How to quickly get going with pyEvolve

The purpose of this page is to get you up-and-going using pyEvolve quickly, so you can use it in your own exercises. There are bound to be features of pyEvolve not covered in this page. For thosefeatures, check out the official pyEvove documentation at http://pyevolve.sourceforge.net/0_6rc1/ .

Almost everything you need in order to "evolve something" in pyEvolve has a default. If you don't tell pyEvolve otherwise, it will use those defaults and run "fairly well." One thing that you absolutely have to tell pyEvolve is what is the fitness function. More on that later.

- Step 1: tell pyEvolve what your genome looks like.
    - **genome = G1DList.G1DList(20)**
    - this tells pyEvolve that your genome is a list of twenty integers.
- Step 2: tell pyEvolve what is your fitness function.
    - **genome.evaluator.set(eval_func)**
    - this tells pyEvolve that the fitness of these genomes is determined by a function called *eval_func*.
    - somewhere else you have to declare *eval_func*.
- Step 3: create the genetic algorithm engine.
    - **ga = GSimpleGA.GSimpleGA(genome)**
- Step 4: run the evolutionary process.
    - **ga.evolve(freq_stats=10)**
    - this runs the genetic engine, giving reports every 10 generations.

That's it. Those few lines will run a genetic process. If you think there's a lot of things that you needed to specify, but that you haven't, you're right. pyEvolve is using a bunch of defaults. Here are some of those defaults.

- what values get assigned to those genomes, to those strings of twenty integers?
    - random integers between 1 and 10.
    - if you want to have a different range for those random numbers, use this:

        -
          ```
          genome.setParams(rangemin=0, rangemax=10)
          ```
- what type of crossover gets used?
    - one point crossover.
    - if you want to change that, use something like genome.crossover.set(Crossovers.G1DListCrossoverUniform)
    - the whole list of crossover types you can possibly use is at http://pyevolve.sourceforge.net/0_6rc1/module_crossovers.html#Crossovers.G1DListCro

- for the selection method, the default is rank based. The list of other available options is at http://pyevolve.sourceforge.net/0_6rc1/module_selectors.html
- here are some other default values, as mentioned in the pyEvolve documentation:
    - *By default, the GA will evolve for 100 generations with a population size of 80 individuals, it will use the mutation rate of 2% and a crossover rate of 80%, the default selector is the Ranking Selection (`Selectors.GRankSelector()`) method. Those default parameters was not random picked, they are all based on the commom used properties.*
    - *if you want to change those defaults, use the following methods:*
        - ga.setGenerations(50)
        - ga.setCrossoverRate(1.0)
        - ga.setMutationRate(0.25)
        - ga.setPopulationSize(800)
- If you want to have real values in the genome, instead of integers: http://pyevolve.sourceforge.net/0_6rc1/examples.html#example-2-real-numbers-gaussian-mutator
- for binary values: http://pyevolve.sourceforge.net/0_6rc1/examples.html#example-10-the-1d-binary-string
- for two-dimentional arrays:
    - http://pyevolve.sourceforge.net/0_6rc1/examples.html#example-9-the-2d-list-genome
    - http://pyevolve.sourceforge.net/0_6rc1/examples.html#example-16-the-2d-binary-string

One thing is pending: defining a fitness function. Remember all the way at the beginning where we defined what the name of the evaluation function was? (genome.evaluator.set(eval_func)). Now we have to define eval_func, like this:

```
def eval_func(chromosome):
   score = 0.0

   # iterate over the chromosome elements (items)
   for value in chromosome:
      if value==0:
         score += 1.0

   return score
```

Last modified: Wednesday, September 28, 2016, 10:44 PM