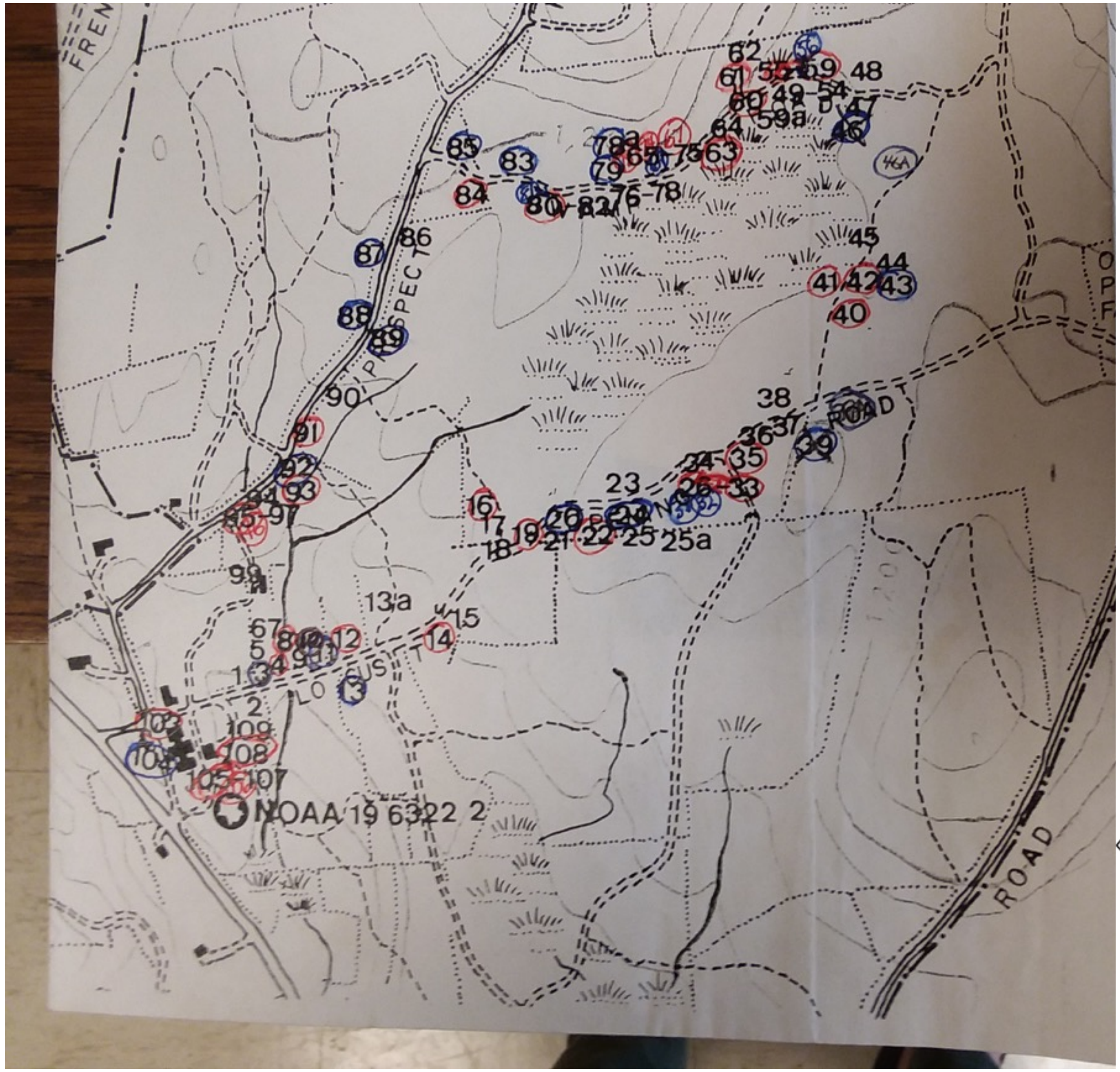
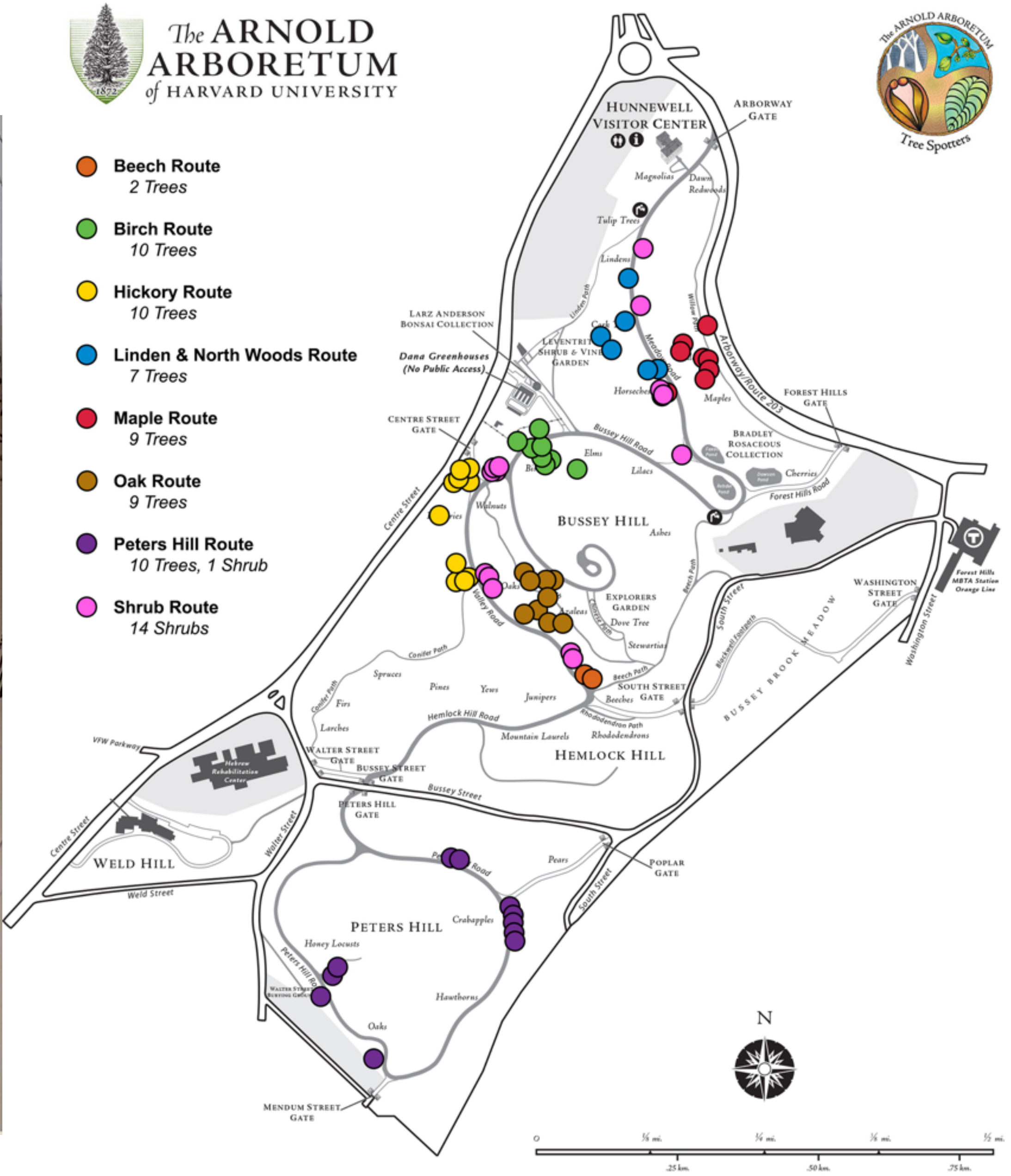
HARVARD FOREST

ARNOLD ARBORETUM


NOAA 19 6322 2
ROAD


The ARNOLD ARBORETUM of HARVARD UNIVERSITY
Beech Route
2 Trees
Birch Route
10 Trees
Hickory Route
10 Trees
Linden & North Woods Route
7 Trees
Maple Route
9 Trees
Oak Route
9 Trees
Peters Hill Route
10 Trees, 1 Shrub
Shrub Route
14 Shrubs
The Arnold Arboretum Tree Spotters
Hunnewell Visitor Center
Arborway Gate
Bussey Hill
Hemlock Hill
Peters Hill
Weld Hill
Forest Hills Gate
Centre Street Gate
South Street Gate
Walter Street Gate
Bussey Street Gate
Peters Hill Gate
Poplar Gate
Washington Street Gate
Mendum Street Gate
N

Main Aims:

1. How does the method used for measuring daily weather influence GDDs until budburst?
   - Where does the error fall?
   - Is there actually less interspecific variation when using Weather Station data as we'd expect?
   - Is anything else influencing our results, like provenance latitude?

2. How does sampling frequency influence GDDs until budburst?

3. Build a PMM in stan

Just using Urban predictor

$$y_i = \alpha_{species[i]} + \beta_{urban_{species[i]}} X + \epsilon_i$$

$$\epsilon_i \sim N(0, \sigma_y)$$

$$\alpha_{species} \sim N(\mu_\alpha, \sigma_\alpha)$$

$$\beta_{urbanspecies} \sim N(\mu_{urban}, \sigma_{urban})$$

Adding in provenance

$$y_i = \alpha_{species[i]} + \beta_{urban_{species[i]}} U + \beta_{provenance_{species[i]}} P + \epsilon_i$$

$$\epsilon_i \sim N(0, \sigma_y)$$

$$\alpha_{species} \sim N(\mu_\alpha, \sigma_\alpha)$$

$$\beta_{urbanspecies} \sim N(\mu_{urban}, \sigma_{urban})$$

$$\beta_{provenancespecies} \sim N(\mu_{provenance}, \sigma_{provenance})$$

Main Aim 1: Flagging system

```
use.sims = TRUE
use.hobo = FALSE ### We expect less species variation using weather station data, so i
use.urban = TRUE
use.provenance = TRUE
use.highsitevariation = FALSE ## Not sure if I will use these but here just in case
use.highprovvariation = FALSE

#check.diags = TRUE ## Do you want to check diagnostics?
#save.stan = TRUE  ## Do you want to save your model?

if(use.urban==FALSE & use.highsitevariation==TRUE){
  print("Error was made in flags!! Adjust accordingly!")
}

if(use.provenance==FALSE & use.highprovvariation==TRUE){
  print("Error was made in flags!! Adjust accordingly!")
}


########################################################################
if (use.sims==TRUE & use.hobo==FALSE & use.urban==TRUE & use.provenance==FALSE &
    use.highsitevariation==FALSE & use.highprovvariation==FALSE){
```

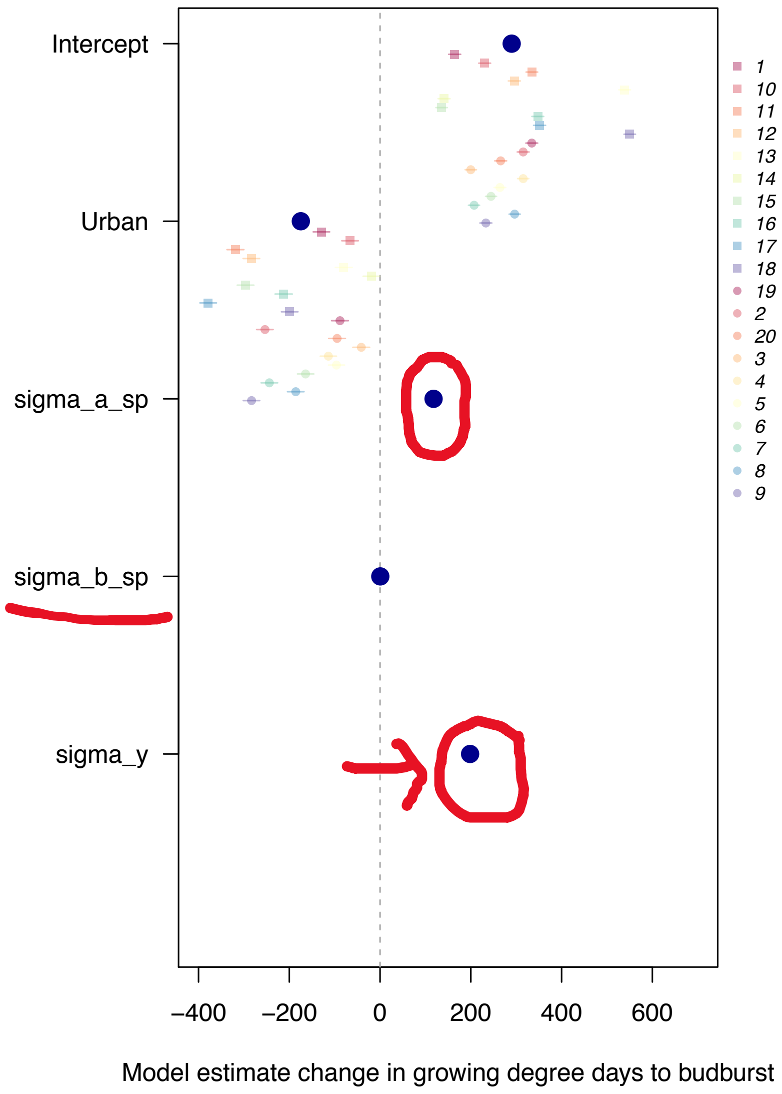
Fast-forward…

1. Many divergent transitions with real data using just urban predictor (0 = forest; 1 = urban)

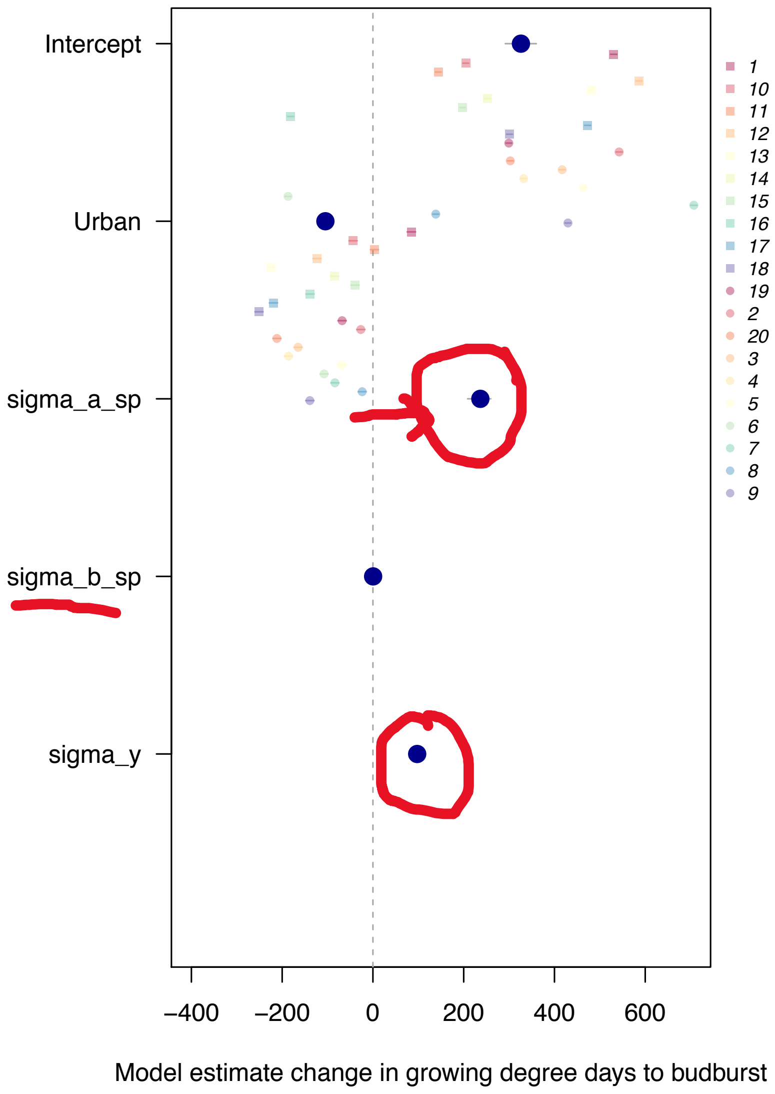
2. Build new model in rstan using ncp – works!! Results to follow

3. Provenance won't fit with real data in rstan or with ncp. Need to build a vcov in rstan as a next step.

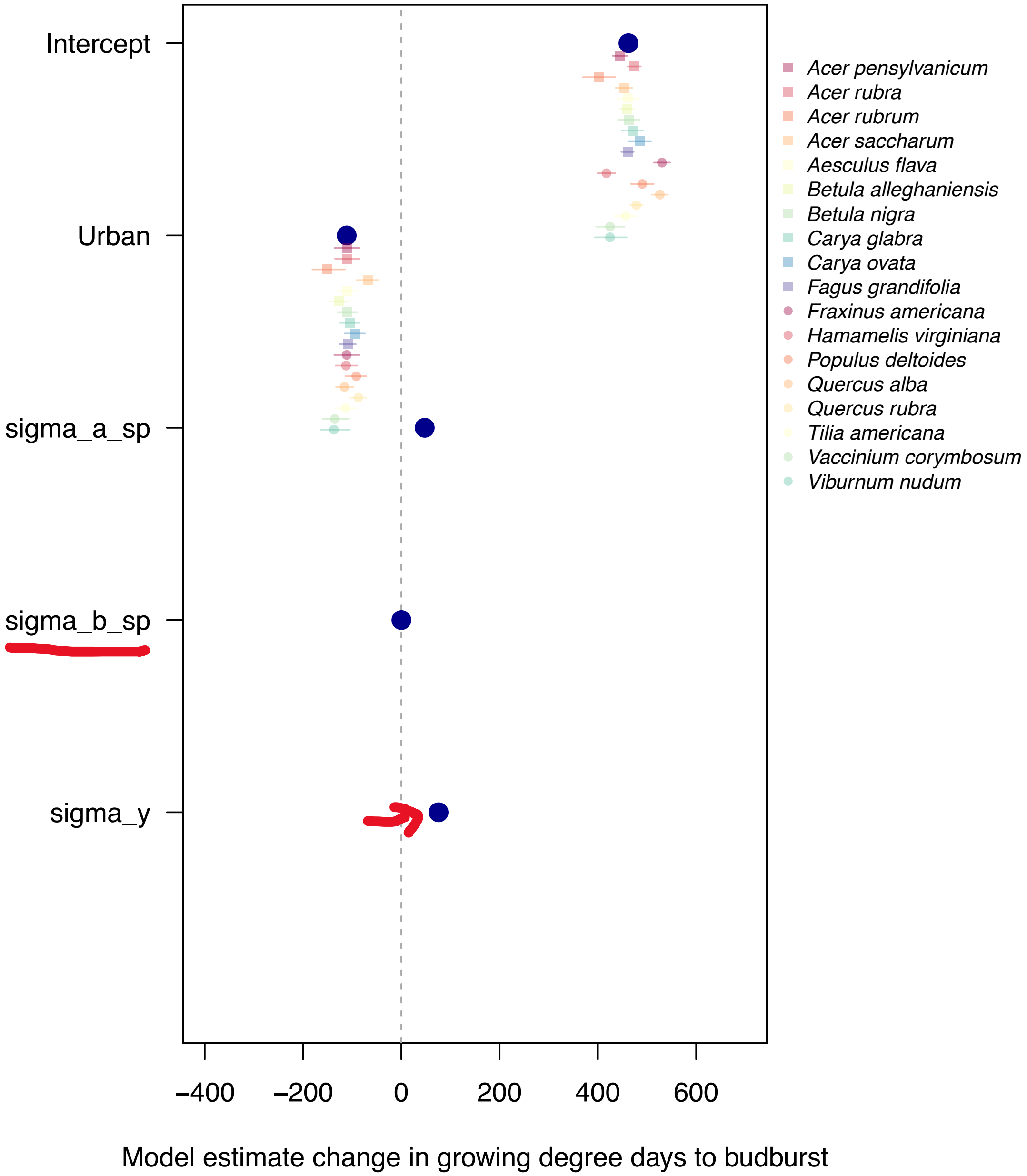
Weather Station Data: less interspecific variation



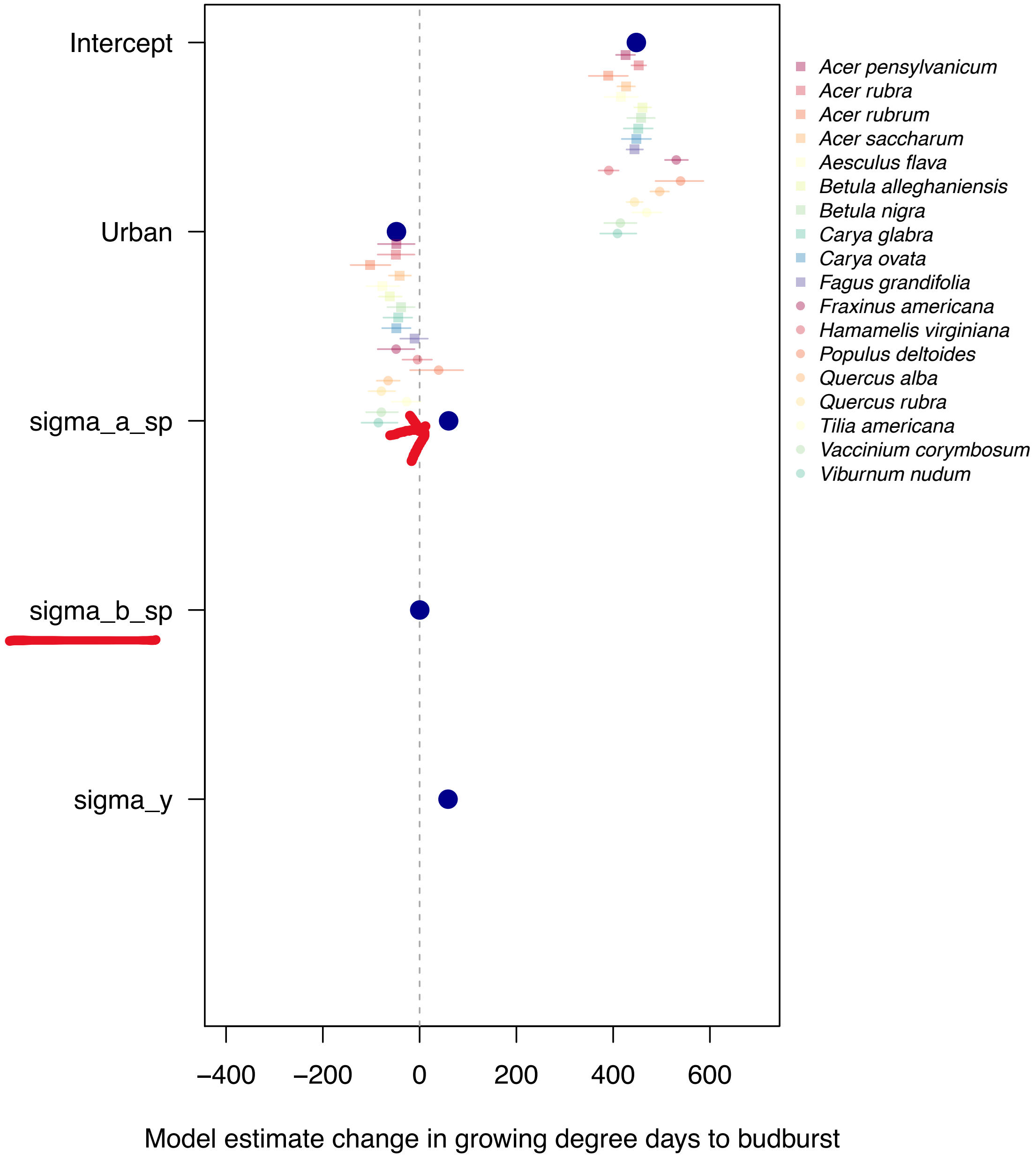
Hobo Logger Data: more interspecific variation



Weather Station Data: real data



Hobo Logger Data: real data



Weather Station Data: less interspecific variation

```
> ws_urb_fake.sum[grep("mu_", rownames(
                  mean    se_mean       sd
mu_a_sp      290.0472 0.1852628 26.8287
mu_b_tx_sp  -174.5789 0.5063822 26.3186
> ws_urb_fake.sum[grep("sigma_", rownam
                   mean      se_mean
sigma_a_sp     117.9317642 0.159600524
sigma_b_tx_sp    0.5705779 0.001788124
sigma_y        198.4758864 0.014090791
```

Hobo Logger Data: more interspecific variation

```
> hl_urb_fake.sum[grep("mu_", rownames(
                  mean    se_mean       sd
mu_a_sp      326.1430 0.4102013 51.7673
mu_b_tx_sp  -104.7482 0.4893669 21.5324
> hl_urb_fake.sum[grep("sigma_", rownam
                   mean      se_mean
sigma_a_sp     236.7172261 0.325133390
sigma_b_tx_sp    0.4757195 0.001654222
sigma_y         97.5189362 0.008039312
```

Weather Station Data: real data

```
> ws_urb.sum[grep("mu_", rownames(ws_u
                  mean    se_mean       s
mu_a_sp      462.2213 0.2827039 17.309
mu_b_tx_sp  -111.0530 0.3356388 19.549
> ws_urb.sum[grep("sigma_", rownames(w
                   mean      se_mean
sigma_a_sp     47.6937006 0.313856974
sigma_b_tx_sp   0.1897722 0.002496276
sigma_y        75.8962972 0.040104918
>
```

Hobo Logger Data: real data

```
> hl_urb.sum[grep("mu_", rownames(hl_u
                   mean    se_mean       s
mu_a_sp      447.96428 0.3309690 21.3523
mu_b_tx_sp   -47.81398 0.4442283 26.2192
> hl_urb.sum[grep("sigma_", rownames(h
                   mean      se_mean
sigma_a_sp     60.0082139 0.576286819
sigma_b_tx_sp   0.2943233 0.003882313
sigma_y        58.7665758 0.125103985
```