Contrasting between- and within-rows stem profiles in poplar plantations - Modelling tree and 'Treat' effect - Piecewise approach

Puletti N., Scotti R., ???

june 2018

Contents

Intro

Analysis of differences (or ratios) between orthogonal tree profiles implies a nested design:
measurements from the same tree are not 'independent' observations!
Differences (and ratios) are also dependent on 'section height'.

Tree level (level 1) modelling

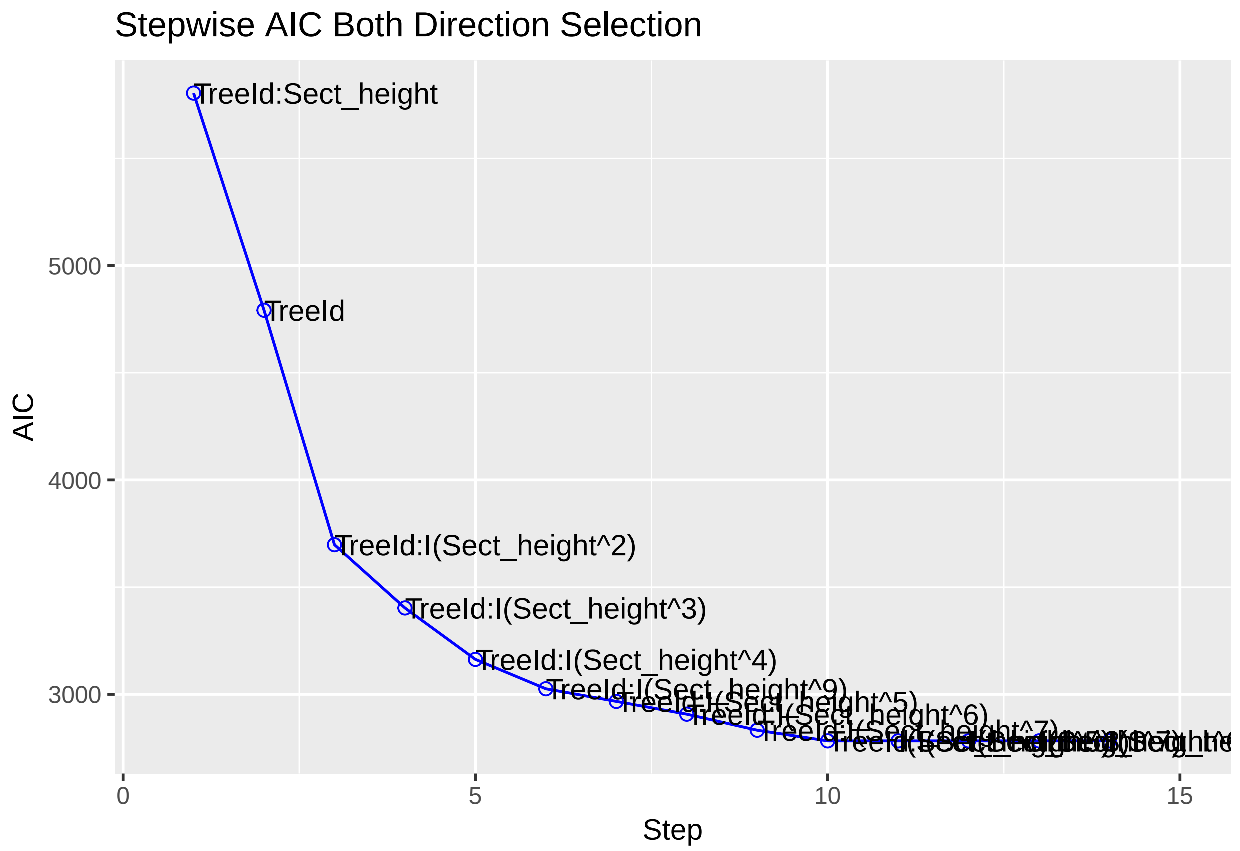
Here we focus on the most convenient way to model that relation.
The graph is generally 'U' shaped, hence a straight line is particularly inefficient as a model.
Since the 'U' is not symmetrical, a polynomial of order grather than 2 could be required.
Selection of model variables is performed adopting a stepwise procedure.

```
source("DataWrangling.R")

fm4 <- lm(data=stff, delta_d_cm ~ poly(Sect_height,4) * TreeId)

# First we try the classical 'p' based (or 'F' based) stepwise procedure (pent = 0.1, prem = 0.3)

#   ols_step_both_p(fm4)

# Not working!! it doesn't recognize the distinct powers as variables!

## Stepwise Selection Method
## ---------------------------
## Candidate Terms:
## 1. poly(Sect_height, 4)
## 2. TreeId
## 3. poly(Sect_height, 4):TreeId
## We are selecting variables based on p value...

# Limited to power 2, ------------------------\|/
```

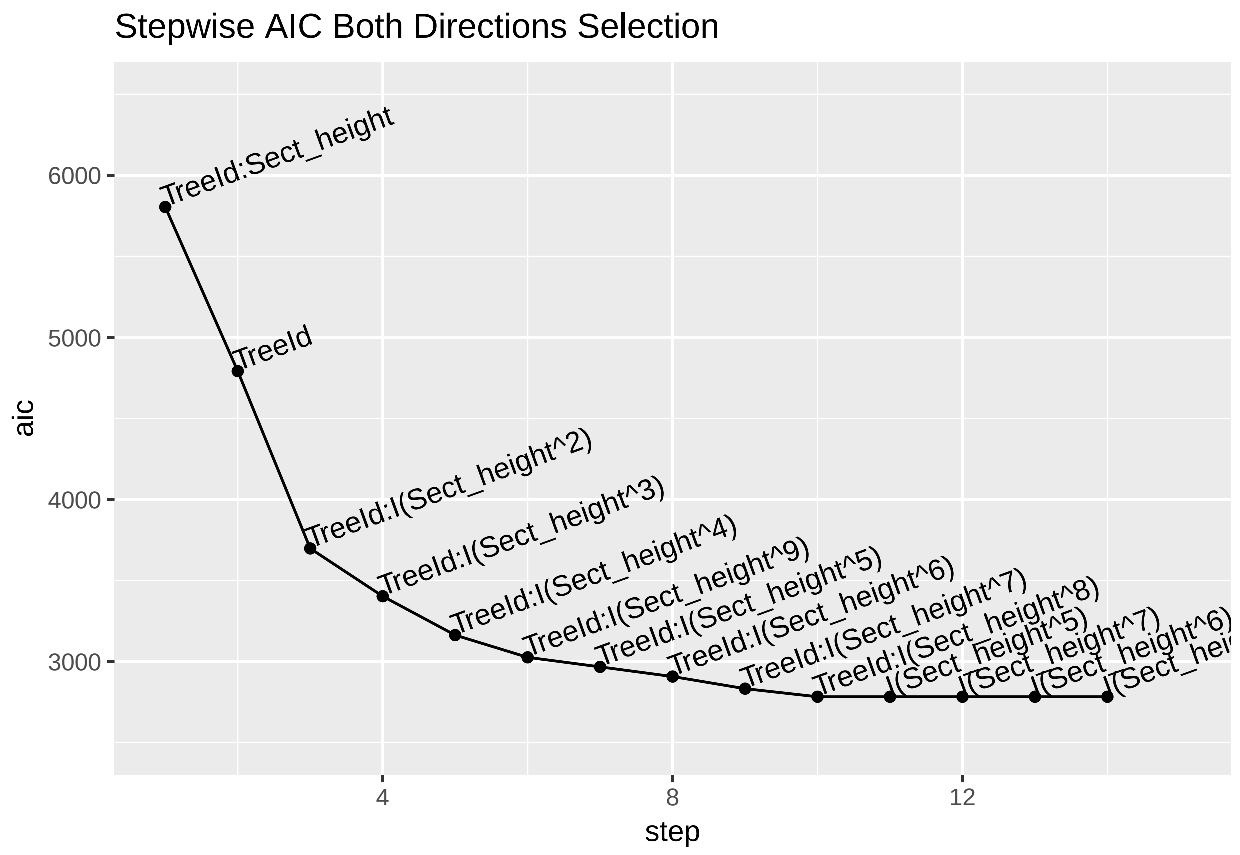
```r
fp <- lm(data=stff, delta_d_cm ~ TreeId
        +  Sect_height     * TreeId
        + I(Sect_height^2) * TreeId
        + I(Sect_height^3) * TreeId )  # with this "ols_step_both_p" crashes
# (ols_step_both_p(fp)) %>% plot()

# Swithching to AIC based selection

ft <- lm(data=stff, delta_d_cm ~ TreeId
        +  Sect_height     * TreeId
        + I(Sect_height^2) * TreeId
        + I(Sect_height^3) * TreeId
        + I(Sect_height^4) * TreeId
        + I(Sect_height^5) * TreeId
        + I(Sect_height^6) * TreeId
        + I(Sect_height^7) * TreeId
        + I(Sect_height^8) * TreeId
        + I(Sect_height^9) * TreeId
        )
ol <- ols_step_both_aic(ft)
```

```
## Stepwise Selection Method
## -------------------------
##
## Candidate Terms:
##
## 1 . TreeId
## 2 . Sect_height
## 3 . I(Sect_height^2)
## 4 . I(Sect_height^3)
## 5 . I(Sect_height^4)
## 6 . I(Sect_height^5)
## 7 . I(Sect_height^6)
## 8 . I(Sect_height^7)
## 9 . I(Sect_height^8)
## 10 . I(Sect_height^9)
## 11 . TreeId:Sect_height
## 12 . TreeId:I(Sect_height^2)
## 13 . TreeId:I(Sect_height^3)
## 14 . TreeId:I(Sect_height^4)
## 15 . TreeId:I(Sect_height^5)
## 16 . TreeId:I(Sect_height^6)
## 17 . TreeId:I(Sect_height^7)
## 18 . TreeId:I(Sect_height^8)
## 19 . TreeId:I(Sect_height^9)
##
##
## Variables Entered/Removed:
##
## <U+2714> TreeId:Sect_height
## <U+2714> TreeId
## <U+2714> TreeId:I(Sect_height^2)
## <U+2714> TreeId:I(Sect_height^3)
## <U+2714> TreeId:I(Sect_height^4)
## <U+2714> TreeId:I(Sect_height^9)
## <U+2714> TreeId:I(Sect_height^5)
## <U+2714> TreeId:I(Sect_height^6)
## <U+2714> TreeId:I(Sect_height^7)
## <U+2714> TreeId:I(Sect_height^8)
## <U+2714> I(Sect_height^5)
## <U+2714> I(Sect_height^7)
## <U+2714> I(Sect_height^6)
## <U+2714> I(Sect_height^4)
##
## No more variables to be added or removed.
```

```
plot(ol)  # labels
```



```
ol[-3] %>%
  as.data.frame() %>%
  ggplot(aes(x = 1:ol[[3]], y = aic, label = predictors)) +
    ggtitle("Stepwise AIC Both Directions Selection") +
    geom_line() +
    geom_point() +
    geom_text(angle = 20, hjust = 0, vjust = 0) + xlab("step") +
    xlim(c(1, ol[[3]] + 1)) +
    ylim(c(pretty(ol$aic)[1], pretty(max(ol$aic) * 1.05)[2]))
```

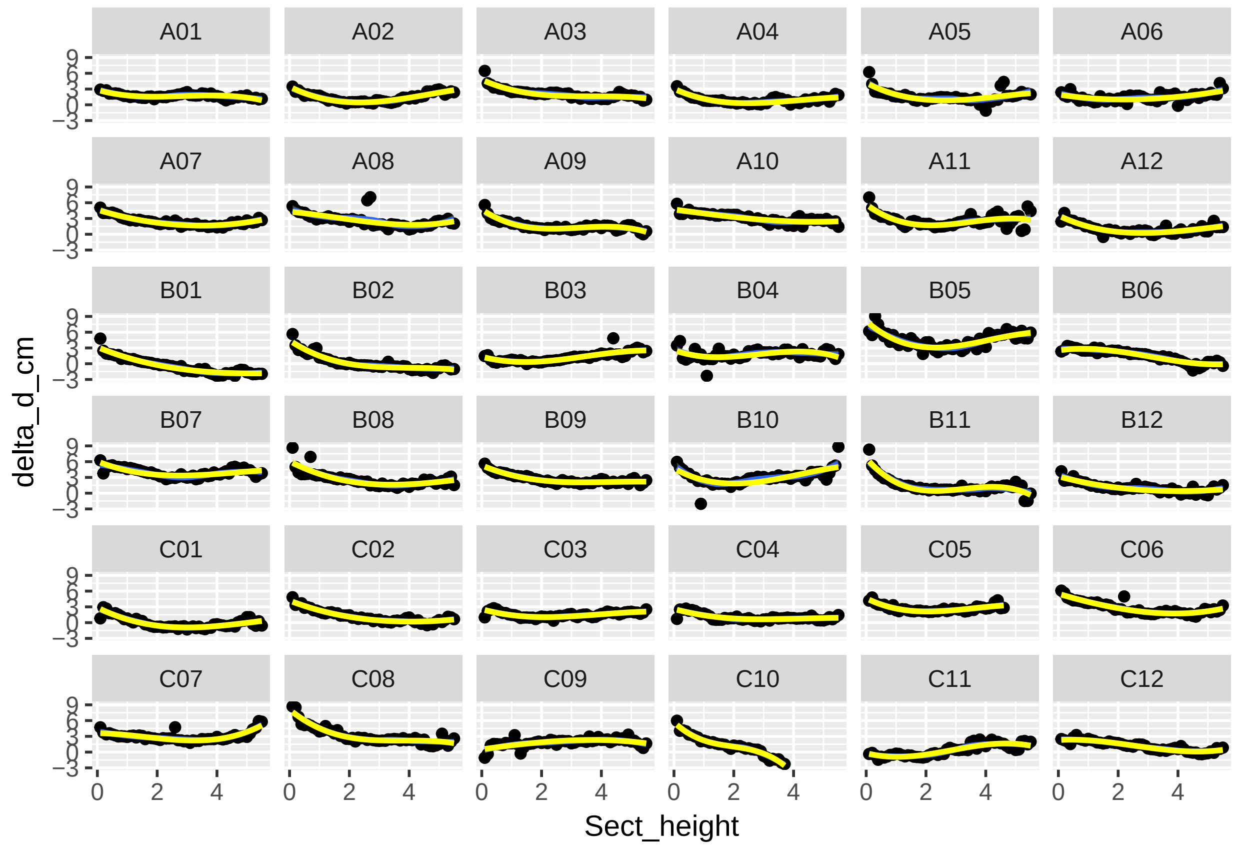


```
opt_m_f <- paste0("delta_d_cm ~ ", paste(ol$predictors[1:4], collapse = " + "))
```

Above power 4 (INCLUDED, in ver2), it is alwais the highest power available that is selected. This behaviour is probably due to local irregularities that the model should not consider. (Even limiting the model to power 3, some cases diplay singular shapes) Best model is hence the following:
* **delta_d_cm ~ TreeId:Sect_height + TreeId + TreeId:I(Sect_height^2) + TreeId:I(Sect_height^3)**

```
opt_m <- lm(data=stff, opt_m_f)

stff %>% ggplot(aes(x = Sect_height, y = delta_d_cm)) +
  geom_point() +
  geom_smooth() +
  geom_line(aes(x = Sect_height, y = fitted(opt_m)), color = "yellow", size = 1) +
  facet_wrap(~TreeId)
```

```
## `geom_smooth()` using method = 'loess'
```



```r
# Actually even power 4 could be too high, some bendings are in excess

# Insted of wrangling with opt_m estimates,
# it is more straightforward to recompute power 4 poly coefficients estimates for each tree

l1p <- stff %>%
  group_by(TreeId) %>%
  do(fit = tidy(lm(delta_d_cm ~ poly(Sect_height, 3), .))) %>%
  unnest() %>%
  select(1:3) %>%
  inner_join(unique(stff[, c("Treat", "TreeId")]))
```

```
## Joining, by = "TreeId"
```

Level 2 modelling: evaluation of the effects of ‘Treat’ on level 1 (tree level) parameters

```r
l1p %>%
  spread(term, estimate) %>%
  select(3:6) %>%
  cor()
```

```
##                         (Intercept) poly(Sect_height, 3)1
## (Intercept)              1.00000000            0.12793276
## poly(Sect_height, 3)1  0.12793276            1.00000000
## poly(Sect_height, 3)2  0.28445284           -0.07472778
## poly(Sect_height, 3)3  0.01491582           -0.10804421
##                       poly(Sect_height, 3)2 poly(Sect_height, 3)3
## (Intercept)                      0.28445284            0.01491582
## poly(Sect_height, 3)1           -0.07472778           -0.10804421
## poly(Sect_height, 3)2            1.00000000           -0.27103579
## poly(Sect_height, 3)3           -0.27103579            1.00000000
```

```
# evaluations disregarding coefficients standard deviation

# separate evaluations, one for each delta_d=f(Sect_height) function coefficient
l1p %>%
  group_by(term) %>%
  do(fit = tidy(anova(lm(estimate ~ Treat, .)))) %>%
  unnest()
```

```
## # A tibble: 8 x 7
##   term                  term1        df   sumsq meansq statistic p.value
##   <chr>                 <chr>     <int>   <dbl>  <dbl>     <dbl>   <dbl>
## 1 (Intercept)           Treat         2   0.592  0.296     0.241   0.788
## 2 (Intercept)           Residuals    33  40.6    1.23     NA      NA
## 3 poly(Sect_height, 3)1 Treat         2   6.54   3.27      0.170   0.845
## 4 poly(Sect_height, 3)1 Residuals    33 637     19.3      NA      NA
## 5 poly(Sect_height, 3)2 Treat         2   7.90   3.95      0.812   0.453
## 6 poly(Sect_height, 3)2 Residuals    33 161      4.86     NA      NA
## 7 poly(Sect_height, 3)3 Treat         2   4.03   2.01      0.957   0.394
## 8 poly(Sect_height, 3)3 Residuals    33  69.5    2.10     NA      NA
```

```
# p.values >> .1

# joint evaluation of all delta_d=f(Sect_height) function coefficients for each tree
l1p %>% lm(estimate ~ Treat * term, .) %>% anova()
```

```
## Analysis of Variance Table
##
## Response: estimate
##             Df Sum Sq Mean Sq F value Pr(>F)
## Treat        2   2.60   1.302  0.1894 0.8277
## term         3 843.94 281.312 40.9186 <2e-16 ***
## Treat:term   6  16.46   2.743  0.3989 0.8786
## Residuals  132 907.49   6.875
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
f0 <- lmer(estimate ~     1 + (1 | term), l1p)
f1 <- lmer(estimate ~ Treat + (1 | term), l1p)
anova(f0, f1)
```

```
## refitting model(s) with ML (instead of REML)

## Data: l1p
## Models:
## f0: estimate ~ 1 + (1 | term)
## f1: estimate ~ Treat + (1 | term)
##    Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## f0  3 700.64 709.55 -347.32   694.64
## f1  5 704.24 719.09 -347.12   694.24 0.3941      2     0.8211
```

```
# No effect! Pr(>Chisq) = 0.6836 !!
```

[rmarkdown::render("PiecewiseEvaluation2.Rmd", encoding="UTF-8")]