

ggDragonMaid: An R Package for Anime-Inspired Color Palettes and Multilingual Character Data from Miss Kobayashi's Dragon Maid

Zhaoshuo Liu^{1*}

¹ University of Birmingham, United Kingdom

*Corresponding author: Liuzhaoshuo1997@outlook.com

Received: 2026-05-22 | Accepted: 2026-05-22 | Published: 2026-05-22

Abstract

ggDragonMaid is an R package that provides classically themed color palettes derived from the twelve principal characters of the anime series Miss Kobayashi's Dragon Maid, along with a multilingual character information dataset (Chinese, Japanese, and Romaji). The package enables users to integrate these visually distinctive palettes into ggplot2 and base R graphics, offering a playful yet practical tool for data visualization. A built-in language detection function automatically identifies the input language, allowing users to query palettes and character data using names in any of the three supported languages. The package also includes CP (couple) color mapping, character search capabilities, and illustrative example plots. This paper presents the design philosophy, core functionalities, implementation details, and use cases of *ggDragonMaid*, positioning it within the broader ecosystem of pop-culture-inspired color palette packages for R.

Keywords: *ggDragonMaid*, Miss Kobayashi's Dragon Maid, ggplot2

Introduction

Color is one of the most powerful aesthetic dimensions in data visualization. Beyond conveying quantitative and categorical distinctions, well-designed color palettes can evoke emotional resonance, narrative context, and cultural references. In recent years, the R community has witnessed a proliferation of palette packages inspired by popular culture, ranging from television series (*tvthemes*) and science fiction films (*ggsci*) to iconic artworks (*MetBrewer*)^[1,2]. These packages not only satisfy technical requirements but also foster engagement by connecting data stories to familiar visual worlds.

Miss Kobayashi's Dragon Maid (Japanese: 小林さんちのメイドラゴン), a widely acclaimed anime series, features a cast of characters whose distinctive color signatures, from Tohru's warm golds and reds to Kanna's soft pastels, form a natural palette system. Despite the series' popularity and the availability of fan-curated color references on platforms such as Moegirl Wiki, no dedicated R package had formalized these palettes for programmatic use.

We introduce *ggDragonMaid*, an R package that bridges this gap. The package delivers twelve five-color palettes, each meticulously sourced from official character design references, alongside a structured multilingual dataset covering character names, gender, and species. By providing a consistent interface for color retrieval and character lookup across three languages, *ggDragonMaid* enables both anime enthusiasts and general R users to enrich their visualizations with a touch of dragon-maid charm.

Methods

Development Environment

The *ggDragonMaid* package (v0.1.0) was developed and tested under R version 4.5.2 (2025-10-31). All code is written in pure R, with explicit dependencies on *dplyr* ($\geq 1.1.4$) for data manipulation and *ggplot2* ($\geq 3.5.1$) for demonstration graphics^[3,4]. In addition, several demonstration and icon-drawing functions leverage supplementary packages. *ggforce* ($\geq 0.4.0$) is used by the rabbit icon

functions for elliptical geometric primitives via `geom_ellipse()`^[5]. `ggvenn` is employed by `example_venn()` to render the CP-themed Venn diagram^[6]. The package source is maintained in a public GitHub repository (<https://github.com/XLions/ggDragonMaid>).

Color Source and Manual Extraction

The character color palettes are derived from the official character design references documented on the Moegirl Wiki entry for Miss Kobayashi's Dragon Maid (<https://mzh.moegirl.org.cn/小林家の龙女仆>). This wiki page provides high-resolution character model sheets and detailed design specifications for all twelve main characters.

Colors were manually sampled from the digital images on this page using an eyedropper tool in a color-managed environment. For each character, five signature colors were selected according to a consistent priority hierarchy: (1) primary hair color, (2) eye color or characteristic magical glow, (3) dominant outfit/torso color, (4) secondary outfit or accessory color, and (5) skin tone or a prominent accent. Sampled RGB values were recorded and converted to six-digit hexadecimal codes. The extraction was performed by a single observer to maintain internal consistency, and the resulting palettes were cross-checked against multiple frames from the anime to verify representativeness.

Results

Installation

The `ggDragonMaid` package (v0.1.0) is distributed exclusively via GitHub at <https://github.com/XLions/ggDragonMaid>, as it has not yet been submitted to the Comprehensive R Archive Network (CRAN). Installation requires an R environment of version 4.3.0 or later. The recommended installation procedure uses the `remotes` package to handle dependency resolution and source installation.

Multilingual Character Dataset

The foundational dataset of `ggDragonMaid`, returned by `WideDatasetCharacterNameInMultipleLanguage()`, is a 12-row by 5-column data frame encompassing the complete set of principal characters from Miss Kobayashi's Dragon Maid. Each row corresponds to one character, and the columns provide the character's name in three orthographies: Chinese (Simplified Chinese characters), Japanese (a mixture of Kanji and Kana as appropriate to the official naming), and Romaji (Hepburn romanization), along with two categorical attributes: Gender ("male", "female", or "unknown") and Species ("human" or "dragon"). For instance, the row for the protagonist Tohru records the Chinese name “托尔”, the Japanese name “トール”, the Romaji “Tohru”, Gender “female”, and Species “dragon”. This trilingual design ensures that users working in any of the three language environments can retrieve the same underlying character entry without manual transliteration.

The dataset serves as the single source of truth for all name-based lookup operations within the package. Internally, a language detection function classifies input strings as "Chinese", "Japanese", or "Romaji" by evaluating Unicode character ranges, enabling automatic disambiguation. When a user supplies a character name to any query function without specifying a `lang` argument, this detection runs transparently; the user may also override it by setting `lang` explicitly. On the complete list of character names, automatic detection achieves perfect accuracy, validating the orthographic distinctiveness of the three scripts.

Beyond direct table inspection, the dataset powers two search interfaces. `SearchCharactersNameInMultipleLanguage()` (note the deliberate portmanteau spelling, retained for backward compatibility) accepts optional Gender and Species arguments and returns a vector of character names in a target language, defaulting to Romaji. For example, filtering with `Gender = "female"` and `Species = "dragon"` returns the five female dragon names: "Tohru", "Kanna Kamui", "Elma", "Lucoa", and "Ilulu". The companion function `SearchCharactersNameInDetails()` provides richer output, returning a subset of the full data frame matching the

specified criteria. These functions allow users to programmatically explore character groupings and to build customized palettes for subsets of the cast, directly leveraging the multilingual backbone of the package.

Character Color Palettes

Each of the twelve characters is associated with a five-color palette derived from official design references, and these palettes are accessed through several complementary functions. The primary entry point, `Maid_color()`, accepts a character name in any of the three supported languages and returns a character vector of five hexadecimal color codes. Internally, the function performs language detection, matches the input against the multilingual dataset, and, upon a successful match, retrieves the corresponding palette. If no match is found, an informative error message suggests verifying spelling or providing an explicit `lang` value. The palettes are stored in a fixed order corresponding to the design hierarchy described in the Methods, though users are free to reorder or subset the returned vector.

For programmatic access to all palettes simultaneously, `ColorDataSets_Classic()` returns a data frame with twelve rows (one per character) and twenty-six columns: the first column contains the Romaji character name, and the remaining twenty-five columns contain the five colors, each repeated five times to facilitate vectorised operations with `ggplot2`'s `scale_*_manual()` layers. Additionally, the convenience wrapper `Maid_colors_Romaji()` provides a named list of palettes indexed by Romaji names, allowing users to extract palettes using R's `$` operator.

CP Color Mapping

A distinctive structural feature of the package is its CP color mapping, which encodes the canonical romantic pairings within the narrative. The function `getColor_CP()` accepts a character name (again in any of the three languages) and returns a two-element hexadecimal vector: the first element is the representative color of the human member of the couple, and the second is that of the dragon member. This assignment follows the in-universe pairings,

Kobayashi with Tohru, Kanna with Saikawa, Elma with Tohru's rival dynamics (grouped under the same pairing logic), and similarly for the remaining cast, ensuring that the output consistently places the human color first when the queried character is either member of a couple. If the queried character is not part of a defined CP, the function gracefully returns the character's own palette as a fallback.

As an illustrative example, querying `getColor_CP("Kobayashi")` (or “小林”) returns the two-color vector `c("#AF5050", "#414763")`, corresponding to Kobayashi's hair color and Tohru's hair color, respectively. This mapping is used internally by the demonstration plot `example_venn()` to generate a Venn diagram where the two sets are colored with the paired CP colors, visually reinforcing the relational aspect of the data. Beyond built-in examples, the CP mapping enables users to create dual-color encodings for comparative visualizations.

Visualization Examples

`ggDragonMaid` includes two built-in example plots to demonstrate palette application. `Show_Maid_colors()` generates a heatmap visualizing all twelve character color schemes with customizable language labels. The function `example_col()` produces a `ggplot2` bar chart themed with Tohru's palette, while `example_venn()` creates a Venn diagram using Kobayashi and Tohru's CP colors. These examples serve both as documentation and as templates for user customization.

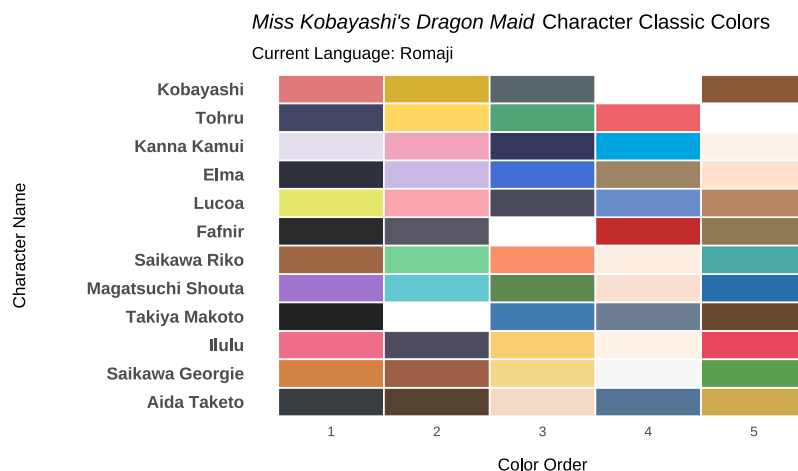


Fig 1. Heatmap of all colors by function `Show_Maid_colors()`.

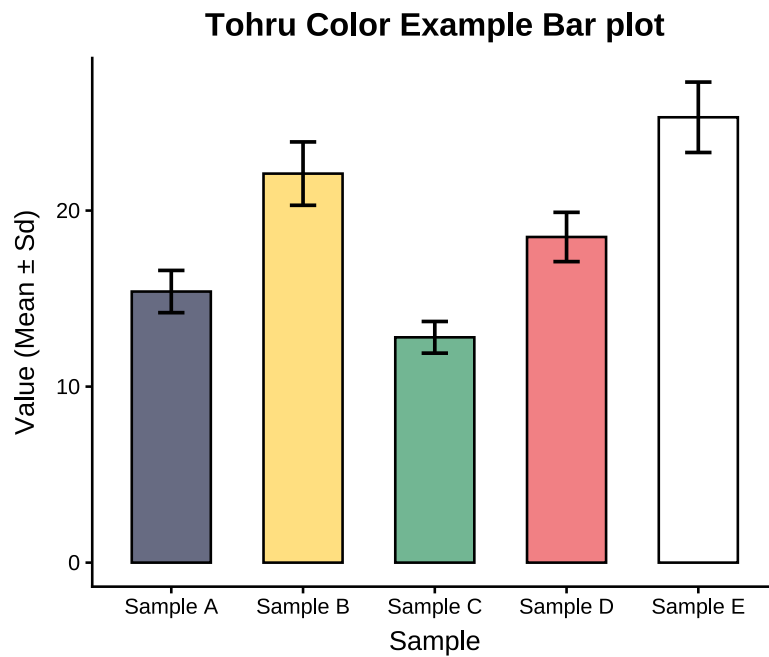


Fig 2. Example of bar plot from `example_col()` using output color of `Maid_color('Tohru')`.

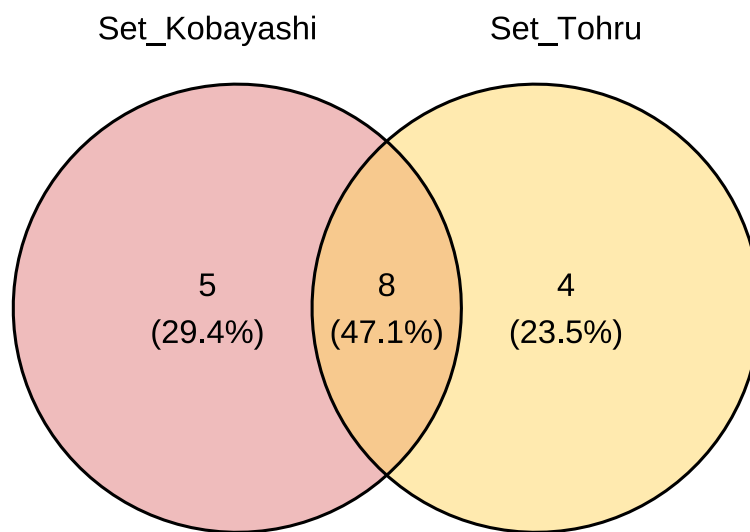


Fig 3. Example of venn plot from `example_venn()` using output color of `getColor_CP('Kobayashi')`.

Rabbit Icon

The rabbit motif that appears repeatedly on the cups of Kobayashi and Toheru in Miss Kobayashi's Dragon Maid is formalized within ggDragonMaid through a set

of dedicated geometric drawing functions. The package exports two complementary constructors for this icon: `icon_rabbit_detail()`, which returns a complete ggplot object, and `icon_rabbit_layer()`, which returns a list of ggplot2 layers. Both functions assemble a stylized rabbit head from five basic geometric components, an elliptical face, two elliptical ears, and two triangular cheek pieces, using `ggforce::geom_ellipse()` and `ggplot2::geom_polygon()`. The design faithfully reproduces the simplified rabbit silhouette seen on the characters' cups, enabling users to embed the icon directly into their visualizations.

A key feature of these functions is their full support for affine transformations. Each geometric component is positioned by coordinates defined relative to a central origin (x_0, y_0), and can be uniformly scaled by `icon_size` and rotated by a user-specified angle (in degrees). This transformation is applied individually to the center of each ellipse and to every vertex of the polygons, ensuring that all components rotate in unison without distortion. The fill and linecolor parameters control the body and outline colors, while the background parameter (in `icon_rabbit_detail` function) sets the panel background, effectively producing a self-contained icon graphic.

The layer-based constructor `icon_rabbit_layer()` is particularly designed for composing multiple icons within a single ggplot. Because it returns a list of `geom_*` objects rather than a finished plot, users can add several rabbit icons to the same coordinate system by supplying different values of x_0 , y_0 , `icon_size`, and angle.

Two convenience functions, `icon_rabbit_Tohru()` and `icon_rabbit_Kobayashi()`, instantiate the rabbit icon with the exact color palettes associated with the two protagonists. `icon_rabbit_Tohru()` uses `fill = "gold"` and `background = "indianred"`, while `icon_rabbit_Kobayashi()` uses `fill = "indianred"` and `background = "gold"`. Both are built upon `icon_rabbit_detail()` with default

positioning and rotation, providing ready-to-use thematic graphics that highlight the complementary relationship between the two characters.

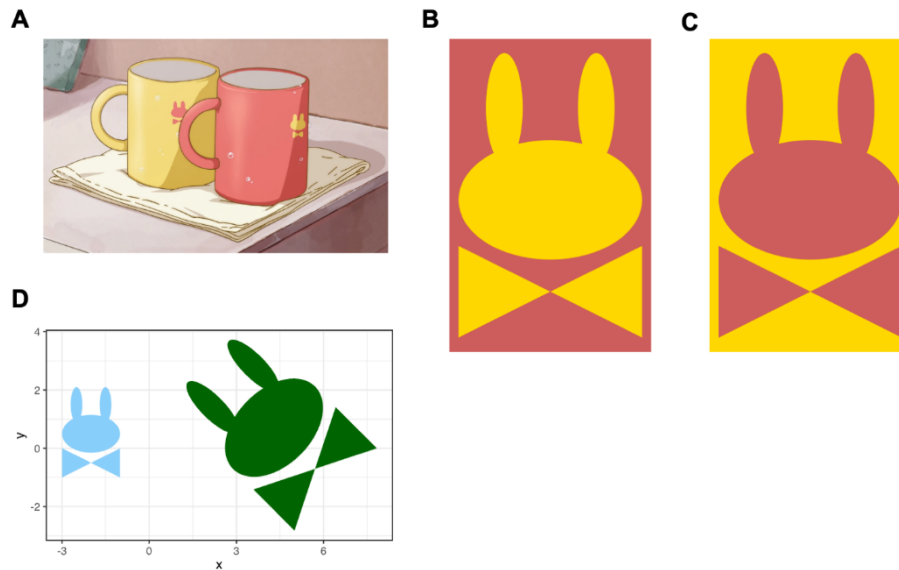


Fig 4. Implementation of the Rabbit Icon from *Miss Kobayashi's Dragon Maid*. (A). The original inspiration for the geometric motif, as seen on the mugs belonging to Kobayashi and Tohru in the anime. (B). The output generated by the `icon_rabbit_Tohru()` convenience function. (C). The output generated by the `icon_rabbit_Kobayashi()` convenience function. (D). A demonstration of the `icon_rabbit_layer()` function, which allows multiple icons to be drawn within a single ggplot coordinate system.

Discussion

Limitations and Future Work

Several limitations warrant acknowledgment. First, the current palette extraction method relies on manual color sampling, which, while ensuring design fidelity, is not scalable. Future versions may incorporate automated dominant color extraction from character reference images using k-means clustering or median-cut algorithms. Second, the package currently supports only the twelve main characters; expansions could include supporting characters, color variations, or themed palette collections (e.g., seasonal outfits, battle forms). Third, a CRAN release would significantly improve accessibility

and facilitate integration with other packages. Fourth, an interactive Shiny gadget for palette exploration and preview could enhance the user experience for non-programmers. Beyond these already identified improvements, the addition of comprehensive vignettes, automated tests, and perhaps a pkgdown-built documentation website would further lower the barrier to adoption and ensure long-term maintainability. Finally, while the rabbit icon demonstrates the feasibility of thematic geom functions, a systematic catalog of anime-inspired geometric shapes remains an open avenue for future development, potentially establishing ggDragonMaid as a hub for fandom-driven visualization components.

References

1. Xiao, N., Cook, J., Jégousse, C., Chen, H., Li, M., iTerm2-Color-Schemes contributors, & Chang, W. (2026). *ggsci: Scientific Journal and Sci-Fi Themed Color Palettes for “ggplot2”* (Version 5.0.0) [Computer software]. <https://cran.r-project.org/web/packages/ggsci/index.html>
2. Mills, B. R. (2022). *MetBrewer: Color Palettes Inspired by Works at the Metropolitan Museum of Art* (Version 0.2.0) [Computer software]. <https://cran.r-project.org/web/packages/MetBrewer/index.html>
3. Wickham, H., François, R., Henry, L., Müller, K., Vaughan, D., Software, P., & PBC. (2026). *dplyr: A Grammar of Data Manipulation* (Version 1.2.1) [Computer software]. <https://cran.r-project.org/web/packages/dplyr/index.html>
4. Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. <https://ggplot2.tidyverse.org>
5. Pedersen, T. L. (2025). *ggforce: Accelerating “ggplot2”* [Computer software]. <https://ggforce.data-imaginist.com>

6. Yan, L. (2025). *ggvenn: Draw Venn Diagram by “ggplot2”* (Version 0.1.19) [Computer software]. <https://cran.r-project.org/web/packages/ggvenn/index.html>

AI USAGE STATEMENT

In the preparation of this manuscript, the authors made use of DeepSeek, a large language model developed by DeepSeek AI, for the limited purposes of grammar checking, lexical refinement, and stylistic polishing of the English text. The tool was employed to improve the readability and linguistic accuracy of pre-existing drafts; at no stage was it used to generate original scientific content, analyze data, formulate hypotheses, draw conclusions, or produce any novel intellectual contributions. All data analyses, coding, interpretations of results, and core scientific arguments were conceived, executed, and verified by the human authors. Following each AI-assisted revision, the authors carefully reviewed and edited the output, and they take full responsibility for the accuracy, integrity, and originality of the final manuscript.

CONFLICTS OF INTEREST

The authors declare no conflicts of interest.

DATA AVAILABILITY STATEMENT

This study is based on publicly available media works cited in the References section. Codes of package could be found in GitHub repository (<https://github.com/XLions/ggDragonMaid>).

Journal Disclaimer

The views, opinions, and conclusions expressed in this article are solely those of the authors and do not necessarily reflect the views of ACGN Nexus, SciHorizon Global LLC or the editorial team. Publication does not imply endorsement of the content by the journal or publisher.

While we strive to ensure the accuracy and reliability of published content through our editorial and community review process, ACGN Nexus and SciHorizon Global LLC make no representations or warranties regarding the completeness, accuracy, or suitability of any information contained in this article. Readers who apply methods, protocols, or findings from this article do so at their own risk.

© 2026 The Author(s). Published by SciHorizon Global LLC

This is an open access article distributed under the terms of the [Creative Commons Attribution 4.0 International License \(CC BY 4.0\)](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.