

Introduction

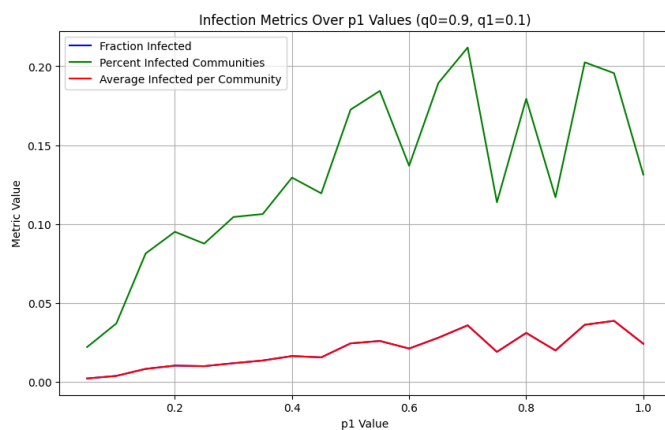
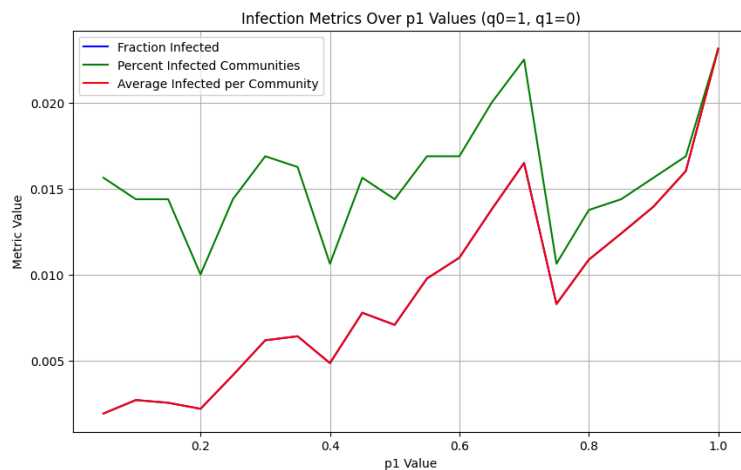
This project aims to develop adaptive group testing algorithms that utilize non-binary test outputs to efficiently identify infected individuals within a population. The first part focuses on utilizing tests that provide the exact number of infected individuals (T1) or categorize the infection count into ranges (T2). In the second part, we will adapt these algorithms to populations with community structures, recognizing that infections often occur in correlated groups, such as families. By comparing community-aware testing strategies with traditional random interleaving methods, we aim to demonstrate significant reductions in the number of tests needed.

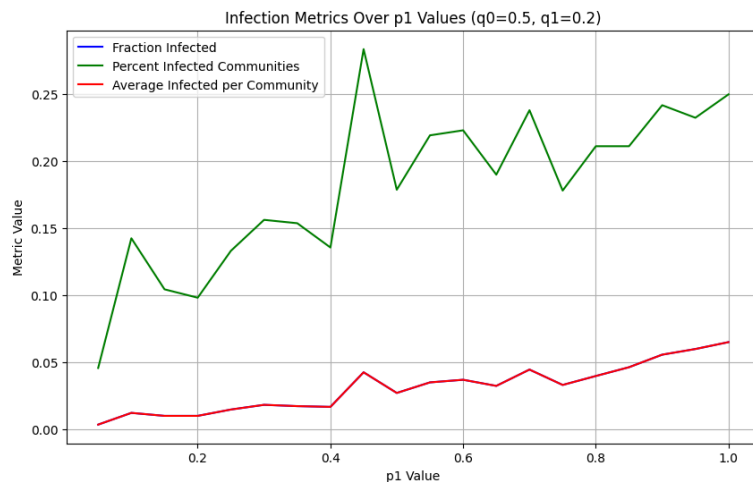
Infection

Table 2: Parameters for synthetic datasets.

N	M	q_0, q_1	p_0	p_1	time steps
256	16	(1, 0)	0.001	[0.05:0.1:1]	2
	16	(0.9, 0.1)	0.001	[0.05:0.1:1]	2
	16	(0.5, 0.3)	0.001	[0.05:0.1:1]	2

Infection data for the table above:





This data appropriately shows the results of increasing q_1 on the percent of infected communities as a higher inter-community connection reasonably allows for the spread of infection to more communities that did not initially have the infection.

Testing

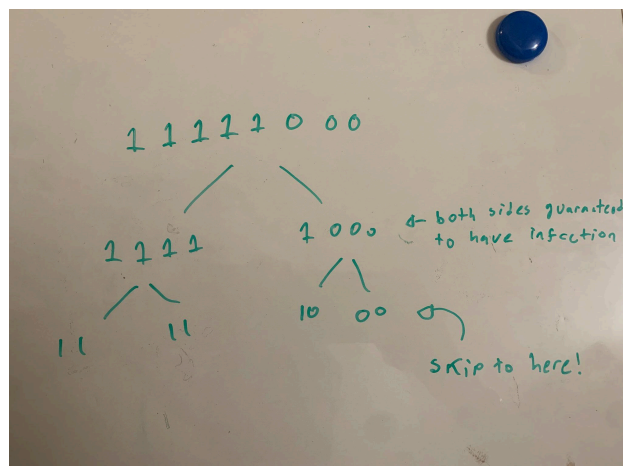
1. Design Adaptive algorithms with Enhanced Test Information

T1 test: Output the exact number of infected individuals in a group.

Traditional binary group testing only detects if at least one infected individual is presented in the group. When applying the **Pigeonhole Principle**, we can efficiently skip one or potentially multiple steps of the binary testing.

For example, suppose there are 16 people in the group, and 9 people are infected. In that case, we can skip the first step of splitting them into 2 groups of 8 people, because it is guaranteed that at least one individual will be infected after splitting them into 2 groups equally. Instead, we split them into 4 groups equally, thus skipping 1 stage and 2 tests in the algorithm, making it more efficient.

The general idea is that suppose there are n people in the group, and the number of infected people is more than $\frac{n}{2}$, then we can skip the first step of splitting them in half. Similarly, when the number of infected people is more than $\frac{3n}{4}$, we can skip the first 2 steps and directly split them into 8 groups equally. Therefore, if the number of infected people is more than $\frac{2^k - 1}{2^k}n$, where $k > 0$, we can skip k steps in binary splitting. This concept is illustrated below:



Similarly, if there are more normal people in the group compared to infected people, we can apply the same logic and find out who are the healthy people, and whoever is left will be the ones infected. This way we can guarantee skipping stages, and thus saving a significant amount of tests when performing the algorithm.

We also implement a pretty robust algorithm for if there is only one infected individual. Here we can use $\log_2(N)$ tests to guarantee we find the infected individual. Essentially, we assign each individual from the test a unique $\log_2(N)$ size binary code where a 1 represents that that test is applied to them. This is commonly known as the Rats and Poison solution, and is illustrated below:

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8
T_1	0	0	1	1	1	0	0	1
T_2	0	0	0	0	1	1	1	1
T_3	0	1	1	0	0	0	1	1

Here, T_1 is administered to Persons 3, 4, 5, and 8; T_2 to Persons 5, 6, 7, and 8; and T_3 to Persons 2, 3, 7, and 8. The results allow us to determine the infected individual as follows:

- If only T_1 is positive, Person 4 is infected.
- If only T_2 is positive, Person 6 is infected.
- If only T_3 is positive, Person 2 is infected.
- If both T_1 and T_2 are positive, Person 5 is infected.
- If both T_1 and T_3 are positive, Person 3 is infected.
- If both T_2 and T_3 are positive, Person 7 is infected.
- If all three tests are positive, Person 8 is infected.
- If none of the tests are positive, Person 1 is infected.

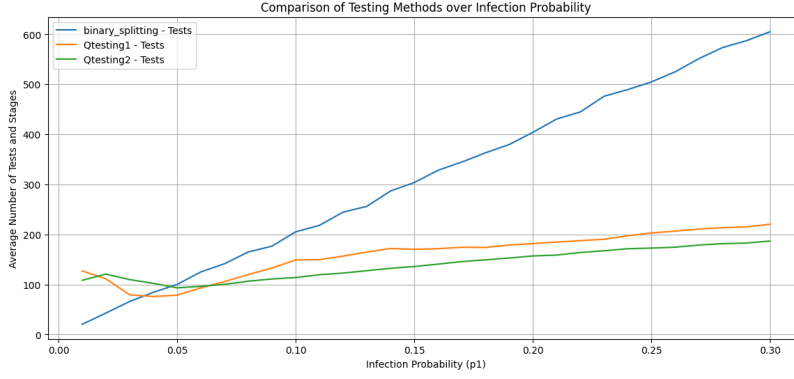
By using this method, we can identify the infected person with just $\log_2(N)$ tests.

Finally, we also employ some basic techniques: If the number of infected is equal to the size of the group, we know the whole group is infected, and if the number of infected is zero, then the whole group is safe.

T_2 test: Output one of five values based on the number of infected individuals.

The strategy delineated above works only for 16 people and less when given test 2 ranges. So simply run binary if the group size is larger than 16. Once we reach size 16, we can use the pigeonhole strategy from above. The rats and poison method described above works here as well.

I.I.D Results.



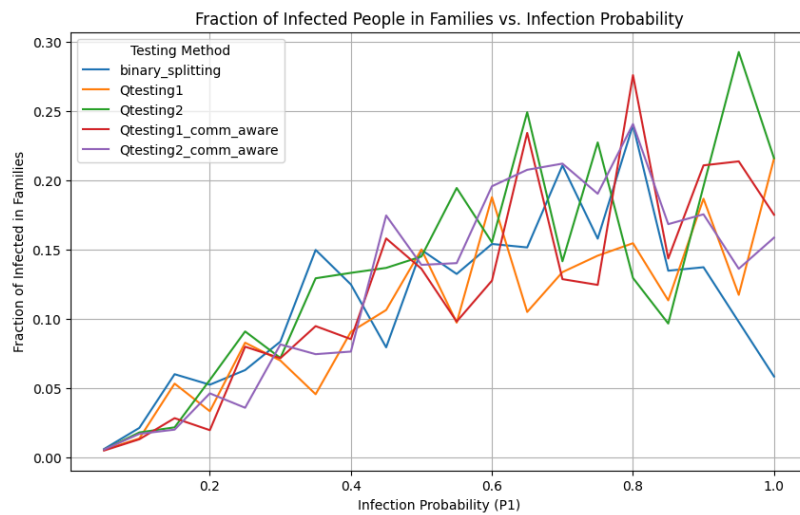
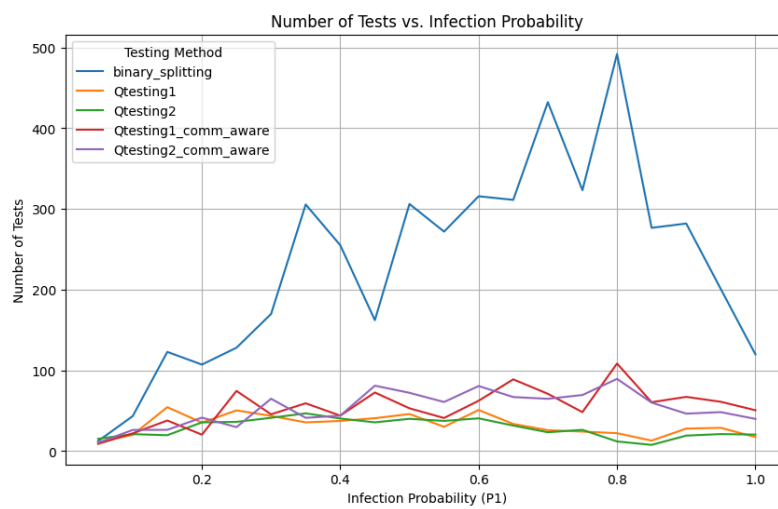
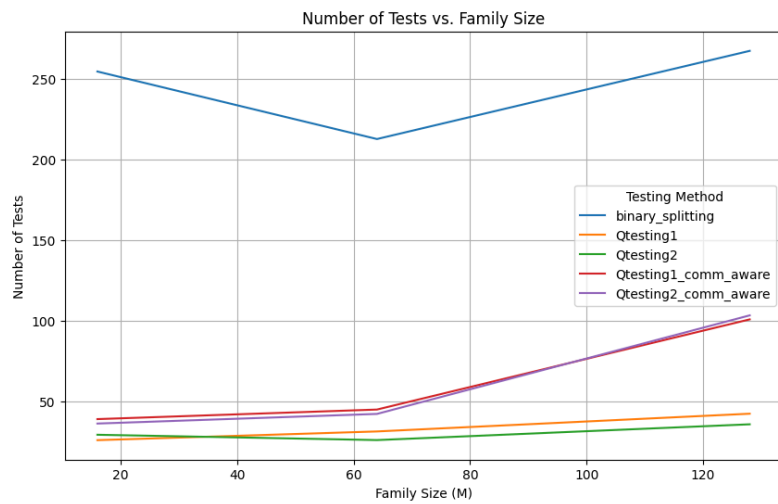
The strategies we employed clearly allowed for significant improvement over the binary testing code with major improvements over higher infection rates. The QTesting1 outperforms QTesting2 which is expected as Test 1 carries more information.

2. Design Adaptive Algorithms for Populations with Community Structure

For community structure, we first run binary on the entire communities to find which communities have an infection as some communities may get an infection while others don't. After identifying the infected communities, we simply run the testing strategies described above on them.

Table 3: Parameters for SBM datasets.

N	M	$q0, q1$	$p0$	$p1$	time steps
256	16	(0.9, 0.1)	0.001	[0.05:0.1:1]	2
	64	(1, 0)	0.001	[0.05:0.1:1]	2
	64	(0.9, 0.1)	0.001	[0.05:0.1:1]	2
	64	(0.5, 0.2)	0.001	[0.05:0.1:1]	2
	128	(0.9, 0.1)	0.001	[0.05:0.1:1]	2



Library Function: math, numpy, seaborn

