
gpmap Documentation

Release 0.1

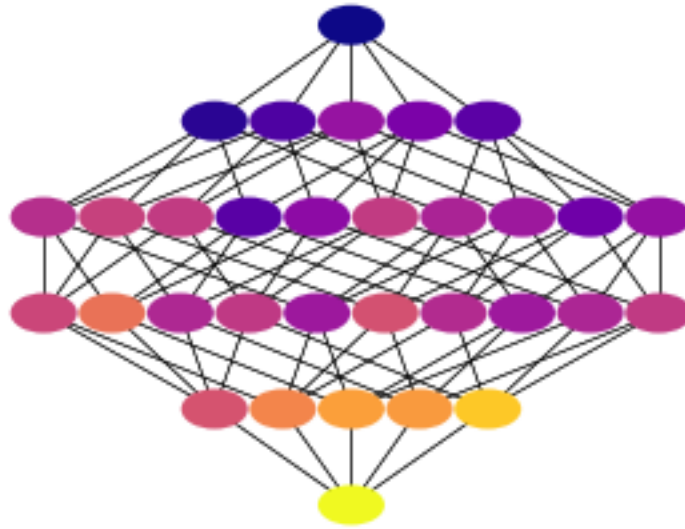
Zach Sailer

May 10, 2018

Contents

1	Basic Example	3
2	Documentation	5
2.1	Simulating genotype-phenotype maps	5
2.2	Reading/Writing	6
2.3	API Documentation	8
3	Indices and tables	9

The Pandas DataFrame for genotype-phenotype (GP) map data.



The `GenotypePhenotypeMap` is a core object for a suite of packages written in the [Harms Lab](#). It organizes and standardizes genotype-phenotype map data.

CHAPTER 1

Basic Example

```
# Import the GenotypePhenotypeMap
from gpmap import GenotypePhenotypeMap

# The data
wildtype = 'AA'
genotypes = ['AA', 'AT', 'TA', 'TT']
phenotypes = [0.1, 0.5, 0.2, 0.8]
stdeviations = [0.05, 0.05, 0.05, 0.05]

# Initialize a GenotypePhenotype object
gpm = GenotypePhenotypeMap(wildtype, genotypes, phenotypes,
                           stdeviations=stdeviations)

# Show the dataFrame
gpm.data
```

	genotypes	n_replicates	phenotypes	stdeviations	binary
0	AA	1	0.1	0.05	00
1	AT	1	0.5	0.05	01
2	TA	1	0.2	0.05	10
3	TT	1	0.8	0.05	11

2.1 Simulating genotype-phenotype maps

The GPMMap package comes with a suite of objects to simulate genotype-phenotype maps following models in the literature. They are found in the `gpmmap.simulate` module.

All Simulation objects inherit the `GenotypePhenotypeMap` object as their base class. Thus, anything you can do with a `GenotypePhenotypeMap`, you can do with the simulation objects.

2.1.1 NK landscape

Construct a genotype-phenotype map using Kauffman's NK Model.¹ The NK fitness landscape is created using a table with binary, length- K , sub-sequences mapped to random values. All genotypes are binary with length N . The fitness of a genotype is constructed by summing the values of all sub-sequences that make up the genotype using a sliding window across the full genotypes.

For example, imagine an NK simulation with $N = 5$ and $K = 2$. To construct the fitness for the 01011 genotype, select the following sub-sequences from an NK table: "01", "10", "01", "11", "10". Sum their values together.

```
# import the NKSimulation class
from gpmmap.simulate import NKSimulation

# Create an instance of the model. Using `from_length` makes this easy.
gpm = NKSimulation.from_length(6, K=3)
```

2.1.2 House of Cards landscape

Construct a 'House of Cards' fitness landscape. This is a limit of the NK model where $K = N$. It represents a fitness landscape with maximum roughness.

¹ Kauffman, Stuart A., and Edward D. Weinberger. "The NK model of rugged fitness landscapes and its application to maturation of the immune response." *Journal of theoretical biology* 141.2 (1989): 211-245.

```
# import the HouseOfCardsSimulation class
from gpmap.simulate import HouseOfCardsSimulation

# Create an instance of the model. Using `from_length` makes this easy.
gpm = HouseOfCardsSimulation.from_length(6)
```

2.1.3 Mount Fuji landscape

Construct a genotype-phenotype map from a Mount Fuji model.²

A Mount Fuji sets a “global” fitness peak (max) on a single genotype in the space. The fitness goes down as a function of hamming distance away from this genotype, called a “fitness field”. The strength (or scale) of this field is linear and depends on the parameters *field_strength*.

Roughness can be added to the Mount Fuji model using a random *roughness* parameter. This assigns a random roughness value to each genotype.

$$f(g) = \nu(g) + c \cdot d(g_0, g)$$

where ν is the roughness parameter, c is the field strength, and d is the hamming distance between genotype g and the reference genotype.

```
# import the HouseOfCardsSimulation class
from gpmap.simulate import MountFujiSimulation

# Create an instance of the model. Using `from_length` makes this easy.
gpm = MountFujiSimulation.from_length(6
    roughness_width=0.5,
    roughness_dist='normal'
)
```

2.1.4 References

2.2 Reading/Writing

The `GenotypePhenotypeMap` object is a Pandas `DataFrame` at its core. Most tabular formats (i.e. Excel files, csv, tsv, ...) can be read/written.

2.2.1 Excel Spreadsheets

Excel files are supported through the `read_excel` method. This method requires *genotypes* and *phenotypes* columns, and can include *n_replicates* and *stdeviations* as optional columns. All other columns are ignored.

Example: Excel spreadsheet file (“data.xlsx”)

Read the spreadsheet directly into the `GenotypePhenotypeMap`.

```
from gpmap import GenotypePhenotypeMap

gpm = GenotypePhenotypeMap.read_excel(wildtype="PTEE", filename="data.xlsx")
```

² Szendro, Ivan G., et al. “Quantitative analyses of empirical fitness landscapes.” *Journal of Statistical Mechanics: Theory and Experiment* 2013.01 (2013): P01005.

2.2.2 CSV File

CSV files are supported through the `read_excel` method. This method requires *genotypes* and *phenotypes* columns, and can include *n_replicates* and *stdeviations* as optional columns. All other columns are ignored.

Example: CSV File

Read the csv directly into the GenotypePhenotypeMap.

```
from gpmap import GenotypePhenotypeMap

gpm = GenotypePhenotypeMap.read_csv(wildtype="PTEE", filename="data.csv")
```

2.2.3 JSON Format

The only keys recognized by the json reader are:

1. *genotypes*
2. *phenotypes*
3. *stdeviations*
4. *mutations*
5. *n_replicates*

All other keys are ignored in the epistasis models. You can keep other metadata stored in the JSON, but it won't be appended to the epistasis model object.

```
{
  "genotypes" : [
    '000',
    '001',
    '010',
    '011',
    '100',
    '101',
    '110',
    '111'
  ],
  "phenotypes" : [
    0.62344582,
    0.87943151,
    -0.11075798,
    -0.59754471,
    1.4314798,
    1.12551439,
    1.04859722,
    -0.27145593
  ],
  "stdeviations" : [
    0.01,
    0.01,
    0.01,
    0.01,
    0.01,
    0.01,
    0.01,
    0.01,
  ]
}
```

(continues on next page)

(continued from previous page)

```
    0.01,  
  ],  
  "mutations" : {  
    0 : ["0", "1"],  
    1 : ["0", "1"],  
    2 : ["0", "1"],  
  }  
  "n_replicates" : 12,  
  "title" : "my data",  
  "description" : "a really hard experiment"  
}
```

2.3 API Documentation

The `GenotypePhenotypeMap` is the main entry point to the `gpmmap` package. Load in your data using the `read` methods attached to this object. The following subpackages include various objects to analyze this object.

2.3.1 Subpackages

`gpmmap.errors` module

`gpmmap.sample` module

`gpmmap.stats` module

`gpmmap.utils` module

`gpmmap.simulate`

`gpmmap.simulate.base` module

`gpmmap.simulate.fuji` module

`gpmmap.simulate.hoc` module

`gpmmap.simulate.nk` module

Module contents

2.3.2 GenotypePhenotypeMap

CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`