓	×

2.8.5 Discussion

Jekyll is defined as a simple, blog-aware, static site generator and it is the most widely used today. Jekyll is also very easy to use. It has default integration with GitHub pages, which makes set-up and updates very simple. Consider what we are going to do right now: the blog post page, and compatible with GitHub pages. Jekyll is an ideal choice.

Middleman is not as widely used. Middleman does not have built-in support with GitHub pages, so it is more difficult to set up and update compared to Jekyll. Middleman is also difficult to write extensions for as it is not well documented. Roots is less used than Middleman, but it relies heavily on extensions and is easily extendable [12].

2.8.6 Selection

Jekyll seems to be the best choice after our discussion. Since we are using GitHub pages, it will be the simplest to use and fits our needs. Because one of our need for our website is to create a page for blog posting, so we intend to use Jekyll mainly for that function, which is considered very convenient.

2.9 Front-end framework

2.9.1 Options

- 1) Bootstrap
- 2) Skeleton
- 3) Foundation

2.9.2 Goals

Our site should look professional and have a simple, user-friendly interface that is also aesthetically pleasing. In order to accomplish this, a front-end framework is desirable.

2.9.3 Criteria

- 1) Well designed interaction
- 2) Professional-looking Interface
- 3) Contains required contents

2.9.4 Comparison

	Bootstrap	Skeleton	Foundation
Abundance of themes	✓	×	✓
Highly mobile friendly	✓	✓	✓
Large community support	✓	×	✓

2.9.5 Discussion

Skeleton is fairly bare-bones, so it isn't ideal for when developers want many themes to choose from [13]. Skeleton is also a much less active project. At the time of writing this document, the last update made on the official GitHub page for Skeleton was made in December of 2014. Bootstrap and Foundation are both very popular and active projects that compare pretty evenly. Bootstrap has a slightly different approach to the mobile version of a site than Foundation does. Foundation was developed under the assumption that "anything not under a media query is considered mobile [14]." Bootstrap will only design something for mobile if specified. This is not very important to COAL's homepage though. Both are compatible with Jekyll, our choice for a static site generator. The deciding factor between Bootstrap and Foundation comes down to the preferences and experiences of the team members. We have experience with Bootstrap and enjoy the look and feel of it, so it is our choice for COAL's homepage's front-end.

2.9.6 Selection

We will be using Bootstrap for the front-end framework of COAL's homepage.

3 CONCLUSION

In this paper we described and compared technologies for implementing the COAL algorithm suite and associated documentation. The core functionality consists of a data processing pipeline that transforms raw imagery and data into meaningful relationships. We have become familiar with relevant literature, developed detailed sketches of our algorithms, and implemented a framework for most of the modules using the libraries we have chosen. In addition to the algorithms, our project will be supported by API documentation for programmers and a homepage for end users. A discussion of implementation choices for each component was provided along with a recommendation based on a comparison of costs and benefits. The team members responsible for each component were identified in the introduction. We hope that this discussion has provided the reader with a better understanding of the design choices that have been and have yet to be made as we continue to collaborate with our client on research and development.

4 REVISIONS

4.1 Feature extraction in AVIRIS data

Revisions to Section 2.1: .

4.2 Classification of AVIRIS data

Revisions to Section 2.2: .

4.3 Identify Mining

Revisions to Section 2.3: .

4.4 Preprocessing Data to Correlate Mining with Environmental Impact

Revisions to Section 2.4: Revised to document our choice to convert GIS to raster data.

4.5 Feature Extraction to Correlate Mining with Environmental Impact

Revisions to Section 2.5: Revised to describe the simple case study chosen by client.

4.6 Rank and Document Changes Over Time

Revisions to Section 2.6: Revised to describe expectation that end user will choose data sets, and updated the selection to be a combination of manual and batch processing.

4.7 API Documentation

Revisions to Section 2.7: .

4.8 Static site generator

Revisions to Section 2.8: .

4.9 Front-end framework

Revisions to Section 2.9: .

REFERENCES

- [1] T. A. Brown, H. Clayton, and X. Wang, Requirements document, Fall 2016.
- [2] <http://www.spectralpython.net/>
- [3] <https://github.com/hyperspy/hyperspy>
- [4] <https://github.com/isaacgerg/matlabHyperspectralToolbox>
- [5] J. A. Benediktsson *et al*, "Classification and Feature Extraction of AVIRIS Data," IEEE Trans. on Geoscience and Remote Sensing, Vol. 33, No. 5, September 1995.
- [6] <http://deeplearning.net/software/pylearn2/>
- [7] <https://github.com/tensorflow/tensorflow>
- [8] G. A. Swayze *et al*, "Using Imaging Spectroscopy to Cost-Effectively Locate Acid-Generating Minerals at Mine Sites: An Example from the California Gulch Superfund Site in Leadville, Colorado," paper presented at JPL Airborne Geoscience Workshop, Leadville, Colorado, 1998.
- [9] R. N. Clark, *et al*, "Mineral Mapping with Imaging Spectroscopy: The Ray Mine, AZ," Summaries of the 7th Annual JPL Airborne Earth Science Workshop, R.O. Green, Ed., JPL Publication 97-21. Jan 12-14, pp67-75, 1998.
- [10] D. Nally, "Moving Mountaintops: Monitoring Surface Mine Expansion and Reclamation Using Landsat Imagery," Tufts Univ., Spring 2011.
- [11] E. Holscher. (2015, March 15). *Why You Shouldn't Use "Markdown" for Documentation* [Online]. Available: <http://ericholscher.com/blog/2016/mar/15/dont-use-markdown-for-technical-docs/>
- [12] M. Christensen. (2015, November 16). *Static Website Generators Reviewed: Jekyll, Middleman, Roots, Hugo* [Online]. Available: <https://www.smashingmagazine.com/2015/11/static-website-generators-jekyll-middleman-roots-hugo-review/>
- [13] *Best CSS Frameworks - Bootstrap vs Foundation vs Skeleton?* [Online]. Available: <http://customwebsitedevelopment.blogspot.com/2016/01/best-css-frameworks-bootstrap-vs-foundation-vs-skeleton.html>
- [14] M. Schenker. (2014, September 15). *Bootstrap vs. Foundation: Which Framework is Better?* [Online]. Available: <https://bootstrapbay.com/blog/bootstrap-vs-foundation/>
- [15] V. Catterson. (2013, November 27). *Understanding data science: dimensionality reduction with R* [Online]. Available: <http://cowlet.org/2013/11/27/understanding-data-science-dimensionality-reduction-with-r.html>