

```
# template_params.py

import numpy as np
import pandas as pd
import os

# set up the landscape
b4 = np.vstack([np.linspace(1, 0, 50) for _ in range(50)])
af = np.vstack([np.linspace(1, 0.5, 50) for _ in range(50)])
stable = np.vstack([np.linspace(1, 0, 50) for _ in range(50)])
K = np.ones((50,50))

# NOTE: time_steps.CSV USED TO SET CLIMATE CHANGE
#       TO START AT TIME STEP 2500
#       AND FINISH AT TIME STEP 2750
if os.getcwd().split('/')[1] == 'home':
    steps = pd.read_csv('/home/deth/Desktop/CAL/research/projects/sim/'
                        'ch2/climate_change_adaptation_and_genomic_arch/sim/'
                        'time_steps.CSV'))
else:
    steps = pd.read_csv('/global/scratch/users/drewhart/'
                        'ch2/climate_change_adaptation_and_genomic_arch/sim/'
                        'time_steps.CSV'))

# set time when environmental change begins
change_T = int(steps[steps['name']=='start']['num'].values[0])
# set time when environmental change ends
T = int(steps[steps['name']=='end']['num'].values[0])

# show the landscape, for debugging, if requested
debug_landscape = False
if debug_landscape:
    import matplotlib.pyplot as plt
    fig = plt.figure()
    ax1 = fig.add_subplot(221)
    im1 = ax1.imshow(b4, cmap='spring', vmin=0, vmax=1)
    plt.colorbar(im1)
    ax3 = fig.add_subplot(223)
    im3 = ax3.imshow(af, cmap='spring', vmin=0, vmax=1)
    plt.colorbar(im3)
    ax2 = fig.add_subplot(222)
    im2 = ax2.imshow(stable, cmap='winter', vmin=0, vmax=1)
    plt.colorbar(im2)
    ax4 = fig.add_subplot(224)
    im4 = ax4.imshow(K, cmap='autumn', vmin=0, vmax=1)
    plt.colorbar(im4)
    plt.show()

# This is a parameters file generated by Geonomics
# (by the gnmx.make_parameters_file() function).
```

[illegible]

```

#####

#####
#### LANDSCAPE ####
#####
'landscape': {

#####
#### main ####
#####
'main': {
    #x,y (a.k.a. j,i) dimensions of the Landscape
    'dim': (50,50),
    #x,y resolution of the Landscape
    'res': (1,1),
    #x,y coords of upper-left corner of the Landscape
    'ulc': (0,0),
    #projection of the Landscape
    'prj': None,
}, # <END> 'main'

#####
#### layers ####
#####
'layers': {

    #layer name (LAYER NAMES MUST BE UNIQUE!)
    'shift': {

#-----#
#--- layer num. 0: init parameters ---#
#-----#

        #initiating parameters for this layer
        'init': {

            #parameters for a 'defined'-type Layer
            'defined': {
                #raster to use for the Layer
                'rast': b4,
                #point coordinates
                'pts': None,
                #point values
                'vals': None,
                #interpolation method {None, 'linear', 'cubic',
                # 'nearest'}
                'interp_method': None,

            }, # <END> 'defined'

        }, # <END> 'init'

#-----#
#--- layer num. 0: change parameters ---#
#-----#

        #landscape-change events for this Layer
        'change': {

            0: {
                #array or file for final raster of event, or directory
                #of files for each stepwise change in event
                'change_rast': af,
                #starting timestep of event
                'start_t': change_T,
                #ending timestep of event
                'end_t': T,
                #number of stepwise changes in event
                'n_steps': T-change_T,
            }, # <END> event 0

        }, # <END> 'change'

    }, # <END> layer num. 0

```

```

#layer name (LAYER NAMES MUST BE UNIQUE!)
'stable': {

#-----#
#--- layer num. 1: init parameters ---#
#-----#

#initiating parameters for this layer
'init': {

#parameters for a 'defined'-type Layer
'defined': {
#raster to use for the Layer
'rast': stable,
#point coordinates
'pts': None,
#point values
'vals': None,
#interpolation method {None, 'linear', 'cubic',
#'nearest'}
'interp_method': None,

}, # <END> 'defined'

}, # <END> 'init'

}, # <END> layer num. 1

#layer name (LAYER NAMES MUST BE UNIQUE!)
'K': {

#-----#
#--- layer num. 2: init parameters ---#
#-----#

#initiating parameters for this layer
'init': {

#parameters for a 'defined'-type Layer
'defined': {
#raster to use for the Layer
'rast': K,
#point coordinates
'pts': None,
#point values
'vals': None,
#interpolation method {None, 'linear', 'cubic',
#'nearest'}
'interp_method': None,

}, # <END> 'defined'

}, # <END> 'init'

}, # <END> layer num. 2

#layer name (LAYER NAMES MUST BE UNIQUE!)
'move': {

#-----#
#--- layer num. 2: init parameters ---#
#-----#

#initiating parameters for this layer
'init': {

#parameters for a 'defined'-type Layer
'defined': {
#raster to use for the Layer
'rast': np.ones((50,50)),

```

```

        #point coordinates
        'pts':          None,
        #point values
        'vals':         None,
        #interpolation method {None, 'linear', 'cubic',
        #'nearest'}
        'interp_method': None,

    }, # <END> 'defined'

}, # <END> 'init'

}

```

#### NOTE: Individual Layers' sections can be copy-and-pasted (and  
 #### assigned distinct keys and names), to create additional Layers.

```

    } # <END> 'layers'

}, # <END> 'landscape'

```

#####

```

#####
#### COMMUNITY ####
#####

```

```

'comm': {

    'species': {

        #species name (SPECIES NAMES MUST BE UNIQUE!)
        'spp_0': {

            #-----#
            #--- spp num. 0: init parameters ---#
            #-----#

            'init': {
                #starting number of individs
                'N':          1000,
                #carrying-capacity Layer name
                'K_layer':    'K',
                #multiplicative factor for carrying-capacity layer
                'K_factor':   2.5,
            }, # <END> 'init'

            #-----#
            #--- spp num. 0: mating parameters ---#
            #-----#

            'mating' : {
                #age(s) at sexual maturity (if tuple, female first)
                'repro_age':  0,
                #whether to assign sexes
                'sex':        False,
                #ratio of males to females
                'sex_ratio':  1/1,
                #whether P(birth) should be weighted by parental dist
                'dist_weighted_birth': False,
                #intrinsic growth rate
                'R':          0.5,
                #intrinsic birth rate (MUST BE 0<=b<=1)
                'b':          0.5,
                #expectation of distr of n offspring per mating pair
                'n_births_distr_lambda': 1,
                #whether n births should be fixed at n_births_dist_lambda
                'n_births_fixed': True,
                #radius of mate-search area
                'mating_radius': 5,
                #whether individs should choose nearest neighs as mates
                'choose_nearest_mate': False,
            }
        }
    }
}

```

```

        #whether mate-choice should be inverse distance-weighted
        'inverse_dist_mating':      False,
    }, # <END> 'mating'

#-----#
#--- spp num. 0: mortality parameters ---#
#-----#

    'mortality'      : {
        #maximum age
        'max_age':      None,
        #min P(death) (MUST BE 0<=d_min<=1)
        'd_min':        0,
        #max P(death) (MUST BE 0<=d_max<=1)
        'd_max':        1,
        #width of window used to estimate local pop density
        'density_grid_window_width': None,
    }, # <END> 'mortality'

#-----#
#--- spp num. 0: movement parameters ---#
#-----#

    'movement': {
        #whether or not the species is mobile
        'move':      True,
        #mode of distr of movement direction
        'direction_distr_mu':      0,
        #concentration of distr of movement direction
        'direction_distr_kappa':  0,
        #mean of distr of movement distance
        'movement_distance_distr_param1':      0.25,
        #variance of distr of movement distance
        'movement_distance_distr_param2':      0.5,
        #movement distance distr to use ('lognormal','levy','wald')
        'movement_distance_distr':      'wald',
        #mean of distr of dispersal distance
        'dispersal_distance_distr_param1':      0.5,
        #variance of distr of dispersal distance
        'dispersal_distance_distr_param2':      0.5,
        #dispersal distance distr to use ('lognormal','levy','wald')
        'dispersal_distance_distr':      'wald',
    }, # <END> 'movement'

#-----#
#--- spp num. 0: genomic architecture parameters ---#
#-----#

    'gen_arch': {
        #file defining custom genomic arch
        'gen_arch_file':      None,
        #num of loci
        'L':      1000,
        #value to use for fixed starting allele freqs (None to draw)
        'start_p_fixed':      0.5,
        #whether to start neutral locus freqs at 0
        'start_neut_zero':      True,
        #genome-wide per-base neutral mut rate (0 to disable)
        'mu_neut':      0,
        #genome-wide per-base deleterious mut rate (0 to disable)
        'mu_delet':      0,
        #shape of distr of deleterious effect sizes
        'delet_alpha_distr_shape': 0.2,
        #scale of distr of deleterious effect sizes
        'delet_alpha_distr_scale': 0.2,
        #NOTE: MAIN SCRIPT OVERRIDES THE FOLLOWING TWO PARAMS
        #TO SET RECOMBINATION RATES TO A FIXED VALUE OF
        #0.5, 0.05, OR 0.005 FOR INDEPENDENT,
        #WEAK, OR STRONG LINKAGE VALUES
        #alpha of distr of recomb rates
        'r_distr_alpha':      1000,
        #beta of distr of recomb rates
        'r_distr_beta':      1e3,
    },

```

```

#whether loci should be dominant (for allele '1')
'dom': False,
#whether to allow pleiotropy
'pleiotropy': False,
#custom fn for drawing recomb rates
'recomb_rate_custom_fn': None,
#number of recomb paths to hold in memory
'n_recomb_paths_mem': int(1e4),
#total number of recomb paths to simulate
'n_recomb_paths_tot': int(1e5),
    #num of crossing-over events (i.e. recombs) to simulate
'n_recomb_sims': 100_000,
#whether to generate recombination paths at each timestep
'allow_ad_hoc_recomb': False,
#whether to save mutation logs
'mut_log': False,
    #whether to jitter recomb bps, to correctly track num_trees
'jitter_breakpoints': False,
    #whether to use tskit (to record full spatial pedigree)
'use_tskit': True,
    #time step interval for simplification of tskit tables
'tskit_simp_interval': 100,

'traits': {

    #-----#
    #---trait 0 parameters ---#
    #-----#
    #trait name (TRAIT NAMES MUST BE UNIQUE!)
    'trait_0': {
        #trait-selection Layer name
        'layer': 'shift',
        #polygenic selection coefficient
        'phi': 1,
        #NOTE: MAIN SCRIPT CHANGES NEXT PARAM TO 4, 20, OR 100
        #FOR LOW-REDUNDANCY SCENARIOS OF DIFF. POLYGENICITY,
        #OR 8, 40, OR 200 FOR HIGH-REDUNDANCY SCENARIOS
        #number of loci underlying trait
        'n_loci': 50,
        #mutation rate at loci underlying trait
        'mu': 0,
        #mean of distr of effect sizes
        'alpha_distr_mu': 0,
        #variance of distr of effect size
        'alpha_distr_sigma': 0,
        #max allowed magnitude for an alpha value
        'max_alpha_mag': None,
        #curvature of fitness function
        'gamma': 1,
        #whether the trait is universally advantageous
        'univ_adv': False
    }, # <END> trait 0

    #-----#
    #---trait 1 parameters ---#
    #-----#
    #trait name (TRAIT NAMES MUST BE UNIQUE!)
    'trait_1': {
        #trait-selection Layer name
        'layer': 'stable',
        #polygenic selection coefficient
        'phi': 1,
        #NOTE: MAIN SCRIPT CHANGES NEXT PARAM TO 4, 20, OR 100
        #FOR LOW-REDUNDANCY SCENARIOS OF DIFF. POLYGENICITY,
        #OR 8, 40, OR 200 FOR HIGH-REDUNDANCY SCENARIOS
        #number of loci underlying trait
        'n_loci': 50,
        #mutation rate at loci underlying trait
        'mu': 0,
        #mean of distr of effect sizes
        'alpha_distr_mu': 0,
        #variance of distr of effect size
        'alpha_distr_sigma': 0,
        #max allowed magnitude for an alpha value

```

```

        'max_alpha_mag':      None,
        #curvature of fitness function
        'gamma':              1,
        #whether the trait is universally advantageous
        'univ_adv':           False
    }, # <END> trait 1

```

#### NOTE: Individual Traits' sections can be copy-and-pasted (and  
 #### assigned distinct keys and names), to create additional Traits.

```

        }, # <END> 'traits'
    }, # <END> 'gen_arch'

}, # <END> spp num. 0

```

#### NOTE: individual Species' sections can be copy-and-pasted (and  
 #### assigned distinct keys and names), to create additional Species.

```

    }, # <END> 'species'

}, # <END> 'comm'

```

#####

#####  
 #### MODEL ####  
 #####

```

'model': {
    # NOTE: NEXT PARAM OVERRIDDEN BY MAIN SCRIPT
    #total Model runtime (in timesteps)
    'T':      100000,
    #min burn-in runtime (in timesteps)
    'burn_T': 30,
    #seed number
    'num':     None,

```

```

#####
#### iterations parameters ####
#####
'its': {
    #num iterations
    'n_its':      1,
    #whether to randomize Landscape each iteration
    'rand_landscape': False,
    #whether to randomize Community each iteration
    'rand_comm':   False,
    #whether to randomize GenomicArchitectures each iteration
    'rand_genarch': True,
    #whether to burn in each iteration
    'repeat_burn': False,
}, # <END> 'iterations'

```

```

#####
#### data-collection parameters ####
#####
'data': {
    'sampling': {
        #sampling scheme {'all', 'random', 'point', 'transect'}
        'scheme':      'all',
        #sample size at each point, for point & transect sampling
        'n':            1000,
        #coords of collection points, for point sampling
        'points':        None,
        #coords of transect endpoints, for transect sampling
        'transect_endpoints': None,
    }

```

```

#num points along transect, for transect sampling
'n_transect_points':    None,
#collection radius around points, for point & transect sampling
'radius':               None,
#when to collect data
'when':                 [change_T-1,
                        int((change_T-1+T-1)/2),
                        T-1],
#whether to save current Layers when data is collected
'include_landscape':    False,
#whether to include fixed loci in VCF files
'include_fixed_sites':  True,
},
'format': {
  #format for genetic data {'vcf', 'fasta'}
  'gen_format':          ['vcf'],
  #format for vector geodata {'csv', 'shapefile', 'geojson'}
  'geo_vect_format':     'csv',
  #format for raster geodata {'geotiff', 'txt'}
  'geo_rast_format':     'geotiff',
  #format for files containing non-neutral loci
  'nonneut_loc_format':  'csv',
},
}, #<END> 'data'

} # <END> 'model'

} # <END> params

```