Image analysis of Festuca seeds

Joan M. Barreto O.

4/20/2020

Methods

Image analysis was used to explore a potential relationship between seed size and ploidy level in Festuca species. Seeds from each accessions were randomly selected and imaged with an Epson Perfection V6 flatbed scanner at 600 dpi (dots per inch) with size 1987 (width) by 2169 (length) pixels. The TIF images (Tagged Image File Format) were processed with a custom Python script (**https://github.com/joanmanbar/ Seed_Morphology**) that transformed the original images to threshold and measure the seed length and width respectively as the length (in pixels) of the major and minor axes of a fitted ellipse, whereas the area is calculated as the number of pixels in the seed. The script returned a dataset containing the image name, the seed number and their dimensional attributes. The data analysis was performed in *R* (R Core Team, 2014) by fitting liner models with seed area, length, and width as response and the ploidy level as predictor. To remove potential artifacts, we filtered out the seeds that were not within the 45th and 55th quantile. Finally, we used the *ggplot2* package (Wickham, 2009) for plots and figures.

Preparing the data

```r
# Get working directory
mydir <- getwd()

# Read data
mydata1 <- read.csv(paste0(mydir, '/Seeds_data_results.csv'))
ploidy <- read.csv(paste0(mydir, '/pic_list_with_ploidy.csv'))

# Substract pic number
mydata1$Image_Name <- as.character(mydata1$Image_Name)
mydata1$Image_Name <- substr(mydata1$Image_Name, 1,
                             nchar(mydata1$Image_Name)-4)
mydata1$Image_Name <- as.factor(mydata1$Image_Name)

ploidy$Image_Name <- as.character(ploidy$Image_Name)
ploidy$Image_Name <- substr(ploidy$Image_Name, 1,
                            nchar(ploidy$Image_Name)-4)
ploidy$Image_Name <- as.factor(ploidy$Image_Name)

# Merge files
mydata2 <- merge(ploidy, mydata1)
colnames(mydata2)[1] <- "Accession"

# Keep seed with area values around the median
mydata3 <- group_by(mydata2, Accession) %>%
  filter(Area > quantile(Area, 0.45)) %>% filter(Area < quantile(Area, 0.55))
mydata3 <- as.data.frame(mydata3)
```

```
# Count total number of seeds per accession
# Total_seeds <- mydata2 %>% group_by (Accession) %>% summarise(n = n())
Total_seeds <- mydata3 %>% group_by (Accession) %>% summarise(n = n())


# Visualize coefficient of variation per accession
Accessions <- group_by(mydata3, Ploidy, Accession)

Accessions <- summarise(Accessions,
                sampled_seeds = n(),
                mean_area = mean(Area, na.rm = TRUE),
                sd_area = sd(Area, na.rm = TRUE),
                cv_area = cv(Area))

kable(Accessions, caption = "Coefficient of Variation")
```

Table 1: Coefficient of Variation

Ploidy	Accession	sampled_seeds	mean_area	sd_area	cv_area
2X	PI_109497	18	565.2222	83.41149	14.757292
2X	PI_115358	30	615.9667	62.15691	10.090954
2X	PI_189146	22	498.0909	57.45052	11.534144
2X	PI_206268	15	523.0667	34.50355	6.596397
2X	PI_206561	38	668.7105	52.23332	7.811050
2X	PI_234478	23	728.6522	274.50014	37.672315
2X	PI_234897	36	769.6389	89.71229	11.656413
2X	PI_237179	13	380.3077	59.10497	15.541355
2X	PI_249739	25	476.2800	81.20218	17.049252
2X	PI_250965	27	711.1111	41.99115	5.905005
2X	PI_251384	21	1397.9524	260.68630	18.647724
2X	PI_251421	19	683.4737	66.04407	9.663001
2X	PI_257740	20	605.2500	26.85218	4.436543
2X	PI_257741	28	679.4643	110.61461	16.279679
2X	PI_314571	15	553.4000	98.10111	17.726981
2X	PI_314687	16	667.0000	132.01212	19.791922
2X	PI_383651	27	449.2222	70.80707	15.762147
2X	PI_383652	25	511.2800	43.94269	8.594643
2X	PI_422463	14	655.0000	54.90832	8.382949
2X	PI_499640	22	742.4091	72.23484	9.729789
2X	PI_538934	19	447.1053	52.80772	11.811026
2X	PI_568183	16	522.1875	89.85931	17.208246
2X	PI_578733	18	604.3333	64.62380	10.693403
2X	PI_595130	11	542.0000	98.08262	18.096424
2X	PI_595140	20	453.1500	121.39465	26.789065
2X	PI_595145	15	678.4000	55.15407	8.130022
2X	PI_595146	15	641.8667	83.58389	13.022002
2X	PI_595158	11	523.7273	109.95189	20.994112
2X	PI_595167	16	541.6875	102.81032	18.979637
2X	PI_595170	25	550.0800	93.94277	17.078019
2X	PI_595178	19	486.1053	115.25040	23.708939
2X	PI_618972	22	626.4545	66.78232	10.660362
2X	PI_618975	21	649.2381	94.28728	14.522758
2X	PI_634303	18	625.3889	90.75056	14.511060

Ploidy	Accession	sampled_seeds	mean_area	sd_area	cv_area
2X	PI_634304	17	470.2941	77.09553	16.393046
2X	PI_636567	26	694.5385	49.77649	7.166844
2X	PI_655206	25	725.8000	76.03234	10.475660
2X	PI_676177	13	645.7692	65.38623	10.125324
4X	PI_227362	20	945.5000	50.45842	5.336692
4X	PI_227507	18	1056.2778	85.33243	8.078598
4X	PI_229453	16	1082.5000	77.83658	7.190446
4X	PI_229497	18	1074.2222	42.88638	3.992319
4X	PI_229502	16	1012.3750	167.57242	16.552406
4X	PI_234758	20	1209.3000	66.74074	5.518956
4X	PI_234895	18	794.9444	140.98705	17.735460
4X	PI_237708	16	765.3125	65.42754	8.549127
4X	PI_251123	20	664.8500	64.81656	9.749051
4X	PI_251125	22	938.9545	70.65306	7.524652
4X	PI_251127	20	922.7000	88.25209	9.564549
4X	PI_251128	16	978.6250	55.56123	5.677479
4X	PI_251131	10	703.5000	239.12352	33.990549
4X	PI_287541	14	860.7857	99.71203	11.583839
4X	PI_302898	10	949.1000	101.53429	10.697954
4X	PI_302900	16	612.8125	209.10578	34.122311
4X	PI_312453	16	1272.7500	98.88074	7.769062
4X	PI_316249	20	692.8000	211.79822	30.571336
4X	PI_340103	15	749.1333	60.91665	8.131616
4X	PI_383650	20	685.7000	137.87336	20.106950
4X	PI_383653	24	718.6250	80.58688	11.214037
4X	PI_383654	15	547.4000	126.66198	23.138835
4X	PI_384860	16	800.5000	250.27718	31.265107
4X	PI_577098	18	644.1667	100.98296	15.676527
4X	PI_577099	20	675.3000	124.38061	18.418571
4X	PI_595049	22	867.0455	67.17956	7.748101
4X	PI_595050	17	867.0000	55.64396	6.417988
4X	PI_595051	14	940.1429	101.15324	10.759348
4X	PI_595059	14	806.8571	53.36645	6.614114
4X	PI_595060	23	740.0435	116.79449	15.782112
4X	PI_595061	20	779.4000	84.74072	10.872558
4X	PI_595062	19	824.3684	85.11770	10.325202
4X	PI_595063	19	866.7368	64.86554	7.483879
4X	PI_634302	24	651.2917	77.84794	11.952855
4X	QUATRO	15	686.3333	81.29722	11.845151
5X	PI_234751	22	1020.7273	62.81809	6.154248
5X	PI_251126	18	867.8889	108.96890	12.555628
5X	PI_274619	16	1310.9375	482.72897	36.823188
5X	PI_311405	16	877.8125	101.97269	11.616682
5X	PI_314522	16	895.1875	76.99023	8.600459
5X	PI_314523	15	1121.2000	103.32280	9.215376
5X	PI_315448	14	1011.0714	64.16381	6.346120
5X	PI_371896	10	1189.5000	70.12093	5.894992
5X	PI_380850	14	1031.2143	90.81018	8.806141
5X	PI_538933	15	1104.4000	95.76519	8.671241
6X	BEACON	13	992.7692	95.48574	9.618120
6X	PI_229456	19	1038.5263	103.93340	10.007777
6X	PI_229533	20	1203.5500	125.84890	10.456474

Ploidy	Accession	sampled_seeds	mean_area	sd_area	cv_area
6X	PI_230247	21	1288.3333	77.41985	6.009303
6X	PI_234896	22	1033.6364	191.97211	18.572499
6X	PI_234898	11	1371.4545	96.86316	7.062805
6X	PI_250967	22	1068.7727	74.89035	7.007136
6X	PI_251385	22	1030.5909	106.43785	10.327846
6X	PI_268234	22	1256.5000	115.35607	9.180746
6X	PI_380849	20	921.4500	62.20973	6.751286
6X	PI_380852	13	838.0769	115.58724	13.791961
6X	PI_380853	15	926.5333	55.37775	5.976876
6X	PI_380854	15	865.2667	52.24320	6.037815
6X	PI_380855	20	1019.9500	46.79347	4.587820
6X	PI_380857	25	974.0400	66.73298	6.851153
6X	PI_384861	14	1054.4286	62.98561	5.973435
6X	PI_537103	17	930.0588	118.10719	12.698895
8X	PI_234750	13	1616.3846	74.06027	4.581847
8X	PI_302899	17	1230.6471	63.30673	5.144183
8X	PI_311403	16	1090.5000	151.41334	13.884763
8X	PI_318989	46	1449.2174	134.89880	9.308389

Convert pixels to milimiters

Now we must convert seed dimensions in pixels to mm! We used this website (https://www.pixelcalculator.com/) to get the conversion factors from px to mm.

```
## Convert px to mm
# Original Width (or length) = 2169 px (92mm)
W_px <- 2169
W_mm <- 92
# Original Height = 1987 px (84mm)
H_px <- 1987
H_mm <- 84
# 600 dpi

px2mm <- 0.5 * ( (W_px/W_mm) + (H_px/H_mm) )    # Average of both dimensions

# Apply conversion factors
mydata3$Area <- mydata3$Area/ (W_mm*H_mm)
mydata3$Length <- mydata3$Length / px2mm
mydata3$Width <- mydata3$Width / px2mm
```
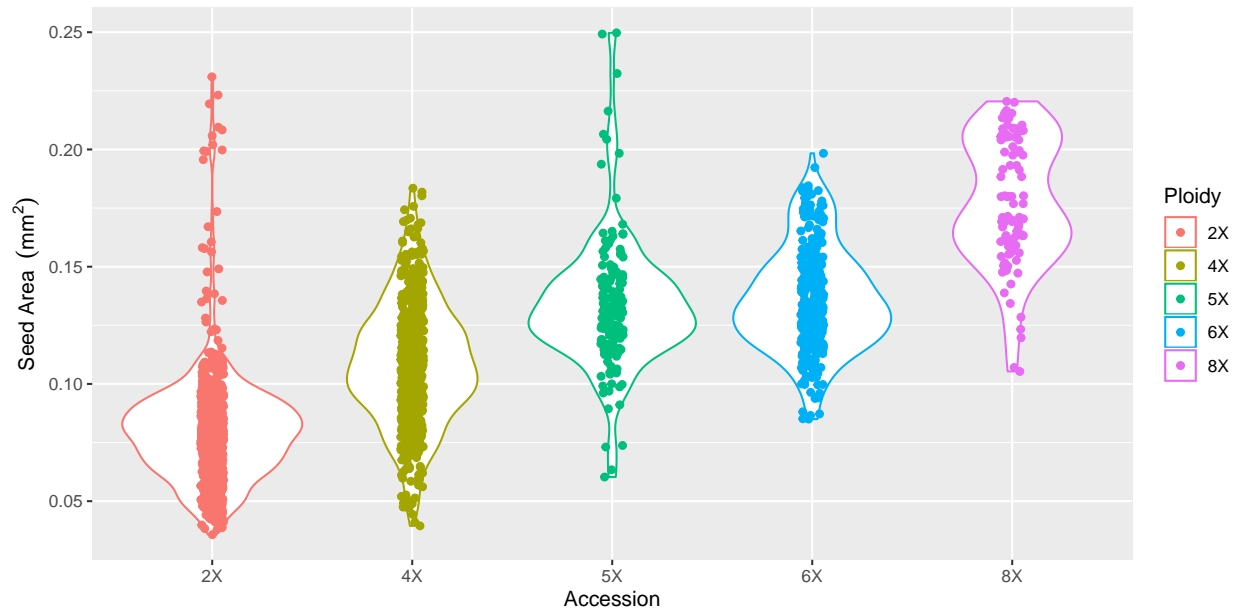
Data visualization and analyses of variance

Seed area

```
# Distribution of the seed area across accessions
vp <- ggplot(mydata3, aes(x=Ploidy, y=Area, color=Ploidy)) +
  geom_violin(position = position_dodge(width = 0.9)) +
  geom_point(position = position_jitterdodge(seed = 1, dodge.width = 0.9)) +
  labs(y= expression(paste("Seed Area  ", (mm^2))), x = "Accession")
```

vp


```r
# Model1 <- lm(Area ~ Ploidy + Accession, data = mydata3)
Model1 <- lm(Area ~ Ploidy, data = mydata3)
# summary(Model1)
# kable(anova(Model1))
kable(HSD.test(Model1, "Ploidy")$groups)
```

	Area	groups
8X	0.1772859	a
6X	0.1360331	b
5X	0.1337892	b
4X	0.1082374	c
2X	0.0807685	d
\		
\		

Seed length

```r
# Distribution of the seed length across accessions
vp <- ggplot(mydata3, aes(x=Ploidy, y=Length, color=Ploidy)) +
  geom_violin(position = position_dodge(width = 0.9)) +
  geom_point(position = position_jitterdodge(seed = 1, dodge.width = 0.9)) +
  labs(y= "Seed Length (mm)", x = "Accession")

vp
```
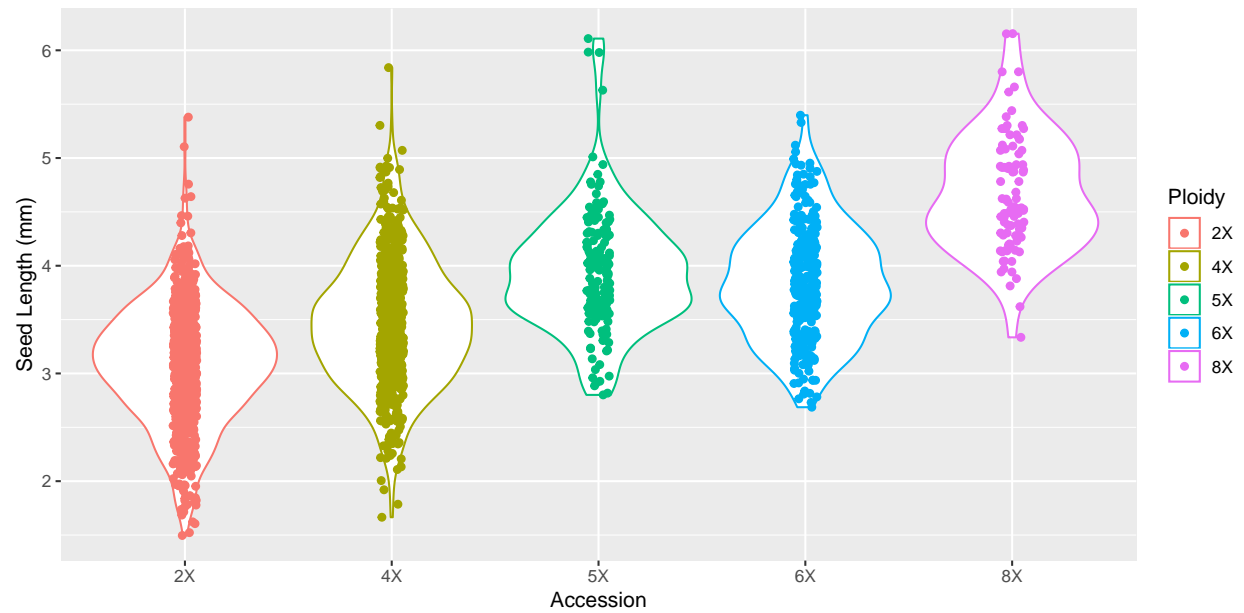
```
# Regress the seed length on ploidy level
Model2 <- lm(Length ~ Ploidy, data = mydata3)
# summary(Model2)
# kable(anova(Model2))
kable(HSD.test(Model2, "Ploidy")$groups)
```

	Length	groups
8X	4.676767	a
5X	3.940450	b
6X	3.844965	b
4X	3.463788	c
2X	3.087498	d

Seed width

```
# Distribution of the seed width across accessions
vp <- ggplot(mydata3, aes(x=Ploidy, y=Length, color=Ploidy)) +
  geom_violin(position = position_dodge(width = 0.9)) +
  geom_point(position = position_jitterdodge(seed = 1, dodge.width = 0.9)) +
  labs(y= "Seed Width (mm)", x = "Accession")

vp
```
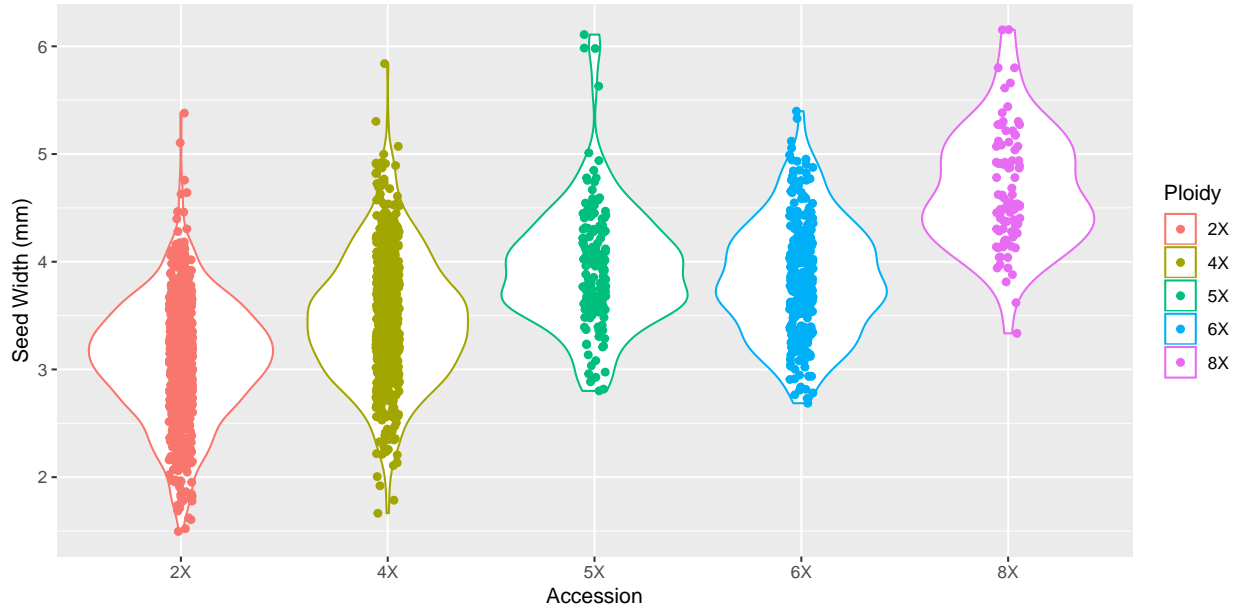
```r
# Regress the seed width using a Type II ANOVA (main effects)
Model3 <- lm(Width ~ Ploidy, data = mydata3)
# summary(Model3)
# kable(anova(Model3))
kable(HSD.test(Model3, "Ploidy")$groups)
```

	Width	groups
8X	0.7173019	a
5X	0.6815768	ab
6X	0.6788670	b
4X	0.6334165	c
2X	0.5317359	d

Results

Our results indicate that there are statistical differences in seed area, length and width between four different ploidy levels of Festuca at the $\alpha = 5$.

Notes

- Please read the methods.
- These results coincide with the previous results.
- Some accession has more variability than other, do you want to keep them all? Look at the plots and let me know if there is any accession you want to remove or include.
- The Github repository has been updated. I would only talk about the latest Python code in the methods and would not mentioned the Matlab one.
- Let me know if there are some questions or concerns.

References

R Core Team (2014). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/

Wickham, H. (2009) ggplot2: elegant graphics for data analysis. Springer New York.