Lumpkin Foundation - Illinois Bundleflower Final Report

Spencer Barriball, Bo Meyering, Brandon Schlautman

2025-03-25

Introduction and Summary

Period Covered by Grant

Please list the time period this grant covered (e.g. 5/20 - 4/21)

3/22 - 2/25

Summary of Work

Please provide a 3 to 5 sentence summary of your post grant report. Include project name, geographic area served, and project accomplishments. Response here.

[1,000 characters]

Outcomes

Please indicate which Outcomes you selected on your original application.

- Direct impact on climate, habitat, soil health, or mitigation of extreme weather
- Enhanced public understanding of why your activity is important
- Contribution to climate science that makes future land uses
- Model for communities/organizations how your work improves the lives of people and the environment

Additional Outcomes

- If previously listed on your application, add your additional outcome not listed above that applies to your request.*

NA

Metrics

Please indicated which Metrics you selected on your original application.

- Rate of carbon sequestration or expected carbon storage
- Media, community presentations or programs that spread the word on why your project is important
- Volunteers recruited and engaged in your activity
- Acres benefiting from your activity
- ~~Trees newly planted or saved~~
- Scientific papers, presentations, or contacts that add to climate science

Additional Metrics

If previously listed on your application, add your additional metric not listed above that applies to your request.

- Rate of nitrogen export

Updated Project Evaluation Table

Please upload your updated Project Evaluation Table to include the Actual data column. If your actual data missed the target data, please discuss this in your Project Evaluation Narrative.

Here's our Project Evaluation Table to be updated.

Materials and Methods

Accession Germination Test

Field Trial Design

Harvesting

Sample Processing

NIR Spectra Sampling

Plant tissues from harvet were ground and filtered using a 1mm sieve .... and scanned using the NeoSpectra NIR handheld scanner ... FILL THIS OUT

Wet Chemistry

A total of 75 samples out of the entire population were selected for targeted wet chemistry analysis. Samples were selected out of the entire population using the Kennard Stone algorithm to select samples with the greatest Euclidean distance from each other, ensuring that the samples reflected the availble NIR variance in the population. Samples were processed by xyz method, dried and sent to ABC labs for targeted wet chemistry analysis.. FILL THIS OUT

NIR Modeling

A total of ten wet chemistry metrics were selected for modeling. These included forage moisture content (%), forage dry matter content (%), ADF (units), NDF (units), relative feed value (units), total digestible nutrients (units), net energy gain (Mcal/cwt), net energy maintenance (Mcal/cwt), net energy lactation (Mcal/cwt), and forage yield (kg/Ha) all calculated on a dry matter basis. Spectra were presented as the percent reflectance per wavelength measured. Low quality spectra, i.e. low variance among the response values for all wavelengths, were filtered out of the dataset programatically. All spectra were then linearly resampled by wavelength to ensure equal wavelength intervals between each response value. The spectra were smoothed and detrended using a Savitsky-Golay filter by fitting a 5^{th} order polynomial over a window length of 15 frames and taking the second derivative. Standard mean subtraction scaling was applied per wavelength, but were not scaled by the colun variance. Training and testing populations (70% and 30%, respectively) were selected from the calibration samples by using the Kennard-Stone selection algorithm. A partial least squares regression (PLSR) model was fit to the data for each of the target variables, allowing the number of model components to vary from 1 to 16 total components. Root Mean Squared Error (RMSE) defined below was used to evaluate each models performance on the training and testing populations.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y_i})^2}{N}}$$

where y_i represents the ground truth target value of the i^{th} sample and $\hat{y_i}$ represents the model prediction for that sample. The model with the lowest testing population RMSE was selected as the final model.

Forage Quality Breeding BLUE and Repeatability

We modeled each of the forage quality trait as $ X $

Repeatabilty was then defined as the variance of the random effect 'Sample ID' divided by the total variance in the model

$$\frac{\sigma_G^2}{\sigma_G^2 + \frac{\sigma_{resid.}^2}{r}}$$

where r is equal to the total number of replicates in the experiment, σ_G^2 is the sample variance from the random accession effects, and $\sigma_{resid.}^2$ is the residual model variance.

Selection Index

For each field replicate plot of a given accession, we defined a selection index based on the importance of 3 traits: Forage yield, total digestible nutrients, and crude protein such that the selection index for sample i can be represented as:

$$SI_i = w \cdot x_i$$

where w is a column vector of weights that sum to 1, and x_i is a vector of traits from the i^{th} sample expressed as $x_i = [\text{forage yield}, \text{tdn}, \text{crude protein}]$

To balance the tradeoff between the average selection index value and variability within the index estimates for any given accesssion, we found the mean value for each accesssion's samples ($\bar{I} = \sum_{i=1}^{N} SI_i, n = 4$) and then penalized it based on the accessions coefficient of variation, resulting in the below definition.

$$I_{penalized} = \bar{I} \cdot (1 - \lambda \cdot \frac{\sigma_I}{\bar{I}})$$

where $\bar{I}$ is a given replicates selection index calculated, σ_I is the standard deviation of the selection indices over all 4 replicates, $\lambda \in [0, 1]$ is a tuning penalty parameter to downweight samples with a high coefficient of variation

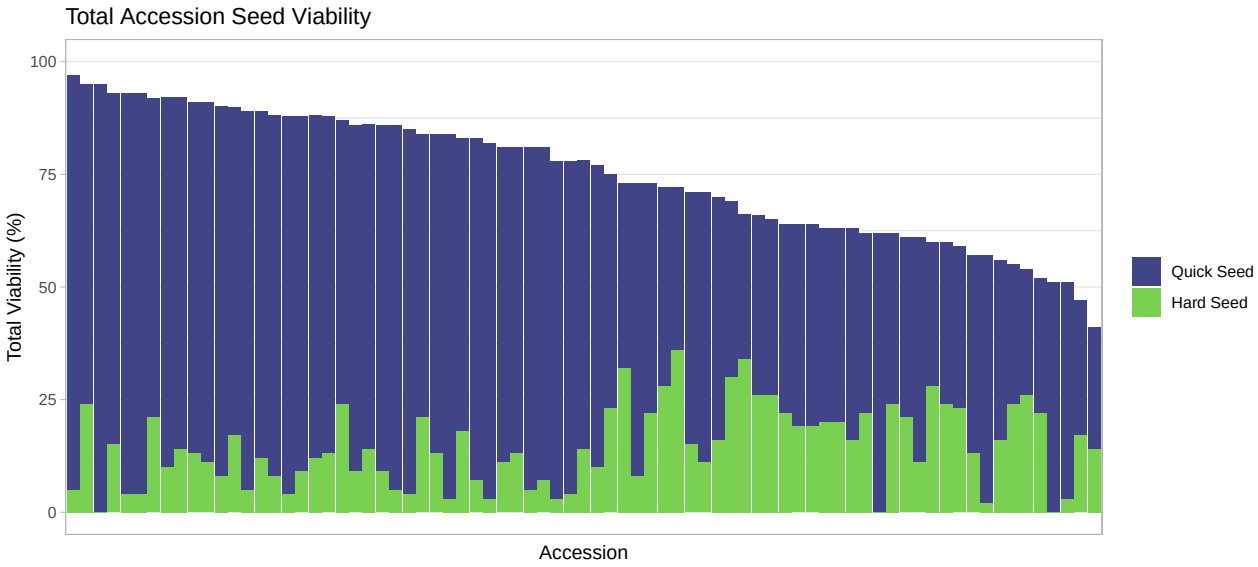
Results

Germination Results

We performed germination tests on all of the historical accesssions collected. We sowed 100 seeds of each line, and tallied the number of seeds that germinated quickly, and the total number of hard seeds that germinated late. Overall average of quick germination was 45.6% of the seeds, while an additional 18.6% were hard seeds on average for a total mean viability of 64.2%. There was wide variability

```
germination_plot <- ggplot(germination_long, aes(x=reorder(sample_id, -total_viable_pct), y=values, fil
        geom_bar(stat = 'identity', position='stack', width=.9)+
        theme_light()+
        scale_fill_viridis_d(option='D', begin=0.2, end=0.8)+
        labs(x = 'Accession', y='Total Viability (%)', fill="Germination", title="Total Accession Seed
        ylim(min=0, max=100)+
        theme(axis.text.x = element_blank(),
              axis.ticks.x = element_blank(),
              panel.grid.major.x = element_blank(),
              panel.grid.minor.x = element_blank(),
              legend.title = element_blank())

germination_plot
```



Yield Data

```
trial_layout <- fread('../data/trial_layout_metadata.csv') %>%
        mutate(sample_name = paste('LMP', plots, sep='-')) %>%
        dplyr::select(-V1)
yield_data <- fread('../data/lumpkin_ibf_forage_yield.csv') %>%
        left_join(trial_layout, by='sample_name') %>%
        dplyr::select(sample_name, trts, block, rep, row, column, forage_yield_kg_ha) %>%
        rename('sample_id' = 'trts')

# Filter out plots with really low yield
yield_data <- yield_data %>%
        filter(forage_yield_kg_ha > 500)

yield_summaries <- yield_data %>%
        group_by(sample_id) %>%
        summarize(mean_yield = mean(forage_yield_kg_ha, na.rm=TRUE),
                  sd_yield = sd(forage_yield_kg_ha, na.rm=TRUE),
                  n = n(),
                  se_yield = sd_yield / sqrt(n))

yield_mm <- lmer(forage_yield_kg_ha ~ sample_id + (1|block), data=yield_data, )
anova(yield_mm)
```

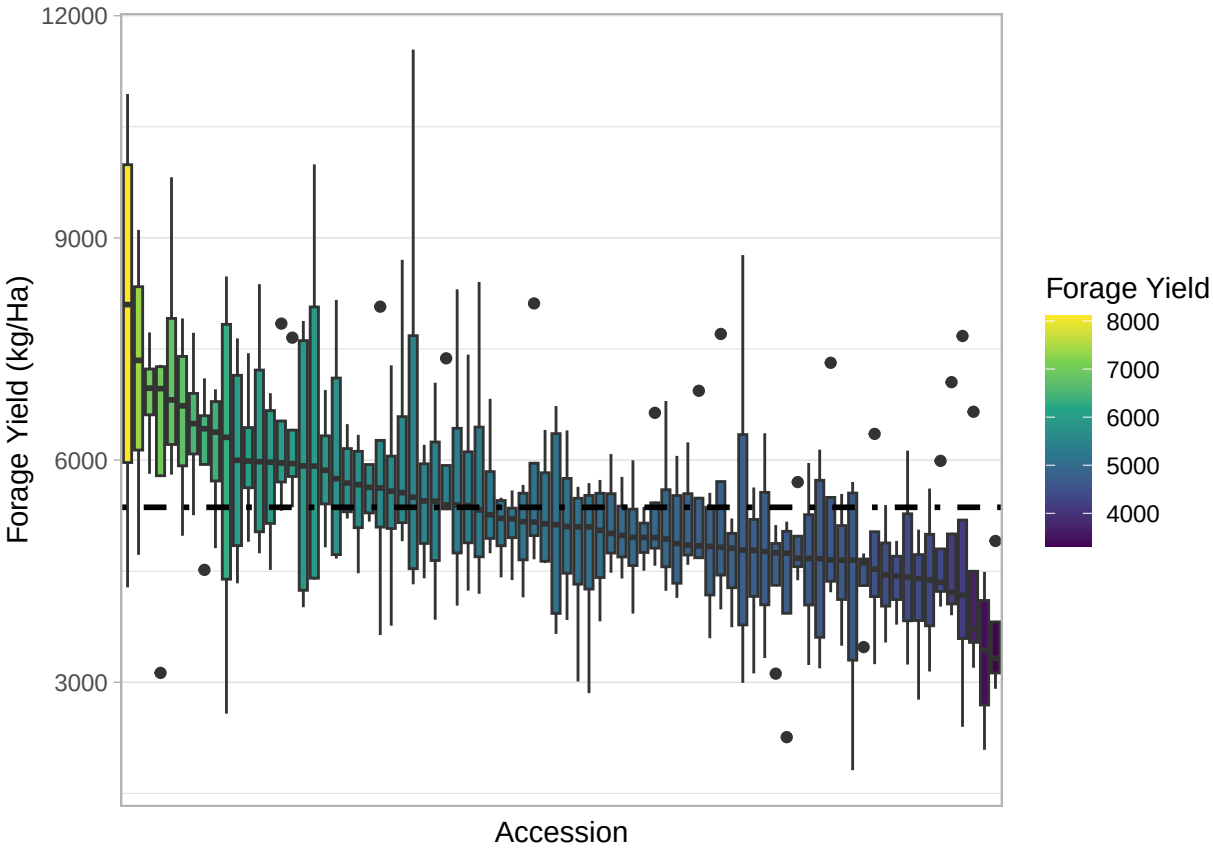
```
## Type III Analysis of Variance Table with Satterthwaite's method
##              Sum Sq Mean Sq NumDF  DenDF F value  Pr(>F)
## sample_id 216456293 2739953    79 226.05  1.4796 0.01361 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# summary(yield_mm)
```

NIR Modeling

Forage Trait	Forage Size	Num. Components	Test RMSE	Train RMSE
NEL	10mm	1	3.273	3.031
NEG	10mm	2	1.926	1.724
NEM	10mm	2	2.095	1.877
TDN	10mm	3	1.544	1.315
RFV	10mm	2	10.543	8.080
NDF	10mm	2	2.975	2.224
ADF	10mm	1	3.852	3.660
Crude Protein	10mm	3	1.957	0.907
Dry Matter Content	10mm	1	2.739	2.637
Moisture Content	10mm	1	2.739	2.637
NEL	1mm	4	3.937	3.050
NEG	1mm	2	2.896	2.246
NEM	1mm	2	3.142	2.439
TDN	1mm	4	1.848	1.559
RFV	1mm	5	11.293	8.491
NDF	1mm	5	2.889	2.174
ADF	1mm	2	4.744	4.029
Crude Protein	1mm	2	1.159	1.053
Dry Matter Content	1mm	1	2.997	2.557
Moisture Content	1mm	1	2.997	2.557

```
## Joining with `by = join_by(sample_id)`
```

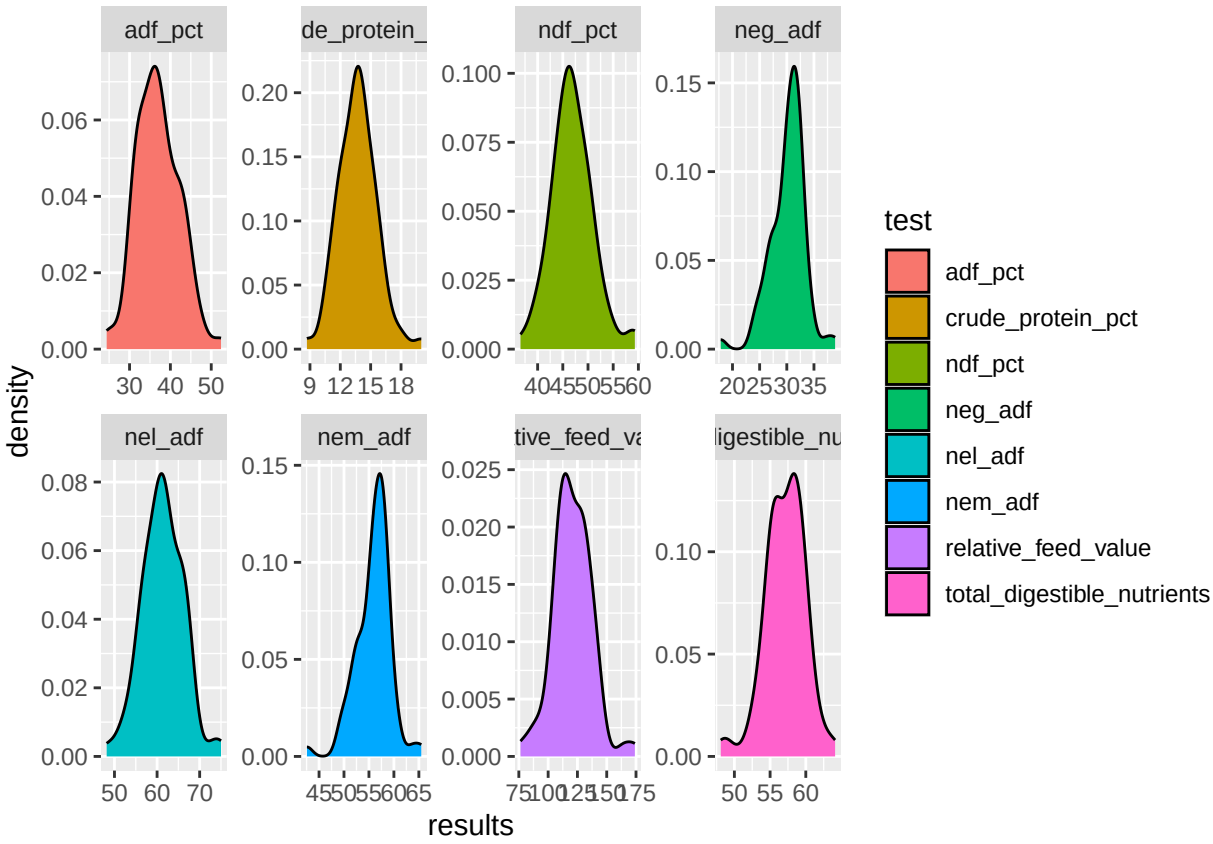



```
# Descending yield boxplot
# median_values = ibf_10mm_summary %>%
```

```
#         left_join(trial_layout %>% dplyr::select(c(sample_name, trts)), by='sample_name') %>%
#         dplyr::select(-sample_name) %>%
#         rename('sample_id' = 'trts') %>%
#         group_by(sample_id) %>%
#         summarize_all(median) %>%
#         filter(!sample_id %in% c('HI-GEST 660', 'Kura Clover', 'Sainfoin'))
#
# ibf_10mm_summary_df = ibf_10mm_summary %>%
#         left_join(trial_layout %>% dplyr::select(c(sample_name, trts)), by='sample_name') %>%
#         dplyr::select(-sample_name) %>%
#         rename('sample_id' = 'trts')
#
#
# cp_plot = ggplot(
#         ibf_10mm_summary_df,
#         aes(x=reorder(sample_id, -adf_pct, FUN=median), y=adf_pct)
#         )+
#         geom_boxplot()+
#         scale_fill_viridis_c(option='D')+
#         theme_light()+
#         labs(x = 'Accession', y='Forage Yield (kg/Ha)', fill = 'Forage Yield')+
#         theme(axis.text.x = element_blank(),
#               axis.ticks.x = element_blank(),
#               panel.grid.major.x = element_blank(),
#               panel.grid.minor.x = element_blank())
#
# cp_plot
```

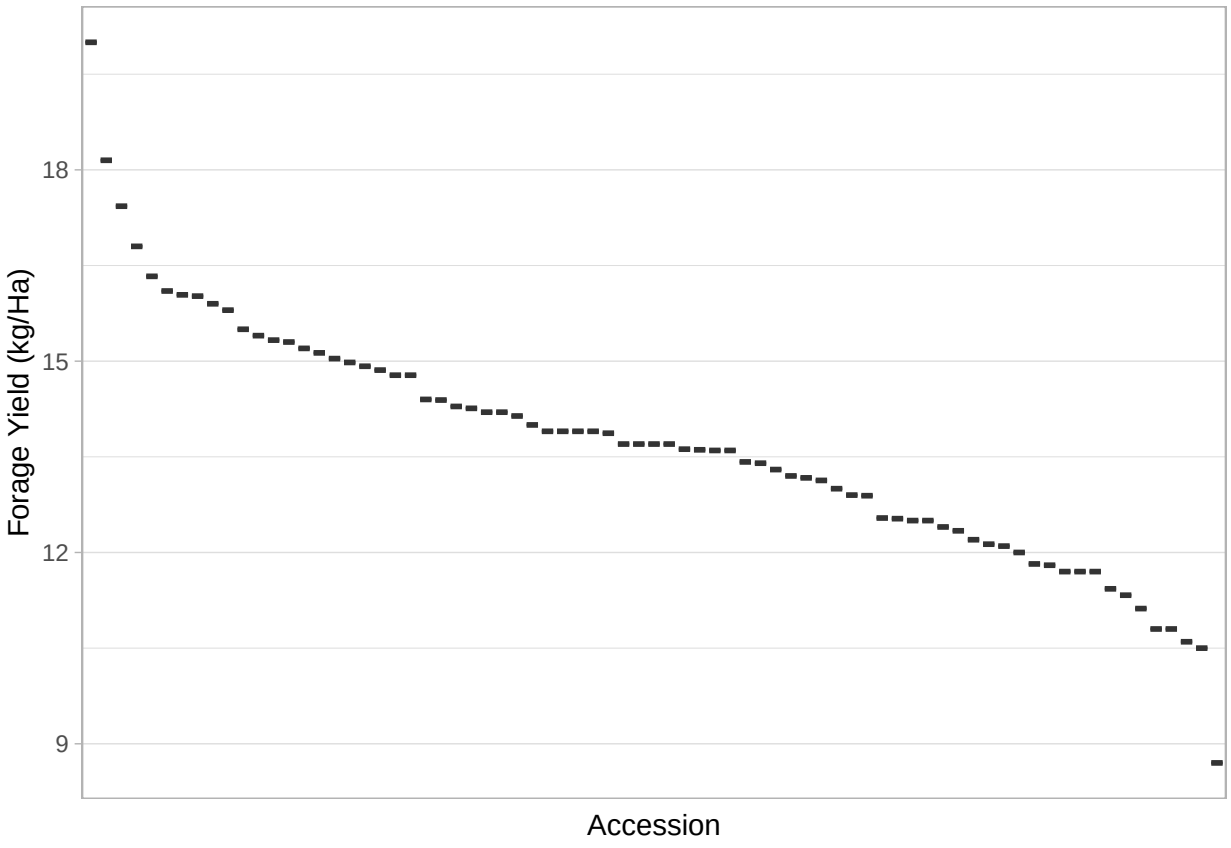
```
ggplot(wetchem_longer, aes(x=results, group=test, fill=test))+
        geom_density()+
        facet_wrap(~test, scales = 'free', nrow=2)
```

```
## Warning: Removed 4 rows containing non-finite outside the scale range
## (`stat_density()`).
```




```
crude_protein_plot = ggplot(wetchem, aes(x=reorder(sample_name, -crude_protein_pct, FUN=median), y=crud
        geom_boxplot()+
        scale_fill_viridis_c(option='D')+
        # geom_hline(yintercept=mean(yield_data$forage_yield_kg_ha), linewidth=1, linetype=4)+
        theme_light()+
        labs(x = 'Accession', y='Forage Yield (kg/Ha)', fill = 'Forage Yield')+
        theme(axis.text.x = element_blank(),
              axis.ticks.x = element_blank(),
              panel.grid.major.x = element_blank(),
              panel.grid.minor.x = element_blank())

crude_protein_plot
```



Trait Predictions and Repeatability

```
# Function to calculate the repeatability of a trait
calc_repeatability <- function(model, r) {
        # Grab the variance components
        var_comp <- as.data.frame(VarCorr(model))

        # Get the variance of the breeding lines (Random Effects)
        var_g <- var_comp[1, "vcov"]
        # Grab the total variance of the population
        var_p <- var_comp[2, "vcov"]
        # Calculate R
        r = var_g / (var_g + (var_p / r))

        r
}

# Construct a list of all the mixed models of traits setting the sample_ids as random effects and block
model_list = list(
        'forage_yield_kg_ha' = fy_mm <- lmer(forage_yield_kg_ha ~ 1 + (1|sample_id) + block, data=ibf_1
        'dry_matter_pct' = dm_mm <- lmer(dry_matter_pct ~ 1 + (1|sample_id) + block, data = ibf_10mm_su
        'moisture_pct' = moi_mm <- lmer(moisture_pct ~ 1 + (1|sample_id) + block, data = ibf_10mm_summa
        'adf_pct' = adf_mm <- lmer(adf_pct ~ 1 + (1|sample_id) + block, data = ibf_10mm_summary),
        'ndf_pct' = ndf_mm <- lmer(ndf_pct ~ 1 + (1|sample_id) + block, data = ibf_10mm_summary),
        'crude_protein_pct' = cp_mm <- lmer(crude_protein_pct ~ 1 + (1|sample_id) + block, data = ibf_1
        'relative_feed_value' = rfv_mm <- lmer(relative_feed_value ~ 1 + (1|sample_id) + block, data =
```

```
        'total_digestible_nutrients' = tdn_mm <- lmer(total_digestible_nutrients ~ 1 + (1|sample_id) + b
        'neg_adf' = neg_mm <- lmer(neg_adf ~ 1 + (1|sample_id) + block, data = ibf_10mm_summary),
        'nel_adf'= nel <- lmer(nel_adf ~ 1 + (1|sample_id) + block, data = ibf_10mm_summary),
        'nem_adf' = nem <- lmer(nem_adf ~ 1 + (1|sample_id) + block, data = ibf_10mm_summary)
)

# Extract and store the R values
r_values <- c()
for (name in names(model_list)) {
        print(name)
        model = model_list[[name]]
        r = calc_repeatability(model, r=4)
        r_values <- c(r_values, r)
}
```

```
## [1] "forage_yield_kg_ha"
## [1] "dry_matter_pct"
## [1] "moisture_pct"
## [1] "adf_pct"
## [1] "ndf_pct"
## [1] "crude_protein_pct"
## [1] "relative_feed_value"
## [1] "total_digestible_nutrients"
## [1] "neg_adf"
## [1] "nel_adf"
## [1] "nem_adf"
```

```
# Build a dataframe with the repeatability values
r_df <- data.frame('test_trait' = names(model_list), 'r' = r_values)

# Get the whole population means
ibf_population_means <- ibf_10mm_summary %>%
        filter(!sample_id %in% c('HI-GEST 660', 'Kura Clover', 'Sainfoin')) %>%
        dplyr::select(-c(1:4)) %>%
        summarize_all(.funs = list('mean' = mean, 'min'=min, 'max'=max, 'var' = var)) %>%
        pivot_longer(cols = everything(), names_to="test_trait", values_to = "value") %>%
        separate(test_trait, into=c('test_trait', 'metric'), sep='_(?=[^_]+$)') %>%
        pivot_wider(names_from = metric)

ibf_population_means <- r_df %>%
        left_join(ibf_population_means) %>%
        mutate(range = paste(round(min, 2), round(max, 2), sep = " - ")) %>%
        dplyr::select(test_trait, mean, range, var, r)
```

```
## Joining with `by = join_by(test_trait)`
```

```
rownames(ibf_population_means) <- c("Forage Yield kg/Ha", "Dry Matter Pct.", "Moisture Content Pct.", "A
ibf_population_means <- ibf_population_means %>%
        dplyr::select(-test_trait)

kable(
        ibf_population_means,
        digits = 4,
        col.names = c('Mean', 'Range', 'Variance', 'R'),
        row.names = TRUE,
```

```
        align = 'c'
) %>%
        kable_material()
```

	Mean	Range	Variance	R
Forage Yield kg/Ha	5433.7723	1812.16 - 11541.71	2154779.5734	0.2966
Dry Matter Pct.	91.8370	89.96 - 94.12	0.4125	0.4589
Moisture Content Pct.	8.1630	5.88 - 10.04	0.4125	0.4589
ADF Pct.	37.0031	24.21 - 48.33	14.5010	0.4948
NDF Pct.	46.1693	33.14 - 56.69	12.4639	0.6044
Crude Protein Pct.	13.8688	9.41 - 21.07	2.7116	0.1186
RFV	122.7109	78.4 - 174.54	224.4228	0.5435
TDN	57.5679	49.54 - 64.57	6.9219	0.5695
NEG ADF	30.8304	22.34 - 38.11	6.6452	0.6360
NEL ADF	61.4325	49.94 - 74.29	14.7500	0.4958
NEM ADF	56.6110	47.35 - 64.56	7.9221	0.6345

Selection Index

We calculated the penalized selection index for 11 different λ weights ranging from 0.0 to 1.0 in 0.1 increment steps, and then calculated the ranking of each accession to see if the rankings were stable (indicating selections that had low variability) or if changing the λ parameter caused a lot of rank change in the selection.

```
LAMBDA = c(0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0)

for(lambda in LAMBDA){
        SI_column_name = paste('SI', lambda, sep='_')
        rank_column_name = paste('rank', lambda, sep = '_')

        # Calculate the new SI values
        SI_df[[SI_column_name]] <- SI_df$mean * (1 - lambda * SI_df$cv)

        # Rank across all samples (higher SI = better)
        SI_df[[rank_column_name]] <- rank(-SI_df[[SI_column_name]], ties.method = "min")
}

SI_df_ordered <- SI_df %>%
        arrange(desc(I_bar)) %>%
        pivot_longer(cols=starts_with('rank'), names_to = 'key', values_to = 'I_penalized')

# Subset for the top 30 lines
subset_df <- SI_df[1:30,]

# Convert to long format for rank columns
df_long <- subset_df %>%
        dplyr::select(sample_id, starts_with("rank_")) %>%
        pivot_longer(cols = starts_with("rank_"),
                     names_to = "lambda",
                     values_to = "rank") %>%
        mutate(lambda = str_remove(lambda, "rank_"),
               lambda = as.numeric(lambda))
```

```r
labels_df <- df_long %>%
  filter(lambda == 0)

# Plot with labels at lambda = 0
ggplot(df_long, aes(x = lambda, y = rank, group = sample_id, color=rank)) +
        geom_line(linewidth=1) +
        geom_text(data = labels_df,
                  aes(label = sample_id),
                  color = 'black',
                  hjust = .0,  # nudge to the left of the line
                  size = 1.5,
                  nudge_x = -0.1,
                  check_overlap = TRUE) +
        scale_y_reverse() +
        theme_minimal() +
        labs(title = "Rank Changes Across Lambda Values",
             x = expression(lambda),
             y = "Rank (1 = Best)",
             color='Rank') +
        coord_cartesian(clip = "off") +
        theme(panel.grid.minor = element_blank(),
              plot.margin = margin(5.5, 5.5, 5.5, 5.5)) + # extra margin for labels
        scale_color_viridis_c(option='D')+
        scale_x_continuous(limits = c(-.1, 1))
```