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',

'number\_of\_organisms': 10,

'genes': {

'speed': 0.0

},

'status': {

'coordinates': {

'type': 'Biotope call',

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

} }],

'outlays': { },

'constraints': { },

'mutability': { }

}

# Nota: Estos serán los valores por defecto

# Experiment 1

{'biotope': {},

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

'number\_of\_organisms': 10,

'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': 10,

'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': 'built-in function',

'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': {

'hunting': {

'comparison': {

(aquí habría que expresar de alguna manera que la condición para que un bicho se

pueda comer a otro es que la fuerza (strength) del primero sea mayor que la del segundo)

}

},

'mutability': {

'strength': {

'increments': gaussian(mean=0, variance=0.01),

'mutation\_frequency': 0.05,

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

},

'photosynthesis\_capacity': {

'increments': gaussian(mean=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',

'number\_of\_organisms': 100,

'genes': {

'strength': gaussian(mean=10, variance=2),

'photosynthesis\_capacity': gausian(mean=20, variance=10),

'speed': discrete([

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

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

'generation': 0 },

'mutability': {'mutation\_frequency': 0.1} },

'status': {

'energy\_reserve': 100.0} },

{'category': 'No\_Mutants',

'number\_of\_organisms': 100,

'genes': {

'strength': gaussian(mean=10, variance=2),

'photosynthesis\_capacity': gausian(mean=20, variance=10),

'speed': discrete([

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

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

'generation': 0 },

'mutability': {'mutation\_frequency': 0.0} },

'status': {

'energy\_reserve': 100.0} } ],

'outlays': {

'hunting': linear\_func({

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

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

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

'moving': linear\_func({

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

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

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

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

'procreating': linear\_func({

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

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

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

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

'living': linear\_func({

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

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

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

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

},

'constraints': {

'hunting': {

'comparison': {

(aquí habría que expresar de alguna manera que la condición para que un bicho se

pueda comer a otro es que la fuerza (strength) del primero sea mayor que la del segundo)

}

},

'mutability': {

'strength': {

'increments': gaussian(mean=0, variance=0.01),

'mutation\_frequency': 0.05,

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

},

'photosynthesis\_capacity': {

'percentage\_increments': gaussian(mean=0, variance=0.15),

'mutation\_frequency': 0.001

},

'speed': probabilistic\_automaton({

'states': {0.0, 1.0},

'probabilities\_matrix': [

[0.99, 0.01]

[0.02, 0.98]]

}),

'generation': {

'increments': 1

'mutability': {

'percentage\_increments': gaussian(mean=0, variance=0.15),

'allowed\_interval': [0, 1]

}

}

}

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

# guerra por la supervivencia