

typshade

A Typst package for visualizing multiple-sequence alignments in bioinformatics.

Published: May 23, 2026 Updated: Aug 08, 2026

```
1MBO_A      . . . . . V L S E G E W Q L V L H V W A K V E A D V . A G H G Q D I L I R L F K S H P E T L E K F D R F . K . H 48
1HHO_A      . . . . . V L S P A D K T N V K A A W G K V G A H A . G E Y G A E A L E R M F L S F P T T K T Y F P H F . D L S 49
1HHO_B      . . . . . V H L T P E E K S A V T A L W G K V N V . . . D E V G G E A L G R L L V V Y P W T Q R F F E S F G D L S 49
2LHB_A      P I V D T G S V A P L S A A E K T K I R S A W A P V Y S T Y . E T S G V D I L V K F F T S T P A A . . . . . Q 49
1MBA_A      . . . . . S L S A A E A D L A G K S W A P V F A N K . N A N G L D F L V A L F E K F P D S A N F F A D F . K G K 49
1ECA_A      . . . . . M L D Q Q T I N I I K A T . V P V L K E H G V T I T T T F Y K N L F A K H P E V R P L F D M G . R Q E 49
consensus    ! * * * * * * * * * * ! * * * * * * * * * * ! * * * *
```

Typshade is a Typst package for visualizing multiple sequence alignments in bioinformatics.

It provides a Typst-native interface centered on `shade(...)`, offering a readable and composable way to render alignments, add annotations, and incorporate logos, structure tracks, and graph tracks.

Inspired by TeXshade, Typshade rethinks alignment visualization with a focus on clarity, composability, and a Typst-native user experience.

Why Typshade?

TeXshade is powerful, but its UI reflects TeX: many global commands, implicit state, and order-sensitive setup before the alignment is rendered. Typshade keeps the feature set while making the source read like a figure specification.

TeXshade style:

```
\begin{texshade}{alignment.msfc}
  \shadingmode[similar]{identical}
  \shadingcolors{blues}
  \residuesperline{45}
  \setends{1}{80..125}
  \showruler{top}{1}
  \rulersteps{10}
  \showconsensus{bottom}
  \showsequencelogo{top}
  \shaderegion{1}{NPA}{White}{BrickRed}
  \feature{top}{1}{NXX[ST]N}{box[Yellow]}{motif}
\end{texshade}
```

Typshade style:

```
#let alignment = read("alignment.msfc", encoding: none)

#shade(
  alignment,
  format: "msfc",
  figure: publication(
    similarity: "blues",
    region: "80..125",
    logo: "charge",
    motifs: (
      "NPA": (bg: "BrickRed", text: "active site"),
      "NXX[ST]N": "motif",
    ),
  ),
)
```

The lower-level helpers are still available through commands: when you need precise control, but new documents can usually start from the kind of figure you want: publication figure, motif map, structure map, or logo analysis.

Quick Start

```
#import "@preview/typshade:0.1.4": *

#let alignment = read("alignment.msfa", encoding: none)

#shade(
  alignment,
  format: "msf",
  theme: "screen",
  figure: motif-map(auto),
)
```

`read(..., encoding: none)` remains supported on Typst 0.15 and later. On Typst 0.15 or later, you can additionally pass a resolved project path and let Typshade read the source inside the package:

```
#shade(path("alignment.msfa"), format: "msf", figure: motif-map(auto))
```

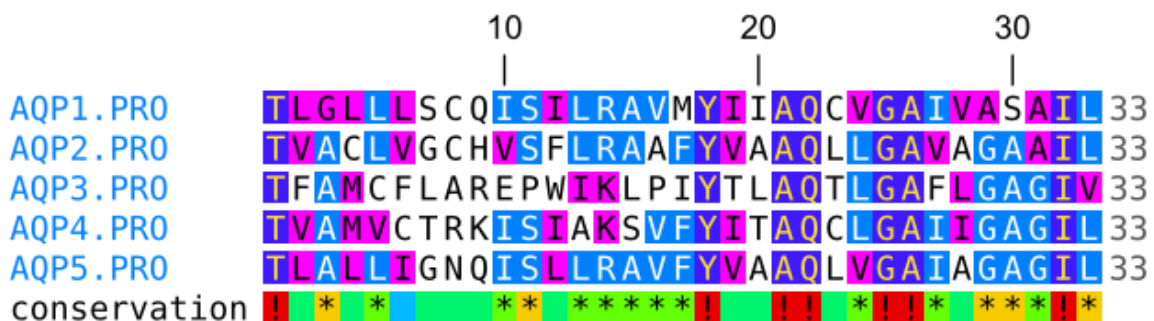
HTML Export

Typst HTML export is experimental and currently requires the HTML feature flag. Typshade detects the HTML target and wraps rendered alignment figures in a scrollable `html.frame` SVG, while `caption:` is emitted as semantic `figure/figcaption` HTML. Data-report helpers such as `selection-table` emit native HTML tables with explicit border styling.

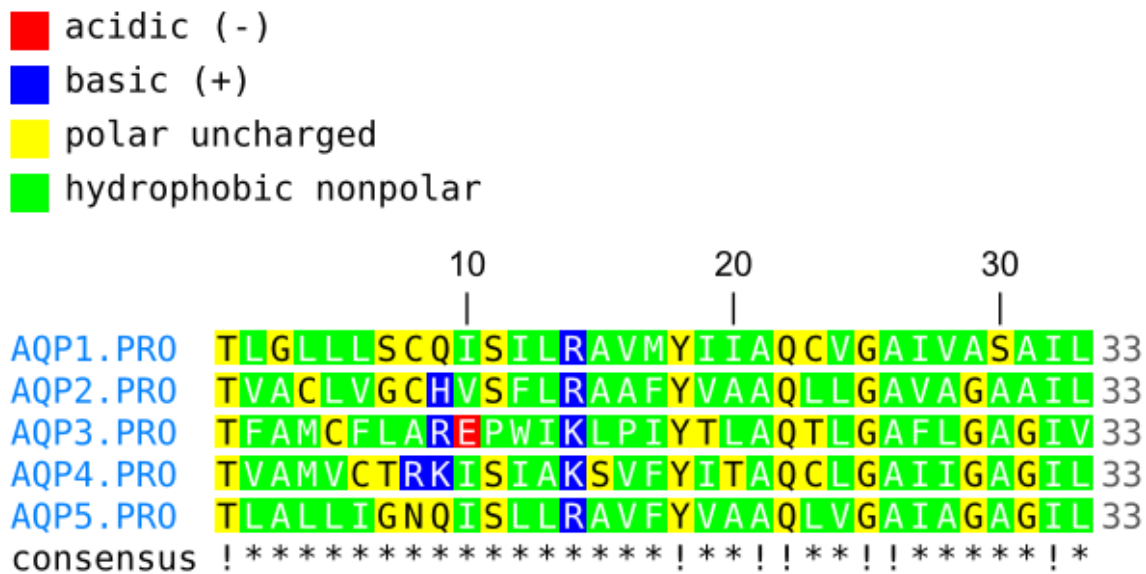
```
typst compile --features html --format html --root . your-document.typ out.html
```

Preview

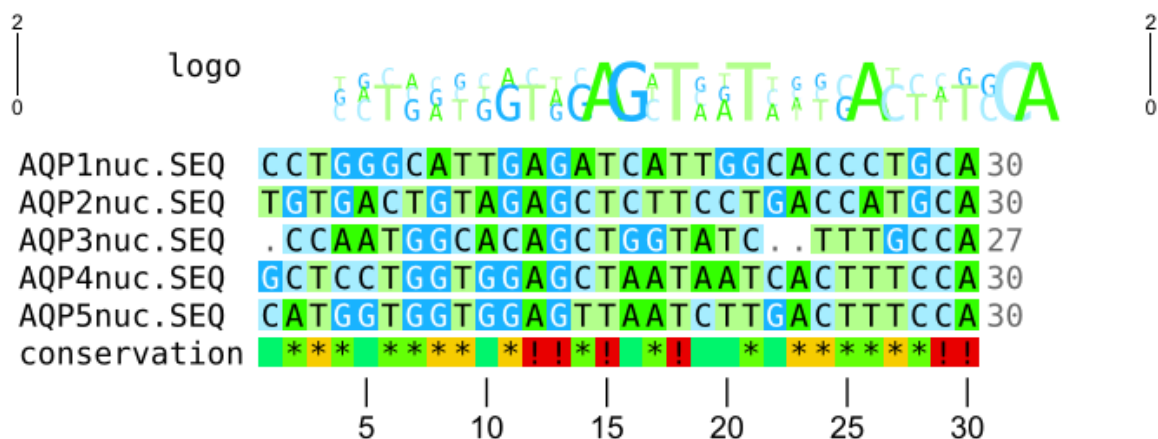
Protein alignment with similarity shading, motif annotations, a ruler, a conservation track, and a legend:



Protein alignment with hydrophathy-based functional coloring:



Nucleotide alignment with DNA coloring, a sequence logo, a conservation track, and a ruler:



Typst-Native Helpers

- `shade(...)`: Named-option alignment renderer for new documents.
- `figure::` Purpose-level recipe slot for complete figure designs.
- `publication`, `motif-map`, `structure-map`, `logo-analysis`, `overview`: High-level recipes.
- `similar`, `identical`, `diverse`, `functional`, `lines`, `window`, `ruler`, `consensus`, `logo`, `legend`: Compact command helpers; `lines(auto)` and `fit`: "container" fit the current Typst container.
- `auto-layout(...)`, `auto-page(...)`, and `fit`: "page": Typst-aware line length and page-aware block splitting for long figures.
- `color-scheme`, `scoring-mode`, `sequence-window`: Small, readable option helpers.
- `ruler-track`, `consensus-track`, `sequence-logo`, `subfamily-logo`, `legend-track`: Track helpers.
- `structure-tracks(...)`: Adds topology/secondary-structure tracks from sidecar files.
- `shade-preset("publication" | "overview" | "logo" | "functional" | "structure")`: Reusable command bundles.
- `shade-theme("classic" | "print" | "screen" | "warm" | "nature")` and `visual-theme(...)`: Color/style bundles.
- `highlight`, `tint`, `emphasize`, `mark`, `motif`, `graph`: Readable command builders for common annotations; graph metrics include conservation, entropy, gap-fraction, coverage, identity, hydrophobicity, molecular weight, and charge.

- `select-range`, `select-motif`, `select-metric`, `select-and`, `select-or`, `select-not`, and `select-pad`: Composable Selection DSL values for windows, highlights, marks, graphs, and analysis helpers.
- `cell-style(ctx => ...)`: Data-driven per-cell styling with Typst functions.
- `pdb-point`, `pdb-line`, `pdb-plane`: Safer constructors for PDB selections.
- `alignment-position("left" | "center" | "right")`: Overrides the default left-aligned block placement.
- `alignment-summary(...)`, `alignment-debug(...)`, `cell-inspect(...)`, and `selection-preview(...)`: In-document inspection helpers.
- `sequence-list(...)` and `selection-table(...)`: Typst tables for data-aware reports.
- `percent-identity(...)`, `percent-similarity(...)`, and `similarity-table(...)`: Pairwise identity/similarity analysis.
- `alignment-data(...)` and `parse-alignment(...)`: Data access helpers for custom Typst logic.
- HTML export support: alignment figures become scrollable SVG frames when compiled with Typst's experimental HTML target.

TeXshade To Typshade

TeXshade idea	Typshade API
<code>texshade</code> environment	<code>shade(read("alignment.msf", encoding: none), format: "msf", figure: publication(...))</code> , or <code>shade(path("alignment.msf"), format: "msf", ...)</code> on Typst 0.15+
<code>shadingmode</code> , <code>shadingcolors</code> , <code>threshold</code>	<code>similar</code> , <code>identical</code> , <code>diverse</code> , <code>functional</code> , or <code>scoring-mode</code> , <code>color-scheme</code> , <code>threshold</code>
<code>residuesperline</code> , <code>setends</code>	<code>lines</code> , <code>window</code> , <code>fit</code> , <code>auto-layout</code> , <code>auto-page</code> , or Selection DSL values with <code>sequence-window</code>
<code>shownames</code> , <code>shownumbering</code> , <code>showconsensus</code> , <code>showruler</code>	<code>names</code> , <code>numbers</code> , <code>consensus</code> , <code>ruler</code> , or the fine-grained track helpers
<code>showsequencelogo</code> , <code>showsubfamilylogo</code> , <code>showlegend</code>	<code>logo</code> , <code>subfamily-logo</code> , <code>legend</code>
<code>shaderegion</code> , <code>tintregion</code> , <code>emphregion</code> , <code>feature</code>	<code>highlight</code> , <code>tint</code> , <code>emphasize</code> , <code>mark</code> , <code>motif</code> , <code>graph</code>
<code>includeDSSP</code> , <code>includeSTRIDE</code> , <code>includeHMMTOP</code> , <code>includePHD*</code>	<code>structure-tracks</code> , <code>dssp-track</code> , <code>stride-track</code> , <code>hmmtop-track</code> , <code>phd-topology-track</code> , <code>phd-secondary-track</code>
font and spacing macros	<code>text-family</code> , <code>text-weight</code> , <code>text-posture</code> , <code>text-size</code> , <code>block-gap</code> , <code>feature-slot-space</code>

See `docs/documentation.typ` for the full guide and a larger correspondence table.

Smart Recipes

Recipes inspect the alignment before rendering. For example, `motif-map(auto)` detects common motifs for the sequence type, focuses the region around them, chooses a readable line length, adds conservation when useful, and enables a logo only when the figure stays readable. Override any option when you need exact control.

Fine-Grained Control

Macro-style command names are intentionally not part of the public API. Use Typst-shaped command helpers when you need detailed control:

- Scoring: `threshold`, `weight-table`, `set-weight`, `gap-penalty`, `residue-style`, `functional-group`.
- Tracks: `names-track`, `numbering-track`, `consensus-name`, `consensus-symbols`, `ruler-name`, `ruler-marker`, `logo-color`.
- Sequence layout: `start-number`, `sequence-length`, `domain`, `hide-sequence`, `sequence-order`, `separation-line`.
- Features and structure: `feature-rule`, `feature-text-label`, `backtranslation-label`, `show-structure-types`, `structure-appearance`, `stride-track`, `dssp-track`, `hmmtop-track`, `phd-topology-track`, `phd-secondary-track`.

- Typography and spacing: text-family, text-weight, text-posture, text-size, character-stretch, line-stretch, block-gap, feature-slot-space.

Custom Control

Use recipes when you know the purpose of the figure:

```
#let alignment = read("alignment.msfa", encoding: none)

#shade(
  alignment,
  format: "msfa",
  figure: publication(
    region: "80..125",
    logo: "charge",
    motifs: (
      "NPA": (bg: "BrickRed", text: "active site"),
      "NXX[ST]N": "glycosylation",
    ),
  ),
)
```

Use commands: when you want to assemble visible parts yourself:

```
#let alignment = read("alignment.msfa", encoding: none)

#shade(
  alignment,
  format: "msfa",
  preset: "publication",
  theme: "screen",
  commands: (
    similar(colors: "blues", threshold: 45),
    lines(45),
    window(1, "80..125"),
    ruler("top", sequence: 1, every: 10),
    consensus("bottom", name: "conservation"),
    logo("top", colors: "charge"),
    legend(),
    highlight(1, "NPA", bg: "BrickRed"),
    motif(1, "NXX[ST]N", text: "motif"),
    graph("bottom", 1, "all", "conservation", kind: "color", options: ("ColdHot")),
  ),
)
```

You can also mix recipe output with explicit, reproducible helper lists:

```
#let alignment = read("alignment.msfa", encoding: none)

#shade(
  alignment,
  format: "msfa",
  figure: publication(region: "80..125"),
  commands: (
    highlight(1, "NXX[DE][KR]XXQ", fg: "White", bg: "BrickRed"),
    sequence-logo(position: "top"),
  ),
)
```

License

This project is distributed under the GPL v2 License. See LICENSE for details.