Propongo que un experimento sea un diccionario. Cada valor puede ser un número o una distribución. Nosotros nos crearemos nuestro paquete de distribuciones.

El objetivo es que al final, si uno dice:

organism.speed = experiment\_data['organisms']['category\_a']['speed']()

Entonces organism.speed toma un valor según la distribución adecuada.

# Default experiment

{'experiment name': "Experiment name"

'biotope': {

'size': (100, 200),

'substances': None },

'organisms': [{'category': 'General\_category',

'number\_of\_organisms': 10,

'genes': {

'speed': 0.0,

'mutation\_frequency': 1.0

},

'status': {

'coordinates': {

'type': 'Biotope call',

'name': 'seek\_free\_position'}

} }],

'outlays': { },

'constraints': { },

'mutability': { }

}

# Nota: Estos serán los valores por defecto

# Experiment 1

{'experiment name': "Strength vs photosyntesis capacity"

'organisms': [{'category': 'Plants',

'number\_of\_organisms': 50,

'genes': {

'strength': {

'type': 'built-in function',

'name': 'gaussian',

'mean': 10,

'variance' 2},

'photosynthesis\_capacity': {

'type': 'built-in function',

'name': 'uniform distribution',

'interval': [10, 30] }},

'status': {

'energy\_reserve': 100.0}}

},

{'category': 'Animals',

'number\_of\_organisms': 50,

'genes': {

'strength': 10,

'speed': {

'type': 'built-in function',

'name': 'discrete distribution'

'values': [

{'value': 0.0, 'probability': 0.25},

{'value': 1.0, 'probability': 0.70},

{'value': 5.0, 'probability': 0.05}] } },

'status': {

'energy\_reserve': 100}

}

],

'outlays': {

'hunting': {

'type': 'built-in function',

'name': 'linear function',

'terms': [

{'parameter': 'strength', 'coefficient': 3.0},

{'parameter': 'speed', 'coefficient': 0.2},

{'parameter': None, 'coefficient': 5.0}]},

'moving': {

'type': 'built-in function',

'name': 'linear function',

'terms': [

{'parameter': 'strength', 'coefficient': 1.0},

{'parameter': 'photosynthesis\_capacity', 'coefficient': 25.0},

{'parameter': 'speed', 'coefficient': 5.0},

{'parameter': None, 'coefficient': 1.0}]},

'procreating': {

'type': 'built-in function',

'name': 'linear function',

'terms': [

{'parameter': 'strength', 'coefficient': 3.0},

{'parameter': 'photosynthesis\_capacity', 'coefficient': 3.0},

{'parameter': 'speed', 'coefficient': 3.0},

{'parameter': None, 'coefficient': 5.0}]},

'living': {

'type': 'la vida loca',

'name': 'linear function',

'terms': [

{'parameter': 'strength', 'coefficient': 1.0},

{'parameter': 'photosynthesis\_capacity', 'coefficient': -1.0},

{'parameter': 'speed', 'coefficient': 2.0},

{'parameter': None, 'coefficient': 5.0}]}

},

'constraints': {

'procreating': {

'type': 'interpreted function',

'name': 'ornanism constraint',

'a': 'energy\_reserve',

'r': ('random number', 'uniform\_distribution [0, 1]'),

'expression': "a\*r > 100.0" },

'hunting': {

'type': 'interpreted function',

'name': 'compare\_predator\_vs\_prey',

'a': ('predator', 'strength'),

'b': ('prey', 'strength'),

'r1': ('random number', 'uniform\_distribution [0, 1]'),

'r2': ('random number', 'uniform\_distribution [0, 1]'),

'expression': "a\*r1 > b\*r2 } },

'dying': {

'type': 'built-in function',

'name': 'death\_because\_of\_low\_energy',

'minimun\_level\_of\_energy': 10.0

} },

'mutability': {

'strength': {

'increments': {

'type': 'built-in function',

'name': 'gaussian',

'mean': 0.0,

'variance' 0.01},

'mutation\_frequency': 0.05,

'allowed\_interval': [0, 'infinity']

},

'photosynthesis\_capacity': {

'increments': {

'type': 'built-in function',

'name': 'gaussian',

'mean': 0.0,

'variance' 0.15},

'mutation\_frequency': 0.001

}

}

}

# Nota: En este experimento los animales podrían evolucionar para ser plantas y viceversa

# Experiment 2

{'organisms': [

{'category': 'Mutants',

'genes': {

'strength': 'default',

'photosynthesis\_capacity': 20.0,

'speed': {

'type': 'built-in function',

'name': 'discrete distribution',

'value\_list': [

{'value': 0.0, 'frequency': 0.25},

{'value': 1.0, 'frequency': 0.75}]},

'generation': 0 ,

'mutation\_frequency': 0.1},

'status': {

'energy\_reserve': 100.0} },

{'category': 'No\_Mutants',

'genes': {

'strength': 'default',

'photosynthesis\_capacity': 20.0,

'speed': {

'type': 'built-in function',

'name': 'discrete distribution',

'value\_list': [

{'value': 0.0, 'frequency': 0.25},

{'value': 1.0, 'frequency': 0.75}]},

'generation': 0,

'mutation\_frequency': 0.0},

'status': {

'energy\_reserve': 100.0} },

'outlays': {

'load outlays': "/Pepito/Experiments/cool experiment.exp" },

'constraints': {},

'mutability': {

'strength': {

'percentage\_increments': {

'type': 'built-in function',

'name': 'gaussian',

'mean': 0.0,

'variance' 0.01},

'mutation\_frequency': 0.05,

'allowed\_interval': [0, 'infinity']

},

'speed': {

'type': probabilistic\_automaton',

'states': {0.0, 1.0},

'probabilities\_matrix': [

[0.99, 0.01]

[0.02, 0.98]] },

'generation': {

'increments': 1},

'mutability': {

'percentage\_increments': {

'type': 'built-in function',

'name': 'gaussian',

'mean': 0.0,

'variance' 0.01},

'allowed\_interval': [0, 1]

}

}

}

# Nota: En este experimento veremos si los mutantes ganan a los no mutantes en la

# guerra por la supervivencia

# Experiment 3

{'organisms': [

{'genes': {

'attack\_capacity': {

'type': 'built-in function',

'name': 'chi-squared',

'k': 3},

'defense\_capacity': {

'type': 'built-in function',

'name': 'chi-squared',

'k': 3},

'photosynthesis\_capacity': 20.0,

'energy\_reserve\_at\_birth': 100.0,

'minimun\_energy\_reserve\_at\_procreation': 200.0},

'status': { 'energy\_reserve': 200.0} }],

'outlays': {

'load outlays': "/Pepito/Experiments/cool experiment 2.exp",

'living': {

'type': 'built-in function',

'name': 'n-linear function',

'terms': [

{'parameters': ['attack\_capacity', 'defense\_capacity'], 'coefficient': 1.0},

{'parameters': ['photosynthesis\_capacity'], 'coefficient': -1.0},

{'parameters': ['attack\_capacity', 'photosynthesis\_capacity'], 'coefficient': 25.0},

{'parameters': [], 'coefficient': 0.5}]} },

'constraints': {

'procreating': {

'type': 'threshold',

'parameter': 'energy\_reserve',

'threshold': 'minimun\_energy\_reserve\_at\_procreation'},

'hunting': {

'type': 'interpreted function',

'name': 'compare\_predator\_vs\_prey',

'a': ('predator', 'attack\_capacity'),

'b': ('prey', 'defense\_capacity'),

'r1': ('random number', 'uniform\_distribution [0, 1]'),

'r2': ('random number', 'uniform\_distribution [0, 1]'),

'expression': "a\*r1 > b\*r2 } },

},

'mutability': {

'all genes': {

'percentage\_increments': {

'type': 'built-in function',

'name': 'gaussian',

'mean': 0.0,

'variance' 0.01},

'mutation\_frequency': 0.05,

'allowed\_interval': [0, 'infinity'] }

}

}

# Nota: En este experimento veremos si los mutantes ganan a los no mutantes en la

# guerra por la supervivencia