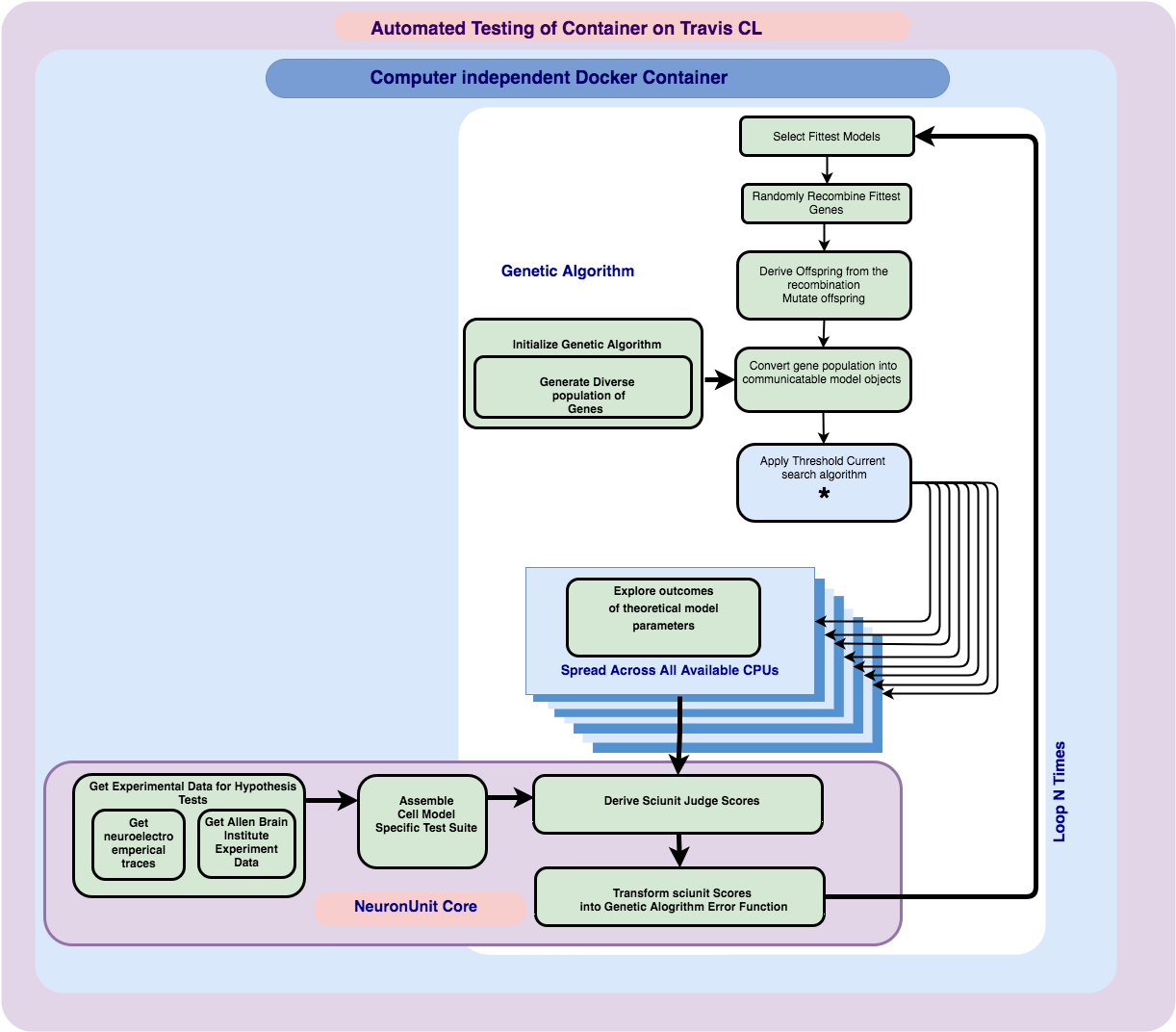
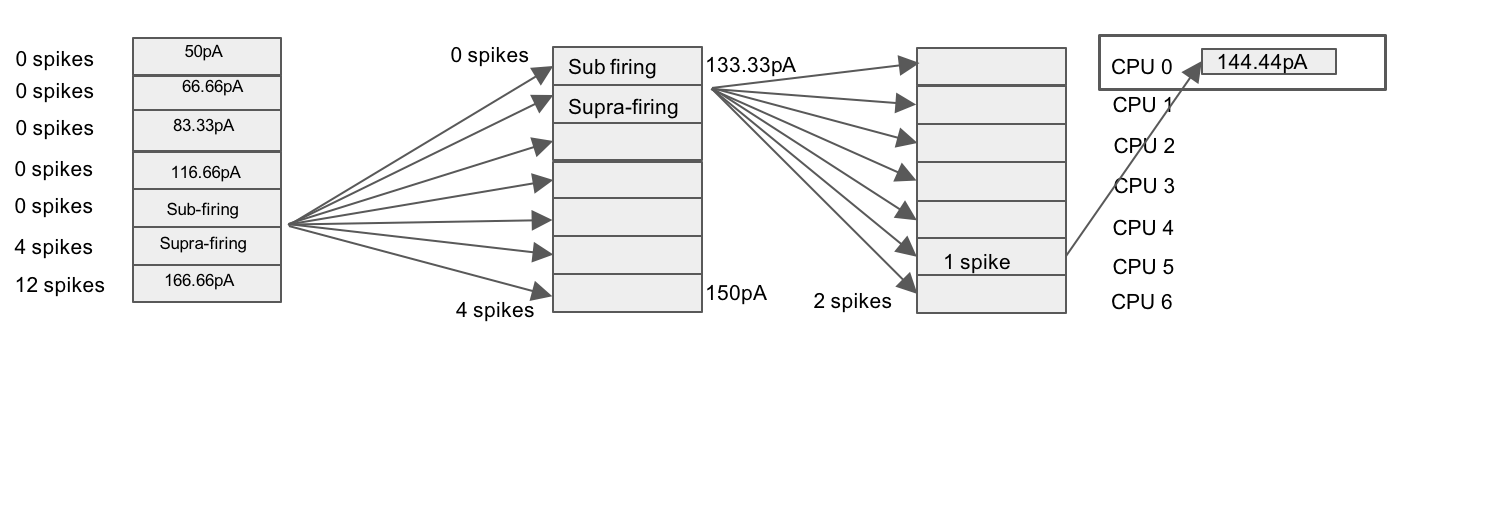
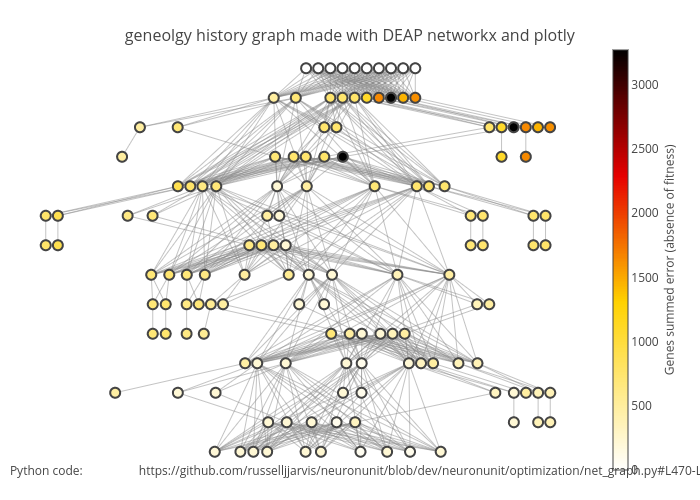
Are the fonts in the boxes too small? Should the box shapes be more similar.

*Figure 1*: The optimization algorithm consisting of a sequence of operations for exploring differences in neuronal models that correspond to genetic algorithm evolution. Another major series of instructions is responsible for evaluating the fitness of output model behavior, inside the NeuronUnit test suite. The algorithm scales by spreading workload onto unsused CPUs. Code blocks that are implemented with parallel map functions is depicted in the diagram using depth and diverging arrows. Parallel scaling of the program workload occurs at two of computational intensity, where invocation of a neuronal simulator occurs. Both of these points, or bottle necks would otherwise cause severe performance degradation, delegating out work in an embarrassingly parallel fashion, facilitates timeliness of the optimizers convergence.



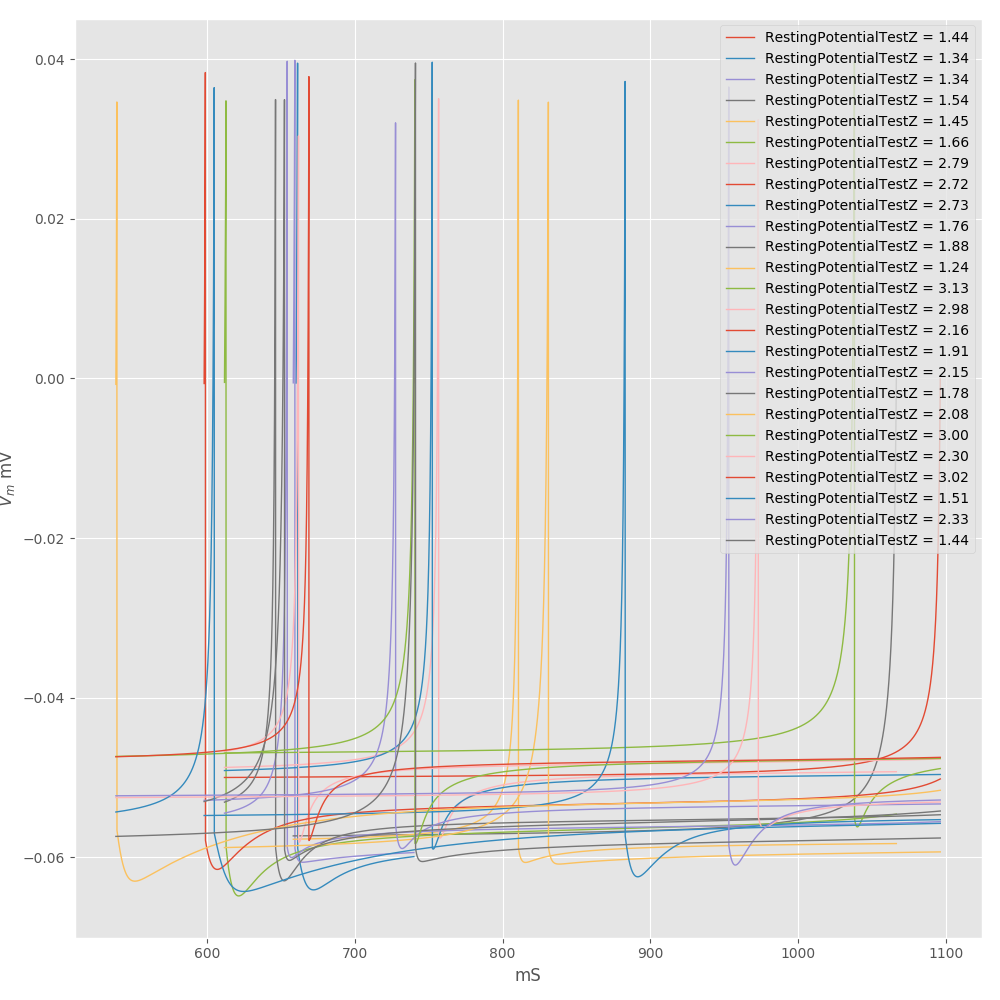
*Figure 2*. As described in figure 1, there were several computationally intensive sites in the program workflow, chief among these sites was the search for threshold current injection value which varies between models so it must be performed on a per model basis. Many of the NeuronUnit test criteria depend on a threshold current value for the model being known in advance. The search for a threshold current injection value, resulting in only one neuronal spike, was speed up using brute force. Intervals in a plausible range of current injection values were at first coarsely sampled across 7 CPUs, as the workers return from their coarse grained sampling, their results inform a narrower interval to search, this narrower interval is then sampled at a higher granularity, this pattern continues until a threshold current injection value is found.



*Figure 3*

The DEAP library and other python libraries such as *networkx* *plotly* were enabled a visualization of changes in model fitness that result from gene evolution progress. This diagram serves as reflection of the genetic algorithms performance, as it illustrates how unfit models are rapidly discarded, and that error is minimized over relatively few generations.

The diagram provides heuristic value about the workings of Genetic Algorithms by clarifying the relationship between model fitness genetic recombination and inheritance.



*Figure 4*

Figure 4 demonstrates that a diversity of model behavior was evaluated and appropriately scored by the combined action of *NeuronUnit* and *DEAP*. Models exhibiting Z-scores that are furtherest from zero were deemed fitter. In this case models that had a resting membrane potential close to -68.24 *mV,* an experimental mean, were actively selected for. Other important wave form features that are also visible in this figure include variation in spike height, and spike length.



*Figure 5* shows an error surface corresponding to parameter a and parameter b, in the Izhiketich neuronal model. Each of the pair of parameters, in the case a, and b are systematically varied in a way where one variable is held constant while the other variable increments. Planes through the error surface were obtained in this fashion. Inspection of these planes reveals complex surfaces with multiple minima, thus warranting a more robust search strategy over less robust gradient descent based search strategies.

This figure belongs near the text.

For a single model class and biological neuron type, we obtained an error surface via a Non-Dominated Sort Genetic Algorithm (NSGA), corresponding to plausible subsets of candidate parameter values that respect intrinsic biological diversity.

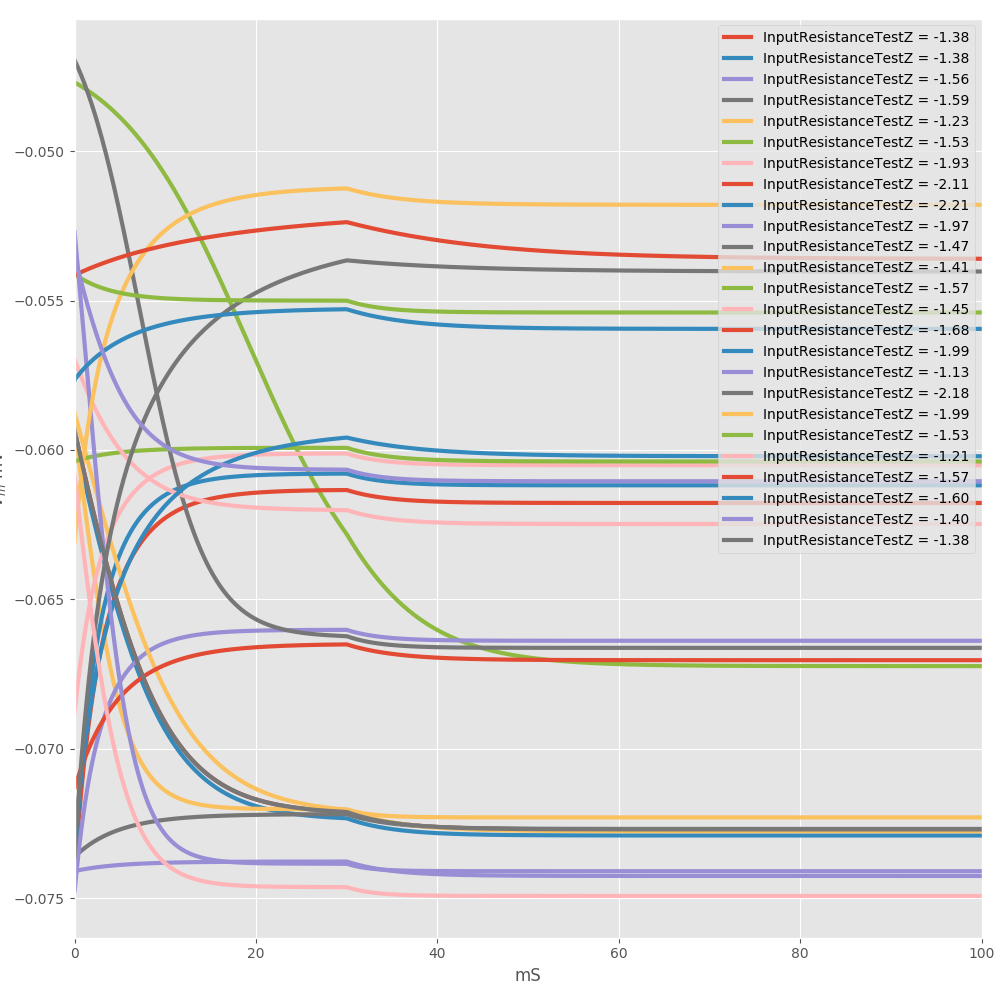


Figure 4 shows the time course of the membrane potential for a diverse population of models of

**A paragraph that I would like to add somewhere inside the body of the poster:**

Importantly the entire software optimization program is featured inside a dedicated docker container. Such containers provide a machine and Operating System independent environment such that users of the software can be guaranteed a greater level of software environment convergence and compatibility. Utilization of Docker is important, the burden model reproducibility should extend to all the way down to the operating system level.

