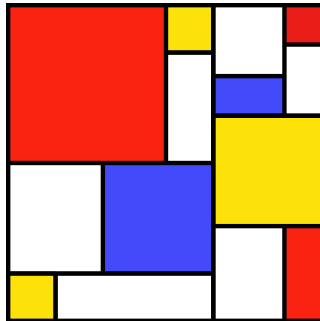


SLiM

Workshop Series

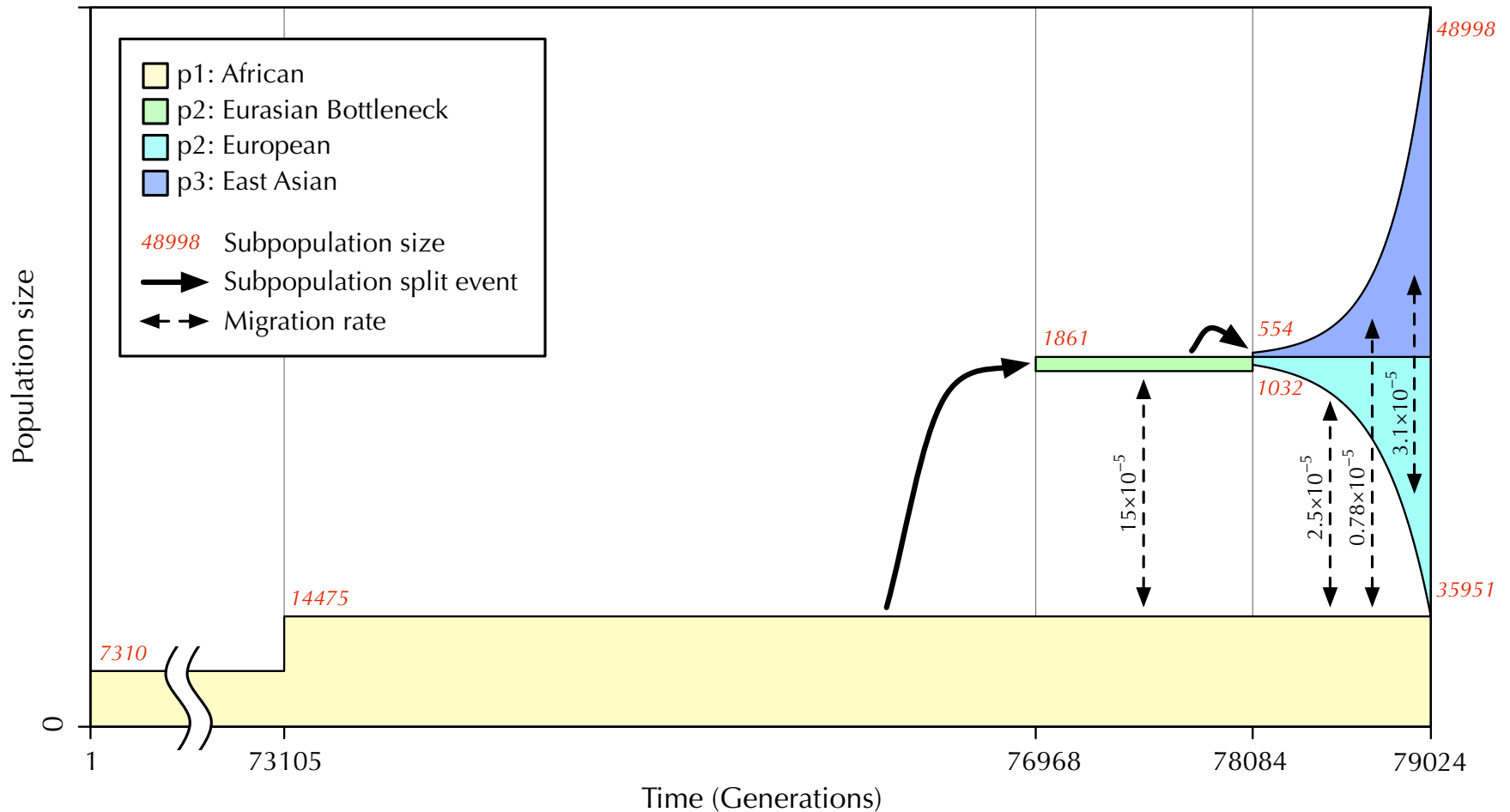


#5: Demography

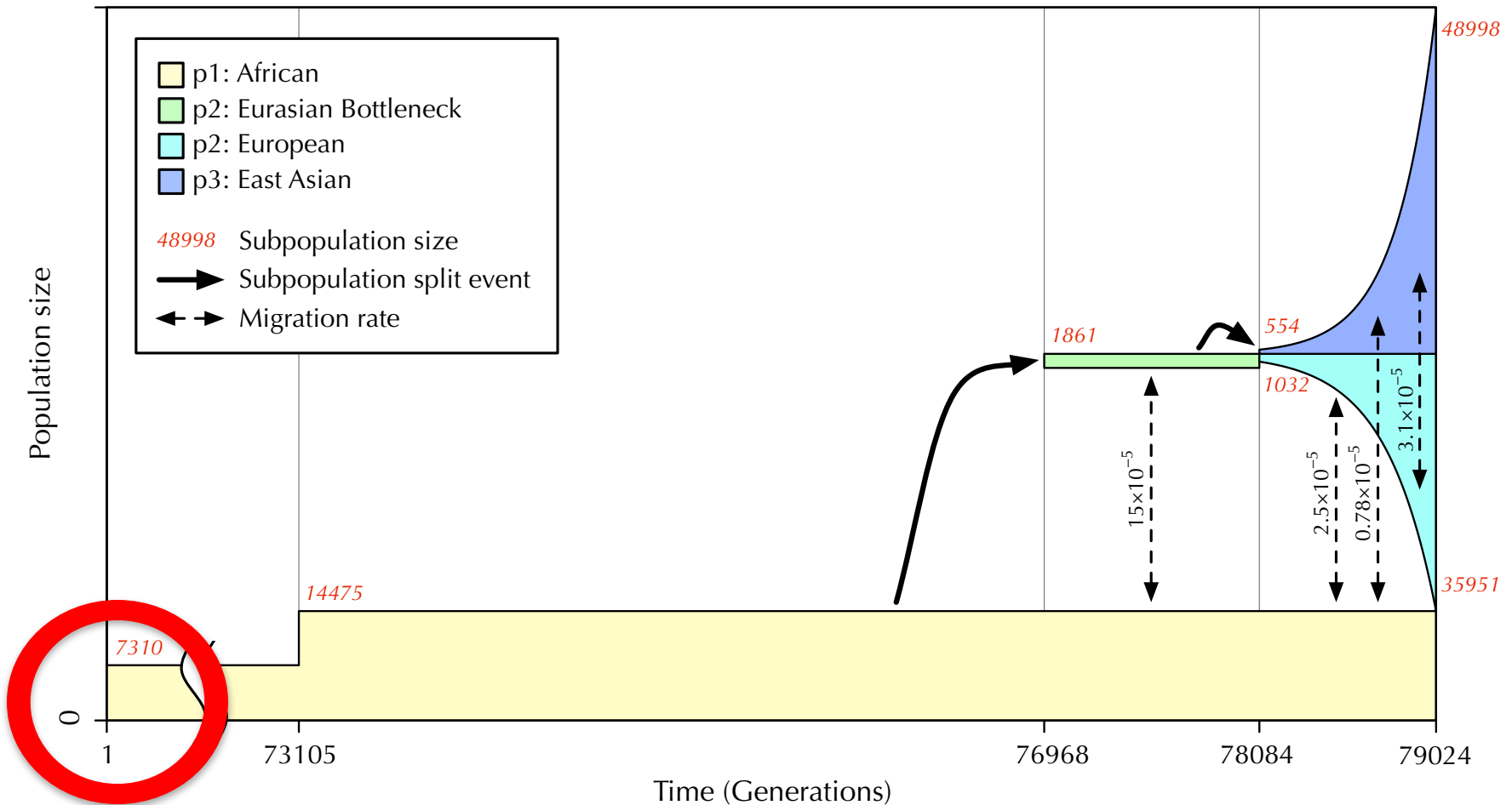
Demography

- Demographic events in SLiM:
 - new subpopulation: `addSubpop()`
 - splits a subpopulation: `addSubpopSplit()`
 - change in size: `setSubpopulationSize()`
 - remove a subpop: `setSubpopulationSize(0)`
 - set migration rate: `setMigrationRates()`
 - merge subpops:
`addSubpop() + setMigrationRates()`

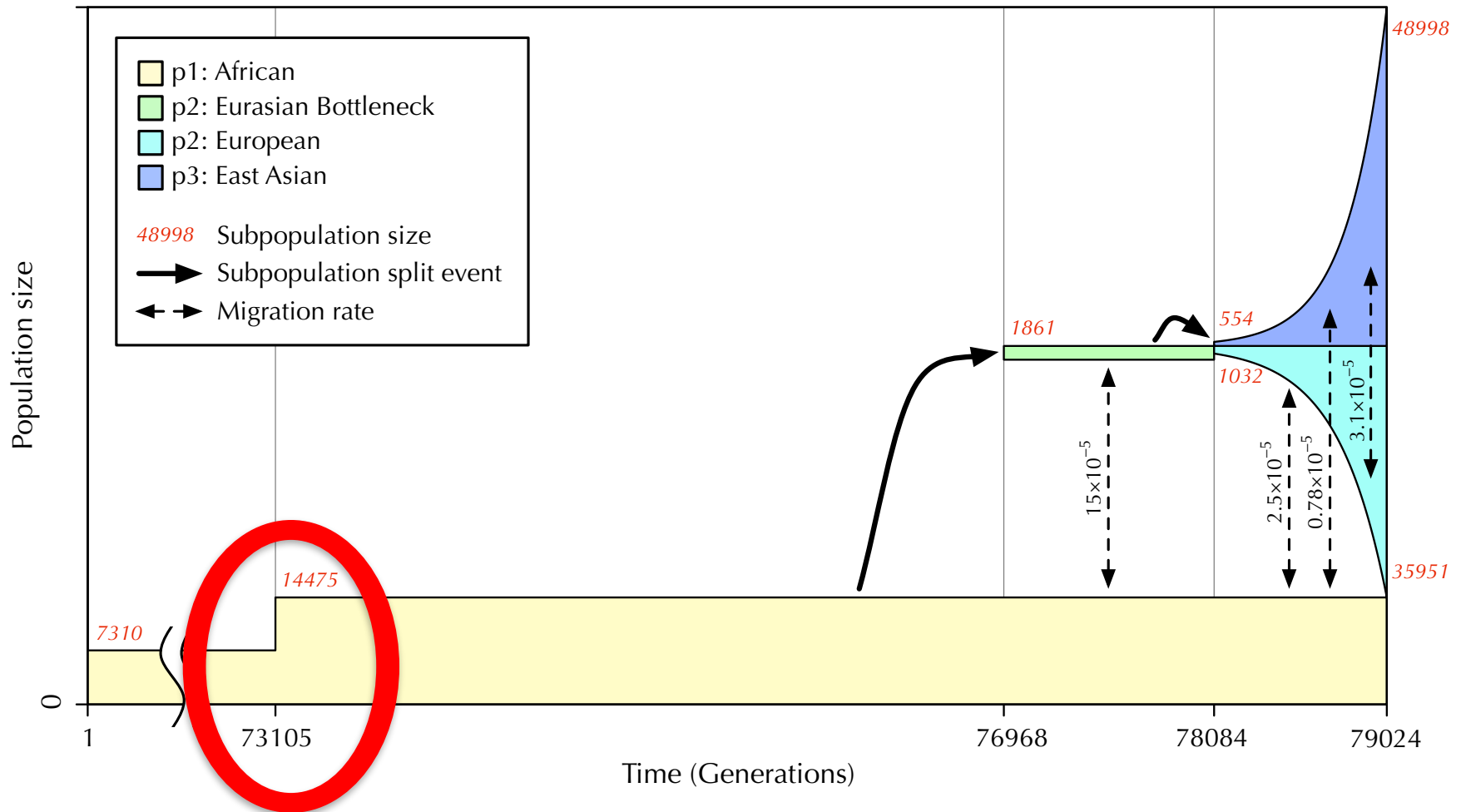
The Gravel Model



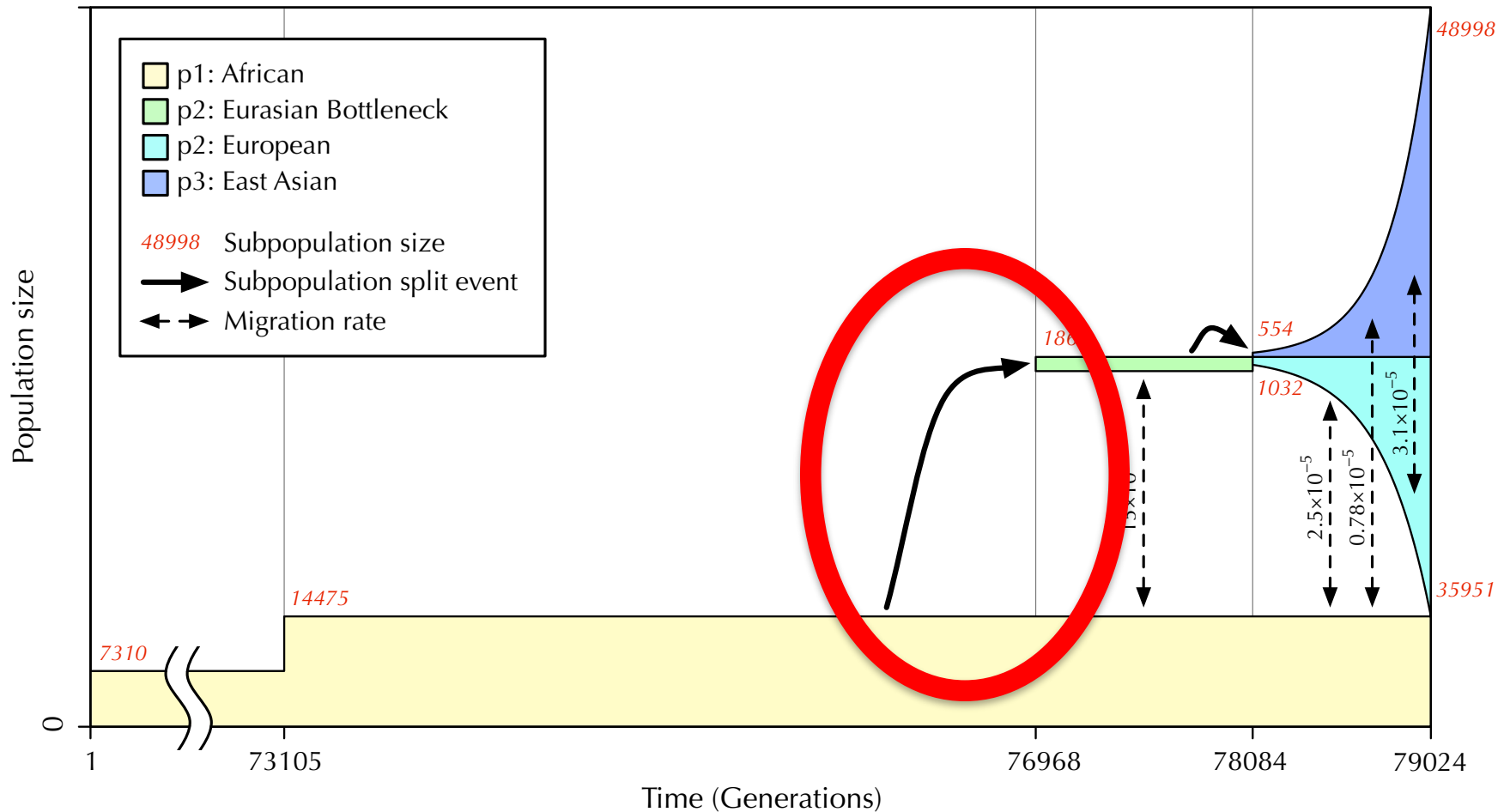
The Gravel Model



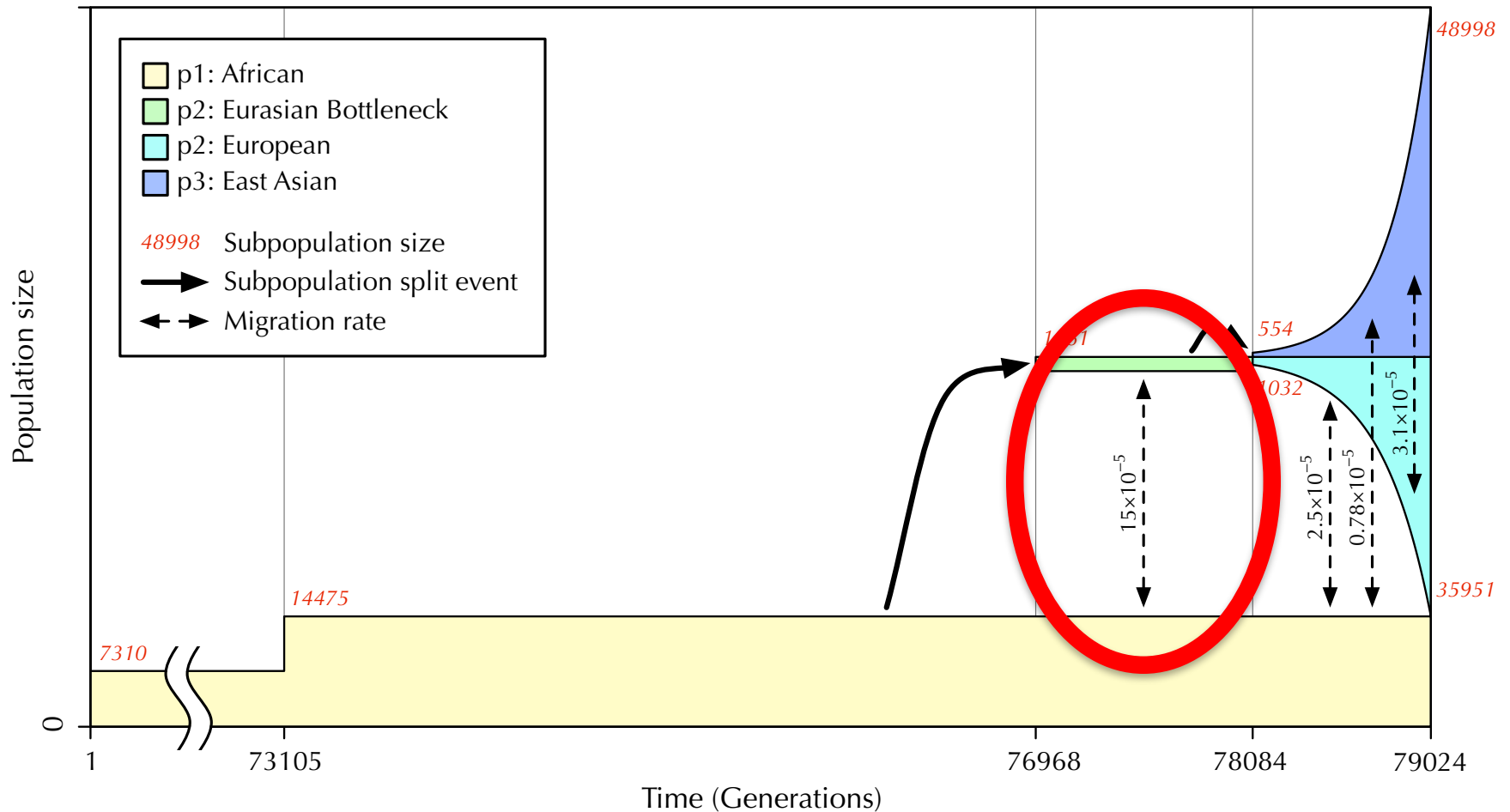
The Gravel Model



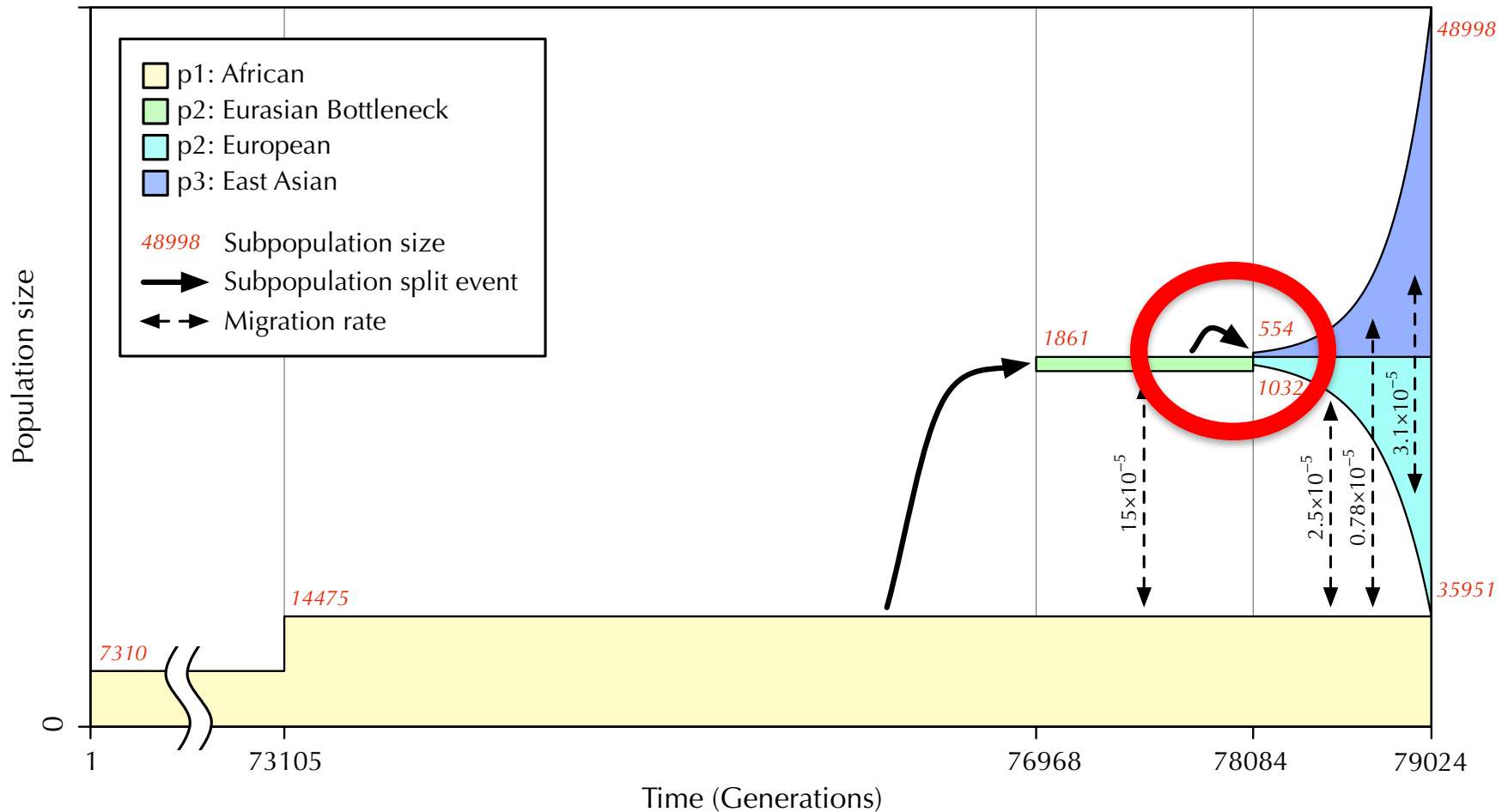
The Gravel Model



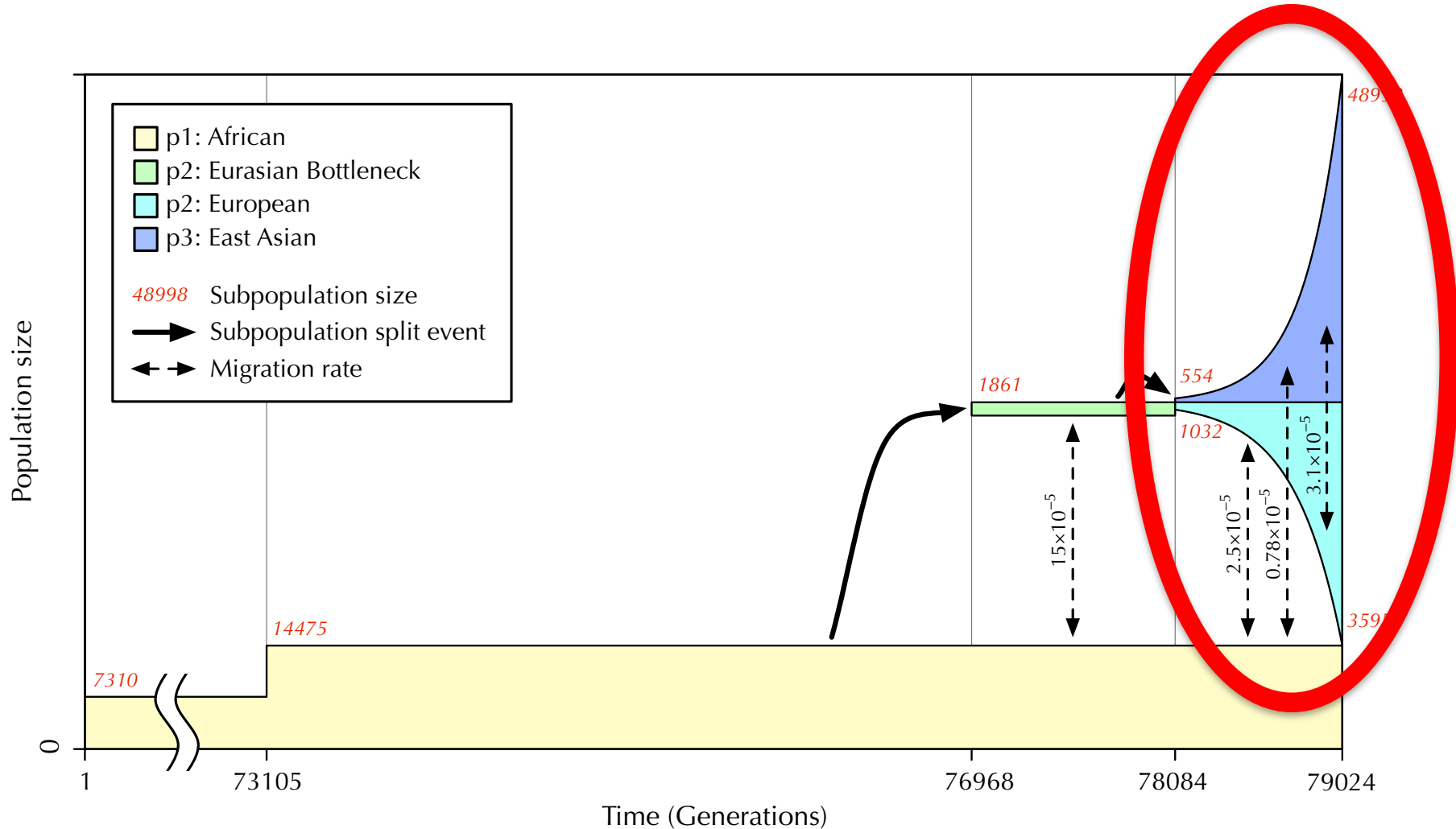
The Gravel Model



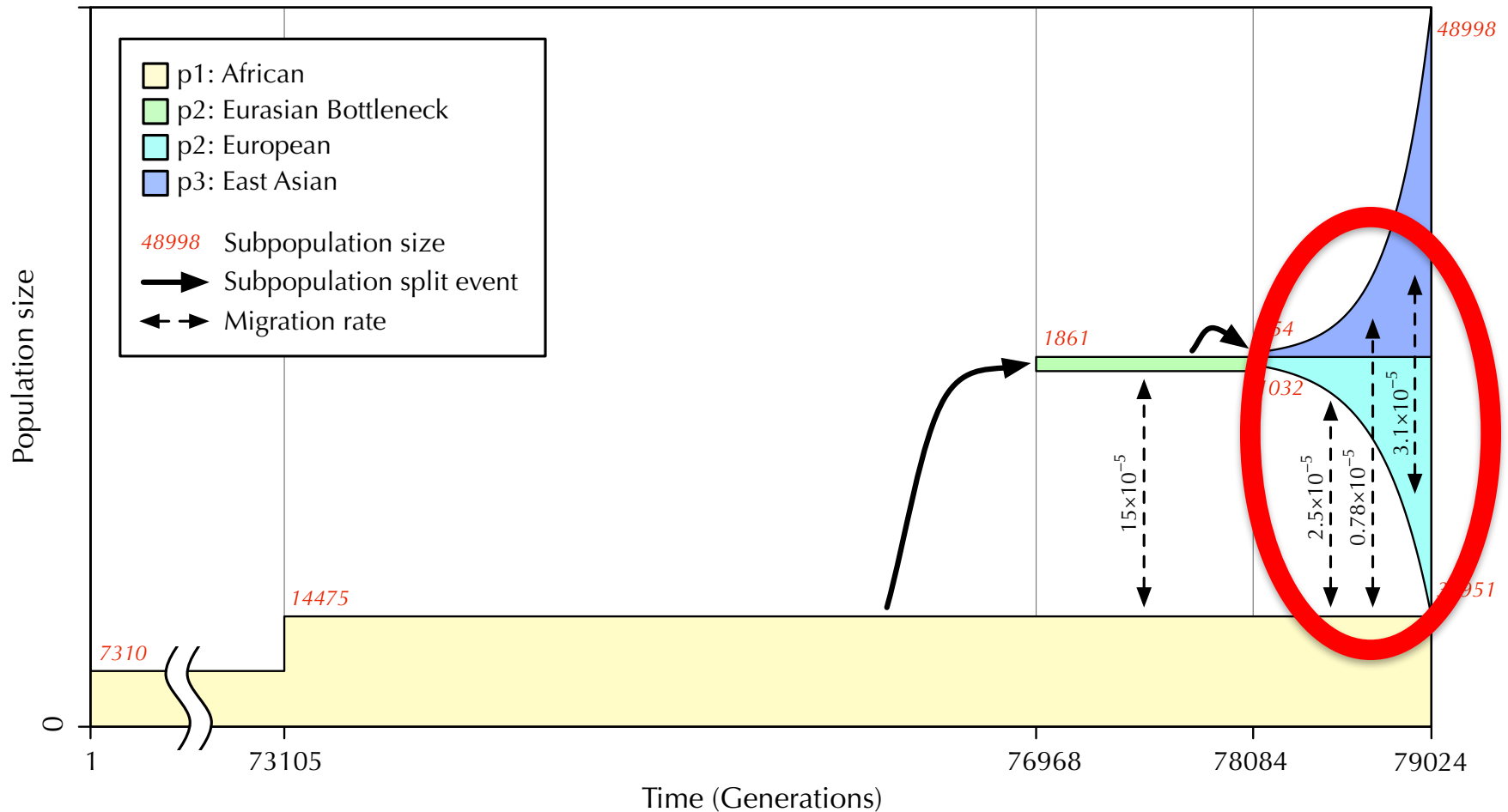
The Gravel Model



The Gravel Model



The Gravel Model



Mating Systems

- Demographic parameters in SLiM:
 - enabling separate sexes: `initializeSex()`
 - sex ratio: `setSexRatio()`
 - cloning rate: `setCloningRate()`
 - selfing rate: `setSelfingRate()`

Mating Systems

- Demographic parameters in SLiM:
 - enabling separate sexes: `initializeSex()`
 - sex ratio: `setSexRatio()`
 - cloning rate: `setCloningRate()`
 - selfing rate: `setSelfingRate()`
- nonWF models are more flexible:
 - monogamous mating
 - alternation of generations
 - haplodiploidy

The WF Generation Cycle

The sequence of events within one generation in WF models.

1. Execution of `early()` events

2. Generation of offspring:

2.1. Choose source subpop

2.2. Choose parent 1

2.3. Choose parent 2
(`mateChoice()` callbacks)

2.4. Generate the offspring
(including `mutation()` and
`recombination()` callbacks)

2.5. Suppress/modify child
(`modifyChild()` callbacks)

3. Removal of fixed mutations

4. Offspring become parents

5. Execution of `late()` events

6. Fitness value recalculation
using `fitness()` callbacks

7. Generation count increment

- The generation cycle
 - repeated by SLiM each generation
 - `early()` events
 - offspring generation
 - `late()` events
 - fitness recalculation

The WF Generation Cycle

The sequence of events within one generation in WF models.

1. Execution of `early()` events

2. Generation of offspring:

2.1. Choose source subpop

2.2. Choose parent 1

2.3. Choose parent 2
(`mateChoice()` callbacks)

2.4. Generate the offspring
(including `mutation()` and
`recombination()` callbacks)

2.5. Suppress/modify child
(`modifyChild()` callbacks)

3. Removal of fixed mutations

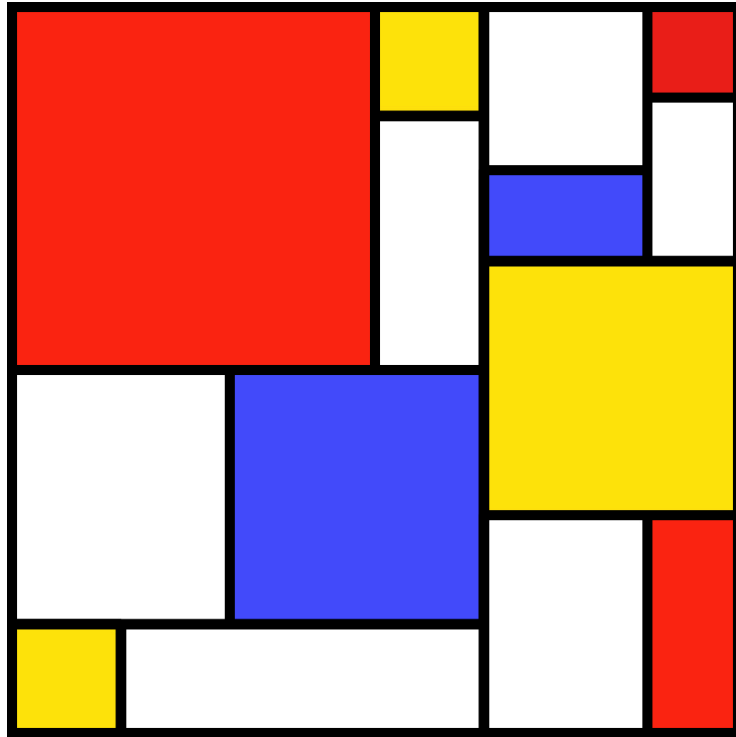
4. Offspring become parents

5. Execution of `late()` events

6. Fitness value recalculation
using `fitness()` callbacks

7. Generation count increment

- The generation cycle
 - repeated by SLiM each generation
 - `early()` events
 - offspring generation
 - `late()` events
 - fitness recalculation
- Future requests
 - many changes take effect the next time offspring are generated!



SLiM Workshop Exercise #5