PeakLearner: an interactive web interface for genomic data

Jacob Christiansen, Allen Clarke, Yuanyuan Fu, John Jackson

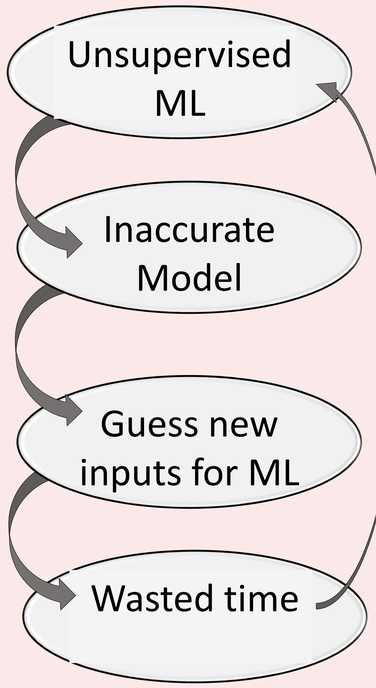
School of Informatics, Computing and Cyber Systems

NORTHERN ARIZONA UNIVERSITY | College of Engineering, Forestry, and Natural Sciences
School of Informatics, Computing, and Cyber Systems

Abstract

- Observing genomic data is a very visual task.
- Scientists need a way to correct data models after they are made to save time when analyzing genes and determining which mutations could be the cause of cancer.

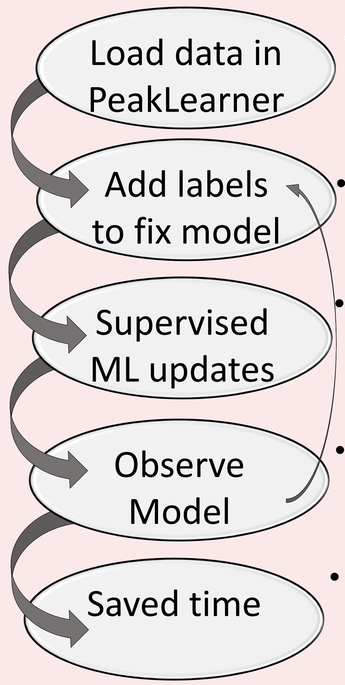
Problem



- Unsupervised Machine Learning yields inaccurate models.
- These models are hard to correct.
- Lots of guessing to correct models.
- No visual graph or chart for a very visual type of data.
- Time is wasted fixing models and dealing with False Positives/Negatives.

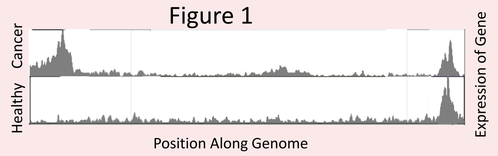
Solution

Our solution features:
- A simple to use web interface.
- The same look and feel as known Genome Browsers.
- Click and drag functionality to add labels.
- 4 different label types.
- Automatic model redrawing.
- Dr. Hocking's machine learning algorithm to calculate new models.



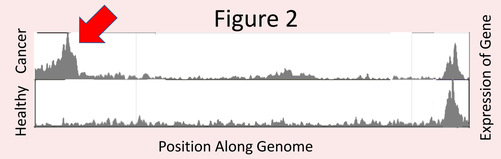
- The user loads their data in to PeakLearner
- Users add labels to correct the model.
- The supervised ML algorithm outputs a better model.
- Users continue adding labels to improve model.
- The user has an accurate model in less time.

Figure 1



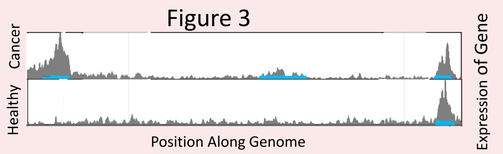
- Figure 1 shows two ChIP-Seq datasets
- One is a healthy sample, the other a cancer sample.
- Clearly, they are visually different.

Figure 2



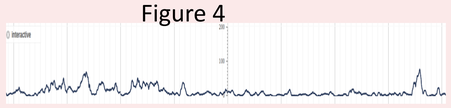
- The gene on the left end of Figure 2 is expressed by the cancer sample and not by the healthy sample.
- This is called a "peak".

Figure 3



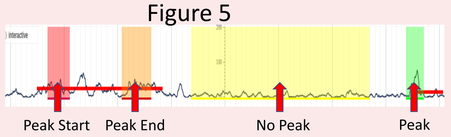
- The blue lines in Figure 3 represent a model.
- Each line denotes a region where the model thinks a peak exists.
- There should be no peak in the middle, but the model is not accurate before adding labels.

Figure 4



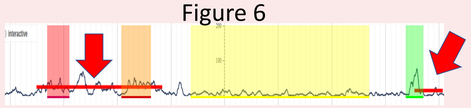
- Figure 4 is the alpha for PeakLearner.
- Data looks similar to the genomic data shown in Figures 1-3.
- PeakLearner allows for the same data viewing capabilities as existing genome browsers.

Figure 5



- The user can now input new labels on to the data by clicking and dragging.
- There is 4 kinds of labels, each shown in Figure 5.
- Each label provides different information to the ML Algorithm.

Figure 6



- Once the label is saved the model is drawn on top of the data as a thick red line where the algorithm expects a peak should be (Figure 6).
- The model is redrawn every time a label is added.

Challenges

- BedGraph files could not handle colors for labels.
- Solution: implementing a color callback system.
- BigWig data files are very large, too big to send.
- Solution: implement range requests in the server to send only pieces of the files.

Testing

- Unit tests to ensure each individual function is working as intended.
- Integration testing to make sure the browser, server, and database communicate properly.
- Our integration testing tools are shown below.

Technologies

- BerkeleyDB - https://www.oracle.com/database/technologies/related/berkeleydb-downloads.html
- CMDColin - https://github.com/cmdcolin
- Python - https://www.python.org/
- Javascript - https://www.javascript.com/
- Jbrowse - http://jbrowse.org/

Acknowledgements

- Dr. Toby Hocking
- Dr. Eck Doerry
- Mahsa Keshavarz
- Colin Diesh