Using *ipmr* output

ipmr further analyses

```
my_ipm <- make_ipm(my_ipm,
                   iterations = 100)
```

```
lambda_ipmr <- lambda(my_ipm)
repro_value <- left_ev(my_ipm)
stable_dist <- right_ev(my_ipm)
```

Other options

```
# For stochastic IPMs
mean_kernel()

# Extract for calculation
# manual or other packages
my_ipm$sub_kernels

my_ipm$pop_state
```

General IPMs

What are general IPMs

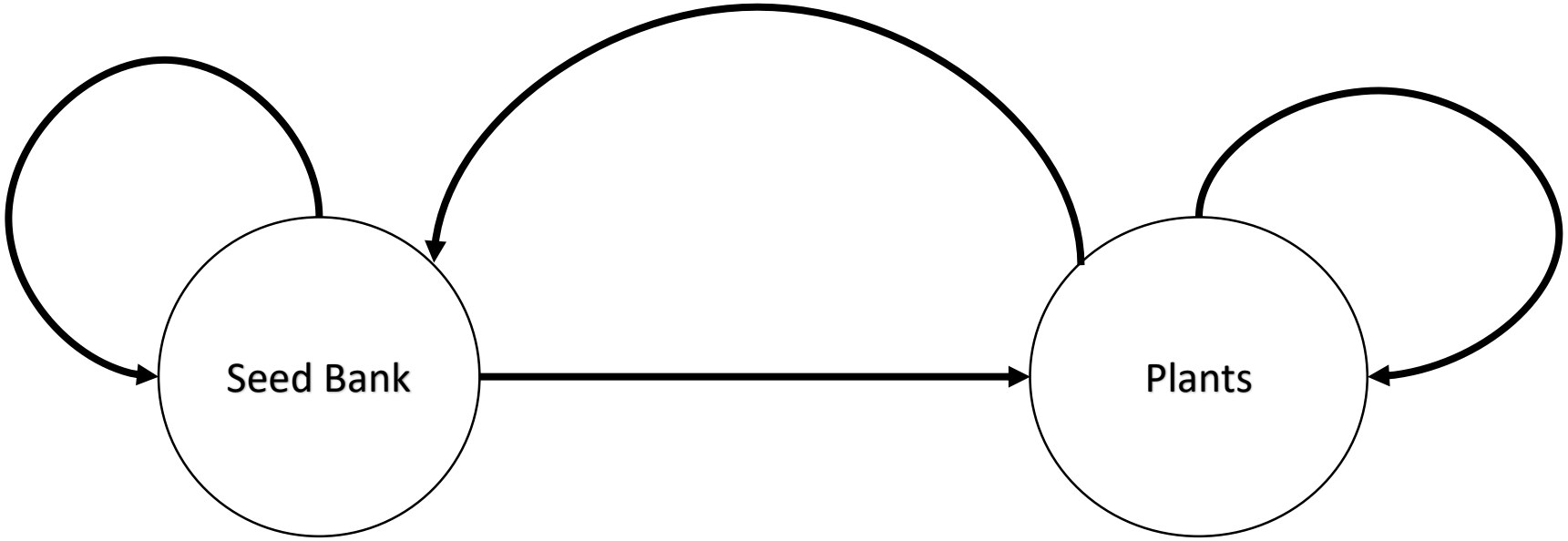
Incorporate ≥ 2 continuous state variables

and/or

include transitions between continuous and discrete states

- Height (C) and diameter at breast height (C) of trees
- Developmental stage(s) (D) and body size (C) of animals

Example: discrete + continuous state variable

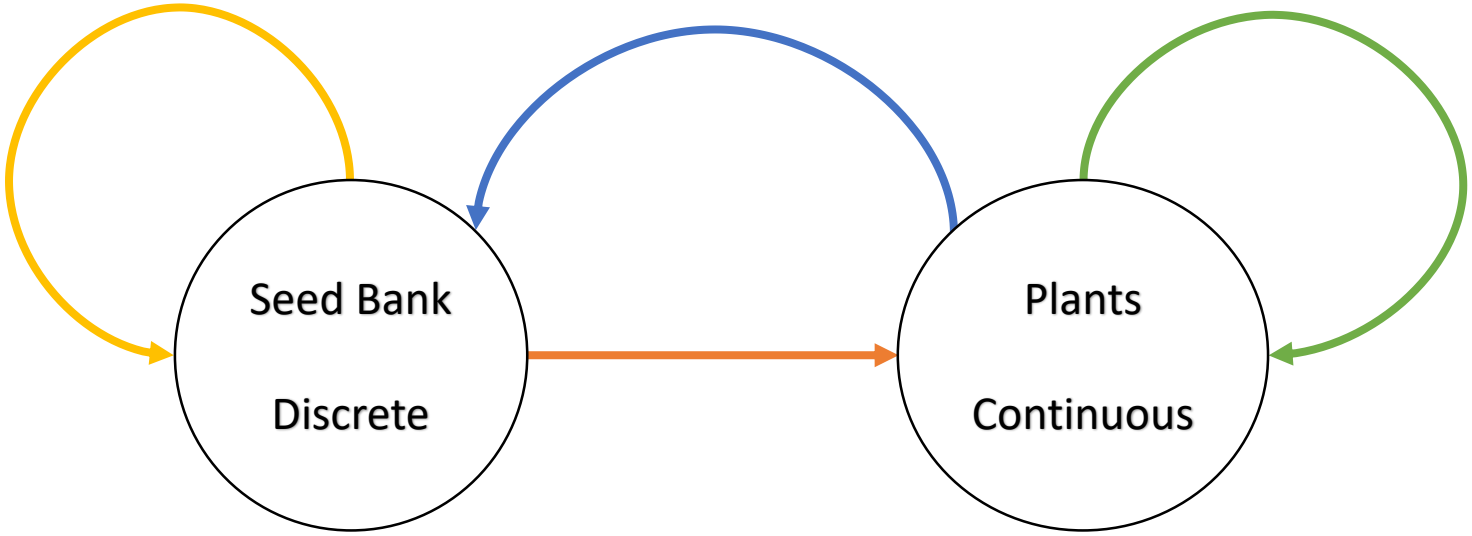


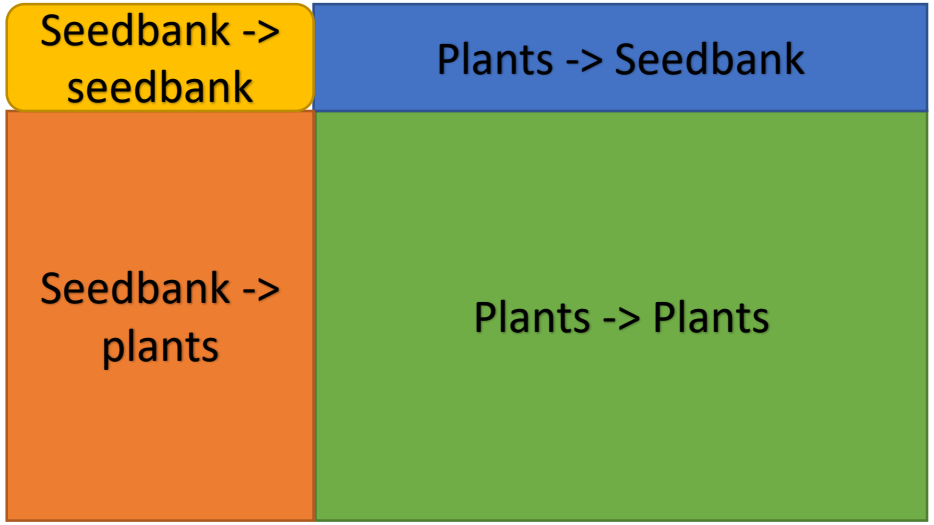
The discrete “addition”

Closely resembles the simple IPM in our previous exercise

Example: discrete + continuous state variable

- $n(x', T+1) = p_{germ} * size_{at\ germ} * B(T) + \int_L^U [P(x', x)] n(x, T) dx$
- $B(T+1) = (1 - p_{germ}) * B(T) + \int_L^U [F] n(x, T) dx$





Changes & additions for general IPMs with *ipmr*

General ipm changes

```
my_ipm <- init_ipm(sim_gen = "simple",
                   di_dd = "di",
                   det_stoch = "det")
```

```
my_ipm <- init_ipm(sim_gen = "general",
                   di_dd = "di",
                   det_stoch = "det")
```

General ipm changes

```
my_ipm <- define_kernel(
                        family     = "CC",
                        ...,
                        states     = list(c('dbh')),
)
```

```
my_ipm <- define_kernel(
                        family     = "CD", # Or "CC", "DD", "DC"
                        ...,
                        states     = list(c('dbh', 'seedlings')),
)
```

General ipm changes

```
my_ipm <- define_kernel(
  family     = "CC",

  formula    = s * g,
  ...,
)
```

```
my_ipm <- define_kernel(
  family     = "CC",

  formula    = s * g * d_dbh,
)
```

General ipm changes

```
my_ipm <- define_impl(
  proto_ipm = my_ipm,
  kernel_impl_list = list( ...,
    F = list(..., state_start = "dbh", state_end   = "dbh")
  )
)
```

```
my_ipm <- define_impl(
  proto_ipm = my_ipm,
  kernel_impl_list = list(...,
    F = list(..., state_start = "dbh", state_end   = "seedling")
  )
)
```

General ipm changes

```
my_ipm <- define_pop_state(my_ipm,
                           n_dbh = rep(1/200, 200))
```

```
my_ipm <- define_pop_state(my_ipm,
                           n_dbh = rep(1/200, 200),
                           n_seedling = 25)
```