Package ‘soilKey’

July 7, 2026

Type Package

Title Automated Soil Profile Classification per WRB 2022, 'SiBCS' 5 and USDA Soil Taxonomy 13

Version 0.9.184

Date 2026-07-07

Description Implements deterministic classification keys for the World Reference Base for Soil Resources 2022 (4th edition) and the Brazilian System of Soil Classification ('SiBCS', 5th edition). Provides a unified profile representation with explicit per-attribute provenance, multimodal extraction from field reports and photos via vision-language models, spatial priors from 'SoilGrids' and national soil maps, and gap-filling of soil attributes from Vis-NIR or MIR spectra via the Open Soil Spectral Library ('OSSL'). The taxonomic key itself is never delegated to a language model; LLMs are restricted to schema-validated extraction. Each classification result reports a key trace, a provenance-aware evidence grade, and ambiguities that further measurement would resolve.

License MIT + file LICENSE

URL `https://github.com/HugoMachadoRodrigues/soilKey`, `https://hugomachadorodrigues.github.io/soilKey/`

BugReports `https://github.com/HugoMachadoRodrigues/soilKey/issues`

Encoding UTF-8

LazyData true

LazyDataCompression xz

Depends R (>= 4.1)

Imports R6, data.table, yaml, cli, rlang

Suggests aqp, SoilTaxonomy, mpspline2, terra, foreign, sf, chromote, munsellinterpol, pls, prospectr, resemble, ellmer, httr, jsonlite, jsonvalidate, pdftools, magick, shiny (>= 1.7.0), DT, bslib, shinyWidgets, plotly, leaflet, htmltools, withr, DBI, RSQLite, base64enc, maps, testthat (>= 3.0.0), knitr, rmarkdown

Config/testthat/edition 3

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Hugo Rodrigues [aut, cre] (ORCID:
<`https://orcid.org/0000-0002-8070-8126`>)

Maintainer Hugo Rodrigues <`rodrigues.machado.hugo@gmail.com`>

Repository CRAN

Date/Publication 2026-07-07 19:40:02 UTC

Contents

`abrupt_textural_difference` *Abrupt textural difference (WRB 2022 Ch 3.2.1)*

Description

Sharp clay-content increase between two superimposed mineral layers meeting all of:

- underlying clay \>= 15% AND thickness \>= 7.5 cm;
- underlying starts \>= 10 cm below mineral soil surface;
- underlying has, vs overlying: 2x clay if overlying < 20%, OR \>= 20pp (absolute) more clay if overlying \>= 20%;
- transitional layer, if any, \<= 2 cm.

v0.3.3 enforces criteria 1, 2, 3. The transitional-layer check is deferred (the canonical horizon schema does not carry a "transitional" marker; it can be added later via boundary_distinctness inspection).

Usage

```
abrupt_textural_difference(pedon)
```

Arguments

`pedon`	A `PedonRecord`.

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`acrisol` *Acrisol RSG diagnostic (WRB 2022)*

Description

Tests whether a profile satisfies the Acrisol RSG criteria: an argic horizon with low-activity clay (CEC < 24 cmol_c/kg clay) AND low base saturation (BS < 50%) within at least one argic layer.

Usage

```
acrisol(pedon, max_cec = 24, max_bs = 50)
```

Arguments

`pedon`	A `PedonRecord`.
`max_cec`	Maximum CEC per kg clay (default 24).
`max_bs`	Maximum base saturation % (default 50).

Value

A `DiagnosticResult`.

References

IUSS Working Group WRB (2022), Chapter 5, Acrisols.

`aeolic_material` *Aeolic material (WRB 2022 Ch 3.3.1)*

Description

Wind-deposited material in the upper 20 cm: rounded matt-surfaced sand grains OR aeroturbation features, AND < 1% SOC in the upper 10 cm. v0.3.3 detects via `rock_origin == "aeolian"` OR `layer_origin == "aeolic"`.

Usage

```
aeolic_material(pedon)
```

Arguments

`pedon`	A `PedonRecord`.

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`albeluvic_glossae` *Albeluvic glossae (WRB 2022 Ch 3.2.2)*

Description

Tongues of bleached, coarser-textured material penetrating an argic horizon. v0.3.3 detects via designation pattern `glossic|albeluvic` on a layer that overlies an argic-horizon-passing layer.

Usage

```
albeluvic_glossae(pedon)
```

Arguments

`pedon`	A `PedonRecord`.

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`albic` *Albic horizon (WRB 2022)*

Description

A bleached eluvial horizon – claric material that has lost iron oxides and/or organic matter due to clay migration, podzolization, or redox under stagnant water. Diagnostic for parts of Podzols, Retisols and Planosols qualifiers.

Usage

```
albic(pedon, min_thickness = 1)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Minimum thickness in cm (default 1, per WRB 2022). The albic horizon has no canonical thickness gate; we keep a token min so that fully-NA layers don't pass.

Details

Sub-tests:

- `test_claric_munsell` – Munsell criteria of claric material (Ch 3.3.4).

Designation pattern `E` or `Eg` also serves as positive evidence when Munsell columns are missing (proxy path).

Value

A `DiagnosticResult`.

References

IUSS Working Group WRB (2022), Ch 3.1 – Albic horizon.

`alisol` *Alisol RSG diagnostic (WRB 2022)*

Description

argic + CEC >= 24 cmol_c/kg clay + Al saturation >= 50%.

Usage

```
alisol(pedon, min_cec = 24, min_al_sat = 50)
```

Arguments

`pedon`	A `PedonRecord`.
`min_cec`	Minimum CEC per kg clay (default 24).
`min_al_sat`	Minimum Al saturation % (default 50).

Value

A `DiagnosticResult`.

References

IUSS Working Group WRB (2022), Chapter 5, Alisols.

`andic_properties` *Andic properties (WRB 2022)*

Description

Tests for the andic property complex – volcanic-ash-derived allophanic / imogolitic / Al-humus material. Diagnostic of Andosols. Two alternative qualifying paths per WRB 2022 Ch 3.2:

1. **Al-Fe oxalate + low BD**: (Al_ox + 0.5*Fe_ox) >= `min_alfe` (default 2.0%) AND bulk_density <= `max_bd` (default 0.9 g/cm^3) on the same layer.
2. **Phosphate retention**: phosphate_retention_pct >= `min_p_retention` (default 70%).

Either path qualifies. The volcanic-glass criterion is the separate `vitric_properties` diagnostic; Andosols key on (andic OR vitric) at the RSG-gate level (`andosol`).

Usage

```
andic_properties(
  pedon,
  min_alfe = 2,
  max_bd = 0.9,
  min_p_retention = 70,
  min_oc_proxy = 4,
  max_bd_proxy = 0.9
)
```

Arguments

`pedon`	A `PedonRecord`.
`min_alfe`	Minimum (Al_ox + 0.5*Fe_ox) percent for the Al-Fe path (default 2.0).
`max_bd`	Maximum bulk density g/cm^3 for the Al-Fe path (default 0.9).
`min_p_retention`	Minimum phosphate retention % for the P path (default 70).
`min_oc_proxy`	Minimum SOC % for the v0.9.80 OC+BD proxy path (default 4.0). Only consulted when the proxy is enabled via `options(soilKey.andic_oc_bd_proxy = TRUE)`.
`max_bd_proxy`	Maximum bulk density g/cm^3 for the v0.9.80 OC+BD proxy path (default 0.9). Only consulted when the proxy is enabled.

Value

A `DiagnosticResult`.

v0.9.80 OC + BD proxy (opt-in)

Field-described volcanic-ash soils (e.g.\ AfSP, KSSL/NASIS, SOTER) routinely lack oxalate Al/Fe and phosphate retention measurements, so the canonical paths return `NA` and Andosols cascade to other RSGs. The genetic signature is still detectable from coarser data: very high SOC (>= 4-5%) plus low bulk density (<= 0.9 g/cm^3) typical of allophanic / Al-humus complexation.

With `options(soilKey.andic_oc_bd_proxy = TRUE)` the function adds a third path that fires when both canonical paths fail and the surface horizon shows `oc_pct >= min_oc_proxy` AND `bulk_density_g_cm3 <= max_bd_proxy` (or OC alone >= 5% when BD is missing). Default is `FALSE` (canonical behaviour preserved).

v0.9.85 proxy contiguous-layer extension (opt-in)

When `options(soilKey.andic_oc_bd_proxy_extend = TRUE)` (only meaningful with `soilKey.andic_oc_bd_proxy = TRUE`), iteratively extend the proxy layers to include contiguous deeper layers whose `oc_pct >= min_oc_proxy / 2` AND whose `bulk_density_g_cm3` is missing OR `<= max_bd_proxy + 0.15`. The extension stops at the first horizon failing either constraint, so a ferralic / argic subsoil cannot accidentally inflate the andic thickness. Default is `FALSE` – canonical proxy behaviour preserved.

References

IUSS Working Group WRB (2022), Chapter 3, Andic properties.

`andosol` *Andosol RSG gate (WRB 2022 Ch 4, p 104)*

Description

WRB-canonical: layer(s) with *andic* OR *vitric* properties, combined thickness \>= 30 cm within 100 cm starting \<= 25 cm; OR \>= 60% of the entire soil thickness when a limiting layer starts 25-50 cm. Plus: no argic, ferralic, petroplinthic, pisoplinthic, plinthic or spodic horizon \<= 100 cm (unless buried below 50 cm).

Usage

```
andosol(
  pedon,
  min_thickness = 30,
  max_top_cm = 25,
  buried_below_cm = 50,
  strict = NULL
)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Numeric threshold or option (see Details).
`max_top_cm`	Numeric threshold or option (see Details).
`buried_below_cm`	Numeric: layers of the exclusion diagnostics whose top_cm \>= this depth are treated as buried and do NOT exclude the Andosol (default 50, per WRB 2022 Ch 4 p 104).
`strict`	Logical or `NULL`. When `NULL` (default) it resolves via `getOption("soilKey.rsg_strict", FALSE)`. `TRUE` disables the buried-exclusion tolerance.

Details

v0.3.4 enforces (1) andic OR vitric AND (2) combined thickness \>= 30 cm starting in the upper 25 cm AND (3) the negative-list exclusions on argic / ferralic / plinthic / spodic.

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

v0.9.85 buried-exclusion fix

WRB 2022 Ch 4 p 104 specifies the Andosol exclusion list (argic / ferralic / petroplinthic / pisoplinthic / plinthic / spodic) as "<= 100 cm *unless buried below 50 cm*". The earlier implementation excluded an Andosol whenever any of those diagnostics passed anywhere in the profile, including on layers starting deeper than 50 cm – which mis-fires on AfSP Andosol references like `CM W3_0047`, where an argic layer at 56-72 cm wrongly excluded the andic surface stack. v0.9.85 restricts the exclusion check to layers starting <= 50 cm: a buried argic / ferralic / plinthic / spodic at deeper levels no longer disqualifies the surface andic stack from Andosol.

Tier-3 strict mode (v0.9.98)

With `strict = TRUE` the v0.9.85 buried-exclusion tolerance is switched off: *any* argic / ferralic / plinthic / spodic horizon anywhere in the profile excludes the Andosol, regardless of depth.

---

annotate_wrb_from_usda

Annotate KSSL/NASIS pedons with a derived WRB Reference Soil Group

Description

Applies `usda_to_wrb_rsg` to each pedon's USDA classification (preserved as `site$reference_usda` + `site$reference_usda_suborder` by `load_kssl_pedons_gpkg`) and writes the result to `site$reference_wrb_from_usc` – a "best-guess" expected WRB label for benchmark comparison.

Usage

```
annotate_wrb_from_usda(pedons)
```

Arguments

`pedons`	List of `PedonRecord` objects.

Details

Pedons that already have `site$reference_wrb` populated (e.g.\ from external sources) are left untouched.

Value

The same list, with `site$reference_wrb_from_usda` populated where USDA classification is present.

`anthraquic` *Anthraquic horizon (WRB 2022): puddled-rice / paddy plough layer. v0.3.3 detects via designation pattern* `Apl|Ap|Hh`.

Description

Anthraquic horizon (WRB 2022): puddled-rice / paddy plough layer. v0.3.3 detects via designation pattern `Apl|Ap|Hh`.

Usage

```
anthraquic(pedon, min_thickness = 20, max_top_cm = 50)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Numeric threshold or option (see Details).
`max_top_cm`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`anthric_horizons` *Anthric horizons (WRB 2022)*

Description

Tests for any of five anthropogenic surface horizons recognised by WRB 2022 (hortic, irragric, plaggic, pretic, terric). Diagnostic of Anthrosols. Two alternative paths qualify:

1. **Designation**: any layer's designation contains one of `hortic|irragric|plaggic|pretic|terric`.
2. **Property-based**: a surface layer (top_cm <= 5) at least `min_thickness_cm` cm thick (default 20) with elevated dark colour (Munsell value moist <= `max_munsell_value`, default 4) AND elevated plant-available P (`p_mehlich3_mg_kg` >= `min_p_mg_kg`, default 50).

Either path qualifies.

Usage

```
anthric_horizons(
  pedon,
  min_thickness_cm = 20,
  min_p_mg_kg = 50,
  max_munsell_value = 4
)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness_cm`	Minimum thickness for the property-based path (default 20).
`min_p_mg_kg`	Minimum plant-available P (Mehlich 3, mg/kg) for the property-based path (default 50).
`max_munsell_value`	Maximum Munsell value moist for the property-based path (default 4).

Value

A `DiagnosticResult`.

References

IUSS Working Group WRB (2022), Chapter 5, Anthrosols.

`apply_soilgrids_depth_prior` *Fill missing horizon attributes from a SoilGrids depth prior*

Description

For each horizon and each requested attribute, interpolates the value at the horizon's mid-depth from the six standard SoilGrids 2.0 depth slices (0-5, 5-15, 15-30, 30-60, 60-100, 100-200 cm) and writes it into the pedon with `source = "inferred_prior"`. Existing values are preserved unless `overwrite = TRUE`; the `PedonRecord` authority order means a SoilGrids prior can never silently displace a measured, spectra-predicted or VLM-extracted value.

Usage

```
apply_soilgrids_depth_prior(
  pedon,
  attrs = NULL,
  depth_profiles = NULL,
  overwrite = FALSE
)
```

Arguments

`pedon`	A `PedonRecord` with at least one horizon. For the live fetch it must also carry `site$lat` and `site$lon`.
`attrs`	Character vector of horizon columns to fill. Defaults to all SoilGrids-backed attributes: `clay_pct`, `sand_pct`, `silt_pct`, `ph_h2o`, `oc_pct`, `cec_cmol`.

`depth_profiles`	Optional named list mapping an attribute to a numeric vector of six slice values (0-5 ... 100-200 cm). When supplied the SoilGrids network call is skipped entirely – this is the path the test suite and offline users take.
`overwrite`	If `FALSE` (default) only `NA` cells are filled. If `TRUE`, every requested cell is overwritten (subject to the provenance authority order).

Details

This is the depth-resolved companion to `spatial_prior_soilgrids` (which returns a site-level RSG probability vector, not horizon attributes), and the attribute-fill stage of `classify_from_photos`.

Value

Invisibly, the mutated pedon. An attribute `"soilgrids_depth_fill"` on the return value records how many cells were filled.

Examples

```
## Not run:
p <- make_cambisol_canonical()
p$horizons$clay_pct <- NA_real_
# Offline: supply the six-slice profiles directly.
apply_soilgrids_depth_prior(
  p, attrs = "clay_pct",
  depth_profiles = list(clay_pct = c(18, 20, 24, 28, 30, 30)))

## End(Not run)
```

`apply_spectral_preprocessing` *Apply a step-by-step Vis-NIR / MIR preprocessing pipeline*

Description

Composes the canonical soil-spectroscopy sequence, each step optional and applied in this fixed scientific order: reflectance → (absorbance) → (Savitzky-Golay smoothing) → (Savitzky-Golay 1st or 2nd derivative). Each Savitzky-Golay pass trims `(window - 1) / 2` columns from each edge; the wavelength axis is trimmed to match and returned so callers can plot the treated spectrum on the correct axis.

Usage

```
apply_spectral_preprocessing(
  X,
  wavelengths = NULL,
  absorbance = FALSE,
  sg_smooth = FALSE,
  sg_derivative = 0L,
```

```
  window = 11L,
  poly = 2L
)
```

Arguments

`X`	Numeric matrix (rows = samples/horizons, columns = wavelengths) or a numeric vector (treated as one sample).
`wavelengths`	Optional numeric wavelength axis. Defaults to the numeric part of `colnames(X)`, else `1:ncol(X)`.
`absorbance`	Logical; apply $A = \log_{10}(1/R)$.
`sg_smooth`	Logical; apply Savitzky-Golay smoothing (`m = 0`).
`sg_derivative`	Integer `0`, `1` or `2`; Savitzky-Golay derivative order (`0` = none).
`window`	Odd Savitzky-Golay window (default `11`); coerced to a valid odd value in `[3, ncol)`.
`poly`	Savitzky-Golay polynomial order (default `2`); clamped to `[1, window - 1]`.

Details

The transform is robust: reflectance that looks like a percentage (maximum > `1.5`) is rescaled to a 0–1 fraction before the absorbance log, values are clamped away from zero to avoid `log(0)`, and a Savitzky-Golay step that cannot fit the requested window into the available wavelengths is skipped (recorded in `steps`) rather than erroring.

Value

A list with `X` (the treated numeric matrix, wavelength column names trimmed to match), `wavelengths` (numeric) and `steps` (an ordered character vector describing the transforms actually applied, starting with `"Reflectance"`).

See Also

`preprocess_spectra`

Examples

```
X <- matrix(seq(0.1, 0.5, length.out = 3 * 60), nrow = 3, byrow = TRUE)
colnames(X) <- seq(400, 2400, length.out = 60)
res <- apply_spectral_preprocessing(X, absorbance = TRUE,
                                    sg_smooth = TRUE, sg_derivative = 1L)
res$steps          # ordered treatment labels
dim(res$X)         # columns trimmed by the two SG passes
```

`arenic_texture` *Arenic texture (WRB 2022)*

Description

Tests whether the upper 100 cm is uniformly coarser than sandy loam (i.e., `silt + 2 * clay < 30` in every layer). Diagnostic of Arenosols.

Usage

```
arenic_texture(pedon, max_top_cm = 100, engine = NULL)
```

Arguments

`pedon`	A `PedonRecord`.
`max_top_cm`	Maximum top depth (cm) of layers to be tested (default 100, per WRB 2022).
`engine`	One of `"soilkey"` (default; strict WRB sand threshold via `test_coarse_texture_throughout`) or `"aqp"` (LUCAS-friendly fallback: passes when sand >= 70\ reads `getOption("soilKey.diagnostic_`

Details

Sub-test: `test_coarse_texture_throughout`.

v0.3 limitations: WRB 2022 Arenosol also requires that no other diagnostic horizon (argic, ferralic, etc.) is present, but those exclusions happen at the key level via canonical RSG order.

Value

A `DiagnosticResult`.

References

IUSS Working Group WRB (2022), Chapter 5, Arenosols.

`argic` *Argic horizon (WRB 2022)*

Description

Tests whether any horizon meets the argic horizon criteria per Chapter 3 of the WRB 2022 (4th edition). Argic is a subsurface horizon with distinctly higher clay content than the overlying horizon, qualified by three depth-conditional clay-increase rules; it must also have texture of sandy loam or finer, satisfy a minimum thickness, and not exhibit albeluvic glossic features (which would direct the profile to the Retisol path).

Usage

```
argic(
  pedon,
  min_thickness = 7.5,
  system = c("wrb2022", "usda"),
  engine = NULL,
  require_t = NULL
)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Minimum thickness in cm (default 7.5).
`system`	One of `"wrb2022"` (default) or `"usda"`. Selects the clay-increase threshold set: WRB uses 6/1.4/20 pp/ratio/pp; KST 13ed uses 3/1.2/8 (looser). See `test_clay_increase_argic` for the table.
`engine`	v0.9.63+. One of `"soilkey"` (the hand-coded path, default for back-compat) or `"aqp"` (canonical NRCS dispatch via `aqp::getArgillicBounds`). When `NULL` (the new default) the function reads `getOption("soilKey.diagnostic_engine", "soilkey")` so a global `options(soilKey.diagnostic_engine = "aqp")` flips every `argic()` call without modifying call sites. See `argic_aqp`.
`require_t`	v0.9.63+. Forwarded to `aqp::getArgillicBounds` when `engine = "aqp"`: `TRUE` requires a "t" suffix in the horizon designation (the strict KST 13ed text); `FALSE` accepts argic by clay-increase alone (more permissive on data-sparse profiles). `NULL` (default) auto-picks: `TRUE` for `system = "usda"`, `FALSE` for `system = "wrb2022"`. Ignored when `engine = "soilkey"`.

Details

Sub-tests called (each a list with `passed`, `layers`, `missing`, `details`, `notes`):

- `test_clay_increase_argic` – the three-pronged WRB 2022 clay-increase rule.
- `test_minimum_thickness` – thickness >= 7.5 cm (configurable via `min_thickness`).
- `test_texture_argic` – texture of sandy loam or finer (`silt + 2 * clay >= 30`).
- `test_not_albeluvic` – excludes profiles with glossic tongues (Retisol path).

v0.1 limitations: clay-increase distance (<= 30 cm vertical, or <= 15 cm with abrupt textural change) is not yet enforced; that is scheduled for v0.2 and depends on horizon boundary descriptions.

Value

A `DiagnosticResult`.

References

IUSS Working Group WRB (2022). *World Reference Base for Soil Resources*, 4th edition. International Union of Soil Sciences, Vienna. Chapter 3 – Argic horizon.

`argic_aqp` *Argic / argillic horizon via aqp::getArgillicBounds()*

Description

Wraps `aqp::getArgillicBounds()` (Beaudette et al.) in soilKey's `DiagnosticResult` contract. The aqp implementation is the canonical NRCS R port and uses the tiered USDA-NRCS clay-increase thresholds:

- Eluvial clay < 15\
- Eluvial clay 15-40\
- Eluvial clay \>= 40\

(vs. soilKey's hand-coded `argic` which uses the WRB 6/1.4/20 thresholds). For BDsolos / FEBR / KSSL profiles the aqp rule is closer to KST 13ed and BDsolos field practice.

Usage

```
argic_aqp(pedon, require_t = FALSE, ...)
```

Arguments

`pedon`	A `PedonRecord`.
`require_t`	Whether to require an explicit "t" suffix in the horizon designation (default `FALSE` for BDsolos / FEBR; `TRUE` matches the strict KST 13ed text).
`...`	Reserved for future arguments.

Details

By default aqp requires a "t" suffix in the horizon designation (`require_t = TRUE`); we expose this so callers can be permissive on datasets where designation is missing or non-conforming (BDsolos exports often drop the "t").

Value

A `DiagnosticResult` with `name = "argic_aqp"`. `$layers` are the row indices of horizons in the argillic / argic depth interval. `$evidence` carries the raw aqp `c(ubound, lbound)` bounds for traceability.

See Also

`argic` (soilKey hand-coded; WRB 6/1.4/20), `aqp::getArgillicBounds`.

`argic_with_strong_clay_films` *Test whether a pedon's argic horizon has strong clay films*

Description

Wraps `argic()` and inspects the `clay_films_amount` field at the argic-passing layers. Returns a structured result that `B_latossolico()` uses to decide whether the SiBCS Cap 18 strong-films exclusion fires.

Usage

```
argic_with_strong_clay_films(pedon)
```

Arguments

`pedon`	A `PedonRecord`.

Value

A list with:

- `passed` – logical, `TRUE` only when argic passes AND at least one argic-passing layer has a strong (*comum* / *abundante*) film qualifier.
- `layers` – integer vector of argic-passing layer indices (empty when `passed` is `FALSE`).
- `argic` – the underlying `DiagnosticResult` from `argic()`.
- `films` – character vector of the `clay_films_amount` values at the argic-passing layers.

`argillic_clay_films_test` *Test for clay-illuviation evidence (KST 13ed Ch 3 p 4)*

Description

KST 13ed argillic horizon requires "evidence of illuvial accumulation of clay" alongside the clay-increase rule. Acceptable evidence:

- oriented clays bridging sand grains in >= 1% of the horizon;
- clay films lining pores or coating ped faces;
- lamellae more than 5 mm thick.

Usage

```
argillic_clay_films_test(pedon)
```

Arguments

`pedon`	A `PedonRecord`.

Details

This test reads three complementary slots, in order of evidence strength:

1. `pedon$site$nasis_diagnostic_features` – the NASIS `pediagfeatures.featkind` vector. The surveyor's explicit "Argillic horizon" entry directly confirms clay-illuviation evidence (~13 500 entries in the 2021 NASIS snapshot). Strongest evidence.
2. `pedon$horizons$clay_films_amount` – per-horizon clay-film abundance derived from NASIS `phpvsf`. Values: `"few"`, `"common"`, `"many"`, `"continuous"`. Direct measurement.
3. `pedon$horizons$designation` containing a 't' master suffix (e.g. `Bt`, `Btk`, `Btx`, `Bt1`, `2Bt`). v0.9.28: the pedologist who wrote that designation explicitly identified the horizon as clay-illuvial – per KST 13ed Ch 18, the 't' suffix means "accumulation of silicate clay" – so it counts as positive evidence even when NASIS records are absent. This unlocks the KST 13ed argillic thresholds for the ~47 pediagfeatures and phpvsf records.

Any of the three sources counts as positive evidence (logical OR). `passed = NA` when none is populated AND no horizon designation field is present at all (lab-only loaders without horizon descriptions). `passed = FALSE` when designations exist but none has a 't' suffix and NASIS slots are empty.

Value

A `DiagnosticResult`.

References

Soil Survey Staff (2022), Keys to Soil Taxonomy 13th ed., Ch. 3, argillic horizon (clay-illuviation criteria, p. 4); Ch. 18, master horizon symbols (`t`: silicate-clay accumulation, p. 332).

`artefacts` *Artefacts (WRB 2022 Ch 3.3.2)*

Description

Per the canonical definition: human-made / human-altered / human- excavated material. v0.3.3 returns the layers where `artefacts_pct >= 1`.

Usage

```
artefacts(pedon, min_pct = 1)
```

Arguments

`pedon`	A `PedonRecord`.
`min_pct`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`as_aqp` *Convert one or more PedonRecord objects to an aqp SoilProfileCollection*

Description

Builds a `aqp::SoilProfileCollection` from one `PedonRecord` or a list of them. Standard soilKey columns (`top_cm`, `bottom_cm`, `designation`, `clay_pct`, `sand_pct`, `silt_pct`) are renamed to aqp's canonical convention (`top`, `bottom`, `name`, `clay`, `sand`, `silt`). All other columns are passed through unchanged. Site-level slots (`lat`, `lon`, `country`, `parent_material`, `reference_*`, `nasis_diagnostic_features`, etc.) are attached to the SPC's site table.

Usage

```
as_aqp(x)
```

Arguments

`x`	A `PedonRecord` or a list of them.

Details

Requires the aqp package, listed in Suggests; the function raises a clear error if aqp is not installed.

Value

A `aqp::SoilProfileCollection`.

See Also

`from_aqp`, the inverse conversion.

Examples

```
## Not run:
library(soilKey)
library(aqp)

pedons <- list(make_ferralsol_canonical(), make_luvisol_canonical())
spc <- as_aqp(pedons)
length(spc)         # 2 profiles
aqp::horizons(spc)  # one row per horizon, aqp-named columns

## End(Not run)
```

`attach_lucas_spectra` *Attach LUCAS 2018 Vis-NIR spectra to a list of PedonRecord objects*

Description

Joins the LUCAS Soil 2018 Spectral Library (separate ESDAC release, ~83 GB) onto the pedons returned by `load_lucas_soil_2018`, by matching the LUCAS `POINT_ID` of the spectra against `pedon$site$id`. Each matched pedon gets `$spectra$vnir` populated as a numeric matrix (rows = horizons, cols = wavelengths).

Usage

```
attach_lucas_spectra(
  pedons,
  spectra,
  point_id_col = "POINT_ID",
  verbose = TRUE
)
```

Arguments

`pedons`	List of `PedonRecord` objects.
`spectra`	A wide or long `data.frame` as described above.
`point_id_col`	Name of the LUCAS point-id column in `spectra`. Default `"POINT_ID"`.
`verbose`	If `TRUE` (default), reports the join hit rate.

Details

Two input shapes are accepted:

- A wide `data.frame` keyed by an integer `POINT_ID` column with one column per wavelength (column names parseable as numeric nm). One row per LUCAS point.
- A long `data.frame` with columns `POINT_ID`, `wavelength_nm`, `reflectance`.

Spectra are attached only to the topsoil horizon (row 1); the subsoil horizon (if any) is left without spectra. After this call, `benchmark_lucas_2018(..., fill_topsoil_from = "spectra",ossl_models = ...)` feeds the spectra through `predict_from_spectra` (v0.9.46) to fill any chemistry / texture gap not already populated by SoilGrids.

Value

The list of pedons (mutated in place; returned invisibly).

See Also

`predict_from_spectra`, `predict_munsell_from_spectra`, `load_lucas_soil_2018`.

`audit_argic_strong_films` *Audit the strong-clay-films exclusion across a list of pedons*

Description

Applies `argic_with_strong_clay_films()` to every pedon in pedons and returns a per-pedon table summarising how the v0.9.61 `B_latossolico()` latossolic-vs-argic rule resolves on the benchmark sample.

Usage

```
audit_argic_strong_films(pedons, reference_filter = NULL)
```

Arguments

`pedons`	List of `PedonRecord` objects.
`reference_filter`	Optional regex applied to `p$site$reference_sibcs` to keep only pedons whose reference matches (case-sensitive, ICU). Default `NULL` keeps every pedon.

Details

Useful for empirical validation of the SiBCS Cap 18 precedence rule on field-described datasets such as BDsolos and Redape, where clay-film qualifiers are recorded in mixed Portuguese / English tokenisation. The audit is read-only and never invokes `classify_sibcs()`.

Value

A `data.frame` with columns `id`, `reference_sibcs`, `argic_passed`, `has_films_at_argic`, `strong_films_at_argic`, and `would_exclude_from_latossolo`.

Examples

```
## Not run:
peds <- load_bdsolos_csv("RJ.csv")
a <- audit_argic_strong_films(peds, reference_filter = "LATOSSOLO")
table(a$would_exclude_from_latossolo)

## End(Not run)
```

`auto_set_proj_env` *Auto-detect PROJ_LIB and GDAL_DATA directories*

Description

Probes the common system locations for PROJ `proj.db` and GDAL data directories, on macOS Homebrew (Apple silicon and Intel), Linuxbrew, conda / mamba environments, and Debian / Ubuntu / Fedora apt or dnf installs. Sets the corresponding environment variables only when they are not already set, so a user-provided value always wins. Idempotent: safe to call repeatedly.

Usage

```
auto_set_proj_env(verbose = FALSE)
```

Arguments

`verbose`	If `TRUE`, emits a `cli` message confirming what was detected.

Details

Called automatically from `.onLoad`; call manually after installing PROJ / GDAL via Homebrew if you want to refresh the env without restarting R.

Value

Invisibly, a named list with `PROJ_LIB` and `GDAL_DATA` (the values that were set, or `NA_character_` if a value was already present or no candidate was found).

`available_esdb_attributes` *List ESDB Raster Library attributes available at a given root*

Description

Walks ‘raster_root‘ and returns the folder names that contain a valid ‘<NAME>.tif‘ raster. Useful for discovery before calling `lookup_esdb`.

Usage

```
available_esdb_attributes(raster_root)
```

Arguments

`raster_root`	Path to the unpacked ESDB raster directory (typically ‘<some>/ESDB-Raster-Library-1k-GeoTIFF-...‘).

Value

A character vector of attribute names (sorted).

Examples

```
## Not run:
available_esdb_attributes("~/data/ESDB-Raster-Library-1k-GeoTIFF-20240507")
#> [1] "AGLI1NNI" "AGLI2NNI" "AGLIM1" "AGLIM2" "ALT" "ATC" "AWC_SUB" ...
#>     [continued: 71 attributes]

## End(Not run)
```

`batch_robustness` *Batch robustness across many pedons*

Description

Runs `classification_robustness` on each pedon in a list and returns a tidy data.frame with one row per pedon. Useful for paper-grade claims like "85 to a 5

Usage

```
batch_robustness(pedons, ...)
```

Arguments

pedons	List of `PedonRecord` objects.
...	Passed to `classification_robustness`.

Value

A data.frame with columns `id`, `baseline`, `robustness`, `n_flipped`.

Examples

```
## Not run:
pedons <- list(make_ferralsol_canonical(),
                 make_luvisol_canonical(),
                 make_chernozem_canonical())
batch_robustness(pedons, system = "wrb2022", n = 50)
#>            id   baseline robustness n_flipped
#> 1 FR-canon-01 Ferralsols       0.96         2
#> 2 LV-canon-01   Luvisols       1.00         0
#> 3 CH-canon-01 Chernozems       0.94         3

## End(Not run)
```

`benchmark_afsp` *Benchmark soilKey WRB predictions against AfSP ground truth*

Description

Benchmark soilKey WRB predictions against AfSP ground truth

Usage

```
benchmark_afsp(pedons, verbose = TRUE)
```

Arguments

`pedons`	List of `PedonRecord` from `load_afsp_pedons` or `load_afsp_sample`.
`verbose`	Print progress.

Value

List with `accuracy`, `n_compared`, `confusion`, `per_class_recall`.

`benchmark_bdsolos` *Benchmark soilKey classifiers against BDsolos national reference labels*

Description

Runs `classify_wrb2022`, `classify_sibcs`, and `classify_usda` on each `PedonRecord` loaded from a BDsolos CSV via `load_bdsolos_csv`, then compares each predicted classification against the corresponding BDsolos reference label (`reference_sibcs`, `reference_wrb`, `reference_st`) and reports per-system accuracy, per-class recall, and a confusion matrix.

Usage

```
benchmark_bdsolos(
  pedons,
  systems = c("wrb2022", "sibcs", "usda"),
  sibcs_level = c("order", "subordem"),
  max_n = NULL,
  verbose = TRUE
)
```

Arguments

`pedons`	A list of `PedonRecord` objects, typically produced by `load_bdsolos_csv`.
`systems`	Character vector. Any subset of `c("wrb2022","sibcs", "usda")`. Default runs all three.
`sibcs_level`	One of `"order"` (default) or `"subordem"`. Forwarded to `normalise_febr_sibcs`.
`max_n`	Optional integer; cap classification at the first `max_n` pedons. `NULL` (default) classifies every pedon.
`verbose`	If `TRUE` (default), emits cli progress messages.

Value

A list with elements:

- `per_system` – named list (one entry per requested system) of `list(accuracy, n_compared, n_correct, n_errors, confusion, per_class)` (or `list(accuracy = NA_real_, message)` when no reference labels were present).
- `coverage` – named list of `list(n_with_ref, n_total, pct)` per system.
- `config` – named list capturing `n_pedons`, `systems`, `sibcs_level`, `soilKey_version`, `timestamp`.

Reference label coverage

BDsolos densely populates `reference_sibcs` (~82 of the v0.9.59 audit) but sparsely populates `reference_wrb` and `reference_st` (UF-dependent; ~5 states). The function always reports the per-system label coverage (`$coverage`) so the caller can judge how representative each accuracy figure is.

Comparison level

SiBCS comparison is at `level = "order"` by default, which converts the BDsolos all-caps Portuguese label (e.g. `"ARGISSOLO VERMELHO Tb EUTROFICO ..."`) to the soilKey plural Title Case form (`"Argissolos"`) via `normalise_febr_sibcs`. Set `sibcs_level = "subordem"` to compare the first two SiBCS tokens (Ordem + Subordem).

WRB and USDA comparisons are at the Reference Soil Group / Order level: `normalise_febr_wrb()` strips qualifier parens and pluralises the bare RSG (`"Xanthic Ferralsol"` -> `"Ferralsols"`); `normalise_febr_usda()` maps the suffix of the last subgroup token to the USDA Order (`"Typic Haplorthox"` -> `"Oxisols"`).

Errors and missing-label handling

Pedons without a reference label for a given system are silently excluded from THAT system's comparison (but still classified for the other two systems). If a system has zero pedons with a reference label, the corresponding `$per_system` entry has `accuracy = NA_real_` and `message = "no_reference_labels"`. Classifier errors are caught per-pedon and recorded in `n_errors`; they do not abort the run.

See Also

`load_bdsolos_csv`, `benchmark_lucas_2018`, `classify_all`, `normalise_febr_sibcs`, `normalise_febr_wrb`, `normalise_febr_usda`.

Examples

```
## Not run:
# Single UF -- typical SiBCS-dense slice
peds <- load_bdsolos_csv("RJ.csv")
bench <- benchmark_bdsolos(peds, systems = c("sibcs", "wrb2022", "usda"))
bench$coverage      # how many pedons had each reference label
bench$per_system$sibcs$accuracy
bench$per_system$sibcs$confusion

# Subordem level
bench2 <- benchmark_bdsolos(peds, systems = "sibcs",
                               sibcs_level = "subordem")

## End(Not run)
```

`benchmark_lucas_2018` *Run the LUCAS Soil 2018 / ESDB WRB benchmark*

Description

For each pedon in pedons, attaches the canonical Reference Soil Group at its coordinate via `lookup_esdb`, runs `classify_wrb2022` (or `classify_sibcs`), and tabulates predicted vs reference. Optionally fills missing texture from ISRIC SoilGrids 250m before classifying so that WRB diagnostic horizons that depend on clay (argic, ferralic, nitic) are reachable.

Usage

```
benchmark_lucas_2018(
  pedons,
  esdb_root,
  attribute = "WRBLV1",
  fill_texture_from = NULL,
  fill_topsoil_from = c("none", "soilgrids", "spectra"),
  fill_subsoil_from = c("none", "soilgrids"),
 fill_properties = c("clay", "sand", "silt", "phh2o", "soc", "cec", "bdod", "nitrogen",
    "cfvo"),
  ossl_models = NULL,
  classify_with = c("wrb2022", "sibcs"),
  max_n = NULL,
  soilgrids_lookup_fn = lookup_soilgrids,
  verbose = TRUE
)
```

Arguments

`pedons`	List of `PedonRecord` objects, typically from `load_lucas_soil_2018`.
`esdb_root`	Path to the unpacked ESDB raster directory (containing the `WRBLV1/` sub-folder).
`attribute`	ESDB attribute to use as reference. Default `"WRBLV1"` (Reference Soil Group, 31 codes). Other sensible choices: `"FAO90LV1"` (legacy FAO 1990).
`fill_texture_from`	Deprecated alias for `fill_topsoil_from` (v0.9.49 signature). When `"soilgrids"`, treated as `fill_topsoil_from = "soilgrids"` with `fill_properties = c("clay", "sand", "silt")` and `fill_subsoil_from = "none"`.
`fill_topsoil_from`	One of `"none"` (default), `"soilgrids"` (fills topsoil 0-20 cm from SoilGrids 250m at 0-5 cm), or `"spectra"` (runs `predict_from_spectra` with the supplied `ossl_models`; pedons must have `$spectra$vnir` attached, e.g. via `attach_lucas_spectra`).
`fill_subsoil_from`	One of `"none"` (default) or `"soilgrids"` (synthesises a 30-60 cm B horizon from SoilGrids 250m). Unlocks WRB diagnostic horizons that depend on subsoil features (cambic, argic, mollic).
`fill_properties`	Character vector of SoilGrids properties to fill when `fill_topsoil_from = "soilgrids"` or `fill_subsoil_from = "soilgrids"`. Default uses all 9 properties: clay, sand, silt, phh2o, soc, cec, bdod, nitrogen, cfvo. Set to `c("clay", "sand", "silt")` to recover the v0.9.49 behaviour. `cfvo` is mapped to `coarse_fragments_pct`, which drives the Leptosols diagnostic (>= 90 within 25 cm).
`ossl_models`	Required when `fill_topsoil_from = "spectra"`. A list of `soilKey_pls_model` objects from `train_pls_from_ossl` (v0.9.46).
`classify_with`	One of `"wrb2022"` (default) or `"sibcs"`.
`max_n`	Optional integer cap on the number of pedons benchmarked. Useful for quick development runs.
`soilgrids_lookup_fn`	Internal: SoilGrids lookup function (defaults to `lookup_soilgrids`). Override for unit tests to inject a deterministic stub.
`verbose`	If `TRUE` (default), prints progress.

Details

This closes Route B of the v0.9.27 EU-LUCAS roadmap end-to-end: v0.9.44 `lookup_esdb` provides the reference label; v0.9.49 (this) provides the loader and the comparison loop; v0.9.48 `lookup_soilgrids` fills texture; v0.9.46 `predict_from_spectra` and v0.9.47 `predict_munsell_from_spectra` can fill the chemistry / Munsell gaps when Vis-NIR is available.

Value

A list with elements:

`predictions` data.frame with one row per pedon: `point_id`, `lon`, `lat`, `country`, `predicted`, `reference_code`, `reference_name`, `agree`.

`confusion`	Confusion table (predicted vs reference) over in-scope rows.
`accuracy`	Overall fraction of correct classifications among in-scope rows.
`per_rsg`	Per-RSG recall data.frame.
`n_in_scope`	Number of pedons with both predicted and reference set.
`n_total`	Total pedons benchmarked.
`n_errors`	Number of pedons where the classifier errored out.
`errors`	List of `(i, id, error)` tuples for classifier errors.
`config`	Recap of arguments used.

See Also

`load_lucas_soil_2018`, `lookup_esdb`, `lookup_soilgrids`.

Examples

```
## Not run:
pedons <- load_lucas_soil_2018(
  "soil_data/eu_lucas/LUCAS-SOIL-2018-data-report-readme-v2/LUCAS-SOIL-2018-v2",
  countries = c("ES"), max_n = 50)
bench <- benchmark_lucas_2018(
  pedons,
  esdb_root = "soil_data/eu_lucas/ESDB-Raster-Library-1k-GeoTIFF-20240507",
  fill_texture_from = "soilgrids")
bench$accuracy
bench$per_rsg

## End(Not run)
```

`benchmark_performance` *Run the soilKey performance benchmark*

Description

Generates `n` synthetic pedons (5 horizons each, with the chemistry / morphology populated for typical Argissolo / Latossolo / Cambissolo cases), calls each classifier on each pedon, and reports per-call latency + total throughput.

Usage

```
benchmark_performance(
  n = 100L,
  systems = c("wrb2022", "sibcs", "usda"),
  include_familia = FALSE,
  seed = 42L,
  verbose = TRUE
)
```

Arguments

`n`	Integer. Number of synthetic pedons to generate. Default 100; pass 1000 or higher for batch-level measurements.
`systems`	Character vector. Which classifiers to time. Default `c("wrb2022", "sibcs", "usda")` (all three).
`include_familia`	Pass-through to `classify_sibcs` when `"sibcs"` is in `systems`. Default `FALSE`.
`seed`	RNG seed for reproducibility. Default 42.
`verbose`	If `TRUE` (default), prints a per-system summary line.

Details

Designed to be a one-shot reproducible benchmark: the synthetic pedons use a fixed RNG seed so timings on the same machine are comparable across releases.

Value

A list with elements:

`summary`	data.frame: `system`, `n_pedons`, `total_seconds`, `mean_seconds`, `median_seconds`, `pedons_per_minute`.
`per_pedon`	data.frame with one row per (pedon, system) call: `i`, `system`, `seconds`, `status`.
`config`	list with `n`, `seed`, `soilKey_version`, `R_version`, `platform`.

Examples

```
## Not run:
bench <- benchmark_performance(n = 100)
bench$summary
#>     system n_pedons total_seconds mean_seconds median_seconds pedons_per_minute
#> 1 wrb2022      100        ~ 5-12      0.05-0.12      ~          ~
#> 2   sibcs      100        ~ 5-15      0.05-0.15      ~          ~
#> 3    usda      100        ~ 4-10      0.04-0.10      ~          ~

## End(Not run)
```

`benchmark_redape` *Benchmark soilKey SiBCS predictions against the Redape gold standard*

Description

Runs `classify_sibcs` on each pedon and compares against the curator-validated reference label (Order / Suborder / Great Group / Subgroup). Returns per-level accuracy and the confusion matrix at the requested granularity.

Usage

```
benchmark_redape(
  pedons,
  level = c("order", "subordem", "gde_grupo", "subgrupo"),
  verbose = TRUE
)
```

Arguments

`pedons`	List of `PedonRecord` objects (typically from `load_redape_pedons`).
`level`	One of `"order"` (default), `"subordem"`, `"gde_grupo"`, or `"subgrupo"`.
`verbose`	Print progress (default `TRUE`).

Value

A list with `accuracy`, `n_compared`, `confusion`, `per_class_recall`, and the per-pedon `predictions` table. `predictions` now also includes columns `ref_norm` and `pred_norm` – the canonical comparison keys – for downstream auditing.

v0.9.81 level-aware comparison

Earlier versions accepted the `level` argument but always used `rsg_or_order` for the prediction and the order field for the reference, so all four levels reported identical accuracy. v0.9.81 reads the level-specific slots from `res$trace` (subordem, grande_grupo, subgrupo) and concatenates the matching reference fields, applying SiBCS-aware Portuguese pluralisation so the comparison key matches the predictor's plural Title Case form.

`benchmark_run_classification`

Run a benchmark across one of the loaded pedon lists

Description

Classifies each pedon in `pedons` against the named system, compares against the published reference (e.g. `site$reference_wrb`), and returns a confusion matrix + top-1 / top-3 accuracy + bootstrap CI on top-1.

Usage

```
benchmark_run_classification(
  pedons,
  system = c("wrb2022", "sibcs", "usda"),
  level = c("order", "subgroup", "subordem", "great_group", "suborder"),
  boot_n = 1000L
)
```

Arguments

`pedons`	List of `PedonRecord` objects (output of one of the `load_*` functions).
`system`	One of `"wrb2022"`, `"sibcs"`, `"usda"`.
`level`	Granularity of the comparison: • `"order"` (default) – the top-level RSG / Ordem / Order, compared against `cls$rsg_or_order`; • `"subgroup"` – the full classified name (Subgroup in USDA, Subgrupo in SiBCS, RSG + qualifiers in WRB), compared against `cls$name` after case-insensitive token normalisation; • `"subordem"` – SiBCS-only, the 2nd-level "Ordem + Subordem" (e.g. "Latossolos Vermelhos"). Comparison via the first two normalised tokens of the predicted name vs the reference; • `"great_group"` (USDA, v0.9.24) – the LAST token of the subgroup name (e.g. `"typic hapludalfs"` -> `"hapludalfs"`). Isolates whether the Great Group machinery is correct independent of subgroup modifiers (Typic / Aquic / Vertic / Cumulic / Pachic / etc.). Reads `site$reference_usda_grtgroup`; • `"suborder"` (USDA, v0.9.24) – maps the Great Group prediction to its canonical Suborder suffix (`"hapludalfs"` -> `"udalfs"`) using the KST 13ed Ch 4 ~70-Suborder list. Reads `site$reference_usda_suborder`.
`boot_n`	Bootstrap replicates for CI (default 1000).

Value

A list with elements `accuracy_top1`, `accuracy_ci`, `confusion`, and `per_pedon` (one row per pedon with predicted vs reference).

`benchmark_spectral_fill`

Benchmark the accuracy lift of spectral gap-fill (ON vs OFF), k-fold

Description

The honest measurement that has been data-blocked until a spectra-bearing, labelled dataset exists. For each cross-validation fold it calibrates a spectral library on the training profiles, then classifies the held-out profiles twice – **OFF** (spectra-only pedon, no lab attributes) and **ON** (`fill_from_spectra` predicts the lab attributes from the scan first) – and scores both against the reference label. Non-circular: the calibration library never includes a test profile.

Usage

```
benchmark_spectral_fill(
  reflectance,
  metadata,
  id_col = "id",
  system = c("sibcs", "wrb2022", "usda"),
```

```
  profile_col = NULL,
  folds = 5L,
  properties = NULL,
  method = c("mbl", "plsr_local", "pretrained"),
  wavelengths = NULL,
  resample_to = NULL,
  property_map = NULL,
  label_map = NULL,
  normalize = c("auto", "none", "percent"),
  fold_id = NULL,
  verbose = TRUE
)
```

Arguments

`reflectance`	Reflectance data: a matrix / data.frame with rows = samples and columns named by wavelength (nm); OR a long data.frame with `id_col`, `wavelength_nm`, `reflectance`; OR a path to a CSV in either form.
`metadata`	A data.frame with one row per sample carrying `id_col` plus lab attributes and optional taxonomic labels and `lat`/ `lon`. Rows are aligned to `reflectance` by `id_col`.
`id_col`	Sample identifier column shared by both tables (default `"id"`).
`system`	One of `"sibcs"` (default), `"wrb2022"`, `"usda"`.
`profile_col`	Column grouping samples into profiles (default `id_col`).
`folds`	Number of CV folds (default 5).
`properties`	Attributes to predict from spectra (default the `fill_from_spectra` set).
`method`	Spectral model: `"mbl"`, `"plsr_local"` or `"pretrained"` (passed to `fill_from_spectra`).
`wavelengths`	Optional explicit wavelength vector (nm) when the reflectance columns are not wavelength-named.
`resample_to`	Optional target wavelength grid (nm) to linearly resample every spectrum onto (e.g. `350:2500`); default keeps the native grid.
`property_map`, `label_map`	Optional named lists overriding the alias auto-detection, e.g. `property_map = list(clay_pct = "ARGILA")`.
`normalize`	One of `"auto"` (divide by 100 when values look like percent), `"percent"`, or `"none"`.
`fold_id`	Optional integer vector (one per profile, in sorted-id order) to use fixed folds instead of the deterministic modulo split.
`verbose`	Print a one-line summary (default `TRUE`).

Value

A list with `accuracy_off`, `accuracy_on`, `delta`, `n`, per-fold rows, and the per-profile `predictions` frame.

See Also

`read_spectral_library`, `fill_from_spectra`

`benchmark_unified` *Unified cross-dataset benchmark across SiBCS / WRB / USDA*

Description

Runs a system's soilKey classifier on every dataset that has reference labels for that system, then pools the results into a single nation-/world-wide accuracy estimate.

Usage

```
benchmark_unified(
  systems = c("all", "wrb2022", "sibcs", "usda"),
  datasets = c("all", "bdsolos", "febr", "kssl", "lucas_esdb"),
  paths = NULL,
  max_n_per_dataset = NULL,
  engine = c("soilkey", "aqp", "both"),
  harmonize = FALSE,
  gapfill = FALSE,
  verbose = TRUE
)
```

Arguments

`systems`	Character vector. Any subset of `c("wrb2022","sibcs", "usda")`. Default `"all"` runs all three.
`datasets`	Character vector. Any subset of `c("bdsolos", "febr", "kssl", "lucas_esdb")`. Default `"all"` pools every dataset that has reference labels for the requested systems. Datasets without reference labels for a system are silently excluded from that system's pooled result.
`paths`	Named list of dataset paths. Element names should match those in `datasets`. If `NULL` (default), soilKey looks for canonical paths under `"~/soil_data/"`.
`max_n_per_dataset`	Optional integer to cap per-dataset sample size (useful for development / debugging). `NULL` (default) classifies every available pedon.
`engine`	Currently forwarded to Phase-1 aqp wiring. One of `"soilkey"` (default), `"aqp"`, `"both"`. When `"aqp"`, sets `options(soilKey.diagnostic_engine = "aqp")` for the duration of the benchmark, which routes `argic()` / `cambic()` through the canonical `aqp::getArgillicBounds` / `getCambicBounds`.
`harmonize`	If `TRUE` (default `FALSE`), applies `harmonize_to_gsm` to each dataset's pedons before classification, putting all chemistry/texture on the GSM depth grid (0-5 / 5-15 / 15-30 / 30-60 / 60-100 / 100-200 cm). Required for cross-dataset pooling integrity (Phase 2.3) but slow (~1-2 min for 1k pedons) and may degrade per-dataset accuracy slightly because the splined depths are approximations.

`gapfill`	If not `FALSE` (the default), applies `gapfill_within_pedon` to each dataset's pedons before classification, filling interior `NA` cells of the continuous depth-trending attributes by within-pedon linear interpolation. Accepts the same values as the `gapfill` argument of `classify_all` (`TRUE`, a character vector of attributes, or a named list). Lets you measure the ON/OFF accuracy lift of gap-fill reproducibly through the harness.
`verbose`	If `TRUE` (default), emits cli progress.

Value

A list with elements:

- `per_system` – per-system pooled `list(accuracy, n_compared, n_correct, confusion, per_class)`.
- `per_system_per_dataset` – per-(system, dataset) same shape, for breakdown.
- `coverage` – per-(system, dataset) sample sizes and label coverage.
- `config` – captures `systems`, `datasets`, `engine`, `soilKey_version`, `timestamp`.

Datasets and their reference labels

Dataset	Systems with reference labels
BDsolos	SiBCS (dense), WRB (sparse), USDA (sparse)
FEBR superconjunto	SiBCS, WRB, USDA (most rows have all 3)
KSSL+NASIS	USDA only (samp_taxsubgrp universal)
LUCAS + ESDB raster	WRB (via lookup_esdb on coords)

For each (system, dataset) pair, this function:

1. Loads pedons via the appropriate `load_*` helper.
2. Filters to pedons with a populated reference label for the requested system.
3. Normalises both reference and predicted labels via `normalise_febr_*()` / KSSL canonicalisation helpers.
4. Calls the system's classifier and records pred-vs-ref.

Then pools per-system results across datasets.

Engine selection (Phase 1 wiring)

For datasets with morphological data (BDsolos / FEBR), the diagnostics that pivot Argissolos / Latossolos / Cambissolos classification can be run with two engines:

- `engine = "soilkey"` (default) – the hand-coded WRB 6/1.4/20 thresholds.
- `engine = "aqp"` – aqp::getArgillicBounds / getCambicBounds (KST 13ed 3/1.2/8 thresholds).

On the v0.9.62 RJ benchmark (722 perfis), aqp was 14.8 pp stricter on argic and 40.6 pp more permissive on cambic; the SiBCS Argissolos / Latossolos / Cambissolos boundary is sensitive to both. `engine` is currently forwarded to a future v0.9.63 wired `argic()` / `cambic()`; for now, `benchmark_unified()` reports separately per engine when `engine = "both"`.

See Also

`benchmark_bdsolos`, `benchmark_lucas_2018`, `benchmark_run_classification`, `harmonize_to_gsm`.

`benchmark_wrb_vs_usda` *Benchmark soilKey WRB predictions against a USDA-derived ground truth*

Description

Convenience wrapper: applies `annotate_wrb_from_usda` to attach derived WRB labels, runs `classify_wrb2022` on each pedon, and returns top-1 accuracy + per-RSG recall.

Usage

```
benchmark_wrb_vs_usda(pedons, verbose = TRUE)
```

Arguments

`pedons`	List of `PedonRecord` objects with `site$reference_usda` populated (typically from `load_kssl_pedons_gpkg`).
`verbose`	Print progress.

Value

A list with `accuracy`, `n_compared`, `confusion`, `per_class_recall`.

`build_taxon_profiles` *Build per-taxon mean depth profiles for predicted-taxon gap-fill*

Description

For each taxon (the first word of the reference label at the requested level), averages each attribute across the calibration pedons into the six standard depth slices (0-5 ... 100-200 cm). The result feeds `gapfill_by_predicted_taxon`. Calibrate on a set DISJOINT from the pedons you will fill (e.g. a train split) to keep the fill non-circular.

Usage

```
build_taxon_profiles(pedons, ref_field = "reference_sibcs", attrs = NULL)
```

Arguments

`pedons`	A list of `PedonRecord` with a reference label.
`ref_field`	Site field holding the reference label (default `"reference_sibcs"`; e.g. `"reference_usda"` / `"reference_wrb"`).
`attrs`	Attributes to profile (default the continuous gap-fill set).

Value

A named list `taxon` -> `attr` -> `numeric(6)` (NA where a taxon has no measured value in a slice).

See Also

`gapfill_by_predicted_taxon`

`B_espodico` *Horizonte B espodico (SiBCS Cap 2, p 62-65; v0.7)*

Description

Subsuperficial com acumulo iluvial de Al + Fe + materia organica; espessura \>= 2.5 cm. Tipos: Bs, Bhs, Bh, ortstein. Reuso de `spodic` (WRB) que ja codifica criterios essencialmente identicos.

Usage

```
B_espodico(pedon, ...)
```

Arguments

`pedon`	A `PedonRecord`.
`...`	Reserved for future arguments.

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`B_incipiente` *Horizonte B incipiente (SiBCS Cap 2, p 59-61; v0.7)*

Description

Subsuperficial sob A/Ap/AB com alteracao fisica e quimica incipiente, NAO satisfazendo a B textural / latossolico / nitico / espodico / planico, com:

- espessura \>= 10 cm;
- textura francoarenosa ou mais fina;
- < 50% estrutura da rocha original;
- evidencias de pedogenese (cor mais viva OR remocao de carbonatos OR designation `Bw`/`Bi`);
- NAO satisfaz: argic, ferralic, espodic, planic, e nao tem duripa/petrocalcico/fragipa.

Usage

```
B_incipiente(pedon, min_thickness = 10)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`B_latossolico` *Horizonte B latossolico (SiBCS Cap 2, p 57-59; v0.7 strict)*

Description

Adicionalmente a `ferralic` (WRB), o B latossolico SiBCS exige:

- Espessura minima de 50 cm;
- Textura francoarenosa ou mais fina;
- Estrutura granular muito pequena/pequena ou em blocos subangulares fraco/moderado;
- < 5% volume mostrando estrutura da rocha original;
- Ki \<= 2.2 (geralmente \<= 2.0);
- Cerosidade no maximo pouca e fraca.

v0.7 enforce thickness, texture, e ausencia de estrutura primaria herdada via designation e clay; Ki/Kr quantitativos sao v0.8 (precisa de SiO2/Al2O3 lab-data nao no schema).

Usage

```
B_latossolico(
  pedon,
  min_thickness = 50,
  max_cec_per_clay = NULL,
  engine = NULL,
  ...
)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Numeric threshold or option (see Details).
`max_cec_per_clay`	Numeric threshold or option (see Details). Defaults to `NULL` (engine-aware): 17 in soilkey engine (the SiBCS-loose threshold, slightly more permissive than strict WRB ferralic 16) or 20 in aqp engine (v0.9.68 regional tolerance for Embrapa lab methodology offset).
`engine`	One of `"soilkey"` (default) or `"aqp"`; `NULL` reads `getOption("soilKey.diagnostic_engine")`. Forwarded to `ferralic`.
`...`	Reserved for future arguments.

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`B_nitico` *Horizonte B nitico (SiBCS Cap 2, p 61-62; v0.7)*

Description

Subsuperficial nao hidromorfico, textura argilosa/muito argilosa (clay \>= 35% desde a superficie), com pequeno incremento de argila (B/A \<= 1.5), estrutura em blocos sub/angulares ou prismatica grau moderado/forte, cerosidade no minimo comum + moderada, espessura \>= 30 cm. Argila ativ baixa OR ativ alta + carater aluminico.

Usage

```
B_nitico(
  pedon,
  min_thickness = 30,
  min_clay_pct = 35,
  max_b_a_ratio = 1.5,
  min_cerosidade = c("common", "many", "abundant", "strong")
)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Numeric threshold or option (see Details).
`min_clay_pct`	Numeric threshold or option (see Details).
`max_b_a_ratio`	Numeric threshold or option (see Details).
`min_cerosidade`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`B_planico` *Horizonte B planico (SiBCS Cap 2, p 65-66; v0.7)*

Description

Tipo especial de B textural com mudanca textural abrupta + permeabilidade lenta + cores neutras/escurecidas + cromas baixos.

Usage

```
B_planico(pedon)
```

Arguments

`pedon`	A `PedonRecord`.

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`B_textural` *Horizonte B textural (SiBCS Cap 2, p 54-57; v0.7 strict)*

Description

Horizonte mineral subsuperficial com incremento de argila + cerosidade OR aumento gradativo, satisfazendo criterios de espessura e relacao textural B/A. v0.7 enforce as alternativas (a)-(j) do SiBCS por delegacao parcial ao WRB `argic` (criterios de clay-increase essencialmente identicos) acrescidos de:

- espessura \>= 7.5 cm OR \>= 10% da soma das espessuras dos sobrejacentes; e
- textura \>= francoarenosa.

Refinamentos pendentes para v0.8: cerosidade obrigatoria sob certas estruturas (criterio i.1 / i.2 / i.3); lamelas \>= 15 cm combinadas.

Usage

```
B_textural(pedon, ...)
```

Arguments

`pedon`	A `PedonRecord`.
`...`	Reserved for future arguments.

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`calcaric_material` *Calcaric material (WRB 2022 Ch 3.3.3): \>= 2% CaCO3 throughout the fine earth, primary carbonates from the parent material.*

Description

Calcaric material (WRB 2022 Ch 3.3.3): \>= 2% CaCO3 throughout the fine earth, primary carbonates from the parent material.

Usage

```
calcaric_material(pedon, min_caco3_pct = 2)
```

Arguments

`pedon`	A `PedonRecord`.
`min_caco3_pct`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`calcic` *Calcic horizon (WRB 2022)*

Description

Tests whether any horizon meets the calcic horizon criteria. The calcic horizon is a horizon of secondary carbonate accumulation, diagnostic for Calcisols and qualifying many other RSGs.

Usage

```
calcic(pedon, min_thickness = 15, min_caco3_pct = 15)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Minimum thickness in cm (default 15).
`min_caco3_pct`	Minimum CaCO3 percent in fine earth (default 15).

Details

Sub-tests called:

- `test_caco3_concentration` – CaCO3 >= 15%.
- `test_minimum_thickness` – thickness >= 15 cm.

v0.2 limitations: the WRB criterion of "5% absolute or relative more CaCO3 than the underlying horizon" is not enforced; this captures true calcic horizons but may also mark uniformly carbonate-rich substrates that are not pedologically calcic. Cementation (petrocalcic) is not yet detected. Both refinements are scheduled for v0.3.

Value

A `DiagnosticResult`.

References

IUSS Working Group WRB (2022). *World Reference Base for Soil Resources*, 4th edition. International Union of Soil Sciences, Vienna. Chapter 3 – Calcic horizon.

`cambic` *Cambic horizon (WRB 2022)*

Description

Tests whether any horizon meets the cambic horizon criteria. The cambic horizon is a subsurface horizon with evidence of pedological alteration that does not meet the criteria for any stronger diagnostic horizon. It is the diagnostic of Cambisols.

Usage

```
cambic(pedon, min_thickness = 15, min_top_cm = 5, engine = NULL)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Minimum thickness in cm (default 15).
`min_top_cm`	Minimum top depth (cm) for a horizon to be considered cambic-eligible (default 5). Anchors the candidate set to subsurface layers.

`engine`	v0.9.63+. One of `"soilkey"` (hand-coded path, default for back-compat) or `"aqp"` (canonical NRCS dispatch via `aqp::getCambicBounds`). When `NULL` (the new default) the function reads `getOption("soilKey.diagnostic_engine", "soilkey")`, so a global `options(soilKey.diagnostic_engine = "aqp")` flips every `cambic()` call without modifying call sites. The aqp engine fired 40.6 soilkey 0 v0.9.50 LUCAS WRB benchmark from 0 100 `cambic_aqp`.

Details

v0.2 implementation tests three conditions:

- thickness >= 15 cm (`test_minimum_thickness`)
- texture sandy loam or finer (`test_texture_argic`)
- NOT `argic` AND NOT `ferralic`

v0.2 limitations: WRB 2022 also excludes profiles with spodic, calcic, gypsic, plinthic, vertic, and several other diagnostic horizons. Those exclusions, plus the WRB criteria of "evidence of alteration" (color/structure differences from parent material, carbonate removal), are scheduled for v0.3.

Value

A `DiagnosticResult`.

References

IUSS Working Group WRB (2022), Chapter 3, Cambic horizon.

`cambic_aqp` *Cambic horizon via aqp::getCambicBounds()*

Description

Wraps `aqp::getCambicBounds()` in soilKey's `DiagnosticResult` contract. The aqp test enforces the KST 13ed cambic criteria:

- Texture finer than loamy fine sand (i.e. NOT in the sandy-texture pattern).
- Soil structure or absence of rock structure.
- Evidence of pedogenic alteration (chroma / value / clay).
- NOT meeting argic / oxic / spodic / mollic criteria.

soilKey's `cambic` (and the SiBCS proxy `B_incipiente`) implements similar logic but with SiBCS / WRB-flavoured exclusions; the aqp engine here is an independent canonical reference.

Usage

```
cambic_aqp(pedon, argi_bounds = NULL, ...)
```

Arguments

`pedon`	A `PedonRecord`.
`argi_bounds`	Optional `c(ubound, lbound)` for argillic bounds (forwarded to aqp). `NULL` (default) means the aqp internals re-detect.
`...`	Reserved for future arguments.

Value

A `DiagnosticResult` with `name = "cambic_aqp"`.

See Also

`cambic` (soilKey hand-coded), `aqp::getCambicBounds`.

`canonicalise_kst13ed_gg` *Canonicalise a USDA Great Group label to a KST 13ed-compatible key*

Description

Maps both obsolete (pre-KST 13ed) and modern Great Group names to a single canonical key, so that direct equality between predicted and reference Great Group names ignores edition-driven renaming. Names that have no known mapping pass through unchanged.

Usage

```
canonicalise_kst13ed_gg(gg)
```

Arguments

`gg`	Character vector of Great Group names (lower case, no whitespace).

Details

Examples of the canonicalisation (each pair is rendered equivalent):

- `"haplaquolls"` (KST 8) === `"endoaquolls"` (KST 13ed)
- `"pellusterts"` (KST 8) === `"hapluderts"` (KST 13ed)
- `"camborthids"` (KST 8) === `"haplocambids"` (KST 13ed)
- `"vitrandepts"` (KST 8) === `"vitrudands"` (KST 13ed)

Value

Character vector of canonical keys. Unmapped names pass through. NA stays NA. Empty input returns empty vector.

References

Soil Survey Staff (2022), Keys to Soil Taxonomy 13ed, Ch 4 (Order keys); previous editions for the obsolete names.

`canonical_reference` *Load a canonical reference dataset from soilKey or SoilTaxonomy*

Description

Resolution order:

1. If the `SoilTaxonomy` package is installed AND the `prefer_pkg` argument is `TRUE` (default), load the dataset from the installed package (always fresh).
2. Otherwise, load from the vendored copy at `inst/extdata/canonical/<name>.rda`.

Usage

```
canonical_reference(
  name = c("WRB_4th_2022", "ST_criteria_13th", "ST_features"),
  prefer_pkg = TRUE
)
```

Arguments

`name`	One of `"WRB_4th_2022"`, `"ST_criteria_13th"`, `"ST_features"`.
`prefer_pkg`	If `TRUE` (default), prefer the installed SoilTaxonomy package over the vendored copy. Set to `FALSE` to force the vendored copy (e.g. for reproducibility of a specific soilKey release).

Value

The dataset as the original R object (list or data.frame).

See Also

`wrb2022_canonical`, `kst13_canonical`, `st_features_canonical`.

`cerosidade` *Cerosidade quantitativa (SiBCS Cap 13, p 207; Cap 1)*

Description

Diagnostico parametrizado quantidade x intensidade de cerosidade (clay films / cutans). Consume as colunas v0.7.2 `clay_films_amount` (ordinal: few/pouca, common/comum, many/abundante, continuous/continua) e `clay_films_strength` (ordinal: weak/fraca, moderate/moderada, strong/forte; "shiny" mapeado a "strong"), introduzidas em substituicao ao legado `clay_films`.

Usage

```
cerosidade(pedon, min_amount = "common", min_strength = "moderate")
```

Arguments

`pedon`	A `PedonRecord`.
`min_amount`	Quantidade minima: `"few"`, `"common"`, `"many"`, `"continuous"` (ou equivalentes em PT-BR). Default `"common"`.
`min_strength`	Intensidade minima: `"weak"`, `"moderate"`, `"strong"`. Default `"moderate"`. Pass `NULL` para ignorar a dimensao de intensidade.

Details

Discriminante critico Nitossolos vs Argissolos no Cap 13: Nitossolos exigem cerosidade $\geq$ comum + $\geq$ moderada (defaults).

Value

`DiagnosticResult`; `passed = TRUE` se ao menos um horizonte B atende ambos os limiares.

References

Embrapa (2018), SiBCS 5a ed., Cap 13 (Nitossolos), p 207; Cap 1 (atributos diagnosticos).

`chernic` *Chernic horizon (WRB 2022): the cherozemic-style mollic with very high biological activity (worm holes, casts, coprolites). v0.3.3: delegates to mollic + worm_holes_pct >= 50 (proxy for "biological homogenization").*

Description

Chernic horizon (WRB 2022): the cherozemic-style mollic with very high biological activity (worm holes, casts, coprolites). v0.3.3: delegates to mollic + worm_holes_pct >= 50 (proxy for "biological homogenization").

Usage

```
chernic(pedon, min_worm_pct = 50)
```

Arguments

`pedon`	A `PedonRecord`.
`min_worm_pct`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`chernozem` *Chernozem RSG diagnostic (WRB 2022)*

Description

Tests whether a profile satisfies the Chernozem RSG criteria: a mollic horizon plus secondary carbonates somewhere in the profile, plus chroma (moist) <= 2 in at least one layer of the upper 20 cm.

Usage

```
chernozem(pedon, max_chroma_upper = 2)
```

Arguments

`pedon`	A `PedonRecord`.
`max_chroma_upper`	Maximum moist chroma in the upper part (default 2, per WRB 2022).

Value

A `DiagnosticResult`.

References

IUSS Working Group WRB (2022), Chapter 5, Chernozems.

`chernozem_strict` *Chernozem RSG gate (strengthened, WRB 2022 Ch 4, p 111)*

Description

WRB-canonical: chernic horizon AND, starting \<= 50 cm below the lower limit of the mollic horizon and (if a petrocalcic horizon is present) above it, a layer with protocalcic properties \>= 5 cm thick OR a calcic horizon AND base saturation \>= 50% from the surface to the protocalcic / calcic layer throughout.

Usage

```
chernozem_strict(pedon, min_bs = 50, max_top_cm = 50, strict = NULL)
```

Arguments

`pedon`	A `PedonRecord`.
`min_bs`	Numeric threshold or option (see Details).
`max_top_cm`	Numeric threshold or option (see Details).
`strict`	Logical or `NULL`. When `NULL` (default) it resolves via `getOption("soilKey.rsg_strict", FALSE)`. `TRUE` raises the base-saturation floor to 80%.

Details

v0.3.4 strengthens the previous v0.2 chernozem (which only required mollic + chernic_color) by adding the protocalcic / calcic gate and the BS \>= 50% requirement.

Note: the v0.2 `chernozem()` diagnostic remains available as a less-strict variant; `chernozem_strict()` is what the v0.3.4 key.yaml uses for the CH RSG.

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

Tier-3 strict mode (v0.9.98)

With `strict = TRUE` the base-saturation floor above the carbonate-bearing layer is raised from 50% to 80%, in line with the very high base status expected of a textbook Chernozem.

`claric_material` *Claric material (WRB 2022 Ch 3.3.4): light-coloured fine earth with Munsell criteria.*

Description

Claric material (WRB 2022 Ch 3.3.4): light-coloured fine earth with Munsell criteria.

Usage

```
claric_material(pedon)
```

Arguments

`pedon`	A `PedonRecord`.

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`ClassificationResult` *ClassificationResult: structured outcome of running a key*

Description

ClassificationResult: structured outcome of running a key

ClassificationResult: structured outcome of running a key

Details

Returned by `classify_wrb2022` (and the future `classify_sibcs`). Carries the full decision trace — which RSGs were tested, which passed, which failed, which were indeterminate because of missing data — plus the assigned class, qualifiers, ambiguities (RSGs that nearly satisfied), missing data that would refine the result, the provenance-aware evidence grade, and any biogeographical or prior-based warnings.

Public fields

`system` Character. `"WRB 2022"` or `"SiBCS 5"`.

`name` Character. Full classification name with qualifiers (e.g. `"Rhodic Ferralsol (Clayic, Humic, Dystric)"`).

`rsg_or_order` Character. Bare RSG (WRB) or order (SiBCS), e.g. `"Ferralsols"`.

`qualifiers` List. Principal and supplementary qualifiers in canonical order.

`trace` List. One element per RSG tested (in key order), each with `code`, `name`, `passed`, `evidence`, `missing`.

`ambiguities` List. RSGs that came close to passing — useful hints for follow-up measurements.

`missing_data` Character vector. Attributes whose measurement would refine or resolve the result.

`evidence_grade` Character. `"A"` (measured), `"B"` (spectra-predicted), `"C"` (prior-inferred), `"D"` (VLM-extracted), `"E"` (user-assumed), or `NA_character_`.

`prior_check` List or NULL. Result of the spatial-prior sanity check (consistent / inconsistent / not run).

`warnings` Character vector. Free-form warnings.

Methods

Public methods:

- `ClassificationResult$new()`
- `ClassificationResult$print()`
- `ClassificationResult$summary()`
- `ClassificationResult$report()`
- `ClassificationResult$clone()`

Method `new()`**:** Build a ClassificationResult.

Usage:

```
ClassificationResult$new(
  system,
  name,
  rsg_or_order = NA_character_,
  qualifiers = list(),
  trace = list(),
  ambiguities = list(),
  missing_data = character(0),
  evidence_grade = NA_character_,
  prior_check = NULL,
  warnings = character(0)
)
```

Arguments:

`system` System name.

`name` Classification name.

`rsg_or_order` RSG (WRB) or order (SiBCS).

`qualifiers` List of qualifier names.

`trace` List of per-RSG test entries.

`ambiguities` List of close-call RSGs.

`missing_data` Character vector.

`evidence_grade` Single character A/B/C/D or NA.

`prior_check` List or NULL.

`warnings` Character vector.

Method `print()`**:** Pretty-print the result with key trace, ambiguities, and warnings.

Usage:

```
ClassificationResult$print(...)
```

Arguments:

`...` Ignored (S3 print signature compatibility).

Method `summary()`**:** Compact summary list.

Usage:

```
ClassificationResult$summary(...)
```

Arguments:

`...` Ignored (S3 summary signature compatibility).

Method `report()`**:** Render this classification as a self-contained report (delegates to the package-level `report` generic). HTML output is dependency-free; PDF requires `rmarkdown` and a working LaTeX engine.

Usage:

```
ClassificationResult$report(
  file,
  format = c("auto", "html", "pdf"),
  pedon = NULL,
  ...
)
```

Arguments:

`file` Output path. Format is inferred from the extension.

`format` One of "html" or "pdf" (defaults to "auto", which infers from the extension).

`pedon` Optional `PedonRecord` whose horizons / provenance are added to the report.

`...` Forwarded to `report_html` or `report_pdf`.

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
ClassificationResult$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

`classification_robustness`

Robustness of classification under input perturbation

Description

For a given `PedonRecord`, perturb a chosen list of horizon attributes by a configured fractional amount, re-classify under the requested system, and report how often the classification `$rsg_or_order` (or full `$name`) matches the unperturbed baseline.

Usage

```
classification_robustness(
  pedon,
  system = c("wrb2022", "sibcs", "usda"),
  level = c("order", "name"),
  n = 50L,
  perturbations = NULL,
  provenance_aware = FALSE,
  seed = 42L
)
```

Arguments

`pedon`	A `PedonRecord`.
`system`	One of `"wrb2022"`, `"sibcs"`, `"usda"`.
`level`	Either `"order"` (compare `$rsg_or_order`) or `"name"` (compare full classification name).
`n`	Number of Monte-Carlo perturbed runs (default 50).
`perturbations`	Named list. Each name is a horizon column; each element is a function taking the original value and returning a perturbed value. NA-tolerant. Ignored when `provenance_aware = TRUE`.
`provenance_aware`	If `FALSE` (default) every cell is perturbed by the fixed `perturbations` panel – the exact v0.9.42 behaviour. If `TRUE`, each `(horizon,attribute)` cell is perturbed by an amount scaled to its provenance evidence grade, and `perturbations` is ignored. See `classify_with_uncertainty` for the full provenance-weighted posterior.
`seed`	Random seed for reproducibility.

Details

Default perturbation panel:

- `clay_pct`: ±5
- `sand_pct`: ±5
- `silt_pct`: ±5
- `ph_h2o`: ±0.2 absolute
- `oc_pct`: ±10

Value

A list with elements `baseline` (the unperturbed classification name), `n` (number of MC runs), `robustness` (fraction of perturbed runs matching baseline), `flipped_to` (table of alternative classifications when the perturbation flipped the result).

Examples

```
## Not run:
p <- make_ferralsol_canonical()
classification_robustness(p, system = "wrb2022", n = 50)
#> $baseline    : "Ferralsols"
#> $robustness  : 0.96  (48 / 50 perturbed runs landed on Ferralsols)
#> $flipped_to  : table(c("Cambisols" = 1, "Acrisols" = 1))

## End(Not run)
```

`classify_all` *Classify a pedon across all three taxonomic systems*

Description

Convenience wrapper that runs `classify_wrb2022`, `classify_sibcs`, and `classify_usda` on the same `PedonRecord` and returns a single named list with one entry per system (plus a `summary` table that's handy for reports).

Usage

```
classify_all(
  pedon,
  systems = "all",
  on_missing = c("warn", "silent", "error"),
  include_familia = TRUE,
  include_family = FALSE,
  specifiers = FALSE,
  gapfill = FALSE,
  ...
)
```

Arguments

`pedon`	A `PedonRecord`.
`systems`	Character vector. Any subset of `c("wrb2022","sibcs", "usda")`, or the literal `"all"` (default) to run every system.
`on_missing`	One of `"warn"` (default), `"silent"`, `"error"`. Forwarded verbatim to each classifier.
`include_familia`	Forwarded to `classify_sibcs` (default `TRUE`). Has no effect on the other systems.
`include_family`	Forwarded to `classify_usda` (default `FALSE`) to derive the USDA 5th-level family. No effect on the other systems.
`specifiers`	Forwarded to `classify_wrb2022` (default `FALSE`) to auto-attach WRB depth specifiers. No effect on the other systems.

`gapfill`	Forwarded to all three classifiers (default `FALSE` => byte-identical). Opt-in within-pedon depth gap-fill; see `gapfill_within_pedon`. Applied independently per system on a deep copy, so the caller's pedon is never mutated.
`...`	Additional named arguments are silently ignored.

Details

Each classifier still produces its own `ClassificationResult` with the full key trace and evidence grade – nothing is collapsed or homogenised. The wrapper exists for ergonomics, not abstraction.

Value

A named list with elements:

- `wrb` – `ClassificationResult` from `classify_wrb2022()` (or `NULL` if the system was skipped or errored).
- `sibcs` – as above, from `classify_sibcs()`.
- `usda` – as above, from `classify_usda()`.
- `summary` – a 1-row `data.frame` with one column per system, holding the resulting `$name` (or `NA` when the system was skipped / errored). Useful for tabulating many pedons in one shot.

Selecting a subset of systems

Pass `systems = c("wrb2022", "sibcs")` (or any other subset) to skip systems you don't need. Default `systems = "all"` runs all three.

Errors and partial results

If a single classifier raises an error, the corresponding slot of the returned list is set to `NULL` and a one-line warning is emitted (so you can rerun the offender on its own to see the full traceback). The other classifiers still run and their results are returned. This matches the spirit of `on_missing = "warn"` on the individual classifiers.

Side effects

None. The classifiers do not mutate `pedon`; the wrapper does not attach any side-channel state.

See Also

`classify_wrb2022`, `classify_sibcs`, `classify_usda`.

Examples

```
pr <- make_ferralsol_canonical()
all_three <- classify_all(pr)
all_three$summary

# WRB + USDA only (skip SiBCS):
classify_all(pr, systems = c("wrb2022", "usda"))$summary
```

`classify_by_spectral_neighbours` *Classify a soil by spectral similarity to OSSL reference profiles*

Description

Given a Vis-NIR (or MIR) spectrum and an OSSL reference library enriched with WRB / SiBCS / USDA labels, returns the K most spectrally similar profiles plus a probabilistic class prediction aggregated from their labels.

Usage

```
classify_by_spectral_neighbours(
  spectrum,
  ossl_library,
  system = c("wrb2022", "sibcs", "usda"),
  k = 25L,
  preprocess = "snv+sg1",
  region = NULL,
  verbose = TRUE
)
```

Arguments

`spectrum`	Numeric vector or 1-row matrix (the query spectrum). Must align (after preprocessing) with the column space of `ossl_library$Xr`.
`ossl_library`	A list with `Xr` (numeric matrix, rows = OSSL training profiles, cols = wavelengths) and `Yr` (data frame keyed by property; *must include* a column named `wrb_rsg` and / or `sibcs_ordem` / `usda_order` for the labels to aggregate over). `ossl_library` may also carry `lat` and `lon` columns in `Yr` for the regional filter.
`system`	One of `"wrb2022"` (default), `"sibcs"`, `"usda"`. Controls which label column of `Yr` is aggregated.
`k`	Number of nearest neighbours (default 25).
`preprocess`	Pre-processing pipeline; passed to `preprocess_spectra`. Default `"snv+sg1"`.
`region`	Optional `list(lat, lon, radius_km)` for a regional filter on `ossl_library$Yr$lat` / `lon`.
`verbose`	Emit a `cli` summary.

Details

This is the **spectral analogy** classifier. It does not replace the deterministic key in `classify_wrb2022` / `classify_sibcs` / `classify_usda`; instead it provides a high-prior "expected class" before the user has lab data, reducing the search space when collecting confirming attributes.

Value

A list with three elements:

`distribution` A `data.table` with columns `class`, `n_neighbours`, `probability` (= `n_neighbours / k`), sorted by probability.

`neighbours` A `data.table` with one row per neighbour (top K), columns `rank`, `distance`, `class`, plus any other columns present in `ossl_library$Yr`.

`query` The query metadata (system, k, region filter, n_library_rows, n_filtered).

Distance metric

By default we compute distances on PLS scores (matching the resemble / OSSL recipe), with PLS components fit on the OSSL reference Yr matrix. When `resemble` is unavailable, we fall back to PCA scores from `stats::prcomp` on the preprocessed Xr – a defensible-but-simpler heuristic.

Region filter

Optional `lat` / `lon` / `radius_km` arguments filter the OSSL library to profiles within `radius_km` (great-circle) of the query location before computing distances. This implements the "biome-aware" use case the architecture document calls for: a Cerrado profile shouldn't have its class inferred from spectral neighbours in the Boreal taiga.

See Also

`predict_ossl_mbl` (predicts attributes), `classify_wrb2022` (the deterministic key).

Examples

```
## Not run:
# Toy run against the bundled demo library (synthetic):
data(ossl_demo_sa)
# Inject a fake label column for the demo (real OSSL has it):
ossl_demo_sa$Yr$wrb_rsg <- sample(c("FR", "AC", "LX", "AL"),
                                    nrow(ossl_demo_sa$Yr),
                                    replace = TRUE)
query <- ossl_demo_sa$Xr[1, ]
res <- classify_by_spectral_neighbours(query, ossl_demo_sa,
                                        k = 10)
res$distribution    # ranked classes
res$neighbours      # the 10 most similar profiles

## End(Not run)
```

`classify_csv` *Classify a horizon spreadsheet in all three systems - one file, one line*

Description

The shortest path from a spreadsheet to an answer. Reads `file` with `read_pedon_csv` and returns the WRB 2022 / SiBCS / USDA names as a one-row `data.frame`. Missing attributes are handled silently (`on_missing = "silent"`).

Usage

```
classify_csv(
  file,
  site = NULL,
  sep = "auto",
  systems = c("wrb", "sibcs", "usda")
)
```

Arguments

`file`	Path to a `.csv` (comma) or `.tsv` (tab) file, one row per horizon.
`site`	Optional named list of site metadata (see `PedonRecord`). Defaults to `list(id = <file base name>)`.
`sep`	Field separator. `"auto"` (default) uses a tab for `.tsv` files and a comma otherwise.
`systems`	Character vector of systems to run; any of `"wrb"`, `"sibcs"`, `"usda"` (default: all three).

Details

The full `ClassificationResult` objects (with the key trace, evidence grade, qualifiers, ...) and the parsed `PedonRecord` are attached as attributes `"results"` and `"pedon"` for anyone who wants to dig deeper.

Value

A one-row `data.frame` with one column per system.

See Also

`read_pedon_csv`, `classify_all`.

Examples

```
f <- system.file("extdata", "perfil_exemplo.csv", package = "soilKey")
classify_csv(f)
```

`classify_from_documents`

Build a fully-classified 'PedonRecord' from documents in one call

---

Description

Highest-level entry point of the soilKey VLM pipeline. Given a soil-description PDF and / or a profile-wall photograph, this function:

Usage

```
classify_from_documents(
  pdf = NULL,
  image = NULL,
  fieldsheet = NULL,
  pedon = NULL,
  provider = "auto",
  model = NULL,
  systems = c("wrb", "sibcs", "usda"),
  report = NULL,
  overwrite = FALSE,
  verbose = TRUE
)
```

Arguments

`pdf`	Optional path to a soil-description PDF.
`image`	Optional path to a profile-wall image (JPG / PNG); if supplied, Munsell extraction is attempted with the configured provider.
`fieldsheet`	Optional path to a site-metadata field sheet (image or PDF).
`pedon`	Optional existing `PedonRecord`; when supplied, the function fills only the fields VLM extraction can fill (subject to the provenance-authority order).
`provider`	Either a provider name passed to `vlm_provider` (default `"ollama"`) OR a pre-built ellmer chat object (when you want full control over `system_prompt`, `api_key`, ...).
`model`	Optional model identifier; passed through to `vlm_provider()` when `provider` is a string. Defaults to the per-provider default from `default_model`.
`systems`	Character vector listing which classification systems to run; subset of `c("wrb", "sibcs", "usda")`. Default: all three.
`report`	Optional output path for a self-contained report (`.html` or `.pdf`). When supplied, `report` is called on the classification results + pedon. Default `NULL` (no report file).
`overwrite`	When merging extracted values into an existing pedon, allow VLM-extracted attributes to clobber already-recorded ones. Default `FALSE` – the provenance authority order (`measured` > `extracted_vlm`) is enforced by `PedonRecord$add_measurement()`.
`verbose`	Emit cli progress messages. Default `TRUE`.

Details

1. Constructs a vision-language provider chat object via `vlm_provider` (defaults to local Ollama with Gemma 4 edge for institutional independence and data sovereignty).
2. Extracts horizons from `pdf` via `extract_horizons_from_pdf`, Munsell colours from `image` via `extract_munsell_from_photo`, and site metadata from `fieldsheet` via `extract_site_from_fieldsheet`. Every extracted attribute is stamped `source = "extracted_vlm"` in the PedonRecord's provenance log.
3. Runs the three deterministic keys (`classify_wrb2022`, `classify_sibcs`, `classify_usda`). The VLM never classifies – the package's architectural invariant is preserved.
4. Optionally renders a one-pager HTML / PDF report via `report`.

At least one of `pdf`, `image` or `fieldsheet` must be supplied; you can also pass an existing partially-filled `PedonRecord` via pedon and let this function fill the gaps.

Value

A list with elements:

`pedon` The (mutated) `PedonRecord`.

`classifications` Named list with up to three `ClassificationResult` objects keyed by `wrb`, `sibcs`, `usda`.

`report` Path to the rendered report file (if `report = ...` was supplied), else `NULL`.

`provider` The chat-provider object actually used (useful for downstream debugging or cost accounting).

Why local-first by default

The default `provider = "ollama"` runs the entire VLM pipeline on the user's machine via Gemma 4 (edge variant, ~3 GB, multimodal text+image). No part of the soil description, photograph or field sheet ever leaves the local network. This is the recommended configuration for governmental surveys, indigenous land studies, and unpublished research data; it also makes the pipeline reproducible without an internet connection. Cloud providers (`"anthropic"`, `"openai"`, `"google"`) remain one argument away when they are the right call.

Architectural invariants preserved

- The VLM never classifies. Every extracted value carries `source = "extracted_vlm"`; the deterministic keys consume the resulting `PedonRecord` unaware of how each value was obtained.
- Provenance is preserved end-to-end. The `evidence_grade` on each `ClassificationResult` reflects whether decisive attributes came from `measured`, `predicted_spectra`, `extracted_vlm`, `inferred_prior`, or `user_assumed` – so a caller always knows how robust the classification is.
- Authority order is enforced. A pre-existing `measured` value is never silently overwritten by a later `extracted_vlm` value (unless `overwrite = TRUE`).

See Also

`vlm_provider`, `extract_horizons_from_pdf`, `classify_wrb2022`, `report`.

Examples

```
## Not run:
# The simplest possible end-to-end call -- local Gemma 4 edge.
res <- classify_from_documents(
  pdf      = "perfil_042_descricao.pdf",
  image    = "perfil_042_parede.jpg",
  report   = "perfil_042.html"
)
res$classifications$wrb$name
#> "Geric Ferric Rhodic Chromic Ferralsol (Clayic, Humic, Dystric, Ochric, Rubic)"

# Cloud provider for a one-shot, production run
res <- classify_from_documents(
  pdf      = "perfil_042_descricao.pdf",
  provider = "anthropic"
)

# Different Gemma 4 size on Ollama
res <- classify_from_documents(
  pdf      = "perfil_042_descricao.pdf",
  provider = "ollama",
  model    = "gemma4:31b"
)

## End(Not run)
```

`classify_from_photos` *Classify a soil profile from field photographs alone*

Description

A no-lab-data pipeline: profile photographs are sent to a vision-language model for Munsell-colour and (optionally) site-metadata extraction; the missing horizon attributes are back-filled from a SoilGrids depth prior; and the WRB 2022, SiBCS 5 and USDA Soil Taxonomy keys are run on the assembled `PedonRecord`.

Usage

```
classify_from_photos(
  images,
  lat = NULL,
  lon = NULL,
  country = NULL,
  provider = NULL,
```

```
  systems = c("wrb", "sibcs", "usda"),
  soilgrids = TRUE,
  depth_profiles = NULL,
  on_missing = "silent"
)
```

Arguments

`images`	Either a character vector of profile-photo paths, or a named list with elements `profile` (character vector, required) and `fieldsheet` (character vector, optional).
`lat, lon`	Optional decimal-degree coordinates. When supplied they seed `pedon$site` and are used for the SoilGrids fetch; a field sheet can also supply them through extraction.
`country`	Optional ISO-2 country code; passed through to the constructed pedon's site metadata.
`provider`	A vision-language provider: an **ellmer** chat object for live use, or a `MockVLMProvider` for testing and offline demos. Required – there is no default, so a real classification is never produced from canned data by accident.
`systems`	Character vector, any subset of `c("wrb", "sibcs","usda")`.
`soilgrids`	If `TRUE` (default) missing horizon attributes are back-filled from a SoilGrids depth prior via `apply_soilgrids_depth_prior`.
`depth_profiles`	Optional named list of six-slice SoilGrids depth profiles, forwarded to `apply_soilgrids_depth_prior`. Supplying it skips the network call.
`on_missing`	Forwarded to the classifiers; default `"silent"`.

Details

Because every value originates from a photograph or a spatial prior, the classification's evidence grade is low by construction (`D` for VLM-extracted attributes, `C` where a SoilGrids prior contributed). The result is a screening estimate, not a substitute for a described and sampled profile.

Value

A named list with one `ClassificationResult` per requested system (`$wrb`, `$sibcs`, `$usda`), the constructed `$pedon`, its `$provenance` ledger, and a one-row `$summary` data frame. If extraction yields no horizons the list instead carries `$error` and a `NULL` pedon.

See Also

`extract_munsell_from_photo`, `apply_soilgrids_depth_prior`, `compute_per_attribute_evidence_grade`.

Examples

```
## Not run:
# Live use with an ellmer chat:
res <- classify_from_photos(
  images   = list(profile = "profile.jpg", fieldsheet = "sheet.jpg"),
```

```
  lat = -22.7, lon = -43.6, country = "BR",
  provider = ellmer::chat_anthropic())
res$wrb$name
res$wrb$evidence_grade   # "D" or "C"

## End(Not run)
```

`classify_sibcs` *Classifica um pedon segundo o SiBCS 5a edicao (1o + 2o + 3o + 4o niveis)*

Description

v0.7 ligou as 13 ordens; v0.7.1 desce ao 2o nivel (subordens) via `run_sibcs_subordem`; v0.7.3 desce ao 3o nivel (Grandes Grupos) via `run_sibcs_grande_grupo` para as ordens progressivamente wiradas em `inst/rules/sibcs5/grandes-grupos/<ordem>.yaml` (Cap 14 Organossolos primeiro). Quando a subordem ainda nao tem bloco de Grandes Grupos, ou quando nenhum Grande Grupo passa (e nao ha catch-all default), a classificacao para no 2o nivel.

Usage

```
classify_sibcs(
  pedon,
  rules = NULL,
  on_missing = c("warn", "silent", "error"),
  include_familia = FALSE,
  gapfill = FALSE
)
```

Arguments

`pedon`	A `PedonRecord`.
`rules`	Conjunto de regras pre-carregado.
`on_missing`	Um de `"warn"` (default), `"silent"`, `"error"`.
`include_familia`	Quando `TRUE` (default `FALSE`), adiciona o 5o nivel categorico (Familia) via `classify_sibcs_familia`. O label textual da Familia aparece em `$trace$familia_label`, e a lista de `FamilyAttributes` em `$trace$familia`.
`gapfill`	Preenchimento opcional de lacunas por interpolacao intra-perfil, default `FALSE` (no-op; classificacao byte-identica). `TRUE` preenche celulas `NA` interiores dos atributos continuos por profundidade; um vetor de caracteres restringe aos atributos citados; uma lista nomeada e repassada a `gapfill_within_pedon`. Celulas preenchidas recebem proveniencia `inferred_prior`, baixando o grau de evidencia para `"C"`. Opera sobre copia profunda – o pedon do chamador nunca e modificado.

Value

Um `ClassificationResult` cujo `name` eh o nome completo da classe atribuida no nivel mais profundo (Grande Grupo > Subordem > Ordem) e `rsg_or_order` eh o nome da ordem (e.g. "Organossolos"). Os codigos de cada nivel e o trace ficam em `$trace`.

Examples

```
pedon <- make_latossolo_canonical()
res <- classify_sibcs(pedon)
res$name
```

`classify_sibcs_familia`

Classifica um perfil no 5o nivel categorico do SiBCS (Familia)

Description

Aplica as dimensoes pertinentes a ordem do solo e devolve uma lista nomeada de `FamilyAttribute`. O label textual da Familia eh formado adicionando-se cada `value` nao-nulo apos a designacao do 4o nivel, separados por virgulas (Cap 18, p 281).

Usage

```
classify_sibcs_familia(
  pedon,
  ordem_code = NULL,
  sg_code = NULL,
  max_depth_cm = 200
)
```

Arguments

`pedon`	A `PedonRecord`.
`ordem_code`	Codigo da ordem (1 letra: "P", "L", ...). Se `NULL`, sera derivado de `sg_code`.
`sg_code`	Codigo do subgrupo do 4o nivel (e.g. "PVdAr"). Opcional; usado para ajustes especificos por SG (e.g. forcar subgrupamento textural em arenicos/espessarenicos).
`max_depth_cm`	Profundidade da secao de controle (default 200 cm).

Details

Esta funcao NAO eh uma chave determinista: cada perfil recebe SIMULTANEAMENTE todos os adjetivos pertinentes (multi-rotulo).

Value

Lista nomeada de `FamilyAttribute`.

Status v0.7.14.A

Implementadas 5 dimensoes – grupamento textural, subgrupamento textural, distribuicao de cascalhos, constituicao esqueletica, tipo de horizonte superficial. Outras dimensoes (prefixos epi/meso/endo, saturacao de bases, alico, mineralogia, atividade da argila, oxidos de ferro, andico, especificos de Organossolos) adicionadas em sub-commits subsequentes.

References

Embrapa (2018), SiBCS 5a ed., Cap 18, pp 281-288.

`classify_usda` *Classify a pedon under USDA Soil Taxonomy (13th edition)*

Description

Walks the canonical USDA key (Order -> Suborder -> Great Group -> Subgroup) using YAML rule files at:

- `inst/rules/usda/key.yaml`: Order key (12 entries)
- `inst/rules/usda/suborders/<order>.yaml`
- `inst/rules/usda/great-groups/<order>.yaml`
- `inst/rules/usda/subgroups/<order>.yaml`

Usage

```
classify_usda(
  pedon,
  rules = NULL,
  on_missing = c("warn", "silent", "error"),
  include_family = FALSE,
  infer_temperature = TRUE,
  gapfill = FALSE
)
```

Arguments

`pedon`	A `PedonRecord`.
`rules`	Optional pre-loaded rule set.
`on_missing`	One of `"warn"` (default), `"silent"`, `"error"`.
`include_family`	If `TRUE`, derive and prepend the 5th-level family modifiers. Default `FALSE` (output byte-identical to earlier versions).
`infer_temperature`	When deriving the family, infer the soil temperature regime from latitude/elevation if `site$soil_temperature_regime` is absent (default `TRUE`). See `family_temperature_regime_usda`.

`gapfill`	Opt-in within-pedon depth gap-fill, default `FALSE` (no-op, classification stays byte-identical). `TRUE` fills interior `NA` cells of the continuous depth-trending attributes by linear interpolation from the profile's own measured horizons; a character vector restricts it to those attributes; a named list is passed to `gapfill_within_pedon`. Filled cells carry `inferred_prior` provenance, so the evidence grade drops to `"C"`. Runs on a deep copy – the caller's pedon is never mutated.

Details

With `include_family = TRUE` it additionally derives the 5th category, the **family** – a set of class modifiers (particle-size, mineralogy, CEC-activity, reaction, temperature regime, depth) PREPENDED to the subgroup name, e.g. *"fine, kaolinitic, isohyperthermic Rhodic Hapludox"*. See `classify_usda_family`.

Value

A `ClassificationResult` with deepest-level taxon name. Each level's trace is in `$trace`; the family attributes are in `$trace$family`.

References

Soil Survey Staff (2022). Keys to Soil Taxonomy, 13th edition. USDA Natural Resources Conservation Service.

Examples

```
pedon <- make_ferralsol_canonical()
res <- classify_usda(pedon)
res$name
# include the 5th (family) level:
classify_usda(pedon, include_family = TRUE)$name
```

`classify_usda_family` *Classify the USDA family (5th level) of a pedon*

Description

Runs the applicable family-modifier dimensions and returns them as a named list of `FamilyAttribute` objects (multi-label; each dimension is orthogonal). Mirrors `classify_sibcs_familia`.

Usage

```
classify_usda_family(
  pedon,
  order_code = NULL,
  subgroup_code = NULL,
  infer_temperature = TRUE
)
```

Arguments

`pedon`	A `PedonRecord`.
`order_code`	Optional USDA order code (selects applicable dimensions).
`subgroup_code`	Optional subgroup code (reserved for refinements).
`infer_temperature`	Passed to `family_temperature_regime_usda`.

Value

Named list of `FamilyAttribute` objects.

References

Soil Survey Staff (2022), KST 13th ed., Ch. 16–17.

See Also

`family_label_usda`, `classify_usda`.

`classify_via_smartsolos_api` *Classify a PedonRecord via Embrapa's SmartSolosExpert REST API*

Description

Sends a soilKey `PedonRecord` to the SmartSolosExpert REST endpoint maintained by Embrapa (Glauber Vaz's PROLOG-based implementation of the SiBCS classifier) and returns the resulting four-level classification (Ordem / Subordem / Grande Grupo / Subgrupo) wrapped in a soilKey `ClassificationResult`.

Usage

```
classify_via_smartsolos_api(
  pedon,
  api_key = Sys.getenv("AGROAPI_TOKEN"),
  endpoint = c("classification", "verification"),
  drenagem = NULL,
  reference_sibcs = NULL,
  base_url = "https://api.cnptia.embrapa.br/smartsolos/expert/v1",
  timeout_seconds = 30,
  post_fn = NULL,
  verbose = TRUE
)
```

Arguments

`pedon`	A `PedonRecord`.
`api_key`	Bearer token. Defaults to `Sys.getenv("AGROAPI_TOKEN")`. Required unless `post_fn` is supplied (test injection).
`endpoint`	One of `"classification"` (default; classify only) or `"verification"` (classify + compare against user-supplied `reference_sibcs`).
`drenagem`	Optional drainage class. Integer 1..8 or Portuguese string (`"bem drenado"` etc.).
`reference_sibcs`	Optional named list (`ordem`, `subordem`, `gde_grupo`, `subgrupo`) used by the `"verification"` endpoint as the user's reference.
`base_url`	Override base URL. Default `"https://api.cnptia.embrapa.br/smartsolos/expert/v1"`.
`timeout_seconds`	HTTP timeout (default 30).
`post_fn`	Internal: function with signature `function(payload) -> response_list` for unit tests. When supplied, the network is bypassed.
`verbose`	If `TRUE` (default), emits a one-line summary.

Details

This is an **external classifier** – the package does not host or replicate the PROLOG rules. The function exists so soilKey users can cross-validate the local classifier against an authoritative Embrapa-hosted reference. Use the `"verification"` endpoint to compare against your own user-supplied reference classification (the API returns a per-level match `summary` with counters `L0..L4`).

Authentication: register a free AgroAPI account at `https://www.agroapi.cnptia.embrapa.br/portal/`, subscribe to the SmartSolosExpert API and generate an access token. Pass it via the `AGROAPI_TOKEN` environment variable or the `api_key` argument.

Value

A `ClassificationResult` with `system = "SiBCS 5a edicao (SmartSolosExpert API)"` and the four taxonomic levels in `rsg_or_order` (Ordem) and `qualifiers` (Subordem / GdeGrupo / Subgrupo). Verification-mode responses additionally carry `trace$smartsolos_summary` (the per-level match counters `L0..L4`).

References

Vaz, G. J., Silva Neto, L. de F. da, & Barbedo, J. G. A. (2025). SmartSolos Expert: an expert system for Brazilian soil classification. *Smart Agricultural Technology*, 10, 100735. doi:10.1016/j.atech.2024.100735.

Vaz, G. J., Silva Neto, L. de F. da, Lima, R. N., & Oliveira, S. R. de M. (2019). Uma API para a classificacao de solos do Brasil. In *Anais do 12 Congresso Brasileiro de Agroinformatica* (SBIAGRO 2019), pp. 63-72. Ponta Grossa.

Vaz, G. J., Silva Jr, A. F., & Silva Neto, L. de F. da (2023). Brazilian soil data for taxonomic classification. *Redape*, V1. doi:10.48432/PYKKA7.

See Also

`classify_sibcs` for the local PROLOG-free classifier; `compare_smartsolos` for a side-by-side comparison helper; `benchmark_redape` for the gold-standard curated dataset published by the same authors.

Examples

```
## Not run:
Sys.setenv(AGROAPI_TOKEN = "<your token>")
res <- classify_via_smartsolos_api(make_argissolo_canonical())
res$rsg_or_order      # "ARGISSOLO"
res$qualifiers
#> $subordem  "VERMELHO"
#> $gde_grupo "Distrofico"
#> $subgrupo  "tipico"

## End(Not run)
```

`classify_with_engine_heuristic`

Classify a pedon with the engine chosen by 'pick_engine()'

Description

Convenience wrapper that routes `classify_wrb2022` / `classify_sibcs` / `classify_usda` through whichever engine the heuristic recommends for the specific pedon.

Usage

```
classify_with_engine_heuristic(
  pedon,
  system = c("wrb2022", "sibcs", "usda"),
  min_score = 3L,
  ...
)
```

Arguments

`pedon`	A `PedonRecord`.
`system`	One of `"wrb2022"`, `"sibcs"`, `"usda"`.
`min_score`	Forwarded to `pick_engine`.
`...`	Forwarded to the underlying classifier.

Value

The result of the chosen classifier (a `ClassificationResult`). The chosen engine is captured in `$trace$engine_used`.

`classify_with_uncertainty`

Posterior distribution over classification outcomes

Description

Runs `n` Monte-Carlo perturbations of a pedon and tallies the resulting classes into an empirical posterior. Unlike `classification_robustness`, the perturbation magnitude of every `(horizon, attribute)` cell is scaled by its provenance evidence grade (see `get_perturbation_scale`): an A-grade measurement is nudged by a few percent, an E-grade assumption by a third of its value. The posterior therefore reflects not just how close the profile sits to a key boundary, but how trustworthy the inputs that placed it there actually are.

Usage

```
classify_with_uncertainty(
  pedon,
  n = 200L,
  system = c("wrb2022", "sibcs", "usda"),
  level = c("rsg", "name"),
  scales = NULL,
  sensitivity = TRUE,
  seed = 42L
)
```

Arguments

`pedon`	A `PedonRecord`.
`n`	Number of Monte-Carlo draws (default 200).
`system`	One of `"wrb2022"`, `"sibcs"`, `"usda"`.
`level`	`"rsg"` (default; compare the RSG / order) or `"name"` (compare the full classification name, qualifiers included – strictly more uncertain).
`scales`	Optional named list overriding the default per-grade magnitudes; each element has the shape returned by `get_perturbation_scale`, keyed by grade letter.
`sensitivity`	If `TRUE` (default) also computes a leave-one-attribute-out sensitivity ranking. Set `FALSE` to skip that extra pass when only the posterior is needed.
`seed`	Random seed for reproducibility.

Value

A list of class `"soilkey_uncertainty"` with elements: `posterior` (named numeric vector summing to 1, sorted descending), `top1` (the modal class), `entropy` (Shannon entropy of the posterior, natural log), `sensitivity` (a `data.table` of `attribute / importance`, or `NULL`), `n_runs`, `n_success`, `baseline`, `system` and `level`.

See Also

`classification_robustness`, `get_perturbation_scale`, `compute_per_attribute_evidence_grade`.

Examples

```
p <- make_ferralsol_canonical()
u <- classify_with_uncertainty(p, n = 50, system = "wrb2022")
u$posterior   # P(RSG = x)
u$entropy     # near 0 for a robust profile
```

`classify_wrb2022` *Classify a pedon under WRB 2022*

Description

High-level classification entry point. Pre-computes the implemented diagnostic horizons (argic, ferralic, mollic) for transparent reporting, runs the key, and assembles a `ClassificationResult` with the trace, ambiguities, missing-data hints, evidence grade, and (in future) prior sanity check.

Usage

```
classify_wrb2022(
  pedon,
  prior = NULL,
  prior_threshold = 0.01,
  on_missing = c("warn", "silent", "error"),
  rules = NULL,
  strict = NULL,
  specifiers = FALSE,
  gapfill = FALSE
)
```

Arguments

`pedon`	A `PedonRecord`.
`prior`	Optional spatial prior – a `data.table` with columns `rsg_code` and `probability`, typically the return value of `spatial_prior`. If supplied, the result records a `prior_check` entry from `prior_consistency_check`; an inconsistent prior also emits a warning. The deterministic key is NEVER overridden by the prior.
`prior_threshold`	Probability below which the prior triggers an "inconsistent" warning (default 0.01).
`on_missing`	One of `"warn"` (default), `"silent"`, `"error"`. Behaviour when the trace reports missing attributes.
`rules`	Optional pre-loaded rule set.

`strict`	Logical or `NULL`. Controls WRB Tier-3 strict mode for the per-RSG numerical gates (Vertisols, Andosols, Gleysols, Planosols, Ferralsols, Chernozems, Kastanozems). When `NULL` (default) the gates follow `getOption("soilKey.rsg_strict", FALSE)`. Passing `TRUE` or `FALSE` forces strict mode on or off for the duration of this call; see the individual RSG-gate help pages (e.g. `ferralsol`) for the strengthened thresholds.
`specifiers`	Logical. When `TRUE`, auto-attach WRB 2022 Ch 5 depth specifiers (Epi-/Endo-/Bathy-/Amphi-/Panto-/Kato-) to depth-anchored qualifiers based on the diagnostic feature's actual depth – e.g. a gleyic feature confined to 50–100 cm yields `Endogleyic` instead of `Gleyic`. Default `FALSE` keeps the canonical names byte-identical. Surface / epipedon qualifiers are excluded (their depth is definitional).
`gapfill`	Opt-in within-pedon depth gap-fill, default `FALSE` (no-op, classification stays byte-identical). `TRUE` fills interior `NA` cells of the continuous depth-trending attributes by linear interpolation from the profile's own measured horizons; a character vector restricts it to those attributes; a named list is passed to `gapfill_within_pedon`. Filled cells carry `inferred_prior` provenance, so the evidence grade drops to `"C"`. Runs on a deep copy – the caller's pedon is never mutated.

Value

A `ClassificationResult`.

Examples

```
pedon <- make_ferralsol_canonical()
res <- classify_wrb2022(pedon)
res$name
```

---

clear_kst13_cache *Clear the in-memory KST13 cache*

Description

Useful when the vendored JSON files are updated mid-session. Frees ~3.1 MB.

Usage

```
clear_kst13_cache()
```

Value

`NULL`, invisibly. Called for its side effect of emptying the KST 13th-edition lookup cache.

`clear_ossl_cache` *Clear the soilKey OSSL cache*

Description

Removes the per-region cache files written by `download_ossl_subset`. Useful when a stale cache is suspected or when disk space is tight.

Usage

```
clear_ossl_cache(region = NULL, cache_dir = NULL, verbose = TRUE)
```

Arguments

`region`	Optional character vector of regions to clear; the default `NULL` clears every cached file under ‘tools::R_user_dir("soilKey", "cache")‘.
`cache_dir`	Cache directory (defaults to the soilKey user-cache dir).
`verbose`	If `TRUE`, prints which files were removed.

Value

Invisibly, the character vector of files that were removed.

`combine_priors` *Combine multiple spatial priors via weighted geometric mean*

Description

Given a list of priors (each a data.table with `rsg_code,probability`), pools them into a single distribution using a weighted geometric mean and renormalises to sum to 1.

Usage

```
combine_priors(priors, weights = NULL, epsilon = 1e-06)
```

Arguments

`priors`	A list of `data.tables` with columns `rsg_code` and `probability`.
`weights`	Optional non-negative numeric vector of length `length(priors)`. Defaults to equal weights. Will be renormalised to sum to 1.
`epsilon`	Smoothing floor for classes missing from a prior (default 1e-6). Must be > 0 – otherwise any class missing from a single prior is suppressed entirely.

Details

Geometric pooling has two desirable properties for soil-class priors:

1. externally Bayesian (the pooled posterior under any common likelihood matches what one would get by individual updates), and
2. zero-preserving: a class assigned probability 0 by any prior is suppressed in the pooled distribution. To avoid that, classes absent from a given prior are imputed with the smoothing constant `epsilon`.

Value

A `data.table` with columns `rsg_code`, `probability`, sorted by descending probability.

`compare_engines` *Side-by-side comparison of soilKey vs aqp diagnostic engines*

Description

Runs the soilKey hand-coded diagnostic and the aqp wrapper on the same pedon, returns both results plus an agreement flag. Useful for A/B benchmarks and for choosing which engine to use per dataset.

Usage

```
compare_engines(pedon, diagnostic = c("argic", "cambic"))
```

Arguments

`pedon`	A `PedonRecord`.
`diagnostic`	One of `"argic"` or `"cambic"`.

Value

A list with `soilkey`, `aqp`, `agree`.

`compare_smartsolos` *Cross-validate the local SiBCS classifier against the SmartSolosExpert API*

Description

Runs both `classify_sibcs` (local) and `classify_via_smartsolos_api` (remote PROLOG via Embrapa AgroAPI) on the same `PedonRecord` and tabulates agreement at each of the four SiBCS categorical levels.

Usage

```
compare_smartsolos(pedon, ...)
```

Arguments

`pedon`	A `PedonRecord`.
`...`	Forwarded to `classify_via_smartsolos_api`.

Value

A list with `local` and `remote` `ClassificationResults` plus a one-row `agreement` data.frame with columns `ordem`, `subordem`, `gde_grupo`, `subgrupo`, `n_match`.

Examples

```
## Not run:
Sys.setenv(AGROAPI_TOKEN = "<your token>")
cmp <- compare_smartsolos(make_argissolo_canonical())
cmp$agreement

## End(Not run)
```

`compute_ki` *Ki (silica:alumina molar) – SiBCS Cap 1, p 32*

Description

Calcula o indice molar Ki = SiO2 / Al2O3 a partir de teores percentuais por ataque sulfurico-NaOH (Embrapa Manual de Metodos). Massas molares: 60.08 (SiO2), 101.96 (Al2O3):

Usage

```
compute_ki(sio2_pct, al2o3_pct)
```

Arguments

`sio2_pct`	Teor de SiO2 por ataque sulfurico (%).
`al2o3_pct`	Teor de Al2O3 por ataque sulfurico (%).

Details

Ki (molar) = (% SiO2 / 60.08) / (% Al2O3 / 101.96) $\approx$ 1.6973 $\times$ (% SiO2 / % Al2O3)

Value

Ki molar (numeric); NA se algum input for NA ou Al2O3 ≤ 0.

References

Embrapa (2018), SiBCS 5a ed., Cap 1, p 32; Embrapa Manual de Metodos de Analise de Solo (3a ed., 2017).

Examples

```
compute_ki(sio2_pct = 18, al2o3_pct = 20)  # ~1.53, abaixo do limite latossolico
```

`compute_kr` *Kr (silica:sesquioxidos molar) – SiBCS Cap 1, p 32*

Description

Calcula o indice molar Kr = SiO2 / (Al2O3 + Fe2O3) usando massas molares 60.08 (SiO2), 101.96 (Al2O3) e 159.69 (Fe2O3):

Usage

```
compute_kr(sio2_pct, al2o3_pct, fe2o3_pct)
```

Arguments

`sio2_pct`	Teor de SiO2 por ataque sulfurico (%).
`al2o3_pct`	Teor de Al2O3 por ataque sulfurico (%).
`fe2o3_pct`	Teor de Fe2O3 por ataque sulfurico (%).

Details

Kr (molar) = (% SiO2 / 60.08) / (% Al2O3 / 101.96 + % Fe2O3 / 159.69)

Value

Kr molar (numeric); NA se algum input for NA ou denominador ≤ 0.

References

Embrapa (2018), SiBCS 5a ed., Cap 1, p 32.

Examples

```
compute_kr(sio2_pct = 18, al2o3_pct = 20, fe2o3_pct = 12)
```

`compute_per_attribute_evidence_grade`

Per-attribute provenance-aware evidence grade

Description

Resolves the evidence grade of every `(horizon, attribute)` cell that carries a provenance entry. Where a cell has more than one entry (a value re-sourced over the profile's lifetime) the most authoritative source wins, mirroring `PedonRecord`'s own authority order.

Usage

```
compute_per_attribute_evidence_grade(pedon)
```

Arguments

`pedon`	A `PedonRecord`.

Details

Grades: `A` measured, `B` predicted from spectra, `C` inferred from a spatial prior, `D` extracted by a vision-language model, `E` user-assumed.

Value

A `data.table` with columns `horizon_idx`, `attribute` and `grade`, sorted by horizon then attribute. A pedon with no provenance entries yields a zero-row table.

See Also

`classify_from_photos`, the global evidence grade reported on every `ClassificationResult`.

Examples

```
p <- make_ferralsol_canonical()
compute_per_attribute_evidence_grade(p)   # all-measured -> all grade A
```

`continuous_rock` *Continuous rock (WRB 2022 Ch 3.2.5)*

Description

Consolidated material below the soil. v0.3.3: detects via designation `R` or `Cr` on the lowermost (or any) layer.

Usage

```
continuous_rock(pedon)
```

Arguments

`pedon`	A `PedonRecord`.

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`coverage_report` *Honest taxonomic-completeness report*

Description

Measures, by NAME, exactly which canonical taxa/qualifiers the package's deterministic rule base registers, replacing hand-maintained coverage claims with an auditable, reproducible diff. For `"usda_subgroup"` the canonical reference is the Soil Taxonomy 13th-edition subgroup set from `kst13_codes`; for `"wrb_qualifiers"` it is the WRB 2022 principal + supplementary qualifier set from `wrb2022_canonical`.

Usage

```
coverage_report(
  system = c("usda_subgroup", "usda_great_group", "usda_suborder", "wrb_qualifiers",
    "sibcs"),
   write = FALSE,
   report_dir = NULL
)
```

Arguments

`system`	Which axis to measure. USDA taxon levels against the Soil Taxonomy 13th-edition code set (`kst13_codes`): `"usda_subgroup"` (default), `"usda_great_group"`, `"usda_suborder"`. WRB 2022 qualifiers against `wrb2022_canonical`: `"wrb_qualifiers"` – here "covered" means the `qual_*` function exists *and* is a genuine implementation (not an unconditional passed = `NA` stub), and the inert ones are returned in `$stubs`. `"sibcs"` has no external canonical class list, so it honestly reports registered class counts per level only (no percentage).
`write`	If `TRUE`, also write a Markdown summary to `report_dir`. Default `FALSE`.
`report_dir`	Directory for the Markdown report when `write` = `TRUE`. Defaults to `inst/benchmarks/reports` inside the installed package.

Value

Invisibly, a list with `$overall` (one-row data frame: `system`, `level`, `canonical_n`, `registered_n`, `covered_n`, `missing_n`, `pct`), `$by_group` (per order, or per principal/supplementary), `$missing` (canonical names not registered), `$extra` (registered names absent from the canonical set), and – for `"wrb_qualifiers"` – `$stubs` (functions that exist but are inert). A compact summary is printed as a side effect.

Examples

```
cov <- coverage_report("usda_subgroup")
cov$overall
head(cov$missing)
```

`cryic_conditions` *Cryic conditions (WRB 2022)*

Description

Tests whether continuous frozen / permafrost material occurs within the upper `max_top_cm`. Two alternative paths qualify per WRB 2022:

1. **Permafrost temperature**: a layer at top_cm <= `max_top_cm` (default 100) with `permafrost_temp_C` <= `max_temp_C` (default 0 C).
2. **Designation pattern**: a layer at top_cm <= `max_top_cm` with designation containing suffix `"f"` (frozen) or matching `"^Cf"` / `"perma"`. Used as a fallback when the temperature field is not in the pedon (typical of legacy survey data).

Either path qualifies. Diagnostic of Cryosols.

Usage

```
cryic_conditions(pedon, max_top_cm = 100, max_temp_C = 0)
```

Arguments

`pedon`	A `PedonRecord`.
`max_top_cm`	Maximum top depth (cm) (default 100).
`max_temp_C`	Maximum mean annual permafrost-zone temperature (deg C) for the temperature path (default 0).

Value

A `DiagnosticResult`.

References

IUSS Working Group WRB (2022), Chapter 5, Cryosols.

`DiagnosticResult` *DiagnosticResult: structured outcome of a diagnostic test*

Description

DiagnosticResult: structured outcome of a diagnostic test

DiagnosticResult: structured outcome of a diagnostic test

Details

Returned by every WRB or SiBCS diagnostic function (e.g. `argic`, `ferralic`, `mollic`). A DiagnosticResult never reduces to a bare TRUE/FALSE — it always carries (a) which layers satisfied the criteria, (b) the per-sub-test evidence, (c) which attributes would have been required but are missing, and (d) the literature reference for the diagnostic definition.

`passed` is `TRUE`/`FALSE`/`NA`; `NA` means the test could not be evaluated because critical attributes were missing. This three-valued semantics propagates through the rule engine — an indeterminate test does not silently fail.

Public fields

`name` Character. Name of the diagnostic (e.g. `"argic"`).

`passed` Logical. `TRUE`, `FALSE`, or `NA`.

`layers` Integer vector. Indices of horizons that satisfy the diagnostic.

`evidence` Named list. Sub-test results, each itself a list with at least `passed`, `layers`, and `missing`.

`missing` Character vector. Attribute names that would have been needed but were NA.

`reference` Character. Literature citation for this diagnostic.

`notes` Character. Free-form notes (interpretation choices, edge cases hit).

Methods

Public methods:

- `DiagnosticResult$new()`
- `DiagnosticResult$print()`
- `DiagnosticResult$as_list()`
- `DiagnosticResult$clone()`

Method `new()`**:** Build a DiagnosticResult.

Usage:

```
DiagnosticResult$new(
  name,
  passed = NA,
  layers = integer(0),
  evidence = list(),
  missing = character(0),
  reference = NA_character_,
  notes = NA_character_
)
```

Arguments:

`name` Diagnostic name.

`passed` `TRUE`/`FALSE`/`NA`.

`layers` Integer vector of horizon indices that satisfied.

`evidence` Named list of sub-test results.

`missing` Character vector of missing attribute names.

`reference` Citation string.

`notes` Free-form notes.

Method `print()`**:** Pretty-print the result with sub-test breakdown.

Usage:

```
DiagnosticResult$print(...)
```

Arguments:

`...` Ignored (S3 print signature compatibility).

Method `as_list()`**:** Return the result as a plain list (for serialization).

Usage:

```
DiagnosticResult$as_list()
```

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
DiagnosticResult$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

`distrofico` *Solo distrofico (SiBCS Cap 1, p 30)*

Description

Negacao operacional de `eutrofico`: V < 50% no horizonte diagnostico subsuperficial.

Usage

```
distrofico(pedon, max_v = 50)
```

Arguments

`pedon`	A `PedonRecord`.
`max_v`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`dolomitic_material` *Dolomitic material (WRB 2022 Ch 3.3.5): \>= 2% Mg-rich carbonate, CaCO3/MgCO3 < 1.5. v0.3.3: detects via designation pattern* `kdo|do|magn` *as proxy when ratio data missing.*

Description

Dolomitic material (WRB 2022 Ch 3.3.5): \>= 2% Mg-rich carbonate, CaCO3/MgCO3 < 1.5. v0.3.3: detects via designation pattern `kdo|do|magn` as proxy when ratio data missing.

Usage

```
dolomitic_material(pedon)
```

Arguments

`pedon`	A `PedonRecord`.

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`download_bdsolos` *Download the BDsolos consulta-publica CSV (experimental, requires chromote)*

Description

Drives the Embrapa BDsolos web form via headless Chrome (`chromote`) to produce a CSV of all profiles + all attributes. Marked **experimental**: heavy queries (no UF filter) frequently overload the Embrapa server. Prefer `filter_uf` = batches of one or two states at a time and stitch the resulting CSVs.

Usage

```
download_bdsolos(
  out_path,
  accept_terms = FALSE,
  filter_uf = NULL,
  attributes = "default",
  timeout_seconds = 600,
  chromote_session = NULL,
  verbose = TRUE
)
```

Arguments

`out_path`	File path for the downloaded CSV.
`accept_terms`	Logical. Must be `TRUE` to proceed; the function aborts otherwise. Documents informed consent to the BDsolos terms (personal/academic use, ABNT citation).
`filter_uf`	Optional 2-letter UF code (e.g. `"RJ"`, `"SC"`). Strongly recommended – the full-table query often times out.
`attributes`	Character vector. Which attribute groups to request. Defaults to the full SiBCS-classification-relevant set (Identificacao + Localizacao + Classificacao for Pontos de Amostragem, Identificacao + Morfologicas + Fisicas + Quimicas for Horizontes; Mineralogicas excluded for performance). Pass `"all"` to include Mineralogicas.
`timeout_seconds`	Total timeout for the AJAX query. Default 600 (10 min).
`chromote_session`	Optional pre-built `chromote::ChromoteSession`. Useful to share a session across calls.
`verbose`	If `TRUE` (default), prints progress.

Details

Per the Embrapa terms-of-use, the data is licensed for personal / academic use and publications must cite the source per ABNT. **Set `accept_terms = TRUE` to acknowledge this and let the function click "Concordo" on your behalf.**

Value

File path to the downloaded CSV (invisible).

See Also

`load_bdsolos_csv`, `inspect_bdsolos_csv`.

Examples

```
## Not run:
# Single UF (fast, recommended)
download_bdsolos("soil_data/bdsolos/RJ.csv",
                  accept_terms = TRUE,
                  filter_uf    = "RJ")

# Stitch multiple UFs
for (uf in c("RJ", "SP", "MG", "ES")) {
  download_bdsolos(file.path("soil_data/bdsolos",
                               paste0(uf, ".csv")),
                    accept_terms = TRUE, filter_uf = uf)
}

# Then load all of them
csvs <- list.files("soil_data/bdsolos", "\\.csv$", full.names = TRUE)
all_pedons <- unlist(lapply(csvs, load_bdsolos_csv), recursive = FALSE)
length(all_pedons)

## End(Not run)
```

`download_extdata_cache` *Download one or more soilKey lazy-fetch caches from GitHub Release*

Description

soilKey ships four large benchmark caches (KSSL, KSSL+NASIS, AfSP, WoSIS stratified) that are too large to embed in the CRAN source tarball. Since v0.9.94 they are pinned to a versioned GitHub Release and downloaded on demand into the user cache directory at `tools::R_user_dir("soilKey", "data")`.

Usage

```
download_extdata_cache(
  which = "all",
  release = .SOILKEY_LAZY_FETCH_RELEASE,
  overwrite = FALSE,
  verbose = TRUE
)
```

Arguments

`which`	Character vector of cache names to download. `"all"` (default) downloads every lazy-fetch cache. Valid names: `"afsp_sample"`, `"kssl_sample"`, `"kssl_nasis_sample"`, `"wosis_stratified_sample"`.
`release`	GitHub Release tag to pull from (default `"v0.9.94-data"`). Override only if you maintain a local mirror.
`overwrite`	If `TRUE`, redownload even if the file is already present in the user cache (default `FALSE`).
`verbose`	Print progress (default `TRUE`).

Details

On first call to any of `load_kssl_sample()`, `load_kssl_nasis_sample()`, `load_afsp_sample()`, or `load_wosis_stratified_sample()`, soilKey checks for the file in the user cache. If missing, the loader prompts (interactive sessions only) to download. Use `download_extdata_cache()` to eagerly populate the cache without prompting.

Value

Invisibly, a named character vector of local paths to the downloaded files.

Examples

```
## Not run:
# Download every lazy-fetch cache once, ahead of any benchmark run:
download_extdata_cache()

# Or just the WRB AfSP sample:
download_extdata_cache("afsp_sample")

## End(Not run)
```

`download_ossl_subset` *Download an OSSL subset and return an ‘ossl_library‘ artefact*

Description

Fetches a region-filtered subset of the Open Soil Spectral Library (Sanderman et al. 2024) and assembles it into the ‘list(Xr, Yr, metadata)‘ shape consumed by `predict_ossl_mbl` and `predict_ossl_plsr_local`. The result is cached under ‘tools::R_user_dir("soilKey", "cache")‘ so subsequent calls in the same session (or future R sessions) skip the network.

Usage

```
download_ossl_subset(
  region = c("global", "south_america", "north_america", "europe", "africa", "asia",
    "oceania"),
  properties = c("clay_pct", "sand_pct", "silt_pct", "cec_cmol", "bs_pct", "ph_h2o",
    "oc_pct", "fe_dcb_pct", "caco3_pct"),
  wavelengths = 350:2500,
  endpoint = NULL,
  cache_dir = NULL,
  force = FALSE,
  verbose = TRUE
)
```

Arguments

`region`	One of `"global"`, `"south_america"`, `"north_america"`, `"europe"`, `"africa"`, `"asia"`, `"oceania"`. Filters the OSSL training rows by their site coordinates' continent.
`properties`	Character vector of OSSL property names to keep in 'Yr' (drops other reference columns to keep the artefact small). Defaults to the WRB-relevant set used by `fill_from_spectra`.
`wavelengths`	Integer vector of wavelengths (nm) the returned `Xr` matrix will be interpolated to. Defaults to Vis-NIR/SWIR (350-2500 nm at 1-nm resolution, 2151 columns).
`endpoint`	OSSL HTTP endpoint serving the JSON manifest; overrideable via `options(soilKey.ossl_endpoint = ...)` for testing or for using a private mirror. The default is the public Soil Spectroscopy GG bucket.
`cache_dir`	Cache directory; defaults to `tools::R_user_dir("soilKey", "cache")`.
`force`	If `TRUE`, re-fetches even when a cached subset exists.
`verbose`	If `TRUE`, emits a 'cli' summary of the fetch.

Details

This function intentionally does **not** fall back to the synthetic predictor on network failure – a missing OSSL artefact is a real condition that the caller must handle, and silent fallback would make benchmarks meaningless.

Value

A list with elements `Xr` (numeric matrix, rows = training profiles, columns = wavelengths in nm), `Yr` (data.frame with the requested property columns, rows aligned to `Xr`), and `metadata` (snapshot date, region, n profiles, source URL, and the SHA-256 of the cache file). Pass it as the `ossl_library` argument to `fill_from_spectra` or `predict_ossl_mbl`.

References

Sanderman, J., Savage, K., Dangal, S.R.S., Duran, G., Rivard, C., Cardona, M.T., Sandzhieva, A., Aramian, A. & Safanelli, J.L. (2024). Soil Spectroscopy for Global Good – the Open Soil Spectral Library (OSSL). `https://soilspectroscopy.org/`.

`download_ossl_subset_with_labels` *Download an OSSL subset and attach WRB / SiBCS / USDA labels*

Description

Fetches a region-filtered slice of the Open Soil Spectral Library via `download_ossl_subset` and post-joins WRB Reference Soil Group labels from WoSIS GraphQL by spatial nearest-neighbour. The resulting artefact has the canonical `list(Xr, Yr, metadata)` shape – with extra columns in `Yr`: `wrb_rsg`, `wrb_label_source`, `wrb_label_distance_km`, plus optionally `sibcs_ordem` and `usda_order` when `translate_systems = TRUE`.

Usage

```
download_ossl_subset_with_labels(
  region = c("global", "south_america", "north_america", "europe", "africa", "asia",
    "oceania"),
  max_distance_km = 5,
  wosis_endpoint = NULL,
  translate_systems = TRUE,
  max_to_label = Inf,
  verbose = TRUE,
  query_fn = NULL,
  ...
)
```

Arguments

`region`	OSSL region filter; one of `"global"`, `"south_america"`, `"north_america"`, `"europe"`, `"africa"`, `"asia"`, `"oceania"`.
`max_distance_km`	WoSIS spatial-join tolerance in kilometres (default 5). Profiles whose nearest WRB-labeled WoSIS neighbour is farther than this are left unlabeled.
`wosis_endpoint`	Override for the WoSIS GraphQL endpoint (default `getOption("soilKey.wosis_graphql")`). The canonical value is `"https://graphql.isric.org/wosis/graphql"`.
`translate_systems`	If `TRUE` (default), also adds `sibcs_ordem` and `usda_order` columns derived from the WRB label via the Schad (2023) Annex Table 1 / SiBCS 5ª ed. Annex A correspondence. Those translations are 1:N for some classes; we pick the most-common partner and tag rows where the translation is genuinely ambiguous.
`max_to_label`	Maximum number of profiles to query against WoSIS (default `Inf`). WoSIS throttles aggressive queries; cap this when running interactive demos.
`verbose`	Emit `cli` progress messages.

`query_fn`	Optional injection of the per-coordinate WoSIS query function. Default uses `.query_nearest_wosis_wrb`. Tests pass a stub here to exercise the join logic without network.
`...`	Forwarded to `download_ossl_subset`.

Value

A list with `Xr` (numeric matrix), `Yr` (data frame with the labels attached), and `metadata` (list with the OSSL fetch metadata + the join statistics: number of profiles labeled, average / max distance, WoSIS endpoint, snapshot date).

Why this function exists

OSSL stores Vis-NIR / MIR spectra and lab data but typically lacks WRB Reference Soil Group labels on most profiles (KSSL data is USDA-flavoured; non-US contributions are inconsistent). WoSIS, by contrast, archives ~228 000 profiles with WRB labels but no spectra. This function bridges the two so the user can run `classify_by_spectral_neighbours` on a real-data OSSL library without having to do the spatial join themselves.

Caveats and provenance

WRB labels obtained via spatial join are **weak labels**. The same physical location may have been classified differently across surveys (different WRB editions, different interpretations). Each row carries:

- `wrb_label_source = "wosis_spatial_join"`: label inherited from a WoSIS neighbour within `max_distance_km`.
- `wrb_label_distance_km`: the distance to that neighbour (NA when no neighbour was found within tolerance).
- `wrb_label_source = "ossl_native"`: label was already present in OSSL Yr (rare; preserved verbatim).
- `wrb_label_source = "missing"`: no neighbour within tolerance; the row stays unlabeled and will be skipped downstream.

Treat the labels as priors, not ground truth.

See Also

`download_ossl_subset`, `classify_by_spectral_neighbours`.

Examples

```
## Not run:
# Real OSSL South-America subset with WRB labels:
lib <- download_ossl_subset_with_labels(
  region          = "south_america",
  max_distance_km = 10
)
table(lib$Yr$wrb_rsg, useNA = "always")
```

```
table(lib$Yr$wrb_label_source)

# Drop into the spectral analogy classifier:
res <- classify_by_spectral_neighbours(
  spectrum     = my_query_spectrum,
  ossl_library = lib,
  k            = 25,
  region       = list(lat = -22.7, lon = -43.7,
                      radius_km = 500)
)

## End(Not run)
```

`download_redape_dataset` *Download the curated Redape GeoTab dataset (Vaz et al 2023)*

Description

Enumerates the dataset via the Dataverse API and downloads all JSON profile files (the structured / interoperable format used by the curators) into `dest_dir`. Skips files already present unless `overwrite = TRUE`.

Usage

```
download_redape_dataset(
  dest_dir,
  dataset_doi = .REDAPE_GEOTAB_DOI,
  include_rtf = FALSE,
  overwrite = FALSE,
  verbose = TRUE
)
```

Arguments

`dest_dir`	Destination directory for the JSON files.
`dataset_doi`	DOI of the dataset (default: the Vaz 2023 dataset).
`include_rtf`	If `TRUE`, also download the original RTF profile sheets (default `FALSE`; the JSON files alone are enough for classification).
`overwrite`	If `TRUE`, re-download files that already exist locally.
`verbose`	Print progress (default `TRUE`).

Value

Character vector of paths to the downloaded files.

References

Vaz, G. J., Silva Jr, A. F., & Silva Neto, L. de F. da (2023). Brazilian soil data for taxonomic classification. Redape, V1. doi:10.48432/PYKKA7.

`duric_horizon` *Duric horizon (WRB 2022)*

Description

Tests for >= 10% volume of duripan nodules (Si-cemented) within a horizon at least 10 cm thick. Diagnostic of Durisols.

Usage

```
duric_horizon(pedon, min_thickness = 10, min_duripan_pct = 10)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Minimum thickness (cm; default 10 per WRB 2022).
`min_duripan_pct`	Minimum duripan volume % (default 10 per WRB 2022).

Value

A `DiagnosticResult`.

v0.3.1: thresholds aligned with WRB 2022 Ch 3.1.7 (10%, 10 cm) – previous v0.3 used 15%/15 cm. Petroduric (cemented continuous duripan) detection still deferred and will be added in v0.4.

References

IUSS Working Group WRB (2022), Chapter 3.1.7 – Duric horizon (p. 41).

`duripa` *Duripa (SiBCS Cap 2, p 74; v0.7)*

Description

Reuso de `duric_horizon` (WRB Ch 3.1): subsuperficial cimentado por silica, continuo ou em \>= 50% volume.

Usage

```
duripa(pedon, ...)
```

Arguments

`pedon`	A `PedonRecord`.
`...`	Reserved for future arguments.

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`ensure_horizon_schema` *Coerce a horizons-like data.frame to the canonical schema*

Description

Adds any missing canonical columns as NAs of the right type and reorders canonical columns first. Extra user-supplied columns are preserved at the end. Coerces character values to numeric where the schema requires it.

Usage

```
ensure_horizon_schema(h)
```

Arguments

`h`	Input data.frame or data.table.

Value

A `data.table` with the canonical horizon columns present, in canonical order, with extra columns preserved at the end.

Examples

```
h <- ensure_horizon_schema(data.frame(top_cm = 0, bottom_cm = 20))
"designation" %in% names(h)
```

`eutrofico` *Solo eutrofico (SiBCS Cap 1, p 30)*

Description

Returns TRUE se a saturacao por bases (V%) >= 50% no horizonte diagnostico subsuperficial (B ou C). 65% para A chernozemico.

Usage

```
eutrofico(pedon, min_v = 50)
```

Arguments

`pedon`	A `PedonRecord`.
`min_v`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`evaluate_rsg_tests` *Evaluate the test block of a single RSG*

Description

Given a parsed `tests` block from a YAML key entry, evaluates the appropriate combinator and returns a list with `passed`, `evidence`, `missing`, and (optionally) `notes`.

Usage

```
evaluate_rsg_tests(pedon, tests)
```

Arguments

`pedon`	A `PedonRecord`.
`tests`	A `tests` block from the YAML.

Value

A list summarising the test outcome.

`extract_horizons_from_pdf` *Extract horizons from a soil description PDF*

Description

Reads a PDF (typically a soil survey chapter, field-sheet scan, or thesis appendix), prompts the configured VLM to extract horizon attributes against `inst/schemas/horizon.json`, and merges the result into `pedon`. Every extracted attribute is recorded with `source = "extracted_vlm"` and the model's reported confidence and verbatim source quote.

Usage

```
extract_horizons_from_pdf(
  pedon,
  pdf_path = NULL,
  provider,
  max_retries = 3L,
  overwrite = FALSE,
  prompt_name = "extract_horizons",
  schema_name = "horizon",
  pdf_text = NULL
)
```

Arguments

`pedon`	A `PedonRecord` to merge into. Mutated in place AND returned invisibly.
`pdf_path`	Path to the PDF file. Either `pdf_path` or `pdf_text` must be supplied.
`provider`	A chat provider from `vlm_provider` (or a `MockVLMProvider` for testing).
`max_retries`	Integer; how many times to re-prompt on validation failure. Default 3.
`overwrite`	If `TRUE`, lower-authority values are allowed to clobber higher-authority ones. Default `FALSE`.
`prompt_name`	Override the default prompt template (`"extract_horizons"`).
`schema_name`	Override the default schema (`"horizon"`).
`pdf_text`	Optional alternative to `pdf_path`: the already-extracted description text. Useful for smoke tests, unit tests without `pdftools`, and for already-OCR'd field-sheet text.

Details

The PedonRecord's authority order guarantees that values already tagged `"measured"` are never silently overwritten by VLM extraction unless `overwrite = TRUE`.

If the PDF is long (more than ~30,000 characters), it is chunked page-by-page and each page is sent independently. This is a conservative-but-simple strategy; for very long surveys callers should pre-chunk and call this function once per profile.

Value

Invisibly, the (mutated) `pedon`. Carries a `"vlm_extraction"` attribute with the parsed response, number of attempts, and number of provenance entries added.

Failure modes

- If `pdftools` is not installed -> error.
- If the PDF cannot be read -> error.
- If the VLM response fails JSON parse / schema validation after `max_retries` + 1 attempts -> error from `validate_or_retry`.

`extract_munsell_from_photo`

Extract Munsell color from a profile photo

Description

Sends the photo to a multimodal VLM with a prompt that asks the model to estimate Munsell hue / value / chroma per visible horizon (when a Munsell reference card is in frame). Recorded as `extracted_vlm` with the model's self-reported confidence; photos without a reference card should yield confidence below 0.5 per the prompt specification.

Usage

```
extract_munsell_from_photo(
  pedon,
  image_path,
  provider,
  max_retries = 3L,
  overwrite = FALSE,
  prompt_name = "extract_munsell_from_photo",
  schema_name = "horizon"
)
```

Arguments

`pedon`	A `PedonRecord`.
`image_path`	Path to the image file (JPG / PNG).
`provider`	A chat provider from `vlm_provider` (or a `MockVLMProvider` for testing).
`max_retries`	Integer; how many times to re-prompt on validation failure. Default 3.
`overwrite`	If `TRUE`, lower-authority values are allowed to clobber higher-authority ones. Default `FALSE`.
`prompt_name`	Override the default prompt template (`"extract_horizons"`).
`schema_name`	Override the default schema (`"horizon"`).

Details

Quantitative non-color attributes (clay %, CEC, pH, etc.) are **never** extracted from photos, by prompt-level instruction. If the model returns one anyway, it is silently dropped.

Value

Invisibly, the mutated `pedon`, with the photo added to `pedon$images`.

`extract_site_from_fieldsheet`

Extract site metadata from a field-sheet image

Description

Sends a photographed / scanned field sheet to a multimodal VLM and merges the extracted site-level metadata (lat, lon, elevation, parent material, land use, etc.) into `pedon$site`. Existing fields are preserved unless `overwrite = TRUE`; only NULL fields are filled.

Usage

```
extract_site_from_fieldsheet(
  pedon,
  image_path,
  provider,
  max_retries = 3L,
  overwrite = FALSE,
  prompt_name = "extract_site_metadata",
  schema_name = "site"
)
```

Arguments

`pedon`	A `PedonRecord`.
`image_path`	Path to the field-sheet image.
`provider`	A chat provider from `vlm_provider` (or a `MockVLMProvider` for testing).
`max_retries`	Integer; how many times to re-prompt on validation failure. Default 3.
`overwrite`	If `TRUE`, lower-authority values are allowed to clobber higher-authority ones. Default `FALSE`.
`prompt_name`	Override the default prompt template (`"extract_horizons"`).
`schema_name`	Override the default schema (`"horizon"`).

Value

Invisibly, the mutated `pedon`.

`familia_mineralogia_argila_geral` *Familia: mineralogia da fracao argila (geral, nao-Latossolos)*

Description

Classifica a mineralogia da argila para Argissolos, Cambissolos, Plintossolos, Luvissolos, Nitossolos, Vertissolos, Chernossolos, Planossolos, Gleissolos quando ha informacao quantitativa de atividade da argila e/ou Ki/Kr. Cobre as classes nao endereçadas por `familia_mineralogia_argila_latossolo`:

- `esmectitica`: T_argila >= `ta_threshold` (default 27 cmolc/kg argila), indicando dominancia de argilas 2:1 expansivas (esmectita / vermiculita / micas hidratadas).
- `caulinitica`: Ki >= `ki_caulinitico_min` (default 0.75) e Kr >= `kr_caulinitico_min` (default 0.75), alem de T_argila < `ta_threshold`.
- `oxidica`: Kr < `kr_caulinitico_min`, indicando predominancia de oxihidrooxidos de Fe e Al.
- `mista`: nenhum dos outros gates fechou conclusivamente – evidencia heterogenea ou incompleta.

Quando os tres atributos (T_argila, Ki, Kr) estiverem ausentes, o resultado fica `NULL` e os atributos faltantes sao reportados.

Usage

```
familia_mineralogia_argila_geral(
  pedon,
  max_depth_cm = 200,
  ta_threshold = 27,
  ki_caulinitico_min = 0.75,
  kr_caulinitico_min = 0.75
)
```

Arguments

`pedon`	A `PedonRecord`.
`max_depth_cm`	Profundidade da secao de controle (default 200).
`ta_threshold`	Limite cmolc/kg argila para esmectitica (default 27).
`ki_caulinitico_min`	Limite Ki para caulinitica (default 0.75).
`kr_caulinitico_min`	Limite Kr para caulinitica vs oxidica (default 0.75).

Value

`FamilyAttribute`.

References

Embrapa (2018), SiBCS 5a ed., Cap 18, p 286-287.

`FamilyAttribute` *Classe S4-like para atributos de Familia (5o nivel SiBCS)*

Description

Classe S4-like para atributos de Familia (5o nivel SiBCS)

Classe S4-like para atributos de Familia (5o nivel SiBCS)

Details

Estrutura categorica (em vez de booleana) que representa um adjetivo composto da Familia. `value` eh o adjetivo atribuido (string) ou `NULL` quando a dimensao nao se aplica ou nao foi possivel determinar.

Public fields

`name` Nome da dimensao (e.g. "grupamento_textural").

`value` Adjetivo atribuido (e.g. "argilosa") ou NULL.

`evidence` Lista nomeada com valores intermediarios.

`missing` Vetor de colunas necessarias mas indisponiveis.

`reference` String com referencia bibliografica.

Methods

Public methods:

- `FamilyAttribute$new()`
- `FamilyAttribute$print()`
- `FamilyAttribute$clone()`

Method `new()`**:** Build a FamilyAttribute.

Usage:

```
FamilyAttribute$new(
  name,
  value = NULL,
  evidence = list(),
  missing = character(0),
  reference = ""
)
```

Arguments:

`name` Nome da dimensao (e.g. "grupamento_textural").

`value` Adjetivo atribuido (e.g. "argilosa") ou `NULL`.

`evidence` Lista nomeada com valores intermediarios.

`missing` Vetor de colunas necessarias mas indisponiveis.

`reference` String com referencia bibliografica.

Method `print()`**:** Pretty-print the attribute.

Usage:

```
FamilyAttribute$print(...)
```

Arguments:

`...` Ignored (S3 print signature compatibility).

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
FamilyAttribute$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

`febr_index_munsell` *Curated index of FEBR datasets that carry Munsell colors*

Description

Returns a data.frame listing FEBR dataset IDs that have at least one Munsell-related column populated in their `camada` table, with metadata: `n_horizons`, `n_finite_munsell`, `coverage`, `column_pattern`.

Usage

```
febr_index_munsell(min_coverage = 0.1, refresh = FALSE, verbose = TRUE)
```

Arguments

`min_coverage`	Drop datasets whose Munsell coverage (fraction of horizons with non-NA hue) is below this. Default 0.1.
`refresh`	Logical. If `TRUE`, re-scan FEBR over the network instead of using the bundled May-2026 cache.
`verbose`	If `TRUE` (default), prints a one-line summary.

Details

Backed by a precomputed cache shipped in `R/sysdata.rda` (`.FEBR_MUNSELL_INDEX`; results of the May 2026 scan over 249 datasets). On first call after install, returns the cache instantly. Pass `refresh = TRUE` to re-scan FEBR live (slow, network-dependent; updates the in-memory copy but does not modify the bundled cache).

Value

A `data.frame` sorted by `n_finite_munsell` descending.

See Also

`read_febr_pedons`.

---

ferralic *Ferralic horizon (WRB 2022)*

Description

Tests whether any horizon meets the ferralic horizon criteria. The ferralic horizon is a subsurface horizon resulting from long and intense weathering, characterized by very low cation exchange capacity per unit clay – the canonical "low-activity clay" signal that defines the Ferralsol RSG.

Usage

```
ferralic(pedon, min_thickness = 30, max_cec = NULL, engine = NULL)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Minimum thickness in cm (default 30).
`max_cec`	Maximum CEC (1M NH4OAc, pH 7) per kg clay (default `NULL` = 16 in soilkey engine, 20 in aqp engine; see `engine`).
`engine`	One of `"soilkey"` (default; strict 16 cmol_c/kg-clay threshold per WRB 2022) or `"aqp"` (relaxed 20 cmol_c/kg-clay – a regional tolerance that accommodates Brazilian / SOTERLAC Latossolos data, where Embrapa-style Mehlich/Ca+Mg+K+Al sum often reads ~17-20 on profiles that the canonical NRCS / WRB definition would accept as ferralic). `NULL` reads `getOption("soilKey.diagnostic_engine")`. The numeric threshold can also be overridden directly via `options(soilKey.ferralic_max_cec = ...)`.

Details

Sub-tests called:

- `test_ferralic_texture` – texture sandy loam or finer.
- `test_cec_per_clay` – CEC / clay <= 16 (or 20 under `engine = "aqp"`) cmol_c/kg clay.
- `test_ferralic_thickness` – thickness >= 30 cm.

v0.3.1 alignment with WRB 2022 Ch 3.1.10 (p. 44): the older "ECEC <= 12 cmol_c/kg clay" gate was removed because it is not in the canonical text – only CEC (1M NH4OAc, pH 7) <= 16 is required.

v0.9.67 regional tolerance: BDsolos RJ benchmark (n=722 perfis) showed 88/115 Latossolos failing the strict 16-cmol gate because Embrapa lab methodology often reads CEC at 17-20 on profiles that are unambiguously Latossolos by every other criterion. The `engine = "aqp"` threshold of 20 closes that gap without redefining the WRB threshold itself; users targeting strict WRB 2022 fidelity should keep `engine = "soilkey"`.

The weatherable-mineral test (<= 10% by volume), water-dispersible-clay test, and stratification / rock-structure exclusions remain deferred (they need mineralogical data outside the canonical horizon schema) and are refinements rather than gates.

Value

A `DiagnosticResult`.

References

IUSS Working Group WRB (2022). *World Reference Base for Soil Resources*, 4th edition. International Union of Soil Sciences, Vienna. Chapter 3.1.10 – Ferralic horizon (p. 44).

`ferralsol` *Ferralsol RSG gate (WRB 2022 Ch 4, p 110)*

Description

WRB-canonical: ferralic horizon \<= 150 cm AND no argic horizon starting above (or at the upper limit of) the ferralic, UNLESS the argic in its upper 30 cm or throughout has one or more of:

- < 10% water-dispersible clay; OR
- DeltapH (pH_KCl - pH_water) \>= 0; OR
- \>= 1.4% soil organic carbon.

v0.3.4 enforces all three exception paths. The DeltapH check uses `ph_kcl` and `ph_h2o`; the WDC check uses `water_dispersible_clay_pct` (introduced in v0.3.3 schema).

Usage

```
ferralsol(pedon, strict = NULL)
```

Arguments

`pedon`	A `PedonRecord`.
`strict`	Logical or `NULL`. When `NULL` (default) it resolves via `getOption("soilKey.rsg_strict", FALSE)`. `TRUE` requires two of the three argic exception paths.

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

Tier-3 strict mode (v0.9.98)

When an argic horizon sits above the ferralic, the default gate keeps the profile as a Ferralsol if *any one* of the three exception paths (WDC \< 10%, DeltapH \>= 0, SOC \>= 1.4%) holds. With `strict = TRUE` the gate requires *at least two* of the three – a single weak indicator no longer rescues a profile with a translocated-clay argic from being keyed out of Ferralsols.

`ferric` *Ferric horizon (WRB 2022)*

Description

A horizon of iron accumulation that does not reach the cementation / redness levels of plinthic. Diagnostic for the Ferric qualifier.

Usage

```
ferric(pedon, min_thickness = 15, min_fe_dith_pct = 5)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Minimum thickness (cm; default 15).
`min_fe_dith_pct`	Minimum dithionite-extractable iron percent (default 5).

Value

A `DiagnosticResult`.

References

IUSS Working Group WRB (2022), Chapter 3.1, Ferric horizon.

`fibrico` *Material organico fibrico (SiBCS Cap 14)*

Description

Material organico pouco decomposto: >= 40% de fibras esfregadas OU indice de von Post H1-H4. Discrimina Organossolos Fibricos no 3o nivel.

Usage

```
fibrico(pedon)
```

Arguments

`pedon`	A `PedonRecord`.

Value

`DiagnosticResult`.

References

Embrapa (2018), SiBCS 5a ed., Cap 14 (Organossolos), pp 224-226.

`fill_from_spectra` *Fill missing soil attributes from spectra via OSSL*

Description

Given a `PedonRecord` carrying a `spectra$vnir` matrix (rows = horizons, columns = wavelengths in nm), pre-processes the spectra, predicts the requested soil properties using the chosen OSSL-backed method, and writes the predictions into the pedon's horizons table via `pedon$add_measurement(..., source = "predicted_spectra")`. Each call updates the pedon's provenance log so that downstream classification can derive an evidence grade.

Usage

```
fill_from_spectra(
  pedon,
  library = "ossl",
  region = c("global", "south_america", "north_america", "europe", "africa"),
 properties = c("clay_pct", "sand_pct", "silt_pct", "cec_cmol", "bs_pct", "ph_h2o",
    "oc_pct", "fe_dcb_pct", "caco3_pct"),
  method = c("mbl", "plsr_local", "pretrained"),
  preprocess = "snv+sg1",
  k_neighbors = 100L,
  overwrite = FALSE,
  ossl_library = NULL,
  ossl_models = NULL,
  verbose = TRUE
)
```

Arguments

`pedon`	A `PedonRecord` with a `spectra$vnir` matrix.
`library`	Currently only `"ossl"` is supported.
`region`	One of `"global"`, `"south_america"`, `"north_america"`, `"europe"`, `"africa"`. Used to subset the OSSL training data when supported by the underlying backend.
`properties`	Character vector of OSSL-supported property names to predict. Default covers the most-requested WRB/SiBCS-relevant attributes.
`method`	One of `"mbl"`, `"plsr_local"`, `"pretrained"`.
`preprocess`	Pre-processing pipeline; passed to `preprocess_spectra`.
`k_neighbors`	Number of neighbours for memory-based methods.
`overwrite`	If `FALSE` (default), only fill cells whose existing provenance is weaker than `predicted_spectra`.

`ossl_library`	Optional OSSL library object (see `predict_ossl_mbl`).
`ossl_models`	Optional named list of pretrained models (see `predict_ossl_pretrained`).
`verbose`	If `TRUE`, prints a cli summary.

Details

By default, predicted values do **not** overwrite measured values (the `add_measurement()` authority logic protects them). Setting `overwrite = TRUE` forces overwrite of any non-measured value.

Value

The mutated pedon, invisibly. Provenance entries with `source = "predicted_spectra"` are added per (horizon, property).

See Also

`preprocess_spectra`, `predict_ossl_mbl`, `predict_ossl_plsr_local`, `predict_ossl_pretrained`, `pi_to_confidence`.

`fill_munsell_from_spectra` *Fill missing Munsell colors on a PedonRecord from Vis-NIR spectra*

Description

High-level helper that runs `predict_munsell_from_spectra` per horizon over the Vis-NIR spectra in `pedon$spectra$vnir` and writes the resulting hue / value / chroma back to the matching horizon rows via `pedon$add_measurement(..., source = "predicted_spectra")`.

Usage

```
fill_munsell_from_spectra(pedon, overwrite = FALSE, verbose = TRUE)
```

Arguments

`pedon`	A `PedonRecord` that has `$spectra$vnir` populated (rows = horizons, cols = wavelengths).
`overwrite`	If `TRUE`, overwrite existing Munsell measurements. Default `FALSE` (only fills horizons whose Munsell is currently NA).
`verbose`	If `TRUE` (default), prints a per-horizon summary.

Details

This is the operational answer to the v0.9.35 Argissolo color confusion: when surveyor Munsell colors are missing and the user has Vis-NIR (e.g. from OSSL), call this helper, then re-run `classify_sibcs` – the v0.9.45 "color-undetermined" fallback will lift, and the classification will descend to subordem / grande grupo / subgrupo with proper `evidence_grade`.

Value

The pedon, invisibly. Provenance entries with `source = "predicted_spectra"` are appended.

`fluvic_material` *Fluvic material (WRB 2022)*

Description

Tests whether the profile shows fluvic material features: alternating textures across consecutive horizons within the upper 100 cm AND an irregular (non-monotone) organic carbon pattern with depth. Diagnostic of Fluvisols.

Usage

```
fluvic_material(pedon, max_top_cm = 100, min_clay_swing = 8)
```

Arguments

`pedon`	A `PedonRecord`.
`max_top_cm`	Maximum top depth (cm) considered (default 100).
`min_clay_swing`	Minimum absolute clay-percent change between consecutive layers required to count as alternation (default 8 percentage points).

Details

Sub-test: `test_fluvic_stratification`.

v0.3 limitations: WRB 2022 fluvic material also requires age (typically <100 years for sediment freshness), which v0.3 does not check (no temporal fields in the schema). The stratification proxy is conservative – truly heterogeneous floodplain profiles with dramatic texture swings will pass; subtle alluvial sequences may miss. v0.4 will refine.

Value

A `DiagnosticResult`.

References

IUSS Working Group WRB (2022), Chapter 3, Fluvic material.

`format_wrb_name` *Format a WRB 2022 soil name with qualifiers*

Description

Format a WRB 2022 soil name with qualifiers

Usage

```
format_wrb_name(
  rsg_name,
  principal = character(0),
  supplementary = character(0)
)
```

Arguments

`rsg_name`	Full RSG name (e.g. "Ferralsols").
`principal`	Character vector of principal-qualifier names.
`supplementary`	Character vector of supplementary-qualifier names (default empty in v0.9).

Value

Formatted string per Ch 6 p 154 ("Rhodic Ferralsol (Clayic, Humic, Dystric)").

`fragic` *Fragic horizon (WRB 2022): a high-bulk-density horizon with restricted rooting. v0.3.3: detects via bulk_density_g_cm3 >= 1.65 AND structure grade massive/very firm OR designation pattern* `x`*/*`Bx`*.*

Description

Fragic horizon (WRB 2022): a high-bulk-density horizon with restricted rooting. v0.3.3: detects via bulk_density_g_cm3 >= 1.65 AND structure grade massive/very firm OR designation pattern `x`/`Bx`.

Usage

```
fragic(pedon, min_thickness = 15, min_bd = 1.65)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Numeric threshold or option (see Details).
`min_bd`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`fragipa` *Fragipa (SiBCS Cap 2, p 73-74; v0.7)*

Description

Reuso de `fragic` (WRB v0.3.3): horizonte subsuperficial endurecido quando seco, baixa MO, BD elevada, quebradicidade.

Usage

```
fragipa(pedon, ...)
```

Arguments

`pedon`	A `PedonRecord`.
`...`	Reserved for future arguments.

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`from_aqp` *Convert an aqp SoilProfileCollection back to a list of PedonRecord*

Description

Inverse of `as_aqp`. Walks each profile in the SPC, renames aqp's canonical horizon column names back to soilKey's (`top -> top_cm`, `name -> designation`, `clay -> clay_pct`, ...), assembles a `PedonRecord` per profile, and returns the list.

Usage

```
from_aqp(spc)
```

Arguments

`spc`	A `aqp::SoilProfileCollection`.

Details

Round-trip property: `from_aqp(as_aqp(pedon))` reproduces pedon modulo column ordering.

Value

A list of `PedonRecord` objects (length = `length(spc)`).

See Also

`as_aqp`, the forward conversion.

Examples

```
## Not run:
pedons <- list(make_ferralsol_canonical(), make_luvisol_canonical())
spc <- as_aqp(pedons)
pedons2 <- from_aqp(spc)
identical(pedons[[1]]$horizons$clay_pct, pedons2[[1]]$horizons$clay_pct)
#> [1] TRUE

## End(Not run)
```

`gapfill_by_predicted_taxon`

Fill missing horizon attributes from the predicted taxon's mean profile

Description

Classifies pedon with NO fill to get a provisional taxon, then fills its missing cells from `taxon_profiles[[<that taxon>]]` (built by `build_taxon_profiles`). Non-circular: the fill is keyed on the model's own prediction, not the reference. Each fill is written with source = `"inferred_prior"` (grade C). Reachable via `gapfill = list(method = "taxon", taxon_profiles = <...>)`.

Usage

```
gapfill_by_predicted_taxon(
  pedon,
  taxon_profiles,
  system = c("sibcs", "wrb2022", "usda"),
  attrs = NULL,
  confidence = 0.55
)
```

Arguments

`pedon`	A `PedonRecord`.
`taxon_profiles`	Output of `build_taxon_profiles`.
`system`	One of `"sibcs"` (default), `"wrb2022"`, `"usda"`.
`attrs`	Attributes to fill (default: those present in the matched profile).
`confidence`	Provenance confidence (default 0.55, below a coordinate prior).

Value

Invisibly, the mutated pedon; attribute `"gapfill_by_predicted_taxon"` records the taxon + cells filled.

See Also

`build_taxon_profiles`, `apply_soilgrids_depth_prior`

`gapfill_derive_horizon` *Fill horizon attributes derivable BY DEFINITION from the same horizon*

Description

Recovers cells that are exact closures of other measured columns in the same horizon (not statistical estimates): the texture third (clay/silt/sand) when the other two are present and sum to \< 100; effective CEC as `sum(bases) + al`; aluminium saturation as `100 * al / ecec`; and base saturation as `100 * sum(bases) / cec`. Every fill is written with `source = "inferred_prior"` so the `PedonRecord` authority order keeps it from displacing a measured value and the evidence grade drops to `"C"`. Companion to `gapfill_within_pedon` (depth interpolation) and `apply_soilgrids_depth_prior` (external prior); reachable via the `gapfill = list(method = "derive")` argument of the classifiers.

Usage

```
gapfill_derive_horizon(pedon, overwrite = FALSE)
```

Arguments

`pedon`	A `PedonRecord`.
`overwrite`	If `FALSE` (default) only `NA` target cells are filled.

Value

Invisibly, the mutated pedon; attribute `"gapfill_derive_horizon"` records the count filled.

See Also

`gapfill_within_pedon`, `apply_soilgrids_depth_prior`

`gapfill_within_pedon` *Fill interior missing horizon attributes by within-pedon depth interpolation*

Description

For each requested attribute, builds a depth profile from the horizons in which that attribute is *measured* (non-`NA`) and linearly interpolates the value at the mid-depth of every horizon where it is missing – but only for horizons whose mid-depth falls strictly between the shallowest and deepest measured layer. Cells above the top or below the bottom measured layer are left `NA`: the function interpolates, it never extrapolates. Each fill is written with `source = "inferred_prior"`, so the `PedonRecord` authority order keeps it from displacing a measured, spectra-predicted or VLM-extracted value, and any downstream `compute_evidence_grade` call reports grade `"C"`.

Usage

```
gapfill_within_pedon(pedon, attrs = NULL, confidence = 0.6, overwrite = FALSE)
```

Arguments

`pedon`	A `PedonRecord` with at least two horizons.
`attrs`	Character vector of horizon columns to fill. Defaults to the continuous depth-trending attributes a linear interpolation can reasonably estimate (clay/silt/sand, pH, organic carbon, CEC/ECEC, base/aluminium saturation, bulk density).
`confidence`	Numeric in \[0, 1\] recorded as the provenance confidence of each interpolated cell. Defaults to `0.6` – below a measured value but anchored on the profile's own data, hence above the `0.5` used for an external SoilGrids prior.
`overwrite`	If `FALSE` (default) only `NA` cells are filled. If `TRUE`, non-measured cells are re-interpolated (measured cells are still never overwritten, and the provenance authority order is always respected).

Details

This is the within-pedon companion to `apply_soilgrids_depth_prior` (which fills from an external SoilGrids profile rather than from the profile's own measured layers). It is the mechanism behind the opt-in `gapfill` argument of `classify_wrb2022`, `classify_sibcs`, `classify_usda` and `classify_all`.

Note that this mutates `pedon` in place (as `apply_soilgrids_depth_prior` does). The `gapfill` argument of the classifiers operates on a deep copy instead, so a classification call never alters the caller's pedon.

Value

Invisibly, the mutated pedon. An attribute `"gapfill_within_pedon"` on the return value records how many cells were filled and for which attributes.

See Also

apply_soilgrids_depth_prior, classify_all

Examples

```
h <- data.frame(
  top_cm    = c(0, 20, 40, 60),
  bottom_cm = c(20, 40, 60, 90),
  clay_pct  = c(15, NA, 35, 40)
)
p <- PedonRecord$new(horizons = h)
gapfill_within_pedon(p, attrs = "clay_pct")
p$horizons$clay_pct   # second horizon filled to ~25 by interpolation
```

`get_perturbation_scale` *Monte-Carlo perturbation scale for an evidence grade*

Description

Returns the noise magnitudes used by `classify_with_uncertainty` for a cell of the given evidence grade. A measurement (grade A) is perturbed only slightly; a user-assumed value (grade E) is perturbed heavily, reflecting how little is actually known about it.

Usage

```
get_perturbation_scale(grade = c("A", "B", "C", "D", "E"))
```

Arguments

`grade`	One of `"A"` (measured), `"B"` (spectra-predicted), `"C"` (prior-inferred), `"D"` (VLM-extracted) or `"E"` (user-assumed).

Value

A list with three elements: `pct` (the half-width of the multiplicative perturbation, applied to most numeric attributes), `ph_abs` (the half-width of the additive perturbation applied to pH columns) and `munsell_abs` (the additive half-width for Munsell value / chroma columns).

Examples

```
get_perturbation_scale("A")$pct   # 0.03 -- measured values barely move
get_perturbation_scale("E")$pct   # 0.30 -- assumptions move a lot
```

`gleyic_properties` *Gleyic properties (WRB 2022)*

Description

Tests whether the profile shows gleyic properties – evidence of prolonged saturation by groundwater – within the upper 50 cm. Gleyic properties are diagnostic for Gleysols and qualify many other RSGs (Endogleyic, Epigleyic qualifiers).

Usage

```
gleyic_properties(
  pedon,
  max_top_cm = 50,
  min_redox_pct = 5,
  stagnic_decay_factor = 3
)
```

Arguments

`pedon`	A `PedonRecord`.
`max_top_cm`	Maximum top depth (cm) of a candidate layer (default 50, per WRB 2022).
`min_redox_pct`	Minimum `redoximorphic_features_pct` (default 5).
`stagnic_decay_factor`	Numeric threshold or option (see Details).

Details

Sub-test: `test_gleyic_features` – requires explicit `redoximorphic_features_pct` >= 5% within the upper 50 cm.

v0.2 deliberately does NOT use the Munsell-based shortcut (chroma <= 2 + value >= 4) as a primary criterion: that pattern fits albic / bleached horizons of Podzols just as well as truly reduced gleyic horizons. v0.3 will add reductimorphic / oxidimorphic feature discrimination once we model field-described mottle properties. v0.9.72 adds the designation-suffix path (opt-in).

Value

A `DiagnosticResult`.

v0.9.72 designation morphological inference (opt-in)

Field-described Brazilian Gleissolos profiles (e.g.\ the Embrapa Redape curated dataset) routinely encode gleyic properties via the designation suffix `g` (e.g.\ `Cg`, `Cg1`, `Cgn`, `Apg`) plus low-chroma Munsell colours (chroma \<= 2), without recording `redoximorphic_features_pct` as a numeric percent. The strict canonical test then returns `NA` on every horizon and Gleissolos cascade to other Orders.

With `options(soilKey.gleyic_designation_inference = TRUE)` the function accepts a layer as gleyic when:

1. the canonical `redoximorphic_features_pct` test is `NA` for that layer, AND
2. the designation matches `[A-Z]+g[0-9a-z]?` (a horizon name with a g suffix in the master letter sequence, e.g.\ `Cg`, `Bg2`, `Apg`, `Cgn`), AND
3. the layer has `munsell_chroma_moist <= 2` (low-chroma reduced colour) when Munsell is recorded; if Munsell is missing on the layer the suffix alone is sufficient (designation suffix is the most direct signal of pedologist field judgment).

This is conservative: the suffix g is a master-letter modifier in the FAO/Embrapa horizon nomenclature that explicitly means "gleyic-affected" – the curator already made the call. Default is `FALSE` (canonical behaviour preserved).

References

IUSS Working Group WRB (2022), Chapter 3, Gleyic properties.

`gleysol` *Gleysol RSG gate (WRB 2022 Ch 4, p 103)*

Description

WRB-canonical (multi-path):

1. Layer \>= 25 cm starting \<= 40 cm with gleyic properties throughout AND reducing conditions in some parts of every sublayer; OR
2. Mollic/umbric > 40 cm thick with reducing conditions some parts of every sublayer 40 cm below mineral surface to lower limit, AND directly underneath a layer \>= 10 cm with lower limit \>= 65 cm having gleyic properties + reducing conditions; OR
3. Permanent saturation by water \<= 40 cm.

v0.3.4 enforces path 1 (the dominant path) and path 3 via designation (W / saturated marker). Path 2 is deferred (requires a depth-of- saturation column that's not standard).

Usage

```
gleysol(pedon, strict = NULL)
```

Arguments

pedon	A `PedonRecord`.
strict	Logical or `NULL`. When `NULL` (default) it resolves via `getOption("soilKey.rsg_strict", FALSE)`. `TRUE` tightens path 1 and disables path 3.

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

Tier-3 strict mode (v0.9.98)

With `strict = TRUE` the path-1 gleyic+reducing layer must start within the upper 25 cm (instead of 40 cm), and the path-3 designation-only fallback (a “W” / aquic marker) is disabled: strict mode requires measured gleyic and reducing evidence.

`grapes-or-or-grapes` *Default-value-for-NULL operator*

Description

Returns the left-hand side if it is non-NULL, otherwise the right-hand side. Re-exported so that downstream code can use the same idiom soilKey itself uses internally.

Usage

```
a %||% b
```

Arguments

`a`	The candidate value.
`b`	The fallback used when `a` is NULL.

Value

Either `a` or `b`.

`GSM_DEPTHS` *Default GlobalSoilMap depth intervals (cm)*

Description

GSM standard per Arrouays et al. (2014) "GlobalSoilMap: Toward a fine-resolution global grid of soil properties". Boundaries: 0-5, 5-15, 15-30, 30-60, 60-100, 100-200 cm.

Usage

```
GSM_DEPTHS
```

Format

An object of class `numeric` of length 7.

`gypsic` *Gypsic horizon (WRB 2022)*

Description

Tests whether any horizon meets the gypsic horizon criteria. The gypsic horizon is a horizon of secondary gypsum accumulation, diagnostic for Gypsisols.

Usage

```
gypsic(pedon, min_thickness = 15, min_gypsum_pct = 5)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Minimum thickness in cm (default 15).
`min_gypsum_pct`	Minimum gypsum percent in fine earth (default 5).

Details

Sub-tests called:

- `test_caso4_concentration` – gypsum >= 5%.
- `test_minimum_thickness` – thickness >= 15 cm.

v0.2 limitations: the WRB rule that gypsum content must exceed the underlying horizon by 1% (absolute) is not enforced. Petrogypsic (cemented) horizons are not yet detected. Both deferred to v0.3.

Value

A `DiagnosticResult`.

References

IUSS Working Group WRB (2022). *World Reference Base for Soil Resources*, 4th edition. International Union of Soil Sciences, Vienna. Chapter 3 – Gypsic horizon.

`gypsiric_material`	*Gypsiric material (WRB 2022 Ch 3.3.7): \>= 5% gypsum that is primary (not secondary). Without a "secondary fraction" schema column, v0.3.3 treats any layer with caso4_pct >= 5 as gypsiric unless it explicitly carries gypsic-horizon designation.*

Description

Gypsiric material (WRB 2022 Ch 3.3.7): \>= 5% gypsum that is primary (not secondary). Without a "secondary fraction" schema column, v0.3.3 treats any layer with caso4_pct >= 5 as gypsiric unless it explicitly carries gypsic-horizon designation.

Usage

```
gypsiric_material(pedon, min_caso4_pct = 5)
```

Arguments

`pedon`	A `PedonRecord`.
`min_caso4_pct`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`harmonize_to_gsm`	*Harmonise pedons to GlobalSoilMap depth intervals*

Description

Runs `mpspline2::mpspline_tidy()` on each requested numeric horizon attribute, producing a new PedonRecord per input pedon whose horizons table covers the canonical GSM intervals (`GSM_DEPTHS`). Categorical attributes (designation, Munsell hue) are propagated by mode-over-depth-overlap.

Usage

```
harmonize_to_gsm(
  pedons,
 attributes = c("clay_pct", "silt_pct", "sand_pct", "ph_h2o", "oc_pct", "cec_cmol",
    "base_saturation_pct", "munsell_value_moist", "munsell_chroma_moist",
    "redoximorphic_features_pct"),
  depths = GSM_DEPTHS,
  lam = 0.1,
  verbose = TRUE
)
```

Arguments

`pedons`	A list of `PedonRecord` objects.
`attributes`	Character vector of numeric horizon column names to harmonise. Default covers the chemistry / texture / Munsell numeric columns the soilKey diagnostics use.
`depths`	Numeric vector of GSM depth boundaries (n+1 values for n intervals). Default `GSM_DEPTHS`.
`lam`	Smoothing parameter for the spline (default 0.1, per Bishop et al. 1999 recommendation).
`verbose`	If `TRUE` (default), emits cli progress.

Value

A list of new `PedonRecord` objects with harmonised horizons.

Why mass-preserving

The Bishop et al. (1999) spline conserves the integral of the attribute over depth: if the original pedon has 30 g/kg OC over 0-15 cm, the harmonised pedon will report 30 g/kg integrated over 0-15 cm (split between 0-5 and 5-15 in proportion to the spline-implied gradient). This is a critical property for benchmark integrity: simple linear interpolation does not preserve mass and biases means upward / downward systematically.

Categorical handling

`designation` and `munsell_hue_moist` (and other character columns in the horizon schema) cannot be splined. Instead, for each target GSM interval, we pick the modal value weighted by the depth-overlap fraction with the input horizons. Ties broken by uppermost-input-horizon precedence.

References

Bishop, T.F.A., McBratney, A.B., Laslett, G.M. (1999). "Modelling soil attribute depth functions with equal-area quadratic smoothing splines." *Geoderma* 91: 27-45.

Arrouays, D. et al. (2014). "GlobalSoilMap: Toward a fine-resolution global grid of soil properties." *Advances in Agronomy* 125: 93-134.

See Also

`mpspline2::mpspline_tidy`, `GSM_DEPTHS`.

`hemico` *Material organico hemico (SiBCS Cap 14)*

Description

Material organico em decomposicao intermediaria: 17-40% de fibras esfregadas OU indice de von Post H5-H6. Discrimina Organossolos Hemicos no 3o nivel.

Usage

```
hemico(pedon)
```

Arguments

`pedon`	A `PedonRecord`.

Value

`DiagnosticResult`.

References

Embrapa (2018), SiBCS 5a ed., Cap 14 (Organossolos), pp 224-226.

`histic_horizon` *Histic horizon (WRB 2022)*

Description

A surface (or near-surface, after drainage) horizon of organic material; diagnostic of Histosols. Two alternative qualifying paths per WRB 2022:

- **Contiguous**: a single layer of organic material (OC % >= `min_oc`) reaching the surface and at least `min_thickness` cm thick (default 10 cm).
- **Cumulative**: organic material totalling `cumulative_min_cm` cm (default 40) within the upper `cumulative_max_depth_cm` (default 80). Relevant for folic / mossy Histosols on slopes.

Either path qualifies. The "after drainage" qualifier (recently drained organic soils) is treated as implicit since the same OC and thickness criteria apply.

Usage

```
histic_horizon(
  pedon,
  min_thickness = 10,
  min_oc = 12,
  surface_top_cm = 0,
  cumulative_min_cm = 40,
  cumulative_max_depth_cm = 80
)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Minimum thickness (cm) for the contiguous path (default 10).
`min_oc`	Minimum organic carbon % (default 12, WRB 2022; equivalent to `>= 20%` organic matter).
`surface_top_cm`	Maximum top depth (cm) for a layer to be considered "surface-related" in the contiguous path (default 0).
`cumulative_min_cm`	Minimum cumulative thickness (cm) for the cumulative path (default 40).
`cumulative_max_depth_cm`	Depth window (cm) for the cumulative path (default 80).

Value

A `DiagnosticResult`.

References

IUSS Working Group WRB (2022), Chapter 3, Histic horizon and organic material.

`horizon_column_spec` *Canonical horizon column specification*

Description

Returns the schema for the `horizons` `data.table` carried by a `PedonRecord`: an ordered named list mapping column names to their R type (`"numeric"` or `"character"`). Adding a new attribute means editing this single function.

Usage

```
horizon_column_spec()
```

Value

Named list of column types in canonical order.

Examples

```
spec <- horizon_column_spec()
head(names(spec))
```

`hortic` *Hortic horizon (WRB 2022): garden / kitchen-midden topsoil. Diagnostic criteria: thickness \>= 20 cm, dark colour (mollic-like), high P (Mehlich-3 P >= 100 mg/kg or P2O5_1pct_citric >= 175 mg/kg), high SOC.*

Description

Hortic horizon (WRB 2022): garden / kitchen-midden topsoil. Diagnostic criteria: thickness \>= 20 cm, dark colour (mollic-like), high P (Mehlich-3 P >= 100 mg/kg or P2O5_1pct_citric >= 175 mg/kg), high SOC.

Usage

```
hortic(pedon, min_thickness = 20, min_oc = 1, min_p_mehlich3 = 100)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Numeric threshold or option (see Details).
`min_oc`	Numeric threshold or option (see Details).
`min_p_mehlich3`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`hydragric` *Hydragric horizon (WRB 2022): subsoil hydric horizon under anthraquic. v0.3.3 detects via designation pattern* `Bg|Brg` *immediately below an anthraquic-like topsoil.*

Description

Hydragric horizon (WRB 2022): subsoil hydric horizon under anthraquic. v0.3.3 detects via designation pattern `Bg|Brg` immediately below an anthraquic-like topsoil.

Usage

```
hydragric(pedon, min_thickness = 20)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`hypersulfidic_material` *Hypersulfidic material (WRB 2022 Ch 3.3.8): \>= 0.01% inorganic sulfidic S, pH \>= 4, capable of severe acidification on aerobic incubation.*

Description

Hypersulfidic material (WRB 2022 Ch 3.3.8): \>= 0.01% inorganic sulfidic S, pH \>= 4, capable of severe acidification on aerobic incubation.

Usage

```
hypersulfidic_material(pedon, min_s_pct = 0.01, min_pH = 4)
```

Arguments

`pedon`	A `PedonRecord`.
`min_s_pct`	Numeric threshold or option (see Details).
`min_pH`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`hyposulfidic_material` *Hyposulfidic material (WRB 2022 Ch 3.3.9): same inorganic sulfidic S and field pH as hypersulfidic but does NOT consist of hypersulfidic (criterion 3 – does not acidify to pH < 4 on aerobic incubation, usually self-neutralised by carbonate). Reachable from v0.9.128: when* `incubation_ph` *is measured, a sulfidic + pH>=4 layer that stays >= 4 on incubation is the set-complement of* `hypersulfidic_material` *and is reported here. Without an incubation pH the two cannot be told apart, so this returns empty (the layer is reported as potential hypersulfidic instead).*

Description

Hyposulfidic material (WRB 2022 Ch 3.3.9): same inorganic sulfidic S and field pH as hypersulfidic but does NOT consist of hypersulfidic (criterion 3 – does not acidify to pH < 4 on aerobic incubation, usually self-neutralised by carbonate). Reachable from v0.9.128: when `incubation_ph` is measured, a sulfidic + pH>=4 layer that stays >= 4 on incubation is the set-complement of `hypersulfidic_material` and is reported here. Without an incubation pH the two cannot be told apart, so this returns empty (the layer is reported as potential hypersulfidic instead).

Usage

```
hyposulfidic_material(pedon, min_s_pct = 0.01, min_pH = 4)
```

Arguments

`pedon`	A `PedonRecord`.
`min_s_pct`	Numeric threshold or option (see Details).
`min_pH`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`inspect_bdsolos_csv` *Diagnostic inspection of a BDsolos CSV before loading*

Description

Reads the CSV header, attempts to map each column to the soilKey horizon schema via `.bdsolos_match_column`, and prints three sections:

Usage

```
inspect_bdsolos_csv(path, sep = NULL)
```

Arguments

`path`	Path to the CSV downloaded from BDsolos.
`sep`	Field separator (default `","`; some BDsolos exports use `";"` or tab).

Details

- **Mapped columns** – BDsolos name -> soilKey name
- **Unmapped columns** – columns the loader will ignore (review these before running `load_bdsolos_csv` to make sure no critical attribute is silently dropped)
- **Munsell coverage** – whether matiz / valor / croma are present in either umido or seco variants

Run this before `load_bdsolos_csv` on any new CSV from BDsolos, especially if the export schema looks unfamiliar (BDsolos has shipped multiple schema versions over the years).

Value

Invisibly, a list with `mapped`, `unmapped`, `munsell_present`, `taxon_column`.

`irragric` *Irragric horizon (WRB 2022): topsoil thickened by irrigation deposits. v0.3.3: thickness >= 20 cm + sediment-derived structure proxied via designation* `Apk|Apg|Au`.

Description

Irragric horizon (WRB 2022): topsoil thickened by irrigation deposits. v0.3.3: thickness >= 20 cm + sediment-derived structure proxied via designation `Apk|Apg|Au`.

Usage

```
irragric(pedon, min_thickness = 20)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`kastanozem` *Kastanozem RSG diagnostic (WRB 2022)*

Description

Tests whether a profile satisfies the Kastanozem RSG criteria: a mollic horizon plus secondary carbonates plus NOT-Chernozem colour (chroma (moist) > 2 in the upper 20 cm).

Usage

```
kastanozem(pedon, max_chroma_upper = 2)
```

Arguments

`pedon`	A `PedonRecord`.
`max_chroma_upper`	Maximum moist chroma to qualify as Chernozem (default 2). Kastanozem requires the upper-20-cm chroma to EXCEED this value.

Value

A `DiagnosticResult`.

References

IUSS Working Group WRB (2022), Chapter 5, Kastanozems.

`kastanozem_strict` *Kastanozem RSG gate (strengthened, WRB 2022 Ch 4, p 112)*

Description

Same structure as `chernozem_strict` but using the mollic horizon (no chernic gate) and starting \<= 70 cm of mineral soil surface.

Usage

```
kastanozem_strict(pedon, min_bs = 50, max_top_cm = 70, strict = NULL)
```

Arguments

`pedon`	A `PedonRecord`.
`min_bs`	Numeric threshold or option (see Details).
`max_top_cm`	Numeric threshold or option (see Details).
`strict`	Logical or `NULL`. When `NULL` (default) it resolves via `getOption("soilKey.rsg_strict", FALSE)`. `TRUE` raises the base-saturation floor to 75%.

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

Tier-3 strict mode (v0.9.98)

With `strict = TRUE` the base-saturation floor above the carbonate-bearing layer is raised from 50% to 75%. The 70 cm carbonate-depth window is left unchanged.

`key_trace_table` *Flatten a classification key trace into a tabular form*

Description

Normalises the system-dependent decision trace carried by a `ClassificationResult` into a single, ordered data frame of display rows. WRB 2022 stores a flat list of reference-soil-group test steps; the hierarchical SiBCS and USDA keys store a nested list of phases (orders, suborders, great groups, subgroups, family, ...), each holding candidate steps, an assigned-taxon record, family attributes, or a bare label. This function walks all of those shapes and returns one row per step, in the order the key was evaluated, so every consumer (`print()`, the HTML and PDF reports, the Shiny app) can render the trace uniformly.

Usage

```
key_trace_table(x)
```

Arguments

`x`	A `ClassificationResult`, or the `trace` list taken from one.

Value

A `data.frame` with one row per trace step and columns:

`phase` Key phase / level the step belongs to (e.g. `"orders"`, `"subgrupos"`); empty for the flat WRB trace.

`code` Taxon or attribute code.

`name` Taxon or attribute name (or attribute value).

`status` One of `"passed"`, `"failed"`, `"indeterminate"` (a test that could not be evaluated for want of data), `"selected"` (the taxon assigned at a level), or `"info"` (a family attribute or label, not a pass/fail test).

`missing` Comma-separated attributes that were missing for the step (empty when none).

`n_missing` Integer count of missing attributes.

A zero-row data frame with those columns when the trace is empty.

Examples

```
res <- classify_sibcs(make_ferralsol_canonical())
head(key_trace_table(res))
```

`kst13_canonical` *Keys to Soil Taxonomy 13th edition canonical reference*

Description

Convenience wrapper for `canonical_reference("ST_criteria_13th")`. Returns a nested list of 3,153 parsed Keys-to-Soil-Taxonomy clauses per chapter / page / key / taxon / code / clause / logic.

Usage

```
kst13_canonical(prefer_pkg = TRUE)
```

Arguments

`prefer_pkg`	If `TRUE` (default), prefer the installed SoilTaxonomy package over the vendored copy. Set to `FALSE` to force the vendored copy (e.g. for reproducibility of a specific soilKey release).

Details

Source: NCSS-tech `SoilTaxonomy` R package. Original: USDA-NRCS (2022). *Keys to Soil Taxonomy*, 13th edition.

Value

The canonical *Keys to Soil Taxonomy* (13th ed.) criteria reference (a list / data.frame).

`kst13_codes` *Load the canonical KST 13ed code -> taxon-name lookup table*

Description

Returns the 3,153-row data.frame from `inst/rules/usda/canonical/2022_KST_codes.json`, vendored from NCSS-tech/SoilKnowledgeBase. Each row is a (code, name) pair.

Usage

```
kst13_codes()
```

Details

Code structure:

- Single letter (`"A"`-`"L"`): Soil Order (Gelisols, Histosols, ..., Entisols)
- Two letters (`"AB"`, `"AC"`, ...): Suborder
- Three letters: Great Group
- Four letters: Subgroup

Value

A data.frame with columns `code, name`.

See Also

`kst13_criteria`, `kst13_canonical`.

`kst13_criteria` *Load the canonical KST 13ed criteria for a single taxon code*

Description

Returns the parsed clause data.frame for one code (e.g. `"A"` for Gelisols, `"ABA"` for Histels.Folistels, etc.). Each row is one clause of the diagnostic text with `content`, `chapter`, `page` columns.

Usage

```
kst13_criteria(code)
```

Arguments

`code`	Character. Taxon code in the KST 13ed code system (e.g. `"A"` for Gelisols, `"ABCDA"` for the Lithic Folistels subgroup).

Details

For the full 3,153-element nested list (all codes), use `kst13_canonical` (which loads the SoilTaxonomy R-package RDA equivalent).

Value

A data.frame with the parsed clauses for that code, or `NULL` if the code is not present.

See Also

`kst13_codes`, `kst13_canonical`.

`leptic_features` *Leptic features (WRB 2022)*

Description

Tests whether continuous rock or rock-like material occurs within `max_depth` cm of the surface. Two alternative paths qualify per WRB 2022:

1. **Designation**: a layer at depth <= `max_depth` with designation matching `"^R"` or `"^Cr"` (continuous rock or weathered rock-like substrate).
2. **Coarse fragments**: a layer at depth <= `max_depth` with coarse_fragments_pct >= `min_coarse_pct` (default 90% by volume), interpreted as rock-dominated even when not R / Cr-designated.

Either path qualifies.

Usage

```
leptic_features(pedon, max_depth = 25, min_coarse_pct = NULL, engine = NULL)
```

Arguments

`pedon`	A `PedonRecord`.
`max_depth`	Maximum depth (cm) at which continuous rock or rock-dominated material must appear (default 25).
`min_coarse_pct`	Minimum coarse-fragment percent for the coarse-fragments path (default 90 in soilkey engine, 50 in aqp engine; `NULL` picks a default per engine).
`engine`	One of `"soilkey"` (default; strict 90\ cfvo threshold) or `"aqp"` (LUCAS-friendly relaxed 50\ requiring positive evidence of rock contact – v0.9.66 tightening). The thin-topsoil path fires only when a horizon ending within `max_depth` also satisfies *at least one* of: (a) designation contains "R" (e.g.\ AR, BR, Cr, R, Rk), (b) `coarse_fragments_pct` >= `30` (gravelly), or (c) a deeper horizon is R/Cr-designated. Users with a strong external prior (e.g.\ a parent-material survey that documents rock < 25 cm but did not record it in the horizon table) can opt back into the original v0.9.65 loose behaviour with `options(soilKey.leptic_assume_rock_below = TRUE)`. `NULL` (the default) reads `getOption("soilKey.diagnostic_engine")`.

Value

A `DiagnosticResult`.

References

IUSS Working Group WRB (2022), Chapter 5, Leptosols.

`limnic_material`	*Limnic material (WRB 2022 Ch 3.3.10): subaquatic deposits (coprogenous earth, diatomaceous earth, marl, gyttja). v0.3.3: detects via* `rock_origin %in% c("lacustrine", "marine")` *or designation pattern.*

Description

Limnic material (WRB 2022 Ch 3.3.10): subaquatic deposits (coprogenous earth, diatomaceous earth, marl, gyttja). v0.3.3: detects via `rock_origin %in% c("lacustrine", "marine")` or designation pattern.

Usage

```
limnic_material(pedon)
```

Arguments

`pedon`	A `PedonRecord`.

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`limonic`	*Limonic horizon (WRB 2022 Ch 3.1)*

Description

From Greek *leimon* = meadow. A subaqueous / wet-meadow horizon showing accumulation of secondary Fe/Mn (oxi)hydroxides from fluctuating redox cycles. Distinct from *limnic material* (Ch 3.3.10), which is the parent material; the limonic horizon is the *soil* horizon derived from such material plus subsequent pedogenesis.

Usage

```
limonic(pedon, min_thickness = 5, min_redox_pct = 5)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Numeric threshold or option (see Details).
`min_redox_pct`	Numeric threshold or option (see Details).

Details

v0.3.5 detection: redoximorphic_features_pct \>= 5 AND designation pattern `Bm` / `Bjm` / `m` as proxy for past meadow wetness.

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`lithic_discontinuity` *Lithic discontinuity (WRB 2022 Ch 3.2.7)*

Description

Significant abrupt change in parent material between two layers. v0.3.3 simplified: detects via large discontinuity in coarse_fragments_pct (>= 10pp absolute jump) OR rock_origin difference between consecutive layers.

Usage

```
lithic_discontinuity(pedon)
```

Arguments

`pedon`	A `PedonRecord`.

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`lixisol` *Lixisol RSG diagnostic (WRB 2022)*

Description

argic + CEC < 24 cmol_c/kg clay + BS >= 50%.

Usage

```
lixisol(pedon, max_cec = 24, min_bs = 50)
```

Arguments

`pedon`	A `PedonRecord`.
`max_cec`	Maximum CEC per kg clay (default 24).
`min_bs`	Minimum base saturation % (default 50).

Value

A `DiagnosticResult`.

References

IUSS Working Group WRB (2022), Chapter 5, Lixisols.

`load_afsp_pedons` *Load Africa Soil Profiles (AfSP) v1.2 as PedonRecord objects*

Description

Reads the AfSP DBase tables shipped inside `AF-AfSP1.2.zip` (downloadable from `https://files.isric.org/public/af` and converts each profile + its horizons to a soilKey `PedonRecord`. Filters to profiles with a populated WRB 2006 RSG code (i.e.\ classifiable; AfSP has ~7000 of these of the total 18,533).

Usage

```
load_afsp_pedons(
  afsp_dir,
  max_n = NULL,
  countries = NULL,
  wrb_codes = NULL,
  verbose = TRUE
)
```

Arguments

`afsp_dir`	Directory containing the extracted AfSP DBase tables (`AfSP012Qry_Profiles.dbf`, `AfSP012Qry_Layers.dbf`).
`max_n`	Optional integer; take a random sample of this size from the classifiable profiles.
`countries`	Optional character vector of ISO country codes to keep (e.g.\ `c("MW", "ET", "TZ")`).
`wrb_codes`	Optional character vector of WRB 2006 RSG codes to keep (e.g.\ `c("VR", "FR", "AC")`).
`verbose`	Print progress.

Value

A list of `PedonRecord` objects.

References

Leenaars, J. G. B., van Oostrum, A. J. M., & Ruiperez Gonzalez, M. (2014). Africa Soil Profiles Database, Version 1.2. ISRIC Report 2014/01. ISRIC – World Soil Information, Wageningen. Project page: `https://isric.org/projects/africa-soil-profiles-database-afsp`.

`load_afsp_sample` *Load the bundled AfSP stratified sample (v0.9.77)*

Description

Returns a 130-profile snapshot from AfSP v1.2 stratified by WRB RSG (5 profiles per RSG x 26 RSGs), pre-built so users can run the African WRB benchmark offline without the 35 MB ZIP download.

Usage

```
load_afsp_sample()
```

Details

This is the African analogue of `load_wosis_stratified_sample` (global WoSIS) and `load_kssl_nasis_sample` (US KSSL+NASIS).

Value

A list with `pedons`, `pulled_on`, `source`, `filter`.

Reference

Leenaars, J. G. B., van Oostrum, A. J. M., & Ruiperez Gonzalez, M. (2014). Africa Soil Profiles Database, Version 1.2. ISRIC Report 2014/01.

`load_bdsolos_csv` *Load a BDsolos CSV export as a list of PedonRecord objects*

Description

Reads the long-format BDsolos CSV (one row per horizon, with a profile-id key) and returns a list of `PedonRecord` objects. Auto-detects the column-name convention via `inspect_bdsolos_csv` and maps to the soilKey horizon schema. Texture (argila / silte / areia) is converted from g/kg to percent (BDsolos canonical unit).

Usage

```
load_bdsolos_csv(path, sep = NULL, verbose = TRUE)
```

Arguments

`path`	Path to the BDsolos CSV.
`sep`	Field separator. Default `","`; BDsolos sometimes exports with `";"` or tab – pass it explicitly.
`verbose`	If `TRUE` (default), prints a one-line summary.

Details

Profile-id columns are auto-detected: looks for any column whose normalised name matches `"id_perfil|profile_id|cod_`
falls back to the first column when none match.

Value

A list of `PedonRecord` objects. Each pedon has `site$id` from the profile-id column, the taxonomic reference (when present) at `site$reference_sibcs`, and one horizon row per CSV row matching the profile id.

See Also

`inspect_bdsolos_csv`, `download_bdsolos`.

`load_embrapa_pedons` *Load Embrapa dadosolos pedons with reference SiBCS classification*

Description

Reads the Embrapa BDsolos CSV export (or the dadosolos R package data frame, if present). Assembles a list of `PedonRecord` objects with the SiBCS classification attached as `pedon$site$reference_sibcs`.

Usage

```
load_embrapa_pedons(csv_path, head = NULL, verbose = TRUE)
```

Arguments

`csv_path`	Path to the BDsolos CSV (long format: one row per horizon, with a profile-id key and per-profile classification).
`head`	Optional integer for parser validation.
`verbose`	If `TRUE` (default), emits a summary.

Details

The dadosolos / BDsolos archive ships with ~5k profiles in PT-BR with full SiBCS classification, lab data, and horizon morphology – the primary validation set for Brazilian-context use. Available from `https://www.bdsolos.cnptia.embrapa.br/`.

Value

A list of `PedonRecord` objects.

`load_febr_pedons` *Load the Embrapa FEBR superconjunto into a list of PedonRecords*

Description

Reads the FEBR `febr-superconjunto.txt` export (one row per camada / horizon, with the profile-level classification denormalised onto every row), groups rows by `(dataset_id, observacao_id)`, and returns a list of `PedonRecord` objects with all three reference taxa attached on `$site`: `reference_sibcs` (raw FEBR string, e.g. "LATOSSOLO VERMELHO"), `reference_usda`, `reference_wrb`.

Usage

```
load_febr_pedons(
  path,
  head = NULL,
  require_classification = c("sibcs", "any", "wrb", "usda"),
  verbose = TRUE
)
```

Arguments

`path`	Path to `febr-superconjunto.txt`.
`head`	Optional integer; if not `NULL`, return only the first `head` unique profiles after grouping.
`require_classification`	One of `c("any", "sibcs", "wrb","usda")`. Default `"sibcs"`: drop profiles whose chosen classification is NA. `"any"` keeps profiles with at least one of the three.
`verbose`	If `TRUE` (default), summarises the load.

Details

Drops profiles whose `taxon_sibcs` is NA (no usable reference). Drops camadas with no horizon-depth information (no `profund_sup` / `profund_inf`).

Value

A list of `PedonRecord` objects.

`load_kssl_nasis_sample`

Load the bundled KSSL + NASIS morphological-enriched sample (v0.9.75)

Description

Returns a 99-profile snapshot built by joining the NCSS Lab Data Mart (`ncss_labdata.gpkg`) with the companion NASIS Morphological sqlite (`NASIS_Morphological_*.sqlite`) via `load_kssl_pedons_with_nasis`. Pre-annotated with derived WRB Reference Soil Group via `usda_to_wrb_rsg`.

Usage

```
load_kssl_nasis_sample()
```

Details

Compared to `load_kssl_sample` (KSSL lab tables only), this sample carries the morphological evidence that several WRB diagnostic horizons need:

| Field | KSSL-only | KSSL + NASIS | |——-|————-:|—————-:| | munsell_hue_moist | 0 | munsell_value_moist | 0 | munsell_chroma_moist | 0 | munsell_hue_dry | 0 | structure_grade | 0 | structure_type | 0 | clay_films_amount | 0 | slickensides | 0

First-ever benchmark on this enriched sample (soilKey v0.9.75, full v0.9.69-72 fallback stack):

- Top-1 baseline: 19.1\ **+3.5pp lift purely from NASIS morphology**)
- Top-1 full stack: 20.6\
- Phaeozem: 1/33 -> 2/33 (Munsell-driven mollic detection)
- Podzol: 0/15 -> 1/15

Remaining ceiling driven by attributes neither dataset preserves: Solonetz needs Na/ESP, Vertisols need slickensides + cracks (NASIS records 1.7 on subsoil samples NASIS often lacks.

Value

A list with `pedons`, `pulled_on`, `source`, `join_helper`, `cross_walk`.

Reference

Beaudette, D., Skovlin, J., Roecker, S., Brown, A. (2024). aqp: Algorithms for Quantitative Pedology. R package version 2.x. `https://github.com/ncss-tech/aqp`.

Examples

```
## Not run:
s <- load_kssl_nasis_sample()
length(s$pedons)
#> 99
# Munsell now populated (KSSL-only sample had 0%):
mean(vapply(s$pedons,
            function(p) any(!is.na(p$horizons$munsell_hue_moist)),
            logical(1)))
#> 0.99

## End(Not run)
```

`load_kssl_pedons` *Load NCSS / KSSL pedons with reference USDA Soil Taxonomy classification*

Description

Reads the KSSL pedon CSV export (typically named `NCSS_Pedon_Layer.csv` or similar) plus the lab-data CSV, joins on `pedon_key`, and assembles a list of `PedonRecord` objects. The published USDA Soil Taxonomy classification (from the `Series` or `Subgroup` field) is attached as `pedon$site$reference_usda`.

Usage

```
load_kssl_pedons(pedon_csv, layer_csv, head = NULL, verbose = TRUE)
```

Arguments

`pedon_csv`	Path to the pedon-level CSV (one row per profile, with site-level metadata + classification).
`layer_csv`	Path to the layer-level CSV (one row per horizon, with horizon properties).
`head`	Optional integer; if not `NULL`, returns only the first head pedons (useful for parser validation).
`verbose`	If `TRUE` (default), emits a summary of the load.

Details

KSSL is the de-facto standard for validating USDA Soil Taxonomy keys (~50k profiles, lab-grade analytical data, professional pedon descriptions). Get the export from the USDA-NRCS NCSS Lab Data Mart (`ncsslabdatamart.sc.egov.usda.gov`).

Value

A list of `PedonRecord` objects.

`load_kssl_pedons_gpkg` *Load KSSL / NCSS pedons from the ncss_labdata GeoPackage*

Description

Reads the ‘lab_combine_nasis_ncss‘ / ‘lab_site‘ / ‘lab_layer‘ / ‘lab_chemical_properties‘ / ‘lab_physical_properties‘ views from the NCSS Lab Data Mart GeoPackage and assembles a list of `PedonRecord` objects. Each pedon has its USDA Soil Taxonomy Order attached as `site$reference_usda`, normalised to match ‘classify_usda()‘ output ("Mollisols", "Alfisols", ...).

Usage

```
load_kssl_pedons_gpkg(
  gpkg,
  head = NULL,
  require_b_horizon = TRUE,
  verbose = TRUE
)
```

Arguments

`gpkg`	Path to `ncss_labdata.gpkg`.
`head`	Optional integer; load only the first N classified pedons. Useful for parser validation.
`require_b_horizon`	If `TRUE` (default), drops pedons whose deepest horizon’s bottom_cm < 30. Most non-Entisol Order gates need a B horizon.
`verbose`	If `TRUE` (default), emits progress messages.

Value

A list of `PedonRecord` objects.

`load_kssl_pedons_with_nasis` *Load KSSL pedons enriched with NASIS morphology*

Description

Joins the NCSS Lab Data Mart GeoPackage with the NASIS Morphological SQLite to produce PedonRecord objects whose horizons table has BOTH lab chemistry + physics AND field morphology (Munsell, structure, clay films, slickensides, cracks). Required for the morphological-evidence diagnostics (`argic` clay-films, `vertic_horizon` slickensides, `mollic_epipedon_usda` Munsell, etc.) to fire on KSSL profiles – the lab gpkg alone has none of those.

Usage

```
load_kssl_pedons_with_nasis(
  gpkg,
  sqlite,
  head = NULL,
  require_b_horizon = TRUE,
  verbose = TRUE
)
```

Arguments

`gpkg`	Path to `ncss_labdata.gpkg`.
`sqlite`	Path to `NASIS_Morphological_*.sqlite`.
`head`	Optional integer; load only the first N classified pedons. Useful for parser validation / scaling.
`require_b_horizon`	If `TRUE` (default), drops pedons whose deepest horizon's bottom_cm < 30.
`verbose`	If `TRUE` (default), emits progress messages.

Value

A list of `PedonRecord` objects.

`load_kssl_sample` *Load the bundled KSSL/NCSS lab-data sample (v0.9.74)*

Description

Returns a 100-profile snapshot from the NCSS Lab Data Mart (KSSL gpkg, `head = 100`) pre-annotated with derived WRB Reference Soil Group via `usda_to_wrb_rsg`.

Usage

```
load_kssl_sample()
```

Details

This is the bundled offline counterpart to `load_kssl_pedons_gpkg` – use this for tests and demos when the 5.5 GB gpkg is not available locally.

Each pedon has BOTH:

- `site$reference_usda` (Order, Suborder, Greatgroup, Subgroup) – the canonical KSSL classification.
- `site$reference_wrb_from_usda` – the derived WRB RSG via the IUSS WRB 2022 Annex 6 cross-walk.

First-ever KSSL WRB benchmark (soilKey v0.9.74, full v0.9.69-72 fallback stack):

- Top-1 accuracy: 20.1\
- Calcisol 69\
- Phaeozem / Kastanozem / Solonetz 0\ data not in KSSL lab tables (in companion NASIS).

Value

A list with `pedons`, `pulled_on`, `source`, `cross_walk`.

Reference

Beaudette, D., Skovlin, J., Roecker, S., Brown, A. (2024). aqp: Algorithms for Quantitative Pedology. R package version 2.x. `https://github.com/ncss-tech/aqp`.

Examples

```
## Not run:
s <- load_kssl_sample()
length(s$pedons)
#> 100
table(vapply(s$pedons, function(p) p$site$reference_wrb_from_usda,
             character(1)))

## End(Not run)
```

`load_lucas_pedons` *Load EU-LUCAS / ESDB pedons with reference WRB classification*

Description

Reads the EU-LUCAS topsoil dataset joined with the ESDB profile archive (the v3 release produced by JRC). Assembles a list of `PedonRecord` objects with the WRB Reference Soil Group attached as `pedon$site$reference_wrb`.

Usage

```
load_lucas_pedons(lucas_csv, head = NULL, verbose = TRUE)
```

Arguments

`lucas_csv`	Path to the LUCAS topsoil CSV.
`head`	Optional integer for parser validation.
`verbose`	If `TRUE` (default), emits a summary.

Details

LUCAS is harvested every 3-6 years on a regular grid; the ESDB classification is updated synchronously. ~28k profile cells with WRB labels in the 2015-2018 release.

Value

A list of `PedonRecord` objects.

`load_lucas_soil_2018` *Load the LUCAS Soil 2018 Topsoil release as a list of PedonRecord objects*

Description

Reads the canonical European Soil Data Centre (ESDAC) release of LUCAS Soil 2018 Topsoil chemistry as published in the JRC report (ESDAC dataset `https://esdac.jrc.ec.europa.eu/content/lucas-2018-tops` The release ships ~18,984 European topsoil samples at 0-20 cm with pH (H2O and CaCl2), EC, OC, CaCO3, P, N, K and oxalate-extractable Al / Fe; a separate `BulkDensity_2018_final-2.csv` carries bulk density at 0-10 / 10-20 / 20-30 / 0-20 cm for ~6,272 of those points and is joined automatically when present.

Usage

```
load_lucas_soil_2018(
  path,
  attach_bulk_density = TRUE,
  countries = NULL,
  max_n = NULL,
  verbose = TRUE
)
```

Arguments

`path`	Folder containing `LUCAS-SOIL-2018.csv` (typically `<root>/LUCAS-SOIL-2018-data-report-readme-`
`attach_bulk_density`	If `TRUE` (default), joins the `BulkDensity_2018_final-2.csv` sister file on `POINTID` when present.
`countries`	Optional character vector of NUTS_0 codes (e.g. `c("ES", "FR")`) to filter pedons. Default `NULL` (all countries).
`max_n`	Optional integer cap on the number of pedons returned (after country filter). Useful for development.
`verbose`	If `TRUE` (default), prints a summary line.

Details

What's NOT in the release (and how to fill it):

- **Texture (clay / sand / silt)** – not in this CSV. Pass `benchmark_lucas_2018(..., fill_texture_from = "soilgrids")` to fill from ISRIC SoilGrids 250m via `lookup_soilgrids`.
- **Munsell colors** – not collected by LUCAS Soil 2018. If the user has Vis-NIR spectra (release separate ~83 GB), use `predict_munsell_from_spectra` (v0.9.47).
- **Vis-NIR spectra** – distributed separately by ESDAC. Once downloaded and attached to the pedon's `$spectra`, `predict_from_spectra` (v0.9.46) fills clay / sand / silt / pH / OC / CEC.
- **Taxonomic reference** – not in the LUCAS release; `benchmark_lucas_2018` attaches the canonical WRB Reference Soil Group via `lookup_esdb` (v0.9.44) at the pedon's coordinates.

Unit conversions applied (LUCAS -> soilKey schema):

- OC, N, CaCO3, Ox_Al, Ox_Fe: g/kg ->
- EC: mS/m -> dS/m (* 0.01)
- P, K: mg/kg unchanged
- pH: unitless

Special LUCAS string values `"< LOD"`, `"<LOD"`, empty cells and `"n.d."` / `"ND"` are converted to `NA` before numeric coercion.

Value

A list of `PedonRecord` objects (one per LUCAS point). Each pedon has a `site$id` matching the LUCAS `POINTID`, `site$lat` / `site$lon` in WGS84, and either one or two horizons (the second being 20-30 cm when the subsoil OC / CaCO3 columns are populated). Provenance entries from the loader use `source = "measured"`.

See Also

`benchmark_lucas_2018`, `lookup_esdb`, `lookup_soilgrids`.

Examples

```
## Not run:
path <- "soil_data/eu_lucas/LUCAS-SOIL-2018-data-report-readme-v2/LUCAS-SOIL-2018-v2"
pedons <- load_lucas_soil_2018(path, countries = c("ES", "PT"),
                                max_n = 100)
length(pedons)
pedons[[1]]

## End(Not run)
```

`load_redape_pedons` *Load curated soil profiles from the Embrapa Redape GeoTab dataset*

Description

Reads the structured JSON files (one profile per file) published by Vaz et al. 2023 at the Embrapa Redape repository (DOI `10.48432/PYKKA7`) and converts each one to a soilKey `PedonRecord`.

Usage

```
load_redape_pedons(json_dir, max_n = NULL, verbose = TRUE)
```

Arguments

`json_dir`	Directory containing the GeoTab JSON files (or a character vector of file paths).
`max_n`	If non-`NULL`, take a random sample of this size.
`verbose`	Print progress (default `TRUE`).

Details

The dataset is unique in two ways:

- Every profile was hand-reviewed by experienced pedologists (the curation note and author list are preserved on each pedon site record), so it is suitable as a gold-standard benchmark.
- Unlike BDsolos, all profiles ship the full exchange complex (Ca, Mg, K, Na, Al *and H*), so `cec_cmol` (Valor T = S + H + Al) is computed directly without any fallback option.

Value

A list of `PedonRecord` objects.

Reference

Vaz, G. J., Silva Jr, A. F., & Silva Neto, L. de F. da (2023). Brazilian soil data for taxonomic classification. Redape, V1. doi:10.48432/PYKKA7.

See Also

`download_redape_dataset`, `benchmark_redape`.

`load_rules` *Load a soilKey rule set (YAML)*

Description

Load a soilKey rule set (YAML)

Usage

```
load_rules(system = c("wrb2022", "usda", "sibcs5"), package = "soilKey")
```

Arguments

`system`	One of `"wrb2022"` (full WRB 2022 key, v0.2 wires 16/32 RSGs), `"usda"` (USDA Soil Taxonomy, v0.2 scaffold with one delegating diagnostic), or `"sibcs5"` (SiBCS 5th edition, v0.2 scaffold with one delegating diagnostic).
`package`	Package owning the rule files (default `"soilKey"`).

Value

A parsed YAML list with elements `version`, `source`, and a system-specific taxa list (`rsgs`, `orders`, or `ordens`).

`load_wosis_sample` *Load the bundled WoSIS South-America sample*

Description

Returns a 40-profile snapshot of WoSIS GraphQL data pulled on 2026-05-03 with `continent = "South America"`. The data is a frozen artefact – do NOT use it for current paper-grade benchmarks (the WoSIS database is updated periodically; the bundled snapshot is for reproducible tests and offline development only).

Usage

```
load_wosis_sample()
```

Details

For up-to-date benchmarks, call `run_wosis_benchmark_graphql()` directly against the live ISRIC GraphQL endpoint.

Value

A list as described above.

Returned data

A list with elements:

- `profiles_raw` – the parsed GraphQL response (one element per profile; nested layer arrays).
- `pedons` – `PedonRecord` objects ready for classification (one per profile).
- `pulled_on` – `Date` of the snapshot.
- `endpoint`, `filter`, `n_pulled` – metadata.

Examples

```
## Not run:
sample <- load_wosis_sample()
length(sample$pedons)
#> 40
classify_wrb2022(sample$pedons[[1]])$rsg_or_order

## End(Not run)
```

`load_wosis_stratified_sample`

Load the bundled WoSIS stratified RSG-balanced sample (v0.9.73)

Description

Returns a 130-profile snapshot of WoSIS GraphQL data pulled on 2026-05-09 with **stratified sampling by WRB Reference Soil Group**: 5 profiles per RSG across 26 RSGs (Acrisol, Andosol, Arenosol, Calcisol, Cambisol, Chernozem, Cryosol, Ferralsol, Fluvisol, Gleysol, Gypsisol, Histosol, Kastanozem, Leptosol, Luvisol, Nitisol, Phaeozem, Planosol, Plinthosol, Podzol, Regosol, Solonchak, Solonetz, Stagnosol, Umbrisol, Vertisol).

Usage

```
load_wosis_stratified_sample()
```

Details

This is the recommended cache for global WRB benchmarking. Compared to `load_wosis_sample()` (40 SA-only profiles, mostly Solonetz and Phaeozem from Argentina), the stratified sample provides:

- Even coverage across the 26 most important RSGs.
- Richer analytical attributes – CEC available on 26 ECEC on 37 in the SA snapshot).
- Geographic diversity (Angola, Brazil, USA, China, Russia, South Africa, Indonesia, Argentina, etc.).

First-ever benchmark on this sample (soilKey v0.9.73, full v0.9.69-72 fallback stack):

- Overall top-1 accuracy: 16.2\
- Histosol 100\ from 20\ Cambisol 60\
- 18 RSGs at 0\ expose (Munsell colours, base saturation, sodium for Solonetz, slickensides for Vertisols, etc.). Documented data ceiling.

For the live API, call `run_wosis_benchmark_graphql()` or the `read_wosis_profiles_graphql(wrb_rsg = "...", n_max = N)` helper (small RSG-filtered queries are tractable; large unfiltered pulls time out as of 2026-05).

Value

A list with:

- `pedons`: list of 130 `PedonRecord` objects.
- `meta`: named integer vector of profiles per RSG.
- `pulled_on`: pull date.
- `endpoint`: ISRIC GraphQL endpoint URL.
- `filter`: pull strategy metadata.
- `n_pulled`: 130.

Reference

Batjes, N. H., Ribeiro, E., van Oostrum, A. (2020). Standardised soil profile data to support global mapping and modelling (WoSIS snapshot 2019). *Earth System Science Data*, 12, 299-320. doi:10.5194/essd122992020.

Examples

```
## Not run:
s <- load_wosis_stratified_sample()
length(s$pedons)
#> 130
table(vapply(s$pedons, function(p) p$site$wosis_rsg, character(1)))
#> 5 of each: Acrisol, Andosol, ... Vertisol

## End(Not run)
```

`lookup_esdb` *Look up an ESDB raster value at WGS84 coordinates*

Description

Loads the requested attribute raster, reprojects WGS84 lat/lon input to the raster's native CRS (typically LAEA Europe, EPSG:3035), and extracts the value(s). When a Value Attribute Table (`.vat.dbf`) is available, the integer raster value is decoded to its coded string (e.g. '21' -> '"LV"' -> Luvisol).

Usage

```
lookup_esdb(coords, attribute, raster_root, decode = TRUE)
```

Arguments

`coords`	A two-column matrix or data.frame with ‘lon‘ and ‘lat‘ (WGS84 decimal degrees) – in that order. A single `c(lon, lat)` vector is also accepted.
`attribute`	Name of the ESDB attribute folder, e.g. `"WRBLV1"` or `"WRBFU"`. See `available_esdb_attributes`.
`raster_root`	Path to the unpacked ESDB raster directory.
`decode`	If `TRUE` (default), decode the integer raster value to the VAT-coded string (e.g. `"21"` -> `"LV"`). If `FALSE`, return the raw integer.

Details

Coordinates outside the European raster footprint return ‘NA‘ silently (rather than erroring) so vectorised calls degrade gracefully.

Value

Character vector (decoded codes) or numeric vector (raw values) of the same length as `nrow(coords)`. `NA` for points outside the raster footprint.

See Also

`available_esdb_attributes`

Examples

```
## Not run:
root <- "~/data/ESDB-Raster-Library-1k-GeoTIFF-20240507"

# Single point: Wageningen, Netherlands (5.66 E, 51.97 N)
lookup_esdb(c(5.66, 51.97), "WRBLV1", root)
#> [1] "GL"   # Gleysol per the ESDB 1km raster

# Vector: Lisbon + Berlin + Helsinki
coords <- rbind(c(-9.14, 38.72), c(13.40, 52.52), c(24.94, 60.17))
lookup_esdb(coords, "WRBLV1", root)
#> [1] "CM" "LV" "PZ"   # Cambisol, Luvisol, Podzol

## End(Not run)
```

`lookup_mapbiomas_solos` *Look up a MapBiomas Solos raster value at WGS84 coordinates*

Description

MapBiomas Solos (Project MapBiomas, Brazil) distributes a national raster of SiBCS classes at 30 m, downloadable from `https://mapbiomas.org/en/produtos`. This helper mirrors the shape of `lookup_esdb` but is local-file only: pass the path of the unpacked GeoTIFF and the function reprojects the user's WGS84 lat/lon to the raster's native CRS, extracts the pixel and (optionally) decodes the integer class code via a user-supplied legend.

Usage

```
lookup_mapbiomas_solos(coords, raster_path, legend = NULL)
```

Arguments

`coords`	A 2-column matrix or data.frame with `lon`, `lat` (WGS84 decimal degrees), or a length-2 numeric vector for a single query.
`raster_path`	Path to the unpacked MapBiomas Solos GeoTIFF.
`legend`	Optional two-column data.frame (first column = numeric value, second = SiBCS class name). When provided, the integer raster value is decoded; when `NULL`, the raw integer is returned.

Details

MapBiomas does not bundle a '.vat.dbf'; the canonical legend is published as a CSV / dictionary on their website. Pass it via `legend` as a two-column data.frame (`value, class_name`) to enable decoding.

Value

Character vector of decoded class names (when `legend` is supplied) or numeric vector of raster values. Same length as `nrow(coords)`. `NA` for points outside the raster footprint.

See Also

`lookup_esdb`, `lookup_soilgrids`.

Examples

```
## Not run:
tif <- "~/data/mapbiomas_solos_collection2_2023.tif"
legend <- data.frame(
  value = c(1L, 2L, 3L, 4L, 5L, 6L, 7L, 8L, 9L, 10L, 11L, 12L, 13L),
  class_name = c("Latossolo Vermelho-Amarelo",
                 "Latossolo Amarelo",
```

```
                      "Argissolo Vermelho-Amarelo",
                      "Argissolo Amarelo",
                      "Neossolo Quartzarenico",
                      "Cambissolo Haplico",
                      "Espodossolo",
                      "Gleissolo",
                      "Nitossolo",
                      "Planossolo",
                      "Plintossolo",
                      "Vertisolo",
                      "Outros")
)
lookup_mapbiomas_solos(c(-43.0, -22.0), tif, legend)

## End(Not run)
```

`lookup_soilgrids` *Look up a SoilGrids 250m soil property at WGS84 coordinates*

Description

Reads ISRIC SoilGrids 250m (Hengl et al. 2017, 2021) directly from the ISRIC Cloud-Optimized GeoTIFF (COG) endpoint at `https://files.isric.org/soilgrids/latest/data/` – no download required, only the pixel under each query coordinate is transferred over HTTPS.

Usage

```
lookup_soilgrids(
  coords,
  property = c("clay", "sand", "silt", "phh2o", "soc", "cec", "bdod", "nitrogen", "ocd",
    "ocs", "cfvo"),
  depth = c("0-5cm", "5-15cm", "15-30cm", "30-60cm", "60-100cm", "100-200cm"),
  quantile = c("mean", "Q0.05", "Q0.5", "Q0.95", "uncertainty"),
  baseurl = "https://files.isric.org/soilgrids/latest/data",
  raw = FALSE
)
```

Arguments

`coords`	A 2-column matrix or data.frame with `lon`, `lat` (WGS84 decimal degrees), or a length-2 numeric vector for a single query.
`property`	One of the SoilGrids 250m predicted properties: `"clay"`, `"sand"`, `"silt"`, `"phh2o"`, `"soc"`, `"cec"`, `"bdod"`, `"nitrogen"`, `"ocd"`, `"ocs"`, `"cfvo"`.
`depth`	Depth interval. One of `"0-5cm"`, `"5-15cm"`, `"15-30cm"`, `"30-60cm"`, `"60-100cm"`, `"100-200cm"`.
`quantile`	Output quantile. One of `"mean"` (default), `"Q0.05"`, `"Q0.5"`, `"Q0.95"`, `"uncertainty"`.
`baseurl`	Base URL of the SoilGrids COG endpoint. Default is the canonical ISRIC location; override only for a local mirror.

`raw`	If `TRUE`, returns the integer raster value without scaling. Default `FALSE` (returns the value in conventional units).

Details

SoilGrids stores integer rasters scaled per property; this helper applies the canonical conversion factor so the returned value is in conventional soil units (%, pH, g/kg, cmol(c)/kg, g/cm^3).

Value

Numeric vector of length `nrow(coords)`. `NA` outside the SoilGrids footprint or on network errors.

See Also

`lookup_esdb`, `lookup_mapbiomas_solos`.

Examples

```
## Not run:
# Single point
lookup_soilgrids(c(-43.0, -22.0),
                 property = "phh2o",
                 depth = "0-5cm",
                 quantile = "mean")

# Vector + multiple properties
coords <- rbind(c(-43.0, -22.0), c( -9.14, 38.72))
lookup_soilgrids(coords, "clay",  "0-5cm", "mean")
lookup_soilgrids(coords, "phh2o", "0-5cm", "mean")

## End(Not run)
```

`luvisol` *Luvisol RSG diagnostic (WRB 2022)*

Description

argic + CEC >= 24 cmol_c/kg clay + Al saturation < 50%.

Usage

```
luvisol(pedon, min_cec = 24, max_al_sat = 50)
```

Arguments

`pedon`	A `PedonRecord`.
`min_cec`	Minimum CEC per kg clay (default 24).
`max_al_sat`	Maximum Al saturation % (default 50).

Value

A `DiagnosticResult`.

References

IUSS Working Group WRB (2022), Chapter 5, Luvisols.

`make_acrisol_canonical` *Build the canonical Acrisol fixture*

Description

Synthetic tropical-humid Acrisol on weathered gneiss: argic horizon at Bt1 with low-activity clay (CEC/clay ~ 17 cmol_c/kg clay) and low base saturation (BS ~ 25%). By construction:

- `argic`: PASSES on Bt1.
- `acrisol`: PASSES (CEC low, BS low).
- `lixisol`, `alisol`, `luvisol`: FAIL.
- Other diagnostics: FAIL.

Usage

```
make_acrisol_canonical()
```

Value

A `PedonRecord`.

`make_alisol_canonical` *Build the canonical Alisol fixture*

Description

Synthetic humid-tropical Alisol on weathered shale: argic horizon at Bt1 with high-activity clay (CEC/clay ~ 34) AND high Al saturation (Al sat ~ 70%); the canonical "young weathering on a 2:1 clay parent that has not yet released enough Al into the precipitate-stabilised pool". By construction:

- `argic`: PASSES on Bt1.
- `alisol`: PASSES (CEC high, Al sat high).
- `acrisol`, `lixisol`, `luvisol`: FAIL.

Usage

```
make_alisol_canonical()
```

Value

A PedonRecord.

`make_andosol_canonical`

Build the canonical Andosol fixture

Description

Synthetic Andosol on volcanic tephra: very dark surface with low bulk density (0.7 g/cm^3) and high active Al + Fe (Al_ox + 0.5 * Fe_ox = 2.25%). By construction `andic_properties` passes.

Usage

```
make_andosol_canonical()
```

Value

A PedonRecord.

`make_anthrosol_canonical`

Build the canonical Anthrosol fixture

Description

Synthetic Anthrosol with a hortic horizon – a long-cultivated dark surface from sustained organic-matter additions (typical of centuries-old kitchen-garden / homegarden soils). By construction `anthric_horizons` passes via the designation pattern.

Usage

```
make_anthrosol_canonical()
```

Value

A PedonRecord.

`make_arenosol_canonical`

Build the canonical Arenosol fixture

Description

Synthetic coastal-dune Arenosol: sandy throughout the upper 100 cm (silt + 2*clay « 30). By construction `arenic_texture` passes uniformly while every clay-dependent diagnostic fails.

Usage

```
make_arenosol_canonical()
```

Value

A `PedonRecord`.

`make_argissolo_canonical`

Perfil canonico de Argissolo (SiBCS 5a ed., Cap 5)

Description

B textural com gradiente significativo, argila ativ baixa ou alta + V baixa. Catch-all final na chave – tipica do Brasil tropical.

Usage

```
make_argissolo_canonical()
```

Value

A `PedonRecord` populated with the canonical horizons and site metadata for this reference profile.

`make_calcisol_canonical`

Build the canonical Calcisol fixture

Description

Synthetic semi-arid Calcisol on calcareous loess: A horizon with modest secondary carbonate; a thick Bk1 with the diagnostic calcic horizon (35% CaCO3 over 40 cm); deepening accumulation in Bk2. By construction:

- `calcic`: PASSES on Bk1 and Bk2.
- `gypsic`, `salic`: FAIL.
- `argic`, `ferralic`, `mollic`: FAIL.

Usage

```
make_calcisol_canonical()
```

Value

A `PedonRecord`.

`make_cambisol_canonical`

Build the canonical Cambisol fixture

Description

Synthetic temperate-zone Cambisol on weathered colluvium: modest subsurface alteration in Bw without meeting argic clay-increase or ferralic CEC criteria. By construction:

- `cambic`: PASSES on Bw (thickness 35 cm, sandy clay loam, no argic / no ferralic).
- `argic`, `ferralic`, `mollic`, `calcic`, `gypsic`, `salic`: FAIL.

Usage

```
make_cambisol_canonical()
```

Value

A `PedonRecord`.

`make_cambissolo_canonical`

Perfil canonico de Cambissolo (SiBCS 5a ed., Cap 6)

Description

Reusa fixture WRB Cambisol – B incipiente sem ser plintico, vertico, planico, etc.

Usage

```
make_cambissolo_canonical()
```

Value

A `PedonRecord` populated with the canonical horizons and site metadata for this reference profile.

`make_chernossolo_canonical`

Perfil canonico de Chernossolo (SiBCS 5a ed., Cap 7)

Description

Reusa fixture WRB Chernozem – A chernozemico + Bk com argila Ta + V alta. SiBCS strictos exigem (a) Bi/Bt + Ta + V alta, OR (b) calcico/petrocalcico/carbonatico + A chernozemico.

Usage

```
make_chernossolo_canonical()
```

Value

A `PedonRecord` populated with the canonical horizons and site metadata for this reference profile.

`make_chernozem_canonical`

Build the canonical Chernozem fixture

Description

Synthetic Ukrainian / Russian steppe Chernozem on loess: thick dark Ah, granular structure, secondary carbonates accumulating in the Bk. By construction:

- `mollic`: PASSES on horizon Ah1 (moist value 2, chroma 1, dry value 3; SOC 4%; BS 89%; thickness 30 cm; strong granular structure).
- `argic`: FAILS (essentially no clay differentiation; ratios all close to 1).
- `ferralic`: FAILS (CEC/clay ~ 120 cmol_c/kg clay – high-activity 2:1 clay).

Usage

```
make_chernozem_canonical()
```

Value

A `PedonRecord`.

`make_cryosol_canonical`

Build the canonical Cryosol fixture

Description

Synthetic Arctic Cryosol on weathered shale with permafrost at 50 cm: thawed A horizon over a frozen Bf horizon. By construction `cryic_conditions` passes via the designation pattern.

Usage

```
make_cryosol_canonical()
```

Value

A `PedonRecord`.

`make_durisol_canonical` *Build the canonical Durisol fixture*

Description

Synthetic semi-arid Durisol with a Si-cemented subsurface horizon (35% duripan nodules over 45 cm). By construction `duric_horizon` passes on Bdu.

Usage

```
make_durisol_canonical()
```

Value

A `PedonRecord`.

`make_empty_horizons` *Build an empty horizons data.table with the canonical schema*

Description

Build an empty horizons data.table with the canonical schema

Usage

```
make_empty_horizons(n = 0L)
```

Arguments

`n`	Number of rows (default 0).

Value

A `data.table` with all canonical horizon columns filled with NAs of the correct type.

Examples

```
h <- make_empty_horizons(3)
nrow(h)
```

`make_espodossolo_canonical`

Perfil canonico de Espodossolo (SiBCS 5a ed., Cap 8)

Description

Reusa fixture WRB Podzol – B espodico imediatamente abaixo de E.

Usage

```
make_espodossolo_canonical()
```

Value

A `PedonRecord` populated with the canonical horizons and site metadata for this reference profile.

`make_ferralsol_canonical`

Build the canonical Ferralsol fixture

Description

Synthetic but realistic Brazilian Latossolo Vermelho (Ferralsol per WRB 2022): deeply weathered, kaolinitic / oxidic, with the canonical "low-activity clay" signature. Diagnostic outcomes are deterministic by construction:

- `ferralic`: PASSES on horizons Bw1 and Bw2 (CEC/clay = 8.3 cmol_c/kg clay; ECEC/clay = 3.6 cmol_c/kg clay; texture sandy clay / clay; thickness >= 30 cm).
- `argic`: FAILS (gradual clay increase, all pairwise ratios < 1.2; absolute increment too small for the >= 40% rule).
- `mollic`: FAILS (chroma > 3, BS < 50%, A horizon < 20 cm thick).

Usage

```
make_ferralsol_canonical()
```

Value

A `PedonRecord`.

`make_fluvisol_canonical`

Build the canonical Fluvisol fixture

Description

Synthetic floodplain Fluvisol: stratified textures across consecutive C horizons, OC pattern non-monotone with depth (because C2 is more recently deposited, OC-richer than C1). By construction `fluvic_material` passes.

Usage

```
make_fluvisol_canonical()
```

Value

A `PedonRecord`.

`make_gleissolo_canonical`

Perfil canonico de Gleissolo (SiBCS 5a ed., Cap 9)

Description

Reusa fixture WRB Gleysol – horizonte glei dentro de 50 cm.

Usage

```
make_gleissolo_canonical()
```

Value

A `PedonRecord` populated with the canonical horizons and site metadata for this reference profile.

`make_gleysol_canonical`

Build the canonical Gleysol fixture

Description

Synthetic Gleysol from a high-water-table floodplain: A with low chroma but no explicit redox features (so gleyic test is anchored on Bg); Bg with diagnostic redoximorphic features (35% by volume) within the upper 50 cm. By construction:

- `gleyic_properties`: PASSES on Bg.
- `argic`, `ferralic`, `mollic`, `cambic`, `plinthic`, `spodic`, `calcic`, `gypsic`, `salic`: FAIL.

Usage

```
make_gleysol_canonical()
```

Value

A `PedonRecord`.

`make_gypsisol_canonical`

Build the canonical Gypsisol fixture

Description

Synthetic Gypsisol on gypsiferous parent material: shallow A; gypsum accumulation rising sharply in the By1 horizon (35% gypsum over 50 cm) – the diagnostic gypsic horizon. By construction:

- `gypsic`: PASSES on By1 and By2.
- `calcic`, `salic`: FAIL.
- `argic`, `ferralic`, `mollic`: FAIL.

Usage

```
make_gypsisol_canonical()
```

Value

A `PedonRecord`.

`make_histosol_canonical` *Build the canonical Histosol fixture*

Description

Synthetic boreal-mire Histosol: thick (50 cm) surface organic horizon with OC ~ 35%, low chroma, no exchangeable-base data reported (typical of histic profiles where laboratory chemistry on organic material is reported separately). By construction:

- `histic_horizon`: PASSES on Oa.
- Mineral horizons below; mollic / umbric NA (no BS reported).

Usage

```
make_histosol_canonical()
```

Value

A `PedonRecord`.

`make_kastanozem_canonical` *Build the canonical Kastanozem fixture*

Description

Synthetic continental-semiarid Kastanozem on loess-like substrate: mollic surface (chroma 3, value 3) – dark enough for mollic but not dark enough for Chernozem (chroma 3 > 2 in the upper 20 cm); secondary carbonates accumulating in the Bk. By construction:

- `mollic`: PASSES.
- `kastanozem`: PASSES.
- `chernozem`, `phaeozem`: FAIL.

Usage

```
make_kastanozem_canonical()
```

Value

A `PedonRecord`.

`make_latossolo_canonical`
Perfil canonico de Latossolo (SiBCS 5a ed., Cap 10)

Description

Reusa fixture WRB Ferralsol – B latossolico imediatamente abaixo de A, sem horizonte argilico acima.

Usage

```
make_latossolo_canonical()
```

Value

A `PedonRecord` populated with the canonical horizons and site metadata for this reference profile.

`make_leptosol_canonical`
Build the canonical Leptosol fixture

Description

Synthetic mountain-slope Leptosol on metamorphic rock: a thin A (10 cm) directly over continuous rock. By construction:

- `leptic_features`: PASSES (R at 10 cm <= 25).
- Other diagnostics fail on thickness, missing data, or absent diagnostic features.

Usage

```
make_leptosol_canonical()
```

Value

A `PedonRecord`.

`make_lixisol_canonical` *Build the canonical Lixisol fixture*

Description

Synthetic Mediterranean / sub-tropical Lixisol on weathered calcareous parent material: argic horizon at Bt1 with low-activity clay (CEC/clay ~ 20) but high base saturation (BS ~ 70%) thanks to carbonate-buffered weathering. By construction:

- `argic`: PASSES on Bt1.
- `lixisol`: PASSES (CEC low, BS high).
- `acrisol`, `alisol`, `luvisol`: FAIL.

Usage

```
make_lixisol_canonical()
```

Value

A `PedonRecord`.

`make_luvisol_canonical` *Build the canonical Luvisol fixture*

Description

Synthetic temperate-zone Luvisol on loess: clear textural differentiation, Bt with clay coatings, high base saturation, high- activity clay. By construction:

- `argic`: PASSES on horizon Bt1 (clay increase from E (18%) to Bt1 (35%) gives ratio 1.94 in the 15-40% band; thickness 25 cm; texture clay loam; no glossic features).
- `ferralic`: FAILS (CEC/clay ~ 45 cmol_c/kg clay in the Bt – well above the 16 cmol_c/kg threshold).
- `mollic`: FAILS (A horizon: moist value 4 > 3, thickness 10 cm < 20 cm).

Usage

```
make_luvisol_canonical()
```

Value

A `PedonRecord`.

`make_luvissolo_canonical` *Perfil canonico de Luvissolo (SiBCS 5a ed., Cap 11)*

Description

Solo com B textural argila Ta + V alta. Tipico do semiarido com rocha basica.

Usage

```
make_luvissolo_canonical()
```

Value

A `PedonRecord` populated with the canonical horizons and site metadata for this reference profile.

`make_neossolo_canonical` *Perfil canonico de Neossolo Litolico (SiBCS 5a ed., Cap 12)*

Description

Solo raso sobre rocha continua dura. Sem horizonte B diagnostico.

Usage

```
make_neossolo_canonical()
```

Value

A `PedonRecord` populated with the canonical horizons and site metadata for this reference profile.

`make_nitisol_canonical`

Build the canonical Nitisol fixture

Description

Synthetic East-African Nitisol on weathered basalt: clay-rich (>= 50%), Fe-rich (DCB ~ 6%), polyhedral structure with shiny ped surfaces. By construction `nitic_horizon` passes.

Usage

```
make_nitisol_canonical()
```

Value

A `PedonRecord`.

`make_nitossolo_canonical`

Perfil canonico de Nitossolo Vermelho (SiBCS 5a ed., Cap 13)

Description

Solo argiloso (>= 35% argila desde superficie) com B nitico (estrutura forte em blocos + cerosidade), gradiente textural baixo (B/A <= 1.5).

Usage

```
make_nitossolo_canonical()
```

Value

A `PedonRecord` populated with the canonical horizons and site metadata for this reference profile.

`make_organossolo_canonical`

Perfil canonico de Organossolo (SiBCS 5a ed., Cap 14)

Description

Solo organico saturado, com horizonte H histico >= 60 cm e SOC alto. Tipico de varzea / brejo.

Usage

```
make_organossolo_canonical()
```

Value

A `PedonRecord` populated with the canonical horizons and site metadata for this reference profile.

`make_phaeozem_canonical`

Build the canonical Phaeozem fixture

Description

Synthetic humid-temperate Phaeozem on non-calcareous loess: mollic (chroma 2, value 2-3) and high BS, but no secondary carbonates anywhere – typical of more leached / less-arid steppe-forest transition. By construction:

- `mollic`: PASSES.
- `phaeozem`: PASSES.
- `chernozem`, `kastanozem`: FAIL (no carbonates).

Usage

```
make_phaeozem_canonical()
```

Value

A `PedonRecord`.

`make_planosol_canonical` *Build the canonical Planosol fixture*

Description

Synthetic temperate Planosol with abrupt textural change: sandy E (clay 12%) overlies a clay-rich Bt (35%) at 25 cm with an abrupt boundary. By construction `planic_features` passes.

Usage

```
make_planosol_canonical()
```

Value

A `PedonRecord`.

`make_planossolo_canonical` *Perfil canonico de Planossolo (SiBCS 5a ed., Cap 15)*

Description

Solo com horizonte E sobrejacente a B planico (mudanca textural abrupta + cores neutras + cromas baixos).

Usage

```
make_planossolo_canonical()
```

Value

A `PedonRecord` populated with the canonical horizons and site metadata for this reference profile.

`make_plinthosol_canonical`

Build the canonical Plinthosol fixture

Description

Synthetic seasonally-saturated tropical Plinthosol: A horizon with typical Cerrado SOC; Btv with diagnostic plinthite (25% by volume over 60 cm); persistent plinthite at depth. By construction:

- `plinthic`: PASSES on Btv and Cv.
- `argic`, `ferralic`, `mollic`, `spodic`, `calcic`, `gypsic`, `salic`: FAIL.

Usage

```
make_plinthosol_canonical()
```

Value

A `PedonRecord`.

`make_plintossolo_canonical`

Perfil canonico de Plintossolo (SiBCS 5a ed., Cap 16)

Description

Reusa fixture WRB Plinthosol – horizonte plintico iniciando dentro de 40 cm.

Usage

```
make_plintossolo_canonical()
```

Value

A `PedonRecord` populated with the canonical horizons and site metadata for this reference profile.

`make_podzol_canonical` *Build the canonical Podzol fixture*

Description

Synthetic boreal / temperate-coniferous Podzol: bleached E (low clay, low CEC), illuvial Bs with diagnostic Al/Fe oxalate accumulation, weathered C. By construction:

- `spodic`: PASSES on Bs (Al_ox + 0.5*Fe_ox = 0.6, pH 4.5, 40 cm thick).
- `argic`, `ferralic`, `mollic`, `cambic`, `plinthic`, `calcic`, `gypsic`, `salic`: FAIL.

Usage

```
make_podzol_canonical()
```

Details

E horizon Munsell is set to chroma 3 (rather than canonical 1-2 of a true albic) to keep `gleyic_properties` clearly negative under the conservative v0.2 criterion.

Value

A `PedonRecord`.

`make_retisol_canonical`

Build the canonical Retisol fixture

Description

Synthetic temperate Retisol on loess over clay-rich substrate: bleached E with glossic tongues penetrating the underlying argic Bt. By construction `retic_properties` passes via the "glossic" designation pattern; `argic` also passes (this is correct – Retisols are argic + retic features, and the WRB key tests RT before AC/LX/AL/LV).

Usage

```
make_retisol_canonical()
```

Value

A `PedonRecord`.

`make_solonchak_canonical`

Build the canonical Solonchak fixture

Description

Synthetic Solonchak from a coastal-arid setting: surface salt accumulation gives the diagnostic salic horizon (EC 25 dS/m over 25 cm); EC declines but stays elevated in the Bz; non-saline C below. By construction:

- `salic`: PASSES on Az.
- `gypsic`, `calcic`: FAIL.
- `argic`, `ferralic`, `mollic`: FAIL.

Usage

```
make_solonchak_canonical()
```

Value

A `PedonRecord`.

`make_solonetz_canonical`

Build the canonical Solonetz fixture

Description

Synthetic Solonetz on saline-sodic substrate: argic Btn with columnar structure and high exchangeable Na (ESP ~ 28%). By construction `natric_horizon` passes.

Usage

```
make_solonetz_canonical()
```

Value

A `PedonRecord`.

`make_stagnosol_canonical`

Build the canonical Stagnosol fixture

Description

Synthetic Stagnosol: redoximorphic features in a perched layer (Bg, 15-50 cm; redox 25%) but the deeper subsoil is well-drained (BC redox 2%, C redox 0). The decay-with-depth contrast is what distinguishes stagnic from gleyic. By construction `stagnic_properties` passes and `gleyic_properties` also passes (the surface redox qualifies for both); the WRB key tests Stagnosols (#16) and Gleysols (#9), so a real Stagnosol-typed fixture lands at Gleysols if both pass – the criteria differ in depth pattern, which is enough for the diagnostic functions but not for key precedence in v0.3. This is documented in the test as known overlap; v0.4 will add a stronger discriminator.

Usage

```
make_stagnosol_canonical()
```

Value

A `PedonRecord`.

`make_synthetic_pedon_with_spectra`

Build a synthetic PedonRecord with attached spectra (testing aid)

Description

Generates a small, deterministic `PedonRecord` with `n_horizons` horizons and a Vis-NIR spectral matrix (`350:2500` nm). Useful for exercising `fill_from_spectra` in tests and vignettes.

Usage

```
make_synthetic_pedon_with_spectra(
  n_horizons = 5L,
  wavelengths = 350:2500,
  seed = 1L
)
```

Arguments

`n_horizons`	Integer number of horizons (default 5).
`wavelengths`	Integer vector of wavelengths (default `350:2500`).
`seed`	Integer seed for the RNG used to generate the spectra.

Value

A `PedonRecord` with a `$spectra$vnir` matrix attached.

`make_technosol_canonical`

Build the canonical Technosol fixture

Description

Synthetic urban / industrial Technosol: surface horizon with 30% anthropogenic artefacts (brick, glass, slag, plastic). By construction `technic_features` passes.

Usage

```
make_technosol_canonical()
```

Value

A `PedonRecord`.

`make_umbrisol_canonical`

Build the canonical Umbrisol fixture

Description

Synthetic humid-temperate Umbrisol on weathered acidic schist: deep organic-rich dark surface with low base saturation – the acid analogue of a Phaeozem. By construction `umbric_horizon` passes; `mollic` fails on BS < 50.

Usage

```
make_umbrisol_canonical()
```

Value

A `PedonRecord`.

`make_vertisol_canonical` *Build the canonical Vertisol fixture*

Description

Synthetic Vertisol from a smectite-rich plain: deep clay (50-55%) with strong slickensides in the Bss horizon. Surface chroma 4 (above the mollic cap) so that vertic_properties is the only v0.2 diagnostic that passes. By construction:

- `vertic_properties`: PASSES on Bss and BC.
- `argic`, `ferralic`, `mollic`, `cambic`, `plinthic`, `spodic`, `calcic`, `gypsic`, `salic`: FAIL.

Usage

```
make_vertisol_canonical()
```

Value

A `PedonRecord`.

`make_vertissolo_canonical` *Perfil canonico de Vertissolo (SiBCS 5a ed., Cap 17)*

Description

Solo argiloso (>= 30% argila desde superficie) com horizonte vertico (slickensides + fendas + clay alto) iniciando dentro de 100 cm. Reusa structure / fixture do WRB Vertisol.

Usage

```
make_vertissolo_canonical()
```

Value

A `PedonRecord` populated with the canonical horizons and site metadata for this reference profile.

`mineral_material` *Mineral material (WRB 2022 Ch 3.3.11): < 20% SOC AND < 35% volume artefacts containing >= 20% organic carbon. The complement of organic_material / organotechnic_material.*

Description

Mineral material (WRB 2022 Ch 3.3.11): < 20% SOC AND < 35% volume artefacts containing >= 20% organic carbon. The complement of organic_material / organotechnic_material.

Usage

```
mineral_material(pedon, max_oc = 20, max_organotechnic = 35)
```

Arguments

`pedon`	A `PedonRecord`.
`max_oc`	Numeric threshold or option (see Details).
`max_organotechnic`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`mollic` *Mollic horizon (WRB 2022)*

Description

Tests whether any near-surface horizon meets the mollic horizon criteria. The mollic horizon is the diagnostic surface horizon of Chernozems, Phaeozems, Kastanozems, and several other RSGs; it indicates a thick, dark, base-rich, organic-matter-enriched topsoil formed under steppe or comparable vegetation.

Usage

```
mollic(
  pedon,
  min_thickness = 20,
  min_oc = 0.6,
  min_bs = 50,
  surface_top_cm = 5
)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Minimum thickness in cm (default 20).
`min_oc`	Minimum SOC % (default 0.6).
`min_bs`	Minimum base saturation % (default 50).
`surface_top_cm`	Maximum top depth (cm) for a horizon to be considered "surface-related" (default 5). v0.1 uses this as a proxy for the WRB rule that mollic must form continuously from the soil surface (after mixing of upper 20 cm if required).

Details

Sub-tests called:

- `test_mollic_color` – moist value <= 3, moist chroma <= 3, dry value <= 5.
- `test_mollic_organic_carbon` – SOC >= 0.6%.
- `test_mollic_base_saturation` – BS (NH4OAc, pH 7) >= 50%.
- `test_mollic_thickness` – horizon thickness >= 20 cm.
- `test_mollic_structure` – not simultaneously massive AND very hard when dry.

v0.1 limitations: cumulative thickness across contiguous mollic- qualifying horizons is not yet supported – this matters for profiles where mollic criteria are met by an A1+A2 sequence but no single horizon is >= 20 cm thick. Mixing of upper 20 cm before the test (per WRB) is also deferred to v0.2.

Value

A `DiagnosticResult`.

References

IUSS Working Group WRB (2022). *World Reference Base for Soil Resources*, 4th edition. International Union of Soil Sciences, Vienna. Chapter 3 – Mollic horizon.

`mudanca_textural_abrupta`

Mudanca textural abrupta (SiBCS Cap 1, p 30-31)

Description

Aumento consideravel de argila em pequena distancia vertical (\<= 7.5 cm) na transicao A/E -> B:

- argila A < 200 g/kg: argila B \>= 2x A; OR
- argila A 200-400 g/kg: incremento absoluto \>= 200 g/kg (i.e. de 300 -> 500); OR
- argila A \>= 400 g/kg: incremento absoluto \>= 220 g/kg (i.e. de 420 -> 640).

Reuso de `abrupt_textural_difference` (WRB Ch 3.2.1) que ja codifica criterios essencialmente equivalentes.

Usage

```
mudanca_textural_abrupta(pedon)
```

Arguments

`pedon`	A `PedonRecord`.

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`mulmic_material` *Mulmic material (WRB 2022 Ch 3.3.12): mineral material developed from organic material; \>= 8% SOC, with low BD, structural / chroma criteria.*

Description

Mulmic material (WRB 2022 Ch 3.3.12): mineral material developed from organic material; \>= 8% SOC, with low BD, structural / chroma criteria.

Usage

```
mulmic_material(pedon, min_oc = 8, max_chroma = 2)
```

Arguments

`pedon`	A `PedonRecord`.
`min_oc`	Numeric threshold or option (see Details).
`max_chroma`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`natric_horizon` *Natric horizon (WRB 2022)*

Description

Tests for the natric horizon: an argic horizon with diagnostic sodium accumulation (ESP >= 15%) within at least one argic layer. Diagnostic of Solonetz.

Usage

```
natric_horizon(pedon, min_esp = 15, min_pH_h2o = 7)
```

Arguments

`pedon`	A `PedonRecord`.
`min_esp`	Minimum ESP % (default 15).
`min_pH_h2o`	Minimum pH(H2O) for the ESP-only path (default 7.0; alkaline gate to exclude false-positive acidic Bt horizons).

Value

A `DiagnosticResult`.

v0.9.76 designation + ESP-only inference (opt-in)

Field-described Solonetz profiles in NCSS / KSSL data routinely reach the natric ESP threshold (computed from `na_cmol` / `cec_cmol`) without satisfying the strict `argic()` clay-increase test, because surveyors record `Btk`-suffix designations (carbonates dominate the horizon designation choice) rather than `Btn`/`Bn` or `clay_pct` is missing.

With `options(soilKey.natric_designation_inference = TRUE)` the function accepts a layer as natric when the canonical argic test returns NA or FALSE AND *either*:

1. the designation matches `[A-Z][a-z0-9]*n` (an `n` master-letter modifier in the horizon name – e.g.\ `Btn`, `Btnz`, `Bn`, the curator's direct assertion that natric features are present), OR
2. ESP >= `min_esp` on a B-prefixed subsoil layer (`top_cm > 20`) AND the layer's pH(H2O) >= 7 (alkaline – typical of true natric, excludes acidic Bt horizons that happen to read high Na from sea-spray).

Default is `FALSE` (canonical behaviour preserved).

References

IUSS Working Group WRB (2022), Chapter 3, Natric horizon.

`nitic_horizon` *Nitic horizon (WRB 2022)*

Description

Tests for the nitic horizon: a clay-rich (>= 30%), Fe-rich (DCB Fe >= 4%) subsurface horizon at least 30 cm thick. Diagnostic of Nitisols. WRB 2022 additionally requires polyhedral / nutty structure with shiny ped surfaces and a gradual (non-abrupt) clay decrease with depth.

Usage

```
nitic_horizon(
  pedon,
  min_clay = 30,
  min_fe_dcb = 4,
  min_thickness = 30,
  max_clay_drop_pct = 8,
  max_decrease_depth = 50
)
```

Arguments

`pedon`	A `PedonRecord`.
`min_clay`	Minimum clay % (default 30).
`min_fe_dcb`	Minimum DCB-extractable Fe % (default 4).
`min_thickness`	Minimum thickness in cm (default 30).
`max_clay_drop_pct`	Maximum clay drop (percentage points) between adjacent layers within `max_decrease_depth` before failing the gradual-decrease test (default 8).
`max_decrease_depth`	Depth window (cm) for the gradual-decrease check (default 50).

Details

Required (AND-combined) sub-tests:

- Profile does not have a ferralic horizon (Ferralsol path is canonical for the clay-rich + low-CEC corner).
- clay % >= `min_clay`.
- fe_dcb_pct >= `min_fe_dcb`.
- thickness >= `min_thickness`.

Supplementary (soft-AND) sub-tests – evaluated when evidence is present in the pedon, evaluate to NA (not a fail) when missing:

- structure_type matches polyhedral / nutty / (sub)angular blocky.

- slickensides / shiny ped surfaces present (proxy for WRB's "shiny ped surfaces").
- clay does not decrease abruptly between adjacent layers within 50 cm of the surface (gradual-decrease pattern; drop > 8 percentage points fails).

Supplementary tests fail (return passed = FALSE) only when evidence actively contradicts the criterion; missing evidence is permissive.

Value

A `DiagnosticResult`.

References

IUSS Working Group WRB (2022), Chapter 3, Nitic horizon.

`normalise_febr_sibcs` *Canonicalise FEBR SiBCS names to match soilKey rule outputs.*

Description

FEBR ships SiBCS labels in mixed legacy/modern form (`"Podzolicos"` for old name of Argissolos, singular vs plural, Portuguese accents). This helper folds them to the form produced by `run_sibcs_key()` so that benchmark accuracies can be computed without false negatives.

Usage

```
normalise_febr_sibcs(x, level = c("order", "subordem"))
```

Arguments

`x`	Character vector of FEBR SiBCS names.
`level`	One of `"order"` (default) or `"subordem"`.

Value

Character vector of normalised SiBCS names; `NA` for labels that are out-of-scope for the comparison (e.g.\ legacy `"Solos"` category).

See Also

`normalise_febr_wrb`, `normalise_febr_usda`

`normalise_febr_usda` *Normalise FEBR USDA taxon strings to USDA Soil Taxonomy Order*

Description

FEBR ships USDA Soil Taxonomy labels at the subgroup or great-group granularity (e.g. "TYPIC HAPLUDULT", "ACRUSTOX"). The suffix of the final word encodes the Order: `...OX` -> Oxisols, `...ULT` -> Ultisols, `...EPT` -> Inceptisols, etc. This helper extracts the Order from the suffix so the benchmark can compare against `classify_usda()$rsg_or_order` at `level = "order"`.

Usage

```
normalise_febr_usda(x)
```

Arguments

`x`	Character vector of FEBR USDA names.

Value

Character vector of normalised Order names ("Oxisols", "Ultisols", "Inceptisols", ...).

`normalise_febr_wrb` *Normalise FEBR WRB taxon strings to RSG-only*

Description

FEBR ships WRB names with full qualifier strings, e.g. "HUMIC FERRALSOL", "HAPLIC ACRISOL (ALUMIC, HYPERDYSTRIC, ...)". The trailing word (before any qualifier parens) is the RSG. This helper extracts and normalises it to soilKey's plural Title Case form ("Ferralsols", "Acrisols"), matching `ClassificationResult$rsg_or_order`.

Usage

```
normalise_febr_wrb(x)
```

Arguments

`x`	Character vector of FEBR WRB names.

Value

Character vector of normalised RSG names.

`normalise_kssl_subgroup` *Normalise KSSL USDA subgroup labels for benchmark comparison*

Description

KSSL stores ‘samp_taxsubgrp‘ in lower-case, space-separated form ("typic hapludalfs", "aquic argiudolls"). soilKey’s ‘classify_usda()‘ returns Title Case names ("Typic Hapludalfs"). The benchmark runner at ‘level = "subgroup"‘ lowercases both sides and trims whitespace, but this helper makes the normalisation explicit when users want to compare KSSL labels against arbitrary classifier output. Idempotent.

Usage

```
normalise_kssl_subgroup(x)
```

Arguments

`x`	Character vector of KSSL subgroup names.

Value

Lowercase, single-space-separated vector.

`ollama_is_running` *Is the local Ollama HTTP API reachable?*

Description

Probes `http://127.0.0.1:11434/api/tags` (the standard Ollama endpoint) with a short HTTP HEAD-style GET. Returns `TRUE` only if the request returns HTTP 200 in under `timeout_s` seconds. Used by `vlm_pick_provider` for the `provider = "auto"` fallback chain. Override the URL via `options(soilKey.ollama_url = "http://host:port")`.

Usage

```
ollama_is_running(url = NULL, timeout_s = 1.5)
```

Arguments

`url`	Override URL to probe (default reads `getOption("soilKey.ollama_url",default = "http://127.0.0.1:11434/api/tags")`).
`timeout_s`	Request timeout in seconds (default 1.5).

Value

Logical scalar.

`organic_material` *Organic material (WRB 2022 Ch 3.3.13): \>= 20% SOC + recognisability criteria. v0.3.3: SOC threshold only.*

Description

Organic material (WRB 2022 Ch 3.3.13): \>= 20% SOC + recognisability criteria. v0.3.3: SOC threshold only.

Usage

```
organic_material(pedon, min_oc = 20)
```

Arguments

`pedon`	A `PedonRecord`.
`min_oc`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`organotechnic_material` *Organotechnic material (WRB 2022 Ch 3.3.14): \>= 35% volume of artefacts that themselves contain \>= 20% organic C. Soil itself has < 20% SOC.*

Description

Organotechnic material (WRB 2022 Ch 3.3.14): \>= 35% volume of artefacts that themselves contain \>= 20% organic C. Soil itself has < 20% SOC.

Usage

```
organotechnic_material(pedon, min_artefacts = 35, max_oc = 20)
```

Arguments

`pedon`	A `PedonRecord`.
`min_artefacts`	Numeric threshold or option (see Details).
`max_oc`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`ornithogenic_material` *Ornithogenic material (WRB 2022 Ch 3.3.15): bird-influenced topsoil. Mehlich-3 P >= 750 mg/kg + designation pattern* `Aornit|Bornit`.

Description

Ornithogenic material (WRB 2022 Ch 3.3.15): bird-influenced topsoil. Mehlich-3 P >= 750 mg/kg + designation pattern `Aornit|Bornit`.

Usage

```
ornithogenic_material(pedon, min_p_mehlich3 = 750)
```

Arguments

`pedon`	A `PedonRecord`.
`min_p_mehlich3`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`ossl_demo_sa` *Synthetic OSSL South America demo subset*

Description

A small, deterministic, OSSL-shaped artefact for use in vignettes, examples and tests when the real Open Soil Spectral Library data is not available (no network, sensitive deployment, CI). The object has the canonical `list(Xr, Yr, metadata)` shape consumed by `predict_ossl_mbl` / `fill_from_spectra`, so the in-package demo path is identical to the real-data path.

Usage

```
ossl_demo_sa
```

Format

A list with three elements:

`Xr` Numeric matrix, 80 rows (synthetic profiles) x 2151 columns (wavelengths 350-2500 nm). Reflectance values in [0.05, 0.85].

`Yr` Data frame, 80 rows x 9 columns (`clay_pct`, `sand_pct`, `silt_pct`, `cec_cmol`, `bs_pct`, `ph_h2o`, `oc_pct`, `fe_dcb_pct`, `caco3_pct`). Property ranges follow the OSSL global summary statistics.

`metadata` Named list with provenance information (`region`, `n_profiles`, `snapshot`, `seed`, `note`, ...).

Details

This is a **synthetic** placeholder: the spectra are generated from a tropical-soil baseline plus property-correlated absorption bands (1400 nm OH-water, 1900 nm clay-OH, 2200 nm Al-OH, 900 nm Fe-oxide) with deterministic noise. It is *not* a substitute for real OSSL measurements. For paper-grade work, populate a real OSSL artefact via:

```
ossl_lib <- download_ossl_subset(region = "south_america")
```

Re-build the demo with `source("data-raw/build_ossl_demo.R")`.

Source

Synthetic; built by `data-raw/build_ossl_demo.R` with seed 20260430. The OSSL property ranges that drove the simulation come from Sanderman, J. *et al.* (2024), *Open Soil Spectral Library*, `https://soilspectroscopy.org/`.

Examples

```
data(ossl_demo_sa)
dim(ossl_demo_sa$Xr)
#> [1]   80 2151
head(ossl_demo_sa$Yr)

## Not run:
# Use it as the ossl_library argument to predict_ossl_mbl():
pedon <- make_synthetic_pedon_with_spectra()
fill_from_spectra(pedon,
                  library      = "ossl",
                  method       = "mbl",
                  ossl_library = ossl_demo_sa)

## End(Not run)
```

`ossl_library_template` *Canonical schema for an 'ossl_library' object*

Description

`predict_ossl_mbl` and `predict_ossl_plsr_local` take an `ossl_library` argument that must be a list with two named elements:

Usage

```
ossl_library_template(
  wavelengths = 350:2500,
 properties = c("clay_pct", "sand_pct", "silt_pct", "cec_cmol", "bs_pct", "ph_h2o",
    "oc_pct", "fe_dcb_pct", "caco3_pct")
)
```

Arguments

`wavelengths`	Integer vector of wavelengths (default `350:2500` nm for Vis-NIR/SWIR).
`properties`	Character vector of property column names to seed the empty `Yr` data.frame with.

Details

- `Xr`: numeric matrix, rows = OSSL training spectra, columns = wavelengths. Must align (after preprocessing) with the column space used by the spectra you predict on.
- `Yr`: data.frame keyed by property name (e.g. `clay_pct`, `cec_cmol`), one row per training spectrum.

This function returns an empty template you can populate from a real OSSL extract (e.g. via the `ossl-import` Python package or the public S3 mirror at `https://storage.googleapis.com/soilspec4gg-public/`).

soilKey does **not** bundle OSSL data; until you populate this template with real values, all 'predict_ossl_*' calls fall back to the deterministic synthetic predictor (which prints a warning).

Value

A list with `Xr` (a 0-row matrix of the right column dimension) and `Yr` (an empty data.frame with the requested columns).

`oxic_usda` *Oxic horizon (USDA Soil Taxonomy)*

Description

The USDA oxic horizon is the diagnostic of Oxisols. Its central criteria match the WRB 2022 ferralic horizon closely enough that v0.2 simply delegates: every fixture that classifies as Oxisol via USDA also classifies as Ferralsol via WRB and vice-versa. The fine-grained differences (USDA's water-dispersible-clay test, the sand-fraction weatherable-mineral cut-offs) are tracked in the diagnostics.yaml for v0.8 refinement.

Usage

```
oxic_usda(pedon, ...)
```

Arguments

`pedon`	A `PedonRecord`.
`...`	Passed to `ferralic`.

Value

A `DiagnosticResult` (with `name = "oxic_usda"`).

References

Soil Survey Staff (2014). *Keys to Soil Taxonomy*, 12th edition. USDA-NRCS, Washington DC. Chapter 3 – Diagnostic Horizons; oxic.

`panpaic` *Panpaic horizon (WRB 2022 Ch 3.1)*

Description

From Quechua *p'anpay* = "to bury". A buried diagnostic horizon (any horizon whose original surface was subsequently overlain by younger material). Used by the Panpaic qualifier and by the Cambisols / Anthrosols branches.

Usage

```
panpaic(pedon)
```

Arguments

`pedon`	A `PedonRecord`.

Details

v0.3.5 detection: designation pattern starting with a digit other than 1 (e.g. `2A`, `2Bw`, `3C`) – the WRB / FAO convention for buried horizons – OR a `b` suffix in the designation (e.g. `Ahb`, `Bwb`).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`PedonRecord` *PedonRecord: structured representation of a single pedon*

Description

PedonRecord: structured representation of a single pedon

PedonRecord: structured representation of a single pedon

Details

The central data carrier in soilKey. A PedonRecord bundles everything we know about one soil profile: site metadata, the horizons table (with a fixed canonical schema — see `horizon_column_spec`), optional Vis-NIR/MIR spectra, profile photographs, source documents, and a provenance log that records, per (horizon, attribute) pair, where each value came from (`measured`, `extracted_vlm`, `predicted_spectra`, `inferred_prior`, `user_assumed`).

All diagnostic functions (`argic`, `ferralic`, `mollic`, ...) consume a PedonRecord directly. The provenance log is what allows the final `ClassificationResult` to assign a meaningful evidence grade.

Value

An R6 object of class `PedonRecord`.

Public fields

`site` List. Site-level metadata: `lat`, `lon`, `crs` (default 4326), `date`, `country`, `elevation_m`, `slope_pct`, `aspect_deg`, `landform`, `parent_material`, `land_use`, `vegetation`, `drainage_class`, plus an arbitrary `id`.

`horizons` data.table with the canonical horizon schema.

`spectra` List with optional `vnir` matrix (rows = horizons, cols = wavelengths in nm), `mir` matrix, and `metadata` list.

`images` List of named lists describing profile photographs.

`documents` List of named lists describing source documents.

`provenance` data.table with columns `horizon_idx`, `attribute`, `source`, `confidence`, `notes`.

Methods

Public methods:

- `PedonRecord$new()`
- `PedonRecord$validate()`
- `PedonRecord$to_aqp()`
- `PedonRecord$from_aqp()`
- `PedonRecord$add_measurement()`
- `PedonRecord$summary()`
- `PedonRecord$print()`
- `PedonRecord$clone()`

Method `new()`**:** Construct a PedonRecord.

Usage:

```
PedonRecord$new(
  site = NULL,
  horizons = NULL,
  spectra = NULL,
  images = NULL,
  documents = NULL,
  provenance = NULL
)
```

Arguments:

`site` List of site-level metadata.

`horizons` data.frame/data.table of horizons.

`spectra` Optional list with `vnir`, `mir`, `metadata`.

`images` Optional list of image descriptors.

`documents` Optional list of document descriptors.

`provenance` Optional provenance data.table; if NULL, an empty one is created.

Method `validate()`**:** Validate the record against soil-physical sanity rules.

Checks: top < bottom for every horizon; no overlapping depths; clay+silt+sand sum to 100 ± 2 where all three are reported; pH values plausible (1..12); CEC >= sum of exchangeable bases (Ca, Mg, K, Na); Munsell value/chroma in plausible ranges; coarse fragments percent in [0, 100]; OC geographic ranges. Returns a list with `valid`, `errors`, `warnings`, `n_horizons`.

Usage:

```
PedonRecord$validate(strict = FALSE, verbose = TRUE)
```

Arguments:

`strict` If `TRUE`, throws on errors instead of returning.

`verbose` If `TRUE`, prints messages via cli.

Returns: Invisibly, a list summarising the validation outcome.

Method `to_aqp()`**:** Coerce to an aqp `SoilProfileCollection`.

Usage:

```
PedonRecord$to_aqp()
```

Returns: A `SoilProfileCollection`. Requires the aqp package.

Method **`from_aqp()`:** Populate this record from an aqp `SoilProfileCollection`.

Usage:

```
PedonRecord$from_aqp(spc, top_col = "top_cm", bottom_col = "bottom_cm")
```

Arguments:

`spc` A `SoilProfileCollection`.

`top_col` Name of the top-depth column in `spc` (mapped to `top_cm`).

`bottom_col` Name of the bottom-depth column (mapped to `bottom_cm`).

Returns: Invisibly self (mutated in place).

Method **`add_measurement()`:** Add a measurement (or extracted/predicted value) and record its provenance.

Usage:

```
PedonRecord$add_measurement(
  horizon_idx,
  attribute,
  value,
  source = "measured",
  confidence = 1,
  notes = NA_character_,
  overwrite = FALSE
)
```

Arguments:

`horizon_idx` Integer horizon index (1-based).

`attribute` Name of the horizon column to set.

`value` New value for that cell.

`source` One of "measured", "extracted_vlm", "predicted_spectra", "inferred_prior", "user_assumed".

`confidence` Numeric in [0, 1].

`notes` Optional free-text note.

`overwrite` If `FALSE` (default) and the cell already has a value from a more authoritative source, leave it alone. If `TRUE`, overwrite.

Returns: Invisibly self.

Method **`summary()`:** Compact summary list (for serialization or testing).

Usage:

```
PedonRecord$summary(...)
```

Arguments:

`...` Ignored (S3 summary signature compatibility).

Method **`print()`:** Pretty-print the record.

Usage:

```
PedonRecord$print(...)
```

Arguments:

`...` Ignored (S3 print signature compatibility).

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
PedonRecord$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Examples

```
# The canonical fixtures return ready-built PedonRecords:
pedon <- make_ferralsol_canonical()
pedon$site$id
nrow(pedon$horizons)
```

`pedons_from_spectral_table`

Build PedonRecords with attached Vis-NIR/MIR spectra from a table

Description

Groups a reflectance + metadata table by profile and returns one `PedonRecord` per profile, with each profile's sample rows stacked into `$spectra$vnir` (rows = horizons, cols = wavelengths) and the lab attributes / depths written to the horizons. Taxonomic labels are stored in `$site` (`reference_wrb` / `reference_sibcs` / `reference_st`). These pedons are the query objects for `classify_*(gapfill = list(method = "spectra", ossl_library = <lib>))`.

Usage

```
pedons_from_spectral_table(
  reflectance,
  metadata,
  id_col = "id",
  profile_col = NULL,
  wavelengths = NULL,
  resample_to = NULL,
  property_map = NULL,
  label_map = NULL,
  normalize = c("auto", "none", "percent"),
  keep_properties = FALSE,
  verbose = TRUE
)
```

Arguments

`reflectance`	Reflectance data: a matrix / data.frame with rows = samples and columns named by wavelength (nm); OR a long data.frame with `id_col`, `wavelength_nm`, `reflectance`; OR a path to a CSV in either form.
`metadata`	A data.frame with one row per sample carrying `id_col` plus lab attributes and optional taxonomic labels and `lat`/ `lon`. Rows are aligned to `reflectance` by `id_col`.
`id_col`	Sample identifier column shared by both tables (default `"id"`).
`profile_col`	Column grouping samples into profiles (default `id_col`: one profile per sample, e.g. a topsoil library).
`wavelengths`	Optional explicit wavelength vector (nm) when the reflectance columns are not wavelength-named.
`resample_to`	Optional target wavelength grid (nm) to linearly resample every spectrum onto (e.g. `350:2500`); default keeps the native grid.
`property_map`, `label_map`	Optional named lists overriding the alias auto-detection, e.g. `property_map = list(clay_pct = "ARGILA")`.
`normalize`	One of `"auto"` (divide by 100 when values look like percent), `"percent"`, or `"none"`.
`keep_properties`	If `TRUE`, also write the mapped lab attributes to the horizons (default `FALSE` – a field pedon usually has only the scan, which is the scenario the spectral fill targets).
`verbose`	Print a one-line summary (default `TRUE`).

Value

A list of `PedonRecord` objects.

See Also

`read_spectral_library`, `benchmark_spectral_fill`

`pedon_json_schema` *JSON Schema for a soilKey PedonRecord*

Description

Returns a Draft-2020-12 JSON Schema describing the canonical `PedonRecord` structure: a `site` object with site-level metadata plus a `horizons` array where each element matches the canonical horizon schema documented by `horizon_column_spec`.

Usage

```
pedon_json_schema(as = c("list", "json"), pretty = TRUE)
```

Arguments

`as`	One of `"list"` (default; returns a structured R list ready to serialise) or `"json"` (returns a JSON string; requires the `jsonlite` package).
`pretty`	Logical, only used for `as = "json"`.

Value

A list (default) or a JSON string.

Examples

```
## Not run:
schema <- pedon_json_schema()
names(schema)
#> [1] "$schema"     "$id"         "title"       "type"        "required"   "properties"

# Validate a JSON profile against the schema:
if (requireNamespace("jsonvalidate", quietly = TRUE)) {
  schema_json <- pedon_json_schema(as = "json")
  jsonvalidate::json_validate('{"site":{...},"horizons":[...]}',
                               schema_json, engine = "ajv")
}

## End(Not run)
```

`pedon_to_spc` *Convert a soilKey PedonRecord to an aqp SoilProfileCollection*

Description

The mapping respects aqp's expected column conventions and sets the metadata required by `getArgillicBounds()`, `getCambicBounds()`, and `mollicEpipedon()`:

Usage

```
pedon_to_spc(pedon)
```

Arguments

`pedon`	A `PedonRecord`.

Details

- `id` from `pedon$site$id`
- `top` / `bottom` from `top_cm` / `bottom_cm`
- `name` (designation) from `designation`
- `texcl` (texture class) derived via `texture_class_from_pct`

- `clay`, `silt`, `sand` from `clay_pct` / `silt_pct` / `sand_pct`
- `m_hue`, `m_value`, `m_chroma`, `d_value`, `d_chroma` from `munsell_*_moist` and `munsell_*_dry`

Internal use; the soilKey diagnostics call this on the fly when `engine = "aqp"`. Direct use is supported for users who want to plug additional aqp algorithms (`slab`, `slice`, `glom`) into a soilKey workflow.

Value

A `aqp::SoilProfileCollection` with one site (the pedon) and one row per horizon.

`petrocalcic` *Petrocalcic horizon (WRB 2022)*

Description

A continuously cemented variant of the calcic horizon. Same chemistry (CaCO3 \>= 15%) plus moderate-or-greater cementation in at least 50% of the layer.

Usage

```
petrocalcic(pedon, min_thickness = 10, min_caco3_pct = 15)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Numeric threshold or option (see Details).
`min_caco3_pct`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`petroduric` *Petroduric horizon (WRB 2022): cemented duric.*

Description

Petroduric horizon (WRB 2022): cemented duric.

Usage

```
petroduric(pedon, min_thickness = 10, min_duripan_pct = 10)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Numeric threshold or option (see Details).
`min_duripan_pct`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`petrogypsic` *Petrogypsic horizon (WRB 2022): cemented gypsic.*

Description

Petrogypsic horizon (WRB 2022): cemented gypsic.

Usage

```
petrogypsic(pedon, min_thickness = 10, min_gypsum_pct = 5)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Numeric threshold or option (see Details).
`min_gypsum_pct`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`petroplinthic` *Petroplinthic horizon (WRB 2022): cemented plinthic.*

Description

Petroplinthic horizon (WRB 2022): cemented plinthic.

Usage

```
petroplinthic(pedon, min_thickness = 10, min_plinthite_pct = 15)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Numeric threshold or option (see Details).
`min_plinthite_pct`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`phaeozem` *Phaeozem RSG diagnostic (WRB 2022)*

Description

Tests whether a profile satisfies the Phaeozem RSG criteria: a mollic horizon AND no secondary carbonate accumulation anywhere in the profile.

Usage

```
phaeozem(pedon)
```

Arguments

`pedon`	A `PedonRecord`.

Value

A `DiagnosticResult`.

References

IUSS Working Group WRB (2022), Chapter 5, Phaeozems.

`pick_engine` *Choose the best diagnostic engine for a single pedon*

Description

Per-pedon heuristic: returns `"aqp"` if the pedon's horizon table has the morphological richness that makes aqp's canonical NRCS dispatch reliable, otherwise returns `"soilkey"` (the more permissive hand-coded path).

Usage

```
pick_engine(pedon, min_score = 3L)
```

Arguments

`pedon`	A `PedonRecord`.
`min_score`	Integer (1-5). Minimum completeness score for `"aqp"` engine to fire (default 3).

Value

Character: `"aqp"` or `"soilkey"`.

Heuristic

We score each pedon on a 0-5 morphology-completeness scale; aqp fires when score \>= `min_score` (default 3). The five axes:

1. **Designation present** (any layer has a non-blank `designation`, e.g. "A1", "Bt2", "Bw").
2. **Texture quantitative** (any layer has both `clay_pct` and `sand_pct` populated).
3. **Munsell complete** (any layer has all three of `munsell_hue_moist`, `munsell_value_moist`, `munsell_chroma_moist` populated).
4. **Structure recorded** (any layer has a non-blank `structure_grade`).
5. **Clay films / argic evidence** (any layer has a non-blank `clay_films_amount` or designation pattern matching `Bt`).

Why this matters

On BDsolos RJ (data-rich), the heuristic recommends aqp for ~99 canonical thresholds). On LUCAS topsoil-only (data-sparse), it recommends aqp for ~0 clay-films / designation axes are unfilled. Calling `classify_*(pedon)` routed through the heuristic gives the correct engine per pedon, recovering both the BDsolos RJ lift AND the LUCAS robustness.

See Also

`argic`, `cambic`.

`pick_engine_batch` *Per-pedon batch engine recommendation*

Description

Vectorised version of `pick_engine` returning the recommended engine for each pedon in a list.

Usage

```
pick_engine_batch(pedons, min_score = 3L)
```

Arguments

`pedons`	A list of `PedonRecord` objects.
`min_score`	Integer; forwarded to `pick_engine`.

Value

Character vector of length(pedons) with values "aqp" or "soilkey".

`pisoplinthic` *Pisoplinthic horizon (WRB 2022): pisolitic plinthic. v0.3.3 detects via designation pattern* `Bspl / Bvpi` *or via plinthite \>= 15% AND structure_type containing 'pisol'.*

Description

Pisoplinthic horizon (WRB 2022): pisolitic plinthic. v0.3.3 detects via designation pattern `Bspl / Bvpi` or via plinthite \>= 15% AND structure_type containing 'pisol'.

Usage

```
pisoplinthic(pedon, min_thickness = 15, min_plinthite_pct = 15)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Numeric threshold or option (see Details).
`min_plinthite_pct`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`pi_to_confidence` *Map a 95% prediction interval to a [0, 1] confidence score*

Description

Tightens confidence as the prediction interval narrows relative to the predicted value: `confidence = 1 - (PI95_width / |value|) / 4`, floored at 0 and capped at 1. When `value` is near zero we fall back to an absolute-width heuristic so we never blow up.

Usage

```
pi_to_confidence(pi95_low, pi95_high, value = NULL)
```

Arguments

`pi95_low`	Lower 2.5% quantile of the prediction.
`pi95_high`	Upper 97.5% quantile of the prediction.
`value`	Optional point prediction. When supplied, normalisation is by `|value|`; otherwise by `|midpoint|`.

Details

Properties of the mapping:

- Zero-width interval -> confidence = 1.
- Interval whose width equals `|value| * 4` -> confidence = 0.
- NA value or NA bounds -> confidence = 0.5 (neutral).

Value

Numeric in `[0, 1]`.

`plaggic` *Plaggic horizon (WRB 2022): sod-derived topsoil >= 20 cm with low BD AND independent evidence of human input.*

Description

v0.9.2.C tightening: the v0.3.3 implementation accepted ANY thick, low-BD, OC-rich A horizon, which over-fired across natural mollic / umbric / chernic surfaces. The diagnostic now requires, in addition to the OC + BD + thickness baseline, at least one independent anthropogenic-input marker:

- `p_mehlich3_mg_kg >= 50` (sustained sod / manure additions concentrate Mehlich-3 P in the topsoil), OR

- `artefacts_pct` > `0` (any human artefact volume fraction is sufficient as a presence signal), OR
- designation pattern `Apl` / `Aplg` / `Apk` / explicit "plagg".

Without one of those markers the diagnostic returns FALSE even when OC + BD + thickness pass. This mirrors the v0.9.1 `qual_plaggic` gate but enforces the rule at the diagnostic level so any caller (SiBCS, USDA, future modules) inherits the protection.

Usage

```
plaggic(
  pedon,
  min_thickness = 20,
  max_bd = 1.5,
  min_oc = 0.6,
  min_p_mehlich3 = 100
)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Numeric threshold or option (see Details).
`max_bd`	Numeric threshold or option (see Details).
`min_oc`	Numeric threshold or option (see Details).
`min_p_mehlich3`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`planic_features` *Planic features (WRB 2022)*

Description

Tests whether the profile shows an abrupt textural change between adjacent horizons (clay-doubling within 7.5 cm vertical distance, typically at the E/Bt boundary). Diagnostic of Planosols.

Usage

```
planic_features(pedon, min_ratio = 2, require_abrupt_boundary = TRUE)
```

Arguments

`pedon`	A `PedonRecord`.
`min_ratio`	Minimum clay ratio (default 2.0).
`require_abrupt_boundary`	If TRUE (default), the upper horizon must have `boundary_distinctness` matching "abrupt".

Value

A `DiagnosticResult`.

References

IUSS Working Group WRB (2022), Chapter 5, Planosols.

`planosol` *Planosol RSG gate (WRB 2022 Ch 4, p 107)*

Description

WRB-canonical: abrupt textural difference \<= 75 cm AND, in 5 cm directly above or below the abrupt textural difference, stagnic properties (>= 50% redoximorphic features) AND reducing conditions.

Usage

```
planosol(pedon, strict = NULL)
```

Arguments

`pedon`	A `PedonRecord`.
`strict`	Logical or `NULL`. When `NULL` (default) it resolves via `getOption("soilKey.rsg_strict", FALSE)`. `TRUE` disables the planic-features fallback.

Details

v0.3.4 enforces all three components. The 5-cm-window restriction is relaxed to "the layer immediately above or below the abrupt textural difference satisfies stagnic + reducing".

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

Tier-3 strict mode (v0.9.98)

With `strict = TRUE` the `planic_features` fallback path is disabled. Strict mode requires the canonical evidence – an abrupt textural difference *plus* measured stagnic and reducing conditions in the bracketing layer – and will not accept the simpler clay-doubling proxy on its own.

`plinthic` *Plinthic horizon (WRB 2022)*

Description

Tests whether any horizon meets the plinthic horizon criteria. Plinthite is Fe-rich material that hardens irreversibly on repeated wetting and drying; the plinthic horizon is the diagnostic of Plinthosols.

Usage

```
plinthic(pedon, min_thickness = 15, min_plinthite_pct = 15)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Minimum thickness in cm (default 15).
`min_plinthite_pct`	Minimum volume % plinthite (default 15).

Details

Sub-tests:

- `test_plinthite_concentration` – plinthite volume % >= 15
- `test_minimum_thickness` – thickness >= 15 cm

v0.2 limitations: WRB 2022 also accepts profiles with >= 40% red Fe-rich mottles as alternative criterion – not yet wired. The "irreversibly hardens" criterion is conceptual and requires field observation; v0.2 takes `plinthite_pct` as already representing true plinthite (as opposed to soft mottles).

Value

A `DiagnosticResult`.

v0.9.72 designation morphological inference (opt-in)

Field-described Brazilian Plintossolos profiles (e.g.\ the Embrapa Redape curated dataset) routinely encode plinthite via the designation suffix `f` in the master letter sequence (e.g.\ `Btf`, `2Btf`, `Cf`) – the curator's direct assertion that plinthite is present – without recording `plinthite_pct` as a numeric volume percent.

With `options(soilKey.plinthic_designation_inference = TRUE)` the function accepts a layer as plinthic when:

1. the canonical `plinthite_pct` test is `NA` for that layer, AND
2. the designation matches `[A-Z]+[A-Za-z]*f[0-9]?` (a `f` master-letter modifier in any sub-position).

Default is `FALSE` (canonical behaviour preserved).

References

IUSS Working Group WRB (2022), Chapter 3, Plinthic horizon.

`posterior_classify` *Bayesian posterior classifier (optional)*

Description

Combines a deterministic `ClassificationResult` with a spatial prior. The deterministic key remains authoritative – this function reports only an alternative probabilistic view useful for downstream uncertainty quantification.

Usage

```
posterior_classify(result, prior, epsilon = 0.001)
```

Arguments

`result`	A `ClassificationResult` from `classify_wrb2022`.
`prior`	A spatial-prior data.table (as returned by `spatial_prior`).
`epsilon`	Small smoothing constant added to all prior entries before normalising, so RSGs unseen by the prior do not receive zero posterior.

Details

Posterior is computed under the simple model:

$$P(rsg|site, evidence) \propto L(rsg|evidence) \times P(rsg|site)$$

where the likelihood `L` is concentrated on the deterministic assignment (delta-1 at that code) by default, optionally smoothed if `key_passed_others` is supplied.

Value

A `data.table` with columns `rsg_code`, `prior`, `likelihood`, `posterior`.

`predict.soilKey_pls_model` *Predict from a soilKey_pls_model*

Description

S3 method that applies a trained PLSR model from `train_pls_from_ossl` to a (pre-processed) numeric matrix and returns predictions plus a 95 built from the cross-validated training RMSE.

Usage

```
## S3 method for class 'soilKey_pls_model'
predict(object, X, ...)
```

Arguments

`object`	A `soilKey_pls_model` object.
`X`	A pre-processed numeric matrix (rows = samples, columns = wavelengths). Must have the same column count used at training time.
`...`	Reserved.

Value

A data.frame with columns `value`, `pi95_low`, `pi95_high`, one row per sample.

`predict_from_spectra` *Predict soil properties from spectra*

Description

Ergonomic, named entry point for the OSSL-backed predictive pipeline. Accepts either a `PedonRecord` or a numeric spectra matrix, applies the same preprocessing used at training time (recorded on each model), and returns predictions in the canonical long-form schema.

Usage

```
predict_from_spectra(
  pedon_or_spectra,
  models = NULL,
  properties = NULL,
  overwrite = FALSE,
  verbose = TRUE,
  ...
)
```

Arguments

`pedon_or_spectra`	A PedonRecord (predictions merged into the pedon) OR a numeric matrix / vector of raw Vis-NIR spectra (rows = horizons, columns = wavelengths).
`models`	A named list of `soilKey_pls_model` objects (output of `train_pls_from_ossl`). Required.
`properties`	Character vector of property names to predict. Defaults to all properties in `models`.
`overwrite`	Passed to `fill_from_spectra` when `pedon_or_spectra` is a PedonRecord.
`verbose`	Verbosity passed downstream.
`...`	Ignored (reserved for future backends).

Details

When `pedon_or_spectra` is a `PedonRecord`, this function delegates to `fill_from_spectra` with `method` = `"pretrained"` and the predictions are written back to the pedon (with source = `"predicted_spectra"` provenance). When `pedon_or_spectra` is a numeric matrix or vector, this function returns the prediction data.table directly without touching any pedon.

Value

Either the mutated `PedonRecord` (invisibly) or a data.table with columns `horizon_idx`, `property`, `value`, `pi95_low`, `pi95_high`, `n_neighbors`.

Examples

```
## Not run:
lib <- download_ossl_subset(region = "south_america")
models <- train_pls_from_ossl(lib,
                                properties = c("clay_pct", "ph_h2o"))
predict_from_spectra(my_pedon, models = models)

## End(Not run)
```

---

predict_lab_from_spectra

Predict CIE Lab from Vis-NIR reflectance spectra

Description

Convenience wrapper: `predict_xyz_from_spectra` followed by the standard CIE Lab transform under D65 / 2-degree observer.

Usage

```
predict_lab_from_spectra(spectra, wavelengths)
```

Arguments

`spectra`	Reflectance values, in 0..1 or 0..100. A numeric vector (one sample), a numeric matrix (rows = samples, cols = wavelengths) or a data.frame.
`wavelengths`	Numeric vector of the wavelengths (in nm) corresponding to the columns of `spectra`. Must cover at least 400-700 nm; values outside 380-780 are ignored.

Value

A data.frame with columns `L`, `a`, `b`.

`predict_munsell_from_spectra`

Predict Munsell hue / value / chroma from Vis-NIR reflectance spectra

Description

Combines `predict_xyz_from_spectra` with the Munsell renotation interpolation in **munsellinterpol** (CRAN, GPL). Returns hue (e.g. `"7.5YR"`), value (0..10) and chroma (0..20) per sample, plus the soilKey fields

Usage

```
predict_munsell_from_spectra(spectra, wavelengths, round_chip = TRUE)
```

Arguments

`spectra`	Reflectance values, in 0..1 or 0..100. A numeric vector (one sample), a numeric matrix (rows = samples, cols = wavelengths) or a data.frame.
`wavelengths`	Numeric vector of the wavelengths (in nm) corresponding to the columns of `spectra`. Must cover at least 400-700 nm; values outside 380-780 are ignored.
`round_chip`	If `TRUE` (default), snaps the predicted HVC to the nearest standard Munsell chip grid via `munsellinterpol::roundHVC()`. `FALSE` returns continuous HVC (useful for further numeric work).

Details

The Munsell renotation is defined under *Illuminant C*, while the colorimetry here is computed under D65, so the conversion adapts D65 -> C. It calls `munsellinterpol::XYZtoMunsell(XYZ, white =)` (munsellinterpol >= 3.4-0), which performs that chromatic adaptation internally, and falls back to the numerically identical XYZ -> CIELAB(D65) -> `LabToMunsell()` route on older versions. Feeding D65 chromaticities straight to `xyYtoMunsell()` (with no `white`) would bias every colour toward green-yellow (a perfect neutral would return Chroma ~ 0.65 rather than 0); this routine avoids that. The D65 reference white is derived from the same bundled CIE table the colorimetry integrates against (so a constant-reflectance spectrum maps to an exact neutral, and a perfect reflecting diffuser to Munsell value 10), and the conversion is vectorised over all rows of `spectra` at once. At zero Chroma the Munsell hue is undefined, so a neutral is reported with hue `"N"` in both the rounded and the continuous (`round_chip = FALSE`) notation. `munsell_hue_moist`, `munsell_value_moist`, `munsell_chroma_moist` ready to write into a `PedonRecord` via the pedon's `add_measurement` method (see also `fill_munsell_from_spectra`).

This is the v0.9.47 unblock for the v0.9.35 Argissolo Vermelho / Amarelo / Vermelho-Amarelo color-confusion case: when a user has Vis-NIR spectra (which Embrapa's BDsolos / FEBR do not include but the OSSL does), the Munsell hue can be recovered physically without waiting for the surveyor's morphological description.

Value

A data.frame with columns `munsell_hue_moist`, `munsell_value_moist`, `munsell_chroma_moist`, `munsell_string` (e.g. `"7.5YR 4/6"`), `X`, `Y`, `Z`, one row per sample.

Examples

```
## Not run:
# White reflector across the visible: should map to a near-neutral
# high-value Munsell color.
wl <- seq(380, 780, by = 5)
R  <- rep(0.9, length(wl))
predict_munsell_from_spectra(R, wavelengths = wl)

## End(Not run)
```

`predict_ossl_mbl` *Memory-based learning prediction against the OSSL library*

Description

Predicts a set of soil properties from pre-processed Vis-NIR or MIR spectra using *memory-based learning* (MBL) – the recommended OSSL workflow for heterogeneous libraries. Defaults follow the literature (Ramirez-Lopez et al., 2013): `k = 100` neighbours, PLS-score dissimilarity, local PLS regression with 5 components, internal leave-one-out validation.

Usage

```
predict_ossl_mbl(
  X,
  properties,
  region = "global",
  k = 100L,
  ossl_library = NULL,
  ...
)
```

Arguments

`X`	A pre-processed numeric matrix (rows = horizons, columns = wavelengths).
`properties`	Character vector of OSSL-supported property names.
`region`	One of `"global"`, `"south_america"`, `"north_america"`, `"europe"`, `"africa"`.
`k`	Integer number of neighbours.
`ossl_library`	Optional list with the OSSL training spectra (`Xr`) and reference values (`Yr`, a data.frame keyed by `properties`). When `NULL`, the synthetic path is used.
`...`	Additional arguments forwarded to `resemble::mbl`.

Details

If `resemble::mbl` is installed and an `ossl_library` artefact is supplied (a list with elements `Xr`, `Yr`) the function delegates to `resemble::mbl()`; otherwise it returns a deterministic synthetic prediction conditioned on the input spectra so that downstream code, tests and vignettes run without external dependencies. The fallback is annotated via the `notes` attribute on the returned data.table.

Value

A data.table with columns `horizon_idx`, `property`, `value`, `pi95_low`, `pi95_high`, `n_neighbors`. The `"backend"` attribute records which path was taken (`"resemble"` or `"synthetic"`).

References

Ramirez-Lopez, L., Behrens, T., Schmidt, K., Stevens, A., Demattê, J. A. M., & Scholten, T. (2013). The spectrum-based learner: A new local approach for modeling soil Vis-NIR spectra of complex datasets. *Geoderma*, 195–196, 268–279.

`predict_ossl_plsr_local`

Local PLSR prediction against the OSSL library

Description

Selects the `k` nearest neighbours to each test spectrum in the OSSL training set and fits a local PLS regression. Like `predict_ossl_mbl`, this function dispatches to `resemble::mbl` (with a `local_algorithm = "pls"` setting) when the dependency is available; otherwise it falls back to the synthetic predictor.

Usage

```
predict_ossl_plsr_local(
  X,
  properties,
  region = "global",
  k = 100L,
  ossl_library = NULL,
  ...
)
```

Arguments

`X`	A pre-processed numeric matrix (rows = horizons, columns = wavelengths).
`properties`	Character vector of OSSL-supported property names.
`region`	One of `"global"`, `"south_america"`, `"north_america"`, `"europe"`, `"africa"`.
`k`	Integer number of neighbours.
`ossl_library`	Optional list with the OSSL training spectra (`Xr`) and reference values (`Yr`, a data.frame keyed by `properties`). When `NULL`, the synthetic path is used.
`...`	Additional arguments forwarded to `resemble::mbl`.

Value

A data.table with the same schema as `predict_ossl_mbl`.

`predict_ossl_pretrained`

Pre-trained OSSL prediction

Description

Applies the OSSL-distributed pre-trained PLSR / Cubist models for a set of soil properties to pre-processed spectra. Pre-trained models are loaded from `ossl_models`, a named list of property models that each must implement a `predict(model, X)` interface returning a data.frame with columns `value`, `pi95_low`, `pi95_high`. When `ossl_models` is `NULL`, the synthetic predictor is used.

Usage

```
predict_ossl_pretrained(
  X,
  properties,
  region = "global",
  ossl_models = NULL,
  ...
)
```

Arguments

`X`	A pre-processed numeric matrix (rows = horizons, columns = wavelengths).
`properties`	Character vector of OSSL-supported property names.
`region`	One of `"global"`, `"south_america"`, `"north_america"`, `"europe"`, `"africa"`.
`ossl_models`	Optional named list of pre-trained models, keyed by property name.
`...`	Reserved.

Value

A data.table with columns `horizon_idx`, `property`, `value`, `pi95_low`, `pi95_high`, `n_neighbors`. `n_neighbors` is `NA_integer_` for pre-trained models. The `"backend"` attribute records which path was taken.

`predict_xyz_from_spectra` *Predict CIE XYZ tristimulus values from Vis-NIR reflectance spectra*

Description

Numerically integrates user reflectance against the CIE 1931 2-degree Standard Observer color-matching functions, weighted by the D65 illuminant. Returns the tristimulus values X, Y, Z on the standard scale where $Y = 100$ for a perfect diffuse white.

Usage

```
predict_xyz_from_spectra(spectra, wavelengths)
```

Arguments

`spectra`	Reflectance values, in 0..1 or 0..100. A numeric vector (one sample), a numeric matrix (rows = samples, cols = wavelengths) or a data.frame.
`wavelengths`	Numeric vector of the wavelengths (in nm) corresponding to the columns of `spectra`. Must cover at least 400-700 nm; values outside 380-780 are ignored.

Value

A data.frame with columns `X`, `Y`, `Z`, one row per sample.

See Also

`predict_lab_from_spectra`, `predict_munsell_from_spectra`.

`preprocess_spectra` *Pre-process Vis-NIR or MIR spectra*

Description

Applies a chosen pre-processing pipeline to a numeric matrix of soil spectra. Rows are samples (typically horizons) and columns are wavelengths. Returns a numeric matrix; SG-based methods shorten the spectrum by `w - 1` columns at the edges (default `w = 5` so two columns are dropped from each side).

Usage

```
preprocess_spectra(X, method = c("snv+sg1", "snv", "sg1"), w = 5L, p = 2L)
```

Arguments

`X`	Numeric matrix or data.frame of spectra (rows = samples, columns = wavelengths). Wavelengths should be evenly spaced.
`method`	One of `"snv"`, `"sg1"`, `"snv+sg1"`. Default `"snv+sg1"`.
`w`	Window size for the SG filter. Must be odd; default 5.
`p`	Polynomial order for the SG filter. Default 2.

Details

Supported `method` values:

- `"snv"` Standard Normal Variate. Each row is centered on its own mean and divided by its own standard deviation.
- `"sg1"` Savitzky-Golay 1st derivative with a window of five wavelengths and a quadratic polynomial.
- `"snv+sg1"` SNV followed by SG1 (default; the standard pipeline used by OSSL pretrained models for Vis-NIR).

If `prospectr` is available, we use `prospectr::standardNormalVariate` and `prospectr::savitzkyGolay` (Rcpp implementation, faster and supports arbitrary window/polynomial). The native fallback uses the classical 5-point first-derivative coefficients `(-2, -1, 0, 1, 2) / 10`, which is the closed-form Savitzky-Golay solution for window 5 / polynomial 2 / derivative 1.

Value

A numeric matrix. Column names (wavelengths) are preserved where possible; SG trimming drops `(w - 1) / 2` columns from each edge.

References

Savitzky, A., & Golay, M. J. E. (1964). Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8), 1627–1639.

Barnes, R. J., Dhanoa, M. S., & Lister, S. J. (1989). Standard Normal Variate transformation and de-trending of near-infrared diffuse reflectance spectra. *Applied Spectroscopy*, 43(5), 772–777.

Stevens, A., & Ramirez-Lopez, L. (2024). *prospectr*: Misc. functions for processing and sample selection of spectroscopic data. R package version 0.2.7.

Examples

```
set.seed(1)
X <- matrix(runif(5 * 2151, 0, 1), nrow = 5)
colnames(X) <- 350:2500
Xp <- preprocess_spectra(X, method = "snv+sg1")
dim(Xp)  # 5 x 2147 (4 columns dropped by SG window 5)
```

`pretic` *Pretic horizon (WRB 2022): "Amazonian Dark Earth" (terra preta de indio) horizon – thick anthropogenic surface with high P, SOC, and incorporated charcoal / pottery.*

Description

Pretic horizon (WRB 2022): "Amazonian Dark Earth" (terra preta de indio) horizon – thick anthropogenic surface with high P, SOC, and incorporated charcoal / pottery.

Usage

```
pretic(pedon, min_thickness = 20, min_oc = 1.5, min_p_mehlich3 = 30)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Numeric threshold or option (see Details).
`min_oc`	Numeric threshold or option (see Details).
`min_p_mehlich3`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`print.soilKey_pls_model` *Print method for soilKey_pls_model*

Description

Print method for soilKey_pls_model

Usage

```
## S3 method for class 'soilKey_pls_model'
print(x, ...)
```

Arguments

`x`	A `soilKey_pls_model` object.
`...`	Reserved.

Value

The object, invisibly.

`prior_consistency_check` *Check consistency between a deterministic RSG assignment and a spatial prior*

Description

Returns a list describing whether the assigned RSG is plausible under the given prior. The deterministic classification is never overridden – this is purely a sanity-check signal.

Usage

```
prior_consistency_check(rsg_code, prior, threshold = 0.01)
```

Arguments

`rsg_code`	Two-letter RSG code (e.g. `"FR"`). Either the `rsg_or_order` from a `ClassificationResult` (in which case it must be the RSG name; we try to translate via the trace) or the bare code from a key trace entry.
`prior`	A spatial-prior data.table from `spatial_prior`.
`threshold`	Probability below which an assignment is flagged inconsistent (default 0.01).

Value

A list with elements:

- `consistent`: `TRUE` / `FALSE` / `NA`.
- `p`: probability of the assigned RSG in the prior (or `NA_real_` if not found).
- `threshold`: the threshold used.
- `status`: a short status string – `"consistent"`, `"inconsistent"`, or `"no_data"`.
- `note`: human-readable explanation.
- `top_prior`: `data.table` with the top three classes from the prior (for messages).

`protocalcic_properties` *Protocalcic properties (WRB 2022 Ch 3.2.8)*

Description

Visible secondary carbonate accumulations, less than the calcic gate. Detects via caco3_pct between 0.5 and the calcic threshold (15) AND designation effervescence pattern (k).

Usage

```
protocalcic_properties(pedon, min_caco3_pct = 0.5, max_caco3_pct = 15)
```

Arguments

`pedon`	A `PedonRecord`.
`min_caco3_pct`	Numeric threshold or option (see Details).
`max_caco3_pct`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`protogypsic_properties`

Protogypsic properties (WRB 2022 Ch 3.2.9): visible secondary gypsum \>= 1% but below the gypsic gate.

Description

Protogypsic properties (WRB 2022 Ch 3.2.9): visible secondary gypsum \>= 1% but below the gypsic gate.

Usage

```
protogypsic_properties(pedon, min_caso4_pct = 1, max_caso4_pct = 5)
```

Arguments

`pedon`	A `PedonRecord`.
`min_caso4_pct`	Numeric threshold or option (see Details).
`max_caso4_pct`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`protovertic` *Protovertic horizon (WRB 2022 Ch 3.1)*

Description

A weakly developed vertic horizon – the swelling/shrinking machinery is present but does not reach the full vertic intensity (cracks too narrow, or slickensides only "few", or thickness too small). Used by the Protovertic qualifier; relevant for soils that would be Vertisols if the cracks/slickensides were a notch stronger.

Usage

```
protovertic(pedon, min_clay = 30, min_thickness = 15)
```

Arguments

`pedon`	A `PedonRecord`.
`min_clay`	Numeric threshold or option (see Details).
`min_thickness`	Numeric threshold or option (see Details).

Details

v0.3.5 detection: clay \>= 30% AND any positive vertic evidence (slickensides at \>= "few" OR cracks_width_cm \>= 0.2 OR a wedge/lenticular structure_type) AND thickness \>= 15 cm. The positive cases that pass the strict `vertic_horizon` test are explicitly excluded so the two diagnostics partition the vertic-spectrum cleanly.

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`read_febr_pedons` *Load FEBR datasets as a list of PedonRecord objects*

Description

Wraps `febr::readFEBR()` (CRAN package, FEBR v1.9.9+ recommended) and adapts the returned `camada` (layer) + `observacao` tables to the soilKey schema. Auto-detects Munsell columns across the ~6 distinct conventions found in the 200 FEBR datasets that carry color data, parses PT-BR Munsell strings (`"2,5YR 3/6"`) and converts FEBR's standard units to soilKey conventions.

Usage

```
read_febr_pedons(
  dataset_codes = c("ctb0039"),
  febr_repo = NULL,
  min_munsell_coverage = 0,
  verbose = TRUE
)
```

Arguments

`dataset_codes`	Character vector of FEBR dataset IDs (e.g. `c("ctb0032", "ctb0562")`). Pass `"all"` to download every Munsell-bearing dataset; this is heavy (network calls per dataset). Default: a small curated sample for development.
`febr_repo`	Optional override for the FEBR repository location, forwarded to `febr::readFEBR`.
`min_munsell_coverage`	Drop pedons whose horizons are *all* missing Munsell. Default 0 (keep all); set to 0.5 to keep only pedons with at least 50 horizons having a Munsell hue.
`verbose`	If `TRUE` (default), prints per-dataset join statistics.

Details

Per the May 2026 scan, ~80 `febr_index_munsell` to get the curated list of Munsell-bearing dataset IDs.

Value

A list of `PedonRecord` objects with `site$id` = FEBR `observacao_id`, `site$reference_sibcs` = the surveyor's classification when available, and one horizon per FEBR camada row.

See Also

`febr_index_munsell`, `load_bdsolos_csv`.

Examples

```
## Not run:
# Single dataset (35 perfis, 100% Munsell coverage)
pedons <- read_febr_pedons("ctb0039")

# Multiple datasets
pedons <- read_febr_pedons(c("ctb0032", "ctb0562", "ctb0568"))

# All Munsell-bearing datasets (slow; 200 datasets, ~36k horizons)
all_pedons <- read_febr_pedons("all")

## End(Not run)
```

`read_pedon_csv` *Read a horizon spreadsheet (CSV/TSV) into a PedonRecord*

Description

The everyday entry point for anyone who has a soil profile in a spreadsheet: one row per horizon, one column per attribute, using soilKey's canonical column names (`top_cm`, `bottom_cm`, `designation`, `clay_pct`, `ph_h2o`, `munsell_hue_moist`, ...). The full list is `names(horizon_column_spec())`; the quickest start is the bundled template `system.file("extdata", "perfil_exemplo.csv", package = "soilKey")`.

Usage

```
read_pedon_csv(file, site = NULL, sep = "auto")
```

Arguments

`file`	Path to a `.csv` (comma) or `.tsv` (tab) file, one row per horizon.
`site`	Optional named list of site metadata (see `PedonRecord`). Defaults to `list(id = <file base name>)`.
`sep`	Field separator. `"auto"` (default) uses a tab for `.tsv` files and a comma otherwise.

Details

Only recognised columns are used (any extras are carried through untouched), so a messy export still works - soilKey simply uses the columns it understands. Site metadata (id, lat/lon, soil moisture / temperature regime) is optional and passed via `site`; without it the profile still classifies, just with less specificity where site data would have refined the name.

Value

A `PedonRecord`.

See Also

`classify_csv` to go straight from a file to the three classifications; `classify_all`; `PedonRecord`.

Examples

```
f <- system.file("extdata", "perfil_exemplo.csv", package = "soilKey")
pedon <- read_pedon_csv(f)
classify_all(pedon)$summary
```

`read_spectral_library` *Read a Vis-NIR / MIR reflectance + lab table into an OSSL-shaped library*

Description

Turns an arbitrary spectral dataset (e.g. a Brazilian Vis-NIR/MIR library) into the canonical `list(Xr, Yr, metadata)` object consumed by `fill_from_spectra` and `classify_by_spectral_neighbours`. Column names are mapped to the package's canonical attributes (clay_pct, sand_pct, ..., and the taxonomic label columns `wrb_rsg` / `sibcs_ordem` / `usda_order`) via a built-in alias table (including Portuguese headers such as *argila* / *silte* / *carbono*) or an explicit `property_map` / `label_map`.

Usage

```
read_spectral_library(
  reflectance,
  metadata,
  id_col = "id",
  wavelengths = NULL,
  resample_to = NULL,
  property_map = NULL,
  label_map = NULL,
  normalize = c("auto", "none", "percent"),
  verbose = TRUE
)
```

Arguments

`reflectance`	Reflectance data: a matrix / data.frame with rows = samples and columns named by wavelength (nm); OR a long data.frame with `id_col`, `wavelength_nm`, `reflectance`; OR a path to a CSV in either form.
`metadata`	A data.frame with one row per sample carrying `id_col` plus lab attributes and optional taxonomic labels and `lat`/ `lon`. Rows are aligned to `reflectance` by `id_col`.
`id_col`	Sample identifier column shared by both tables (default `"id"`).
`wavelengths`	Optional explicit wavelength vector (nm) when the reflectance columns are not wavelength-named.
`resample_to`	Optional target wavelength grid (nm) to linearly resample every spectrum onto (e.g. `350:2500`); default keeps the native grid.
`property_map, label_map`	Optional named lists overriding the alias auto-detection, e.g. `property_map = list(clay_pct = "ARGILA")`.
`normalize`	One of `"auto"` (divide by 100 when values look like percent), `"percent"`, or `"none"`.
`verbose`	Print a one-line summary (default `TRUE`).

Value

A list with `Xr` (numeric reflectance matrix), `Yr` (data frame of mapped properties + labels + `lat/lon`), and `metadata` (provenance). Ready to pass as `ossl_library=`.

See Also

`pedons_from_spectral_table`, `benchmark_spectral_fill`, `fill_from_spectra`

`reducing_conditions` *Reducing conditions (WRB 2022 Ch 3.2.10) – per-pedon test wrapping* `test_reducing_conditions`.

Description

Reducing conditions (WRB 2022 Ch 3.2.10) – per-pedon test wrapping `test_reducing_conditions`.

Usage

```
reducing_conditions(pedon, min_redox_pct = 5)
```

Arguments

`pedon`	A `PedonRecord`.
`min_redox_pct`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`report` *Render a soilKey classification report*

Description

Produces a pedologist-facing report from one or more `ClassificationResult` objects, optionally including the source `PedonRecord`. The HTML output is fully self-contained (single file, inline CSS); the PDF output goes through `rmarkdown::render()` and therefore requires a working LaTeX install (or one of the alternative engines accepted by `rmarkdown`).

Usage

```
report(
  x,
  file,
  format = c("auto", "html", "pdf"),
  pedon = NULL,
  title = NULL,
  include_family = FALSE,
  specifiers = FALSE,
  lang = c("en", "pt"),
  ...
)
```

Arguments

`x`	A `ClassificationResult`, a list of `ClassificationResults`, or a `PedonRecord` (in which case all three keys are run automatically).
`file`	Output path. The format is inferred from the extension (`.html` or `.pdf`) unless `format` is given explicitly.
`format`	One of `"auto"`, `"html"`, `"pdf"`.
`pedon`	Optional `PedonRecord`; when provided, its horizons table and provenance log are included.
`title`	Optional report title.
`include_family`	When `x` is a `PedonRecord` (so the three keys are run here), passes through to `classify_usda` to append the USDA family (5th category) to the subgroup. Default `FALSE` keeps the output byte-identical to earlier versions.
`specifiers`	When `x` is a `PedonRecord`, passes through to `classify_wrb2022` to attach WRB depth specifiers (Epi-/Endo-/...) to depth-anchored qualifiers. Default `FALSE`. Both flags are ignored when `x` is already a (list of) `ClassificationResult`.
`lang`	Report language; `"en"` (default) or `"pt"` (Brazilian Portuguese).
`...`	Passed to method-specific renderers.

Details

This is an S3 generic with methods for `ClassificationResult`, `list`, and `PedonRecord`. Most users call `report()` directly with a list of three results (`list(classify_wrb2022(p), classify_sibcs(p), classify_usda(p))`) to get a cross-system one-pager.

Value

The output path, invisibly.

Examples

```
pedon <- make_ferralsol_canonical()
out <- file.path(tempdir(), "soilkey_report.html")
report(pedon, file = out, pedon = pedon)
file.exists(out)
```

`report_html` *Render a soilKey classification report as self-contained HTML*

Description

See `report` for the generic. This function writes a single-file HTML report with inline CSS (no external network requests, no ‘htmltools‘ dependency) so it can be emailed or archived as-is.

Usage

```
report_html(
  x,
  file,
  pedon = NULL,
  title = NULL,
  include_family = FALSE,
  specifiers = FALSE,
  lang = c("en", "pt"),
  ...
)
```

Arguments

`x`	A `ClassificationResult`, list of results, or `PedonRecord`.
`file`	Output `.html` path.
`pedon`	Optional `PedonRecord`.
`title`	Report title.
`include_family, specifiers`	Passed through to the keys when `x` is a `PedonRecord`; see `report`.
`lang`	Report language; `"en"` (default) or `"pt"` (Brazilian Portuguese).
`...`	Currently unused.

Value

The output path, invisibly.

`report_pdf` *Render a soilKey classification report as PDF*

Description

See `report` for the generic dispatcher. This function assembles a temporary '.Rmd' file with the same content as `report_html` (site, cross-system summary, classification cards, horizons, provenance) and renders it via `rmarkdown::render()`.

Usage

```
report_pdf(
  x,
  file,
  pedon = NULL,
  title = NULL,
  include_family = FALSE,
  specifiers = FALSE,
  lang = c("en", "pt"),
  ...
)
```

Arguments

`x`	A `ClassificationResult`, list of results, or `PedonRecord`.
`file`	Output `.pdf` path.
`pedon`	Optional `PedonRecord`.
`title`	Report title.
`include_family, specifiers`	Passed through to the keys when `x` is a `PedonRecord`; see `report`.
`lang`	Report language, `"en"` (default) or `"pt"` (Brazilian Portuguese).
`...`	Passed to `rmarkdown::render()`.

Value

The output path, invisibly.

`report_to_qgis` *Export a classification result + pedon to a QGIS GeoPackage*

Description

Writes a single GeoPackage (`.gpkg`) that QGIS reads natively, containing one POINT layer (the profile location with all classification metadata as attributes) plus two attribute-only tables (the horizons schema and the provenance log). Lets a pedologist overlay the soilKey result on a soil-survey base map or join it with field-campaign vector data without writing R or SQL.

Usage

```
report_to_qgis(
  pedon,
  classifications,
  file,
  report_html = NULL,
  overwrite = TRUE
)
```

Arguments

`pedon`	A `PedonRecord`.
`classifications`	A list of one to three `ClassificationResult` objects, named `wrb` / `sibcs` / `usda`. Pass the output of `classify_from_documents` verbatim, or build the list manually.
`file`	Output path (`.gpkg`). Created with parents.
`report_html`	Optional path to a sibling HTML report (rendered via `report_html`) – stored in the `report_html` attribute of `pedon_point` so QGIS users can launch the report from the feature pop-up.
`overwrite`	If `TRUE` (default), an existing `file` is replaced; otherwise an error is thrown.

Value

The output `file` path, invisibly. Side-effect: writes a multi-layer GeoPackage.

Geometry handling

The point geometry uses the pedon's site CRS (`pedon$site$crs`, default EPSG:4326). When the site has no coordinates, the function still writes the two attribute tables but skips the point layer and emits a warning.

Layer schema

- `pedon_point` site_id, country, year, lat, lon, crs, wrb_name, wrb_rsg, wrb_grade, wrb_principal, wrb_supplementary, sibcs_name, sibcs_ordem, sibcs_grade, usda_name, usda_order, usda_grade, n_horizons, report_html (relative path), generated_at.
- `horizons_table` site_id, horizon_idx, top_cm, bottom_cm, designation, plus the canonical `horizon_column_spec()` attributes when present.
- `provenance_log` site_id, horizon_idx, attribute, source, confidence, notes.

See Also

`report` for HTML / PDF reports; `classify_from_documents` for the high-level one-liner that produces compatible `classifications`.

Examples

```
## Not run:
pedon <- make_ferralsol_canonical()
results <- list(
  wrb   = classify_wrb2022(pedon, on_missing = "silent"),
  sibcs = classify_sibcs(pedon, include_familia = TRUE),
  usda  = classify_usda(pedon)
)
report_to_qgis(pedon, results,
               file        = "perfil_042.gpkg",
               report_html = "perfil_042.html")
# In QGIS: Layer -> Add Layer -> Add Vector Layer -> perfil_042.gpkg

## End(Not run)
```

`resolve_wrb_qualifiers` *Resolve WRB 2022 qualifiers for a Reference Soil Group*

Description

Walks the YAML qualifier list for a given RSG code and tests every principal / supplementary qualifier against the pedon. Returns the resolved canonical name pieces (principal + supplementary) plus a per-qualifier trace.

Usage

```
resolve_wrb_qualifiers(pedon, rsg_code, rules = NULL, specifiers = FALSE)
```

Arguments

`pedon`	A `PedonRecord`.
`rsg_code`	Two-letter RSG code (e.g. `"FR"` for Ferralsols).
`rules`	Optional pre-loaded rules list (saves I/O when many RSGs are tested).
`specifiers`	If `TRUE`, auto-attach WRB Ch 5 depth specifiers (Epi-/Endo-/Bathy-/Amphi-/Panto-/Kato-) to depth-anchored qualifiers based on the feature's actual depth. Default `FALSE` leaves names byte-identical to earlier versions.

Value

A list with `principal` (character vector), `supplementary` (character vector), `trace`, and `trace_supplementary`.

`retic_properties` *Retic properties (WRB 2022)*

Description

Tests whether any horizon designation indicates retic features (glossic tongues of bleached material penetrating into a clay- enriched horizon). v0.3 detects these via designation pattern matching `"glossic|retic|albeluvic"` (case-insensitive). Diagnostic of Retisols.

Usage

```
retic_properties(pedon, pattern = "glossic|retic|albeluvic")
```

Arguments

`pedon`	A `PedonRecord`.
`pattern`	Regex (default `"glossic|retic|albeluvic"`).

Value

A `DiagnosticResult`.

References

IUSS Working Group WRB (2022), Chapter 5, Retisols.

`run_all_benchmarks` *Run the full soilKey benchmark suite and (optionally) write a report*

Description

Auto-detects which reference datasets are available locally, runs each via `benchmark_unified`, adds the offline canonical sanity row and the AfSP sample when present, and returns a tidy accuracy summary. When `report_path` is given, a consolidated Markdown report is written.

Usage

```
run_all_benchmarks(
  datasets = "auto",
  paths = NULL,
  max_n = 300L,
  level = "order",
  report_path = NULL,
  verbose = TRUE
)
```

Arguments

`datasets`	`"auto"` (default) detects available datasets; otherwise any subset of `c("bdsolos", "febr", "kssl","lucas_esdb", "redape")`, the literal `"canonical"` (only the fixture sanity row), or `"all"` (every dataset regardless of availability – absent ones are skipped).
`paths`	Named list of dataset paths (see `benchmark_unified`). `NULL` uses the package defaults (override the root via `options(soilKey.benchmark_root = "...")`).
`max_n`	Cap on pedons per dataset (keeps the run fast). Default 300.
`level`	Comparison level forwarded where supported (currently the suite reports at `"order"` / top level).
`report_path`	File to write the Markdown report to, `TRUE` to auto-name one under `inst/benchmarks/reports/`, or `NULL` (default) for no file.
`verbose`	Print progress.

Value

Invisibly, a list with `summary` (data.frame: dataset, system, n_compared, accuracy), `per_system` (pooled), `raw` (full `benchmark_unified` output), `weak` (zero-recall classes) and `config`.

See Also

`benchmark_unified`, `benchmark_redape`.

Examples

```
## Not run:
res <- run_all_benchmarks(max_n = 250,
                          report_path = TRUE)
res$summary

## End(Not run)
```

`run_classify_app` *Launch the soilKey interactive classification Shiny app*

Description

Opens a local Shiny app ("Pro") that drives the soilKey pipeline from a browser – no R code required: build a pedon from a canonical fixture, a CSV upload, or an interactive horizon editor; classify under WRB 2022 / SiBCS 5 / USDA ST 13 with the full key trace; run VLM photo extraction, OSSL spectral gap-fill, the SoilGrids spatial prior, an interactive **leaflet** map that queries the class prior at a clicked point, and a Monte-Carlo robustness analysis; and download a cross-system HTML or PDF report. The interface is bilingual (English / Portuguese; see `lang`).

Usage

```
run_classify_app(
  ui = c("pro", "classic"),
  lang = c("en", "pt"),
  port = NULL,
  launch.browser = TRUE,
  ...
)
```

Arguments

`ui`	Kept for back-compatibility. `"pro"` (default) launches the professional multi-tab app. `"classic"` – the original single-page uploader – was **retired in v0.9.117**; passing it now emits a deprecation warning and launches the Pro app instead.
`lang`	Initial interface language: `"en"` (default) or `"pt"` (Brazilian Portuguese). Can also be switched live from the app's navbar.
`port`	Port for the local server. Default lets Shiny choose.
`launch.browser`	Whether to open the app in the default browser (default `TRUE`).
`...`	Additional arguments passed to `runApp`.

Details

Needs the optional packages **bslib**, **shinyWidgets**, **plotly** and **leaflet** (all in `Suggests`); the function raises a clear, copy-pasteable error if any are missing.

Value

Invisibly the value returned by `shiny::runApp()`.

Examples

```
## Not run:
run_classify_app()             # professional multi-tab app (English)
run_classify_app(lang = "pt")   # interface em portugues

## End(Not run)
```

`run_demo` *Launch the soilKey Shiny demo (one-screen GUI)*

Description

Opens a Shiny app that lets a non-coder pick one of the 31 canonical profiles or upload a small horizons CSV, click **Classify**, and read the WRB / SiBCS / USDA names plus the deterministic key trace and the evidence grade. Useful for live demos, classroom teaching, and for pedologists who want to verify the package on a profile they already know without writing R code.

Usage

```
run_demo(...)
```

Arguments

`...`	Forwarded to `shiny::runApp()` (e.g. `port = 4321`, `launch.browser = FALSE`, `host = "0.0.0.0"`).

Details

Requires the `shiny` package. The taxonomic key is still deterministic: no VLM is invoked from the GUI.

Value

Invisibly, the value returned by `shiny::runApp()`.

Examples

```
## Not run:
  soilKey::run_demo()

## End(Not run)
```

`run_sibcs_grande_grupo` *Resolve o grande grupo (3o nivel) de um pedon classificado em uma subordem SiBCS*

Description

v0.7.3: itera os Grandes Grupos da subordem em ordem canonica via o engine generico `run_taxa_list`; a primeira test-block que passa captura o perfil. Os Grandes Grupos sao carregados de `inst/rules/sibcs5/grandes-grupo` (split por ordem) e mergeados pelo `load_rules`.

Usage

```
run_sibcs_grande_grupo(pedon, subordem_code, rules = NULL)
```

Arguments

`pedon`	A `PedonRecord`.
`subordem_code`	Codigo da subordem (e.g. "OJ" para Organossolos Tiomorficos).
`rules`	Lista de regras carregada via `load_rules`.

Details

Quando a subordem nao tem bloco de Grandes Grupos definido (ainda nao wirado para todas as ordens), retorna `list(assigned = NULL, trace = list())` – comportamento nao-fatal que permite `classify_sibcs` parar no 2o nivel sem erro.

Value

Lista com `assigned` (entrada YAML do Grande Grupo ou `NULL`) e `trace`.

`run_sibcs_key` *Roda a chave SiBCS 5a edicao sobre um pedon*

Description

Roda a chave SiBCS 5a edicao sobre um pedon

Usage

```
run_sibcs_key(pedon, rules = NULL)
```

Arguments

`pedon`	A `PedonRecord`.
`rules`	Conjunto de regras pre-carregado; se NULL, le `inst/rules/sibcs5/key.yaml`.

Value

Lista com `assigned` (entrada YAML da ordem atribuida) e `trace`.

`run_sibcs_subgrupo` *Resolve o subgrupo (4o nivel) de um pedon classificado em um Grande Grupo SiBCS*

Description

v0.7.3.B: itera os Subgrupos do Grande Grupo em ordem canonica via o engine generico `run_taxa_list`; a primeira test-block que passa captura o perfil. Os Subgrupos sao carregados de `inst/rules/sibcs5/subgrupos/<ordem>.` (split por ordem) e mergeados pelo `load_rules`.

Usage

```
run_sibcs_subgrupo(pedon, gg_code, rules = NULL)
```

Arguments

`pedon`	A `PedonRecord`.
`gg_code`	Codigo do Grande Grupo (e.g. "OJF" para Organossolos Tiomorficos Fibricos).
`rules`	Lista de regras carregada via `load_rules`.

Details

Em contraste com o 3o nivel (Grandes Grupos de Organossolos), Subgrupos de Cap 14 SEMPRE tem catch-all `tests:{default:true}` como ultima entrada de cada lista (subgrupo "tipico"), entao a classificacao sempre desce ao 4o nivel quando o GG foi resolvido.

Value

Lista com `assigned` (entrada YAML do Subgrupo ou `NULL`) e `trace`.

`run_sibcs_subordem` *Resolve a subordem de um pedon ja classificado em uma ordem SiBCS*

Description

Itera as subordens da ordem em ordem canonica via o engine generico `run_taxa_list`; a primeira cuja test-block passa captura o perfil. Se nenhuma passar, retorna a ultima subordem (catch-all `tests:{default:true}`).

Usage

```
run_sibcs_subordem(pedon, ordem_code, rules = NULL)
```

Arguments

`pedon`	A `PedonRecord`.
`ordem_code`	Codigo de uma letra da ordem (e.g. "L" para Latossolos).
`rules`	Lista de regras carregada via `load_rules`.

Value

Lista com `assigned` (entrada YAML da subordem ou `NULL` se a ordem nao tiver bloco) e `trace`.

`run_taxa_list` *Iterate a flat taxa list and evaluate tests in canonical order*

Description

Internal iterator extracted from `run_taxonomic_key` so nested categorical levels (subordens, grandes grupos, subgrupos, familias) can be iterated directly, without going through the `rules[[level_key]]` indirection that only makes sense at the top level.

Usage

```
run_taxa_list(pedon, taxa)
```

Arguments

`pedon`	A `PedonRecord`.
`taxa`	A list of taxon entries; each entry must have `code`, `name`, and `tests` fields, where `tests` is a block parseable by `evaluate_rsg_tests`.

Details

Behavioural note: when `taxa` is empty or `NULL`, returns `list(assigned = NULL, trace = list())` – a sub-level lookup with no canonical entries is non-fatal. The top-level `run_taxonomic_key` keeps the stricter "missing list is an error" semantics by guarding before calling this helper.

Value

A list with `assigned` (the entry of the assigned taxon, or NULL when `taxa` was empty) and `trace`.

`run_taxonomic_key` *Run a taxonomic key (system-agnostic engine)*

Description

Iterates over the taxa list at `rules[[level_key]]` in canonical order; the first taxon whose tests pass is assigned. `evaluate_rsg_tests` is reused as the per-taxon evaluator regardless of system – the test combinator semantics (`all_of` / `any_of` / `default` / `not_implemented_v01`) are the same in all three systems.

Usage

```
run_taxonomic_key(pedon, rules, level_key)
```

Arguments

`pedon`	A `PedonRecord`.
`rules`	A parsed rule set (output of `load_rules`).
`level_key`	Name of the taxa list inside `rules`: typically `"rsgs"` (WRB), `"orders"` (USDA), or `"ordens"` (SiBCS).

Details

Used at the TOP level (RSG / Order / Ordem). For nested categorical levels (subordens, grandes grupos, subgrupos, familias) iterate the flat taxa list directly via `run_taxa_list`.

Value

A list with `assigned` (the YAML entry of the assigned taxon) and `trace` (one entry per taxon tested).

`run_usda_great_group` *Run the USDA Great Group key for a given Suborder*

Description

Run the USDA Great Group key for a given Suborder

Usage

```
run_usda_great_group(pedon, suborder_code, rules = NULL)
```

Arguments

`pedon`	A `PedonRecord`.
`suborder_code`	The Suborder code (e.g. "AA" for Histels).
`rules`	Optional pre-loaded rule set.

Value

A list with `assigned` and `trace`; assigned is NULL if the Suborder has no great-groups YAML.

`run_usda_key` *Run the USDA Soil Taxonomy Order key over a pedon*

Description

Run the USDA Soil Taxonomy Order key over a pedon

Usage

```
run_usda_key(pedon, rules = NULL)
```

Arguments

`pedon`	A `PedonRecord`.
`rules`	Optional pre-loaded rule set; if NULL, reads `inst/rules/usda/key.yaml`.

Value

A list with `assigned` (the YAML entry of the assigned Order) and `trace`.

`run_usda_subgroup` *Run the USDA Subgroup key for a given Great Group*

Description

Run the USDA Subgroup key for a given Great Group

Usage

```
run_usda_subgroup(pedon, great_group_code, rules = NULL)
```

Arguments

`pedon`	A `PedonRecord`.
`great_group_code`	The Great Group code (e.g. "AAA" for Folistels).
`rules`	Optional pre-loaded rule set.

Value

A list with `assigned` and `trace`; assigned is NULL if the Great Group has no subgroups YAML.

`run_usda_suborder` *Run the USDA Suborder key for a given Order*

Description

Run the USDA Suborder key for a given Order

Usage

```
run_usda_suborder(pedon, order_code, rules = NULL)
```

Arguments

`pedon`	A `PedonRecord`.
`order_code`	The Order code (e.g. "GE" for Gelisols).
`rules`	Optional pre-loaded rule set.

Value

A list with `assigned` and `trace`; assigned is NULL if the Order has no suborders YAML.

`run_wrb_key` *Run the WRB 2022 key over a pedon*

Description

Iterates over the RSGs in canonical key order; the first RSG whose tests pass is assigned. RSGs whose tests return NA (stubbed diagnostics or insufficient data) are skipped and recorded in the trace.

Usage

```
run_wrb_key(pedon, rules = NULL)
```

Arguments

`pedon`	A `PedonRecord`.
`rules`	Optional pre-loaded rule set; if NULL, reads `inst/rules/wrb2022/key.yaml`.

Value

A list with `assigned` (the YAML entry for the assigned RSG) and `trace` (one entry per RSG tested, in order).

`salic` *Salic horizon (WRB 2022)*

Description

Tests whether any horizon meets the salic horizon criteria. The salic horizon is a horizon of soluble-salt accumulation, diagnostic for Solonchaks.

Usage

```
salic(
  pedon,
  min_thickness = 15,
  min_ec_dS_m = 15,
  alkaline_min_ec_dS_m = 8,
  alkaline_min_pH = 8.5,
  min_product = 450,
  alkaline_min_product = 240
)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Minimum thickness in cm (default 15).
`min_ec_dS_m`	Primary EC threshold (default 15 dS/m at 25C).
`alkaline_min_ec_dS_m`	Alkaline-path EC threshold (default 8 dS/m, used when pH(H2O) \>= `alkaline_min_pH`).
`alkaline_min_pH`	Required pH(H2O) for alkaline path (default 8.5).
`min_product`	Primary path product (EC * thickness in dS/m * cm) threshold (default 450 per WRB 2022).
`alkaline_min_product`	Alkaline-path product threshold (default 240).

Details

Sub-tests called:

- `test_ec_concentration` – EC \>= 15 dS/m (primary) OR (EC \>= 8 dS/m AND pH(H2O) \>= 8.5) (alkaline).
- `test_minimum_thickness` – thickness \>= 15 cm.
- `test_salic_product` – EC * thickness product \>= 450 (primary) or \>= 240 (alkaline) per qualifying layer.

v0.3.1: alkaline-path and product test added (WRB 2022 Ch 3.1.20, p. 49). Earlier versions only enforced the primary EC + thickness gate.

Value

A `DiagnosticResult`.

References

IUSS Working Group WRB (2022). *World Reference Base for Soil Resources*, 4th edition. International Union of Soil Sciences, Vienna. Chapter 3.1.20 – Salic horizon (p. 49).

`saprico` *Material organico saprico (SiBCS Cap 14)*

Description

Material organico altamente decomposto: < 17% de fibras esfregadas OU indice de von Post H7-H10. Discrimina Organossolos Sapricos no 3o nivel categorico.

Usage

```
saprico(pedon)
```

Arguments

`pedon`	A `PedonRecord`.

Value

`DiagnosticResult`.

References

Embrapa (2018), SiBCS 5a ed., Cap 14 (Organossolos), pp 224-226.

`save_ossl_models` *Save / load trained OSSL-backed PLSR models*

Description

Thin wrappers around `saveRDS` / `readRDS` that also verify the deserialised object's shape. The on-disk file carries the soilKey version, training time and preprocess label as attributes; `load_ossl_models` preserves them.

Usage

```
save_ossl_models(models, path)

load_ossl_models(path)
```

Arguments

`models`	Output of `train_pls_from_ossl`.
`path`	File path. Use `.rds` or `.RData` as the suffix (saveRDS is used regardless).

Value

`save_ossl_models()` returns `path` invisibly. `load_ossl_models()` returns the model list.

`shrink_swell_cracks` *Shrink-swell cracks (WRB 2022 Ch 3.2.12) – per-pedon test wrapping* `test_shrink_swell_cracks`*.*

Description

Shrink-swell cracks (WRB 2022 Ch 3.2.12) – per-pedon test wrapping `test_shrink_swell_cracks`.

Usage

```
shrink_swell_cracks(pedon, min_width_cm = 0.5)
```

Arguments

`pedon`	A `PedonRecord`.
`min_width_cm`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`sideralic_properties` *Sideralic properties (WRB 2022 Ch 3.2.13)*

Description

Mineral material with a relatively low CEC. WRB 2022 (3.2.13) requires BOTH:

1. one or both of: clay >= 8% AND CEC/clay < 24 cmol_c/kg clay; OR bulk CEC < 2 cmol_c/kg soil;
2. evidence of soil formation as defined in criterion 3 of the cambic horizon (`test_cambic_soil_formation`).

Both must be met by the SAME layer. Criterion 2 was added in v0.9.127 (previously only criterion 1 was enforced); where the soil-formation evidence cannot be assessed (no Munsell/clay/Fe/carbonate adjacency data) the result is `NA` rather than a false positive.

Usage

```
sideralic_properties(pedon, max_cec_per_clay = 24, max_bulk_cec = 2)
```

Arguments

`pedon`	A `PedonRecord`.
`max_cec_per_clay`	Numeric threshold or option (see Details).
`max_bulk_cec`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`soilgrids_usda_lut` *SoilGrids -> USDA Soil Order lookup table (placeholder)*

Description

Reserved for the future SoilGrids USDA layer. Currently returns the 12 USDA Order codes mapped to integers 1..12.

Usage

```
soilgrids_usda_lut()
```

Value

Named character vector.

`soilgrids_wrb_lut` *SoilGrids -> WRB code lookup table*

Description

Maps the integer raster values used by the SoilGrids 2.0 "MostProbable WRB" layer to soilKey's two-letter RSG codes (the codes used in `inst/rules/wrb2022/key.yaml`).

Usage

```
soilgrids_wrb_lut()
```

Details

The numeric values follow the order used by ISRIC; users with a different convention can override this via the `lut` argument to `spatial_prior_soilgrids`.

Value

Named character vector: names are integer-as-character (`"1"`, `"2"`, ...), values are RSG codes.

`soil_classes_at_location` *Likely soil classes at a geographic location (spatial classification aid)*

Description

Returns a ranked list of the soil Reference Soil Groups (or SiBCS ordens, or USDA orders) most likely to occur at the given point, based on a global or regional dominant-soil raster (SoilGrids 2.0 by default). This is the **before-you-have-a-pedon helper**: a pedologist arriving in the field can call it with the GPS coordinates of the planned profile pit and see which classes are expected, plus what attributes typically distinguish them.

Usage

```
soil_classes_at_location(
  lat,
  lon,
  system = c("wrb2022", "sibcs", "usda"),
  buffer_m = 1000,
  source_url = NULL,
  top_n = 5,
  verbose = TRUE
)
```

Arguments

`lat, lon`	Numeric WGS-84 coordinates.
`system`	Classification system. One of `"wrb2022"` (default), `"sibcs"`, `"usda"`.
`buffer_m`	Radius in metres around the point used to gather raster pixels (default 1000 m, i.e. roughly 4 SoilGrids pixels).
`source_url`	Path / URL of the dominant-soil raster.
`top_n`	Keep the top N classes by probability (default 5).
`verbose`	Emit a `cli` summary.

Details

This function does **not** classify a profile. The deterministic key in `classify_wrb2022` / `classify_sibcs` / `classify_usda` remains the only thing that assigns a class from horizon data. The output here is purely informational – a "shopping list" of what to confirm.

Value

A list as described under **Output**.

Data source

For real use, point `source_url` at a regional SoilGrids "MostProbable WRB" GeoTIFF / COG (one of the cuts at `https://files.isric.org/soilgrids/latest/data/wrb/`). For tests, `options(soilKey.test_raster = "/tmp/syn.tif")` is honoured. When no source is given, the function emits a `cli_alert_warning()` and returns an empty result – it does **not** pretend to know.

Output

A list with three elements:

`distribution` A `data.table` with columns `rsg_code`, `rsg_name`, `probability`, sorted by descending probability.

`typical_attributes` A `data.table` keyed by `rsg_code` with the canonical attribute ranges that distinguish each class (clay range, CEC range, BS range, etc.). The values come from the WRB 2022 / SiBCS 5 / KST 13ed canonical thresholds, NOT from the raster.

`site` The site list passed in, plus the buffer radius and the source URL.

See Also

`spatial_prior_soilgrids` for the post-classification consistency check.

Examples

```
## Not run:
# Mata Atlântica, Rio de Janeiro state.
res <- soil_classes_at_location(
  lat        = -22.7,
  lon        = -43.7,
  system     = "wrb2022",
  source_url = "https://files.isric.org/soilgrids/latest/data/wrb/MostProbable.vrt"
)
res$distribution         # ranked list of likely RSGs
res$typical_attributes   # canonical thresholds per RSG to confirm

## End(Not run)
```

`soil_organic_carbon` *Soil organic carbon (WRB 2022 Ch 3.3.16): organic C that does NOT belong to artefacts. v0.3.3: any layer with oc_pct >= 0.1 and artefacts_industrial_pct < 35.*

Description

Soil organic carbon (WRB 2022 Ch 3.3.16): organic C that does NOT belong to artefacts. v0.3.3: any layer with oc_pct >= 0.1 and artefacts_industrial_pct < 35.

Usage

```
soil_organic_carbon(pedon, min_oc = 0.1, max_artefacts = 35)
```

Arguments

`pedon`	A `PedonRecord`.
`min_oc`	Numeric threshold or option (see Details).
`max_artefacts`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`solimovic_material` *Solimovic material (WRB 2022 Ch 3.3.17): hetero genous mass-movement material on slopes / footslopes (formerly "colluvic"). v0.3.3: detects via* `rock_origin == "colluvial"` *OR* `layer_origin == "solimovic"`.

Description

Solimovic material (WRB 2022 Ch 3.3.17): hetero genous mass-movement material on slopes / footslopes (formerly "colluvic"). v0.3.3: detects via `rock_origin == "colluvial"` OR `layer_origin == "solimovic"`.

Usage

```
solimovic_material(pedon)
```

Arguments

`pedon`	A `PedonRecord`.

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`sombric` *Sombric horizon (WRB 2022): subsurface accumulation of humus that qualified neither as spodic nor as a true mollic-like horizon (low-base-saturation cool tropical highlands). v0.3.3 detects via designation pattern + OC criteria (BS < 50, OC > 0.6, depth > 25 cm).*

Description

Sombric horizon (WRB 2022): subsurface accumulation of humus that qualified neither as spodic nor as a true mollic-like horizon (low-base-saturation cool tropical highlands). v0.3.3 detects via designation pattern + OC criteria (BS < 50, OC > 0.6, depth > 25 cm).

Usage

```
sombric(
  pedon,
  min_thickness = 15,
  min_oc = 0.6,
  max_bs = 50,
  min_top_cm = 25,
  min_oc_increase = 0.1
)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Numeric threshold or option (see Details).
`min_oc`	Numeric threshold or option (see Details).
`max_bs`	Numeric threshold or option (see Details).
`min_top_cm`	Numeric threshold or option (see Details).
`min_oc_increase`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`spatial_prior` *Spatial prior over RSGs (or Orders) at a pedon's location*

Description

Top-level dispatcher. Reads a categorical raster of soil classes (SoilGrids globally, Embrapa for Brazil), buffers the pedon's coordinates, tallies pixel classes within the buffer, and returns the empirical class frequency as a probability distribution.

Usage

```
spatial_prior(
  pedon,
  source = c("soilgrids", "embrapa"),
  system = c("wrb2022", "usda"),
  ...
)
```

Arguments

`pedon`	A `PedonRecord` with non-NULL `site$lat` / `site$lon`.
`source`	Backend to query: `"soilgrids"` (default) or `"embrapa"`.
`system`	Classification system: `"wrb2022"` (default) or `"usda"`. Embrapa source forces `"sibcs5"` internally regardless of this argument.
`...`	Passed through to the backend (`spatial_prior_soilgrids` or `spatial_prior_embrapa`).

Details

The prior is intentionally separate from the deterministic key. Pass the returned data.table to `classify_wrb2022` via the `prior` argument; the result will then carry a `prior_check` entry (consistent / inconsistent / not_run).

Value

A `data.table` with columns `rsg_code` (character) and `probability` (numeric, summing to 1). Empty if the buffer extracts no valid pixels – callers should check `nrow()`.

`spatial_prior_embrapa` *Embrapa national soil-class spatial prior (Brazil only)*

Description

v0.5 stub. Reads a user-provided categorical raster of SiBCS orders / suborders, buffers the pedon's site, tallies pixel classes, and returns a probability distribution over SiBCS codes (or, with a user-provided LUT, over WRB equivalents).

Usage

```
spatial_prior_embrapa(
  pedon,
  raster_path = NULL,
  buffer_m = 3750,
  lut = NULL,
  n_classes_top = 10,
  ...
)
```

Arguments

`pedon`	A `PedonRecord`.
`raster_path`	Required. Path to a local categorical raster (GeoTIFF) of Embrapa SiBCS classes. There is no built-in file in v0.5 – download the polygon map from `https://www.embrapa.br/solos/sibcs` and rasterise it.
`buffer_m`	Buffer radius in metres (default 3750, i.e. ~15-cell neighbourhood at 250 m resolution).
`lut`	Optional named character vector mapping raster integer values to soil-class codes. If NULL, raster categories are used as-is (terra::levels).
`n_classes_top`	Keep only the top N classes (default 10).
`...`	Reserved.

Details

Unlike SoilGrids, Embrapa does not publish per-pixel probabilities, so the empirical frequency over a neighbourhood window (default 15 x 15 cells = ~3.75 km radius at 250 m resolution) is used as an approximation.

Value

A `data.table` with columns `rsg_code`, `probability`.

`spatial_prior_soilgrids`

SoilGrids spatial prior

Description

Reads a categorical raster of dominant Reference Soil Groups around the pedon's site, buffers the point in metric coordinates, extracts all pixel values within the buffer, and returns the empirical class frequency as a probability distribution over RSG codes.

Usage

```
spatial_prior_soilgrids(
  pedon,
  system = c("wrb2022", "usda"),
  buffer_m = 250,
  source_url = NULL,
  n_classes_top = 10,
  lut = NULL,
  ...
)
```

Arguments

`pedon`	A `PedonRecord` with non-NULL `site$lat` and `site$lon`.
`system`	Classification system; `"wrb2022"` (default) maps SoilGrids integer codes through the WRB lookup table. `"usda"` is reserved for a future SoilGrids-USDA layer.
`buffer_m`	Buffer radius in metres around the point (default 250 m, i.e. one SoilGrids pixel).
`source_url`	Optional. A path or URL accepted by `terra::rast`. If NULL, falls back to `getOption("soilKey.test_raster")`.
`n_classes_top`	Keep only the top N classes by frequency (default 10). Set to `Inf` to keep all.
`lut`	Optional named integer vector mapping raster values to RSG codes. Default is `soilgrids_wrb_lut`; pass a custom one if your raster uses different codes.
`...`	Reserved for future use.

Value

A `data.table` with columns `rsg_code`, `probability`.

Data source

For real use, pass `source_url` pointing at a SoilGrids "MostProbable WRB" GeoTIFF / COG, e.g. one of the regional cuts published at `https://files.isric.org/soilgrids/latest/data/wrb/`. For tests, set `options(soilKey.test_raster = "/path/to/syn.tif")` to point at a local synthetic raster – this avoids network access in CI.

Coordinate handling

We use `sf::st_transform` when sf is available; otherwise we fall back to `terra::project` on a single-point SpatVector. The buffer is constructed in metric (UTM) coordinates so `buffer_m` is in metres regardless of the pedon CRS. The raster itself is queried in its native CRS via terra's automatic reprojection.

See Also

`spatial_prior`, `soilgrids_wrb_lut`.

`spodic` *Spodic horizon (WRB 2022)*

Description

Tests whether any horizon meets the spodic horizon criteria. The spodic horizon is an illuvial horizon with active Al + Fe oxalate- extractable material plus organic matter; diagnostic of Podzols.

Usage

```
spodic(
  pedon,
  min_thickness = 2.5,
  min_alfe = 0.5,
  max_ph = 5.9,
  min_oc_in_b = 0.5,
  engine = NULL
)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Minimum thickness in cm (default 2.5).
`min_alfe`	Minimum (Al_ox + 0.5 * Fe_ox) percent (default 0.5).
`max_ph`	Maximum ph_h2o (default 5.9).
`min_oc_in_b`	Minimum OC % in the candidate Bh / Bs layer for the v0.9.19 morphological inference path when Al / Fe oxalate are missing (default 0.5).
`engine`	One of `"soilkey"` (default; strict v0.9.19 morphological path requires Bh / Bs / Bhs designation + albic E above) or `"aqp"` (relaxed v0.9.84 path: any B* below E* with OC translocation peak). When `NULL`, reads `getOption("soilKey.diagnostic_engine")`.

Details

Sub-tests:

- `test_spodic_aluminum_iron` – (Al_ox + 0.5*Fe_ox) >= 0.5%
- `test_ph_below` – ph_h2o <= 5.9
- `test_minimum_thickness` – thickness >= 2.5 cm

v0.2 limitations: the WRB color criterion (hue 5YR or yellower with chroma <= 5, or specific dark colors) is not enforced. The (Al_ox + Fe_ox)/clay >= 0.05 alternative ratio test is not yet wired. Both deferred to v0.3.

Value

A `DiagnosticResult`.

v0.9.84 engine="aqp" relaxation

KSSL+NASIS Spodosols routinely use generic "B1" / "B2" / "Bw" designations rather than the specific Bh / Bs / Bhs that the v0.9.19 morphological-inference path requires. Of 14 KSSL+NASIS Podzol references, only 1 / 14 passes spodic via the v0.9.19 path; 7 / 14 have BOTH an E-designated albic-eligible horizon above AND an OC peak in a B horizon below (the canonical Podzol illuviation signature) but use generic B / Bw designations and so fail strict morph.

When `engine` = `"aqp"` (read from `getOption("soilKey.diagnostic_engine", "soilkey")` when `engine` is `NULL`) AND Al / Fe oxalate is unmeasured AND the v0.9.19 strict path did not fire, accept any `B*` designation below an `E*`-designated horizon when:

- `ph_h2o` <= `max_ph` in the B horizon, AND
- `oc_pct` >= `min_oc_in_b` in the B horizon, AND
- OC in the B is greater than the maximum OC in any horizon above (the translocation signature).

Default `engine` is `"soilkey"` – canonical behaviour bit-for-bit preserved.

References

IUSS Working Group WRB (2022), Chapter 3, Spodic horizon.

`stagnic_properties` *Stagnic properties (WRB 2022)*

Description

Tests for redoximorphic features driven by perched water. Distinct from gleyic (groundwater): stagnic features appear in upper layers AND redox decreases substantially with depth (the perched layer sits above a slowly permeable subsoil that itself is not saturated).

Usage

```
stagnic_properties(
  pedon,
  max_top_cm = 100,
  min_redox_pct = 5,
  decay_factor = 3
)
```

Arguments

`pedon`	A `PedonRecord`.
`max_top_cm`	Maximum top depth (cm) of candidate shallow layers (default 100).
`min_redox_pct`	Minimum redox feature percent in the shallow layer (default 5).
`decay_factor`	Required factor of redox decrease with depth (default 3, i.e., deeper redox < shallow / 3).

Value

A `DiagnosticResult`.

References

IUSS Working Group WRB (2022), Chapter 3, Stagnic properties.

`st_features_canonical` *USDA Soil Taxonomy diagnostic features canonical table*

Description

Convenience wrapper for `canonical_reference("ST_features")`. Returns an 84-row data.frame with one row per diagnostic feature (epipedon / subsurface horizon / property / material) and columns: `group`, `name`, `chapter`, `page`, `description`, `criteria`. The `criteria` column is a list-column; each element holds the parsed criteria text per feature.

Usage

```
st_features_canonical(prefer_pkg = TRUE)
```

Arguments

`prefer_pkg`	If `TRUE` (default), prefer the installed SoilTaxonomy package over the vendored copy. Set to `FALSE` to force the vendored copy (e.g. for reproducibility of a specific soilKey release).

Value

The canonical Soil Taxonomy diagnostic-features reference (a list / data.frame).

`subgrupo_planossolo_espessos`

Subgrupo "espessos" de Planossolos (B planico profundo, > 100 cm)

Description

Discrimina os Subgrupos espessos de Planossolos (Cap 15: SNs Espessos, SNo Espessos, SXs Espessos, SXal Espessos, SXd Espessos, SXe Espessos): B planico cujo topo ocorre entre `min_top_cm` (exclusivo) e `max_top_cm` (inclusivo).

Usage

```
subgrupo_planossolo_espessos(pedon, min_top_cm = 100, max_top_cm = 200)
```

Arguments

`pedon`	A `PedonRecord`.
`min_top_cm`	Profundidade minima exclusiva do topo do B planico (default 100; passa se top > 100).
`max_top_cm`	Profundidade maxima inclusiva (default 200).

Details

Implementacao: identifica B planico via `B_planico`, captura o topo (mais raso) das camadas que passam, e testa se cai em (`min_top_cm`, `max_top_cm`].

Value

`DiagnosticResult`.

References

Embrapa (2018), SiBCS 5a ed., Cap 15 (Planossolos), pp 251-260.

`subgrupo_planossolo_mesicos`

Subgrupo "mesicos" de Planossolos (B planico topo em [50, 100] cm)

Description

Discrimina os Subgrupos mesicos de Planossolos (Cap 15: SNs Mesicos, SNo Mesicos, SXs Mesicos, SXal Mesicos, SXd Mesicos, SXe Mesicos): B planico cujo topo ocorre entre `min_top_cm` (inclusivo) e `max_top_cm` (inclusivo).

Usage

```
subgrupo_planossolo_mesicos(pedon, min_top_cm = 50, max_top_cm = 100)
```

Arguments

`pedon`	A `PedonRecord`.
`min_top_cm`	Profundidade minima inclusiva (default 50).
`max_top_cm`	Profundidade maxima inclusiva (default 100).

Value

`DiagnosticResult`.

References

Embrapa (2018), SiBCS 5a ed., Cap 15 (Planossolos).

`subgrupo_plintossolo_endico_concrecionario` *Subgrupo "endico" de Plintossolos Concrecionarios (topo de horizonte concrecionario >= 40 cm)*

Description

Discrimina o Subgrupo FFcoEn (Plintossolos Petricos Concrecionarios endicos): horizonte concrecionario cujo topo ocorre a >= `min_top_cm` cm.

Usage

```
subgrupo_plintossolo_endico_concrecionario(pedon, min_top_cm = 40)
```

Arguments

`pedon`	A `PedonRecord`.
`min_top_cm`	Profundidade minima inclusiva (default 40).

Value

`DiagnosticResult`.

References

Embrapa (2018), SiBCS 5a ed., Cap 16, p 264.

`subgrupo_plintossolo_endico_litoplintico`

Subgrupo "endico" de Plintossolos Litoplinticos (topo de horizonte litoplintico >= 40 cm)

Description

Discrimina o Subgrupo FFlpEn (Plintossolos Petricos Litoplinticos endicos): horizonte litoplintico cujo topo ocorre a >= `min_top_cm` cm.

Usage

```
subgrupo_plintossolo_endico_litoplintico(pedon, min_top_cm = 40)
```

Arguments

`pedon`	A `PedonRecord`.
`min_top_cm`	Profundidade minima inclusiva (default 40).

Value

`DiagnosticResult`.

References

Embrapa (2018), SiBCS 5a ed., Cap 16, p 264.

`subgrupo_plintossolo_espessos`

Subgrupo "espessos" de Plintossolos (horizonte plintico topo > 100 cm)

Description

Discrimina os Subgrupos espessos de Plintossolos Argiluvicos (FT*Es) e Haplicos (FXacEs, FXdEs, FXeEs): horizonte plintico cujo topo ocorre entre `min_top_cm` (exclusivo) e `max_top_cm` (inclusivo).

Usage

```
subgrupo_plintossolo_espessos(pedon, min_top_cm = 100, max_top_cm = 200)
```

Arguments

`pedon`	A `PedonRecord`.
`min_top_cm`	Profundidade minima exclusiva (default 100).
`max_top_cm`	Profundidade maxima inclusiva (default 200).

Value

`DiagnosticResult`.

References

Embrapa (2018), SiBCS 5a ed., Cap 16 (Plintossolos), pp 261-272.

`takyric_properties` *Takyric properties (WRB 2022 Ch 3.2.15) – per-pedon test wrapping* `test_takyric_surface`.

Description

Takyric properties (WRB 2022 Ch 3.2.15) – per-pedon test wrapping `test_takyric_surface`.

Usage

```
takyric_properties(pedon)
```

Arguments

`pedon`	A `PedonRecord`.

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`technic_features` *Technic features (WRB 2022)*

Description

Tests for any of three WRB 2022 alternative qualifying conditions for Technosols:

1. Artefacts >= `artefacts_min_pct` (default 20%) by volume within the upper `max_top_cm` (default 100 cm).
2. A continuous geomembrane (`geomembrane_present == TRUE`) within the upper 100 cm.
3. Technic hard material (concrete, asphalt, mine spoil) with `technic_hardmaterial_pct` >= `hardmaterial_min_pct` (default 95%) at the surface (top_cm <= `hardmaterial_max_top_cm`, default 5).

Either path qualifies.

Usage

```
technic_features(
  pedon,
  artefacts_min_pct = 20,
  max_top_cm = 100,
  hardmaterial_min_pct = 95,
  hardmaterial_max_top_cm = 5
)
```

Arguments

`pedon`	A `PedonRecord`.
`artefacts_min_pct`	Minimum artefact percent (default 20).
`max_top_cm`	Maximum top depth (cm) for the artefact and geomembrane paths (default 100).
`hardmaterial_min_pct`	Minimum hard-material coverage (%) for the technic-hard-material path (default 95).
`hardmaterial_max_top_cm`	Surface depth window (cm) for the technic-hard-material path (default 5).

Value

A `DiagnosticResult`.

References

IUSS Working Group WRB (2022), Chapter 5, Technosols.

`technic_hard_material` *Technic hard material (WRB 2022 Ch 3.3.18): consolidated human-made material (asphalt, concrete, worked stones).*

Description

Technic hard material (WRB 2022 Ch 3.3.18): consolidated human-made material (asphalt, concrete, worked stones).

Usage

```
technic_hard_material(pedon)
```

Arguments

`pedon`	A `PedonRecord`.

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`tephric_material` *Tephric material (WRB 2022 Ch 3.3.19): \>= 30% volcanic glass in 0.02-2 mm fraction AND no andic / vitric properties.*

Description

Tephric material (WRB 2022 Ch 3.3.19): \>= 30% volcanic glass in 0.02-2 mm fraction AND no andic / vitric properties.

Usage

```
tephric_material(pedon, min_glass = 30)
```

Arguments

`pedon`	A `PedonRecord`.
`min_glass`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`terric` *Terric horizon (WRB 2022): topsoil thickened by long-term application of mineral material (sediment / sand additions). v0.3.3: thickness >= 20 cm + designation Au / Apc.*

Description

Terric horizon (WRB 2022): topsoil thickened by long-term application of mineral material (sediment / sand additions). v0.3.3: thickness >= 20 cm + designation Au / Apc.

Usage

```
terric(pedon, min_thickness = 20)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`texture_class_from_pct` *NRCS texture-class shorthand from clay / silt / sand percent*

Description

aqp's `getArgillicBounds()` requires an NRCS texture class column (e.g. "SCL", "C", "CL", "FS"). soilKey horizons only carry the percent fractions; this helper derives the class from the standard USDA texture triangle.

Usage

```
texture_class_from_pct(clay, silt, sand)
```

Arguments

`clay`	Numeric vector of clay percent (0-100).
`silt`	Numeric vector of silt percent.
`sand`	Numeric vector of sand percent. (clay + silt + sand should sum to ~100; mild deviations are tolerated.)

Details

Returns the standard NRCS abbreviation:

COS	Coarse sand
S	Sand
FS	Fine sand
VFS	Very fine sand
LS	Loamy sand
LFS	Loamy fine sand
SL	Sandy loam
FSL	Fine sandy loam
L	Loam
SIL	Silt loam
SI	Silt
SCL	Sandy clay loam
CL	Clay loam
SICL	Silty clay loam
SC	Sandy clay
SIC	Silty clay
C	Clay

Implementation follows the canonical USDA texture triangle; vector- ised over the input. NA in / NA out.

Value

Character vector of NRCS texture class abbreviations.

`thionic` *Thionic horizon (WRB 2022): post-oxidation acid sulfate horizon. Requires sulfidic_s_pct >= 0.01 AND pH(H2O) <= 4.*

Description

Thionic horizon (WRB 2022): post-oxidation acid sulfate horizon. Requires sulfidic_s_pct >= 0.01 AND pH(H2O) <= 4.

Usage

```
thionic(pedon, min_thickness = 15, max_pH = 4, min_sulfidic_s = 0.01)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Numeric threshold or option (see Details).
`max_pH`	Numeric threshold or option (see Details).
`min_sulfidic_s`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`train_pls_from_ossl` *Train pre-trained PLSR models from an OSSL library*

Description

Iterates over `properties` and fits one PLSR model per target against the OSSL spectra in `ossl_library$Xr`, with internal cross-validation to pick the optimal number of components per property. The returned list is a drop-in replacement for the `ossl_models` argument of `predict_ossl_pretrained` and `fill_from_spectra`.

Usage

```
train_pls_from_ossl(
  ossl_library,
 properties = c("clay_pct", "sand_pct", "silt_pct", "cec_cmol", "ph_h2o", "oc_pct"),
  ncomp_max = 20L,
  validation = c("CV", "LOO", "none"),
  segments = 10L,
  preprocess = "snv+sg1",
  min_n = 50L,
  verbose = TRUE
)
```

Arguments

`ossl_library`	A list with two named elements: `Xr` (numeric matrix of training spectra) and `Yr` (data.frame keyed by property name, one row per training spectrum). See `ossl_library_template`.
`properties`	Character vector of column names in `ossl_library$Yr` to train models for. Defaults to the six core soil properties exposed by OSSL.
`ncomp_max`	Integer. Upper bound on the number of PLS components to consider during cross-validation. Defaults to 20.
`validation`	One of `"CV"` (default, k-fold), `"LOO"` (leave-one-out, slow), `"none"` (uses `ncomp_max` components without selection).
`segments`	Number of CV segments when `validation = "CV"`. Default 10.
`preprocess`	Pre-processing label passed to `preprocess_spectra`. Stored on the trained models so `predict_from_spectra` can reapply it.
`min_n`	Minimum number of valid training samples (after dropping rows with non-finite y or X). Properties below this threshold are skipped with a warning. Default 50.
`verbose`	If `TRUE` (default), prints a per-property summary on completion.

Details

Spectra are pre-processed inside the function (default `"snv+sg1"`); the same preprocessing is used downstream by `predict_from_spectra` so the user does not have to remember which transform was applied at training time.

Value

A named list of `soilKey_pls_model` objects, one per successfully trained property. Carries `trained_at`, `soilKey_version` and `preprocess` attributes for provenance.

Examples

```
## Not run:
lib <- download_ossl_subset(region = "south_america")
models <- train_pls_from_ossl(lib,
                              properties = c("clay_pct", "ph_h2o"))
```

```
result <- predict_from_spectra(my_pedon, models = models)

## End(Not run)
```

`tsitelic` *Tsitelic horizon (WRB 2022 Ch 3.1)*

Description

From Georgian *tsiteli* = red. A red colour-defined horizon formed on weathered basalt or similar Fe-rich parent material in Caucasian / Mediterranean settings. Used by the Cambisols key (Ch 4 p 123, criterion 4) and by the Tsitelic qualifier.

Usage

```
tsitelic(pedon, min_thickness = 10)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Numeric threshold or option (see Details).

Details

Diagnostic criteria (v0.3.5 simplification):

- Munsell hue \<= 2.5YR (i.e. 2.5YR, 10R, 7.5R, 5R, 2.5R) AND value \<= 4 (moist) AND chroma \>= 4 (moist);
- evidence of soil formation (cambic-style criterion 3) proxied by clay \>= 8% AND `structure_grade` not "single grain" / "massive";
- thickness \>= 10 cm.

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`umbric_horizon` *Umbric horizon (WRB 2022)*

Description

Tests for the umbric horizon – a thick, dark, organic-rich surface horizon like mollic, but with low base saturation (< 50%). Diagnostic of Umbrisols.

Usage

```
umbric_horizon(
  pedon,
  min_thickness = 20,
  min_oc = 0.6,
  max_bs = 50,
  surface_top_cm = 5
)
```

Arguments

`pedon`	A `PedonRecord`.
`min_thickness`	Minimum thickness (cm; default 20).
`min_oc`	Minimum SOC % (default 0.6).
`max_bs`	Maximum base saturation % (default 50; profile must be BELOW this).
`surface_top_cm`	Maximum top_cm for surface-related layers (default 5).

Details

Implementation reuses every mollic sub-test except the BS test, which is inverted via `test_bs_below`.

Value

A `DiagnosticResult`.

References

IUSS Working Group WRB (2022), Chapter 3, Umbric horizon.

`usda_to_wrb_rsg` *USDA Soil Taxonomy <-> WRB Reference Soil Group correlation table*

Description

Returns the single most-common WRB RSG for a given USDA Order + optional Suborder. Based on IUSS WRB (2022) Annex 6.

Usage

```
usda_to_wrb_rsg(usda_order, usda_suborder = NULL)
```

Arguments

`usda_order`	Character vector of USDA Order names. Case- insensitive; trailing 's' stripped (e.g.\ both "Mollisols" and "Mollisol" accepted).
`usda_suborder`	Optional character vector of USDA Suborder names (case-insensitive) used to refine the mapping. Same length as `usda_order` or recycled.

Value

Character vector of WRB Reference Soil Group names (singular, no plural 's'). `NA` for unrecognised inputs.

Caveat

This is a "best-guess" cross-walk for benchmark validation only. Real-world correlation requires per-pedon evaluation of WRB diagnostic horizons. Use this function to derive a reasonable *expected* WRB classification from a USDA-classified pedon (e.g.\ from KSSL/NASIS) so that `classify_wrb2022()` can be validated against an external taxonomy on the same profiles.

References

IUSS Working Group WRB (2022). *World Reference Base for Soil Resources*, 4th edition, Annex 6. International Union of Soil Sciences, Vienna.

Examples

```
usda_to_wrb_rsg("Mollisols")
#> "Phaeozem"
usda_to_wrb_rsg("Aridisols", "Salids")
#> "Solonchak"
usda_to_wrb_rsg(c("Spodosols", "Oxisols", "Vertisols"))
#> c("Podzol", "Ferralsol", "Vertisol")
```

`validate_horizon_geometry`

Validate horizon depth geometry

Description

A pure, side-effect-free check of a horizon table's depth geometry, independent of any `PedonRecord`. The Pro app's Pedon builder calls it to give immediate feedback while horizons are edited, and it is a handy guard before constructing a profile from an untrusted CSV.

Usage

```
validate_horizon_geometry(horizons)
```

Arguments

`horizons`	A data frame with at least numeric `top_cm` and `bottom_cm` columns (and optionally a `designation` column).

Details

It reports two severities:

errors (these make a sane classification impossible) a missing or non-numeric `top_cm`/`bottom_cm`; a negative depth; a horizon whose `top_cm` >= `bottom_cm` (inverted or zero thickness); two horizons whose depths overlap.

warnings (allowed, but worth surfacing) the shallowest horizon not starting at the surface (0 cm); a gap between consecutive horizons; horizons entered out of increasing-depth order; a duplicated horizon designation.

This complements `PedonRecord$validate()`, which additionally checks chemistry (texture sums, pH, CEC vs bases, Munsell ranges); use that for a built record and this for a raw table.

Value

A list with `valid` (logical; `TRUE` when there are no errors), `errors` and `warnings` (character vectors of human-readable English messages), and `details` – a named list of the offending row indices (or values) per check, so a caller can compose its own (e.g. localised) messages.

Examples

```
h <- data.frame(top_cm = c(0, 20, 55), bottom_cm = c(20, 55, 90),
                designation = c("A", "AB", "Bt"))
validate_horizon_geometry(h)$valid          # TRUE

bad <- data.frame(top_cm = c(0, 40), bottom_cm = c(50, 30))  # overlap+inverted
validate_horizon_geometry(bad)$errors
```

`validate_pedon_json` *Validate a PedonRecord against the JSON schema*

Description

Convenience wrapper that converts a `PedonRecord` (or a compatible list) to JSON and validates it via `jsonvalidate::json_validate` against the canonical schema returned by `pedon_json_schema`.

Usage

```
validate_pedon_json(x)
```

Arguments

`x`	A `PedonRecord` or a list with the same shape.

Details

Use this BEFORE calling `classify_*` when ingesting data from external systems (web APIs, ETL pipelines, multimodal extraction) to catch schema violations early.

Value

A logical scalar (`TRUE` when valid). Validation errors appear as the `errors` attribute when `FALSE`.

Examples

```
## Not run:
p <- make_ferralsol_canonical()
validate_pedon_json(p)
#> [1] TRUE

## End(Not run)
```

`vertic_horizon` *Vertic horizon (WRB 2022 Ch 3.1)*

Description

Stricter than the vertic *properties*: the vertic *horizon* requires \>= 30% clay throughout, slickensides at \>= "common" level, AND shrink-swell cracks \>= 0.5 cm wide. Used by Vertisols. v0.9.19 adds an OR-alternative COLE-based linear-extensibility path: summed (`cole_value * thickness`) over the upper 100 cm \>= 6 cm passes the diagnostic even when slickensides + cracks are not recorded (KST 13ed Ch 16 LE alternative, p 343).

Usage

```
vertic_horizon(
  pedon,
  min_clay = 30,
  min_thickness = 25,
  min_le_cm = 6,
  le_max_depth_cm = 100,
  min_crack_width_cm = 0.5
)
```

Arguments

`pedon`	A `PedonRecord`.
`min_clay`	Numeric threshold or option (see Details).
`min_thickness`	Numeric threshold or option (see Details).
`min_le_cm`	Minimum LE sum (cm) for the COLE-based path (default 6, per KST 13ed Ch 16).
`le_max_depth_cm`	Depth window (cm) for the COLE-based path (default 100).
`min_crack_width_cm`	Minimum shrink-swell crack width (cm) for the field-crack path. Defaults to 0.5 (WRB/USDA); the SiBCS `horizonte_vertico` wrapper passes 1.0 per Embrapa (2018) Cap 2 p.73.

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

v0.9.72 designation morphological inference (opt-in)

Field-described Brazilian Vertissolos profiles (e.g.\ the Embrapa Redape curated dataset) encode vertic morphology via a `v` master-letter modifier in the horizon designation (`Bv`, `Bvk1`, `Cv`, `Cvz`) without recording `slickensides` class or `shrink_swell_cracks_cm` as numeric inputs. With `options(soilKey.vertic_designation_inference = TRUE)` the function accepts a layer as vertic when the canonical and COLE paths both fail or are NA AND the layer has `clay_pct >= min_clay` AND its designation matches a `v` master-letter modifier. Default is `FALSE`.

`vertic_properties` *Vertic properties (WRB 2022)*

Description

Tests whether any horizon shows vertic properties – shrink-swell clay behaviour evidenced by slickensides, wedge-shaped peds, and deep cracks. Diagnostic for Vertisols.

Usage

```
vertic_properties(
  pedon,
  min_clay = 30,
  min_thickness = 25,
  slickenside_levels = c("common", "many", "continuous")
)
```

Arguments

`pedon`	A `PedonRecord`.
`min_clay`	Minimum clay percent (default 30, per WRB 2022).
`min_thickness`	Minimum thickness (cm) of the vertic layer (default 25 per WRB 2022 Ch 3.2.x).
`slickenside_levels`	Vector of `slickensides` values accepted as evidence (default `c("common", "many","continuous")`).

Details

Sub-tests:

- `test_clay_above` – clay >= 30%
- `test_slickensides_present` – slickensides at or above the "common" level
- `test_minimum_thickness` – combined vertic layer thickness >= 25 cm (v0.3.1 added per WRB 2022)

v0.3.1: thickness gate added. Limitations remaining: WRB also accepts deep cracks (>= 1 cm wide extending from the surface to >= 50 cm depth, when soil is dry) and wedge-shaped peds as alternative evidence; this implementation requires clay + slickensides. The "after mixing of upper 18 cm" clause from WRB is still deferred.

Value

A `DiagnosticResult`.

References

IUSS Working Group WRB (2022), Chapter 3.2 – Vertic properties.

`vertisol` *Vertisol RSG gate (WRB 2022 Ch 4, p 101)*

Description

WRB-canonical: vertic horizon \<= 100 cm AND \>= 30% clay between the surface and the vertic horizon throughout AND shrink-swell cracks that start at the surface (or below a plough layer / below a self- mulching surface / below a surface crust) and extend to the vertic horizon.

Usage

```
vertisol(pedon, strict = NULL)
```

Arguments

`pedon`	A `PedonRecord`.
`strict`	Logical or `NULL`. When `NULL` (default) it resolves via `getOption("soilKey.rsg_strict", FALSE)`. `TRUE` applies the Tier-3 strengthened threshold.

Details

v0.3.4 enforces (1) vertic horizon, (2) all overlying layers \>= 30% clay, and (3) shrink-swell cracks that start within the upper 20 cm. "Cracks extending to the vertic horizon" is enforced indirectly by the test_shrink_swell_cracks test that already requires an explicit `cracks_width_cm` value.

Value

A `DiagnosticResult`.

Tier-3 strict mode (v0.9.98)

With `strict = TRUE` the overlying-clay threshold is raised from 30% to 35%, tightening the gate against marginally clayey profiles that satisfy the vertic horizon but sit close to the Vertisol cut-off.

`vitric_properties` *Vitric properties (WRB 2022 Ch 3.2.16)*

Description

Volcanic glass \>= 5% in 0.02-2 mm fraction, Al_ox + 1/2 Fe_ox \>= 0.4%, phosphate retention \>= 25%.

Usage

```
vitric_properties(
  pedon,
  min_glass_pct = 5,
  min_alfe = 0.4,
  min_p_retention = 25
)
```

Arguments

`pedon`	A `PedonRecord`.
`min_glass_pct`	Numeric threshold or option (see Details).
`min_alfe`	Numeric threshold or option (see Details).
`min_p_retention`	Numeric threshold or option (see Details).

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

`vlm_pick_provider` *Pick the best available VLM provider*

Description

Selects a provider based on what is reachable in the user's environment, in this preference order: local Ollama (if `ollama_is_running()`), then Anthropic, OpenAI, and Google (each requires the relevant `*_API_KEY` environment variable). Errors with an actionable installation / API-key hint when no provider is reachable.

Usage

```
vlm_pick_provider(verbose = TRUE)
```

Arguments

`verbose`	If `TRUE` (default), emits a one-line `cli` message explaining the chosen provider.

Value

Character scalar: one of `"ollama"`, `"anthropic"`, `"openai"`, `"google"`.

`vlm_provider` *Construct a VLM provider chat object*

Description

Returns an `ellmer` chat object configured for the given provider, ready to be passed to the extraction functions (`extract_horizons_from_pdf`, etc.). The chat object wraps API credentials and model selection; it does not itself send any request.

Usage

```
vlm_provider(
  name = c("auto", "anthropic", "openai", "google", "ollama"),
  model = NULL,
  ...
)
```

Arguments

`name`	Provider name. One of `"anthropic"` (Claude), `"openai"` (GPT-4o family), `"google"` (Gemini), `"ollama"` (local).
`model`	Optional model identifier; defaults to `default_model(name)`.
`...`	Additional arguments forwarded to the corresponding `ellmer::chat_*` constructor (e.g. `system_prompt`, `api_key`, `base_url`, `params`).

Details

This is purely a convenience wrapper: it picks a default model per provider and forwards remaining arguments (e.g. `system_prompt`, `api_key`) to the underlying ellmer constructor. `ellmer` must be installed.

Value

An `ellmer` `Chat` object exposing a `$chat()` method for sending prompts.

Local-first option

Passing `name = "ollama"` runs every extraction locally via an Ollama server (default `gemma4:e4b`, Gemma 4 edge with multimodal text+image+audio support). No data leaves the machine, which is the recommended setting for sensitive field descriptions (e.g. governmental surveys, indigenous land studies) where institutional independence and data sovereignty matter. Pull the model first:

```
ollama pull gemma4:e4b      # ~3 GB edge variant (default)
ollama pull gemma4:31b      # frontier dense variant
ollama pull gemma3:27b      # earlier generation, still solid
```

Then start an Ollama server (`ollama serve`) and the chat object returned here will dispatch over HTTP locally.

Examples

```
## Not run:
# Cloud provider (needs ANTHROPIC_API_KEY)
provider <- vlm_provider("anthropic")

# Local Gemma 4 edge model -- default, ~3 GB, runs anywhere
provider <- vlm_provider("ollama")

# Local Gemma 4 frontier dense model -- best quality
provider <- vlm_provider("ollama", model = "gemma4:31b")

# Any other multimodal model the user has pulled
provider <- vlm_provider("ollama", model = "qwen2.5vl:32b")

## End(Not run)
```

`wrb06_code_to_rsg` *WRB 2006 RSG code -> 2022 RSG name*

Description

AfSP ships WRB 2006 RSG codes (2-letter, e.g.\ LV, AC, AR). The 2-letter codes are stable across WRB editions (2006 -> 2022); only a handful of qualifier names changed. This helper maps the codes to the WRB 2022 RSG names that `classify_wrb2022` emits.

Usage

```
wrb06_code_to_rsg(code)
```

Arguments

`code`	Character vector of WRB 2006 codes.

Value

Character vector of singular WRB 2022 RSG names; `NA` for unrecognised codes.

`wrb2022_canonical` *WRB 2022 canonical reference (parsed IUSS Working Group WRB 2022)*

Description

Convenience wrapper for `canonical_reference("WRB_4th_2022")`. Returns a 3-element list:

- `$rsg` (118 obs): Reference Soil Group + criteria text
- `$pq` (661 obs): principal qualifiers per RSG
- `$sq` (1167 obs): supplementary qualifiers per RSG

Usage

```
wrb2022_canonical(prefer_pkg = TRUE)
```

Arguments

`prefer_pkg`	If `TRUE` (default), prefer the installed SoilTaxonomy package over the vendored copy. Set to `FALSE` to force the vendored copy (e.g. for reproducibility of a specific soilKey release).

Details

Source: NCSS-tech `SoilTaxonomy` R package. Original: IUSS Working Group WRB (2022). *World Reference Base for Soil Resources*, 4th edition.

Value

The canonical WRB 2022 reference data (a list / data.frame of RSG and qualifier criteria), as vendored or sourced from the **SoilTaxonomy** package.

`yermic_properties` *Yermic properties (WRB 2022 Ch 3.2.17) – per-pedon test wrapping* `test_yermic_surface`*.*

Description

Yermic properties (WRB 2022 Ch 3.2.17) – per-pedon test wrapping `test_yermic_surface`.

Usage

```
yermic_properties(pedon)
```

Arguments

pedon	A `PedonRecord`.

Value

A `DiagnosticResult` recording whether the diagnostic is present, the qualifying layers, and the supporting evidence.

Index