

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

S. Mason Garrison <sup>1</sup>✉

<sup>1</sup> 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Published: unpublished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). Such complexity exposes the limitations of existing tools, which often struggle to handle large datasets and complex relationships. 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 specialized proprietary software (e.g., Progeny, GenoPro, Pedigree Viewer) or R packages like kinship2 (Sinnwell et al., 2014), pedtools (Magnus Dehli Vigeland, 2021a), 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 existing R packages focus on base graphics or simple ggplot2 implementations.

First, base R graphics lack the modular design and extensibility needed 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 existing 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 expose  
42 the full theming and layering capabilities familiar to users of ggplot2 (Hadley Wickham, 2016).  
43 In animal-focused workflows, rapid rendering often takes precedence over aesthetic flexibility,  
44 which can hinder interpretability in human-focused research.

45 Recent advances in behavior genetics and genetic epidemiology (Garrison et al., 2023; Lyu,  
46 Hunter, Good, Carroll, & Garrison, 2025; McArdle & McDonald, 1984) have introduced new  
47 visualization demands, particularly for extended families and highly interconnected pedigrees.  
48 Modern pedigree datasets can include hundreds or thousands of individuals across multiple  
49 generations, with overlapping relationships, consanguinity, remarriages, and twin sets that strain  
50 existing layout algorithms. Additionally, researchers increasingly need to overlay phenotypic  
51 information, genetic relatedness values, and model outputs onto pedigree plots to support  
52 interpretation and communication of their findings.

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

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

## 69 Software Architecture

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

## 80 Features

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

## 87 Core Functions

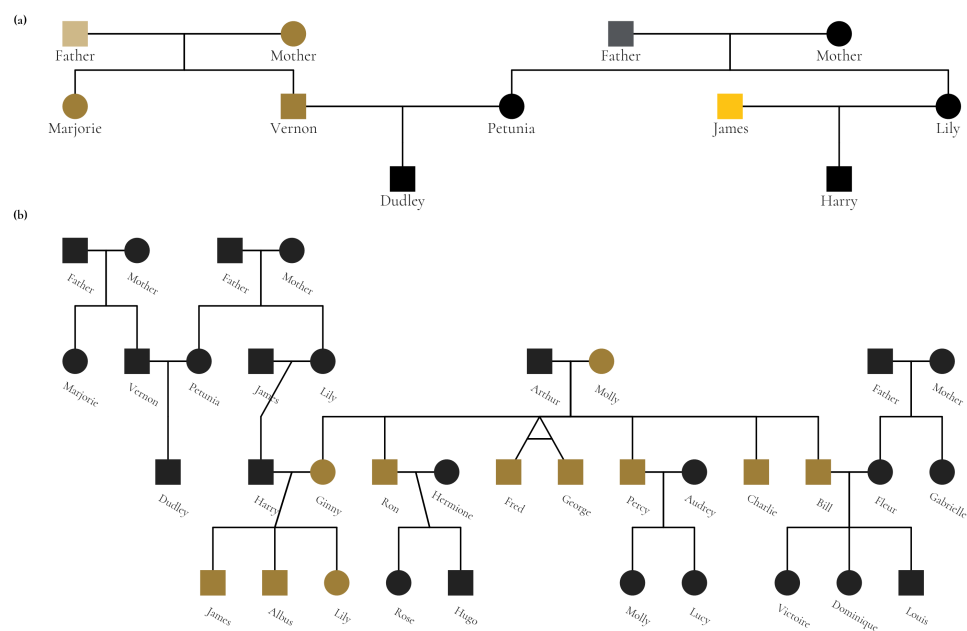
- 88     ▪ Data Standardization and Family Structure Analysis: `ggPedigree()` integrates with  
89       BGmisc functions like `ped2fam()` to organize individuals by family, `recodeSex()` to  
90       standardize sex coding, and `checkParentIDs()` to validate pedigree structures. The  
91       function handles complex scenarios including consanguineous relationships and individuals  
92       appearing in multiple pedigree positions.
- 93     ▪ Coordinate Calculation: `calculateCoordinates()` computes optimal positioning for  
94       individuals using algorithms adapted from `kinship2::align.pedigree`, with enhance-  
95       ments for large multi-generational pedigrees and complex family structures. The function  
96       returns coordinate grids that optimize spacing and minimize visual overlap. These steps  
97       are vectorized as much as possible to ensure computational efficiency, especially for large  
98       pedigrees.
- 99     ▪ Relationship Connection Mapping: `calculateConnections()` generates connection paths  
100       between family members, mapping parent-child, sibling, spousal, and twin relationships.  
101       The function determines midpoints for line intersections and handles overlapping con-  
102       nections with specialized curved segments. These calculations are optimized for large  
103       datasets by using vectorized operations rather than the loop-based approaches used in  
104       `kinship2`.
- 105     ▪ Layer-based Plot Construction: `ggPedigree()` constructs plots using `ggplot2` geome-  
106       try functions, returning standard `ggplot2` objects that integrate with existing R work-  
107       flows. `ggPedigreeInteractive()` extends plots into interactive `plotly` widgets with hover  
108       tooltips and dynamic exploration capabilities. A comprehensive config system allows  
109       customization of over 100 aesthetic and layout parameters.
- 110     ▪ Focal Individual Highlighting: Advanced functionality to highlight specific individuals and  
111       their relatives based on additive genetic, mitochondrial, or other relationship matrices.

## 112 Specific Visualization Functions

- 113     ▪ Pedigree Plotting: `ggPedigree()` creates static pedigree plots using `ggplot2`, supporting  
114       complex family structures, multiple generations, and customizable aesthetics. It can  
115       handle large pedigrees with thousands of individuals while maintaining clarity and  
116       readability.
- 117     ▪ Interactive Pedigree Visualization: `ggPedigreeInteractive()` generates interactive pedi-  
118       gree plots using `plotly`, allowing users to explore large pedigrees dynamically. Users can  
119       configure tooltip content to display individual IDs, names, phenotypic information, and  
120       other relevant data. It supports tooltips, zooming, and panning for detailed exploration  
121       of family structures.
- 122     ▪ Relatedness Matrix Heatmaps: `ggRelatednessMatrix()` creates customizable heatmap  
123       visualizations for genetic relatedness matrices with support for hierarchical clustering,  
124       interactive exploration, and seamless integration with BGmisc relatedness calculations.
- 125     ▪ Phenotype-Relatedness Analysis: `ggPhenotypeByDegree()` produces specialized plots  
126       for visualizing phenotypic correlations as a function of genetic relatedness, including  
127       confidence intervals and statistical summaries for quantitative genetic analysis.

## 128 Illustrative Example of `ggPedigree`

129 I illustrate some of the features by reproducing the figure from (M. D. Hunter et al., 2025), but  
130 restyled to conform with the Wake Forest Style Guidelines. As you can see, the `ggpedigree`  
131 package allows for complex family structures to be visualized in a clear and aesthetically  
132 pleasing manner.



I have combined two figures using patchwork (Pedersen, 2025) to show the pedigree of the Potter family. The first plot (a) highlights the individual mtDNA lines in the Dursley and Evans families, while the second plot (b) shows the entire family structure with mitochondrial relationships highlighted by a gradient color scale. Features such as focal individual highlighting, custom color scales, text labels, and fonts are easily implemented. The package supports both static and interactive plots, allowing users to explore large pedigrees dynamically.

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.

- 160 *Bioinformatics*, 34(6), 1069–1071. doi:[10.1093/bioinformatics/btx705](https://doi.org/10.1093/bioinformatics/btx705)
- 161 Daniel, S., & Sinnwell, J. (2024). Pedgene: Gene-level variant association tests for pedigree  
162 data. doi:[10.32614/CRAN.package.pedgene](https://doi.org/10.32614/CRAN.package.pedgene)
- 163 Garrison, S. M. (2024). Charting new paths in behavior genetics: Developing a comprehensive  
164 r visualization atlas. *Behavior Genetics* (Vol. 54, p. 488488). SPRINGER. Retrieved from  
165 <https://scholar.google.com/scholar?cluster=4710819130852314260&hl=en&oi=scholar>
- 166 Garrison, S. M., Hunter, M. D., Lyu, X., Trattner, J. D., & Burt, S. A. (2024). BGMisc: An r  
167 package for extended behavior genetics analysis. *Journal of Open Source Software*, 9(94).  
168 doi:[10.21105/joss.06203](https://doi.org/10.21105/joss.06203)
- 169 Garrison, S. M., Lyu, X., Hunter, M. D., Rodgers, J. L., Smith, K. R., Coon, H., & Burt, S.  
170 A. (2023). Analyzing extended cousin similarity to unravel the mystery of mtDNA and  
171 longevity. *Behavior Genetics*. doi:[10.1007/s10519-023-10156-9](https://doi.org/10.1007/s10519-023-10156-9)
- 172 Hunter, M. D., Garrison, S. M., Lyu, X., Good, R., & Burt, S. A. (2024). Tools for  
173 biometric modeling in large population databases. *Behavior Genetics* (Vol. 54, p.  
174 531531). SPRINGER. Retrieved from [https://scholar.google.com/scholar?cluster=](https://scholar.google.com/scholar?cluster=16878992125450714238&hl=en&oi=scholar)  
175 [16878992125450714238&hl=en&oi=scholar](https://scholar.google.com/scholar?cluster=16878992125450714238&hl=en&oi=scholar)
- 176 Hunter, M. D., Garrison, S. M., Lyu, X., Good, R., Carroll, S. L., & Burt, S. A. (2025). *Tracing*  
177 *the right path: Determination of large pedigree segmentation and relatedness. Revise and*  
178 *Resubmit at Behavior Genetics*.
- 179 Hunter, M., Garrison, S. M., Burt, S. A., & Rodgers, J. L. (2021). The Analytic Identification  
180 of Variance Component Models Common to Behavior Genetics. *Behavior Genetics*, 51(4),  
181 425–437. doi:[10.1007/s10519-021-10055-x](https://doi.org/10.1007/s10519-021-10055-x)
- 182 Lyu, S. A., X. and Burt, Hunter, M. D., Good, R., Carroll, S. L., & Garrison, S. M. (2025).  
183 Detecting mtDNA effects with an extended pedigree model: An analysis of statistical power  
184 and estimation bias. *Behavior Genetics*. doi:[10.1007/s10519-025-10225-1](https://doi.org/10.1007/s10519-025-10225-1)
- 185 Martin, G. R. (1997). *A game of thrones* (Vol. 1). Bantam.
- 186 Martin, G. R. R. (2018). *Fire & blood: 300 years before a game of thrones*. New York:  
187 Random House Worlds.
- 188 Martin, J., Wolak, M., Johnston, S., & Morrissey, M. (2025). *Pedtricks: Visualize, summarize*  
189 *and simulate data from pedigrees*. doi:[10.32614/CRAN.package.pedtricks](https://doi.org/10.32614/CRAN.package.pedtricks)
- 190 McArdle, J. J., & McDonald, R. P. (1984). Some algebraic properties of the reticular action  
191 model for moment structures. *British Journal of Mathematical and Statistical Psychology*,  
192 37, 234–251. doi:[10.1111/j.2044-8317.1984.tb00802.x](https://doi.org/10.1111/j.2044-8317.1984.tb00802.x)
- 193 McFarlane, S. E., Boutin, S., Humphries, M. M., McAdam, A. G., Gorrell, J. C., & Colt-  
194 man, D. W. (2015, January 21). Data from: Very low levels of direct additive ge-  
195 netic variance in fitness and fitness components in a red squirrel population. Dryad.  
196 doi:[10.5061/DRYAD.N5Q05](https://doi.org/10.5061/DRYAD.N5Q05)
- 197 McFarlane, S. E., Gorrell, J. C., Coltman, D. W., Humphries, M. M., Boutin, S., & McAdam,  
198 A. G. (2014). Very low levels of direct additive genetic variance in fitness and fitness  
199 components in a red squirrel population. *Ecology and Evolution*, 4(10), 1729–1738.  
200 doi:[10.1002/ece3.982](https://doi.org/10.1002/ece3.982)
- 201 McGoldrick, M., Gerson, R., & Petry, S. (2020). *Genograms: Assessment and Treatment*.  
202 Erscheinungsort nicht ermittelbar: W. W. Norton & Company.
- 203 Morrissey, M. B., & Wilson, A. J. (2010). pedantics: an r package for pedigree-based genetic  
204 simulation and pedigree manipulation, characterization and viewing. *Molecular Ecology*  
205 *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)

- 206 Pedersen, T. L. (2025). *Patchwork: The composer of plots*. doi:[10.32614/CRAN.package.patchwork](https://doi.org/10.32614/CRAN.package.patchwork)
- 207
- 208 Ranaweera, T., Makalic, E., Hopper, J. L., & Bickerstaffe, A. (2018). An open-source,  
209 integrated pedigree data management and visualization tool for genetic epidemiology.  
210 *International Journal of Epidemiology*, 47(4), 1034–1039. doi:[10.1093/ije/dyy049](https://doi.org/10.1093/ije/dyy049)
- 211 Shaw, P. D., Graham, M., Kennedy, J., Milne, I., & Marshall, D. F. (2014). Helium: Visualiza-  
212 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)
- 213
- 214 Sinnwell, J. P., Therneau, T. M., & Schaid, D. J. (2014). The kinship2 r package for pedigree  
215 data. *Human Heredity*, 78, 91–93. doi:[10.1159/000363105](https://doi.org/10.1159/000363105)
- 216 Vigeland, Magnus Dehli. (2021b). *Pedigree analysis in r*. London: Academic Press, an  
217 imprint of Elsevier. Retrieved from <https://shop.elsevier.com/books/pedigree-analysis-in-r/vigeland/978-0-12-824430-2>
- 218
- 219 Vigeland, Magnus Dehli. (2021a). Pedigree analysis in {r}.
- 220 Vigeland, Magnus D. (2022). QuickPed: An online tool for drawing pedigrees and analysing  
221 relatedness. *BMC Bioinformatics*, 23(1), 220. doi:[10.1186/s12859-022-04759-y](https://doi.org/10.1186/s12859-022-04759-y)
- 222 Wickham, Hadley. (2016). *ggplot2. Use R!* Cham: Springer International Publishing.  
223 doi:[10.1007/978-3-319-24277-4](https://doi.org/10.1007/978-3-319-24277-4)
- 224 Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L., François, R., Grolemund, G.,  
225 et al. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, 4(43), 1686.  
226 doi:[10.21105/joss.01686](https://doi.org/10.21105/joss.01686)
- 227 Zhang, R., Jia, G., & Diao, X. (2023). geneHapR: An r package for gene haplotypic statistics  
228 and visualization. *BMC Bioinformatics*, 24(1), 199. doi:[10.1186/s12859-023-05318-9](https://doi.org/10.1186/s12859-023-05318-9)

## 229 Appendix

- 230 Source code for the figures above is available in the vignettes/articles/\_paper.Rmd file.  
231 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,
    outline_multiplier = 1.5,
    segment_linewidth = 0.5,
    label_text_size = 12,
    label_text_color = text_color_wfu,

```

```

axis_text_color = text_color_wfu,
label_text_family = family_wfu,
focal_fill_include = TRUE,
label_nudge_y = -0.30,
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
    outline_multiplier = 1.5,
    segment_linewidth = 0.5,
    label_text_size = 9,
    label_text_family = family_wfu,
    label_text_color = text_color_wfu,
    axis_text_color = text_color_wfu,
    label_nudge_y = -0.3,
    label_nudge_x = .1,
    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 = "geom_text",
    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 = 2 * 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"
)
```