Reports  
  
A report on grid, random, bayes GP and RF

Summary of estimation of what I have currently

A report to show tuning methods. Full go high dimension as well as Two step- narrowing of domain space with heatmaps.  
Grid/random search manual step method for first step  
Bayesian Op/ Random two step (Grid for first) for second steps  
Compare to full dimensional base times for each set

**Report structure:**  
  
**Abstract:**  
Some brief details on Neural networks and hyperparameter optimisation. Explain how this project works on optimising neural networks for bioprocess data. State we will investigate grid, random, bayes GP and bayes Random forest, (add more techniques like TPE and EAs later)  
  
**Introduction:**  
More details on neural networks, the concept of hyperparameter optimisation and its merits.  
Some details on what we are investigating, the steps I’ve taken and mentions of what scripts I used and giving due credit. (I will later make a technical document/powerpoint to help other people navigate the code and files I have created, detailing where I made changes)

**Part 1: Overview of Hyperparameter Optimisation techniques ( basically a theory section with some initial estimates of results)**  
  
**Grid search** – basics of how it works and some historic context.   
Advantages: Find the whole domain space, can be parallelised  
Disadvantages: extreme curse of dimensionality, can take long to execute in any case, requires user to select the range and scale of values which may not be obvious.  
Some details on approximate execution times and performance on my desktop.  
  
Simliar for **Random search**Advantages: isn’t stuck to one axis  
  
**Bayesian Optimisation GP** – basics of how it works, details of the key steps and equations showing how the posterior, prior and acquisition work together.  
Advantages: Can automatically find the appropriate scale within the domain space. Finds optimum in relatively few iterations ~ few hundred  
Disadvantages: Also suffers from curse of dimensionalty. Can optimise towards one local maxima when many are available. Higher number of evaluations (< 500) can be detrimental due to the posterior updating and becoming more expensive than the original objective function. Cannot be parallelised.  
Some details on approximate execution times and performance on my desktop.  
  
Similar for **random forests**

A subset of data is sampled with replacement to form a decision tree. This is repeated many times to generate a random forest of varied decision trees. The hyperparameters are passed to each tree and the outcomes totaled up to see if the hyperparameters are considered viable. **B**uiltfrom decision trees, easy to build and implement but not particularly useful, quite inaccurate. They work really well with data used to create them but are not good for predicting new data.  
Random Sampling with replacement (can select the same data point multiple timees) to make a bootstrapped dataset.Advantages: Won’t overfit the model, can handle missing data, can handle high dimensionality

Disadvantage: Good at classification but not so good for continuous, regression based data.  
  
**Evolutionary Algorithms** – Basic concept of how they work; crossover, mutation, populations and generations.  
Advantages: Guaranteed to find the entire set of optimal solutions without getting stuck on local maximas. Can be parallelised.  
Some details on approximate execution times and performance on my desktop. The fact we had to scrap it for now. Options for further work to parallelise or build a surrogate version to replace the expsive objection function.  
  
Brief table and overview of current conclusions

**Part 2: Evaluation of Grid search, Random Search, Gaussian Process Bayesian Optimisation and Random Forest Bayesian Optimisation.**

**Methodology:**

**A: Manually evaluating the entire domain space with exhaustive grid search.**As detailed in my powerpoint, describe how one might use grid search and 2D heat maps to analyse a portion of the domain space enabling approximate identification of the optimal regions.  
The steps I took to search the domain space could also be used as a manual tuning procedure in itself:

* Plot heatmaps of Hidden Layers, neuron numbers and activation functions. Identify the optimal regions for these hyperparameters.
* For a given configuration of neurons, layers and activation function, investigate the relationship between ‘Epochs and learning rate’, ‘Epochs and batch size’ as well as ‘batch size and learning rate’.
* Identifying the optimal regions from each of these plots will help the tuner further constrain the hyperparameters to the optimal region of the domain space.

**B: Two step method utilising Grid search and/or Bayesian Optimisation.**Similar to above but using Grid search for the first bullet point, i.e. things that are low dimensions and can be evaluated quite quickly even with grid search. Then, use Bayesian Optimisation for more high complexity hyperparameters like Epochs, learning rate, batch size.  
  
**C: high dimension hyperparameter optimisation.**Letting Grid and random search run through all possible configurations at extreme computational costs.  
Letting Bayesian Optimisation run free with all hyperparameters, may not find ideal solution, but will get close to some tolerance.

**Results and discussion for sigmoidal, 2 hidden layer ANN.**

Detail the result for each technique and each methodology A,B,C.  
Discuss the possible outcomes and their implications.  
  
**Conclusions**