Overview of Project Documentation - Growth Rate Equations

**Tests Ran**

* Each results file includes multiple tests with varying parameters

Balanus Growth Rate:

1. Temperature vs Growth Rate
   1. Datasets: relevant\_X\_train.csv (temperature only), BalanusGrowthRate.csv
   2. Results File: Temperature Vs Growth Rate
2. Temperature vs Growth Rate with outliers removed
   1. Datasets: X\_train\_Outliers.csv (temperature only), BalanusGrowthRate\_Outliers.csv
   2. Results File: Temperature Vs Growth Rate
3. Day of Year vs Growth Rate with outliers removed
   1. Datasets: X\_train\_Outliers.csv (time only), BalanusGrowthRate\_Outliers.csv
   2. Results File: Day of Year vs Growth Rate
4. Day Since Experiment Start vs Growth Rate with outliers removed
   1. Datasets: time\_Outliers.csv, BalanusGrowthRate\_Outliers.csv
   2. Results File: Day of Year vs Growth Rate
5. Correlation Matrix
   1. Datasets: X\_train.csv, BalanusGrowthRate.csv
   2. Results File: Correlation Matrix
6. Relevant vs Balanus Growth Rate
   1. Datsets: relevant\_X\_train.csv, BalanusGrowthRate.csv
   2. Results File: Relevant vs Balanus Growth Rate
7. Relevant vs Balanus Growth Rate with outliers removed
   1. Datsets: X\_train\_Outliers.csv, BalanusGrowthRate\_Outliers.csv
   2. Results File: Relevant vs Balanus Growth Rate
8. Relevant vs Balanus Growth Rate with outliers removed and log function applied
   1. Datsets: X\_train\_Outliers.csv, BalanusLOG.csv
   2. Results File: Relevant vs Balanus Growth Rate

Mytilus Growth Rate:

1. Relevant vs Mytilus Growth Rate
   1. Datsets: relevant\_X\_train.csv, MytilusGrowthRate.csv
   2. Results File: Relevant vs Mytilus Growth Rate
2. Relevant vs Mytilus Growth Rate with outliers removed
   1. Datsets: X\_train\_Outliers\_M.csv, MytilusGrowthRate\_Outliers.csv
   2. Results File: Relevant vs Mytilus Growth Rate
3. Relevant vs Mytilus Growth Rate with outliers removed and log function applied
   1. Datsets: X\_train\_Outliers\_M.csv, MytilusLOG.csv
   2. Results File: Relevant vs Mytilus Growth Rate

**Takeaways**

* Balance:
  + The key to using the symbolic regressor effectively was finding the right balance of parameters.
  + Key components to balance: population\_size, generations, stopping\_criteria, and parsimony\_coeffecient. Everything else was kept relatively constant
  + The goals were to have the regressor run long enough so that the fitness levels decreased significantly while maintaining a reasonable length. The stopping\_criteria had to be set not too high so that the fitness could decrease but not too low so that the regressor would not run forever if the fitness level reached a minimum.
* Function Set Parameters:
  + Regressor did not work well with trig functions or sqrt
  + Led to more complex equations that were not simplifiable
    - Ex: Regressor would return sqrt(-x) which would cause error in simplifying function due to use of imaginary numbers
  + Decided to use function set ['add', 'sub', 'mul', 'div', 'neg', 'inv']
* Correlation Matrix:
  + Creating a correlation matrix was somewhat useful
  + Helped identify the significant relationships between variables and growth rate
  + Focused only on individual relationships so was not helpful in identifying relationships between groups of variables and growth rate
  + Using the correlation matrix and input from Mark, we identified the relevant variables to use in future tests
* Outliers:
  + Outliers had a significant influence on the regressor
  + Output equations focused on making sure predicted values matched the outliers while the rest of the values had little variation
    - Ex: Equation 6 in Relevant vs Balanus Growth Rate
  + Decided to remove growth rate outliers for testing
* Log Functions:
  + Growth rates were also calculated with log functions to reduce the variation in hopes of producing a better equation
  + Function Applied: ln(countt / countt+1) / (t+1 - t)
  + Led to new equations and uncovered new behaviors of the regressor
* Parsimony Coefficient
  + Log tests revealed that parsimony coefficient is not scaled based on the fitness of the output equations, it is only subtracted from the fitness:
    - From gplearn: return self.raw\_fitness\_ - penalty
  + Therefore, to make parsimony coefficient effective at smaller fitness values, it must be extremely low itself
    - Ex: stopping\_criteria=.0025 -> parsimony\_coefficient=.000008
* Effect of Very Low Parsimony Coeffecient
  + Having a very low parsimony coefficient led to the loss of variable coefficients
  + This is due to each variable coefficient adding 2 to the length (one for the value, one for the multiplication action) and so leaving them out helped the regressor reduce the equation length
  + Ex: parsimony\_coefficient=.000008-> X4\*X5\*\*2\*(X2\*X5\*\*2 - X3)/(-X3\*\*3\*X5\*\*2\*(-X3 + X5\*\*2 + X8) + X3\*\*2\*(X2\*X5\*\*2 - X3) + X5\*\*2\*(X2\*X5\*\*2 - X3)\*(-X1 + X2 + X3))

**Future Extensions**

Ideas for how to continue the project

1. Use another regressor
   1. We only experimented with GPLearn’s symbolic regressor. It was effective but also had its limitations. Using a different regressor might provide features that are more effective for the purposes of this project.
2. Use another machine
   1. We also experimented only on Google Colab servers for convenience of sharing data and running tests. Another machine might provide more capabilities that would better support the regressor’s analysis.
3. Try to extract intermediate equations
   1. One of the limitations of GPLearn was that it only outputted equations after it reached the stopping criteria. Being able to see intermediate results would be very helpful and provide new insight into the regressor’s behavior.
4. Produce more graphs/visuals
   1. Our project unintentionally revealed more about the behavior of symbolic regression than the data and equations itself. Equations can be explored further by producing graphs that compare how the predicted growth rate values change based on manipulating one variable and keeping the rest constant.