Supplementary information

High-resolution mapping of cell cycle dynamics during steady-state T-cell development and regeneration *in vivo*

Heike Kunze-Schumacher, Nikita A. Verheyden, Zoe Grewers, Michael Meyer-Hermann, Victor Greiff, Philippe A. Robert, Andreas Krueger

Figure S1

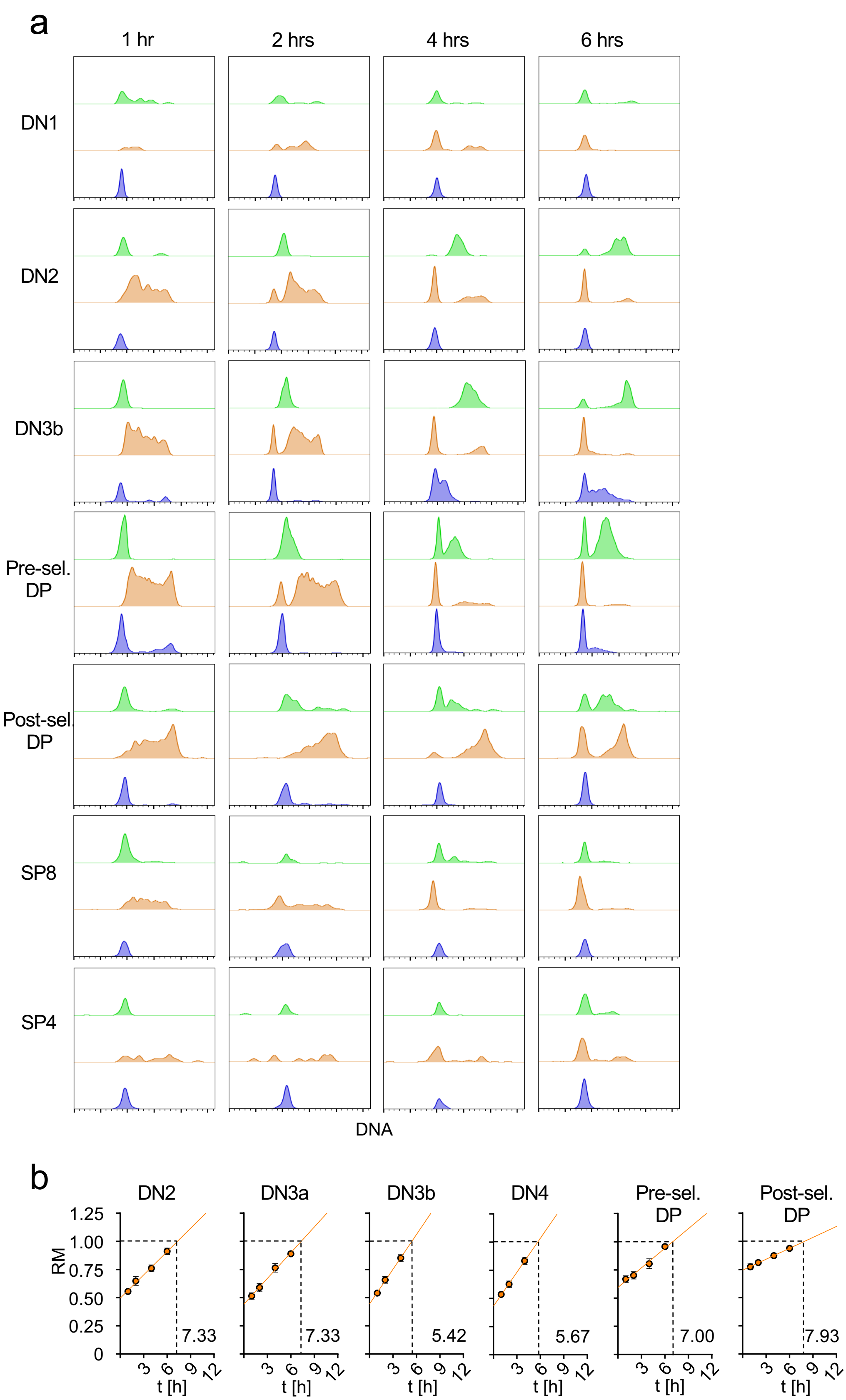


Figure S1. (a) Representative flow cytometric histograms of DNA content of DN1, DN2, DN3b, pre- and post-selection DP, SP8 and SP4 thymocytes of WT mice at indicated time points. Each plot depicts an overlay of the DNA content of $EdU^{-}BrdU^{+}$ (green), $EdU^{+}BrdU^{+}$ (orange) and $EdU^{+}BrdU^{-}$ (blue) cells. **(b)** Statistical analysis of WT thymocyte subpopulations to assess S-phase duration based on RM values of $EdU^{+}BrdU^{+}$ cells (mid/late S phase) over time (orange dots). The orange line represents the resulting linear regression. Numbers adjacent to linear regression show S-phase duration in h calculated based on linear regression, n = 3-5 mice for each point in time, data from 2 independent experiments.

Figure S2

a

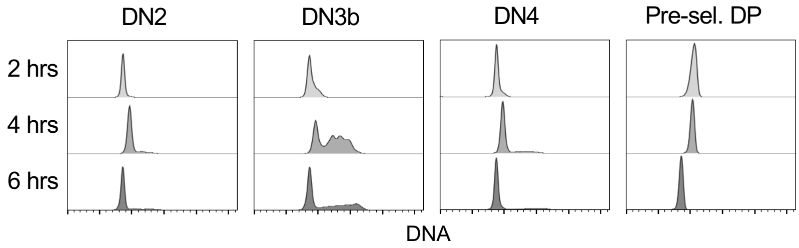


Figure S2. (a) Representative flow cytometric histograms of DNA content of EdU$^-$BrdU$^-$ DN2, DN3b, DN4 and pre-selection DP thymocytes of WT mice over time. Each plot represents an overlay of the DNA content at 2, 4 and 6 h.

Figure S3_1

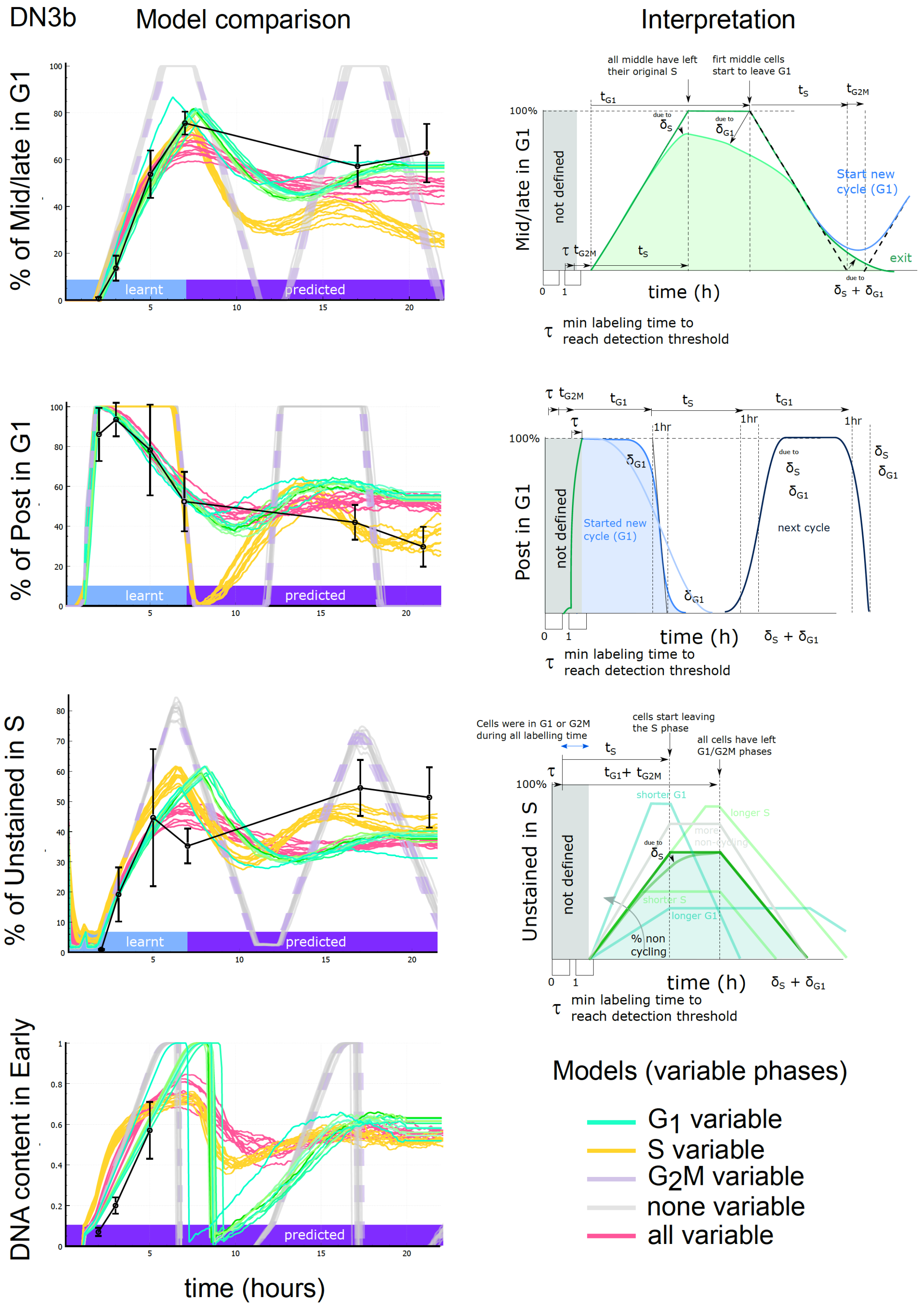
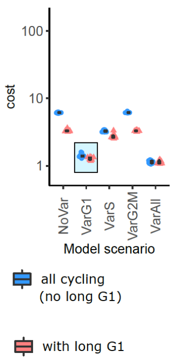


Figure S3_1. Variation in G1-phase but not S-phase duration is required to explain the dual-pulse labeling kinetics of DN3b thymocytes. Left: The simulation curves under the phase heterogeneity hypotheses are shown for the most informative experimental variables (see Figure S3_2 for comparison with all measured variables). Right: The interpretation of the curves is shown, including the expected effect of each phase variation (δS, $\delta G1$ and $\delta G2/M$) and of the amount of “long G1 cells” (% of non cycling).

Figure S3_2

DN3b



all cycling (no long G1 cells)

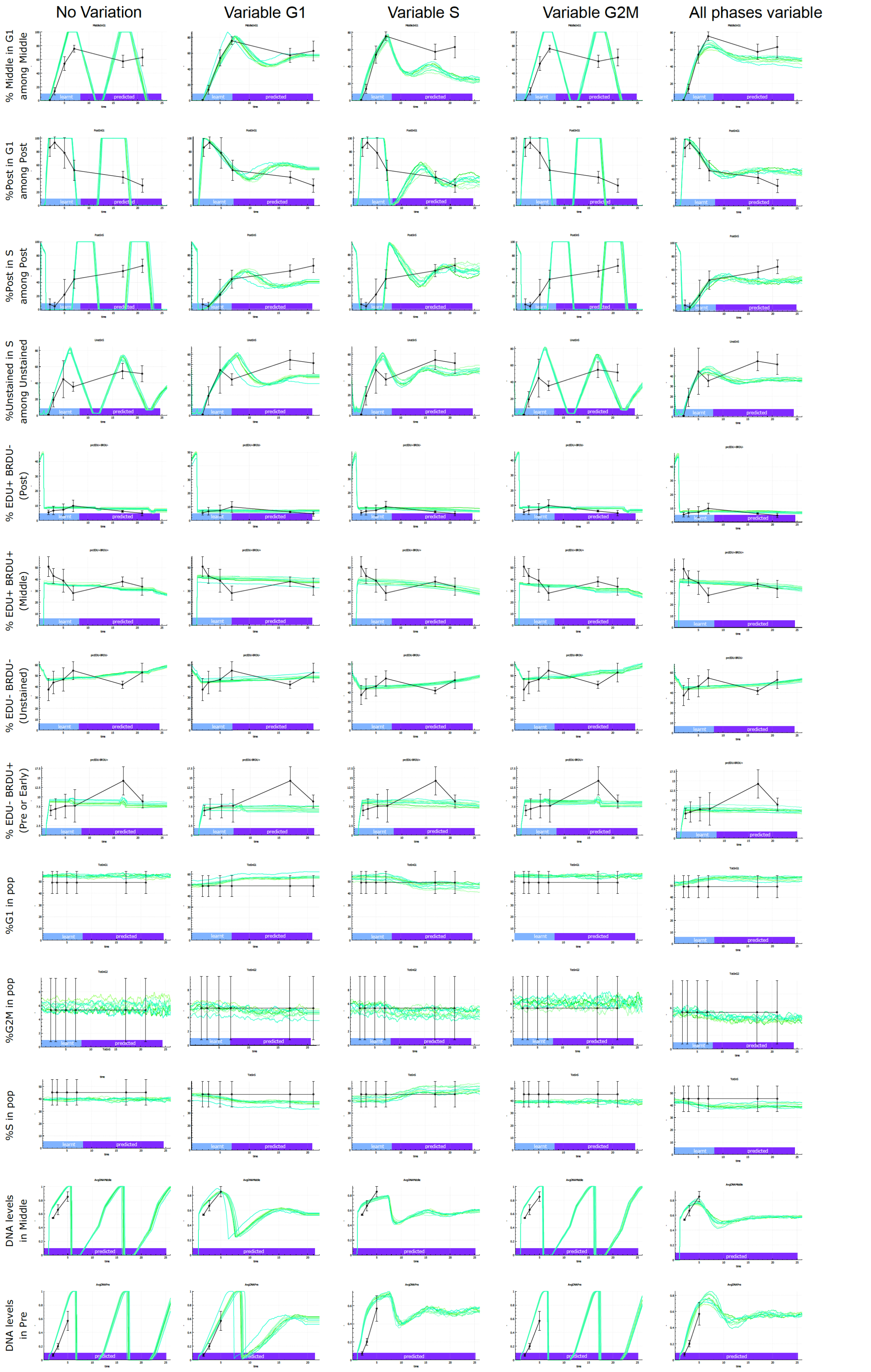


Figure S3_2

Figure S3_2. Variation in G1-phase but not S-phase duration is required to explain the dual-pulse labeling kinetics of DN3b thymocytes (detailed curves). The experimental data is shown in black, and the simulations from 10 independent fits are shown in green according to different model hypotheses on the stochasticity of phase durations at the population level. No variation: all the cells have the exact same duration of each cell cycle phase. Variable phases: only one phase is different between cells, and picked from a lognormal distribution. All phases variables: each phase follows a different log-normal distribution between different cells. For the top 11 observed variables, the 4 early time points are used for fitting (learnt), without knowledge of the remaining two time points (predicted). The bottom two variables, that depict the amount of DNA in EdU$^-$BrdU$^+$ and EdU$^+$BrdU$^+$ cells, were excluded from fitting and used as an independent qualitative validation dataset. No variation in phase durations, or only variation in the S phase cannot explain the dynamics of the EdU$^+$BrdU$^-$ cells or the DNA levels in EdU$^-$BrdU$^+$ cells, while variation in the G1 phase is sufficient to recapitulate all the observed variables. Allowing all phases to be variable does not improve the quality of the curves.

Figure S3_3

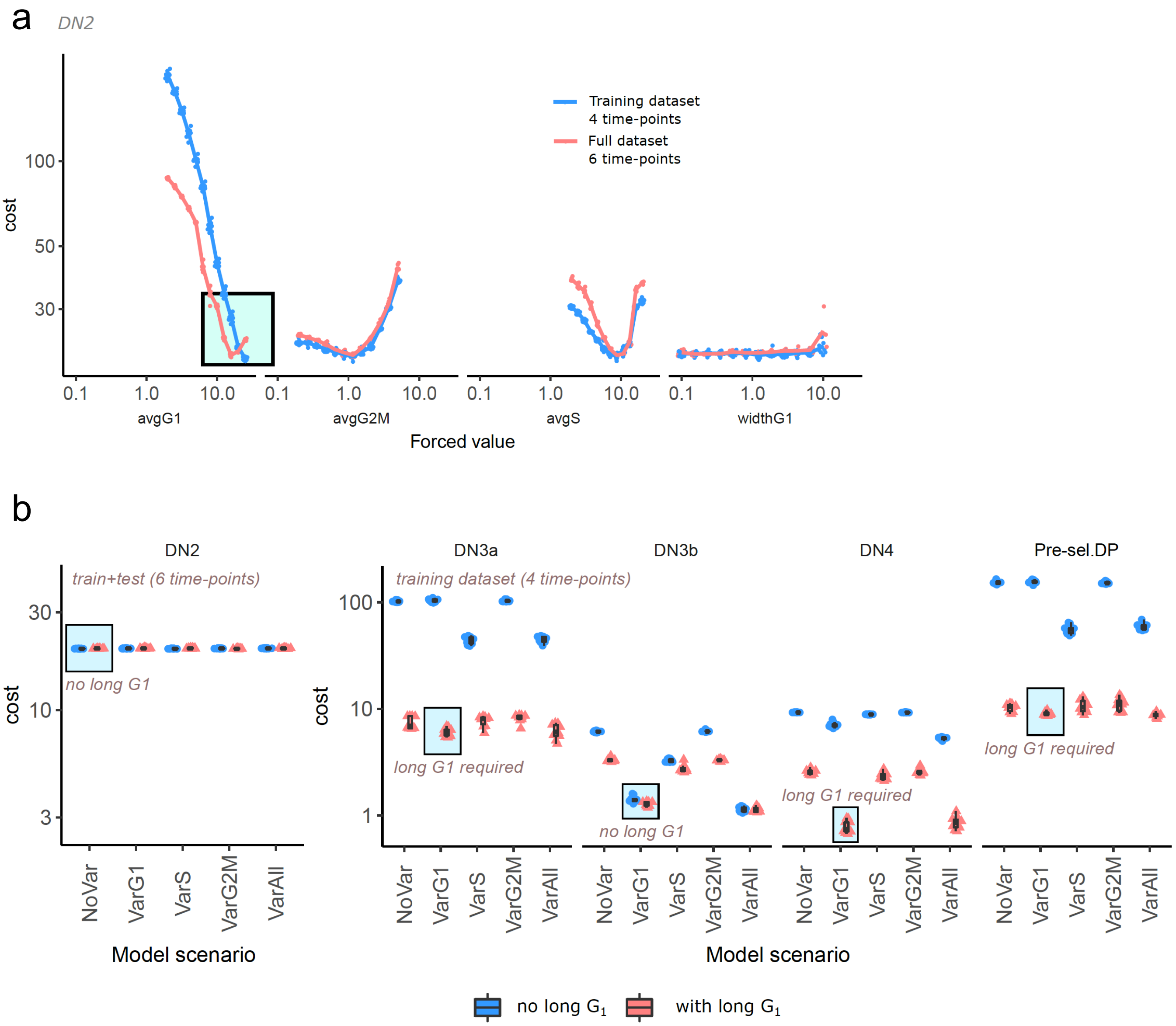


Figure S3_3. Cell cycle inference in DN2 requires the full 6 time points. (a) Identifiability analysis of phase durations with 4 (training dataset) or 6 (full dataset) time points. It is not possible to identify the G1 duration with only four time points. **(b)** Phase heterogeneity or long G1 cells do not improve the cost of simulations, meaning that DN2 labeling can be explained without phase heterogeneity or long G1 cells.

Figure S4

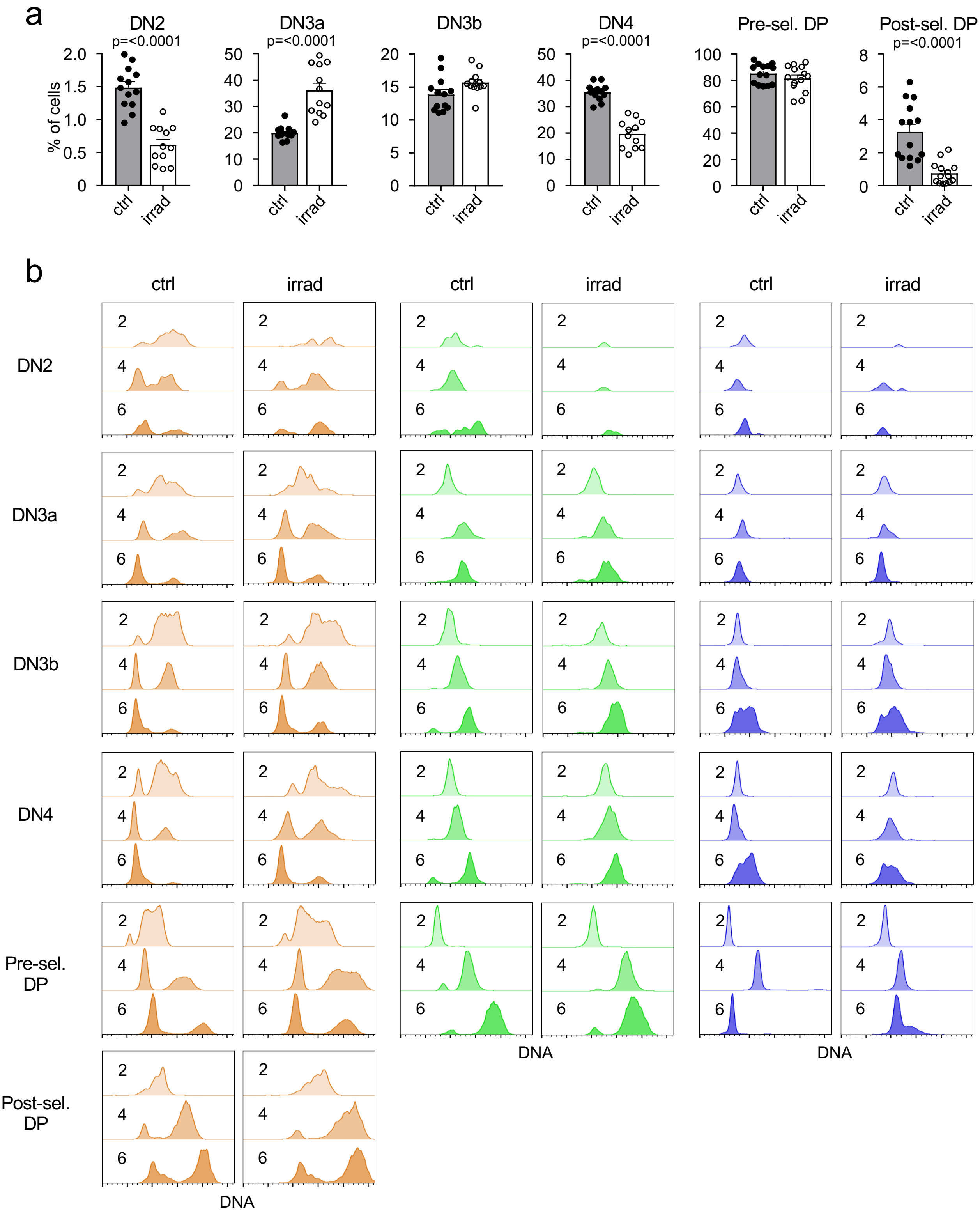


Figure S4. (a) Thymocyte subset composition from ctrl (grey, black dots) and irradiated WT (white, white dots) mice indicated as frequencies with n = 13-14 ctrl mice and n = 12-14 irradiated WT mice. **(b)** Representative flow cytometric histograms of DNA content of different thymocyte subsets of ctrl and irradiated WT mice over time of EdU$^+$BrdU$^+$ (orange), EdU$^-$BrdU$^+$ (green) or EdU$^+$BrdU$^-$ (blue) cells. Each individual plot represents an overlay of the DNA content at 2, 4 and 6 h.

Figure S5

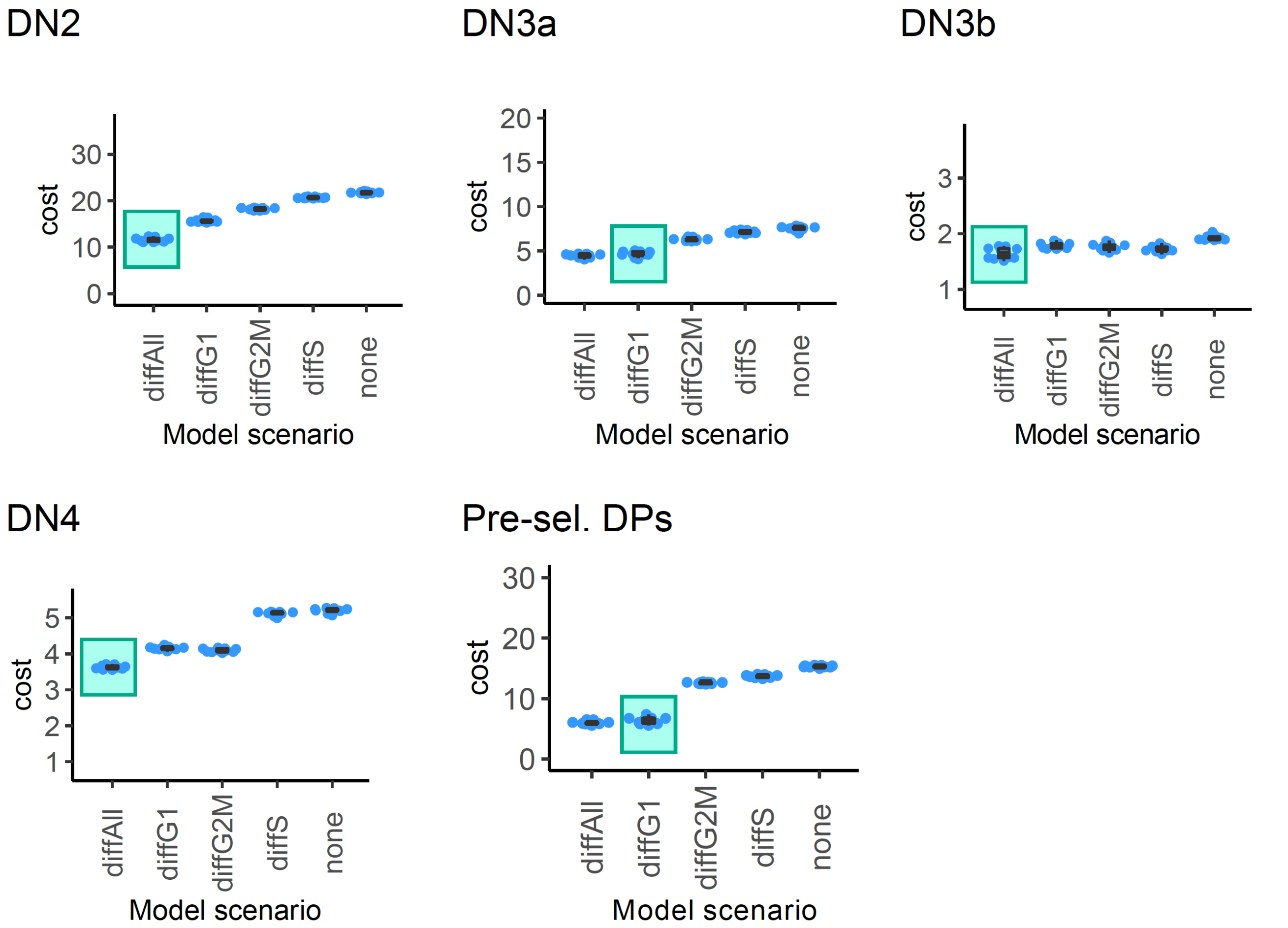
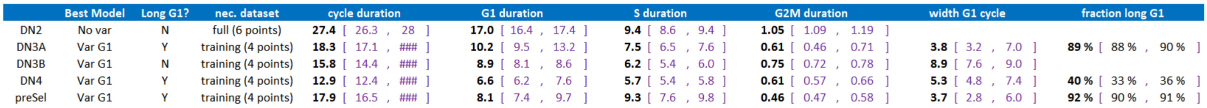


Figure S5. Comparison of hypotheses of modulated phase duration due to irradiation. For each population, cost of simulating the ctrl and irradiated datasets together with the same phase durations except the phases hypothetically modulated by irradiation. The model with minimal complexity raising a minimal cost is kept and shown in a green box.

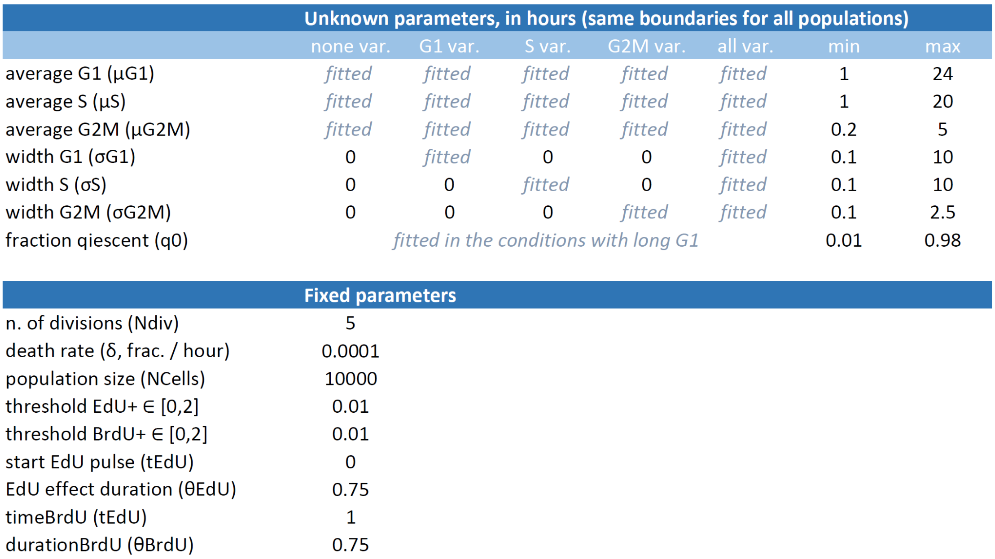
Table S1



	Best Model	Long G1?	nec. dataset	cycle duration	G1 duration	S duration	G2M duration	width G1 cycle	fraction long G1
DN2	No var	N	full (6 points)	**27.4** [26.3 , 28]	**17.0** [16.4 , 17.4]	**9.4** [8.6 , 9.4]	**1.05** [1.09 , 1.19]		
DN3A	Var G1	Y	training (4 points)	**18.3** [17.1 , ###]	**10.2** [9.5 , 13.2]	**7.5** [6.5 , 7.6]	**0.61** [0.46 , 0.71]	**3.8** [3.2 , 7.0]	**89 %** [88 % , 90 %]
DN3B	Var G1	N	training (4 points)	**15.8** [14.4 , ###]	**8.9** [8.1 , 8.6]	**6.2** [5.4 , 6.0]	**0.75** [0.72 , 0.78]	**8.9** [7.6 , 9.0]	
DN4	Var G1	Y	training (4 points)	**12.9** [12.4 , ###]	**6.6** [6.2 , 7.6]	**5.7** [5.4 , 5.8]	**0.61** [0.57 , 0.66]	**5.3** [4.8 , 7.4]	**40 %** [33 % , 36 %]
preSel	Var G1	Y	training (4 points)	**17.9** [16.5 , ###]	**8.1** [7.4 , 9.7]	**9.3** [7.6 , 9.8]	**0.46** [0.47 , 0.58]	**3.7** [2.8 , 6.0]	**92 %** [90 % , 91 %]

Table S1. Identified durations of cycle phases and variation per population. The average value for the best set of 10 independent parameter estimations are shown, with the best model hypothesis. The DN2 population required the full dataset (6 time points), while the training data from other populations was already sufficient to identify the cell cycle phase durations. Confidence intervals are shown, obtained from 10 independent parameter estimation after bootstrapping the experimental data (see Methods).

Table S2

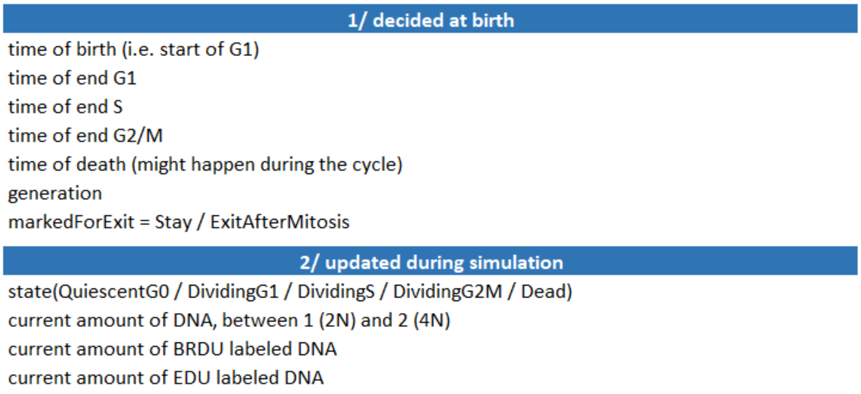


Unknown parameters, in hours (same boundaries for all populations)							
	none var.	G1 var.	S var.	G2M var.	all var.	min	max
average G1 (µG1)	*fitted*	*fitted*	*fitted*	*fitted*	*fitted*	1	24
average S (µS)	*fitted*	*fitted*	*fitted*	*fitted*	*fitted*	1	20
average G2M (µG2M)	*fitted*	*fitted*	*fitted*	*fitted*	*fitted*	0.2	5
width G1 (σG1)	0	*fitted*	0	0	*fitted*	0.1	10
width S (σS)	0	0	*fitted*	0	*fitted*	0.1	10
width G2M (σG2M)	0	0	0	*fitted*	*fitted*	0.1	2.5
fraction qiescent (q0)	*fitted in the conditions with long G1*					0.01	0.98

Fixed parameters	
n. of divisions (Ndiv)	5
death rate (δ, frac. / hour)	0.0001
population size (NCells)	10000
threshold EdU+ ∈ [0,2]	0.01
threshold BrdU+ ∈ [0,2]	0.01
start EdU pulse (tEdU)	0
EdU effect duration (θEdU)	0.75
timeBrdU (tEdU)	1
durationBrdU (θBrdU)	0.75

Table S2. Set of parameters defining a dual-pulse simulation for estimating the cell phase durations. Unknown parameters to be estimated are shown as ‘fitted’ with their respective minimum and maximum boundaries, depending on the heterogeneity hypothesis and the presence of “long G1” cells. Simulation parameters are also given.

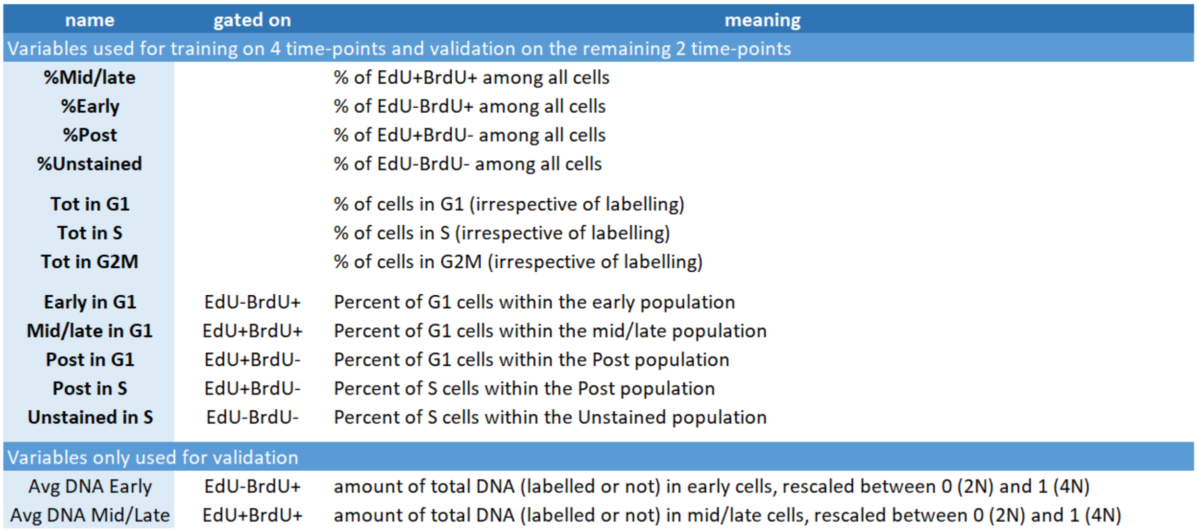
Table S3



1/ decided at birth
time of birth (i.e. start of G1)
time of end G1
time of end S
time of end G2/M
time of death (might happen during the cycle)
generation
markedForExit = Stay / ExitAfterMitosis
2/ updated during simulation
state(QuiescentG0 / DividingG1 / DividingS / DividingG2M / Dead)
current amount of DNA, between 1 (2N) and 2 (4N)
current amount of BRDU labeled DNA
current amount of EDU labeled DNA

Table S3. Description of agent properties used in the simulation. Events are decided at birth by sampling respective distributions. When daughter cells enter the last division in a population, daughter cells will be marked either for direct exit after completion of the cycle, or for another division, with a probability to have on average Ndiv divisions at the population level. DNA levels are also stored for each agent but are continuously updated during the simulation.

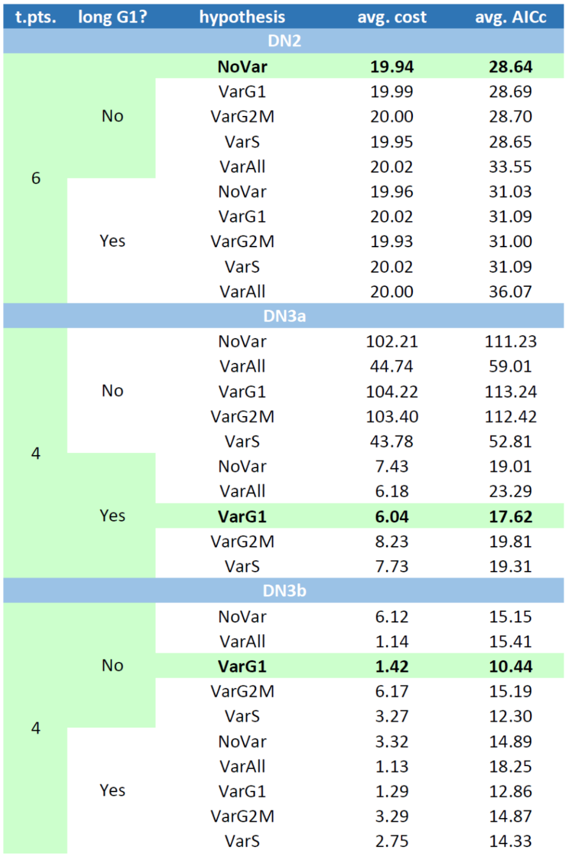
Table S4



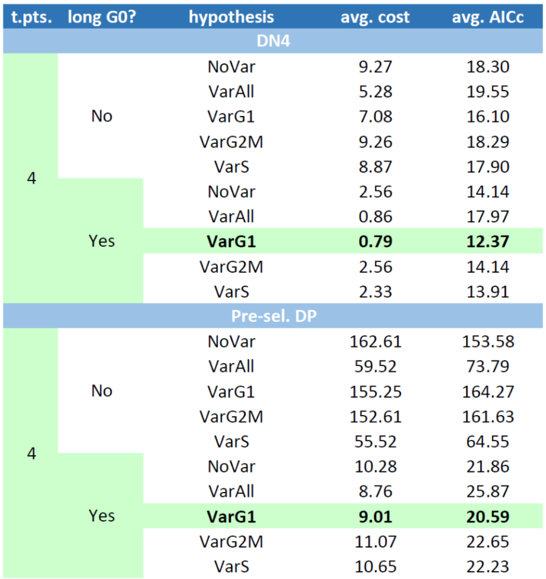
name	gated on	meaning
Variables used for training on 4 time-points and validation on the remaining 2 time-points		
%Mid/late		% of EdU+BrdU+ among all cells
%Early		% of EdU-BrdU+ among all cells
%Post		% of EdU+BrdU- among all cells
%Unstained		% of EdU-BrdU- among all cells
Tot in G1		% of cells in G1 (irrespective of labelling)
Tot in S		% of cells in S (irrespective of labelling)
Tot in G2M		% of cells in G2M (irrespective of labelling)
Early in G1	EdU-BrdU+	Percent of G1 cells within the early population
Mid/late in G1	EdU+BrdU+	Percent of G1 cells within the mid/late population
Post in G1	EdU+BrdU-	Percent of G1 cells within the Post population
Post in S	EdU+BrdU-	Percent of S cells within the Post population
Unstained in S	EdU-BrdU-	Percent of S cells within the Unstained population
Variables only used for validation		
Avg DNA Early	EdU-BrdU+	amount of total DNA (labelled or not) in early cells, rescaled between 0 (2N) and 1 (4N)
Avg DNA Mid/Late	EdU+BrdU+	amount of total DNA (labelled or not) in mid/late cells, rescaled between 0 (2N) and 1 (4N)

Table S4: Description of the experimentally observed variables used for estimating the cycle phases that are directly compared to simulation. Mid/late cells were in the S phase during both labeling periods, while early cells were not yet in the S phase during the first labeling (EdU) and entered the S phase during the second labeling (BrdU). Post cells were in the S phase at first labeling and left before the second labeling. Unstained cells were never in the S phase during the two labeling periods.

Table S5



t.pts.	long G1?	hypothesis	avg. cost	avg. AICc
		DN2		
6	No	**NoVar**	**19.94**	**28.64**
		VarG1	19.99	28.69
		VarG2M	20.00	28.70
		VarS	19.95	28.65
		VarAll	20.02	33.55
	Yes	NoVar	19.96	31.03
		VarG1	20.02	31.09
		VarG2M	19.93	31.00
		VarS	20.02	31.09
		VarAll	20.00	36.07
		DN3a		
4	No	NoVar	102.21	111.23
		VarAll	44.74	59.01
		VarG1	104.22	113.24
		VarG2M	103.40	112.42
		VarS	43.78	52.81
	Yes	NoVar	7.43	19.01
		VarAll	6.18	23.29
		VarG1	**6.04**	**17.62**
		VarG2M	8.23	19.81
		VarS	7.73	19.31
		DN3b		
4	No	NoVar	6.12	15.15
		VarAll	1.14	15.41
		VarG1	**1.42**	**10.44**
		VarG2M	6.17	15.19
		VarS	3.27	12.30
	Yes	NoVar	3.32	14.89
		VarAll	1.13	18.25
		VarG1	1.29	12.86
		VarG2M	3.29	14.87
		VarS	2.75	14.33

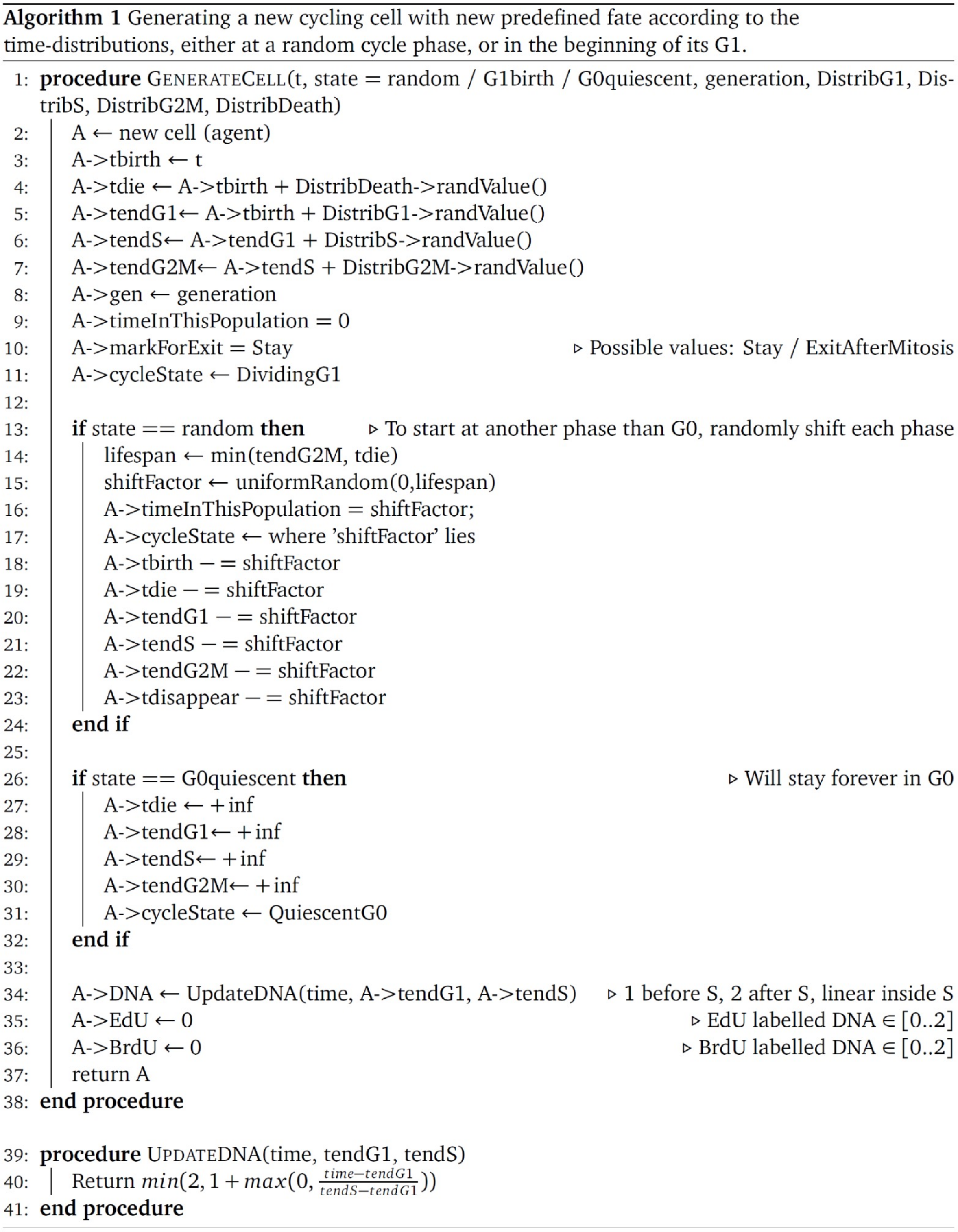


t.pts.	long G0?	hypothesis	avg. cost	avg. AICc
		DN4		
4	No	NoVar	9.27	18.30
		VarAll	5.28	19.55
		VarG1	7.08	16.10
		VarG2M	9.26	18.29
		VarS	8.87	17.90
	Yes	NoVar	2.56	14.14
		VarAll	0.86	17.97
		VarG1	**0.79**	**12.37**
		VarG2M	2.56	14.14
		VarS	2.33	13.91
		Pre-sel. DP		
4	No	NoVar	162.61	153.58
		VarAll	59.52	73.79
		VarG1	155.25	164.27
		VarG2M	152.61	161.63
		VarS	55.52	64.55
	Yes	NoVar	10.28	21.86
		VarAll	8.76	25.87
		VarG1	**9.01**	**20.59**
		VarG2M	11.07	22.65
		VarS	10.65	22.23

Table S5. Comparison of cycle heterogeneity hypotheses. The average cost and average AICc value of 10 independent parameter estimations are shown for each hypothesis on the WT dataset. The best hypothesis is highlighted in green.

Algorithms 1-4

Algorithms 1-4: Algorithmic description of the agent-based model.
Algorithm 1: Generation of a new cell at G0, G1 or at a random phase of the cycle.

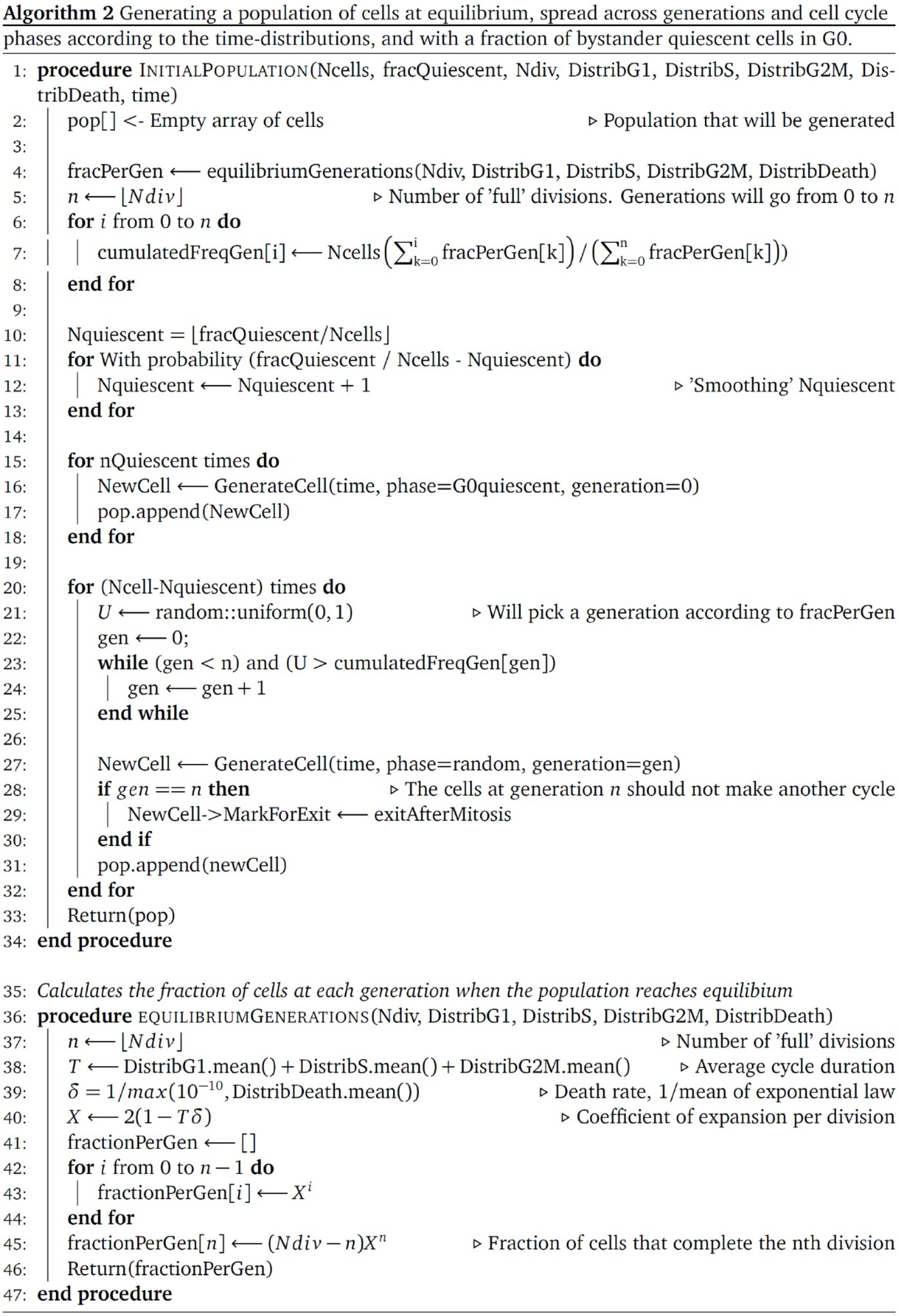


Algorithm 1 Generating a new cycling cell with new predefined fate according to the time-distributions, either at a random cycle phase, or in the beginning of its G1.

```
procedure GenerateCell(t, state = random / G1birth / G0quiescent, generation, DistribG1, DistribS, DistribG2M, DistribDeath)
    A ← new cell (agent)
    A->tbirth ← t
    A->tdie ← A->tbirth + DistribDeath->randValue()
    A->tendG1← A->tbirth + DistribG1->randValue()
    A->tendS← A->tendG1 + DistribS->randValue()
    A->tendG2M← A->tendS + DistribG2M->randValue()
    A->gen ← generation
    A->timeInThisPopulation = 0
    A->markForExit = Stay                          ▷ Possible values: Stay / ExitAfterMitosis
    A->cycleState ← DividingG1

    if state == random then        ▷ To start at another phase than G0, randomly shift each phase
        lifespan ← min(tendG2M, tdie)
        shiftFactor ← uniformRandom(0,lifespan)
        A->timeInThisPopulation = shiftFactor;
        A->cycleState ← where 'shiftFactor' lies
        A->tbirth − = shiftFactor
        A->tdie − = shiftFactor
        A->tendG1 − = shiftFactor
        A->tendS − = shiftFactor
        A->tendG2M − = shiftFactor
        A->tdisappear − = shiftFactor
    end if

    if state == G0quiescent then                              ▷ Will stay forever in G0
        A->tdie ← +inf
        A->tendG1← +inf
        A->tendS← +inf
        A->tendG2M← +inf
        A->cycleState ← QuiescentG0
    end if

    A->DNA ← UpdateDNA(time, A->tendG1, A->tendS)   ▷ 1 before S, 2 after S, linear inside S
    A->EdU ← 0                                           ▷ EdU labelled DNA ∈ [0..2]
    A->BrdU ← 0                                          ▷ BrdU labelled DNA ∈ [0..2]
    return A
end procedure

procedure UpdateDNA(time, tendG1, tendS)
    Return $min(2, 1 + max(0, \frac{time-tendG1}{tendS-tendG1}))$
end procedure
```

Algorithms 1-4

Algorithms 1-4: Algorithmic description of the agent-based model.
Algorithm 2: Creation of an initial population of cells already at steady-state generations and cycle phases.

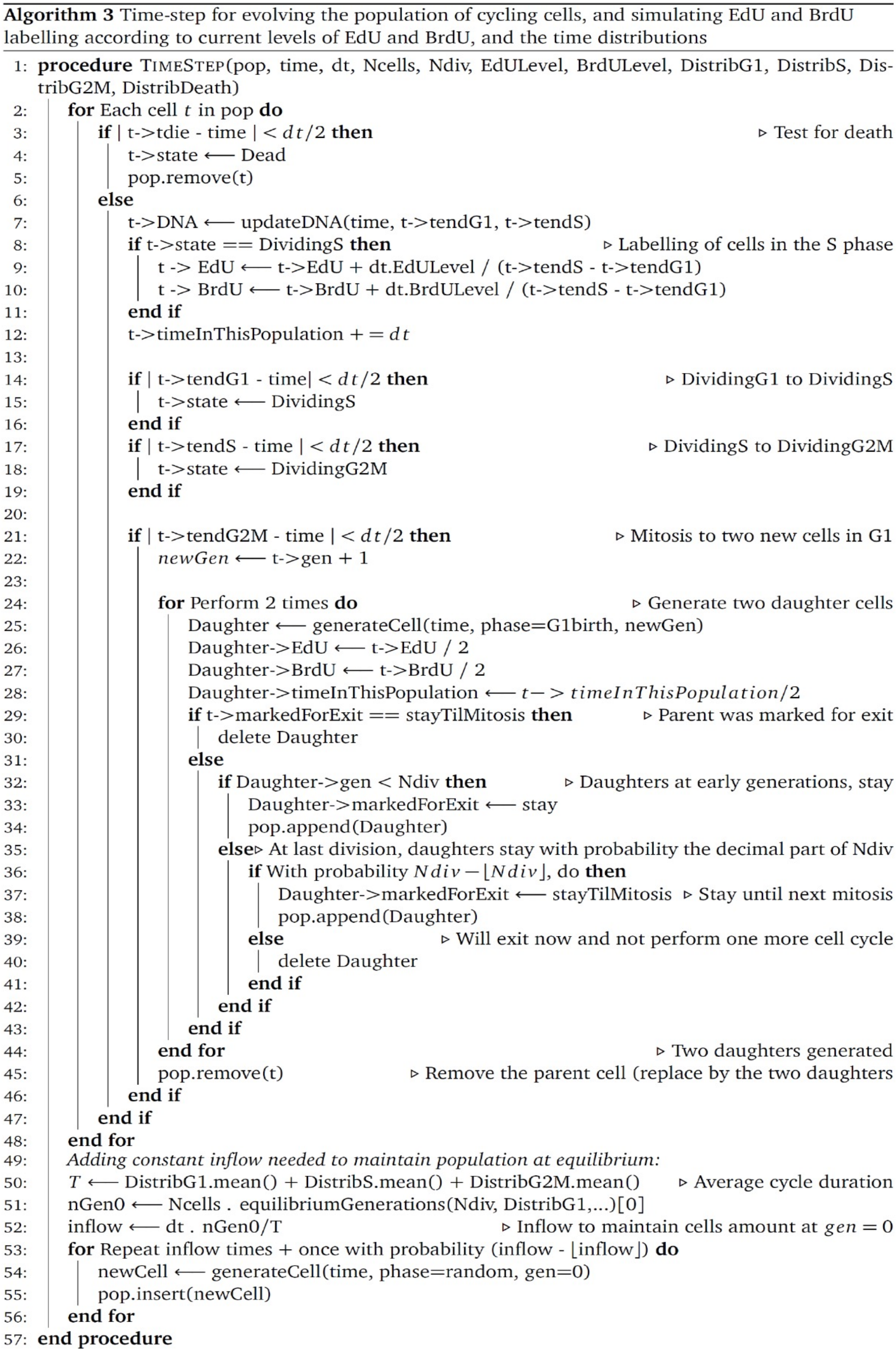


Algorithm 2 Generating a population of cells at equilibrium, spread across generations and cell cycle phases according to the time-distributions, and with a fraction of bystander quiescent cells in G0.

```
procedure InitialPopulation(Ncells, fracQuiescent, Ndiv, DistribG1, DistribS, DistribG2M, DistribDeath, time)
    pop[] <- Empty array of cells                                    ▷ Population that will be generated

    fracPerGen ⟵ equilibriumGenerations(Ndiv, DistribG1, DistribS, DistribG2M, DistribDeath)
    $n \longleftarrow \lfloor Ndiv \rfloor$                ▷ Number of 'full' divisions. Generations will go from 0 to $n$
    for $i$ from 0 to $n$ do
        cumulatedFreqGen[i] ⟵ $\mathrm{Ncells}\left(\sum_{k=0}^{i}\mathrm{fracPerGen[k]}\right)/\left(\sum_{k=0}^{n}\mathrm{fracPerGen[k]}\right))$
    end for

    Nquiescent = ⌊fracQuiescent/Ncells⌋
    for With probability (fracQuiescent / Ncells - Nquiescent) do
        Nquiescent ⟵ Nquiescent + 1                                  ▷ 'Smoothing' Nquiescent
    end for

    for nQuiescent times do
        NewCell ⟵ GenerateCell(time, phase=G0quiescent, generation=0)
        pop.append(NewCell)
    end for

    for (Ncell-Nquiescent) times do
        $U$ ⟵ random::uniform(0, 1)                  ▷ Will pick a generation according to fracPerGen
        gen ⟵ 0;
        while (gen < n) and (U > cumulatedFreqGen[gen])
            gen ⟵ gen + 1
        end while

        NewCell ⟵ GenerateCell(time, phase=random, generation=gen)
        if $gen == n$ then                ▷ The cells at generation $n$ should not make another cycle
            NewCell->MarkForExit ⟵ exitAfterMitosis
        end if
        pop.append(newCell)
    end for
    Return(pop)
end procedure

Calculates the fraction of cells at each generation when the population reaches equilibium
procedure equilibriumGenerations(Ndiv, DistribG1, DistribS, DistribG2M, DistribDeath)
    $n \longleftarrow \lfloor Ndiv \rfloor$                                       ▷ Number of 'full' divisions
    $T$ ⟵ DistribG1.mean() + DistribS.mean() + DistribG2M.mean()      ▷ Average cycle duration
    $\delta = 1/max(10^{-10}$, DistribDeath.mean())             ▷ Death rate, 1/mean of exponential law
    $X \longleftarrow 2(1 - T\delta)$                                ▷ Coefficient of expansion per division
    fractionPerGen ⟵ []
    for $i$ from 0 to $n - 1$ do
        fractionPerGen[$i$] ⟵ $X^{i}$
    end for
    fractionPerGen[$n$] ⟵ $(Ndiv - n)X^{n}$          ▷ Fraction of cells that complete the nth division
    Return(fractionPerGen)
end procedure
```

Algorithms 1-4

Algorithms 1-4: Algorithmic description of the agent-based model.
Algorithm 3: Step-by-step update of the population and cell labeling.

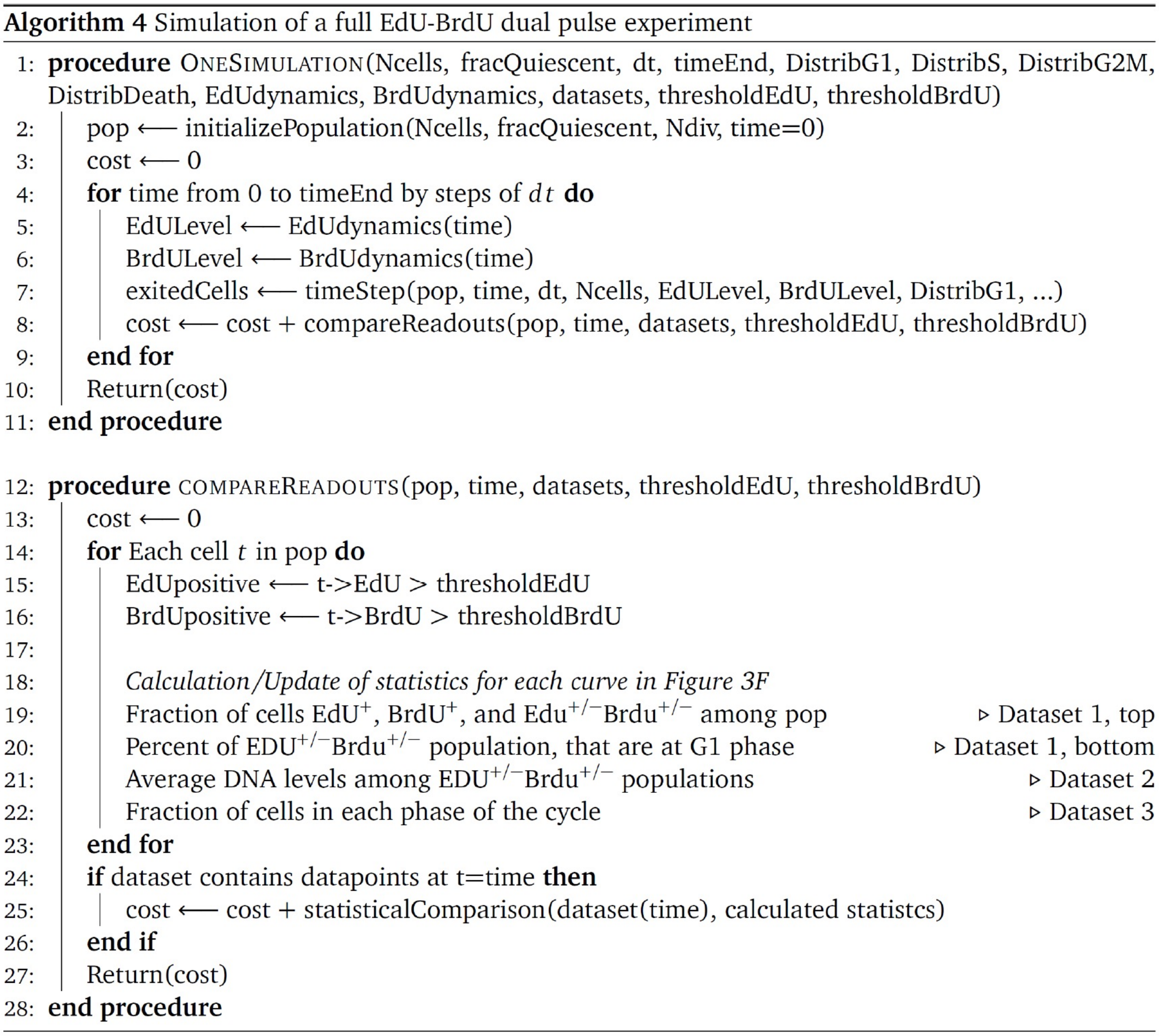


Algorithm 3 Time-step for evolving the population of cycling cells, and simulating EdU and BrdU labelling according to current levels of EdU and BrdU, and the time distributions

```
procedure TimeStep(pop, time, dt, Ncells, Ndiv, EdULevel, BrdULevel, DistribG1, DistribS, DistribG2M, DistribDeath)
    for Each cell $t$ in pop do
        if | t->tdie - time | $< dt/2$ then                                   ▷ Test for death
            t->state ⟵ Dead
            pop.remove(t)
        else
            t->DNA ⟵ updateDNA(time, t->tendG1, t->tendS)
            if t->state == DividingS then                      ▷ Labelling of cells in the S phase
                t -> EdU ⟵ t->EdU + dt.EdULevel / (t->tendS - t->tendG1)
                t -> BrdU ⟵ t->BrdU + dt.BrdULevel / (t->tendS - t->tendG1)
            end if
            t->timeInThisPopulation $+ = dt$

            if | t->tendG1 - time| $< dt/2$ then                        ▷ DividingG1 to DividingS
                t->state ⟵ DividingS
            end if
            if | t->tendS - time | $< dt/2$ then                       ▷ DividingS to DividingG2M
                t->state ⟵ DividingG2M
            end if

            if | t->tendG2M - time | $< dt/2$ then                 ▷ Mitosis to two new cells in G1
                $newGen$ ⟵ t->gen + 1

                for Perform 2 times do                          ▷ Generate two daughter cells
                    Daughter ⟵ generateCell(time, phase=G1birth, newGen)
                    Daughter->EdU ⟵ t->EdU / 2
                    Daughter->BrdU ⟵ t->BrdU / 2
                    Daughter->timeInThisPopulation ⟵ $t -> timeInThisPopulation/2$
                    if t->markedForExit == stayTilMitosis then        ▷ Parent was marked for exit
                        delete Daughter
                    else
                        if Daughter->gen < Ndiv then          ▷ Daughters at early generations, stay
                            Daughter->markedForExit ⟵ stay
                            pop.append(Daughter)
                        else▷ At last division, daughters stay with probability the decimal part of Ndiv
                            if With probability $Ndiv - \lfloor Ndiv \rfloor$, do then
                                Daughter->markedForExit ⟵ stayTilMitosis  ▷ Stay until next mitosis
                                pop.append(Daughter)
                            else             ▷ Will exit now and not perform one more cell cycle
                                delete Daughter
                            end if
                        end if
                    end if
                end for                                              ▷ Two daughters generated
                pop.remove(t)          ▷ Remove the parent cell (replace by the two daughters
            end if
        end if
    end for
    Adding constant inflow needed to maintain population at equilibrium:
    $T$ ⟵ DistribG1.mean() + DistribS.mean() + DistribG2M.mean()       ▷ Average cycle duration
    nGen0 ⟵ Ncells . equilibriumGenerations(Ndiv, DistribG1,...)[0]
    inflow ⟵ dt . nGen0/T                         ▷ Inflow to maintain cells amount at $gen = 0$
    for Repeat inflow times + once with probability (inflow - ⌊inflow⌋) do
        newCell ⟵ generateCell(time, phase=random, gen=0)
        pop.insert(newCell)
    end for
end procedure
```

Algorithms 1-4

Algorithms 1-4: Algorithmic description of the agent-based model.
Algorithm 4: Main organization of a simulation returning the cost of a parameter set.



Algorithm 4 Simulation of a full EdU-BrdU dual pulse experiment

```
procedure ONESIMULATION(Ncells, fracQuiescent, dt, timeEnd, DistribG1, DistribS, DistribG2M, DistribDeath, EdUdynamics, BrdUdynamics, datasets, thresholdEdU, thresholdBrdU)
    pop ⟵ initializePopulation(Ncells, fracQuiescent, Ndiv, time=0)
    cost ⟵ 0
    for time from 0 to timeEnd by steps of $dt$ do
        EdULevel ⟵ EdUdynamics(time)
        BrdULevel ⟵ BrdUdynamics(time)
        exitedCells ⟵ timeStep(pop, time, dt, Ncells, EdULevel, BrdULevel, DistribG1, ...)
        cost ⟵ cost + compareReadouts(pop, time, datasets, thresholdEdU, thresholdBrdU)
    end for
    Return(cost)
end procedure

procedure COMPAREREADOUTS(pop, time, datasets, thresholdEdU, thresholdBrdU)
    cost ⟵ 0
    for Each cell $t$ in pop do
        EdUpositive ⟵ t->EdU > thresholdEdU
        BrdUpositive ⟵ t->BrdU > thresholdBrdU

        Calculation/Update of statistics for each curve in Figure 3F
        Fraction of cells EdU$^+$, BrdU$^+$, and Edu$^{+/-}$Brdu$^{+/-}$ among pop          ▷ Dataset 1, top
        Percent of EDU$^{+/-}$Brdu$^{+/-}$ population, that are at G1 phase           ▷ Dataset 1, bottom
        Average DNA levels among EDU$^{+/-}$Brdu$^{+/-}$ populations                    ▷ Dataset 2
        Fraction of cells in each phase of the cycle                                    ▷ Dataset 3
    end for
    if dataset contains datapoints at t=time then
        cost ⟵ cost + statisticalComparison(dataset(time), calculated statistcs)
    end if
    Return(cost)
end procedure
```