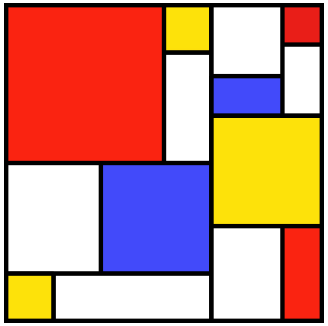
SLiM

Workshop Series



#10: Callbacks I (`fitness()`)

Benjamin C. Haller

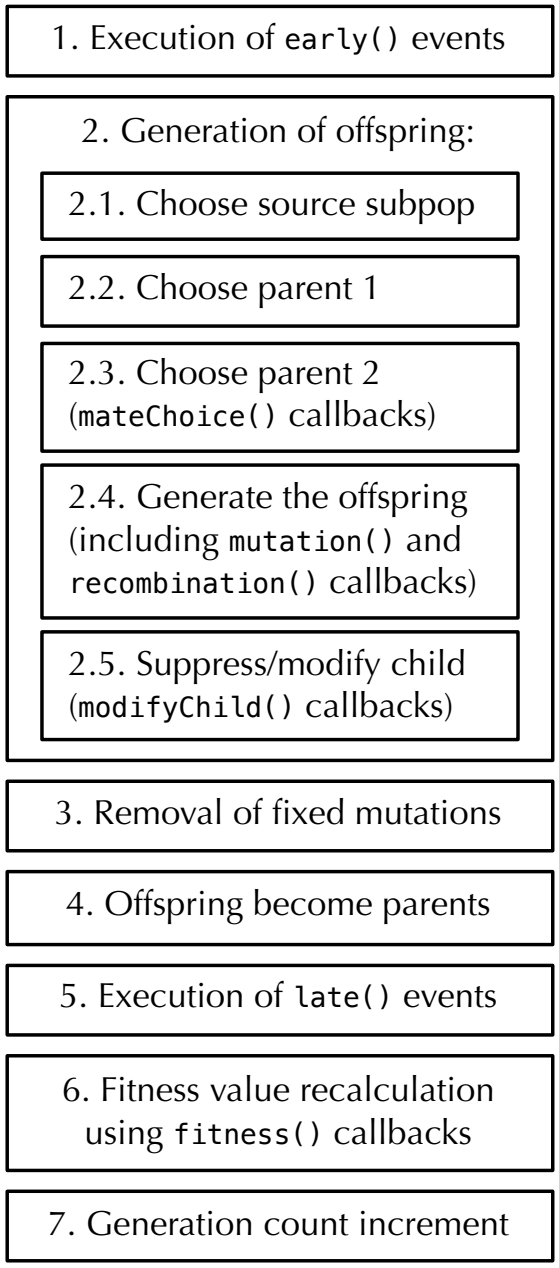
Messer Lab, Cornell University

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

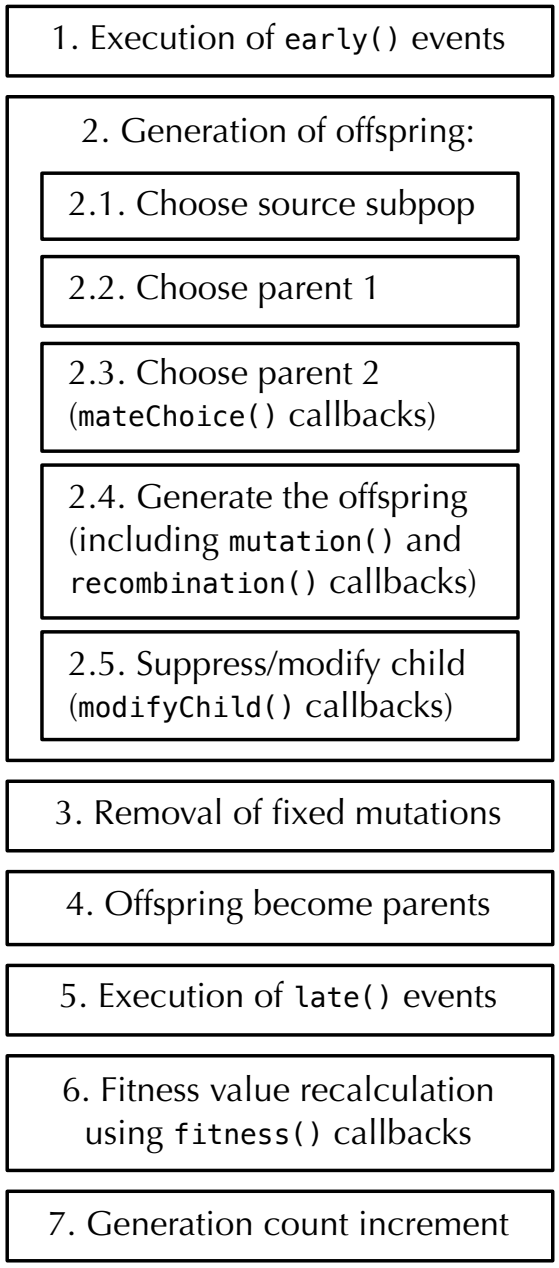
The sequence of events within one generation in WF models.



The WF Generation Cycle

- a set of standard behaviors:
  - generation start
  - mate choice / reproduction
  - parental mortality
  - fitness value recalculation
  - generation end

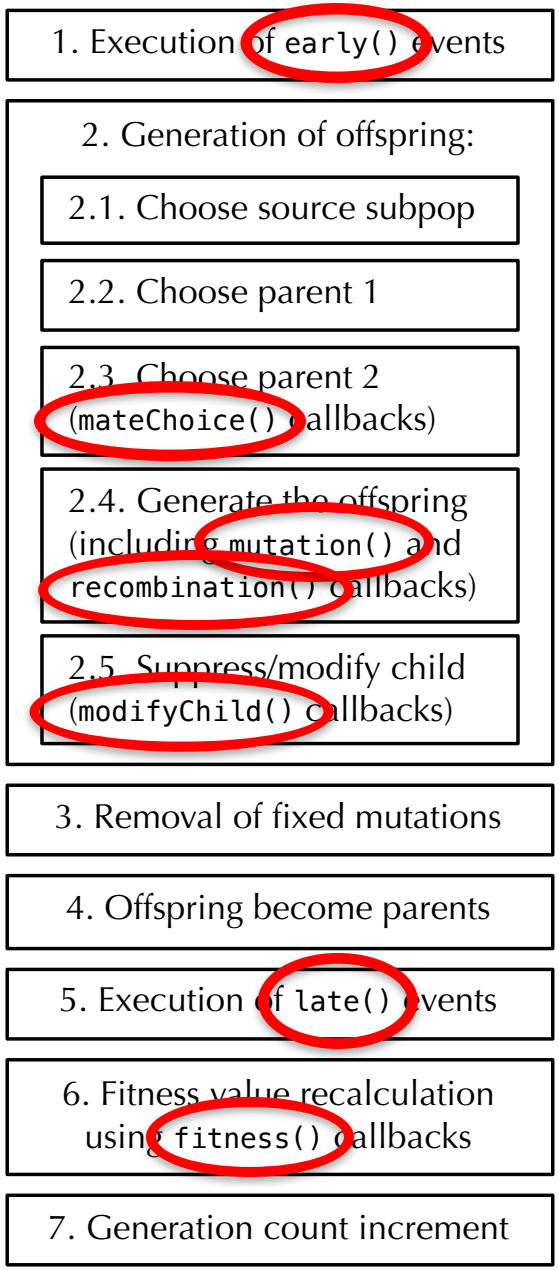
The sequence of events within one generation in WF models.



The WF Generation Cycle

- a set of standard behaviors:
  - generation start
  - mate choice / reproduction
  - parental mortality
  - fitness value recalculation
  - generation end
- modified by *callbacks*
  - script providing new behavior

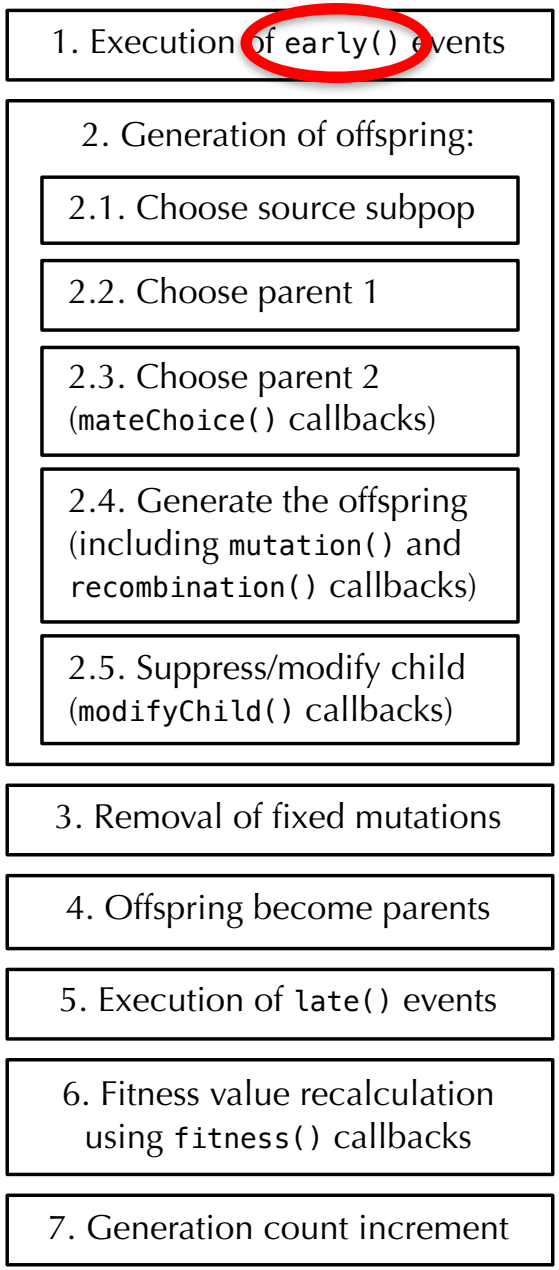
The sequence of events within one generation in WF models.



The WF Generation Cycle

- Callbacks:
  - `early()` events
  - `mateChoice()` callbacks
  - `mutation()` callbacks
  - `recombination()` callbacks
  - `modifyChild()` callbacks
  - `late()` events
  - `fitness()` callbacks
  - `interaction()`, `reproduction()`

The sequence of events within one generation in WF models.

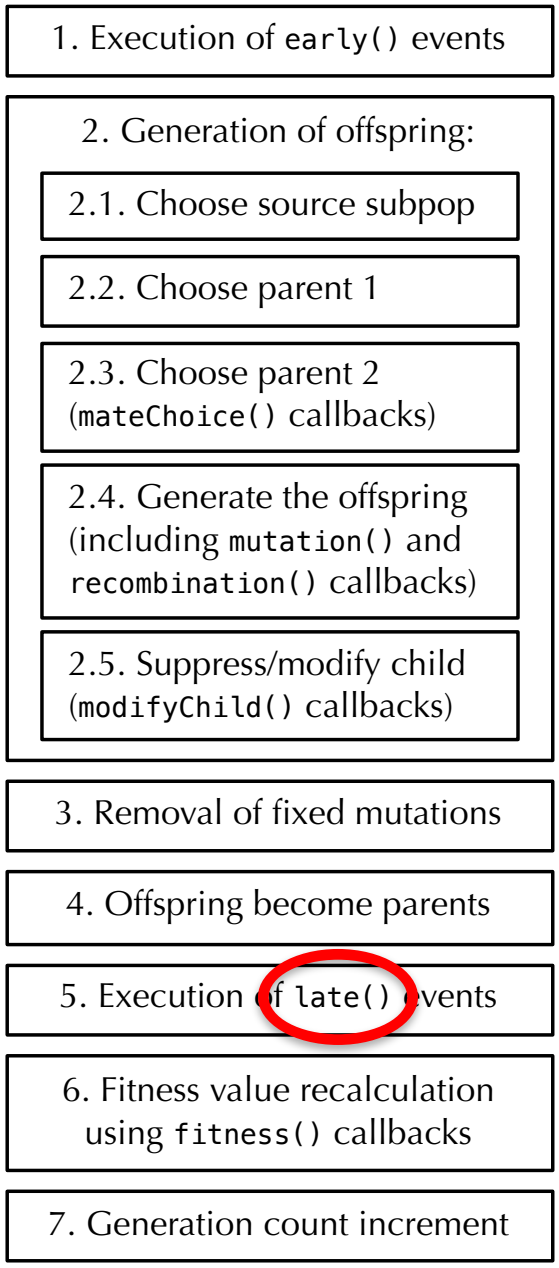


`early()` events

```
early()
{                    {
   ...                  ...
}                    }
```

- called *early* in each generation
- often used to:
  - set up demographic parameters
  - prepare state for mating
  - make a plan for the generation

The sequence of events within one generation in WF models.

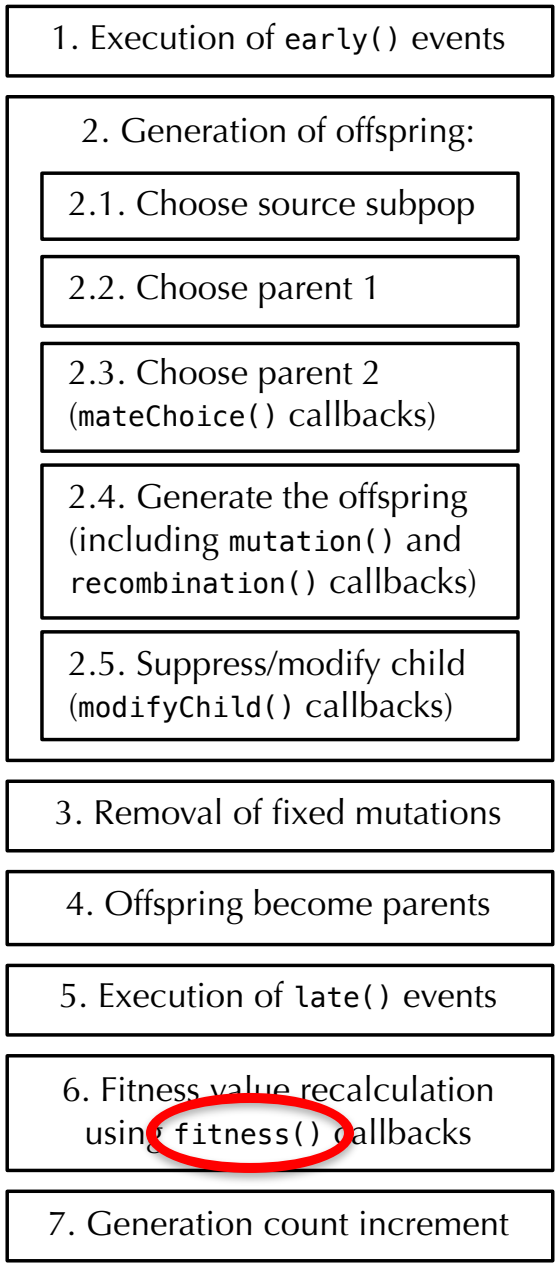


late() events

```
late()
{
   ...
}
```

- called *late* in each generation
- often used to:
  - modify the offspring state
  - run behavioral interactions
  - prepare for fitness calculations
  - produce custom output

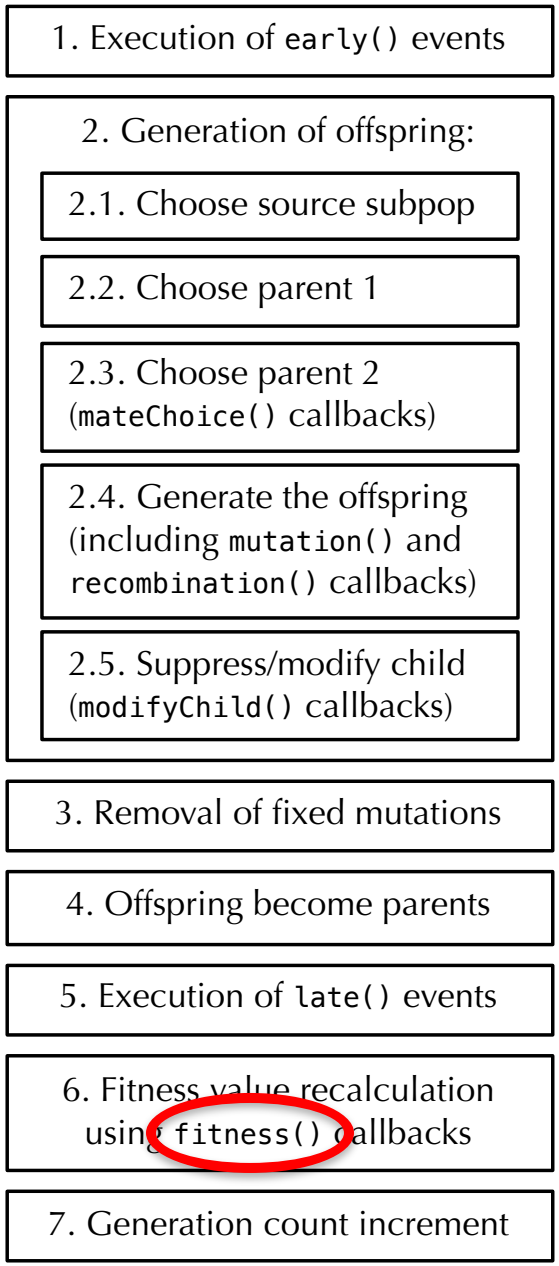
The sequence of events within one generation in WF models.



`fitness()` callbacks

- called during fitness calculation
- override default fitness
- often used to:
  - produce spatial variation
  - produce temporal variation
  - produce epistatic interactions
  - produce frequency dependence
  - make a mutation type neutral
- `fitness(NULL)` callbacks

The sequence of events within one generation in WF models.

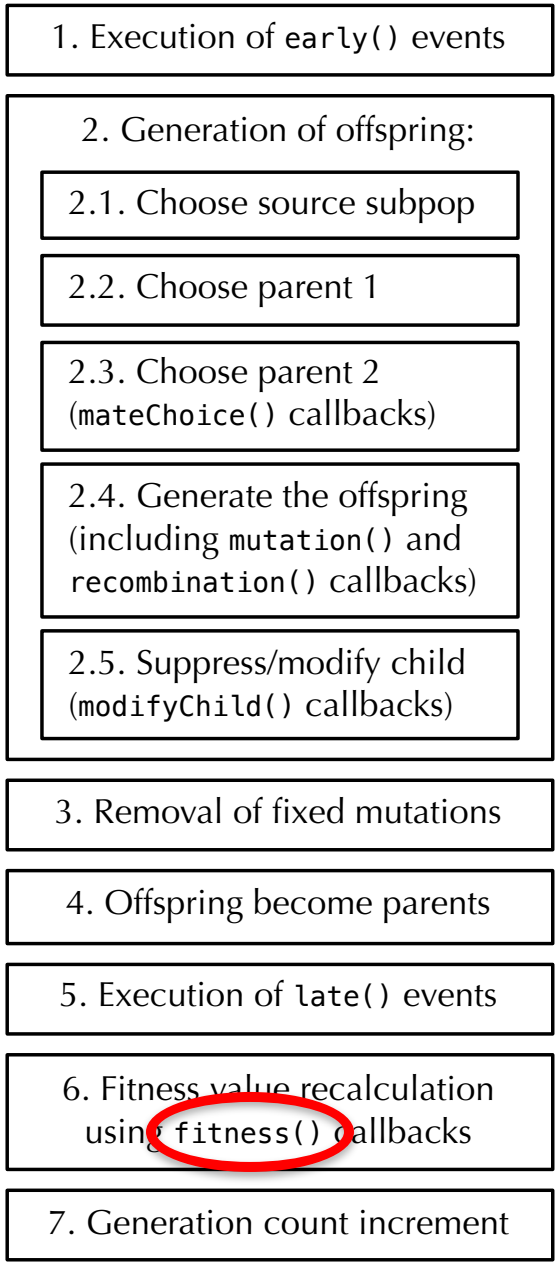


`fitness()` callbacks

```
fitness(<mutTypeID> [, <subpopID>])
{
   ...
}
```

- apply to *one* mutation type
- apply to *one or all* subpops
- called once for:
  - each **focal mutation**,
  - in each **focal individual**
  - allows individual effects
- return a **fitness effect**

The sequence of events within one generation in WF models.



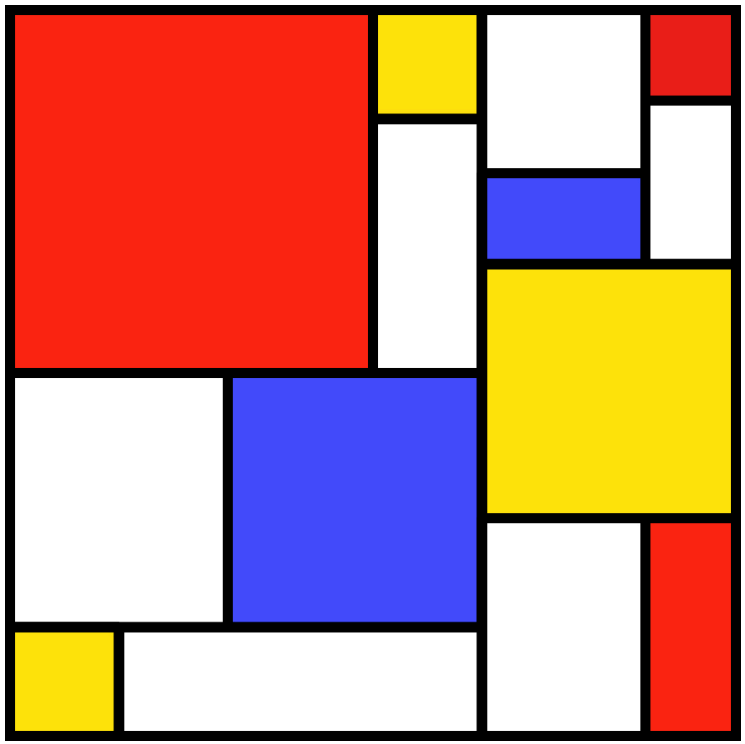
`fitness()` callbacks

- there is a *focal mutation*
- it is in a *focal individual*
- how does the callback know?
- **pseudo-parameters:**
  - `mut`
  - `homozygous`
  - `individual`
  - `genome1`
  - `genome2`
  - `subpop`
  - `relFitness`

Fitness calculation

- Fitness values in SLiM:
  - evaluated per individual, per mutation
  - `homozygous`: `T` if homozygous, `F` if heterozygous
  - `relFitness` has the default fitness value:

```
fitness(m1)
{
   if (homozygous)
      return 1.0 + mut.selectionCoeff;
   else
      return 1.0 + mut.mutationType.dominanceCoeff *
                        mut.selectionCoeff;
}
```



SLiM Workshop Exercise #10

Copyright © 2020–2021 Benjamin C. Haller. All Rights Reserved.