

Progress on Work Package 1C

12 May 2021

Objectives of Work Package 1C

- ▶ Determine how the scale of landscape diversity affects the maintenance of susceptibility alleles

Objectives of Work Package 1C

- ▶ Determine how the scale of landscape diversity affects the maintenance of susceptibility alleles
- ▶ Use individual-based modelling (i.e., agent-based modelling) to simulate genetic architecture and landscapes

Objectives of Work Package 1C

- ▶ Determine how the scale of landscape diversity affects the maintenance of susceptibility alleles
- ▶ Use individual-based modelling (i.e., agent-based modelling) to simulate genetic architecture and landscapes
- ▶ Simulate dynamic application of biopesticide application and crop rotation *in silico*

Objectives of Work Package 1C

- ▶ Determine how the scale of landscape diversity affects the maintenance of susceptibility alleles
- ▶ Use individual-based modelling (i.e., agent-based modelling) to simulate genetic architecture and landscapes
- ▶ Simulate dynamic application of biopesticide application and crop rotation *in silico*
- ▶ Parameterise with data from *Helicoverpa armigera* and data from Work Package 1A

Simulate genetic architecture

##	rG	Diff	Diff2	TRT1	TRT2
## 1	0.47205394	Isolate	Control-Infected	SC	SB
## 2	0.54752141	Isolate	Control-Infected	SC	SM
## 3	0.31357864	Crop	Uninfected	SC	MC
## 4	-0.33459966	Both	Control-Infected	SC	MB
## 5	0.06347942	Both	Control-Infected	SC	MM
## 6	-0.38837056	Crop	Uninfected	SC	TC
## 7	0.06358750	Both	Control-Infected	SC	TB
## 8	-0.09823783	Both	Control-Infected	SC	TM
## 9	0.53766718	Isolate	Isolate Genus	SB	SM
## 10	0.17474769	Both	Control-Infected	SB	MC
## 11	0.26738775	Crop	Infected	SB	MB
## 12	-0.10436088	Both	Isolate Genus	SB	MM
## 13	-0.09942243	Both	Control-Infected	SB	TC
## 14	-0.14326498	Crop	Infected	SB	TB
## 15	-0.42090644	Both	Isolate Genus	SB	TM
## 16	-0.05742982	Both	Control-Infected	SM	MC

Simulate genetic architecture

Correlation matrix derived from Work Package 1A

##		SC	SB	SM	MC	MB	MM	TC	TB	TM
##	SC	1.000	0.472	0.548	0.314	-0.335	0.063	-0.388	0.064	-0.098
##	SB	0.472	1.000	0.538	0.175	0.267	-0.104	-0.099	-0.143	-0.421
##	SM	0.548	0.538	1.000	-0.057	0.137	-0.305	-0.372	-0.010	0.023
##	MC	0.314	0.175	-0.057	1.000	0.551	0.337	-0.268	-0.033	0.021
##	MB	-0.335	0.267	0.137	0.551	1.000	0.518	0.020	-0.275	0.038
##	MM	0.063	-0.104	-0.305	0.337	0.518	1.000	0.010	-0.260	-0.156
##	TC	-0.388	-0.099	-0.372	-0.268	0.020	0.010	1.000	0.250	0.388
##	TB	0.064	-0.143	-0.010	-0.033	-0.275	-0.260	0.250	1.000	0.597
##	TM	-0.098	-0.421	0.023	0.021	0.038	-0.156	0.388	0.597	1.000

Need to create a quantitative genetic architecture that produces these trait correlations from allele values

Traits

1

2

3

4

5

6

7

8

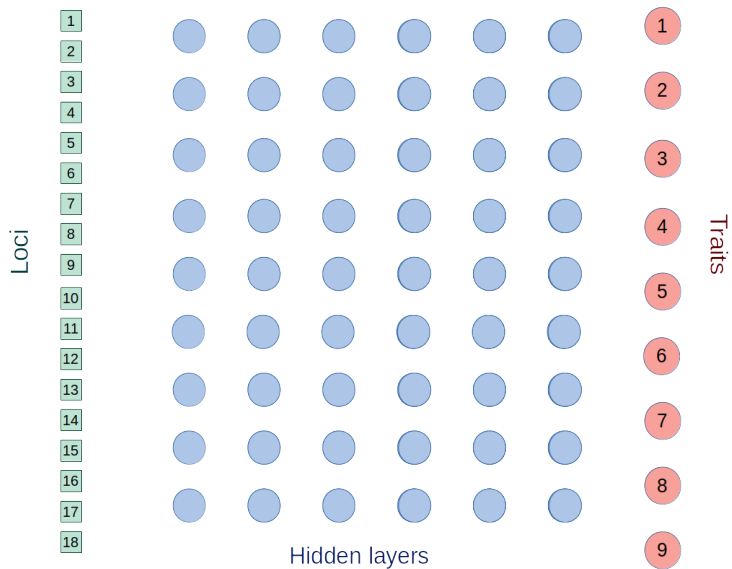
9

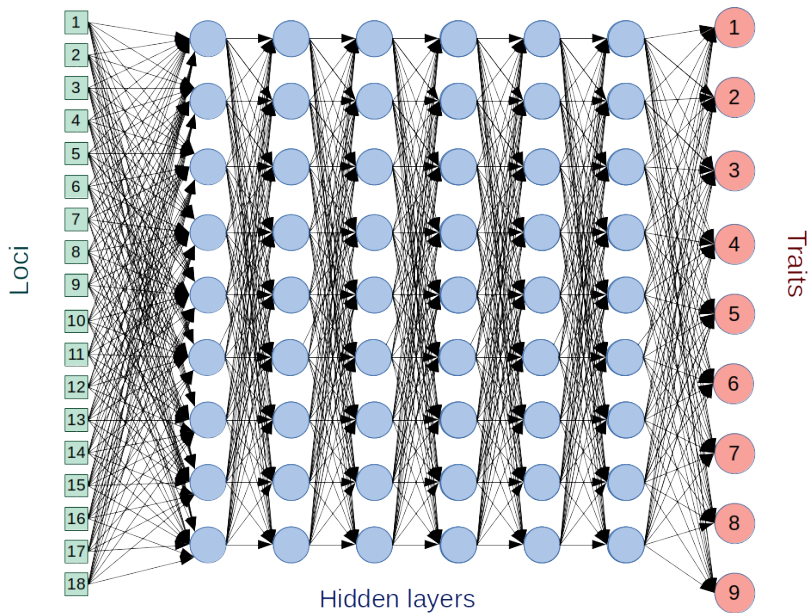
Loci

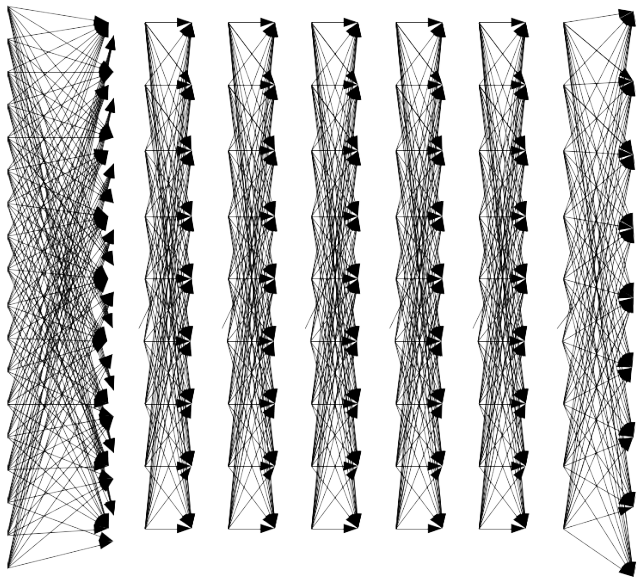
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18

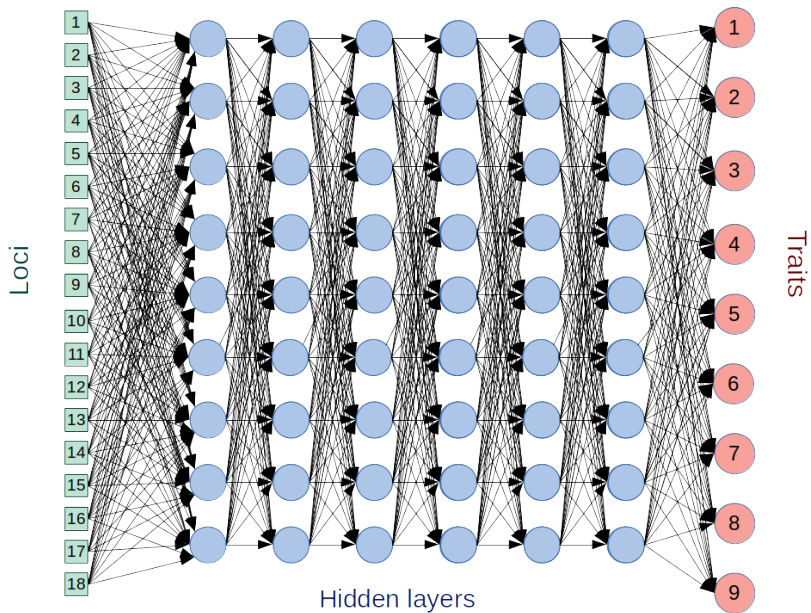
Traits

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9









Getting from loci to traits

Use an evolutionary algorithm to find arrow values

1. Initialise a population of N random networks

Getting from loci to traits

Use an evolutionary algorithm to find arrow values

1. Initialise a population of N random networks
2. Crossover between networks (swap values)

Getting from loci to traits

Use an evolutionary algorithm to find arrow values

1. Initialise a population of N random networks
2. Crossover between networks (swap values)
3. Mutation of network values

Getting from loci to traits

Use an evolutionary algorithm to find arrow values

1. Initialise a population of N random networks
2. Crossover between networks (swap values)
3. Mutation of network values
4. Get network fitnesses (how close to empirical matrix?)

Getting from loci to traits

Use an evolutionary algorithm to find arrow values

1. Initialise a population of N random networks
2. Crossover between networks (swap values)
3. Mutation of network values
4. Get network fitnesses (how close to empirical matrix?)
5. Tournament to select highest fitness networks

Getting from loci to traits

Use an evolutionary algorithm to find arrow values

1. Initialise a population of N random networks
2. Crossover between networks (swap values)
3. Mutation of network values
4. Get network fitnesses (how close to empirical matrix?)
5. Tournament to select highest fitness networks
6. High fitness networks reproduce

Getting from loci to traits

Use an evolutionary algorithm to find arrow values

1. Initialise a population of N random networks
2. Crossover between networks (swap values)
3. Mutation of network values
4. Get network fitnesses (how close to empirical matrix?)
5. Tournament to select highest fitness networks
6. High fitness networks reproduce
7. N offspring networks replace parents

Getting from loci to traits

Use an evolutionary algorithm to find arrow values

1. Initialise a population of N random networks
2. Crossover between networks (swap values)
3. Mutation of network values
4. Get network fitnesses (how close to empirical matrix?)
5. Tournament to select highest fitness networks
6. High fitness networks reproduce
7. N offspring networks replace parents
8. Repeat 2-7 until termination criteria satisfied

End result is a set of network values that gets close the empirical matrix when loci have standard random normal values

The helicoverpa package

An R function `mine_gmatrix` can find network values given:

- ▶ A square empirical matrix
- ▶ Arbitrary loci number
- ▶ Arbitrary number of hidden layers
- ▶ Various algorithm input options

```
eg_net <- mine_gmatrix(gmatrix = gmt, loci = 18,  
                        layers = 6, mu_rate = 0.1,  
                        mu_sd = 0.04, net_N = 8000,  
                        test_N = 1000);
```

The helicoverpa package

```
=====
Initialising gmatrix mining...
=====
```

Gen: 0	Stress: -1.300040	Min: -1.300051
Gen: 1	Stress: -1.300052	Min: -1.300150
Gen: 2	Stress: -1.300119	Min: -1.300985
Gen: 3	Stress: -1.300434	Min: -1.303620
Gen: 4	Stress: -1.301580	Min: -1.311596
Gen: 5	Stress: -1.305397	Min: -1.332303
Gen: 6	Stress: -1.316018	Min: -1.388752
Gen: 7	Stress: -1.342745	Min: -1.537480
Gen: 8	Stress: -1.400662	Min: -1.643793
Gen: 9	Stress: -1.483798	Min: -1.720355
Gen: 10	Stress: -1.552275	Min: -1.832071
Gen: 11	Stress: -1.580273	Min: -1.863808

Building the empirical matrix from random normal loci

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
## [1,] 1.000 0.610 0.809 0.334 -0.214 -0.303 -0.746 -0.220 -0.399
## [2,] 0.610 1.000 0.707 0.391 0.347 -0.146 -0.550 -0.713 -0.768
## [3,] 0.809 0.707 1.000 0.122 -0.037 -0.449 -0.739 -0.275 -0.361
## [4,] 0.334 0.391 0.122 1.000 0.646 0.447 -0.647 -0.499 -0.470
## [5,] -0.214 0.347 -0.037 0.646 1.000 0.681 -0.364 -0.713 -0.583
## [6,] -0.303 -0.146 -0.449 0.447 0.681 1.000 -0.131 -0.475 -0.468
## [7,] -0.746 -0.550 -0.739 -0.647 -0.364 -0.131 1.000 0.411 0.528
## [8,] -0.220 -0.713 -0.275 -0.499 -0.713 -0.475 0.411 1.000 0.891
## [9,] -0.399 -0.768 -0.361 -0.470 -0.583 -0.468 0.528 0.891 1.000
```


Building the empirical matrix from random normal loci

##		[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
##	[1,]	1.000	0.610	0.809	0.334	-0.214	-0.303	-0.746	-0.220	-0.399
##	[2,]	0.610	1.000	0.707	0.391	0.347	-0.146	-0.550	-0.713	-0.768
##	[3,]	0.809	0.707	1.000	0.122	-0.037	-0.449	-0.739	-0.275	-0.361
##	[4,]	0.334	0.391	0.122	1.000	0.646	0.447	-0.647	-0.499	-0.470
##	[5,]	-0.214	0.347	-0.037	0.646	1.000	0.681	-0.364	-0.713	-0.583
##	[6,]	-0.303	-0.146	-0.449	0.447	0.681	1.000	-0.131	-0.475	-0.468
##	[7,]	-0.746	-0.550	-0.739	-0.647	-0.364	-0.131	1.000	0.411	0.528
##	[8,]	-0.220	-0.713	-0.275	-0.499	-0.713	-0.475	0.411	1.000	0.891
##	[9,]	-0.399	-0.768	-0.361	-0.470	-0.583	-0.468	0.528	0.891	1.000

##		SC	SB	SM	MC	MB	MM	TC	TB	TM
##	[1,]	1.000	0.472	0.548	0.314	-0.335	0.063	-0.388	0.064	-0.098
##	[2,]	0.472	1.000	0.538	0.175	0.267	-0.104	-0.099	-0.143	-0.421
##	[3,]	0.548	0.538	1.000	-0.057	0.137	-0.305	-0.372	-0.010	0.023
##	[4,]	0.314	0.175	-0.057	1.000	0.551	0.337	-0.268	-0.033	0.021
##	[5,]	-0.335	0.267	0.137	0.551	1.000	0.518	0.020	-0.275	0.038
##	[6,]	0.063	-0.104	-0.305	0.337	0.518	1.000	0.010	-0.260	-0.156
##	[7,]	-0.388	-0.099	-0.372	-0.268	0.020	0.010	1.000	0.250	0.388
##	[8,]	0.064	-0.143	-0.010	-0.033	-0.275	-0.260	0.250	1.000	0.597
##	[9,]	-0.098	-0.421	0.023	0.021	0.038	-0.156	0.388	0.597	1.000

Next steps

- ▶ Further improve the evolutionary algorithm
- ▶ Model *Helicoverpa* with random initial alleles
- ▶ Simulate *Helicoverpa* on a landscape
- ▶ Observe evolution given changing pesticide and crop scenarios
- ▶ Complete R package and publications

```
mg2[[8]];
```

```
## [1] -5.166129
```