

Lymph node swelling combined with temporary effector T cell retention aids T cell response in a model of adaptive immunity

Supplementary File 2: Supplementary Methods B

This section presents the logic underlying the sub-functions called during the simulation. The order each method is called is depicted in the 'Main' diagram (Fig 2B) and summarised below. Properties stored in T cells and DCs are shown in S1 File, Fig C.

- The Builder file and Constants files are the key files used to create the environment and the agents at time zero. After this the Context file is the key initiator.
- Each time-step, in the context class, the number of T cells to add and the value of properties are calculated. Positions to place T cells are selected from a list of possible coordinates and new instances (copies) of the relevant T cell class. (eg. Cognate or Non-cognate) are produced.
- Calling the '**UpdateTCs**' function applies updates to the history of all agent's present, eg. how long since they have entered.
- Subfunctions that result in removal of an agent, for example, a T cell dying (**UpdateTCsAgeing**, **RemoveTCsAgeing**), also requires removal of any links **RemoveLinks** (projections) to DCs, and the update of relevant DC properties, eg. number of T cells bound.
- T cells move (**TCsMove**), and any DCs to enter are added (**AddDCs**).
- **UpdateDCs** includes updating MHC expression, DC states, such as whether or not they are licenced, and timecounts, such as duration of links to T cells. Interactions are ended by removing Links (**removeLinks**) after checking rules that take into account T cell type and interaction history. The DCs are aged (**AgeDCs**) and may be removed, along with any corresponding links.
- Cognate T cells then undergo updates (**UpdateCognateTCs**) including updating age and history counts, gaining or losing stimulation, and assessing probability of, then carrying out activation/differentiation and proliferation.
- Finally, functions are called in the Context class that update counters of inflammation state (sum of MHC values), count T cells present, and adjust the volume and geometry of the LN accordingly.

2.1 UML diagrams describing the rules used to build the ABM

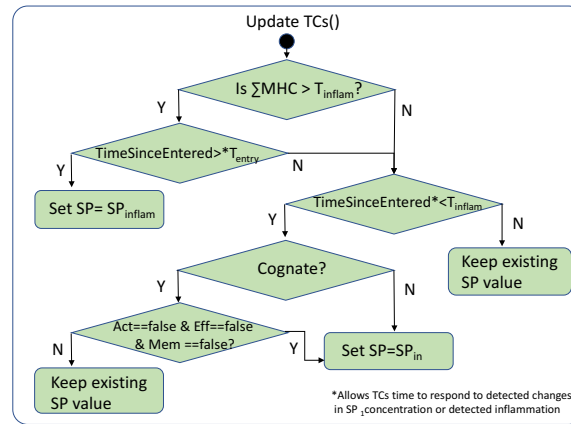


Figure 1: UML diagram for 'UpdateTCs'. By checking the time since the TC entered, a 'lag time' representing time for detection of signals and receptor internalisation is permitted. If the cell is newly entered, and not an activated or effector TC, SP_{1r} is down-regulated to the value of SP_{in}

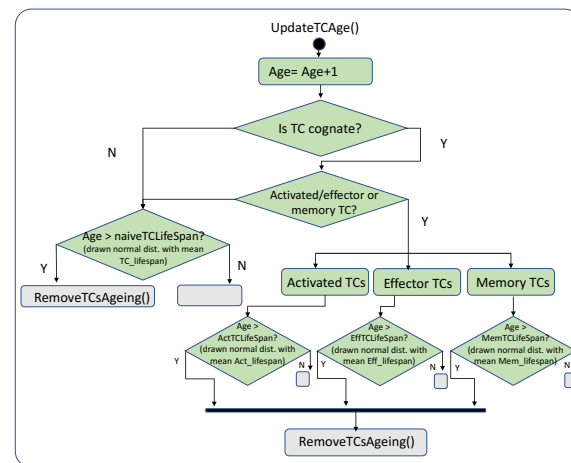


Figure 2: UML diagram for 'UpdateTCsAgeing'. Lifespan is dependent on TC state (see S1 File, Table B).

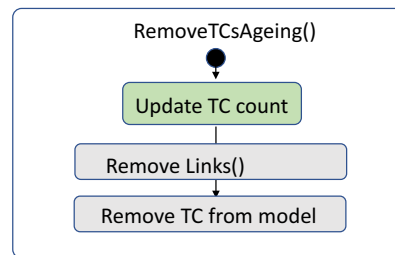


Figure 3: UML diagram for 'RemoveTCsAgeing'. The context stores a counter of how many TCs are present which is updated when a T cell is removed. Any links to DCs are also removed.

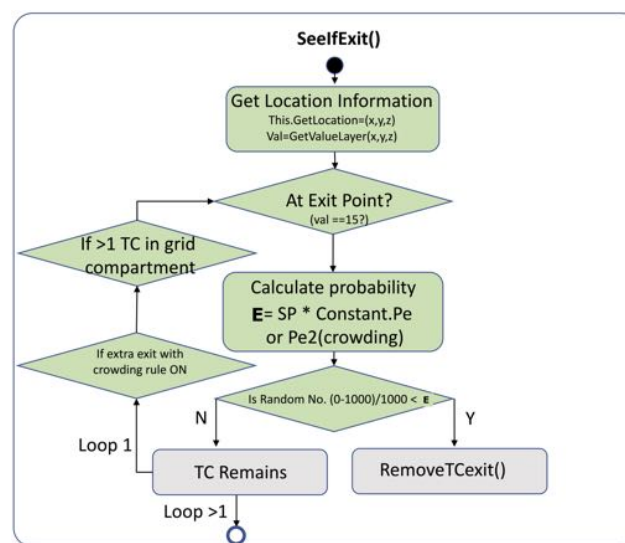


Figure 4: UML diagram for 'SeelfExit'. T cells check their environment by checking the 'value-Layer'. If in an exit area, the probability of exit (E) is calculated by multiplying a constant P_e by the individual TCs relative $S1P1r$ expression. A random number between 0 and 1 is generated and if less than E , the T cell will exit. A second check of TC crowding (crowding rule) is applied in some simulations.

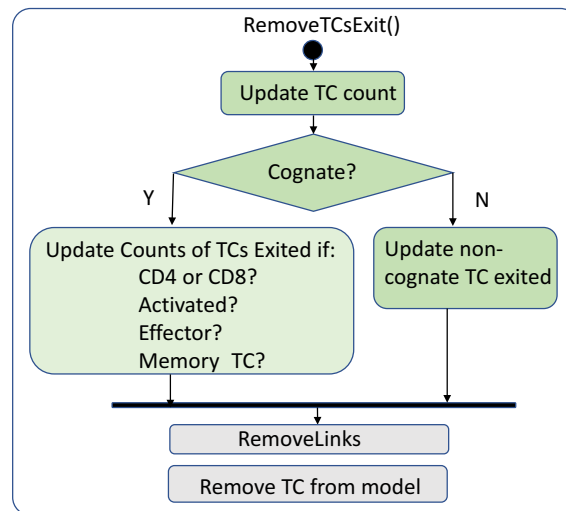


Figure 5: UML diagram for 'RemoveTCExit'. Ageing and TC exit are treated differently due to different counters (stored as Context properties) being updated.

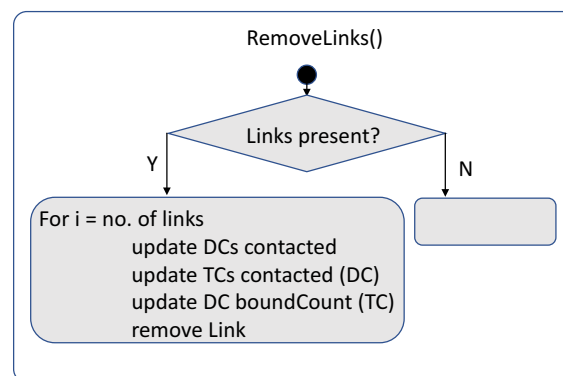


Figure 6: UML diagram for 'RemoveLinks'. Links between TCs and DCs are formed using a projection network, a feature of RepastSimphony. Before removal, properties within the linked agents relating to interactions are updated.

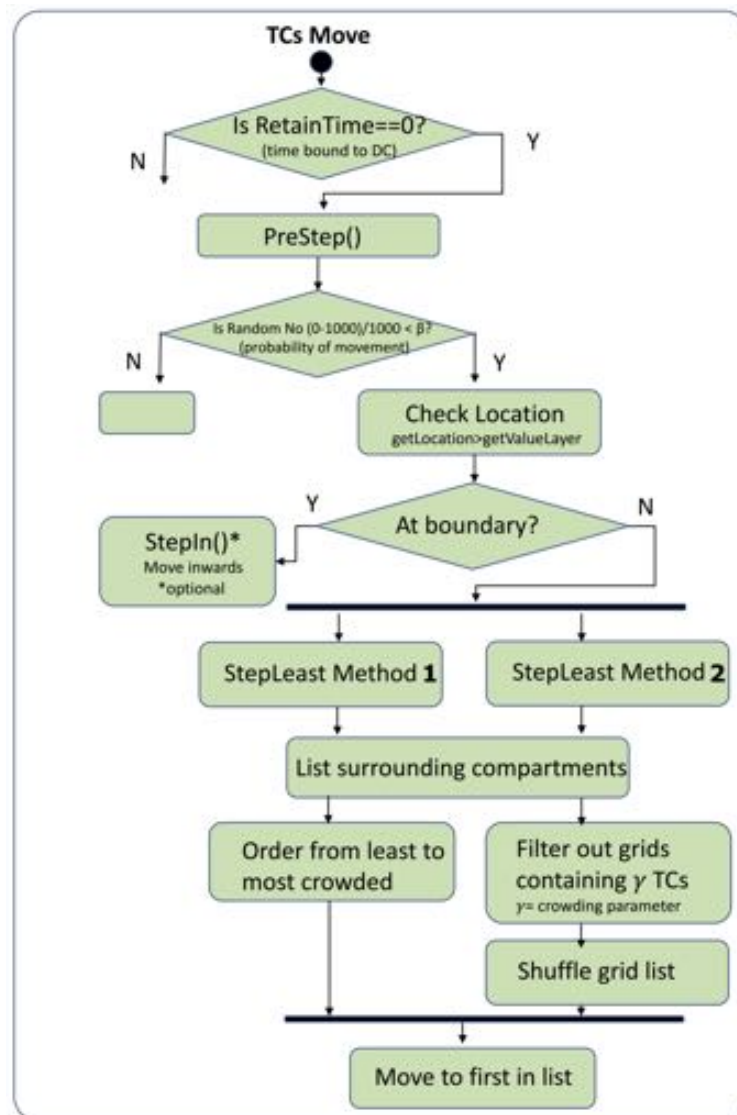


Figure 7: UML diagram for 'TCsMove'. Only unbound T cells will migrate. After determining whether the cell will move this time-step (using probability of movement β), the model has the capacity for different movement options. The GridCell class makes it possible to provide a list of neighbouring compartments and their contents. Method 1 involves iterating over the list and moving to the least crowded neighbouring compartment. Method 2 filters the list, considering the maximum TCs permitted per grid compartment (dictated by crowding parameter γ), then randomly selects from the remaining positions.

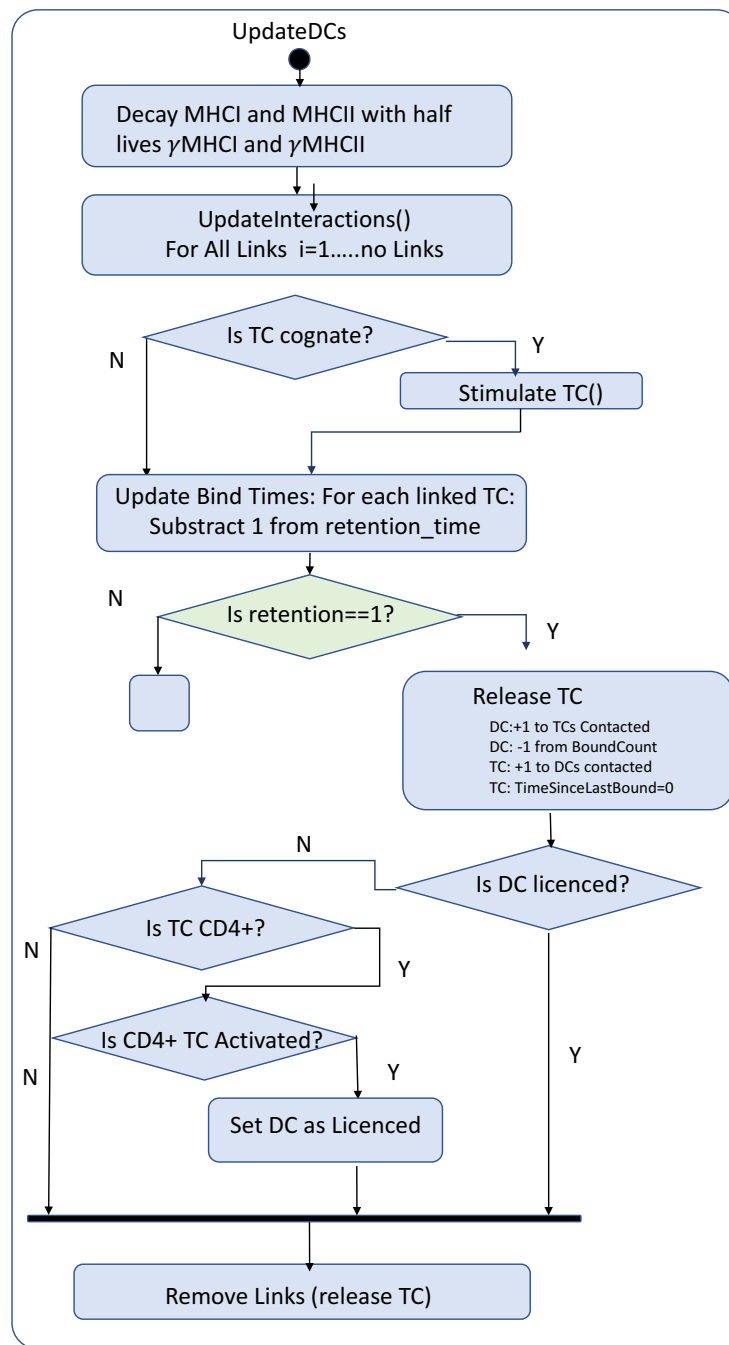


Figure 8: UML diagram for 'UpdateDCs'. DCs express a decaying MHC signal, accessible by TCs that they have formed links with. Each interaction (link) is designated a duration to exist (retention), which is reduced every timestep. When retention falls to 1, the link is removed. DCs enter as mature migrating DCs but may become licenced through interaction with activated CD4+ TCs. This means the DC is more effective at activating CD8+ T cells (implemented by switching activation probability curve parameters to equal those used when interacting with CD4+ TCs).

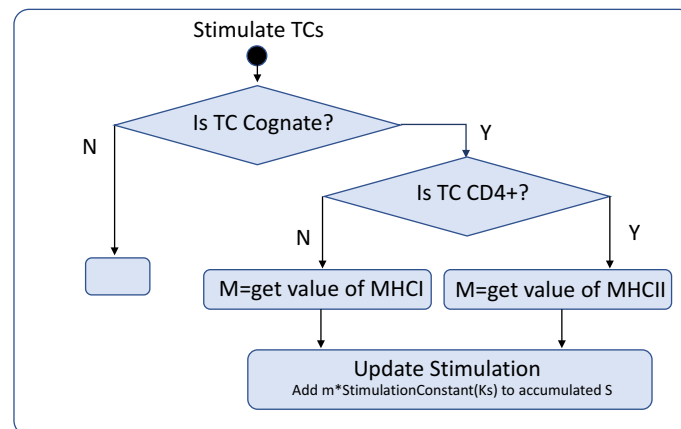


Figure 9: UML diagram for 'StimulateTCs'. DCs can access agent properties that they link with (interact with) and alter the agent's accumulated stimulation based on the DCs MHCI/II presentation and Stimulation constant, Ks.

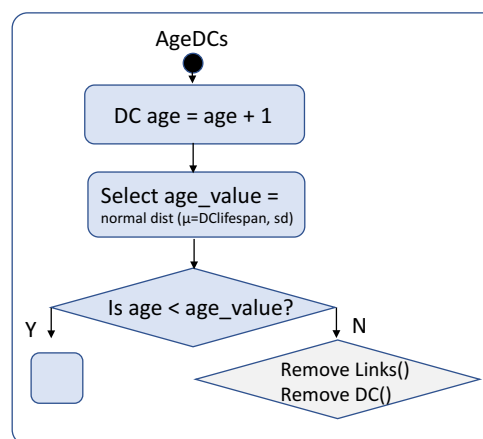


Figure 10: UML diagram for 'AgeDCs'. Mature DCs have a lifespan of around 60hours (+/-2.7hrs), drawn from a normal distribution. When reached, any interacting agents are notified, any links are removed, and the DC is removed.

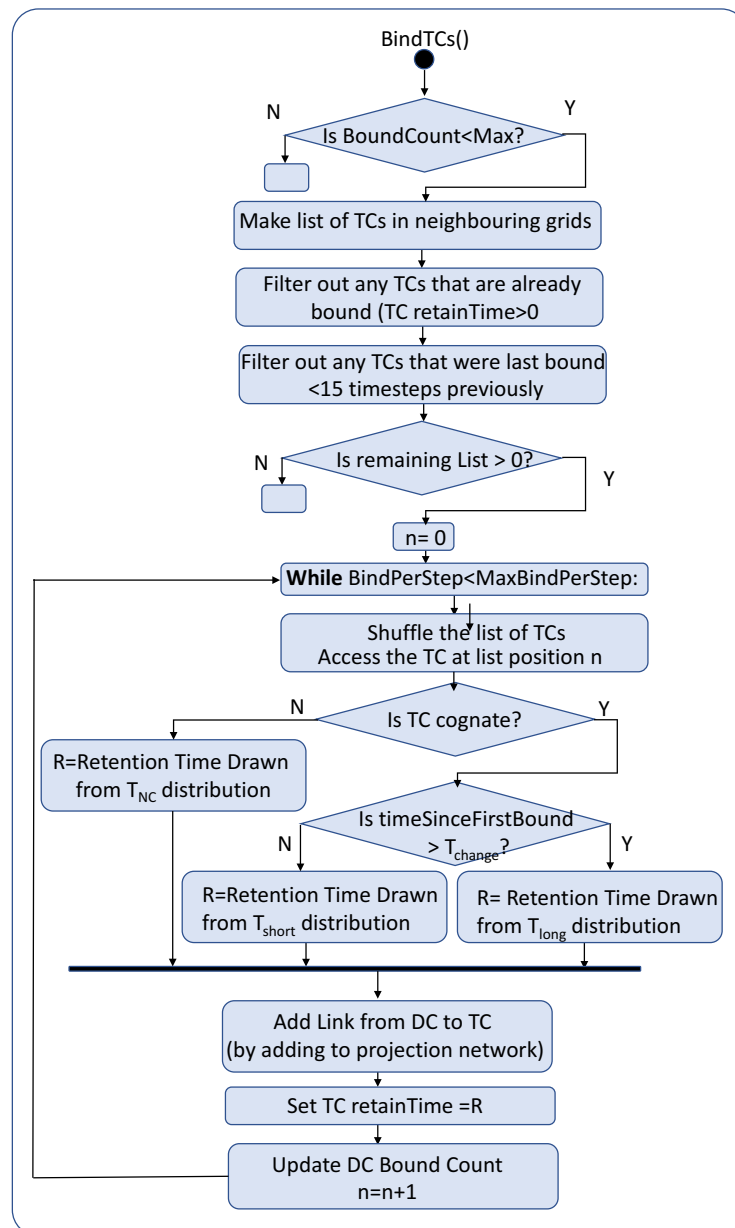


Figure 11: UML diagram for 'BindTCs'. When setting up new interactions, the DCs must first not exceed their maximum number of interactions (BoundCount). The GridCell property that can provide a list of neighbouring compartments and their agents is used again to determine TCs eligible to interact. Once selected, the TCs receive a interaction time stored as a 'retention' property drawn from a relevant probability density function, based on their type and history. This property decreases each timestep, until the value reaches 1 and the T cell is released.

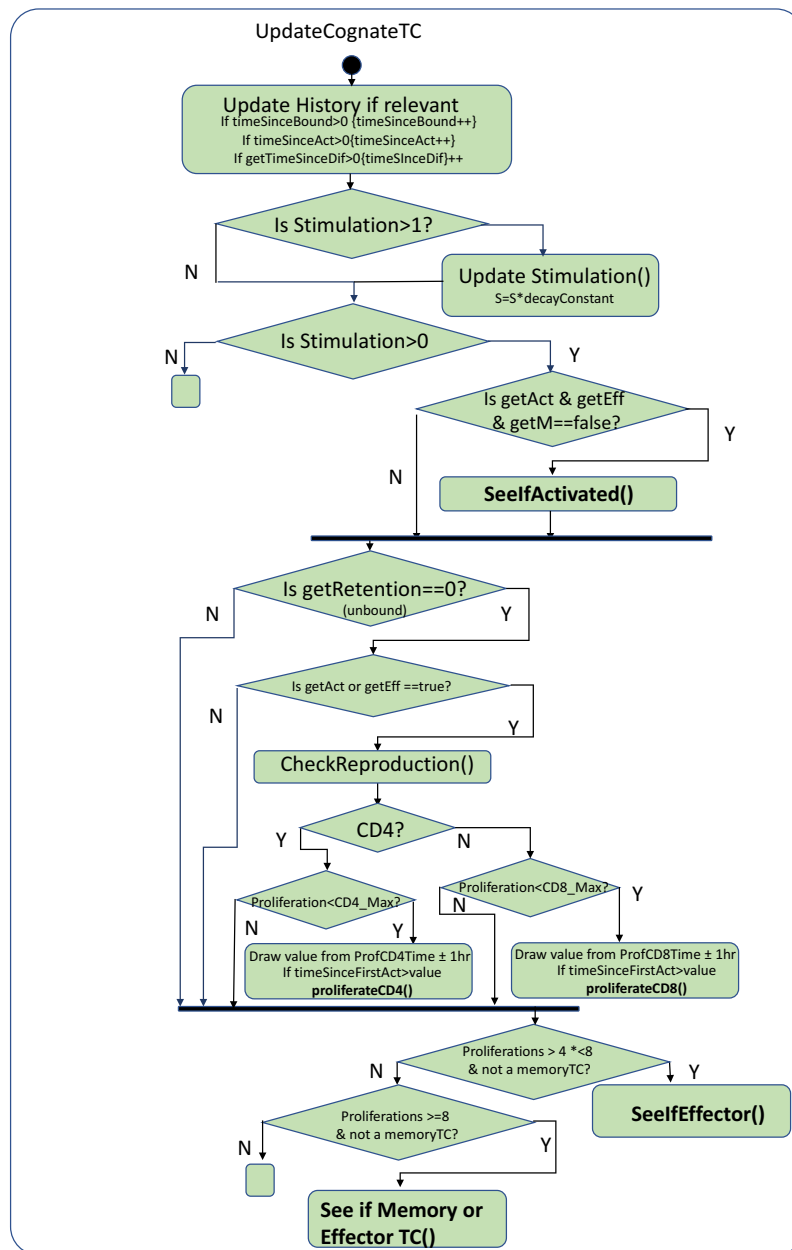


Figure 12: UML diagram for 'UpdateCognateTCs'.

Cognate TCs store properties determining their current state and history. Stimulation is increased during interaction with DCs, and decays each time step. Activation probability is a sigmoidal function of Stimulation (Figure 14). Proliferation and differentiation are permitted only when T cells are unbound. CD4+ and CD8+ T cells follow different rules, using sigmoidal functions that are parameterized differently. SeeIfEffector or Memory use accumulated stimulation as inputs to sigmoidal functions to determine if the cell will differentiate, then a probability ratio to determine which cell type will be produced.

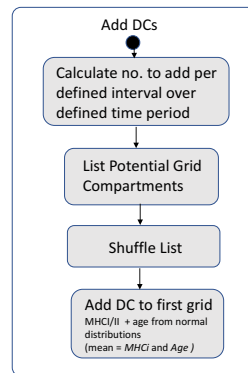


Figure 13: UML diagram for 'AddDCs'.

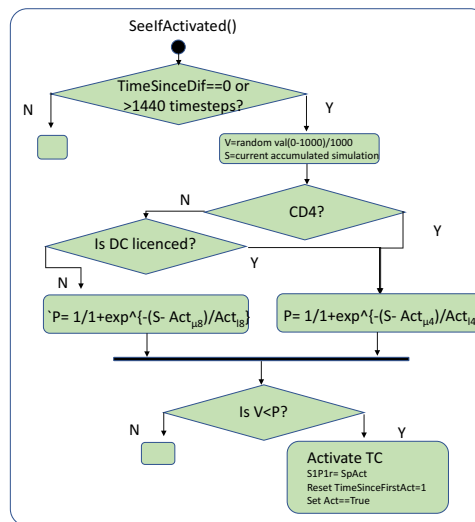


Figure 14: UML diagram for 'SelfActivated'. Stimulation is used as an input to sigmoidal functions that produce a probability of activation. If generation of a random decimal number between 0 and 1 is less than the probability of activation, then the T cell is activated and S1P1r expression is down-regulated.

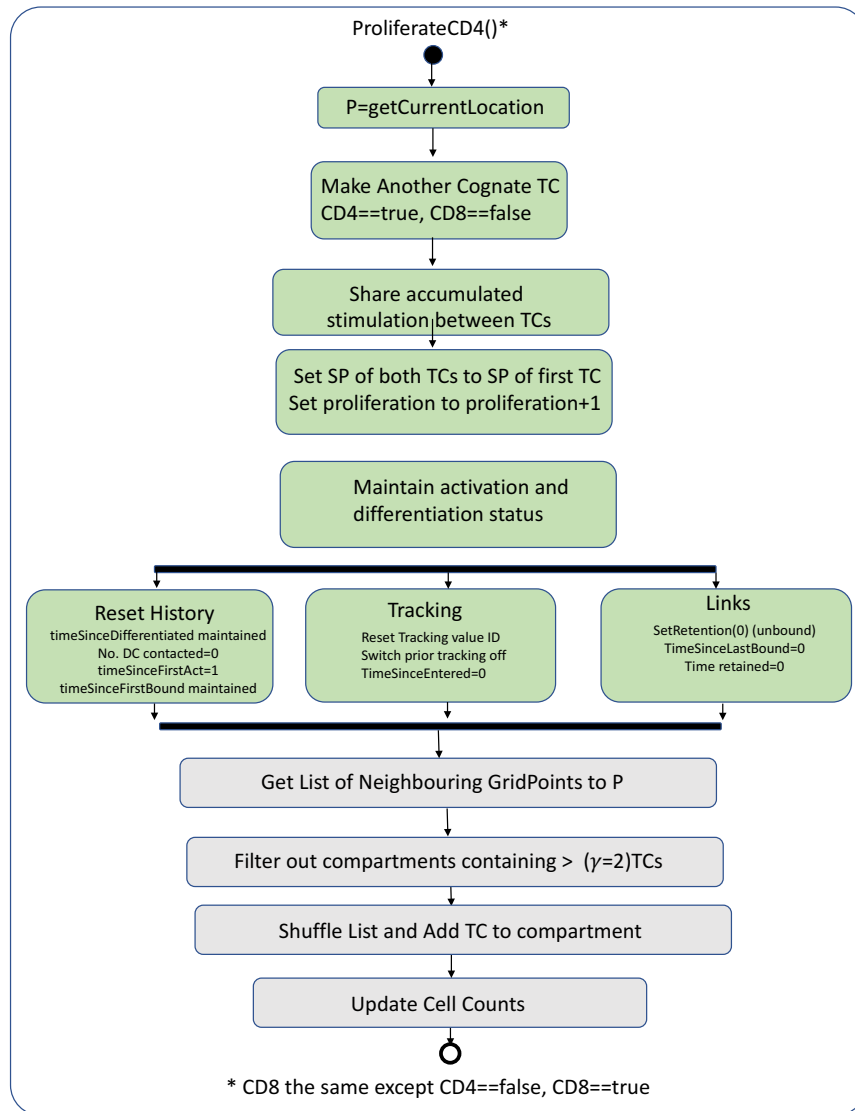


Figure 15: UML diagram for 'Proliferation'. Proliferation of T cells is complex because accumulated Stimulation is shared between the daughter cells (which is in-fact a reset existing cell and a copy). History, tracking value and links are updated to ensure that the two new cells have individual properties

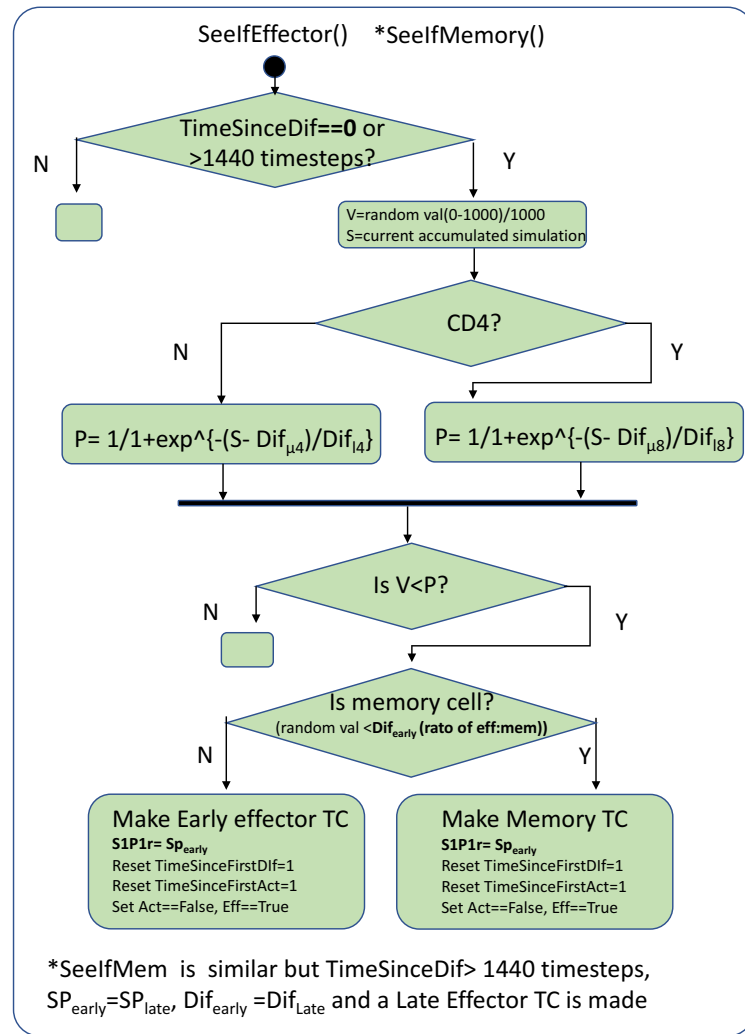


Figure 16: UML diagram for 'SeeIfEffector' or 'SeeIfEffectorMemory'. The two behaviours are similar but for seeIfEffectorMemory, the time since differentiation cannot be zero, as the TC must have previously differentiated, and more memory TCs are made.

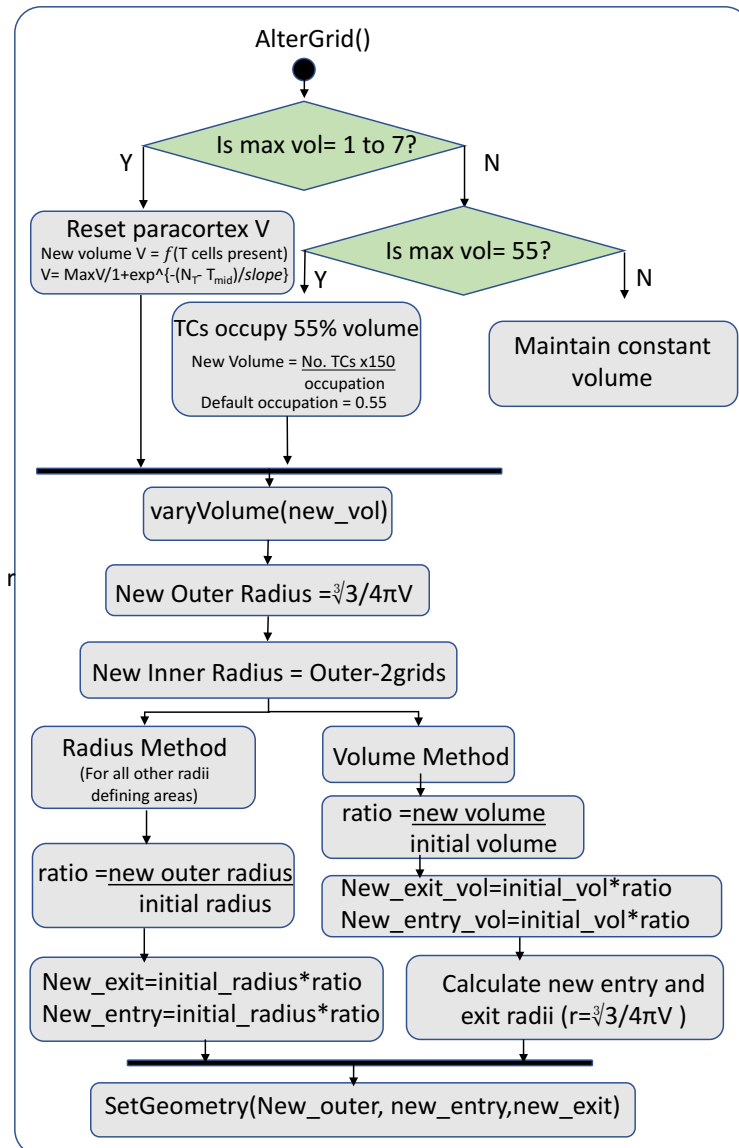


Figure 17: UML diagram for 'AlterGrid'. This methods carries out the calculation of the new paracortex volume then passes the calculated radii to 'Set Geometry'.

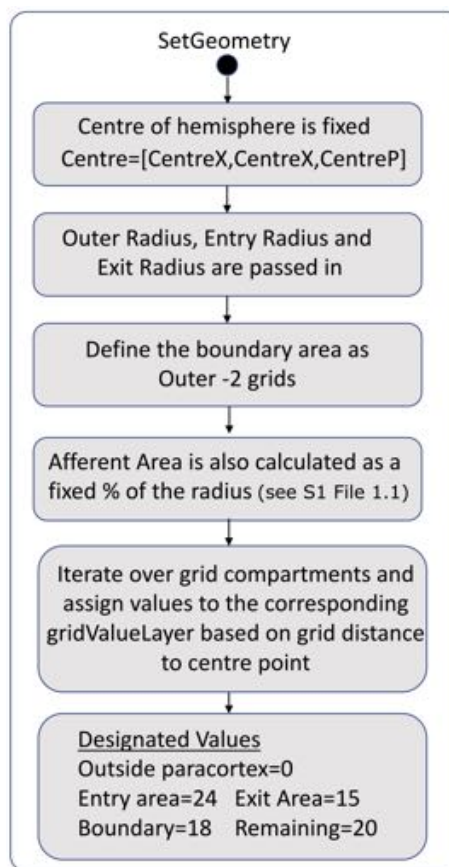


Figure 18: UML diagram for 'SetGeometry'. Only the radii are required as inputs to determine the geometry of the LN as described in S1 File. 1.1