TEAM - Transmission of Epidemic Among Membranes

S. Erba[1], G. Franco[2], F. Reiff[2], J. M. Sempere[3], and C. Zandron[1]

[1] Dept of Computer Systems and Communication, University of Milano Bicocca, Milan, Italy
[2] Dept of Computer Science, University of Verona, Verona, Italy
[3] VRAIN, Universitat Politècnica de València and VALGRAI, Valencia, Spain

Abstract. The experience with COVID-19 pandemic has highlighted the importance of computational models to understand, simulate, and control infectious disease dynamics, also to significantly support decision-making processes for public health. Within this context, membrane computing has shown to be promising for modeling complex epidemiological systems, due to its population based inherent parallelism and compartmental structure.
Two models stand out for their complementary strengths among the existing works that adopt this paradigm. One, known as LOIMOS, focuses on detailed representations of infection and symptom progression, offering a biologically rich modeling of disease stages. The other, referred to as MVT, introduces behavioral dynamics, allowing individuals to adapt their actions based on perceived risk, personal willingness to vaccinate, and inter-provincial mobility preferences.
This work combines the core ideas of LOIMOS and MVT into a unified simulation framework, referred to as TEAM (Transmission of Epidemic Among Membranes), which integrates biological accuracy with behavioral flexibility. The goal is to create a general-purpose, computationally efficient model adaptable to various infectious diseases beyond COVID-19. Central challenges included resolving formal and structural differences between the two source models and harmonizing their rule-based dynamics.

Keywords: Behavioural epidemiology, distributed computing, infection diffusion, massive parallelism, population dynamics.

1 Introduction

Membrane computing is a branch of natural computing inspired by the structure and function of biological cells [29, 11, 10], through the metaphor of nested membranes and chemical reactions occurring within compartments [15].

P systems (introduced by Gheorghe Păun [21, 19]) offer an inherently parallel and non-deterministic framework [20] that has attracted interest from both fields

of theoretical computer science and systems biology [9]. Each membrane defines a region containing multisets of objects (representing chemical substances) and is subject to evolution rules that govern transformations, communications, and structural operations.

Various classes of P systems have been advanced in the literature, while P systems with active membranes are particularly expressive [18, 31]. They extend the basic framework by introducing additional features such as membrane polarization, membrane division, and dissolution, enabling dynamic structural changes and enhanced computational power [33]. Division rules are particularly powerful, by allowing membranes to duplicate together with their contents. This feature enables exponential workspace growth, a critical asset in solving computationally hard problems such as the HPP and SAT [32, 17].

Recent advancements in membrane computing have introduced significant refinements to the semantics of evolution rules, enabling more expressive and biologically realistic models. To provide finer control over the application of rules, three key mechanisms have been proposed:

- Input Population Percentage, which specifies the proportion of objects or membranes to which a rule is applied, enabling partial or population-level interactions;
- Rule Priority Index, which resolves conflicts between applicable rules based on their relative precedence, allowing deterministic selection;
- Rule Probability, which assigns a likelihood to rule execution when given conditions are met, by introducing controlled non-determinism and reflecting the probabilistic nature of many biological processes.

While expanding the model expressive power, these mechanisms remain consistent with the foundational principles of membrane computing: parallel execution, local interactions, and rule-based rewriting.

Numerous works have employed P systems to simulate the spread of infectuous diseases, and evaluate containment strategies under various conditions. Building upon these foundations, we here introduce the TEAM epidemiological model, which extensively incorporates the three features above. We propose this model with the main aim to integrate the strengths and to tune or eliminate the limits of two prior recent models of the literature, LOIMOS [1] and MVT [28], into a unified framework, that enhances flexibility and expressiveness in simulating infection diffusion processes through membrane computing.

LOIMOS is a Greek word meaning pestilence, which more figuratively recalls pestilent fellows. It is the name for the epidemiological model developed at the Universitad Politècnica de València, which introduces a rich biological layer and a predictive statistical structure. It models epidemiological states in detail, assigning individuals to a broader set of location types while tracking a finer granularity of attributes, such as age group, viral load, symptom severity, and immune status. Infection in LOIMOS involves thousands of rules and complex object dynamics, where individuals generate and interact with virus, antivirus, specialized immune elements, and further components. Disease progression is

influenced by both viral load and immune efficiency, producing various symptomatic trajectories, ranging from home isolation to hospitalization or ICU admission, each with associated recovery or mortality probabilities. LOIMOS also models four distinct infection types based on combinations of innate or acquired immunity and symptom intensity, reflecting more nuanced biological variability. LOIMOS enjoys realistic features and nice system properties, and achieves notable simulation results reported in [3, 2].

MVT is a rule-based simulator of disease spread across provinces implemented in Python. The name MVT reflects the collaboration work among the universities of Milano-Bicocca, Verona, and Trieste. In MVT, the population is distributed among a limited set of location types—schools, workplaces, hospitals, and common areas—across multiple provinces. Each individual carries structured attributes including age, mobility data, and epidemiological status, while infection evolves according to simplified disease phases: incubation, active infection, and immunity. Behavioral adaptation [27] is a main feature of this approach, indeed it plays a central role, with individuals responding to local infection prevalence by altering movement patterns and risk behaviors, a mechanism captured by newly introduced Caution Parameter. This model basically integrates disease dynamics with geographic mobility, enabling exploration of how different vaccination strategies, different social restriction rules, and different behavioral responses, affect outcomes in the population. Successful simulations of this model among provinces of the Lombardy region in Italy are presented in [8, 6].

MVT and LOIMOS provide complementary insights: the former focuses on adaptive behavior and geographic mobility, while the latter models immune response and symptomatology in greater biological detail. These models serve as conceptual and technical foundation for TEAM, the unified framework presented in this paper. The name is inspired by both the concept of team membranes, recently circulating in the membrane computing community, and the active collaboration between University of Milano Bicocca, Verona, and Univèrsidad de València.

In next sections we present the source model and its simulator. Section 2 namely introduces the proposed model, detailing its rules for infection, contagion, symptoms, movement, recovery, and behavior, whereas Section 3 focuses on the simulation and validation environment, where scalability, infection generalization, integration of real-world data, and assessment of behavioral parameters and quarantine strategies are reported as the results of this investigation. Section 5 concludes the paper and outlines potential future developments.

2 Model definition

TEAM model is defined as a cell-like P system with active membranes without polarization, formally described by the tuple:

$$\Pi = (V, H, \mu, w_1, w_2, \ldots, w_m, R) \tag{1}$$

where:

1. V is the alphabet of objects. Each membrane contains objects, represented with multisets to map strings of symbols onto an alphabet;
2. H is the alphabet of labels for membranes. They have a label to distinguish them from different membranes of the same type;
3. μ is the initial membrane structure, of degree m, with all membranes labeled with elements of H. A membrane with label h is represented as $[]_h$;
4. w_i are strings over V specifying the multiset of objects initially in the i-th regions defined by μ.
5. R is a finite set of evolution rules.

The membrane structure is tree-like: at the top is the skin membrane, which contains all provinces. Each province includes a variety of places such as houses, hospitals, schools, and more, each with a limited capacity. Individuals, modeled as membranes themselves, move between these places according to rules that reflect behavior and epidemiological state.

Each place serves a specific function: houses host families; hospitals and ICUs treat severe infections; schools, workplaces, and leisure centers capture daily routines; common areas model transit and crowded indoor environments. The leisure center represents both open and closed public spaces, varying in infection risk depending on time and use.

Embedded within places, individuals never leave the system but move through its membrane structure via movement rules that simulate interactions in a population. Every membrane carries objects that define its characteristics.

Objects within membranes represent information, control behavior, and enable evolution through rewriting rules. Each individual contains the temporal objects Hour_i and Day_l to govern daily routines.

Age is represented by tags like *young*, *adult*, or *elderly*, influencing movement patterns. Viral dynamics are modeled using objects such as v_1 for viral load, *antiv* and *antivesp* for immune response, and E_x for symptoms, where $x \in \{1, 2, 3, 4\}$ indicates the level. Higher v_1 values increase infection severity, and a successful immune response eliminates v_1, triggering recovery and temporary immunity.

Each place tracks local infection levels via an object ϕ indicating the number of infected individuals present, affecting contagion probabilities. Additional objects encode individual attributes such as ID, location, hospitalization time, and vaccination status (denoted by the suffix V). These objects are manipulated through evolution rules that drive the simulation.

Rules are described in the following, according to the process they control in the whole epidemiological dynamics. Each evolution rule $\alpha \xrightarrow{p,q} \beta$ has a parametrized application guided by a couple of parameters p and q. The first parameter $p \in [0, 1]$ is a probability value that models the stochastic simulation of the systems, while the second parameter $q \in \mathbb{N}$ represents the value of a priority relation defined over the rules for the application order. We do not explicit this parameter when we have a system with only one rule. This allows P systems to be effectively used in probabilistic modeling of biological networks, as shown in [7].

Infection rules

Infection depends on location, number of infected individuals, individual status, and contextual factors. The general form is:

$$ind_{j,H}\, ind_{k,I}\, \phi \xrightarrow{\text{Infection Probability}} ind_{j,Inc}\, ind_{k,I}\, \phi \tag{2}$$

- Infection Probability: base infection rate $\cdot \frac{\phi}{\text{total individuals}} \cdot \psi(M)$ where ψ is a decreasing function that models awareness of contagiousness [28].
- Base infection rate: specific to the type of membrane.
- ϕ: Current infection count in the location.
- Infection status: H for Healthy, I for Infected, and Inc for Incubation.

When the infection rule is applied and a healthy individual becomes infected, they are assigned five v_1 objects to represent the initial viral load.

Viral load rules

The following rules are some of the many defined in TEAM, selected for their explanatory value in illustrating the management of viral load within an individual. They are listed in order of decreasing priority:

- $antiv^{200}, v_1^{200} \xrightarrow{0.012,4} antivesp, v_1\_ino, antiv^{199}, v_1^{199}, sint^{200}$
  The individual's antibodies (*antiv*) fight the disease (v_1), and a new specialized antibody (*antivesp*) is created. The probabilities for applying this rule depend on the individual's state of health.
- $antivesp, v_1 \xrightarrow{1,3} antivesp, v_1\_ino, sint$
  Each *antivesp* fights the infection with a certain probability, generating harmless viruses, while non-specialized antibodies have a lower probability.
- $antiv, v_1 \xrightarrow{0.001,3} antiv, v_1\_ino, sint$
  Each non-specialized antibody (*antiv*) fights against the infection with a lower probability.
- $v_1 \xrightarrow{0.035,2} v_1, v_1, sint$
  Viruses not occupied by the other rules are free to grow.
- $v_1 \xrightarrow{1,1} v_1, sint$
  Symptoms (*sint*) increase according to the viral load present, even if the v_1 elements follow no specific rule. In this way, each of the evolution rules generates a certain amount of *sint*, useful for managing the symptoms, depending on how many v_1 are involved in the left-hand side of the rule.

Symptoms rules

Each individual has a symptom status object E_x, with $x \in \{1, 2, 3, 4\}$:

- E_1: the host has no symptoms, and may be or may not be infected
- E_2: the host has mild symptoms, may be or may not realize he is infected

- E_3: the host has severe symptoms and needs hospitalization
- E_4: the host has critical symptoms and needs ICU

The following rules manage system variables related to symptoms, which are closely linked to changes in viral load.

- $sint^{700}, flag \xrightarrow{1,2} cont, flag$
  When there are 700 *sint* objects, this rule takes the *flag* object and generates the high viral load state represented by *cont*.
- $E_i, cont \xrightarrow{\text{Symptoms Probability},3} E_{i+1}$
  Transition to a more severe symptom stage (E_{i+1}) due to sustained high viral load. The probabilities for the growth of symptoms reflect real-world worsening rates and are respectively 1, 0.0015, and 0.001 for transitions in the states E_2, E_3, and E_4. These values were initially derived from the LOIMOS model [1], and underwent a balancing phase after the integration and unification with MVT.
- $E_i, cont \xrightarrow{1,2} E_i, cont$
  If the *cont* object is present and the host does not worsen, the current state is maintained.
- $E_i \xrightarrow{1,1} E_1$
  If there is no *cont* object, the viral load is not dangerous, so the host is cured and the symptoms disappear.
- $sint \xrightarrow{1,1} \lambda$
  The object *sint* is used to know how many objects v_1 there were in a previous step. If they are not used, then are deleted.

Infected individuals with medium or high-level symptoms (E_3 or E_4) can, every hour, trigger a dissolving rule that results in death.

- Probability of death for seriously ill patient (E_3): $1.\overline{6} \cdot 10^{-5}$/h.
- Probability of death for critical patient (E_4) in ICU: 10^{-4}/h.
- Probability of death for critical patient (E_4) outside the ICU: $2 \cdot 10^{-4}$/h.

Movement rules

Movements simulate daily routines or inter-provincial transfers. All movement is handled by membrane mobility rules.

$$[[ind_{PV_x,PV_y}\,\text{Hour}_i\,\text{Day}_l]_{PM}]_{PV_x} \xrightarrow{\text{Movement Probability}} [[ind_{PV_x,PV_y}\,\text{Hour}_{i+1}\,\text{Day}_l]_{CA}]_{PV_y} \quad (3)$$

- ind_{PV_x,PV_y}: an individual traveling from province x to province y.
- Hour_i and Day_l: respectively the i-th hour of the day and the l-th day of the week.
- PM and CA: respectively, a generic Place Membrane, and the Common Area of the destination province

– Movement Probability: this value depends on the epidemiological situation in the destination province. In any case, it will be 0 if the movement between regions is not in the routine set up for that individual at time i on day l.

Other movement rules handle transitions within the same province, such as commuting between places (e.g., house and workplace), but follow the same idea as the one shown. These movement rules bring the model to life, allowing individuals to follow their own routines.

Recovering rules

Having described how an individual becomes infected, how the disease evolves, and how symptoms develop, we now present how recovery occurs. In all cases, a recovered individual gains immunity for 180 days. There are three ways to recover:

– Recovery through hospitalization: infected individuals with severe symptoms (E_3 or E_4) have a probability of 0.03 (for hospital) or 0.05 (for ICU) per hour of being admitted to a suitable facility, if available in the region. If already hospitalized and symptoms worsen to critical (E_4), the individual is moved directly to ICU. ICU availability is limited, so simulations must be correctly parameterized. Once admitted, a 7-day recovery cycle begins. The same cycle applies to ICU transfers, as hospitalization days are shared across facilities.
– Recovery through specialized antivirus: if the individual accumulates enough *antivesp* objects, they are considered cured. The production rate and healing threshold of *antivesp* are critical parameters that have undergone a tuning phase.
– Recovery by zeroing the viral load: even without hospitalization or a full antibody response, if *antivesp* and *antiv* objects reduce the viral load to zero, the individual recovers.

Behavior

To enhance realism, TEAM includes individual behaviors that influence epidemic dynamics, inspired by MVT. These behaviors add heterogeneity and decision-making capacity to individuals, preventing uniform reactions across the population and improving the realism of the simulation [27]. Four main behavioral features are implemented:

– Prudence Parameter: models individual awareness of symptoms. People may choose to stay home when mildly symptomatic (E_2), depending on their prudence level. A higher Prudence Parameter reduces the probability of going out while infected, simulating varying civic responsibility and risk perception. A detail of this parameter is described in the related section.

- Likelihood to Change Province: individuals avoid traveling to provinces with high infection rates. The probability of inter-provincial movement decreases with the proportion of infected people in the destination area, discouraging mobility toward high-risk zones. The number of individuals not predisposed to change provinces in their routine is a fixed parameter called Same Province Percentage, which can be set during simulation creation.
- Caution Factor: as infections rise, people become more cautious. This is modeled through a decreasing function that reduces contagion probability as the infection rate increases.
- Vaccination Will: the willingness to get vaccinated increases with perceived risk, based on infection prevalence. Individuals are vaccinated based on a probability modulated by a behavioral function. Each vaccinated person receives a randomly assigned vaccine efficacy and a corresponding protection duration.

3 Scenario and simulations

TEAM was developed using an Object-Oriented Programming (OOP) approach, enabling a clear and efficient translation of the membrane computing paradigm into code. This choice was favored over the use of existing frameworks tailored to membrane computing—such as LOIMOS or P-Lingua [22]—due to the superior flexibility, performance, and compatibility of OOP with the starting MVT model.

To highlight the performance improvement achieved by adopting OOP, simulation times were compared: LOIMOS requires more than 200 seconds to simulate a single day, whereas TEAM, under the same population size and hardware conditions, completes the same task in less than 10 seconds.

In TEAM simulator, individuals are modeled as objects with attributes representing their health status. The rewriting rules of membrane computing are implemented as methods, and a strict constraint is applied: no attribute can change more than once per simulation step, in accordance with membrane computing semantics. Rule application respects both priority and probability, and membrane hierarchy ensures organizational consistency (e.g., provinces contain places, which contain individuals).

Python was used as the programming language due to its simplicity, data handling capabilities, and support for OOP. The simulation progresses in discrete time steps, each representing one hour. At each step, every object can apply one rule, and a new configuration is generated. The system halts when a predefined number of steps (entered as input, in days) is reached.

To broaden the simulator usability, a key development was the creation of a Graphical User Interface (GUI) using Python Tkinter library. The primary goal of the GUI is to make the sophisticated model accessible to researchers and officials who are not programming experts, such as those in epidemiology or public health. The interface organizes the numerous simulation parameters—from

population and behavioral factors to quarantine rules—into a clear, tabbed layout. To further aid usability, it incorporates informative tooltips that explain the function of each parameter. Upon completing a run, the GUI presents the results, including key data and plots, directly within an integrated output panel, facilitating immediate analysis. This feature removes the barrier of direct code manipulation and establishes the simulator as a practical tool for rapid and flexible scenario testing. The complete source code for the TEAM simulator is publicly available [24].

Scalability of the scenario

A fundamental aspect of this work is the generalization of the model in terms of both structure and application. The initial simulators were tailored specifically to COVID-19 and did not have an easily editable scenario. For example, LOIMOS required pre-defined input files for membranes, making structural changes cumbersome, while MVT used a fixed number of regions, limiting flexibility.

TEAM introduces a dynamic system that allocates place membranes based on realistic capacity assumptions. These capacities are now decoupled from hard-coded values and instead reflect typical structural limits observed in real-world contexts. This change ensures a more coherent relationship between population size and the number of available facilities.

Particular attention was given to the modeling of healthcare structures, where capacity constraints are critical for simulating scenarios such as hospital saturation. The new approach enables the simulator to recreate more realistic and flexible settings, allowing for the study of how healthcare overload negatively influences the progression of the disease.

Scalability was also enhanced by introducing parameters to control the number of provinces and the mobility behavior of individuals. A new parameter, called Same Province Percentage SPP, allows control over the proportion of individuals whose destination province matches their origin. For instance, setting SPP to 80% models low inter-province mobility, useful for rural or disconnected regions, while a lower SPP simulates high urban mobility, such as between neighborhoods in a city. This parameter, combined with the ability to vary the number of individuals and provinces, results in a highly adaptable simulation environment suitable for diverse scenarios.

Generalization of infections

Another major generalization concerns the type of infection being modeled. While this work focuses on COVID-19, the simulator is designed to be adaptable to other infectious diseases. The two base models used diverged significantly in infection dynamics: LOIMOS employed a viral load model with high granularity, while MVT used a simplified fixed-day cycle (Incubation $\rightarrow$ Infected $\rightarrow$ Recovered).

To accommodate various infection types, TEAM includes support for both approaches. This flexibility is crucial when modeling novel diseases where detailed clinical parameters may not yet be available.

Additional modularity is provided by enabling or disabling components such as ICUs. While critical for COVID-19 simulations, ICUs may be irrelevant for diseases with low hospitalization rates. The simulator can also incorporate dynamic quarantine policies. By setting parameters for start time and duration, scenarios involving time-specific lockdowns can be tested to determine optimal containment strategies.

Another important element is the Prudence Parameter (PP), which allows the simulation to reflect varying levels of public awareness and responsiveness. For a well-known disease with visible symptoms and public warnings, a high PP simulates a population acting prudently. Conversely, for a new disease with ambiguous or mild symptoms, a low PP represents delayed recognition and high unintentional transmission, capturing important epidemiological dynamics. In such scenarios, individuals may become unwitting vectors, mistaking mild symptoms for unrelated or negligible conditions, and continue interacting socially, thereby facilitating the spread of the disease [16]. These features collectively make TEAM highly generalizable and applicable to a wide range of infectious disease models beyond COVID-19.

Database

Model calibration and result validation constitute a key part of this work. While many simulation parameters were derived from existing models, their integration into a unified framework initially produced results with unrealistically high numbers of infections and deaths. This discrepancy became evident when comparing simulation outputs with empirical data from the Lombardy region, which had not been previously used as a validation benchmark.

In particular, LOIMOS produces a combined infection peak of roughly 46% (summing across the four infection types), which is far higher than observed in reality. By contrast, MVT yields prevalence peaks between 8% and 20%, depending on the value of the caution factor, and therefore provides more realistic outcomes. The integration of the two approaches required careful parameter adjustment to maintain internal consistency.

Parameter tuning focused on epidemiological factors such as transmission probability, viral progression, and mortality rates conditional on individual health and symptom profiles. These parameters were refined to align model outputs with real-world observations. To ground the validation process, we used the publicly available dataset for the Lombardy region [13], covering the period from February 2020 onward. Lombardy, one of the Italian regions most severely affected during the early phase of the COVID-19 pandemic [25], offers detailed and high-quality data, which have been used extensively in previous studies [5, 26, 14]. The model was calibrated primarily against two critical indicators: the number of infected individuals and the number of deaths.

The primary calibration of our model was performed against data from the Lombardy region. However, to ensure the model principles were not narrowly tailored to a single dataset, we conducted a secondary validation using the distinct demographic and epidemiological landscape of Veneto. Using data from Italy's Civil Protection Department, we re-scaled the simulation for Veneto's smaller population. This meant modeling a population of 12,000 individuals across the region's 7 provinces, maintaining the proportional representation used in the Lombardy setup. This test confirmed the model's robustness. In a 365-day run, the simulation once again successfully captured the primary infection peak and mirrored the overall trend of cumulative deaths, providing strong evidence that our framework is adaptable and can generalize to different regional contexts.

Pseudocode. The entire simulator code can be viewed in [24], and Algorithm 2 describes the pseudocode, structured into two main functions:

- `create_scenario()`: Initializes the environment by generating provinces, places, and individuals with the correct demographic distribution. Initial infections are also introduced.
- `run_simulation(days, hours per day)`: Governs the epidemic progression, handling mobility, infections, health status updates, hospitalizations, and data recording.

4 Validation of the model

Here we evaluate the effect of the newly introduced Prudence Parameter (PP), with $0 \leq PP \leq 1$, which controls the probability that symptomatic individuals choose to self-isolate rather than follow their daily routines. The parameter modulates individual behavior in a simple yet effective way: when individuals experience mild symptoms (E_2), the probability of leaving the house is reduced by a factor of $(1-PP)^2$. Thus, a higher PP reflects a more cautious population. For instance, if $PP = 0.3$, individuals are about 50% less likely to go out while symptomatic compared to their asymptomatic behavior.

At the extremes, $PP = 0$ corresponds to a population entirely unaware of or indifferent to the disease, resulting in no behavioral change. In contrast, $PP = 1$ models complete prudence, where symptomatic individuals never leave the house, mimicking the original LOIMOS behavior in which infected individuals are fully isolated upon symptom onset.

This parameter can also be interpreted from a sociological perspective. It offers a way to simulate varying levels of civic awareness and compliance with public health guidelines. Even in contexts where information about the disease is widely available, populations may exhibit limited adherence to testing and isolation directives, especially if institutional trust or civic engagement is low [12]. Such behavior has been documented in the Italian situation, as described in [4, 30]. As a result, simulations may justifiably adopt medium or low values of PP, even under well-informed conditions.

Algorithm 2 Pseudocode of the TEAM simulator

Input: Simulation parameters, days, population
Output: Epidemic evolution data, seconds

```
function CREATE_SCENARIO
    for all province in provinces do
        Create province membrane instance
        Calculate the correct number of place membranes
        Generate place membranes
        Create individuals with the correct age distribution
        Assign individuals to houses
        Introduce initial infections
    end for
end function

function RUN_SIMULATION(days, hours per day)
    Create CSV file
    for all day in days do
        for all hour in hours per day do
            if not a quarantine day and is the correct hour then
                Move individuals between provinces through common areas
                Move workers to workplaces
                Move students to schools
                Move elderly to leisure centers
                Simulate infections in all places with individuals
                Move individuals back home
            end if
            Trigger infection and vaccination progress
            Check for deaths
            Discharge recovered individuals from hospitals
            Check for hospitalization based on hospital capacity
            Track infections and update the scenario
        end for
        Reduce recovery days in hospitalized individuals
    end for
    Write data to CSV file and create charts
end function
```

The impact of PP is illustrated in Fig. 1, which shows the prevalence and deaths under varying values. Simulations with $PP = 0$ were excluded, as they represent implausible scenarios in which no individual, regardless of age or symptom severity, modifies their behavior when symptomatic.

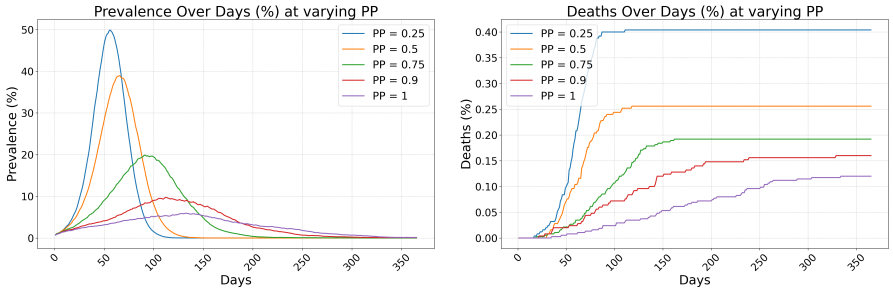

Fig. 1. Validation against PP. Prevalence (on the left side) and deaths (on the right side) values variation in time, according to different values of the parameter PP.

Results demonstrate that PP has a pronounced influence on epidemic dynamics. Lower values lead to unrealistically high peaks in both prevalence and mortality, deviating significantly from real-world data such as that observed in Lombardy and Veneto. In particular, PP values below 0.75 yield outcomes that are inconsistent with known epidemiological patterns, suggesting that models using such parameters may not reflect realistic human behavior during an outbreak.

For this reason, the default value used throughout the rest of the study is $PP = 0.9$, consistent with assumptions made in LOIMOS, which implicitly modeled behavior close to $PP = 1$. It is worth noting that symptom onset does not occur immediately after infection; thus, even high PP values do not eliminate the risk of transmission from individuals unaware of their infectious status.

Note that this current formulation of the parameter PP should be regarded as a high-level simplification. In practice, prudence is not a homogeneous trait, but rather is shaped by a variety of factors, including socioeconomic background, literacy rates, and cultural contexts. PP therefore acts as an umbrella variable that condenses many heterogeneous influences into a single dimension.

Nonetheless, the experiments underscore the importance of behavioral response in epidemic containment. They also suggest that promoting voluntary self-isolation in response to symptoms can significantly mitigate disease spread and reduce mortality.

At this point, we present a comparative analysis of three simulations in which the only varying factor is the quarantine scheduling, while all other parameters, including $PP = 0.9$, remain unchanged. The goal is to evaluate how different quarantine patterns affect the disease progression.

Quarantines are among the most effective measures available to contain epidemics. While they do not offer a definitive solution, they can substantially reduce transmission, ease hospital burden, and buy time for medical response. However, prolonged lockdowns are costly from both economic and psychological standpoints. For this reason, it is crucial to apply them as efficiently as possible, maximizing impact with minimal duration.

Fig. 2 shows three simulations. The first includes no quarantine and serves as the baseline. In the second simulation, a 60-day quarantine starts on day 50, aligned with the rising edge of the infection peak. This leads to a significant drop in active cases, which climb again after restrictions are lifted but remain well below the values observed without intervention. This confirms that a well-timed quarantine can mitigate the epidemic's peak and delay its resurgence.

The third simulation introduces a split quarantine strategy: three 20-day lockdowns alternated with 20-day reopening intervals. Although the total quarantine duration remains 60 days, the effect differs markedly. After the first lockdown, the decline in infections slows, but the second period causes a further sharp drop. By the end of the third, the number of infected falls below the initial level. This configuration delays the return to pre-quarantine levels by over a year, suggesting extended protection despite the same social cost.

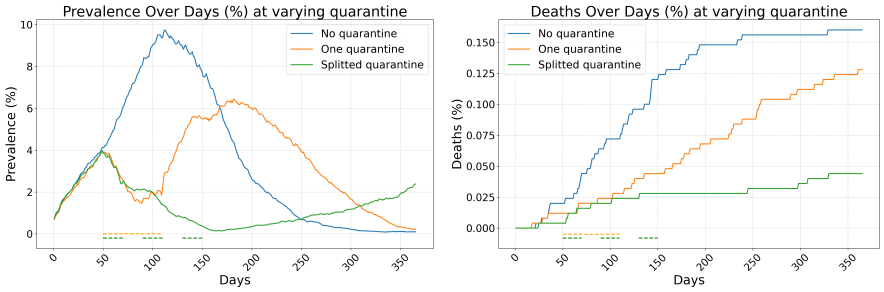

Fig. 2. Validation against targeted quarantines. Prevalence (on the left side) and deaths (on the right side) values variation in time, according to different quarantine periods (dotted lines).

These results support the strategic use of intermittent lockdowns as a cost-effective tool to manage disease progression and reduce fatalities, particularly when long-term eradication is not immediately feasible.

5 Conclusions

This work investigated the application of P systems to the simulation of infectious disease dynamics. The main objective achieved was the combination of two key aspects, behavioral management and infection diffusion complexity, into a single coherent framework.

A major strength of P systems lies in their natural support for parallel computation, which facilitates efficient large-scale simulations while preserving modularity and adaptability. In this work, several features were implemented to generalize the model to various epidemiological scenarios. Among these are the dynamic configuration of provinces and place membranes, the Same Province Percentage (SPP) parameter, controlling mobility patterns, and parameters such as the Prudence Parameter (PP) for behavioral responses, and optional inclusion of ICUs, and viral-load-based infection dynamics.

These features allow TEAM to reproduce complex disease dynamics, including the effects of individual awareness, government interventions such as quarantines, and structural healthcare components. A graphical interface was also developed to ensure usability and support further experimentation.

Simulations were conducted under a range of settings: 25,000 individuals, 12 provinces, 365-day duration, demonstrating the model's robustness across scenarios. Behavioral elements and quarantine enforcement were tested, and results were confirmed about i) the relevance of parameters like PP to shape the observed dynamics, and ii) the cruciality of confinement intervention to significantly impact infection spread. The above validates the model's potential as a decision-support tool. Regional data, particularly from Lombardy and Veneto,

were employed in two different tranches of validation [8, 23], confirming the system's ability to reflect realistic epidemic curves.

To further enhance the model's realism and adaptability, several research directions are envisaged. Parameters related to prudence, infection dynamics, daily schedules, and mobility across place membranes were calibrated specifically for some regions of the Italian context, as mentioned. It has been our starting point, given the availability of detailed datasets. A natural next step will be to test the simulator on other countries and contexts, to increase its generalization and confirm that the included variables can capture diverse population behaviors.

Another important direction concerns the integration of demographic factors such as births and non-disease-related deaths, which would allow for more realistic long-term simulations, accounting for population turnover. Improvements of computational performance could also be achieved by parallelizing the simulation, enabling faster execution, especially in large-scale or long-duration scenarios.

Regarding the Prudence Parameter, while its current formulation as a high-level simplification is effective for modeling purposes, a deeper behavioral and sociological refinement could substantially increase the realism of the simulations without additional computational complexity. A promising direction for future work is the development of a lightweight database that collects country-specific indicators and suggests plausible values—or ranges—for PP and similar behavioral parameters. Such an extension would enhance the model ability to represent diverse populations and improve its applicability across different epidemiological and cultural contexts.

Incorporating seasonal variations, such as changes in behavior during holidays or colder months, would improve the accuracy of infection and mobility patterns. Further differentiation of household structures, for instance by modeling single-person homes or student residences, could refine transmission dynamics within domestic environments.

The expansion of place membranes, including specific environments like universities or transport hubs, would support more granular simulations. Additionally, simulating travel restrictions between provinces may offer insights into the effectiveness of regional containment strategies.

Finally, applying optimization techniques or machine learning could support parameter tuning and predictive accuracy, while testing the model against data from different infectious diseases would assess its generality, flexibility, and robustness.

In conclusion, TEAM combines computational efficiency, epidemiological depth, and behavioral flexibility, providing a strong foundation for further research in disease modeling and public health management strategy.

Acknowledgement and funding. We all wish to thank Marcelino Campos from Universitat Politècnica de València, who is the developer of LOIMOS, for fruitful discussions with him about the software details implementing modeling approaches of LOIMOS.

Sandro Erba is funded by the Department of Computer Science, University of Verona, within the project *Application of Spiking Neural P systems to the federated learning of biomedical data* FUR_FRANCO - UA.VR.050.DPINF.DINF-RATE.

José M. Sempere is funded by the Generalitat Valenciana within the Prometeo program in the project *A disruptive way to ameliorate the diagnosis and treatment of sensorineural diseases.* CIPROM/2023/026.

References

1. F. Baquero, M. Campos, C. Llorens, and J. M Sempere. P systems in the time of covid-19. *Journal of Membrane Computing*, 3:246–257, 2021.
2. M. Campos, J. M. Sempere, J. C. Galán, A. Moya, R. Cantón, C. Llorens, and F. Baquero. Simulating the efficacy of vaccines on the epidemiological dynamics of sars-cov-2 in a membrane computing model. *Microlife*, 3:uqac018, 2022.
3. M. Campos, J. M Sempere, J. C. Galán, A. Moya, C. Llorens, C. de Los-Angeles, F. Baquero-Artigao, R. Cantón, and F. Baquero. Simulating the impact of non-pharmaceutical interventions limiting transmission in covid-19 epidemics using a membrane computing model. *MicroLife*, 2:uqab011, 2021.
4. L. Carlucci, I. D'Ambrosio, and M. Balsamo. Demographic and attitudinal factors of adherence to quarantine guidelines during covid-19: The italian model. *Frontiers in Psychology*, 11, 2020.
5. D Cereda, M Tirani, F Rovida, V Demicheli, M Ajelli, P Poletti, F Trentini, G Guzzetta, V Marziano, A Barone, M Magoni, S Deandrea, G Diurno, M Lombardo, M Faccini, A Pan, R Bruno, E Pariani, G Grasselli, A Piatti, M Gramegna, F Baldanti, A Melegaro, and S Merler. The early phase of the covid-19 outbreak in Lombardy, Italy, 2020.
6. A. D'Onofrio, M. M. Fareed, G. Franco, D. Valcamonica, and C. Zandron. Integrating human behavior and membrane computing in epidemiological models. Submitted, 2025.
7. A. A. Elsayed, R. Ceprián, A. Hafez, C. Llorens, and J. M. Sempere. Enhancing p systems for complex biological simulations. In *Proceedings of the 24th Conference on Membrane Computing (CMC 2024)*, Nice, France, 2024.
8. S. Erba. Covid-19 infection inspired diffusion dynamics: Generalization of a membrane-based simulator. Master's thesis, Department of Computer Systems and Communication, University of Milano-Bicocca, 2025.
9. G. Franco, N. Jonoska, B. Osborn, and A. Plaas. Knee joint injury and repair modeled by membrane systems. *BioSystems*, 91(3):473–88, 2008.
10. R. Freund, I. Pérez-Hurtado, A. Riscos-Núñez, and S. Verlan. A formalization of membrane systems with dynamically evolving structures. *International Journal of Computer Mathematics*, 90(4):801–815, 2013.
11. R. Freund and S. Verlan. A formal framework for static (tissue) p systems. In *International Workshop on Membrane Computing*, pages 271–284. Springer, 2007.
12. S. Hills and Y. Eraso. Factors associated with non-adherence to social distancing rules during the covid-19 pandemic: a logistic regression analysis. *BMC Public Health*, 2021.
13. https://github.com/pcm-dpc/COVID-19/tree/master/dati-regioni.

14. A. Lombardi, N. Amoroso, A. Monaco, S. Tangaro, and R. Bellotti. Complex network modelling of origin–destination commuting flows for the covid-19 epidemic spread analysis in italian Lombardy region. *Applied Sciences*, 11(10), 2021.
15. V. Manca, A. Castellini, G. Franco, L. Marchetti, and R. Pagliarini. Metabolic P systems: A discrete model for biological dynamics. *Chinese Journal of Electronics*, 22(4):717–723, 2013.
16. F. Mowbray, L. Woodland, L. E. Smith, R. Amlôt, and G. J. Rubin. Is my cough a cold or covid? a qualitative study of covid-19 symptom recognition and attitudes toward testing in the UK. *Frontiers in public health*, 9:716421, 2021.
17. L. Pan and A. Alhazov. Solving HPP and SAT by P systems with active membranes and separation rules. *Acta Informatica*, 43:131–145, 2006.
18. A. Paun. On P systems with active membranes. In *UMC*, pages 187–201. Springer, 2000.
19. Gh. Păun. *Membrane computing: an introduction*. Springer Science & Business Media, 2002.
20. Gh. Păun. *Introduction to Membrane Computing*, pages 1–42. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.
21. Gheorghe Păun. Computing with membranes. *Journal of Computer and System Sciences*, 61(1):108–143, 2000.
22. I. Pérez-Hurtado, D. Orellana-Martín, G. Zhang, and M J Pérez-Jiménez. P-lingua in two steps: flexibility and efficiency. *Journal of Membrane Computing*, 1:93–102, 2019.
23. F. Reiff. A Membrane System Simulator for Covid-19 Infection Diffusion. Bachelor Thesis in Bioinformatics, Department of Informatics, University of Verona, 2025.
24. F. Reiff and S. Erba. TEAM: Transmission of Epidemic Among Membranes. GitHub repository, 2025.
25. A. Remuzzi and G. Remuzzi. Covid-19 and Italy: what next? *The Lancet*, 2020.
26. Castaldo L. Rovetta A, Bhagavathula AS. Modeling the epidemiological trend and behavior of covid-19 in Italy. *Cureus*, 2020.
27. M. Ryan, E. Brindal, M. Roberts, and R. Hickson. A behaviour and disease transmission model: incorporating the health belief model for human behaviour into a simple transmission model. *Journal of the Royal Society Interface*, 21(215):20240038, 2024.
28. D. Valcamonica, A. D'Onofrio, M.M. Fareed, G. Franco, and C. Zandron. A dynamic behavior epidemiological model by membrane systems. submitted, 2024.
29. S. Verlan. Using the formal framework for P systems. In *International Conference on Membrane Computing*, pages 56–79. Springer, 2013.
30. R.K. Webster, S.K. Brooks, L.E. Smith, L. Woodland, S. Wessely, and G.J. Rubin. How to improve adherence with quarantine: rapid review of the evidence. *Public Health*, 182:163–169, 2020.
31. C. Zandron. Bounding the space in P systems with active membranes. *Journal of Membrane Computing*, 2(2):137 – 145, 2020.
32. C. Zandron, C. Ferretti, and G. Mauri. Solving np-complete problems using P systems with active membranes. In *Unconventional Models of Computation, UMC'2K: Proceedings of the Second International Conference on Unconventional Models of Computation,(UMC'2K)*, pages 289–301. Springer, 2001.
33. C. Zandron, A. Leporati, C. Ferretti, G. Mauri, and M. J. Pérez-Jiménez. On the computational efficiency of polarizationless recognizer p systems with strong division and dissolution. *Fundamenta Informaticae*, 87(1):79 – 91, 2008.