

2.2 UML diagrams

The following section presents the logic underlying the sub-methods called during the simulation. The order that each method is called is depicted in the 'Main' diagram (Figure 2B).

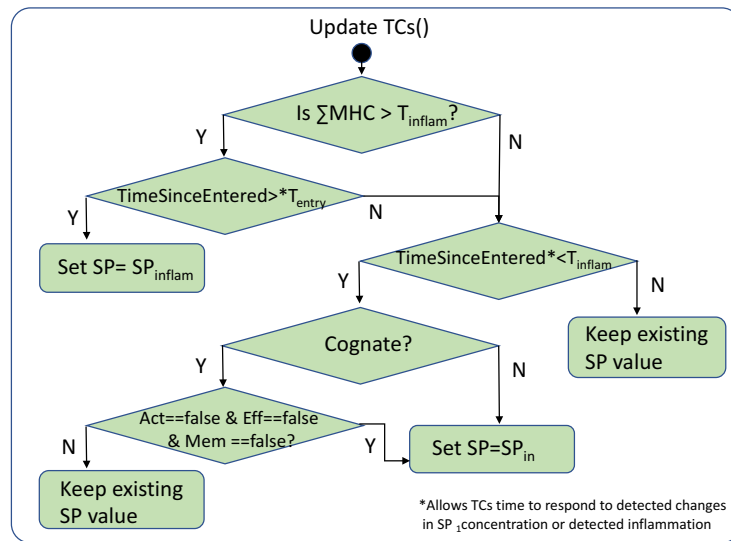


Figure S9. UML diagram for 'UpdateTCs'. By checking the time since the TC entered, a 'lag time' representing time for detection of signals and receptor internalisation to occur.

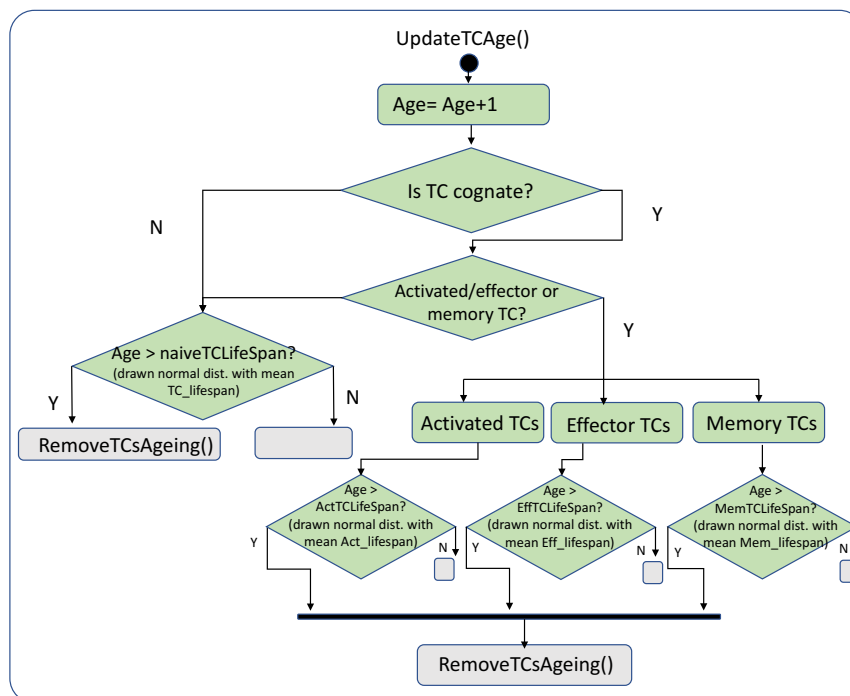


Figure S10. UML diagram for 'UpdateTCsAgeing'.

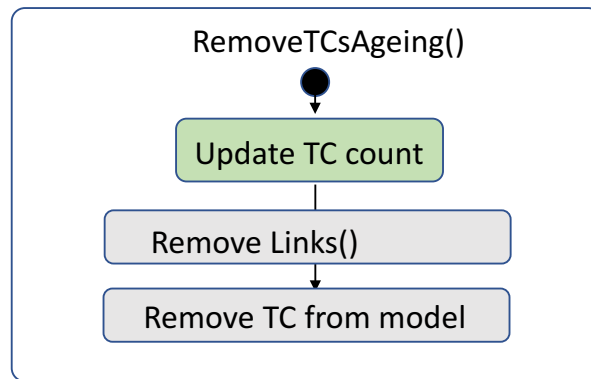


Figure S11. UML diagram for 'RemoveTCsAgeing'. The different types of TCs draw ages from gaussian distributions with a different mean value.

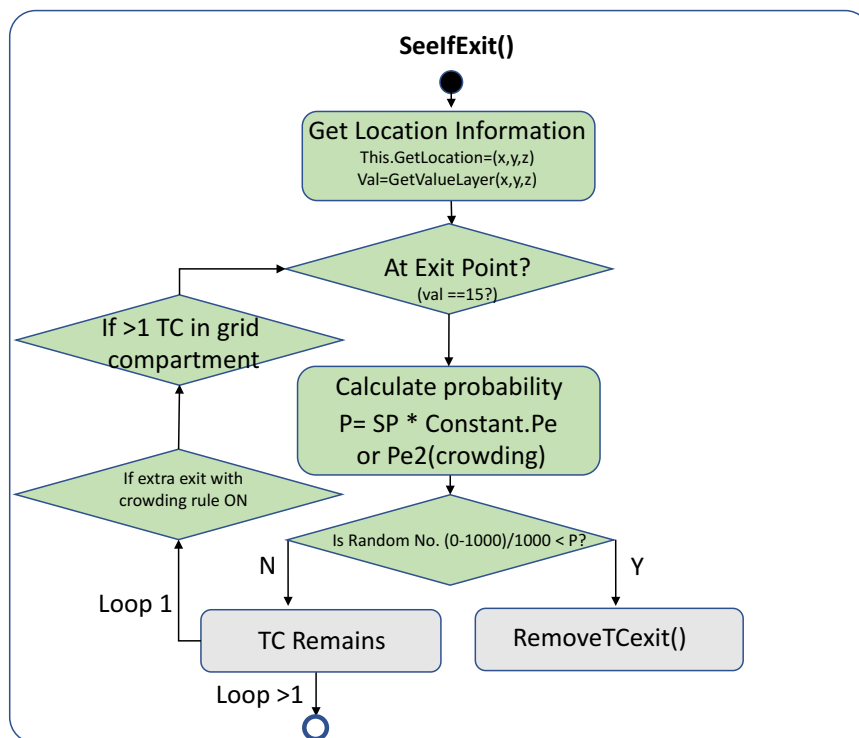


Figure S12. UML diagram for 'SeelfExit'. The original simulations did not incorporate the second check of TC crowding (crowding rule).

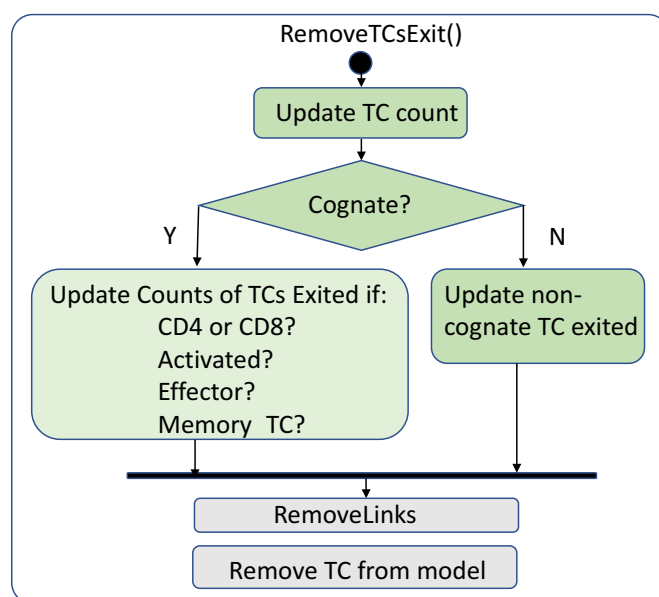


Figure S13. UML diagram for 'RemoveTCExit'. Ageing and TC exit are treated differently due to different counters being updated

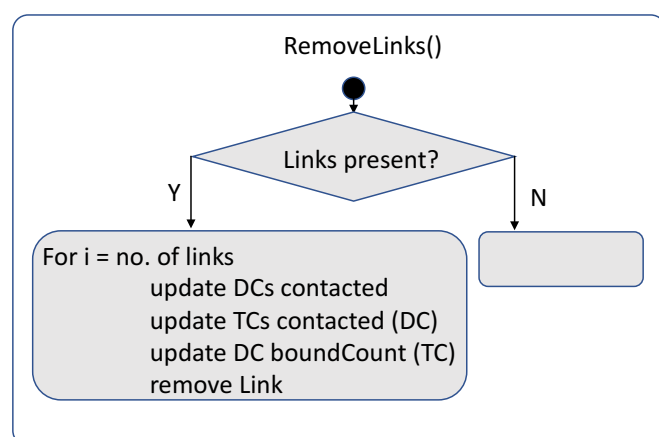


Figure S14. UML diagram for 'RemoveLinks'. Links between TCs and DCs are formed using a projection network, a feature of RepastSimphony, and must be removed when the connection is over.

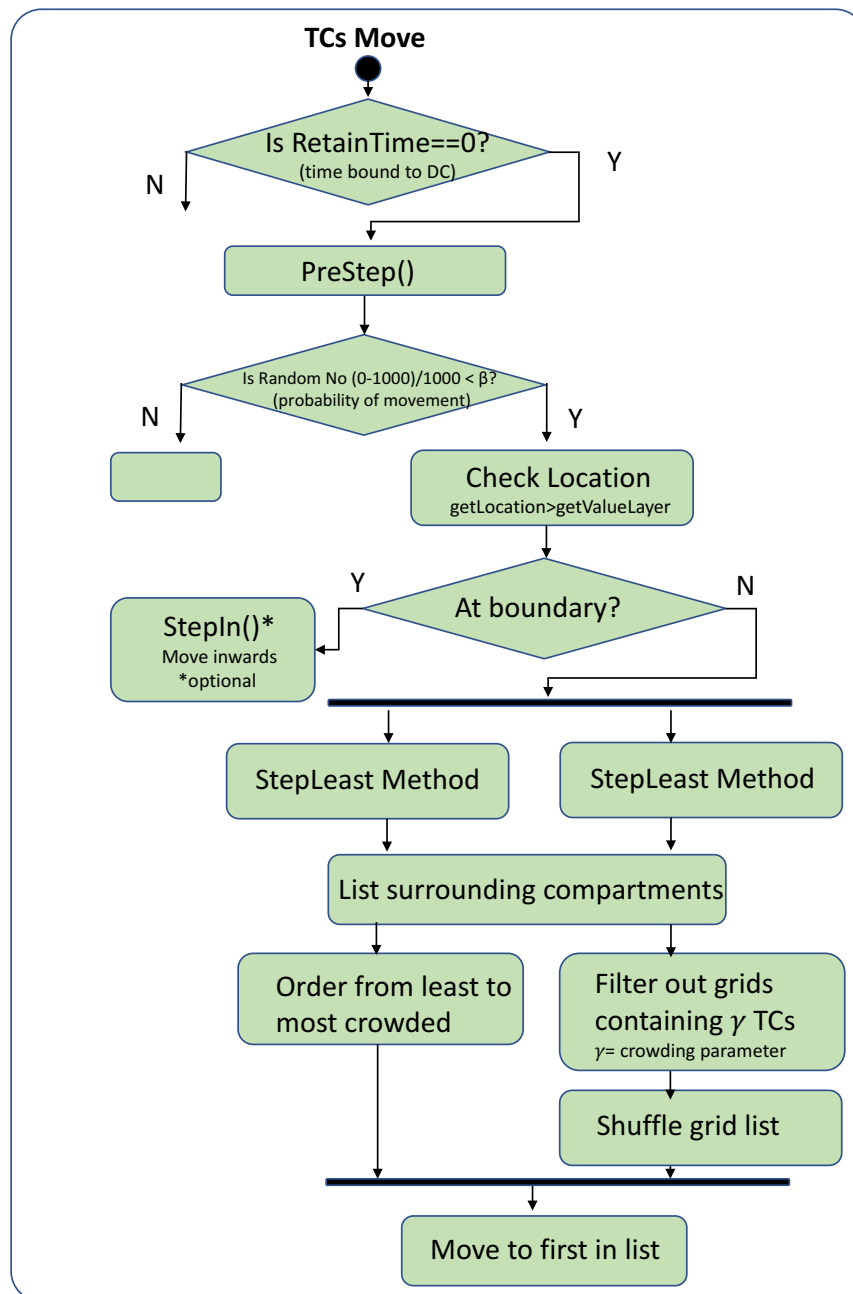


Figure S15. UML diagram for 'TCsMove'. The model has the capacity for different movement options, and maximum TCs per grid is set with crowding parameter γ

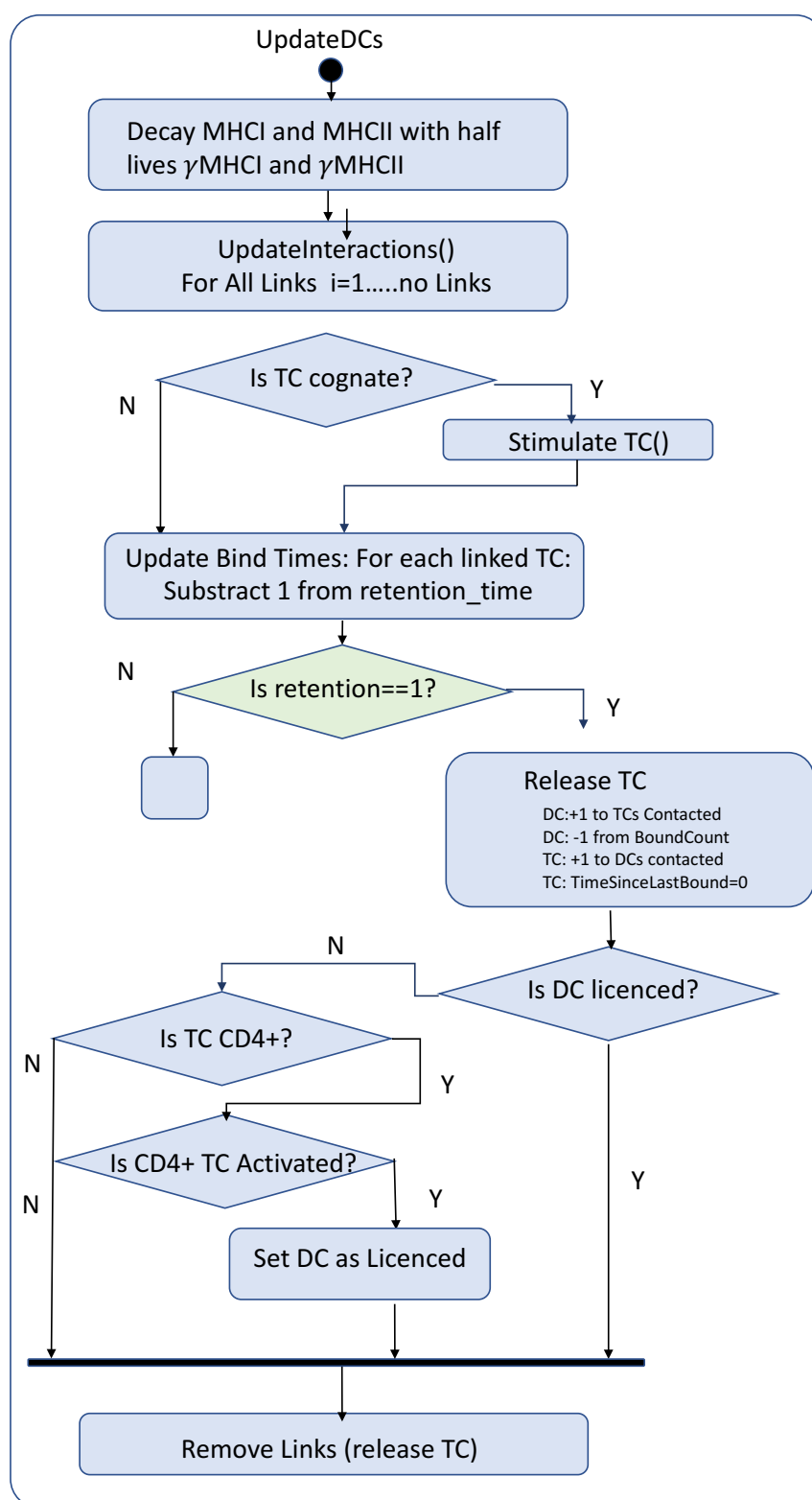


Figure S16. UML diagram for 'UpdateDCs'. DCs enter as mature migrating DCs but may become licenced through interaction with activated CD4+ TCs

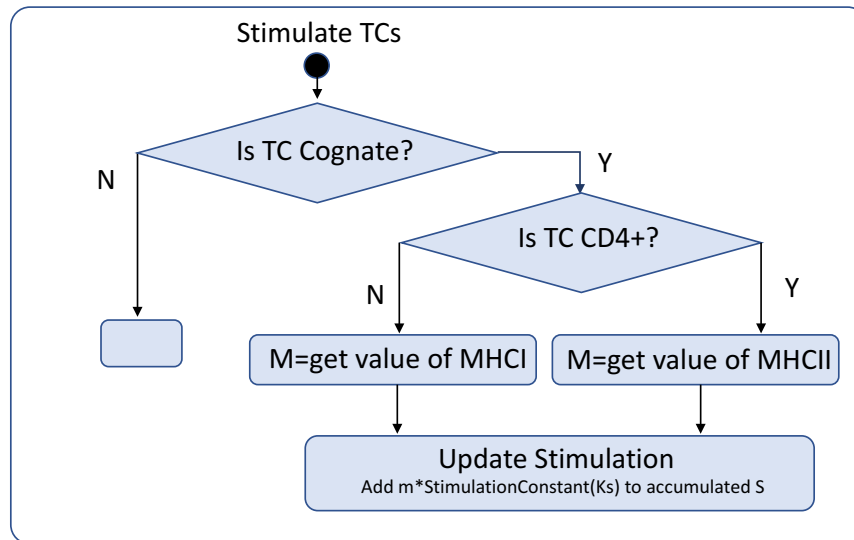


Figure S17. UML diagram for 'StimulateTCs'. DCs can access the variables of the agent attached via a projection link and thus alter their state based on the DCs MHCI/II presentation and the Stimulation constant, Ks.

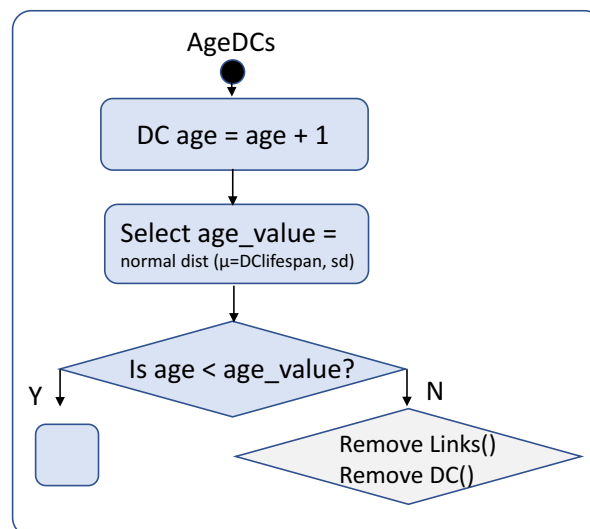


Figure S18. UML diagram for 'AgeDCs'. Mature DCs have a lifespan of around 2.5 days.

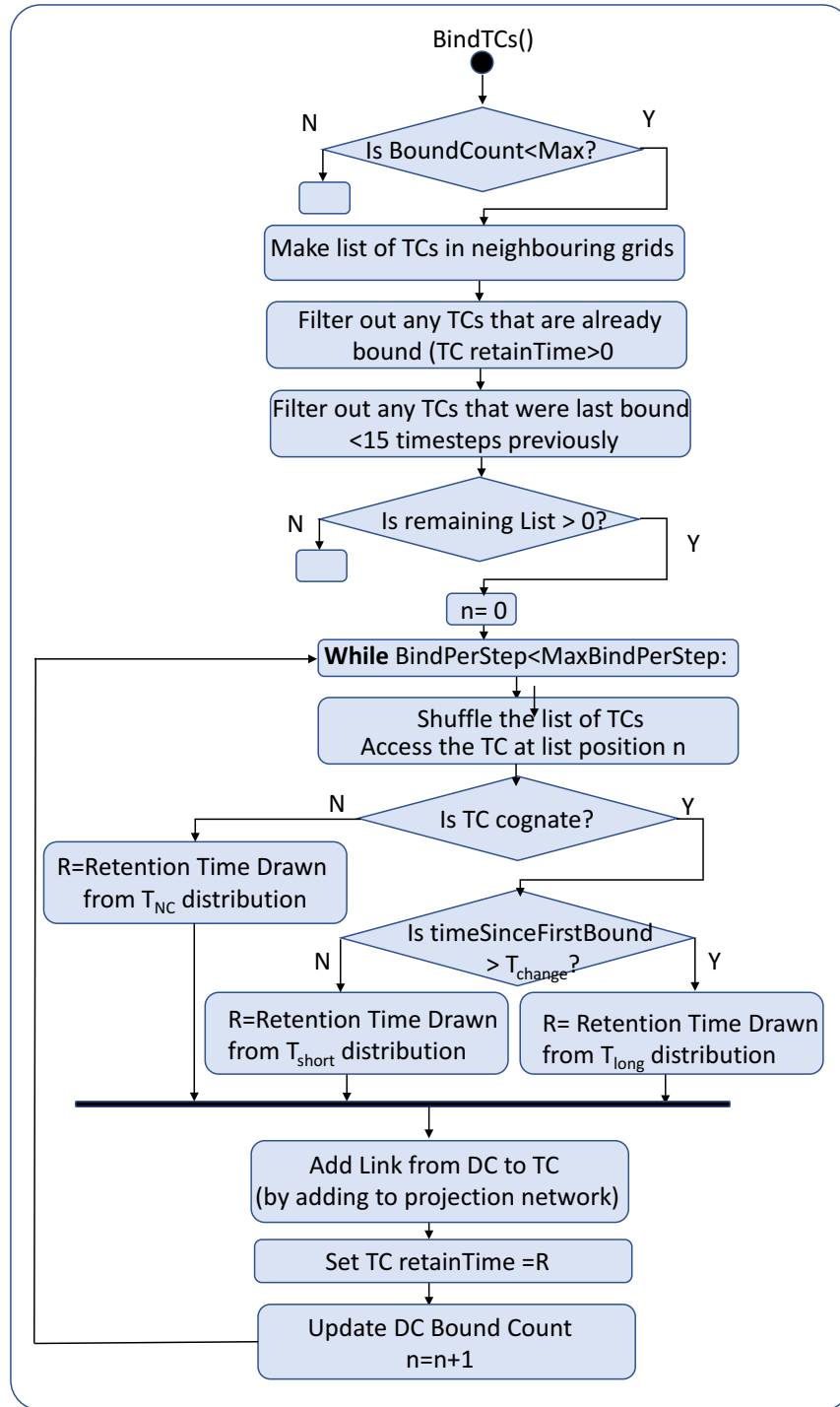


Figure S19. UML diagram for 'BindTCs'. TCs receive a 'interaction time' drawn from a relevant probability density function, based on their type and history.

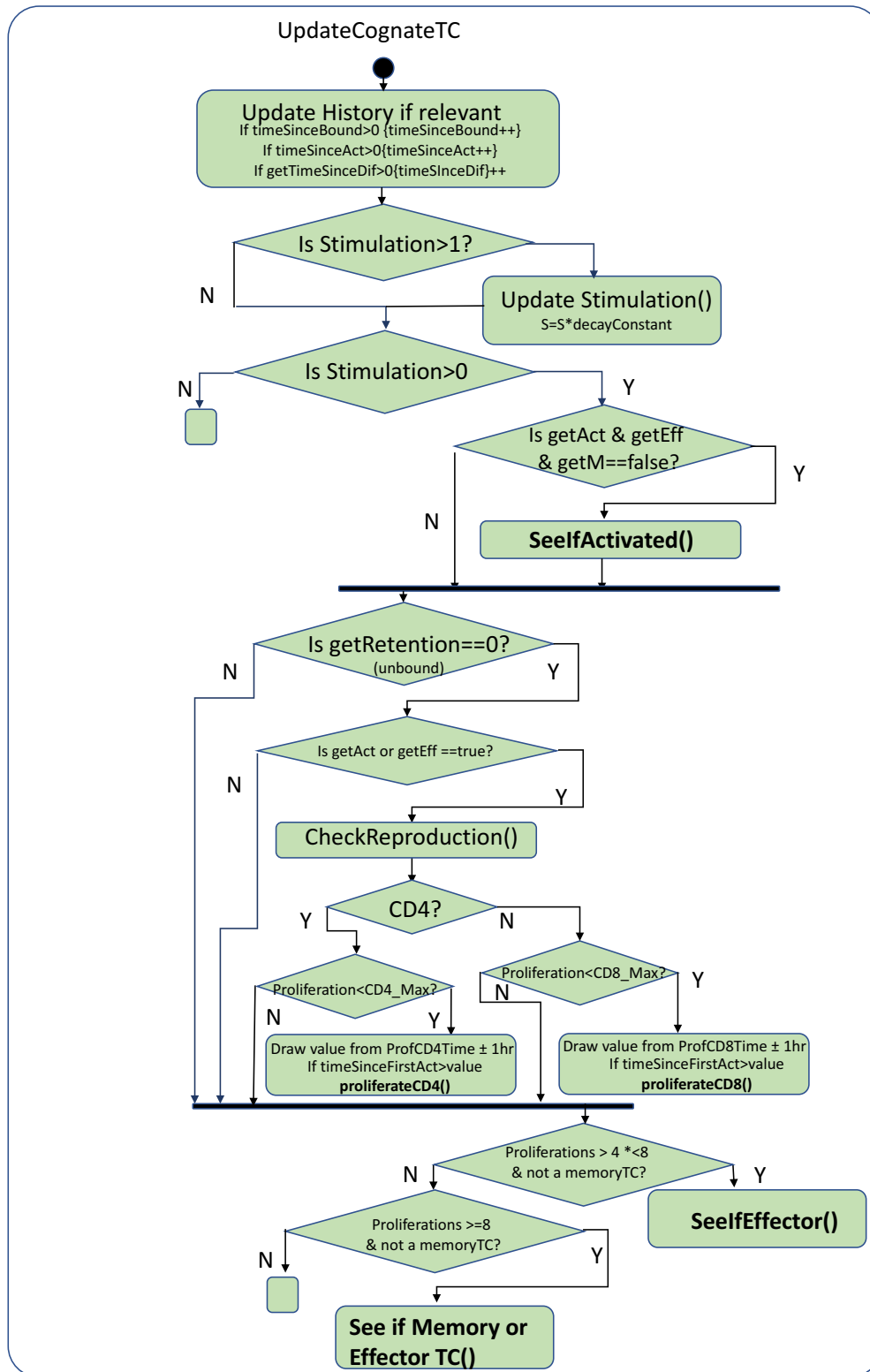


Figure S20. UML diagram for 'UpdateCognateTCs'.

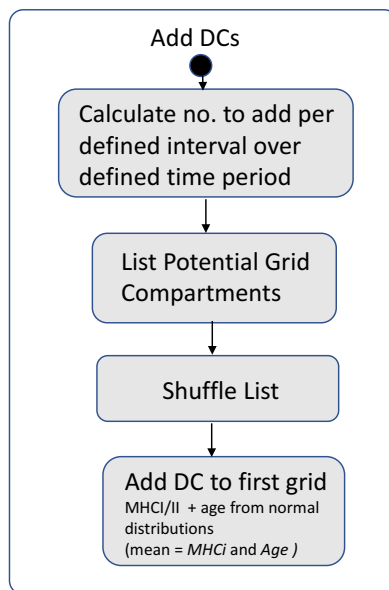


Figure S21. UML diagram for 'AddDCs'.

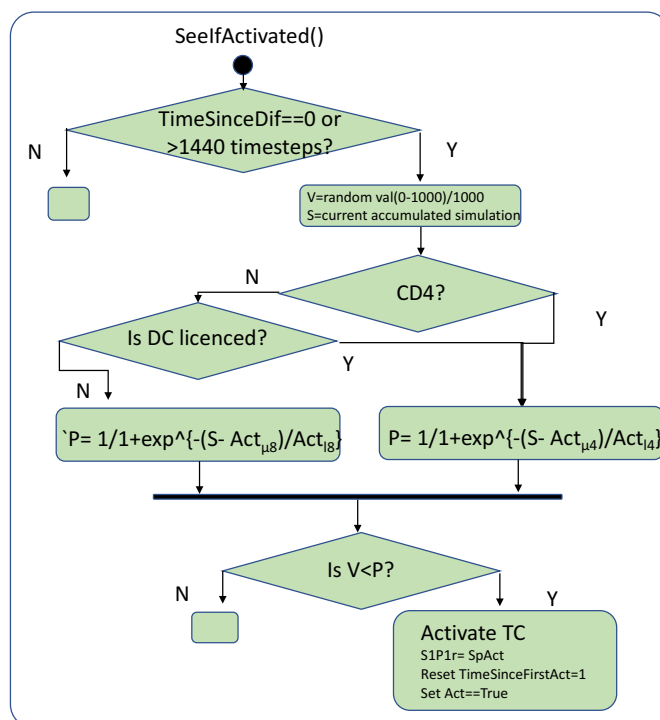


Figure S22. UML diagram for 'SeeIfActivated'.

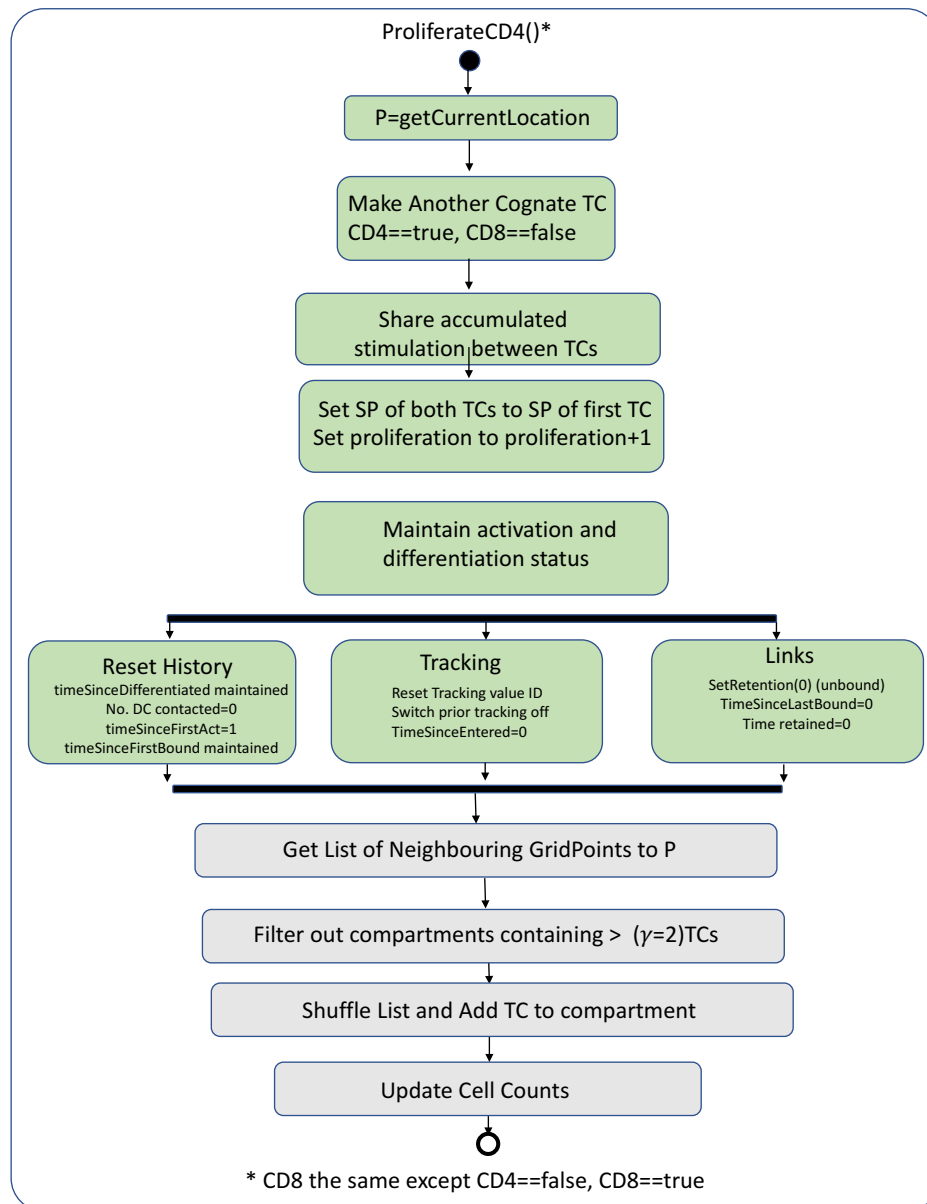


Figure S23. UML diagram for 'Proliferation'.

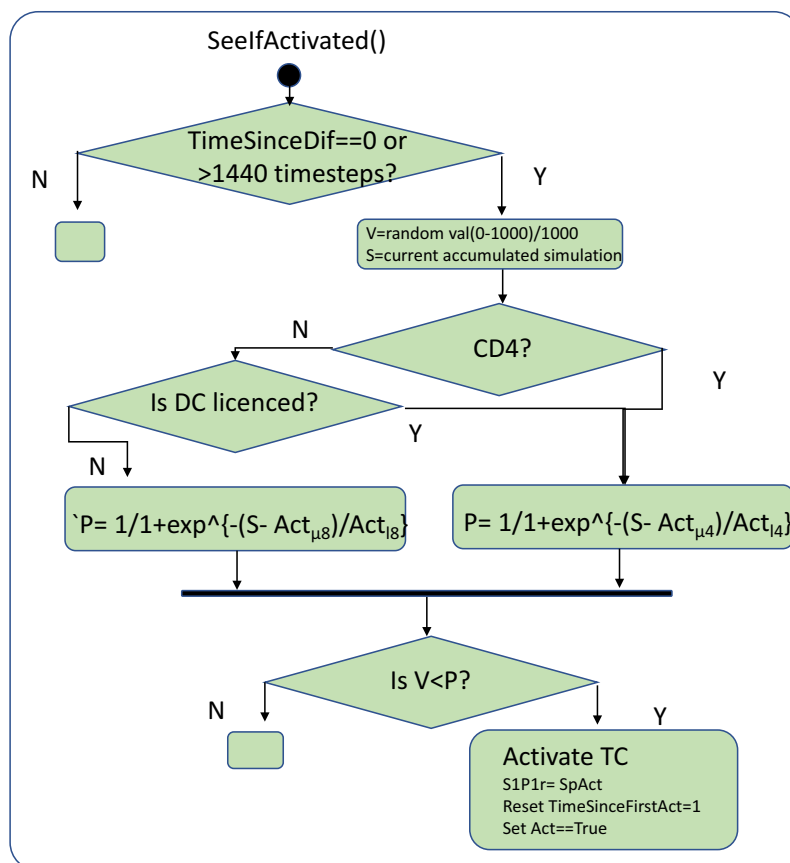


Figure S24. UML diagram for 'SeeIfEffector' or 'SeeIfEffectorMemory'. The two behaviours are similar but for seeIfEffectorMemory, the time since differentiation cannot be zero, as the TC must have previously differentiated, and more memory TCs are made.

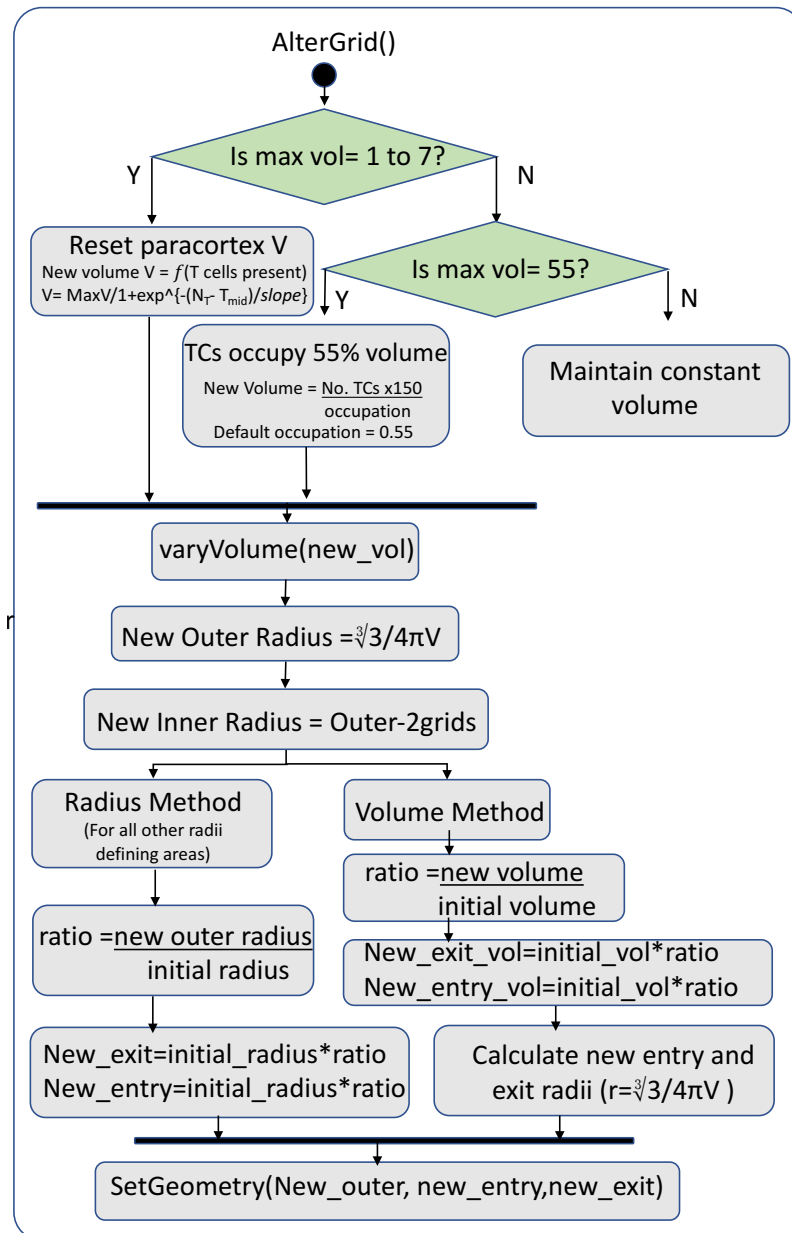


Figure S25. UML diagram for 'AlterGrid'. This methods carries out the calculation of the new paracortex volume then passes the calculated radii to 'Set Geometry'.

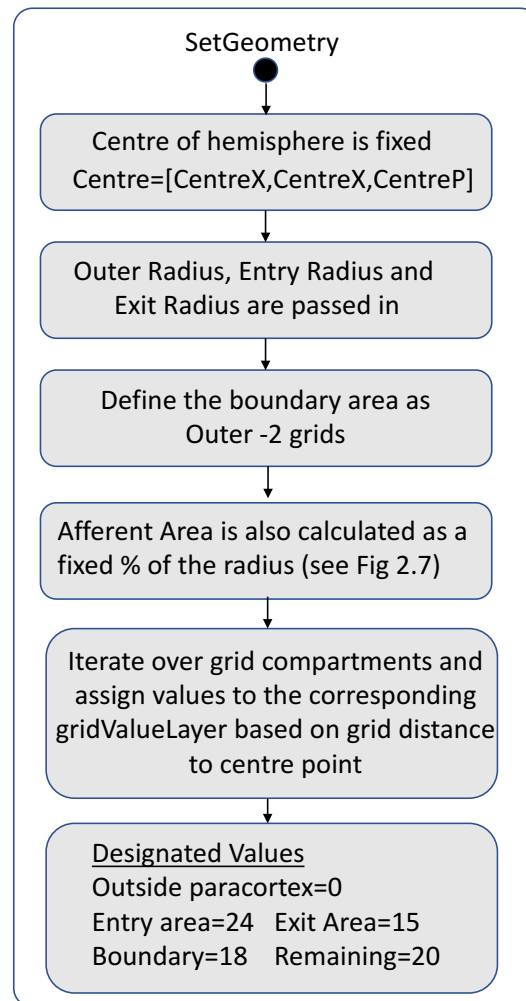


Figure S26. UML diagram for 'SetGeometry'. Only the radii are required as inputs to determine the geometry of the LN