Modelling the remyelination process as a spatial stochastic system

Ludovica Luisa Vissat and Jane Hillston and Anna Williams

Abstract The regenerative process called *remyelination* aims to repair damaged areas of the central nervous system, caused by demyelinating diseases (e.g. multiple sclerosis). In many cases this process does not succeed in completely repairing the demyelinated lesions and the reasons of the failures are not clear. Since complex mechanisms and many factors regulate remyelination, it is helpful to describe this stochastic process using high-level modelling languages and to analyse its dynamics using model checking techniques. This study will help neurologists to reason about the various factors influencing this complex process and to create new hypotheses to test through lab experiments. In this work we introduce a novel process algebra called MELA. This high-level modelling language was used to describe the remyelination process. We have captured and compared different hypotheses about its functioning using a number of MELA models. We present the analysis of the spatio-temporal evolution of remyelination using Signal Spatio-Temporal Logic and Statistical Model Checking.

1 Introduction

Remyelination is a regenerative process that aims to repair damaged regions of the central nervous system, caused by demyelinating diseases, like multiple sclerosis. The central nervous system consists of the brain and the spinal cord; it controls most of the brain and body activity through the combination and transmission of information. It contains a dense network of nerve cells, also called neurons, which are able to transmit and receive information through electrical and chemical signals. They present three basic parts: the cell body (soma), the dendrites, and the axon, as shown in Figure 1. The cell body contains the genetic material (in the nucleus) and controls the cell activity. The dendrites are used to receive the electrical impulses while the axons are used to transmit the information through the axon terminals to other neurons. To guarantee the passage of the electrical impulse and to increase its propagation speed, the axons are surrounded by fatty sheaths known as myelin sheaths, which protect them and work as an electrically insulating layer. The loss of myelin sheaths can be caused by demyelinating diseases and it disrupts communication properties along axons. This damage results in a range of physical and cognitive problems. Remyelination is the process by which new myelin sheaths are created on demyelinated axons by oligodendrocytes. These cells are highly specialized cells which are able to form extensions of their membranes to replace the myelin sheaths. However, the remyelination process can fail to completely repair the demyelinated lesions; complex mechanisms and various factors regulate this process and the reasons for these failures are not clear. The ability to simulate and predict the evolution of the system supports neurologists and neuroscientists to understand the dynamics of this process in different possible scenarios and to reason about the several factors that influence it.

Ludovica Luisa Vissat · Jane Hillston
School of Informatics, University of Edinburgh, UK,

Anna Williams
MRC-Centre for Regenerative Medicine, University of Edinburgh, UK

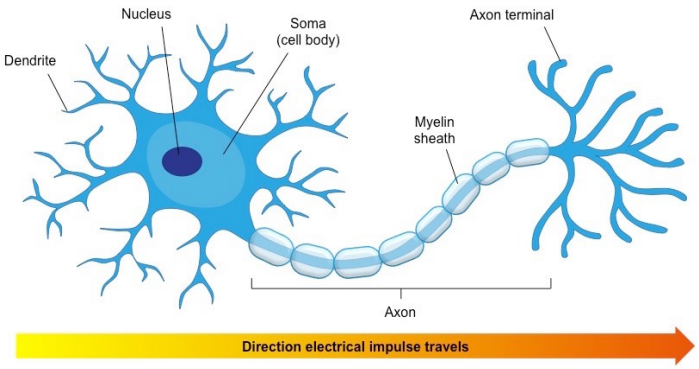


Fig. 1: Schematic representation of a neuron.

In our case study we focus in particular on lesions caused by multiple sclerosis. In the central nervous system, this disease is characterised by focal demyelinated lesions surrounded by normally myelinated white matter, which contains oligodendrocyte precursor cells (OPCs), which are cells of key importance for the remyelination process. OPCs are distributed throughout the brain and spinal cord, and are precursors to oligodendrocytes. These OPCs react to injury by duplicating and one of the OPCs then starts migrating. This activation of OPCs is thought to be due to molecules released by cells within the demyelinated lesion, including the chemotactic signalling proteins Semaphorin 3A and 3F [12]. These proteins are secreted by cells in the lesion, forming a concentration gradient emanating from the lesion and influencing OPCs within their range of action. Chemotactic molecules control the migration of OPCs. Our exemplars Semaphorin 3F and Semaphorin 3A act as an attractive signal and as a repulsive one respectively. If the OPCs reach the lesion, they start differentiating into mature oligodendrocytes. These mature myelin-producing cells recognise demyelinated axons and restore myelin sheaths providing electrical insulation and metabolic support to the underlying axon. If these steps proceed efficiently, then the lesion is remyelinated, and the production of signalling proteins is suspended. However, remyelination in multiple sclerosis is not always efficient, and can fail, leaving chronically demyelinated lesions.

Hypotheses for this failure include an imbalance of chemoattractive and chemorepulsive signals, leading to poor OPC recruitment and/or pro- and anti-differentiation signals in the lesions, or a loss of these signals over time.

Remyelination can be considered as a spatial stochastic system, with movement of cells regulated by signalling and spatial interactions among different parts of the system. In this work we present a novel process algebra MELA, a high-level modelling language recently introduced to model spatial stochastic systems, with a specific focus on spatial population models. We build and compare MELA models of different hypotheses about the functioning of the remyelination process.

MELA is a powerful modelling language incorporating several features which support the modelling of this and other spatial stochastic systems. First of all, it is a simple *high-level* language, which helps us to easily model and better understand complex interactions between agents, the components of the system. Secondly, MELA allows one to model the system in a compositional structure, describing the *individual-based* behaviour of each agent and then compose the set of individuals into a complete *population model*. For example, in the remyelination model, we will describe the individual behaviour of different cells types and signalling proteins and model the interactions between them, to subsequently understand and predict the evolution of the whole system. Moreover, to consider an aspect of key importance in spatial stochastic systems, MELA takes account of the underlying *spatial structure* of the system. It provides location attributes for the agents described in each MELA model and offers different choices of discrete spatial structures, dealing automatically with the set of locations and population vectors. In particular, the definition of the spatial structure and the description of agents' behaviour are introduced separately in the model. This separation can assist the modeller to understand the effect of different spatial assumptions on the dynamics of the system, by defining different spatial scenarios and keeping the agents' behaviour unchanged.

As support for the simulation of the models, MELA is equipped with a structural operational semantics, which provides an interpretation of the expressions of the language as a labelled transition system, which is used to derive the underlying continuous time Markov chain (CTMC). These stochastic processes are a useful alternative method to systems of Ordinary Differential Equations (ODEs), to describe the system dynamics. In particular, these models are more suitable in systems that present variability and individualised behaviour. Moreover, if the space is considered, as in the case of the remyelination process, the system variability increases, and consequently the use of stochastic models is even more appropriate. We will represent our spatial stochastic systems using *spatial population models*, which describe a collection of agents distributed in space, interacting and performing actions.

MELA supports different types of analysis that can be conducted to explore and analyse the behaviour of these systems. Our main focus is on spatio-temporal properties, which can be formally expressed using spatio-temporal logics, such as Signal Spatio-Temporal Logic (SSTL) [1]. This logic provides spatial and temporal modalities to describe system properties and their validity is verified using monitoring algorithms, which are automatic verification procedures over the output of a stochastic simulation. We chose to use SSTL to analyse the system dynamics since it presents features that fit perfectly with MELA settings, working with discrete representation of space and focussing on spatial stochastic systems. We estimate the satisfaction probability of SSTL properties by performing Statistical Model Checking over a set of simulation outputs. We perform our study to analyse the effect of different assumptions on the dynamics of the remyelination process. As main contribution of this work, we present the descriptive power of MELA as a modelling language, and the insights that we can get by modelling and comparing different system dynamics, particularly for a complex mechanism such as the remyelination process.

Related work: There has been only limited previous work on modelling multiple sclerosis reported in the literature, mostly focused on studying and predicting the progression of the disease, and only a few papers on the mechanisms underlying the disease. [2] presents a model of the competition and cross regulation of different T lymphocyte populations. These immune cells play a major role in disease progression. The authors use an agent-based approach and the programming language Netlogo; however, their focus is on the demyelination and they aim to investigate genetic predisposition for developing the disease rather than progression. In contrast, a detailed stochastic model of lymphocyte recruitment is presented in [3], using a process algebra, the biochemical stochastic π-calculus, to model this critical event in the pathogenesis of multiple sclerosis. The biochemical stochastic π-calculus has been used extensively to model and simulate biochemical processes. Such models support the analysis of the dynamics of disease systems and prediction of the outcome of interventions. Thus this work shares similar objectives to our own. In contrast, agent-based simulation is used in [4] to predict progression and long term levels of disability. Non-linear mixed effect modelling is used in [5], also to study disease progression, whilst [6] presents the use of spatially informed approaches for objective classification of multiple sclerosis subtypes, distinct clinical categories with distinct patterns of disease progression. To the best of our knowledge, there is no published work on modelling remyelination.

Chapter structure: This work is structured as follows. Firstly we define the syntax of MELA and we present MELA semantics, in terms of a labelled transition system, in Section 3. Section 4 introduces spatial population models, as well as simulation and analysis techniques. In Section 5 we describe the MELA model of the remyelination process while Section 6 presents the results of our study through a number of different scenarios. Finally, Section 7 report discussion, conclusions and future directions of our investigation.

2 MELA Syntax

Our novel process algebra MELA was initially developed for Modelling in Ecology with Location Attributes. When modelling an ecological system, it is natural to consider agents that can interact with each other and with the environment, making actions which may depend on the current state of the system. It is also crucial to take into account variations in the number of the agents, due to demographic and migration processes. Moreover, we have to be able to consider the interactions among agents, with others that are in their neighbourhood or even with more distant ones, depending on their own needs and characteristics. Therefore, key requirements for a process algebra to model ecological systems are modelling *birth-death* processes, an ability to describe *space*, concepts of *neighbourhood* and *movement*,

and the description of *actions* and the *influence* they may have on the behaviour of the agents. All of these aspects are considered in the specification of MELA models.

We now present the syntax of MELA, designed to capture the different possible agents' behaviours. We initially describe the component *AgentId* that defines the agent P with a location attribute l. While specifying the agents' behaviour, we do not explicitly declare all the locations of the discrete spatial structure. The location attribute l is used to specify that the agents are located in the structure. The component *AgentBody* describes the behaviour of the agent and the different type of actions α that the agent can perform, where α belongs to the set of action names $\mathscr{A}$. These actions can have different updates, varying the state of the agent, its location and the total number of the agent in the system. We will describe all the different type of actions later in this section. The environment is represented as a collection of different environmental conditions defined through *EnvId* and their behaviour is described using *EnvBody*. They can be in effect in multiple locations, as specified by the set L_E, but the component itself has no location. The definition of agents and environment behaviour is specified using *defAgent* and *defEnv*. Within the MELA syntax we distinguish *atomic components*, $\mathscr{C}_{at}$, and *compound components*, $\mathscr{C}_{com}$. The atomic components are described in the first part of the syntax and they represent the behaviour of a single agent, or a single environmental condition. Compound components *Com* and *Env* represent the parallel composition of one or more agents, or none, one or more environmental conditions, respectively. *Init* denotes the initial configuration of the system, which defines the initial distribution of the agents.

$$
\begin{aligned}
\mathit{AgentId} &::= P(l) \\
\mathit{AgentBody} &::= (\alpha,r).\mathit{Com} \mid (\alpha,r)\star P(l) \mid (\alpha,r)\triangleright P(\mathit{new}(l)) \mid \\
&\quad \rightarrow \{L\}(\alpha,r).\mathit{Com} \mid \rightarrow \{L\}(\alpha,r)\star P(l) \mid \rightarrow \{L\}(\alpha,r)\triangleright P(\mathit{new}(l)) \mid \\
&\quad \leftarrow (\alpha,p).\mathit{Com} \mid \leftarrow (\alpha,p)\star P(l) \mid \leftarrow (\alpha,p)\triangleright P(\mathit{new}(l)) \mid \\
&\quad \mathit{AgentBody}+\mathit{AgentBody} \mid \mathbf{nil} \\
\mathit{EnvId} &::= E \\
\mathit{EnvBody} &::= \rightarrow \{L_E\}(\alpha,r).\mathit{EnvId}
\end{aligned}
$$

$$
\begin{aligned}
\mathit{defAgent} &::= \mathit{AgentId} := \mathit{AgentBody} \\
\mathit{defEnv} &::= \mathit{EnvId} := \mathit{EnvBody} \\
\mathit{Com} &::= \mathit{AgentId} \mid \mathit{Com} \parallel \mathit{AgentId} \\
\mathit{Env} &::= \mathbf{nil} \mid \mathit{EnvId} \mid \mathit{Env} \parallel \mathit{EnvId} \\
\mathit{Init} &::= \mathit{Com} \parallel \mathit{Env}
\end{aligned}
$$

In the following paragraphs we explain the different type of actions that we can define in *AgentBody* and *EnvBody*:
$(\alpha,r).\mathit{Com} \mid (\alpha,r)\star P(l) \mid (\alpha,r)\triangleright P(\mathit{new}(l))$, where $\star ::= \uparrow \mid \downarrow$, denotes a *no-influence action*. It represents an action which is performed autonomously by the agent, without influencing or being influenced by others. The first type represents the change of the *state* of the agent, the second type represents a *demographic* action which can create ($\uparrow$) or destroy ($\downarrow$) an agent while the third one represents a *movement* action. When an agent performs a *no-influence action* α, it does so with the given rate r, representing the parameter of an exponential distribution[1], behaving subsequently as the component declared in the action. We assume that every action has a random duration and we can also describe the rate as a function $f(\mathscr{X})$, where $f : \mathscr{X} \mapsto \mathbb{R}_{\geq 0}$ is a function over the current state of the system $\mathscr{X}$, which includes the state of each agent and its location. In the presence of a movement action, the location of the resulting agent is expressed as a distribution over a set of possible locations, specified by $\mathit{new}(l) : \mathscr{L} \rightarrow Pr(N_l)$. $\mathscr{L}$ is the set of all locations which is specified and derived from the MELA model. $\mathit{new}(l)$ assigns a probability distribution to the set N_l ($N_l \subseteq \mathscr{L}$), the set of the possible locations in the defined neighbourhood of l, the current location of the agent. We can either use a given probability distribution over the neighbouring locations (e.g. uniform distribution $U(l_1,l_2,\cdots,l_n)$) or define a distribution empirically (e.g. $(l_1[p_1],l_2[p_2],\cdots,l_n[p_n])$), where p_i may depend on functions $f(\mathscr{X})$, for

[1] A random variable T has *exponential distribution* with parameter λ ($0 \leq \lambda < \infty$) if $P(T > t) = e^{-\lambda t}$, for all $t \geq 0$. The mean, or expected value, of an exponentially distributed random variable T with rate parameter λ is given by $E[T] = \frac{1}{\lambda}$.

example capturing a dependency on the number of the agents in some specific locations. It is important to notice that when the agent $P(l)$ performs a movement action only the location is changed.

$\rightarrow \{L\}(\alpha,r).Com \mid \rightarrow \{L\}(\alpha,r) \star P(l) \mid \rightarrow \{L\}(\alpha,r) \triangleright P(new(l))$, where $L ::= l \mid N(d) \mid all$ is an *influence action* that may affect agents present in the set of locations specified by the set L, $L \subseteq \mathscr{L}$. This set may be the current agent location l, the set $N(d)$ of the locations at given distance d from the current one or all the locations, denoted by *all*. The definition of distance between locations will depend on the chosen spatial structure, as explained in the following section *MELA model structure*. The possible updates given the *influence* action are defined exactly as for the *no-influence* action.

$\leftarrow (\alpha,p).Com \mid \leftarrow (\alpha,p) \star P(l) \mid \leftarrow (\alpha,p) \triangleright P(new(l))$ is a *passive action*, which will be always coupled with an influence action with the same name. It defines the potential effect of the action α on the agents performing it, which are affected with probability p. Also in this case, the probability p can be a constant value or a function of the current state of the system and the possible updates are defined as for the *no-influence* action.

The *choice operator* $+$, that represents the choice over possible behaviours, and the null component **nil** are standard. We use the derived syntax $\uparrow$ and $\downarrow$ to describe the creation and destruction of agents, in order to be able to explicitly track demographic actions. Thus $P(l) := (\alpha,r) \uparrow P(l)$ is simply an alternative representation of $P(l) := (\alpha,r).P(l) \parallel P(l)$ and $P(l) := (\alpha,r) \downarrow P(l)$ represents $P(l) := (\alpha,r).\mathbf{nil}$.

$\rightarrow \{L_E\}(\alpha,r).EnvId$, where $L_E ::= list \mid all$, denotes an environmental factor or condition. This captures an influence which can affect the evolution of the system but is not associated with a specific agent. This factor may be in effect in multiple or all the locations, as specified by the set L_E, but the component itself has no location.

We will now describe the MELA compound components *Com*, *Env* and *Init*.

$Com ::= AgentId \mid Com \parallel AgentId$ denotes a composition of agents, consisting of one or more agents in parallel. Similarly $Env ::= \mathbf{nil} \mid EnvId \mid Env \parallel EnvId$, denotes no, one or more environmental conditions that comprise the *environment* in effect in the system. We assume that the agents E, when they exist, behave independently.

$Init ::= Com \parallel Env$, defines the *initial configuration* of the system, consisting of the components present in the initial state of the system. Maintaining the separation between the agents and the environment means that we can readily experiment with the same composition of agents under different sets of environmental conditions.

MELA model structure The file containing the executable version of a MELA model consists in 4 different sections: #*Space*, #*Parameters*, #*Agents* and #*Initial Conditions*.

- The modeller can choose between different *discrete spatial structures*, defined in the specific section #*Space*, distinct from the agents' behaviour description:

$$\#Space ::= OneD(x) \mid TwoD(x,y) \mid ThreeD(x,y,z) \mid G = (\mathscr{L},E,w)$$

  OneD represents a discrete line, *TwoD* sets a 2D grid and *ThreeD* represents a 3D grid, with the dimensions defined at the beginning ($OneD(x), TwoD(x,y)$, $ThreeD(x,y,z)$) and default *bouncing* boundaries. We can have additional information in the MELA model, that allows us to specify *periodic* or *absorbing* boundaries and the rules of movement. With these spatial structures we consider the Manhattan distance between locations, which will be identified with coordinates. We also have the option of a weighted undirected graph, defined as $G = (\mathscr{L},E,w)$. $\mathscr{L}$ is the set of vertices, E the set of edges and w the function *cost*. We extend w to E^*, the transitive closure of E (set containing all the pairs of connected nodes). w gives the sum of costs of the lowest cost path between two different nodes. We will use this function w to provide the distance between two nodes of the graph.
- In the section #*Parameters* the modeller has to specify the action rates and the associated probabilities, in the presence of an influence action.
- The section #*Agents* presents the different type of agents present in the system and their behaviour, described following the syntax of MELA.
- The initial distribution of agents in the system is described in the section #*Initial Conditions*, using the rules of the structural congruence described in the next section and explicitly locating the agents in the spatial structure.

3 MELA Semantics

Before introducing the semantics, we define a structural congruence, $\equiv$, that will allow us to define the semantics and the models in a more compact and straightforward way.

$$\begin{aligned} P(l)[n] &\equiv \underbrace{P(l) \parallel \cdots \parallel P(l)}_{n \text{ times}} \\ P(l) &\equiv P(l)[1] \\ P(l)[n] \parallel \mathbf{nil} &\equiv P(l)[n] \\ P(l)[x] \parallel P(l)[y] &\equiv P(l)[x+y] \\ P(l) \parallel Q(l) &\equiv Q(l) \parallel P(l) \\ (P(l) \parallel Q(l)) \parallel R(l) &\equiv P(l) \parallel (Q(l) \parallel R(l)) \\ Com_1 \parallel Com_2 &\equiv Com_2 \parallel Com_1 \end{aligned}$$

MELA is equipped with a *structural operational semantics* [7], which describes the possible evolutions of the components, in terms of a *Labelled Transition System* $(\mathscr{C}, \rightarrow, \Theta)$. Here $\mathscr{C}$ is the set of components, $\rightarrow$ is the transition relation between components ($\rightarrow \subseteq \mathscr{C} \times \Theta \times \mathscr{C}$), and Θ is the set of labels ($mode, influence, action, value, location$), containing qualitative and quantitative information about the actions. Two states of the system, $\mathscr{X}_1$ and $\mathscr{X}_2$, are related if there is an action that may change the state from $\mathscr{X}_1$ to $\mathscr{X}_2$. Each label is a tuple ($mode, influence, action, value$, $location$), a rich label to capture the different features of all the possible actions:

- $mode ::= m \mid mm$ where $m ::= . \mid \uparrow \mid \downarrow \mid \triangleright$
  The entry *mode* records the effect of the action on the agents present in the system: . indicates a change of state, $\uparrow$ and $\downarrow$ relate to demographic actions and $\triangleright$ to a movement action. In the presence of an influence action, which is a synchronisation between two agents, we see two different entries, representing the effect on the populations of influencing and influenced agents respectively. We currently restrict to binary interactions so that at most one agent may be influenced in each step of the evolution, caused by the influencing action.
- $influence ::= \emptyset \mid L \mid L_E \mid \leftarrow$
  The second entry *influence* reflects the role of the agent in the action; this may be to be involved in no influence with other agents ($\emptyset$), to influence agents in some locations (set L, L_E) or to be influenced by the action ($\leftarrow$).
- $action ::= \alpha$ ($\alpha \in \mathscr{A}$, set of action names)
  The entry *action* provides the name of the action.
- $value ::= r \mid p$
  The entry *value* represents the *rate* parameter of the action, or the probability of influence when *influence* is equal to $\leftarrow$.
- $location ::= l_{atomic} \mid -$
  The last entry *location* captures the location of the atomic component that is involved in the action. The entry is equal to $-$ for an influence action made by an environment factor E, since E has no location. In the presence of an influence action, we have two atomic components affected by the action and they might be in different locations. In this case we indicate the *location* of the passive agent in the derived label, giving priority to the location where the action is felt.

Our rules are split into those defined for atomic components and those governing the behaviour of compound components. Here we present the atomic rules for the no-influence action. These axioms present the behaviour of the no-influence prefix terms which may give rise to the change of state of an agent, the creation of a new agent ($\uparrow$), the destruction of a present one ($\downarrow$) or the change of location of an agent. We define a function $loc : \mathscr{C}_{at} \rightarrow \mathscr{L}$, that returns the current location of an atomic component. We present the other semantic rules in detail at `https://ludovicalv.github.io/MELA/Semantics.pdf`.

no-influence action . $\dfrac{}{(\alpha, r).Com \xrightarrow{(.,\emptyset,\alpha,r,\bar{l})} Com}$ where $\bar{l} = loc((\alpha, r).Com)$

no-influence action $\uparrow$ $\dfrac{}{(\alpha,r)\uparrow P(l)\xrightarrow{(\uparrow,\emptyset,\alpha,r,\bar{l})}P(l)\parallel P(l)}$ where $\bar{l}=loc((\alpha,r)\uparrow P(l))$

no-influence action $\downarrow$ $\dfrac{}{(\alpha,r)\downarrow P(l)\xrightarrow{(\downarrow,\emptyset,\alpha,r,\bar{l})}\mathbf{nil}}$ where $\bar{l}=loc((\alpha,r)\downarrow P(l))$

no-influence action $\triangleright$

where $\bar{l}=loc((\alpha,r)\triangleright P(new(l)))$

$$\frac{}{(\alpha,r)\triangleright P(new(l))\xrightarrow{(\triangleright,\emptyset,\alpha,r\times\bar{p}_i,\bar{l})}P(l_i)} \qquad \bar{p}_i=\begin{cases}p_i, & \text{if } new(l)=(l_1[p_1],\ldots,l_n[p_n])\\ \frac{1}{n}, & \text{if } new(l)=U(l_1,\ldots,l_n).\end{cases}$$

4 Population model and dynamics

The labelled transition system generated by a MELA model can be mapped to a spatial population model via population semantics. This describes the collection of agents which can interact, take different states and move in space. From the MELA model description we extract the underlying discrete spatial structure, the different agent types, the set of transitions which can change the global state of the system, their rate, and the initial conditions.

4.1 Spatial Population Model

We define a *spatial population model* $\mathscr{M}$ as a tuple $\mathscr{M}=(\mathscr{S},G,\mathbf{X},\mathbf{X}_0,Tr)$ where:

- $\mathscr{S}=\{S_1,\ldots,S_n\}$ is the complete set of states that can be exhibited by the agents of all populations.
- $G=(\mathscr{L},E,w)$, a *finite weighted undirected graph*, representing the choice of underlying spatial structure of the spatial population model. Each possible spatial structure in MELA can be mapped to a weighted graph structure $G=(\mathscr{L},E,w)$:
  - $\mathscr{L}$ is the finite set of locations (nodes)
  - $E\subseteq\mathscr{L}\times\mathscr{L}$ is the set of connections (edges)
  - $w:E\to\mathbb{R}_{\geq 0}$ is the function *cost* (weights).
- $\mathbf{X}:\mathscr{L}\to\mathbb{N}_0^n$, where $\mathbf{X}(l)=(X_1,\ldots,X_n)\in\mathbb{N}_0^n$ is the state vector, that represents the state of the population in each location. The entries of the vector $\mathbf{X}(l)$ represent the number of agents in location l in the i^{th} state; therefore these *counting variables* are $X_i\in\mathbb{N}_0$.
- $\mathbf{X}_0:\mathscr{L}\to\mathbb{N}_0^n$, where $\mathbf{X}_0(l)$ is the initial state of the state vector, for each location.
- Tr is the set of transitions, $\tau_i=(\alpha_i,v_i,r_i)$, describing the events that change the global state of the system. Each transition consists of a label α_i, an update vector, $v_i:\mathscr{L}\to\mathbb{N}_0^n$ recording the change to each counting variable in each location due to the transition, and a rate function r_i, which may depend on the global state of the system.

4.2 MELA Spatial Population Model

A MELA model provides all the necessary information to derive the underlying spatial population model to simulate and analyse. We describe how to derive a spatial population model $\mathscr{M}=(\mathscr{S},G,\mathbf{X},\mathbf{X}_0,Tr)$ from a MELA model definition.

- $\mathscr{S}$: the set of agent states is extracted from the section *#Agents*, where we declare the different type of agents present in the system and their behaviour, following the syntax presented in Section 2.

- $G = (\mathscr{L}, E, w)$: the discrete spatial structure is defined in the section #*Space* of a MELA model. It can be chosen among different options. We can map each of these structures into a undirected weighted graph, with weights equal to 1 in the case of discrete line, 2D grid and 3D grid.
- $\mathbf{X}$: this vector is used to keep track of the system temporal evolution. It presents an entry for each agent type and each location. The length of this vector is equal to $|\mathscr{S}| \times |\mathscr{L}|$.
- $\mathbf{X}_0$: in the section #*Initial Conditions* we define the initial condition of the system, declaring how many agents of the different types are present in the different locations of the spatial structure. To make the declaration more compact, we declare just the agents that are present and their location, assuming that the number of the ones not declared is equal to 0.
- *Tr*: we use MELA semantics to generate the Labelled Transition System and derive the transitions that can change the global state of the system. Given a MELA model description, we extract the possible actions, the parameters for their rates and probabilities, the spatial structure and the population distribution over the spatial domain. Moreover, the labels of the LTS generated by the MELA model provide all the additional information to derive update vectors v_i and global rates r_i.

4.3 Stochastic dynamics and simulation

We can interpret the dynamical evolution of these spatial population models either stochastically as a Markov chain or deterministically as a system of ODEs; in our work we focus on the first interpretation. We can describe the temporal evolution of a spatial population model $\mathscr{M}$ using:

- σ, a *spatio-temporal trajectory* of $\mathscr{M}$. $\sigma : \mathscr{L} \times \mathbb{R}_{\geq 0} \to \mathbb{N}_0^n$ gives the state of the population vector for each location $l \in \mathscr{L}$ and each time $t \in \mathbb{R}_{\geq 0}$;
- Σ, a *set of spatio-temporal trajectories*, that will be used in the analysis.

To generate spatio-temporal trajectories, we implemented a MELA simulator, which performs stochastic simulations using Gillespie's Direct Method Stochastic Simulation Algorithm (SSA) [8]. Following the specification of a MELA model, the simulation algorithm consists of 4 steps.

- Step 0: Initialization. We initialise the number of agents in the system for each location and we extract the parameters for each action. We set the time $t = 0$
- Step 1: Calculation. We calculate the propensity function p_i for each action α_i, which gives the probability that the action α_i will occur in the next dt. We compute $p_f = \sum_i p_i$.
- Step 2: Generation. Two random numbers from the unit-interval uniform distribution are generated to determine the next action to occur as well as the time interval τ, according to p_f. The probability of a given action to be chosen is proportional to the number of agents that can perform it (*mass action law*).
- Step 3: Update. Knowing that a given action took place, we update the population vector according to the update vector and we set the time $t = t + \tau$.

Step 1-3 are iterated. The algorithm terminates typically when a final time $\mathscr{T}_\sigma$ is reached. The output of this discrete event simulation of a MELA model is a *spatio-temporal trajectory*, which keeps track of the population values at each location over time, for each time-step when an event (action) is realised. The domain of our spatio-temporal trajectories is more precisely defined as $\mathscr{L} \times \mathbb{R}_{<\mathscr{T}_\sigma}$, where $\mathscr{T}_\sigma$ is the temporal horizon. In addition, the MELA simulator keeps track of *metadata*, several pieces of information about the state of the system and the actions that are performed. In the metadata, for each action, we store the name, the action type, the type of update for active and passive agent and the location where active and passive agent are located when the action is performed. All of this information can be extracted from the rich labels in the derived LTS. These metadata are of key importance to understand the dynamics of the system. They give straightforward information about the performed actions that would be difficult to extract directly from the population vector changes at the first glance, especially for huge models with many populations and locations.

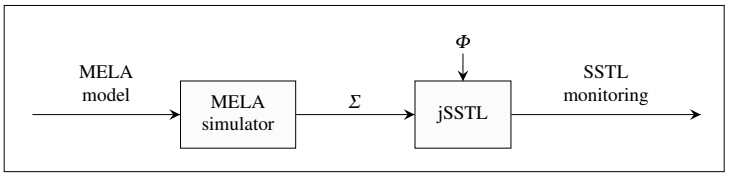


Fig. 2: Structure of modelling, simulation and analysis of spatio-temporal properties of a spatial population model, using MELA and jSSTL

4.4 Analysis via SSTL and Statistical Model Checking

The study of the dynamics of spatial stochastic systems requires suitable formal tools to describe and verify properties of their evolution, expressed in terms of temporal and spatial modalities. Spatio-temporal logics and model checking techniques provide analysis to evaluate the satisfaction probabilities of given properties, which are expressed through logical formulas. To define and verify spatio-temporal properties of our models, we use Signal Spatio-Temporal Logic (SSTL) [9], which is a spatial extension of Signal Temporal Logic (STL) [10], a temporal logic suitable for describing properties of real-valued signals. We will use this spatio-temporal logic, together with its monitoring techniques and model checking, to analyse and compare the spatio-temporal evolution of our MELA models under different scenarios. The syntax of SSTL is given by:

$$\varphi ::= \mu \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \, U^{[t_1,t_2]} \, \varphi_2 \mid \, \diamondsuit_{[w_1,w_2]} \, \varphi \mid \varphi_1 S_{[w_1,w_2]} \varphi_2$$

The SSTL *atomic proposition* μ is of the form $\mu \equiv (f \geq 0)$, $f : \mathbb{N}_0^n \to \mathbb{R}$, an inequality on expressions of population counts, given in the spatio-temporal trajectory. *Negation* $\neg$ and *disjunction* $\vee$ are the standard boolean operators and $U^{[t_1,t_2]}$ is the *bounded until* operator. This temporal operator $U^{[t_1,t_2]}$ is used to verify that the property φ_2 will be satisfied at some time instant in the interval $[t_1,t_2]$ and that at all preceding time instants φ_1 holds. SSTL introduces two spatial operators: the *bounded somewhere* operator $\diamondsuit_{[w_1,w_2]}$ and the *bounded surround* operator $S_{[w_1,w_2]}$, with w_1, w_2 real values, $w_1 \leq w_2$. The *bounded somewhere* operator requires that the property φ holds in a location reachable from the current one, with a cost w, $w \in [w_1,w_2]$. The operator *bounded surround* describes the property of being surrounded by a φ_2-region, while being in a φ_1-region: the formula $\varphi_1 S_{[w_1,w_2]} \varphi_2$ is true in a location l, if l belongs to a set of locations A where φ_1 holds, such that its external boundary $B^+(A)$ contains only locations satisfying φ_2.[2] Moreover, the locations in the $B^+(A)$ have to be reached from location l with a cost w, $w \in [w_1,w_2]$.

SSTL Boolean semantics SSTL presents a *boolean* semantics. Given a SSTL formula φ, it is interpreted over a spatio-temporal trajectory σ of $\mathscr{M}$, for each location $l \in \mathscr{L}$ and at time $t \in \mathbb{R}_{<\mathscr{T}_\sigma}$. It gives values in the boolean set $\mathbb{B} = \{T,F\}$ (T: true, F: false), depending on whether the observed trajectory σ satisfies the SSTL formula φ or not:

$$\beta(\mathscr{M},\sigma,l,t,\varphi) \in \mathbb{B}$$

SSTL is provided with monitoring algorithms to check the validity of the given formulas on a spatio-temporal trajectory. To perform our analysis we used jSSTL, a Java library implemented to support monitoring of SSTL properties, which can be found at `https://bitbucket.org/LauraNenzi/jsstl`.

Statistical Model Checking As illustrated in Figure 2, by simulating MELA models, we generate a set of spatio-temporal trajectories Σ. We perform Statistical Model Checking on Σ to estimate the probability that a given property is satisfied. Given a SSTL property, we verify it for each spatio-temporal trajectory, assigning to each one a truth value according to the boolean semantics. After this step, the estimate of the satisfaction probability is calculated in a frequentist manner, as the proportion of the spatio-temporal trajectories for which the formula is valid compared to the total number of spatio-temporal trajectories under examination. SSTL uses the binomial proportion confidence interval and the results are given at 95% confidence. We will show the results of our analyses in Section 6 of this work.

[2] The external boundary of a subset of locations A is defined as $B^+(A) := \{l \in \mathscr{L} \mid l \notin A \wedge \exists l' \in A \text{ such that } (l,l') \in E\}$

5 Remyelination: modelling and alnalysis

In this section we introduce and illustrate the MELA model, which describes the remyelination, and the analysis framework we used to conduct the study of system properties.

5.1 MELA remyelination model

In this section we present our MELA spatial population model $\mathscr{M} = (\mathscr{S}, G, \mathbf{X}, \mathbf{X}_0, Tr)$ representing a demyelinated area of the brain undergoing remyelination. The initial conditions for the remyelination process present a set of damaged locations in the centre of the area of study, with OPCs in the surrounding and potential signals, initially inactive, over the lesion.

Research in human brain tissue is done postmortem and it is confined to one time point by its nature. To investigate the remyelination and the different mechanisms in action, studies are often conducted on rodent brains. In this case, demyelinated lesions are created through stereotactic neurosurgery in mouse corpus callosum, using a 30 gauge needle, injecting 2 µl lysophosphatidylcholine (LPC, 1% weight/volume) over 4 minutes. The average size of the created lesion is 0.5 mm in all dimensions. Complete remyelination occurs on average after 28 days from the injection, which causes the formation of a lesion in about 3 days. We use this laboratory setting to build our MELA model. We also used the estimated rate of movement of OPCs, equal to 2 µm/min, as observed in *in vitro* cell migration assays.

- $\mathscr{S} = \{P, D, C, M, I, A, R, L, LN\}$: to model the remyelination process, our MELA model presents different types of agents: these represent the different types of cells, the different types of signals and the presence or absence of the lesion. Their different behaviour is presented in detail in the next section. The agents *P* represent the OPCs, initially located outside the lesion. Once these cells are activated by the signalling, we use the agent *D* to model the cell undergoing duplication. This process will generate a new precursor cell (agent *P*) and a moving OPC (agent *C*), which will move according to the signals. To represent the different types of signals, which will guide the OPC migration, we introduce the agent *A* to model an attractor signal (e.g. Semaphorin 3F) and the agent *R* for the repelling one (e.g. Semaphorin 3A). Moreover, we introduce the agent *I* to represents the inactive signal. Once the OPCs reach the lesion, they will differentiate and become mature cells, which we represent with the agents *M*. These mature oligodendrocytes will then support the repair process inside the lesion. We represent the presence of the lesion with the agents *L*. We do not explicitly model the final stage of the repair process, with production and wrapping of myelin sheaths, but we represent this as a simple change of state from *L* to *LN*, where *LN* represents the absence of the lesion. In our model for the moment we assume that the repair process gives lasting recovery: once a lesion is repaired, it cannot relapse. Therefore the only change of state allowed in the damaged area is from *L* to *LN*.
- $G = (\mathscr{L}, E, w)$: in our MELA remyelination model a section of the brain is modelled as a 15×15 grid of locations, each location corresponding to a 0.01 mm^2 planar section of cells. Each simulation therefore takes place over an area of 2.25 mm^2. We do not explicity model each axon in a different location but each location represents a brain area which contains multiple ones. The lesion area represents multiple axons which lack myelin sheaths. We chose this spatial structure to be able to use and compare with the results seen in vitro. The movement of cells in the remyelination process will be modelled considering the neighbouring cells at a Manhattan distance of 1, with the probability distribution over the neighbouring locations dependent on the signalling. The set of influences for the signalling will be of type $N(d)$, which in a 2D grid setting refers to the set of grid cells at a Manhattan distance of d.
- $\mathbf{X}$: the population vector $\mathbf{X}$ keeps track of the number of different agent types presented in the set $\mathscr{S}$ in the different grid locations. We can describe the output of a simulation as a time-indexed family of vectors $(\mathbf{X}(t))_{t \in \mathbb{R}_{<\mathscr{T}_\sigma}}$ where $\mathbf{X}(t)$ correspond to the state of the population vector at time t and $\mathscr{T}_\sigma$ is the temporal horizon.

- $\mathbf{X}_0$: in our MELA model we start with an initial lesion positioned centrally within the area of 15×15 grid cells, as shown in Figure 3. The grid cell size is equal to 0.1 mm and the diameter of the lesion is 0.5 mm. [3] Therefore in our 2D grid spatial structure we will represent the lesion dimension with 5 grid cells. From this point on, we will refer to the grid cells as *locations*, to distinguish from the physical cells that we are modelling in the system. We can find many physical cells inside a location. We chose to model the lesion not as a single location, but as an area of the grid formed by multiple locations, to be able to explore the remyelination process in more detail.
- *Tr*: the description of agent behaviours for the complete MELA model is presented in Figure 4. In the following paragraphs we explain the actions used in the model which give rise to the transitions in the spatial population model and illustrate how a MELA model is constructed.

actSA, **actSR**. These actions represent the activation of the production of signalling proteins inside the lesion, by the agent L. These processes are represented by a change of state of the agent I (inactive signal), to either A (attractor) or R (repellent). These local transitions will be performed according to the given probabilities $phSA$ and $phSR$.

$$I(l) := \leftarrow (actSA, phSA).A(l) + \leftarrow (actSR, phSR).R(l)$$
$$L(l) := \rightarrow \{l\}(actSA, hSA).L(l) + \rightarrow \{l\}(actSR, hSR).L(l)$$

sigA$_i$, **sigR**$_i$. Both signals will activate the precursor cells to duplicate. We assume the activation effect is uniform, with range equal to 0.5 mm. To model this, we define $N(i)$ as influence set, with $i = 1, \ldots, 5$, which indicates the set of grid cells with a Manhattan distance of i from the location where the signal is emitted. The agent D is used to model duplication, which we explain in the next point.

$$\begin{aligned} P(l) := {} & \leftarrow (sigA_1, pA_1).D(l) + \cdots + \leftarrow (sigA_5, pA_5).D(l) + \\ & \leftarrow (sigR_1, pR_1).D(l) + \cdots + \leftarrow (sigR_5, pR_5).D(l) \\ A(l) := {} & \rightarrow \{N(1)\}(sigA_1, hA_1).A(l) + \cdots + \rightarrow \{N(5)\}(sigA_5, hA_5).A(l) \\ R(l) := {} & \rightarrow \{N(1)\}(sigR_1, hR_1).R(l) + \cdots + \rightarrow \{N(5)\}(sigR_5, hR_5).R(l) \end{aligned}$$

dup. Through this action we model the duplication process of precursor cells and their following potential migration. Here the action will cause the generation a new moving OPC (agent C) and a new precursor cell (agent P).

$$D(l) := (dup, d).C(l) || P(l)$$

attr$_i$, **rep**$_i$. The moving OPCs will move according to the different signals produced inside the lesion, which can either attract or repel these cells. To model the movement of OPCs and establish their new position we define a probability distribution over their neighbouring locations, depending on the type of signal that causes their movement. For the action $attr_i$, $new(l)$ defines a uniform probability distribution over the neighbouring locations which are less distant from the source of the attractive signal; for the rep_i action, the process is mirrored, but over the neighbouring locations which are more distant from the source of repellent signal. With this choice, we model the attraction and repulsion of OPCs by the signals A and R. We assume that the effect of these signals will decrease linearly with distance, with a

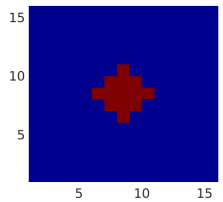


Fig. 3: Initial lesion (red area)

[3] The average size of the lesion is taken to be 0.5 mm in all dimensions. This was chosen as it is the size of an average lesion produced in a commonly used mouse model of demyelination and remyelination in mouse corpus callosum, as explain in Section **??**.

maximum range of influence d equal to 0.5 mm, from the location where the signal is produced. To model this aspect, the influence actions $attr_i$ and rep_i present an influence set defined as $N(i)$ and the respective rate parameters decrease with distance.

$$\begin{aligned} C(l) &:= \leftarrow (attr_1, pat_1) \triangleright C(new(l)) + \cdots + \leftarrow (attr_5, pat_5) \triangleright C(new(l)) + \\ &\quad \leftarrow (rep_1, pre_1) \triangleright C(new(l)) + \cdots + \leftarrow (rep_5, pre_5) \triangleright C(new(l)) \\ A(l) &:= \rightarrow \{N(1)\}(attr_1, at_1).A(l) + \cdots + \rightarrow \{N(5)\}(attr_5, at_5).A(l) \\ R(l) &:= \rightarrow \{N(1)\}(rep_1, re_1).R(l) + \cdots + \rightarrow \{N(5)\}(rep_5, re_5).R(l) \end{aligned}$$

diff. Once a moving OPC reaches the lesion, it will differentiate to become a mature cell, ready to start the repair process. The action *diff* is a local action, which will happen only when the agent C and the agent L are co-located, which represents the presence of the OPC inside the lesion.

$$\begin{aligned} C(l) &:= \leftarrow (diff, pdf).M(l) \\ L(l) &:= \rightarrow \{l\}(diff, df).L(l) \end{aligned}$$

inactiveA, inactiveR. The signals can degrade and become inactive again (I), after some time has elapsed. Using this modelling choice we allow the signal to be activated again, when the damaged area has not been repaired.

$$\begin{aligned} A(l) &:= (inactiveA, iA).I(l) \\ R(l) &:= (inactiveR, iR).I(l) \end{aligned}$$

repair. The action repair represents the final step of the remyelination process. Precursor cells that have differentiated inside the lesion to form mature oligodendrocytes will create new myelin sheaths to repair the lesion. After these agents M have played their role in the repair process, we will have a decrease of their number and a change of state for the lesion, that becomes repaired (LN).

$$\begin{aligned} M(l) &:= \leftarrow (repair, prp) \downarrow M(l) \\ L(l) &:= \rightarrow \{l\}(repair, rp).LN(l) \end{aligned}$$

endA, endR. These actions represent the inactivation of the signal when the lesion is repaired. Since we assume a repaired lesion cannot relapse, the signal cannot be reactivated.

$$\begin{aligned} A(l) &:= \leftarrow (endA, peA).I(l) \\ R(l) &:= \leftarrow (endR, peR).I(l) \\ LN(l) &:= \rightarrow \{l\}(endA, eA).LN(l) + \rightarrow \{l\}(endR, eR).LN(l) \end{aligned}$$

As already explained, once each required section of the MELA model is defined, we can simulate the system dynamics, performing stochastic simulations for a finite time horizon. The output of a stochastic simulation is a *spatio-temporal trajectory*, as defined in Section 4.1, which keeps track of the population counts in the different locations of the discrete spatial structure.

5.2 SSTL properties

In our analysis we will check SSTL properties such as:

$$\varphi := L > 0$$

$$
\begin{aligned}
P(l) &:= \leftarrow (sigA_1, pA_1).D(l) + \cdots + \leftarrow (sigA_5, pA_5).D(l) + \\
&\quad \leftarrow (sigR_1, pR_1).D(l) + \cdots + \leftarrow (sigR_5, pR_5).D(l);
\end{aligned}
$$

$$
D(l) := (dup, d).C(l) || P(l);
$$

$$
\begin{aligned}
C(l) &:= \leftarrow (attr_1, pat_1) \triangleright C(new(l)) + \cdots + \leftarrow (attr_5, pat_5) \triangleright C(new(l)) + \\
&\quad \leftarrow (rep_1, pre_1) \triangleright C(new(l)) + \cdots + \leftarrow (rep_5, pre_5) \triangleright C(new(l)) + \\
&\quad \leftarrow (diff, pdf).M(l);
\end{aligned}
$$

$$
M(l) := \leftarrow (repair, prp) \downarrow M(l);
$$

$$
I(l) := \leftarrow (actSA, phSA).A(l) + \leftarrow (actSR, phSR).R(l);
$$

$$
\begin{aligned}
A(l) &:= \rightarrow \{N(1)\}(sigA_1, hA_1).A(l) + \cdots + \rightarrow \{N(5)\}(sigA_5, hA_5).A(l) + \\
&\quad \rightarrow \{N(1)\}(attr_1, at_1).A(l) + \cdots + \rightarrow \{N(5)\}(attr_5, at_5).A(l) + \\
&\quad \leftarrow (endA, peA).I(l) + (inactiveA, iA).I(l);
\end{aligned}
$$

$$
\begin{aligned}
R(l) &:= \rightarrow \{N(1)\}(sigR_1, hR_1).R(l) + \cdots + \rightarrow \{N(5)\}(sigR_5, hR_5).R(l) + \\
&\quad \rightarrow \{N(1)\}(rep_1, re_1).R(l) + \cdots + \rightarrow \{N(5)\}(rep_5, re_5).R(l) + \\
&\quad \leftarrow (endR, peR).I(l) + (inactiveR, iR).I(l);
\end{aligned}
$$

$$
\begin{aligned}
L(l) &:= \rightarrow \{l\}(actSA, hSA).L(l) + \rightarrow \{l\}(actSR, hSR).L(l) + \\
&\quad \rightarrow \{l\}(diff, df).L(l) + \rightarrow \{l\}(repair, rp).LN(l);
\end{aligned}
$$

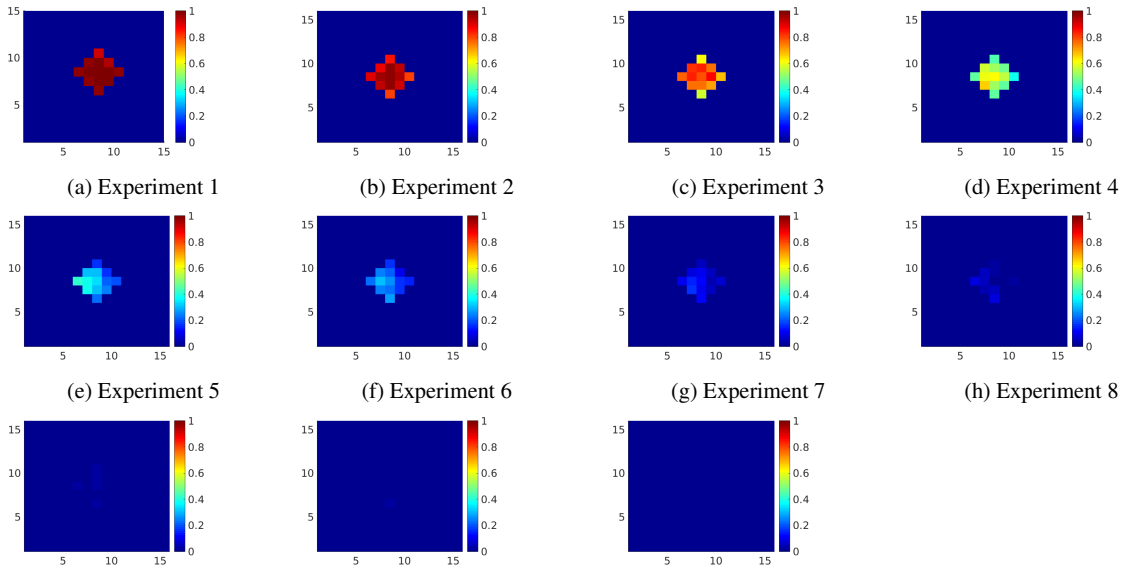
$$
LN(l) := \rightarrow \{l\}(endA, eA).LN(l) + \rightarrow \{l\}(endR, eR).LN(l);
$$

Fig. 4: MELA model: agents description. The complete model can be found at `https://ludovicalv.github.io/ModelMS`

that identifies the presence of damaged locations (presence of agents L). This property φ will be used to study the spatio-temporal evolution of the damaged area. We will analyse how the area evolves, if the lesion is repaired over time and characteristics of the repair process.

6 Case study

We conducted several experiments to analyse the remyelination process under different scenarios. Starting from the initially defined MELA model, we study the model under a range of different hypotheses, analysing their different impact on the dynamics of the system. In particular, in scenario 1, we analyse the remyelination process under different probabilities of expressing attractor and repelling signals. In scenarios 2 and 3 we study the system dynamics under different rates of OPC movement, while in scenarios 4 and 5 we explore different tactics of remyelination. Ranges of activation and signalling, rates of activation, cessation of signalling protein production, and rates of activation, duplication, differentation and maturation of OPCs are not subject to change in these scenarios. The initial dimension of the lesion and the area of interest are also unvaried throughout the different experiments. For each experiment, we will present the settings, the study of the resulting remyelination processes under the different conditions and the conclusions we can draw. In each case we simulate the whole process for 25 days, starting from an existing lesion. This is chosen to correspond to the wet lab experimental set up. We performed the analysis using jSSTL and 100 MELA spatio-temporal trajectories.



(a) Experiment 1 (b) Experiment 2 (c) Experiment 3 (d) Experiment 4

(e) Experiment 5 (f) Experiment 6 (g) Experiment 7 (h) Experiment 8

(i) Experiment 9 (j) Experiment 10 (k) Experiment 11

Fig. 5: Validity of the SSTL property $\varphi := L > 0$ (presence of the lesion), at t = 25 days, for experiments of scenario 1

We chose to analyse this number of spatio-temporal trajectories as balance between efficiency and consistency, looking at exploratory spatio-temporal trajectories. Each simulation required on average 4 minutes.

6.1 Scenario 1: Different probabilities of expressing signals

In this experiment we want to analyse the different impact on the remyelination process depending on the produced signalling proteins Semaphorin3F (attractor agent A) and Semaphorin3A (repellent agent R) produced inside the lesion. We model this aspect using the probabilities *phSA* and *phSR* of activating the different signals which will guide the migration of moving OPCs. In this model the range of both signals is 0.5 mm and the estimated rate of the cell movement is 2 µm/min.

Experiment	1	2	3	4	5	6	7	8	9	10	11
phSA	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
phSR	1.0	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	0.0

In Figure 5 we show the probability of finding damaged locations after 25 days, inside the initially damaged area. The colour scheme goes from red (probability equal to 1) to blue (probability equal to 0). In Figure 6 we show the average probability for the locations that are part of the lesion to remain damaged over time. As expected, all of them are initially damaged so the average probability is equal to 1 at time 0, and then the probability value decreases over time, differently depending on the different experiments. As we were expecting, the remyelination process is more effective when we have a relative higher probability of expressing Sema3F (the attractive signalling) with respect to expressing Sema3A. The probability of finding damaged cells reduces with time and more substantially when we have more attractor signals. Indeed in experiments $10-11$ there is a high probability that the lesion is completely repaired.

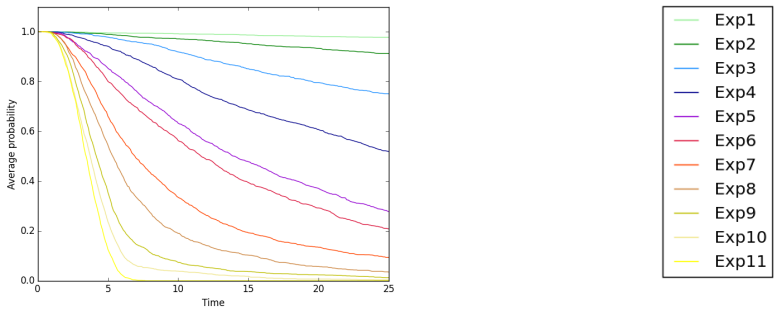
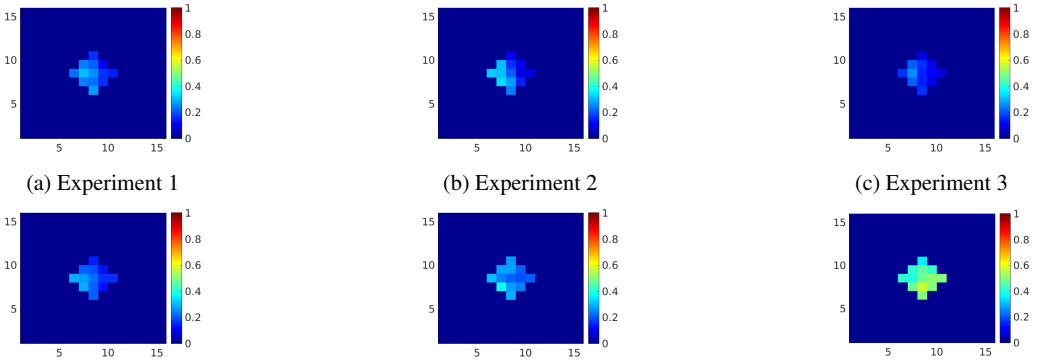


Fig. 6: Average probability of satisfying the SSTL property φ inside the lesion, for experiments of scenario 1

6.2 Scenario 2: Different rates of OPC movement - faster movement

One of the interesting factors that might influence the remyelination process is the rate of movement of OPCs. Based on experimental observation, the expected movement rate of these cells (agents C) is 2 µm/min = 2.88 mm/day, which can vary depending on the concentration of signalling proteins. We explore the dynamics of remyelination with different movement rates for OPCs, when there is equal probability of expressing either attractor or repellent signals. The resulting probabilities of finding damaged locations after 25 days are shown in Figure 7.

Experiment	1	2	3	4	5	6
Increase	×1	×1.2	×1.5	×2	×5	×10
rate (µm/min)	2.0	2.4	3.0	4.0	10.0	20.0
mm/day	2.88	3.456	4.32	5.76	14.4	28.8



(a) Experiment 1 (b) Experiment 2 (c) Experiment 3

(d) Experiment 4 (e) Experiment 5 (f) Experiment 6

Fig. 7: Validity of the SSTL property $\varphi := L > 0$ (presence of the lesion), at t = 25 days, different rate of OPCs, for experiments of scenario 2

Although initially we expected an improvement with faster rates, the results of this experiment show that an increase in the rate of movement of OPCs does not improve the remyelination process. As shown in Figure 8, we can observe that the average probability for each part of the lesion to remain unrepaired does not vary significantly. Moreover, if the speed increases significantly, we observe a worse result with respect to the remyelination process. This result suggests further lab experiments to investigate if the impact of faster movement of OPCs, which can be induced by a stronger gradient of the chemoattractant, is similar *in vitro* with respect to the effectiveness of remyelination. In addition, the increase of speed might have a counter-productive effect on the maturation of the OPCs, as we explain in Section 6.4.

6.3 Scenario 3: Different rates of OPC movement - slower movement

We performed experiments to explore the remyelination process with slower rates of OPC movement. The results of the analysis are shown in Figures 9 and 10. We can observe a slight improvement in the remyelination process with a slower rate of OPC movement. Given the nature of our model, if OPCs move slower, it is more likely that the maturation will occur, since there will be higher probability that the maturation action takes place (*race condition*).

Experiment	1	2	3	4	5	6
Decrease	$\times 1.0$	$\times 0.5$	$\times 0.1$	$\times 0.02$	$\times 0.01$	$\times 0.002$
mm/day	2.88	1.44	0.288	0.0576	0.0288	0.00576

6.4 Failure to repair: not enough OPCs or no maturation process?

In Section 6.2 we showed that an increase of the speed of OPCs does not impact positively on the remyelination process. To have better insight into this scenario, we analyse the average number of OPCs in the system and the number of mature cells inside the lesion, with faster and slower movement rates. We want to understand if the negative effect is given by either not enough OPCs available or insufficient maturation processes that are fundamental to repair inside the lesion. We have analysed the average population of OPCs and mature cells, analysing the output of experiments of scenarios 2 and 3. We have observed a higher number of OPCs in the area with a faster movement than with a slower movement. However, the faster movement acts against maturation, and we have seen that there are fewer mature cells when the movement rate is faster. We have observed that a faster rate reduces the ability of these cells to reach the lesion. By analysing the distribution of OPCs in the system, we have observed that if they move faster, they tend to quickly move outside the range of signalling, becoming unavailable.

6.5 Scenario 4: instant stopping inside the lesion

In scenario 4 we analyse the remyelination process with a different behaviour of the moving OPCs. We assume that the OPCs will stop quickly once they find the lesion. To introduce this aspect in our MELA model, we add a new agent *CS*, which represents the *C* agents that have stopped, having found a damaged location. Splitting the behaviour in this way allows the stop action to be given a very fast rate, making it very likely to have precedence over other possible actions. In this new model, the differentiation process will be carried out by agents *CS*, in the same way as the agents *C* previously.

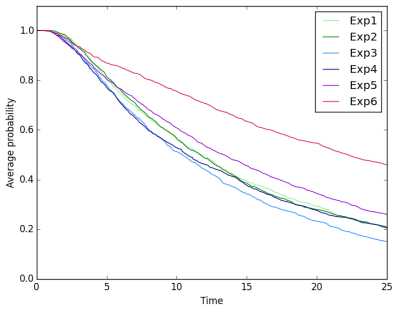
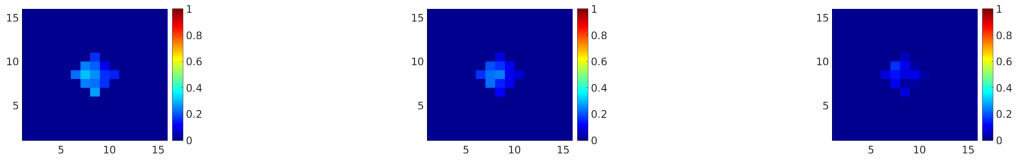


Fig. 8: Average probability of satisfying the SSTL property φ inside the lesion, for the different faster movement rates, for experiments of scenario 2



(a) Experiment 1 (b) Experiment 2 (c) Experiment 3

(d) Experiment 4 (e) Experiment 5 (f) Experiment 6

Fig. 9: Validity of the SSTL property $\varphi := L > 0$, at t = 25 days, different (slower) rate of OPCs, for experiments of scenario 3

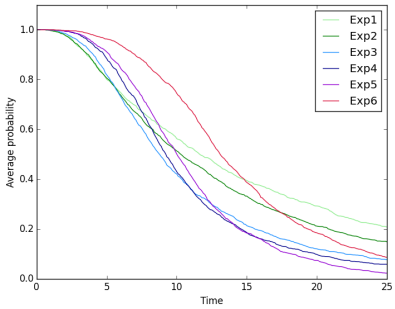


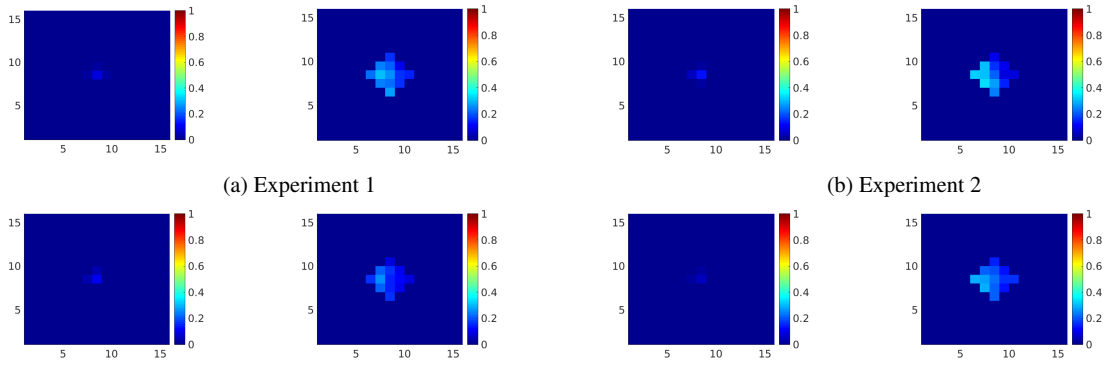
Fig. 10: Average probability of satisfying the SSTL property φ inside the lesion, for different slower movement rates, for experiments of scenario 3

$$
\begin{aligned}
C(l) &:= \leftarrow (stop, ps).CS(l);\\
CS(l) &:= \leftarrow (diff, pdf).M(l);\\
L(l) &:= \rightarrow \{l\}(stop, s).L(l) + \rightarrow \{l\}(diff, df).L(l)
\end{aligned}
$$

The results of these experiments are shown in Figures 11 and 12. In Figure 11, we show the difference of the repair process after 25 days, with and without the stop option, for different rates from the scenario 2. Looking at the figures, we observe a substantial improvement of the repair process when cells stop as soon as they encounter damage and that for each different rate the lesion is almost entirely repaired.

6.6 Scenario 5: stopping in the centre of the lesion

In the previous scenario, the action *stop* was a local and fast action, performed inside the lesion by the moving OPCs. Under this assumption, we observe that the repair process happens initially in the exterior part of the lesion. In this section, we analyse what the impact on the remyelination process would be if a different stopping mechanism was in operation: the moving OPCs will stop once they get to the centre of the lesion. After the centre location gets repaired, the attraction feature will move to the neighbouring locations that are still damaged, and so on. To represent this mechanism, we add to the model the agent *LS*, to indicate the damaged locations that have this attractive feature. The



(a) Experiment 1 (b) Experiment 2

(c) Experiment 3 (d) Experiment 4

Fig. 11: Faster rate, validity of the SSTL property $\varphi := L > 0$ (presence of the lesion), with and without the stop option, t = 25 days

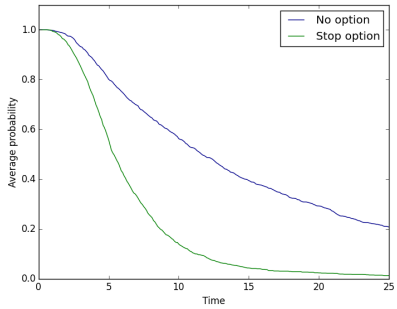


Fig. 12: Comparison between average probabilities, with and without stopping option (Experiment 1)

agents *LS* behave exactly in the same way as agents *L*, with respect to the activation of the signalling, maturation of the cell and repair process. Initially, the centre of the lesion will be the only attractive location. In this new model, we add the local action *stopLS*, that the agents *C* will perform when they reach an attractive location.

$$LS(l) := \rightarrow \{l\}(stopLS, s).LS(l)$$
$$C(l) := \leftarrow (stopLS, ps).M(l)$$

Once the attractive and damaged centre of the lesion is repaired, the neighbouring locations will become attractive. To model this process, we firstly add to the model the agent *LB*.

$$LS(l) := \rightarrow \{l\}(repairS, rpS).LB(l);$$
$$M(l) := \leftarrow (repairS, prpS) \downarrow M(l);$$

The agent *LB* represents an intermediate state. It corresponds to a repaired attractive part of the lesion that will pass the attractive feature to its neighbouring damaged locations, before getting to the final stage *LN*. To represent this action, we add to the model the action *transmit*. This action is fast and performed by the agents *LB*.

$$LB(l) := \rightarrow \{N(1)\}(transmit, c).LB(l);$$
$$L(l) := \leftarrow (transmit, pc).LS(l);$$

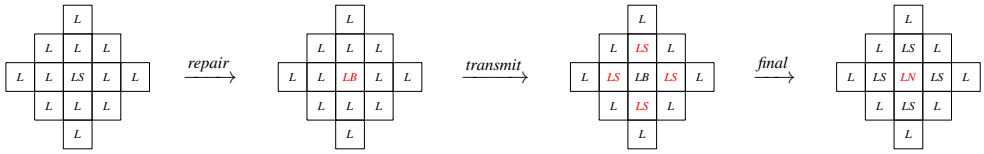


Fig. 13: Evolution of *LS* agent. This figure shows the successful change of the state of all the neighbouring agents, by the agent *LB*, before changing its state in *LN*, the final stage.

By giving a faster rate to the *transmit* action, it is more likely that this action is performed before the action *final*, which will change the state of the agent *LB*, that will eventually become a repaired part of the lesion (agent *LN*). This mechanism is illustrated in Figure 13. As we observe in Figure 14, the resulting analysis of our model shows a repair process, starting from the centre of the lesion. This model might not represent the mechanism that happens in reality, since current lab studies of remyelination in rodents suggest that white matter demyelinated lesions repair from the outside-in.

$$LS(l) := \rightarrow \{l\}(stop,s).LS(l) + \rightarrow \{l\}(repairS,rpS).LB(l);$$
$$LB(l) := (final,f).LN(l)$$

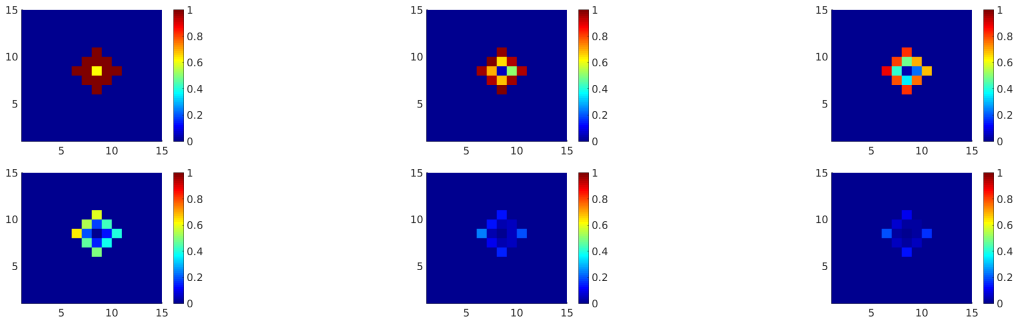


Fig. 14: Validity of the SSTL property $\varphi := L > 0$ (presence of the lesion), time = 2, 5, 7 (on the top row), time = 10, 20, 25 days (on the second row), for scenario 5

7 Conclusions and Future Work

In this work we have presented the process algebra MELA and the analysis techniques we used to model and study the remyelination process in multiple sclerosis under different assumptions.

The process algebra MELA was developed with a specific focus on spatial population models and ecological systems. It provides the modeller with a rich set of possible description of agents' behaviour and discrete spatial representation. MELA agents can perform different types of actions, which may influence other agents in the system, depending also on their respective position. From a MELA model, we can directly extract the underlying spatial population model and simulate its dynamics on a finite temporal horizon using Gillespie's Stochastic Simulation Algorithm. A set of output of this procedure (*spatio-temporal trajectories*) is used to perform Statistical Model Checking and formal verification of spatio-temporal properties of the evolution of the system, expressed using the spatio-temporal logic SSTL.

The versatility of MELA for representing spatial population models allowed us to easily build an initial model of the remyelination process and analyse the system dynamics under different scenarios, comparing the impact of the various assumptions on the evolution of the system. In scenario 1 we explore the repair process with different probabilities of expressing the two different types of signal (attractive and repellent). As expected, we observed an improvement in the repair process with a relatively higher probability of expressing the attractive signal. In scenario 2 and 3 we analysed the remyelination process, assuming different movement rates of OPCs. The results do not show a significant difference among the experiments, except for the extreme cases. These extreme cases have informed our biological work. Therapeutics are being sought that increase the speed of movement of OPCs to demyelinated lesions, aiming to improve remyelination. However, this modelling indicates that increasing the speed of OPCs too much may be detrimental to repair. To analyse these results in more detail, we looked at the number of OPCs inside the lesion in the various cases, observing that the OPCs that move too fast tend to move outside the signalling range. In scenario 4 and 5 we analyse different stopping mechanisms for the OPCs, once they reach the lesion site. We observed an improvement in remyelination if the OPCs are able to detect the lesion as soon as they find it, and then stop and mature. This would suggest that remyelination would start at the edge of a demyelinated lesion and work towards the middle of the lesion, and this is seen in repair of human multiple sclerosis lesions. Conversely, if we assume that they start repairing from the centre of the damaged area, we observe an inside-out repair process which is not observed in biological studies.

We observed some predictable behaviours, such as a more effective remyelination process in the presence of more attractive signalling, and some interesting results, such as the failure of process improvement given faster moving OPCs. These results have generated new hypotheses to be tested both in the lab in mouse models of demyelination and remyelination and by new modelling paradigms. We will next study remyelination including a distinct maturation signal, including proliferation of cells and complex signalling influences. Moreover, we can generate different initial conditions, such as with different sized lesions, multiple lesions occurring at the same time, to analyse the movement of OPCs between them. Our basic model has already given ideas to test in biological systems, and further information from these biological systems will be fed back into this computational model, in an iterative manner.

As future enhancement of our modelling framework, we would consider a finer grain representation of movement, as introduced in [11]. This extension would allow a more detailed description of the cell movement following signalling gradients, providing a combination of process-algebraic and diffusion models.

References

1. L. Nenzi and L. Bortolussi, Specifying and monitoring properties of stochastic spatio-temporal systems in signal temporal logic, in 8th International Conference on Performance Evaluation Methodologies and Tools, VALUETOOLS, 2014.
2. M. Pennisi, A.-M. Rajput, L. Toldo, and F. Pappalardo, Agent based modeling of tregteff cross regulation in relapsing-remitting multiple sclerosis, BMC Bioinformatics, vol. 14, 2013.
3. P. Lecca, C. Priami, P. Quaglia, B. Rossi, C. Laudanna, and G. Constantin, A stochastic process algebra approach to simulation of autoreactive lymphocyte recruitment, SIMULATION, vol. 80, no. 6, pp. 273-288, 2004.
4. M. Veloso, An agent-based simulation model for informed shared decision making in multiple sclerosis, Multiple Sclerosis and Related Disorders, vol.2, pp.377-384, 2013
5. N. Velez de Mendizabal, M. M. Hutmacher, I. F. Troconiz, J. Goñi, P. Villoslada, F. Bagnato and R. R. Bies, Predicting Relapsing-Remitting Dynamics in Multiple Sclerosis Using Discrete Distribution Models: A Population Approach, PLOS ONE 9(1), pp.1-11, 2013
6. B. Taschler, T. Ge, K. Bendfeldt, N. Müller-Lenke, T. D. Johnson and T. E. Nichols, Spatial Modeling of Multiple Sclerosis for Disease Subtype Prediction, MICCAI 2014, pp.797-804, 2014
7. G. D. Plotkin, The origins of structural operational semantics, The Journal of Logic and Algebraic Programming, vol. 60, pp. 3-15, 2004.
8. D. T. Gillespie, Exact stochastic simulation of coupled chemical reactions, J. Phys. Chem., vol. 81, no. 25, pp. 2340-2361, 1977.
9. L. Nenzi, L. Bortolussi, V. Ciancia, M. Loreti, and M. Massink, Qualitative and quantitative monitoring of spatio-temporal properties, in Runtime Verification - 6th International Conference, RV, pp. 21-37, 2015.
10. O. Maler and D. Nickovic, Monitoring Temporal Properties of Continuous Signals, pp. 152-166. Springer Berlin Heidelberg, 2004.
11. M. Klann, L. Paulevé, T. Petrov and H.Koeppl, Coarse-Grained Brownian Dynamics Simulation of Rule-Based Models. Computational Methods in Systems Biology. CMSB 2013. Lecture Notes in Computer Science, vol 8130. Springer, Berlin, Heidelberg
12. A. Williams, G. Piaton, M. Aigrot, A. Belhadi, M. Théaudin, F. Petermann, J. Thomas, B. Zalc and C. Lubetzki, Semaphorin 3A and 3F: key players in myelin repair in multiple sclerosis?, Brain, vol. 130, no. 10, 2007, pp. 2554-2565