

ggpedigree: Visualizing Pedigrees with ‘ggplot2’ and ‘plotly’

S. Mason Garrison ¹✉

¹ Department of Psychology, Wake Forest University, North Carolina, USA ✉ Corresponding author

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Open Journals](#) 

Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

License

Authors of papers retain copyright,
and release the work under a
Creative Commons Attribution 4.0
International License ([CC BY 4.0](#)).

Summary

Pedigree diagrams underpin research and practice across genetics, animal breeding, genealogy, forensics, and counseling. They help medical geneticists trace the inheritance of Mendelian diseases and identify at-risk relatives; enable dairy breeders to plan matings that improve milk yield; support genealogists in reconstructing ancestry; assist forensic scientists in establishing familial connections in criminal investigations; and facilitate family therapists and counselors in understanding their clients' relationships (McGoldrick, Gerson, & Petry, 2020). Early R tools such as kinship2 (Sinnwell, Therneau, & Schaid, 2014) plot simple nuclear families effectively, but they do not scale to today's pedigrees that can exceed 1,000s of individuals. As datasets have grown, researchers now work with increasingly complex family structures, including large-scale plant breeding pedigrees (Shaw, Graham, Kennedy, Milne, & Marshall, 2014), web-based pedigree management systems (Ranaweera, Makalic, Hopper, & Bickerstaffe, 2018), interactive pedigree editors (Carver et al., 2018), and behavior genetic studies of extended family structures (Garrison et al., 2023; M. Hunter, Garrison, Burt, & Rodgers, 2021). That complexity exposes the limitations of existing tools, which often struggle with large and complex datasets. ggpedigree addresses this need by combining a vectorised layout algorithm, ggplot2 output, and optional plotly interactivity.

Statement of need

Pedigree visualization has traditionally relied on proprietary software (e.g., Progeny, GenoPro, Pedigree Viewer) or R packages like kinship2 (Sinnwell et al., 2014), pedtools (Magnus Dehli Vigeland, 2021a), or pedtricks (J. Martin, Wolak, Johnston, & Morrissey, 2025). While these tools are functional for many use cases, their limitations become pronounced when working with complex, modern pedigree datasets or when more detailed customization is required. Most R packages focus on base graphics or simple ggplot2 implementations.

First, base R graphics lack the modular design and extensibility for generating publication-quality pedigree figures. For example, kinship2 (Sinnwell et al., 2014) uses base graphics and loop-based layout functions that do not scale well. pedtricks (J. Martin et al., 2025), by contrast, returns ggplot2-based plots and is designed for large, multigenerational animal pedigrees, but offers minimal support for annotation layering, per-individual theming, or integration with phenotypic and model-based data

Second, most R-based tools offer no interactivity. Static graphics are often sufficient for publication, but interactivity improves exploration and communication during model development or data cleaning. A notable exception is pedtools (Magnus Dehli Vigeland, 2021b), which offers a sister shiny app, QuickPed (Magnus D. Vigeland, 2022). While the R ecosystem includes libraries, like plotly, that support interactive plotting, these features have yet to be integrated into pedigree functions.

41 Third, current solutions are often poorly integrated with tidyverse workflows and do not
42 expose the full theming and layering capabilities familiar to ggplot2 users (Hadley Wickham,
43 2016). In animal-focused workflows, rapid rendering seems to takes precedence over aesthetic
44 flexibility. I suspect that this is because users tend to work with more uniform data and fewer
45 phenotypes. By contrast, human-focused workflows—particularly in behavior genetics and
46 genetic epidemiology (Garrison et al., 2023; Lyu, Hunter, Good, Carroll, & Garrison, 2025;
47 McArdle & McDonald, 1984)—require closer integration with tidyverse pipelines and more
48 flexible plotting systems to accommodate complex pedigree structures and harmonization of
49 phenotypes across data sources. In other words, the needs are different.

50 Several R packages attempt to address these challenges with built-in pedigree plotting functions.
51 kinship2 (Sinnwell et al., 2014) remains widely used but produces static base graphics and
52 relies on non-vectorized recursive layout functions that do not scale well to large families.
53 A partial ggplot2 implementation exists but is incomplete, non-vectorized, and not actively
54 maintained. pedtricks, a revival of pedantics (Morrissey & Wilson, 2010), provides a ggplot2-
55 based implementation for large animal pedigrees but lacks extensibility and interactivity. The
56 geneHapR (Zhang, Jia, & Diao, 2023) package focuses on haplotype visualization rather than
57 general pedigree structure. The pedgene package (Daniel & Sinnwell, 2024) offers some
58 plotting functions but is primarily designed for association testing rather than visualization.
59 The pedigreejs package (Carver et al., 2018) provides an interactive pedigree editor but does
60 not integrate with R or ggplot2, limiting its utility for R users.

61 None of these packages offers the combination of modern ggplot2 integration, interactive
62 capabilities, and extensibility that ggpedigree provides. ggpedigree addresses these limitations
63 by providing a comprehensive visualization framework built on modern R graphics infrastruc-
64 ture. It leverages the extensive customization capabilities of ggplot2 while adding specialized
65 functionality for pedigree-specific visualization challenges.

66 Software Architecture

67 ggpedigree is built on a modular architecture that separates data processing, layout calculation,
68 and visualization layers. The core workflow involves: (1) data standardization and family
69 structure analysis using BGmisc functions, (2) coordinate calculation using algorithms adapted
70 from kinship2, (3) relationship connection mapping, and (4) layer-based plot construction
71 using ggplot2 geometry functions. This design allows users to customize any aspect of the
72 visualization while maintaining computational efficiency for large pedigrees. The package
73 integrates tightly with the broader R ecosystem, particularly the tidyverse (H. Wickham et
74 al., 2019) and BGmisc (Garrison, S. M., Hunter, M. D., Lyu, X., Trattner, J. D., & Burt,
75 S. A., 2024). All functions return standard R objects (ggplot or plotly) that can be further
76 customized.

77 Features

78 I describe the main features of the ggpedigree package below. More detailed descriptions of
79 features and usage is available in the [package vignettes](#), including examples of how to create
80 static and interactive pedigree plots, customize aesthetics, and visualize relatedness matrices.
81 Additional example data include squirrel data from the Kluane Red Squirrel Project (McFarlane
82 et al., 2014, 2015) and Targaryen family data from the Song of Ice and Fire universe (G. R.
83 Martin, 1997; G. R. R. Martin, 2018).

84 Core Functions

- 85 ■ Data Standardization and Family Structure Analysis: `ggPedigree()` integrates with
86 BGmisc functions like `ped2fam()` to organize individuals by family, `recodeSex()` to
87 standardize sex coding, and `checkParentIDs()` to validate pedigree structures. The

88 function handles consanguineous relationships and individuals appearing in multiple
89 pedigree positions.

90 ■ Coordinate Calculation: `calculateCoordinates()` computes optimal positioning for indi-
91 viduals using algorithms adapted from `kinship2::align.pedigree`, with enhancements
92 for large multi-generational pedigrees and complex family structures. These steps are
93 vectorized as much as possible to ensure computational efficiency and compatibility with
94 `ggplot2`.

95 ■ Relationship Connection Mapping: `calculateConnections()` generates connection paths
96 between family members, mapping parent-child, sibling, spousal, and twin relationships.
97 The function determines midpoints for line intersections and handles overlapping con-
98 nections with specialized curved segments. These calculations are optimized for large
99 datasets by using vectorized operations rather than the loop-based approaches used in
100 `kinship2`.

101 ■ Layer-based Plot Construction: `ggPedigree()` constructs plots using `ggplot2` geometry
102 functions, returning standard `ggplot2` objects that integrate with existing R workflows.
103 `ggPedigreeInteractive()` extends plots into interactive `plotly` widgets. A config system
104 allows customization of over 100 aesthetic and layout parameters.

105 ■ Focal Individual Highlighting: Advanced functionality to highlight specific individuals and
106 their relatives based on additive genetic, mitochondrial, or other relationship matrices.

107 Specific Visualization Functions

108 ■ Pedigree Plotting: `ggPedigree()` creates static pedigree plots using `ggplot2`. It can
109 handle large pedigrees with thousands of individuals.

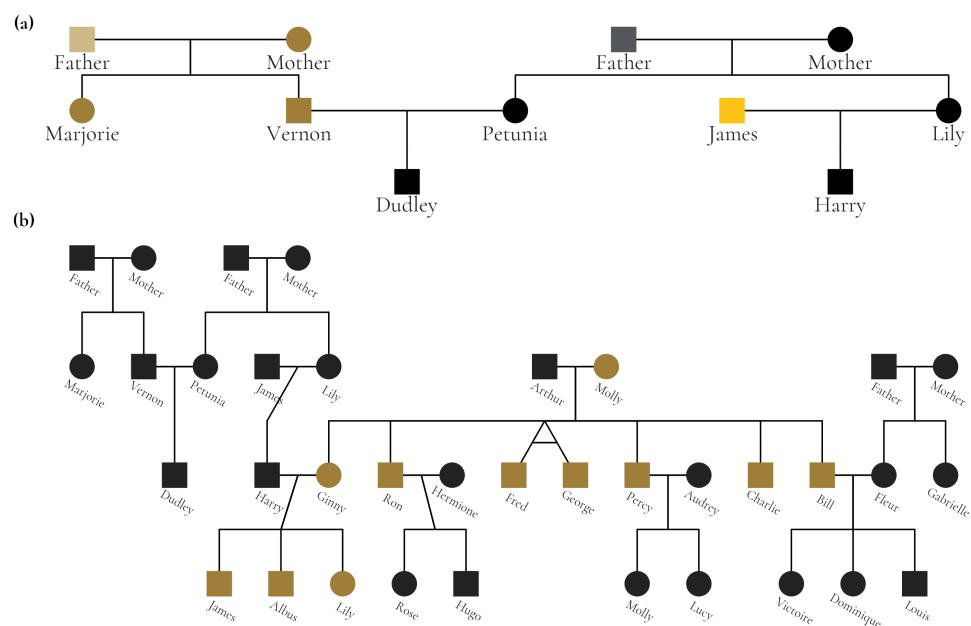
110 ■ Interactive Pedigree Visualization: `ggPedigreeInteractive()` generates interactive pedi-
111 gree plots using `plotly`. Users can configure tooltip content to display individual IDs,
112 names, phenotypic information, and other relevant data. It supports tooltips, zooming,
113 and panning for detailed exploration of family structures.

114 ■ Relatedness Matrix Heatmaps: `ggRelatednessMatrix()` creates customizable heatmap
115 visualizations for relatedness matrices with support for hierarchical clustering, interactive
116 exploration, and seamless integration with `BGmisc` relatedness calculations.

117 ■ Phenotype-Relatedness Analysis: `ggPhenotypeByDegree()` produces specialized plots
118 for visualizing phenotypic correlations as a function of genetic relatedness, including
119 confidence intervals and statistical summaries for quantitative genetic analysis.

120 Illustrative Example of `ggPedigree`

121 I illustrate some of the features by reproducing the figure from (M. D. Hunter et al., 2025),
122 restyled to conform with the Wake Forest Style Guidelines.



I have combined two figures using patchwork (Pedersen, 2025) to show the pedigree of the Potter family. The first plot (a) highlights the individual mitochondrial lines in the Dursley and Evans families, while the second plot (b) shows the entire pedigree with mitochondrial descendants of Molly Weasley highlighted in gold. Features such as focal individual highlighting, custom color scales, text labels, and fonts are easily implemented.

Collectively, these tools provide a valuable resource for behavior geneticists and others who work with extended family data. They were developed as part of a grant and have been used in several ongoing projects, presentations (Garrison, 2024; M. D. Hunter, Garrison, Lyu, Good, & Burt, 2024), and forthcoming papers (Burt et al., 2025; M. D. Hunter et al., 2025; Lyu et al., 2025).

Availability

The ggpedigree package is open-source and available on both GitHub at <https://github.com/R-Computing-Lab/ggpedigree> and the Comprehensive R Archive Network (CRAN) at <https://cran.r-project.org/package=ggpedigree>. It is licensed under the GNU General Public License.

Acknowledgments

The current research is supported by the National Institute on Aging (NIA), RF1-AG073189. The author would like to thank Michael Hunter for his enthusiasm for the development of this package.

References

- Burt, S. A., Garrison, S. M., Lyu, X., Rodgers, J. L., Carroll, S. L., Smith, K. R., & Hunter, M. D. (2025). *Inherited mtDNA contributes to longevity: Evidence from extended pedigrees with 176 million kinship pairs*. *Revise and resubmit at eBioMedicine*.
- Carver, T., Cunningham, A. P., Babb de Villiers, C., Lee, A., Hartley, S., Tischkowitz, M., Walter, F. M., et al. (2018). Pedigreejs: A web-based graphical pedigree editor. *Bioinformatics*, 34(6), 1069–1071. doi:10.1093/bioinformatics/btx705

- 150 Daniel, S., & Sinnwell, J. (2024). Pedgene: Gene-level variant association tests for pedigree
151 data. doi:[10.32614/CRAN.package.pedgene](https://doi.org/10.32614/CRAN.package.pedgene)
- 152 Garrison, S. M. (2024). Charting new paths in behavior genetics: Developing a comprehensive
153 r visualization atlas. *Behavior Genetics* (Vol. 54, p. 488488). SPRINGER. Retrieved from
154 <https://scholar.google.com/scholar?cluster=4710819130852314260&hl=en&oi=scholar>
- 155 Garrison, S. M., Hunter, M. D., Lyu, X., Trattner, J. D., & Burt, S. A. (2024). BGmisc: An r
156 package for extended behavior genetics analysis. *Journal of Open Source Software*, 9(94).
157 doi:[10.21105/joss.06203](https://doi.org/10.21105/joss.06203)
- 158 Garrison, S. M., Lyu, X., Hunter, M. D., Rodgers, J. L., Smith, K. R., Coon, H., & Burt, S.
159 A. (2023). Analyzing extended cousin similarity to unravel the mystery of mtDNA and
160 longevity. *Behavior Genetics*. doi:[10.1007/s10519-023-10156-9](https://doi.org/10.1007/s10519-023-10156-9)
- 161 Hunter, M. D., Garrison, S. M., Lyu, X., Good, R., & Burt, S. A. (2024). Tools for
162 biometric modeling in large population databases. *Behavior Genetics* (Vol. 54, p.
163 531531). SPRINGER. Retrieved from [https://scholar.google.com/scholar?cluster=](https://scholar.google.com/scholar?cluster=16878992125450714238&hl=en&oi=scholar)
164 [16878992125450714238&hl=en&oi=scholar](https://scholar.google.com/scholar?cluster=16878992125450714238&hl=en&oi=scholar)
- 165 Hunter, M. D., Garrison, S. M., Lyu, X., Good, R., Carroll, S. L., & Burt, S. A. (2025). *Tracing*
166 *the right path: Determination of large pedigree segmentation and relatedness. Revise and*
167 *Resubmit at Behavior Genetics*.
- 168 Hunter, M., Garrison, S. M., Burt, S. A., & Rodgers, J. L. (2021). The Analytic Identification
169 of Variance Component Models Common to Behavior Genetics. *Behavior Genetics*, 51(4),
170 425–437. doi:[10.1007/s10519-021-10055-x](https://doi.org/10.1007/s10519-021-10055-x)
- 171 Lyu, S. A., X. and Burt, Hunter, M. D., Good, R., Carroll, S. L., & Garrison, S. M. (2025).
172 Detecting mtDNA effects with an extended pedigree model: An analysis of statistical power
173 and estimation bias. *Behavior Genetics*. doi:[10.1007/s10519-025-10225-1](https://doi.org/10.1007/s10519-025-10225-1)
- 174 Martin, G. R. (1997). *A game of thrones* (Vol. 1). Bantam.
- 175 Martin, G. R. R. (2018). *Fire & blood: 300 years before a game of thrones*. New York:
176 Random House Worlds.
- 177 Martin, J., Wolak, M., Johnston, S., & Morrissey, M. (2025). *Pedtricks: Visualize, summarize*
178 *and simulate data from pedigrees*. doi:[10.32614/CRAN.package.pedtricks](https://doi.org/10.32614/CRAN.package.pedtricks)
- 179 McArdle, J. J., & McDonald, R. P. (1984). Some algebraic properties of the reticular action
180 model for moment structures. *British Journal of Mathematical and Statistical Psychology*,
181 37, 234–251. doi:[10.1111/j.2044-8317.1984.tb00802.x](https://doi.org/10.1111/j.2044-8317.1984.tb00802.x)
- 182 McFarlane, S. E., Boutin, S., Humphries, M. M., McAdam, A. G., Gorrell, J. C., & Colt-
183 man, D. W. (2015, January 21). Data from: Very low levels of direct additive ge-
184 netic variance in fitness and fitness components in a red squirrel population. Dryad.
185 doi:[10.5061/DRYAD.N5Q05](https://doi.org/10.5061/DRYAD.N5Q05)
- 186 McFarlane, S. E., Gorrell, J. C., Coltman, D. W., Humphries, M. M., Boutin, S., & McAdam,
187 A. G. (2014). Very low levels of direct additive genetic variance in fitness and fitness
188 components in a red squirrel population. *Ecology and Evolution*, 4(10), 1729–1738.
189 doi:[10.1002/ece3.982](https://doi.org/10.1002/ece3.982)
- 190 McGoldrick, M., Gerson, R., & Petry, S. (2020). *Genograms: Assessment and Treatment*.
191 Erscheinungsort nicht ermittelbar: W. W. Norton & Company.
- 192 Morrissey, M. B., & Wilson, A. J. (2010). pedantics: an r package for pedigree-based genetic
193 simulation and pedigree manipulation, characterization and viewing. *Molecular Ecology*
194 *Resources*, 10(4), 711–719. doi:[10.1111/j.1755-0998.2009.02817.x](https://doi.org/10.1111/j.1755-0998.2009.02817.x)
- 195 Pedersen, T. L. (2025). *Patchwork: The composer of plots*. doi:[10.32614/CRAN.pack-](https://doi.org/10.32614/CRAN.package.patchwork)
196 [age.patchwork](https://doi.org/10.32614/CRAN.package.patchwork)

- 2197 Ranaweera, T., Makalic, E., Hopper, J. L., & Bickerstaffe, A. (2018). An open-source,
2198 integrated pedigree data management and visualization tool for genetic epidemiology.
2199 *International Journal of Epidemiology*, 47(4), 1034–1039. doi:[10.1093/ije/dyy049](https://doi.org/10.1093/ije/dyy049)
- 2200 Shaw, P. D., Graham, M., Kennedy, J., Milne, I., & Marshall, D. F. (2014). Helium: Visualiza-
2201 tion of large scale plant pedigrees. *BMC Bioinformatics*, 15(1), 259. doi:[10.1186/1471-2105-15-259](https://doi.org/10.1186/1471-2105-15-259)
- 2203 Sinnwell, J. P., Therneau, T. M., & Schaid, D. J. (2014). The kinship2 r package for pedigree
2204 data. *Human Heredity*, 78, 91–93. doi:[10.1159/000363105](https://doi.org/10.1159/000363105)
- 2205 Vigeland, Magnus Dehli. (2021b). *Pedigree analysis in r*. London: Academic Press, an
2206 imprint of Elsevier. Retrieved from <https://shop.elsevier.com/books/pedigree-analysis-in-r/vigeland/978-0-12-824430-2>
- 2207 Vigeland, Magnus Dehli. (2021a). Pedigree analysis in {r}.
- 2209 Vigeland, Magnus D. (2022). QuickPed: An online tool for drawing pedigrees and analysing
2210 relatedness. *BMC Bioinformatics*, 23(1), 220. doi:[10.1186/s12859-022-04759-y](https://doi.org/10.1186/s12859-022-04759-y)
- 2211 Wickham, Hadley. (2016). *ggplot2*. Use R! Cham: Springer International Publishing.
2212 doi:[10.1007/978-3-319-24277-4](https://doi.org/10.1007/978-3-319-24277-4)
- 2213 Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L., François, R., Grolemund, G.,
2214 et al. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, 4(43), 1686.
2215 doi:[10.21105/joss.01686](https://doi.org/10.21105/joss.01686)
- 2216 Zhang, R., Jia, G., & Diao, X. (2023). geneHapR: An r package for gene haplotypic statistics
2217 and visualization. *BMC Bioinformatics*, 24(1), 199. doi:[10.1186/s12859-023-05318-9](https://doi.org/10.1186/s12859-023-05318-9)

2218 Appendix

2219 Source code for the figures above is available in the vignettes/articles/_paper.Rmd file.
2220 The code below reproduces the figure using the ggpedigree package.

```
library(ggpedigree) # ggPedigree lives here
library(BGmisc) # helper utilities & example data
library(tidyverse) # for data manipulation and plotting
library(showtext)
library(sysfonts)
library(patchwork) # for combining plots

# Load the potter pedigree data
data("potter")
# Removing Last names
df_potter <- potter %>%
  mutate(
    name = case_when(
      personID == 1 ~ "Vernon",
      personID == 2 ~ "Marjorie",
      personID == 3 ~ "Petunia",
      personID == 4 ~ "Lily",
      personID == 5 ~ "James",
      personID == 6 ~ "Dudley",
      personID == 7 ~ "Harry",
      personID == 8 ~ "Ginny",
      personID == 9 ~ "Arthur",
      personID == 10 ~ "Molly",
      personID == 11 ~ "Ron",
```



```

personID == 12 ~ "Fred",
personID == 13 ~ "George",
personID == 14 ~ "Percy",
personID == 15 ~ "Charlie",
personID == 16 ~ "Bill",
personID == 17 ~ "Hermione",
personID == 18 ~ "Fleur",
personID == 19 ~ "Gabrielle",
personID == 20 ~ "Audrey",
personID == 21 ~ "James",
personID == 22 ~ "Albus",
personID == 23 ~ "Lily",
personID == 24 ~ "Rose",
personID == 25 ~ "Hugo",
personID == 26 ~ "Victoire",
personID == 27 ~ "Dominique",
personID == 28 ~ "Louis",
personID == 29 ~ "Molly",
personID == 30 ~ "Lucy",
personID == 101 ~ "Mother",
personID == 102 ~ "Father",
personID == 103 ~ "Mother",
personID == 104 ~ "Father",
personID == 105 ~ "Mother",
personID == 106 ~ "Father"
)
)
# Load Google fonts for styling
font_add_google(name = "Cormorant", family = "cormorant")
showtext_auto() # Load the showtext package to render Google fonts

# Set the WFU style guidelines for the plot
text_color_wfu <- "#222222" # dark grey for text labels
focal_fill_color_values_wfu <- c(
  "#9E7E38", "#000000", "#FDC314", "#CEB888", "#53565A"
)
family_wfu <- "cormorant"
text_size_wfu <- 14

# Create Panel A
m1 <- ggPedigree(df_potter %>% filter(personID %in% c(1:7, 101:104)),
  famID = "famID",
  personID = "personID",
  config = list(
    label_include = TRUE,
    label_column = "name",
    point_size = 8,
    focal_fill_personID = 8,
    segment_linewidth = 0.5,
    label_text_size = 17,
    label_text_color = text_color_wfu,
    axis_text_color = text_color_wfu,
    label_text_family = family_wfu,
    focal_fill_include = TRUE,
    label_nudge_y = -0.32,

```

```
focal_fill_method = "manual",
focal_fill_color_values = focal_fill_color_values_wfu,
focal_fill_force_zero = TRUE,
label_method = "geom_text",
focal_fill_na_value = text_color_wfu,
focal_fill_scale_midpoint = 0.40,
focal_fill_component = "matID",
focal_fill_labels = NULL,
sex_legend_show = FALSE,
sex_color_include = FALSE
)
) + guides(shape = "none") + theme(
  plot.title = element_blank(),
  plot.title.position = "plot",
  text = element_text(family = family_wfu, size = 14)
) + coord_cartesian(ylim = c(3.25, 1), clip = "off")
# Create Panel B
m2 <- ggPedigree(df_potter,
  famID = "famID",
  personID = "personID",
  config = list(
    label_include = TRUE,
    label_column = "name",
    point_size = 8,
    focal_fill_personID = 8, # Molly Weasley
    segment_linewidth = 0.5,
    label_text_size = 10, #9.75,
    label_text_family = family_wfu,
    label_text_color = text_color_wfu,
    axis_text_color = text_color_wfu,
    label_nudge_y = -0.25,
    label_nudge_x = .05,
    focal_fill_include = TRUE,
    focal_fill_method = "gradient2",
    focal_fill_high_color = "#9E7E38",
    focal_fill_mid_color = "#9E7E38",
    focal_fill_low_color = text_color_wfu[2],
    focal_fill_scale_midpoint = 0.85,
    focal_fill_component = "mitochondrial",
    focal_fill_force_zero = TRUE,
    label_method = "ggrepel",
    focal_fill_na_value = text_color_wfu,
    label_text_angle = -30,
    sex_legend_show = FALSE,
    sex_color_include = FALSE
  )
) + theme(
  legend.position = "none",
  plot.title = element_blank(),
  plot.title.position = "plot",
  text = element_text(
    family = family_wfu,
    size = text_size_wfu, face = "bold"
  )
) + coord_cartesian(ylim = c(4.25, .9), clip = "off")
```



```
# Combine the two plots using patchwork
showtext_auto()
result <- m1 + m2 +
  plot_layout(
    ncol = 1, heights = c(1.1, 2.5),
    guides = "collect", tag_level = "new"
  ) +
  plot_annotation(
    tag_levels = list(c("(a)", "(b)")),
    theme = theme(plot.margin = margin(0, 0, 0, 0), )
  ) +
  guides(shape = "none") &
  theme(
    legend.position = "none",
    plot.margin = unit(c(0, 0, 0.0, 0), "lines"),
    plot.tag = element_text(
      family = family_wfu,
      size = 3 * text_size_wfu, face = "bold"
    )
  )

# save as a png
ggsave(
  filename = "wfu_potter_pedigree.png",
  plot = result,
  width = 9.5, height = 6, dpi = 300, units = "in"
)
```