M2. Recruitment scheme module

M2.1 Slot, options and selection

Summary. The recruitment slot decides how immune cells enter the grid at vascular sources, once per cellular timestep. Option A, selected by default, recruits probabilistically from the local soluble signals at each source (S8). Option B draws recruits from finite lymph-node and blood compartment populations advanced by an ODE model, so that the onset, magnitude and cognate fraction of the T cell response are outputs.

What the slot controls

M2 specifies the slot, not either option's algorithm. Its subjects are the interface an option must satisfy, the implementation the base class gives every option, how one option is selected and how long the resulting object lives, where the recruitment state resides, and the properties that change with the selection. S8 specifies option A. M2.9 specifies option B as far as the slot needs; its equations are outside this documentation set.

Options

Option	Parameter groups	Specified in
A, probabilistic recruitment from local signals	`Recruit_Prob` and its five subgroups	S8; M2.8
B, lymph-node and blood compartment recruitment	`Recruit_ODE`, with others listed in M2.13	M2.9; equations outside this set

There is no third option. Adding one means deriving a further option class and extending the selector; M2.2 lists what such an option must satisfy.

How an option is selected

The selection is a command-line option, not a parameter-file entry. This follows the division of spine 7.0, "Command-line options and the parameter file": settings that choose among model variants are command-line options, and quantities entering model equations are parameter-file entries.

A value naming neither option reaches the selector's default branch, which throws with the numeric value in its message. The end-of-list sentinel is therefore never accepted, and there is no fall-through to a default option. The rejection happens at selection, not at command-line validation, so the diagnostic a user sees for a mistyped value depends on when the driver maps the option string to an option.

The slot has no default; the driver supplies option A. The simulation's constructor leaves the recruitment object unset and makes no selection. A newly constructed simulation therefore cannot execute a timestep: the invocation (M2.7) dereferences the unset object. The driver selects before the first timestep, option A when the command-line option is omitted (M2.13); the slot itself supplies no default, no fallback and no diagnostic. The diffusion slot, by contrast, installs a placeholder solver at construction so that one exists before a saved state is loaded (M1.1). The recruitment slot needs none, because it saves the object itself (M2.5). The default is held by the driver, not by the slot.

M2.2 The slot interface

The slot is an abstract base class; an option is a concrete class derived from it.

The required members

The base class declares three pure virtual members:

- **the method query**, which returns the option's identity;
- **the recruit call**, taking the simulation, the current timestep and a save-at-last-timestep flag, which performs the whole of one timestep's recruitment;
- **the clone call**, which returns a heap-allocated deep copy through a base pointer.

The method query is the identity test by which the selector decides whether to keep an existing object (M2.4). An option returning the wrong identity would be replaced on every call to the selector.

The base class also declares a virtual destructor, a default constructor, the non-virtual rate computation, ten static predicate functions (two overloads for each of five cell types) and a serialisation member with an empty body. M2.3 specifies the predicates and the rate computation.

What the interface lacks

- **No initialisation hook.** Setup must happen in the constructor, which takes no argument, or on the first recruit call, which the option must detect from the timestep.
- **No separate per-timestep update.** Dynamics that advance whether or not any cell is recruited must advance inside the recruit call.
- **No validation hook.** Cross-parameter checks are made by the parameter system, and none involves a recruitment parameter (S8.13).
- **No accessor to an option's internal state.** That state is visible only through what the option writes to the simulation.
- **No reverse channel.** The recruit call returns nothing, so an option cannot report a count, a failure or an error short of throwing.

Option A also declares a public per-slot recruit routine (S8.11). It is an overload of the recruit name, not an override, and is not part of the slot contract: an option need not decompose its work by source or by slot.

Reads and writes

The simulation is passed by non-const reference, and no restricted interface is interposed. An option may therefore read and write anything reachable from the simulation. The slot constrains what an option is called with and when, not what it may do, so any claim about what recruitment touches is a claim about one option.

- **Reads:** the simulation; the current timestep, by value; the save-at-last-timestep flag, which neither option reads (M2.7).
- **Writes:** agents, through the simulation's factory functions; the per-compartment recruitment counters; the recruitment statistics, including the 18 rate statistics of M2.3; the global random number stream.

S8.1 lists option A's reads and writes.

Contract for a further option

A further recruitment option must:

- derive from the base class and implement the three pure virtual members;
- return a distinct identity from the method query, added to the option enumeration before its sentinel, with an explicit value and a case in its output operator;
- be constructed in a new branch of the selector, with a default constructor;
- carry the serialisation registrations (M2.5) if a saved state is to be reloadable, since the object is saved through the base pointer;
- provide a clone that deep-copies whatever state it holds; a plain copy suffices only for an option with no state;
- call the rate computation once per timestep if the rate statistics are to be meaningful, no hook guaranteeing it (M2.3);
- tolerate the source permutation, which the caller performs and the option cannot decline (M2.7);
- create agents through the simulation's factory functions if occupancy enforcement is to apply (M2.12).

The compiler enforces none of the distinct identity, the serialisation registrations and the rate-computation call.

M2.3 Shared machinery in the base class

The eligibility predicates and the rate computation live in the base class, and both options use both.

The eligibility predicates

Five predicates, one for each recruitable type (macrophage, Tgam, Tcyt, Treg and neutrophil), are static members of the base class, defined in the slot header. Each has a form that returns a recruitment weight and a form that discards it. Because they are static and take the grid rather than the simulation, the local-signal eligibility test is shared machinery of the slot, not a property of option A. S8.4, S8.6 and S8.8 specify the macrophage and T cell predicates; M9.5 specifies the neutrophil predicate.

The macrophage and T cell predicates are pure: they read the grid, the position and parameters, and write only their output argument. The neutrophil predicate also increments a statistic and raises a fatal error on two invariant violations (M9.5). A caller inherits both effects.

- **Option A** uses the weight-returning form at all five of its call sites.
- **Option B** calls the weight-discarding form for macrophages, Tgam, Tcyt and Treg. Its eligibility is the boolean threshold test alone; the weight is computed inside the predicate and discarded.
- **The neutrophil weight-discarding form** is called by neither option. Calls from the visualisation layer, if any, do not affect dynamics.

The rate computation

The rate computation is a non-virtual member of the base class. It recomputes 18 rate statistics: the total, cognate and non-cognate rates for macrophages, neutrophils, dendritic cells, Tgam, Tcyt and Treg, each a cumulative count divided by the current timestep. S9 specifies these statistics and their run-to-date reading.

The simulation does not call the rate computation; the option must. Both options call it once per timestep, so the rate statistics have one definition across the slot; recruits to unsimulated granulomas are excluded. Option A calls it after its source loop. Nothing in the slot guarantees the call: an option omitting it would leave all 18 at their last values, and one calling it twice would recompute the same values. Not every recruitment statistic shares one definition across the options (M2.11).

M2.4 Selection, construction and ownership

The selector

The recruitment object is selected by one call to the selector; nothing else constructs one. If an object exists and its method query matches the request, the selector returns and keeps it. Otherwise it deletes any existing object and default-constructs the requested option. The reason for keeping a match is that the object may have been loaded from a saved state and must not be replaced.

No configuration is passed at construction; both options read the parameter singleton. The rationale is that the selector does no option-specific processing, the caller doing whatever an option needs, and that the option is not changed after program initialisation. The program enforces neither. Option-specific initialisation is therefore the driver's responsibility and is invisible at the slot. For option A it is only the onset draw (M2.6), made through the simulation's initialisation.

Change of option during a run

The selector is public and keeps an object only on a match, so the option can be changed mid-run. The change deletes the outgoing object, with any state it held, and default-constructs the incoming one. Option A holds no state, so for it the change discards nothing. For option B it would discard the compartment populations and ODE coefficients and restart them from their initial conditions. No code path in the simulation makes such a change, and the selector does not prevent one.

Ownership

The selector allocates the object on the heap; the simulation holds it and deletes it in its destructor, so ownership is unique. The object holds no pointer to anything: the simulation is passed afresh on each invocation.

The object outlives repeated modification of what it reads, and the grid's buffers are exchanged on every diffusion step (spine 2.2), so a reference cached at construction would be invalidated. Neither option caches a reference to the simulation; each re-obtains the grid and the statistics object on every invocation.

Cloning

Cloning the simulation copies the object pointer, then replaces a set pointer with a clone of the object. A simulation cloned before selection yields an unselected clone, not a fault. Option A's clone is a plain copy, which is a deep copy because the option has no data members. An option with state needs deep-copying members or an explicit clone; the slot imposes the obligation and provides no mechanism for it.

M2.5 Saved state and restart

The simulation saves the recruitment object itself, through the base pointer, so each option must be registered for reconstruction of its derived type. Option B is registered at every level of its class hierarchy.

The diffusion slot instead saves its method and re-selects on load (M1.1). The two slots therefore restore differently.

What is saved

- **Option A** saves only the base object, whose serialisation is empty, so its saved representation is its type identity and no data. Saving the object or the method restores the same thing.
- **Option B** holds its compartment populations and ODE coefficients on the object, and they are saved with it. On reload the saved coefficients are kept, so changes to the `Recruit_ODE` entries of the parameter file have no effect on a resumed run.

Precedence against a command-line selection

Precedence between a command-line option and a loaded state is option-specific and is indexed in Appendix B (B.11). For `--recr` it is not and cannot be implemented by the slot, which knows nothing of the command line. Which selection wins depends on the order in which the driver selects and loads, and the selector's keep-on-match rule (M2.4) makes the orders behave differently.

Driver order	Outcome
Select, then load	The loaded option wins: the loaded object replaces the one the selector made
Load, then select the same option	The loaded object is kept
Load, then select a different option	A fresh object replaces the loaded one: the command line wins and the loaded state is lost

For option A the outcomes are indistinguishable. For option B, loading and then selecting a different option discards the restored populations and coefficients, and no warning is issued.

M2.6 Where the recruitment state lives

The option objects

Option A has no data members. It retains nothing between invocations: it reads everything from the simulation or the parameter singleton and writes everything to the simulation. It could be a free function taking the simulation, losing only the ability to select among options. Option B holds its compartment populations and ODE coefficients on its object (M2.5).

The onset schedule

Option A's mechanism (S8) has persistent state: the drawn onset timestep and the latched onset flag, both of which must survive a restart. Neither lives on the option object. Both are simulation members, written by simulation routines that read option A's parameters.

Item	Resides on	Written by	Specified in
Drawn onset timestep	Simulation	The onset draw, once, at the start of initialisation	S8.10
Latched onset flag	Simulation	The onset latch, once per timestep, after the statistics block	S8.10
Per-slot eligibility, weights and probabilities	Nowhere; local to one invocation	Option A	S8.4 to S8.9
Per-compartment recruitment counters	Grid	The option's construction routines	S9
Recruitment statistics	Statistics object	The option and the rate computation	S9; M2.3

Option A reads the onset timestep and flag through accessors; neither it nor anything it calls writes them. The two writing routines read `Recruit_Prob_timeRecEnabled`, `Recruit_Prob_timeRecEnabledRange` and `Recruit_Prob_mtbRecEnabled` directly, so they implement part of option A's mechanism while residing on the simulation. The intracellular growth modulator of S4.7 also reads `Recruit_Prob_timeRecEnabled` directly, whatever option is selected. Under option B it switches at the onset drawn from option A's parameters, which option B's T cell recruitment does not read; it has no effect by default (S4.7, M8.12).

Consequences

- **The slot boundary does not coincide with the mechanism boundary.** S8 therefore specifies an onset schedule that the option object does not implement.
- **The schedule is saved with the simulation**, independently of the recruitment object. A restart that changes the option keeps the onset timestep and flag of the discarded option.
- **The schedule runs under every configuration.** Both members, both writing routines and the three parameter reads exist under option B too. Option B reads neither the onset timestep nor the flag, so there the schedule is computed and unused.
- **Moving the schedule into the option would change the timestep.** It would move the latch from after the statistics block to before the next-state pass, removing the one-timestep lag of S8.10 and changing when the bacterial criterion reads its input. A reimplementation can move the ownership or keep the schedule's timing, not both.

M2.7 Invocation

The slot is invoked exactly once per cellular timestep, from the simulation's timestep routine, immediately after the source permutation. The block follows all movement and precedes the per-agent statistics reset and the next-state pass (spine 3.2, S8.1). No check is made that an option was selected: an unselected simulation faults here.

The source permutation

The simulation, not the option, permutes the source list before invoking the slot. The permutation is uniform and consumes draws whatever option is selected and whether or not the option uses the order (S8.1). An option can neither decline it nor choose another order: one wanting sources visited in a different order would have to re-sort a list just shuffled.

The arguments

- **The simulation**, by non-const reference. Option A reads the grid, the statistics, the agent factories, the onset accessors and the module flags through it.
- **The current timestep**, by value. Option A passes it to the ramp and the rate computation, and reads the same value from the simulation for birth times.
- **The save-at-last-timestep flag**, by value. The driver supplies it, and its name refers to the final timestep. Neither option reads it, so it has no effect on dynamics.

M2.8 Option A, selected: probabilistic recruitment from local signals

Summary. Every vascular source is evaluated once per cellular timestep and once per open slot. Eligibility and a recruitment weight come from the soluble signals at the source, and a cascade selects at most one cell type per slot, which is constructed and placed. S8 specifies the algorithm.

Properties

The supply of cells, the imposed response and the limits on influx are compared with option B in M2.11. Under option A the innate-then-adaptive structure of the model follows from the imposed schedule, not from its dynamics (spine 4.2, 4.3).

- **Information is local, except for the floor.** Each eligibility test and weight is computed from the fields at one compartment, and on the neutrophil path from sums over its Moore neighbourhood (M9.5). The exception is the macrophage floor, which reads the whole-grid macrophage count, the only whole-grid quantity read by a recruitment or agent rule (S8.7). Option B applies the same floor (M2.9).

Some parts of the mechanism are visible only through option A but belong to the slot: the predicates and the rate computation in the base class (M2.3), and the onset schedule on the simulation (M2.6). The draws and rationale of the source permutation are in S8.2.

M2.9 Option B, not selected: lymph-node and blood compartment recruitment

Summary. Option B draws recruits from finite lymph-node and blood compartment populations advanced by an ODE model. Each recruited T cell's type and cognate status come from a draw over the blood compartments, so the onset, magnitude and cognate fraction of the T cell response are outputs. Macrophages are recruited by option A's floor rule and, above the floor, with a fixed probability.

Scope

This section specifies option B as far as the slot needs. Its equations, subset transitions and lung-to-node coupling are outside this documentation set, so a study selecting it needs a companion document to reimplement it. Neutrophil recruitment under option B is not described here.

Interface

Option B derives from the base class. It returns its own identity from the method query, implements the recruit call with the same arguments, and provides a clone. It is created by the same selector, held in the same place and invoked at the same point, with the whole simulation by non-const reference. Its saved state is described in M2.5.

Of S8.4 to S8.12, option B keeps only the floor of S8.7; the eligibility tests of S8.4, S8.6 and S8.8, applied through the shared predicates without the weight (M2.3), with the collapse of S8.5; and the T0 construction rule of S8.12. It replaces the rest. Whether it shares the source iteration of S8.2 or the slot accounting of S8.3 is not described here.

T cell recruitment

Each recruited T cell's type and cognate status come from one draw over the eligible blood compartments, weighted by blood concentration. The cognate fraction is therefore a model output, and the `Recruit_Prob_Tcell_<type>_probCognate` parameters are not read. T cell recruitment is limited by blood availability, eligibility and space.

Neither T cell nor macrophage recruitment reads the onset schedule (M2.6), so the `Recruit_Prob` onset and ramp parameters are inert for recruitment under option B. The onset and magnitude of the T cell response are produced by the compartment dynamics, and the two-phase structure of the model becomes emergent (spine 4.2, 4.3). The fraction of T cells that secrete changes without any S6 parameter changing (S6).

Every Treg recruited under option B is non-cognate. S3.13 records that no Treg then suppresses.

Macrophage recruitment

Macrophages are recruited by the same floor rule as under option A, using `Core_Mac_initNumber` (S8.7). Above the floor, a macrophage is recruited when its eligibility test (M2.3) is met and a draw falls below the fixed `Recruit_OLD_Mac_probRec`. A recruited monocyte becomes a dendritic cell instead with probability `Core_DCellLung_percentOfMacInitNumber`, default 0.

Finite supply

The supply of recruitable cells is finite and depleted by recruitment. Influx is therefore limited by the compartment populations as well as by the local signal, and a lesion producing a strong signal may fail to recruit because the supply is exhausted. Under option A, supply does not limit recruitment.

Admission

Option B tests caseation, occupancy and the signal thresholds at the source. It applies no cell-density gate (M2.12).

Statistics

Option B uses the same statistics as option A for the eligible-, active- and crowded-source counts and the per-type recruited counts. Caseated sources are excluded. Recruits to unsimulated granulomas are counted only in the extra-granuloma statistics: six statistics, written as nine output columns (S9.19). The source-active count is incremented whenever the source is not full, a definition different from option A's (S9).

The Treg total recruitment rate is identically zero under option B, and every Treg is counted as both cognate and non-cognate in the rate statistics (S9).

Only the general per-compartment recruitment counter is incremented under option B, and only for monocytes recruited above the macrophage floor. T cell recruits and floor recruits are not counted, and the per-type counters stay zero. Option A increments both kinds of counter (S8.12).

Parameters

The `Recruit_ODE` group has 71 parameters. By subject:

Subject	Content
Blood populations	Concentrations in counts per microlitre for the naive, central memory, effector memory and effector subsets of CD4 and CD8 cells, with separate cognate concentrations and cognate fractions for the two memory subsets; where both forms are supplied, the fraction form supersedes the absolute form
Regulatory T cells	One fraction applied to the effector and effector-memory Tgam populations, cognate and non-cognate, to obtain a Treg population
Second antigen specificity	Four conversion quantities, one per memory subset, and a latency timestep after which non-cognate cells are converted to cognate; a latency of zero disables the conversion
Recruitment and priming	Rate constants and half-saturations for naive recruitment and priming, and for central-memory reactivation, separately for CD4 and CD8
Proliferation and differentiation	Rate constants and half-saturations for precursor proliferation, and for differentiation of precursors to effector and to central memory, and of effector to effector memory, separately for CD4 and CD8
Death	Per-subset death rates for naive, precursor, effector and effector-memory cells of both lineages, and for the antigen-presenting cell population
Efflux	Per-subset rates at which cells leave the lymph node, for eight subsets
Capacity and coupling	A precursor carrying capacity, a weight by which CD4 precursors contribute to CD8 priming, a blood-to-lymph-node volume conversion in microlitres, a lymph node count for the host, and a cognate frequency

Subject	Content
Dendritic-cell proxy	Two scaling factors, one to the host and one for the granuloma size: `Recruit_ODE_scalingLung` and `Recruit_ODE_scalingMDC`, whose product scales the count of dendritic cells leaving the lymphatics before it is added to the lymph-node antigen-presenting cell population (M2.10)

The blood populations are initial conditions, so option B starts from a prescribed state. Option B also reads `Recruit_OLD_Mac_probRec`, `Core_Mac_initNumber`, `Core_DCellLung_percentOfMacInitNumber`, and the per-type `Recruit_Prob` threshold and half-saturation parameters through the shared predicates (M2.13).

M2.10 Entities and machinery serving option B

Dendritic cells

No dendritic cell exists in the default configuration. Option A has no dendritic-cell predicate or construction routine, and its cascade has three top-level categories: macrophage, neutrophil and T cell. No parameter file or command line can make option A recruit a dendritic cell; the absence is structural. The neutrophil path differs: it is live, with a recruitment probability of zero (M9.5).

Under option B a recruited monocyte becomes a dendritic cell with probability `Core_DCellLung_percentOfMacInitNumber` (M2.9). Under either option, dendritic cells can also be created at initialisation outside the slot: by deterministic placement, or by random placement of `Core_DCellLung_percentOfMacInitNumber` times the initial macrophage count (spine 5.4). The default parameter file has no placement entry and sets the fraction to 0, so no route is taken. The absence is structural for option A's recruitment and configured for the entity.

The empty dendritic-cell agent list, the dendritic-cell recruitment counter and the dendritic-cell rate statistics exist and are unused by default (Appendix B; S9).

The lymphatics

The simulation allocates a lymphatics object under every configuration, clones and saves it with itself, and exposes it through a public accessor. The simulation's own timestep never advances it.

Dendritic-cell behaviour adds stimulated dendritic cells to the lymphatics after the lung-exit delay. Option B advances the lymphatics once per timestep, at the start of recruitment. Cells whose exit time, their entry time plus `Core_DCellLung_lymphaticsExitInterval`, has arrived are removed. Their count, scaled by `Recruit_ODE_scalingLung` × `Recruit_ODE_scalingMDC`, is added to the lymph-node antigen-presenting cell population. The lymphatics therefore delay the antigen signal by the exit interval, and their advance writes the lymphatic-exit statistics (S9). With no dendritic cells in the default configuration, the lymphatics stay empty.

Under option A nothing advances the lymphatics. If dendritic cells were placed at initialisation, the queue could only grow.

M2.11 What changes with the selection

Property	Option A, selected	Option B	See
Supply of recruitable cells	Unmodelled and inexhaustible; every event constructs a new agent and nothing is debited, so influx is limited by demand and space, never by supply	Finite; blood and lymph-node populations, depleted by recruitment	M2.8, M2.9
Limits on influx	Demand signal, open slots and the cascade division	Blood availability, the eligibility test at the source, and space; whether a cascade divides the slots is not described here	S8.3, S8.11; M2.9
Onset of the T cell response	Imposed: a timestep drawn once per simulation, then a linear ramp over a sampled interval; with `Recruit_Prob_mtbRecEnabled` = 0 at default, and does not respond to the state of the infection	Produced by the compartment dynamics; the onset schedule is not read	S8.10; M2.9
Magnitude of the T cell response	Five sampled parameters per type (`maxRecProb`, two thresholds, two half-saturations), the hard-coded weight split and the ramp	Compartment dynamics and efflux rates	S8.4, S8.13
Cognate fraction	A per-type Bernoulli draw at construction	An output of one draw over the blood compartments, weighted by concentration; the secreting T cell fraction changes without any S6 parameter	S8.12; M2.9; S6
Immunological memory	None	Naive, central memory, effector memory and effector subsets, outside the tissue	Spine 4.5
Second antigen specificity	Not represented	Non-cognate memory cells converted to cognate after a latency	M2.9
Dendritic cells	Not recruitable	Recruitable from monocytes with probability `Core_DCellLung_percentOfMacInitNumber`; trafficked to the node through the lymphatics	M2.10
Lymphatics	Allocated; never advanced	Advanced each timestep; delays the antigen signal by the exit interval	M2.10
State on the option object	None; saved as type identity only	Compartment populations and ODE coefficients, saved	M2.5, M2.6
Restart against a command-line selection	Immaterial	One driver order discards the restored state, and no warning is issued	M2.5
Source-active statistic	Slots at which a type was a cascade candidate	Incremented whenever the source is not full	S9
Eligible-source, crowded-source and recruited counts	Option A's statistics; the crowded-source count is identically zero	The same statistics; caseated sources excluded	M2.9; S9
Extra-granuloma, lymph-node and blood population, and dendritic-cell statistics	Identically zero	Written	M2.9, M2.10; S9

Property	Option A, selected	Option B	See
Per-compartment recruitment counters	General and per-type counters incremented	General counter only, for monocytes above the floor	S8.12; M2.9
Rate statistics	Base-class computation	The same computation; Treg total rate zero; each Treg counted cognate and non-cognate	M2.3; M2.9
Parameter groups with an effect	`Recruit_Prob` and its subgroups	`Recruit_ODE`; the per-type `Recruit_Prob` thresholds and half-saturations; `Recruit_OLD_Mac_probRec`; `Core_Mac_initNumber`; `Core_DCellLung`	S8.13; M2.13
Non-local reads	One: the whole-grid macrophage count for the floor, the only whole-grid quantity read by a recruitment or agent rule	The same floor, and at least the compartment populations, which are global by construction	S8.7

`Core_DCellLung` has an effect under either option wherever dendritic cells are placed (M2.13).

Consequences for the design concepts

Several statements of the spine's design concepts describe option A. A study selecting option B must revise them:

- **Adaptive immunity arriving late and ramping in** (spine 4.1, principle 5; 4.3) describes the imposed schedule. Under option B it becomes a claim about what the model produces.
- **The imposed-quantities table** (spine 4.2) loses the timing, cognate-fraction and magnitude rows of the T cell response, and gains the initial blood populations and the `Recruit_ODE` parameters. Imposed structure grows in total and shrinks at tissue level.
- **The absence of adaptive memory** (spine 4.5) becomes a claim about the tissue. Option B represents memory subsets outside it, and neither option represents memory within it. A recruit under either option arrives with a fixed specificity and no history, and the S2 and S3 rules are unchanged.

The spine's statement that no agent predicts is unaffected.

M2.12 Slot-invariant couplings

The vascular source list

Both options iterate the grid's fixed source list (spine 5.2), so studies differing only in this slot share vascular geometry for the same seed. The simulation permutes the list before each invocation (M2.7).

Admission and occupancy

The grid's per-type admission predicates, of which S8.9 specifies the recruitment test, are applied inside the simulation's agent factories, which throw if the compartment has no room. An option creating agents through the factories inherits the occupancy rules, the size-class exclusion and the caseation prohibition, and cannot violate them even without testing them. This enforces the slot and size-class limits of spine 2.3 independently of the slot. Option A tests the predicates itself before calling a factory, so for it the factory check always passes.

Nothing prevents an option from placing agents directly. Such an option would bypass both the occupancy enforcement and the statistics updates the factories make. Neutrophil injection and reinfection (M8.9) bypass the slot, so they increment no recruitment counter, but they create cells through the factories, so occupancy is enforced (M9.3).

Position in the timestep

The slot runs exactly once per timestep, at one point, under either option. A recruit computes a next state on its arrival timestep, but neither moves nor secretes until the next one (S8.1). The fields it reads already reflect this timestep's diffusion block.

No density gate at sources

Neither option applies a cell-density gate at sources; option B tests caseation, occupancy and the signal thresholds only. The asymmetry with oxygen delivery, which stops where the local cell density reaches 0.8 (S7, S8.1), is therefore slot-invariant.

Shared machinery

The eligibility test is shared by both options; only option A uses the weight (M2.3). Both options call the rate computation, so the rate statistics share one definition (M2.3).

M2.13 Parameters

Slot selection

Identifier	Default value	Units	Meaning
Recruitment command-line option	Option A	—	Which option the selector constructs (M2.1, M2.4)

Option A

S8.13 specifies option A's parameters. Their groups are:

- `Recruit_Prob`: four parameters governing the onset of T cell recruitment. Three are read by simulation routines rather than by the option object (M2.6), so the grouping follows the mechanism, not the class.
- `Recruit_Prob_Mac`: five (maximum probability, two thresholds, two half-saturations).
- `Recruit_Prob_Tcell_Tgam`, `Recruit_Prob_Tcell_Tcyt` and `Recruit_Prob_Tcell_Treg`: six each, the same five plus the cognate probability.
- `Recruit_Prob_Neutrophil`: ten, inert by default (M9.10.2).

S8 also lists the parameters it reads from other submodels' groups and its hard-coded quantities. Under option B the per-type threshold and half-saturation parameters are read too, through the shared predicates (M2.3).

Option B and dendritic cells

`Recruit_ODE` (71 parameters; subjects in M2.9), `Core_DCellLung` and the per-type `Recruit_OLD` parameters carry no declared default. A parameter file must therefore contain them whichever option is selected; omitting one is fatal at startup (spine 7.0, "Resolution"). Each is range-checked only where its declared minimum and maximum differ (spine 7.0, "Validity ranges"). The five `Recruit_ODE` entries declared with a minimum only, the four second-antigen conversion quantities and the conversion latency, are therefore unchecked. Appendix B indexes the group's numbering and declared ranges.

Spine 5.4 and Appendix B B.5 refer to this table for the `Core_DCellLung` parameters.

Identifier	Default value	Units	Meaning
`Core_DCellLung_percentOfMacInitNumber`	0	fraction	Initial dendritic cells as a fraction of the initial macrophage count; under option B, also the probability that a recruited monocyte becomes a dendritic cell
`Core_DCellLung_lungExitInterval`	—	—	Interval before a stimulated dendritic cell leaves the lung
`Core_DCellLung_lymphaticsExitInterval`	—	—	Transit time through the lymphatics to the draining lymph node (M2.10)

The group is named for the lung dendritic cell, not for the recruitment slot, and is not strictly option-specific: a positive fraction places dendritic cells at initialisation under either option, and the intervals then govern them. All three are inert by default because the fraction is 0.

Superseded scheme

The `Recruit_OLD` groups declare twelve parameters for a superseded recruitment scheme. Option A reads none of them; option B reads `Recruit_OLD_Mac_probRec` (M2.9). Appendix B indexes the groups and their defaults.

Computed parameters

- `Core_nrSources`, 7886: the length of the source list both options iterate (spine 5.2).
- `Core_Mac_initNumber`: sets the macrophage floor under both options (spine 5.3, S8.7).

Fixed structural constants

Two options; 18 rate statistics (three variants for each of six types).

Read but without effect by default

Under option A, as selected: `Recruit_ODE`, the `Core_DCellLung` intervals, the `Recruit_OLD` groups and `Recruit_Prob_Neutrophil`. Under option B: the `Recruit_Prob` onset and ramp parameters, except `Recruit_Prob_timeRecEnabled`, which still sets the onsets of the S4.7 modulator and the M8 post-adaptive factor (both inert by default); and the `Recruit_Prob_Tcell_<type>_probCognate` parameters.

M2.14 Interfaces

Independence of the slot

At the interface the slot is independent of every other slot. No other module reads the recruitment method; construction, cloning, saving and invocation ignore other selections; the interface names no other module's type; and only the driver can reach the selector. No other slot is selected on the basis of this one, and every combination is accepted. Other slots couple to an option directly, as the table records.

The slot and the T cell proliferation of S3 are the only routes by which an agent is created after initialisation by default. The M8 routes, which bypass the recruitment counters, are disabled by default.

With	Provides or receives	See
Spine module-slot table	Provides the M2 summary	M2.1
Spine 2.2	Receives the exchange of grid buffers, which forbids cached references	M2.4
Spine 2.3	Receives the slot and size-class limits the factories enforce	M2.12
Spine 3.2	Receives the position of the recruitment block	M2.7
Spine 4.1, 4.2, 4.3, 4.5	Option B revises the statements on adaptive timing, imposed quantities and memory	M2.11
Spine 5.2, 5.3, 5.4	Receives the source list, `Core_nrSources`, `Core_Mac_initNumber` and the dendritic-cell initialisation routes	M2.10, M2.12, M2.13
Spine 7.0	Receives the division between command-line and parameter-file settings, the precedence rule, and the omitted-parameter and range rules	M2.1, M2.5, M2.13
S3	Under option B every recruited Treg is non-cognate, and no Treg suppresses (S3.13)	M2.9
S4	The intracellular growth modulator reads `Recruit_Prob_timeRecEnabled` under every option (S4.7)	M2.6
S6	Under option B the fraction of secreting T cells changes without any S6 parameter	M2.9, M2.11

With	Provides or receives	See
S7	Oxygen is delivered at the same sources, gated by cell density; recruitment is not	M2.12
S8	Specifies option A; holds the recruitment admission test (S8.9), the density-gate asymmetry (S8.1), the floor both options share (S8.7) and option A's module couplings (S8.12, S8.14)	M2.8, M2.9, M2.12
S9	Holds the recruitment statistics, the counters, the rate definition, option B's statistics and the lymphatic-exit statistics	M2.3, M2.9, M2.10
M1	The diffusion slot installs a placeholder and saves its method (M1.1). The collapse of the three chemokine terms onto CCL2 (S8.5) holds while the S6.14 conditions hold (S8.5; M1.3, Derived assignment); the dependence lies in the shared predicates, so option B inherits it	M2.1, M2.5, M2.3
M3	With MHC class II enabled, recruited T cells are constructed in their T0 state under either option; each construction routine reads the flag directly. The slot neither mediates nor constrains the read, so a restricted interface would have to pass the flag explicitly. The program's help text says MHC class II requires option B, and the program does not require it (M3.6)	M2.2
M4	No interaction	—
M5	Changes the source count and the neutrophil saturation denominators (S8.14, M9.5)	M2.12
M6	Drug flux at the same sources, on the diffusion timestep; reads no recruitment quantity (S8.14)	M2.12
M7	Anti-TNF flux at the same sources alters recruitment through the TNF field (S8.14)	M2.12
M8	Injection and reinfection create cells outside the slot and increment none of its counters (M8.9). M2 sets which recruited T cells are cognate and active, and so eligible for exhaustion and induction; the post-adaptive factor keys on `Recruit_Prob_timeRecEnabled` (M8.12)	M2.6, M2.12
M9	Specifies the neutrophil predicate and the neutrophil parameters (M9.5, M9.10.2); neutrophil recruitment under option B is not specified	M2.3, M2.9
Appendix B	Indexes the selector toggle, `Recruit_ODE`, `Core_DCellLung`, `Recruit_OLD` and the unused dendritic-cell list, counter and statistics	M2.13