IT-Based Landscape Stratification for Rain-Gauge Network Design

Chris Reudenbach

2025-12-10

Introduction

Designing an optimal precipitation-monitoring network is not a simple geostatistical optimisation task. It requires understanding:

- how **landscape structure** affects rainfall organisation,
- how **information gain** varies across space,
- how **hydrological response** depends on spatial rainfall patterns.

This document implements **Layer 1** of a modern three-layer network-design framework:

1. **Physiographic stratification** (structural representativeness)
2. **Information-gain optimisation** (analytical efficiency)
3. **Hydrological coupling** (functional adequacy)

Here we implement **Layer 1**: an objective stratification of the Burgwald landscape using **information-theoretic metrics** and **pattern-based texture signatures**.

Conceptual background

Physiographic stratification

Goal: ensure that rainfall stations represent the dominant structural units of the landscape. Landscape units differ in:

- land-cover composition
- forest–openland transitions
- patch fragmentation and clumpiness
- canopy interception potential

- micro-topographic exposure

We capture these differences using information-theoretic metrics:

- **Entropy H(x)**
  - measures thematic diversity of land cover
  - low → homogeneous (e.g. closed forest)
  - high → heterogeneous, mixed, patchy
- **Relative mutual information U**
  - measures configurational order (clumpiness)
  - low → fragmented patterns
  - high → ordered, aggregated patterns

Each 2×2 km grid cell gets a structural fingerprint in **(H, U)-space**, and clustering these fingerprints yields **objective structural strata**.

Information-gain optimisation

Once strata exist, statistical optimisation (e.g. kriging variance, radar–gauge mismatch, representativeness error) should be applied **within and across strata**, not across the whole catchment at once. This avoids bias towards "statistically convenient" open areas and keeps the optimisation linked to process-relevant structural units.

Hydrological coupling

Finally, a network must improve the prediction of:

- streamflow peaks (P–Q coherence)
- storage changes
- water-balance residuals

Even a statistically optimal station is useless if it does not represent rainfall regimes controlling the hydrograph. This QMD prepares the structural layer required for such coupling.

Load AOI and land cover

```r
# Load AOI and land cover
lc_burgwald  <- terra::rast(clc_file)
aoi_burgwald <- sf::read_sf(aoi_file)

# Harmonise CRS
if (!sf::st_crs(aoi_burgwald) == sf::st_crs(lc_burgwald)) {
  aoi_burgwald <- sf::st_transform(aoi_burgwald, sf::st_crs(lc_burgwald))
}

# Clip raster strictly to AOI
lc_burgwald <- lc_burgwald |>
  terra::crop(terra::vect(aoi_burgwald)) |>
  terra::mask(terra::vect(aoi_burgwald))

# Ensure integer categories for landscapemetrics
terra::values(lc_burgwald) <- round(terra::values(lc_burgwald))
```

Compute IT metrics on a 2×2 km structural grid

Concept:

- Create a regular **2×2 km** grid over the Burgwald.
- Treat each grid cell as a small "landscape".
- Compute **entropy H(x)** and **relative mutual information U** for each grid cell.
- Each cell becomes a point in (H, U)-space.

```r
# Transform AOI to metric CRS so that cellsize is in metres
aoi_utm <- sf::st_transform(aoi_burgwald, 25832)  # ETRS89 / UTM 32N

# Grid resolution (metres)
grid_cellsize <- 2000  # 2 km × 2 km

# Build grid in UTM space
grid_utm <- sf::st_make_grid(
  aoi_utm,
  cellsize = grid_cellsize,
  what     = "polygons"
) |>
  sf::st_as_sf() |>
  dplyr::mutate(grid_id = dplyr::row_number())
```

```r
# Back-transform grid to raster CRS
grid <- sf::st_transform(grid_utm, sf::st_crs(lc_burgwald))

# Compute entropy and relative mutual information per grid cell
lsm_grid_it <- sample_lsm(
  landscape = lc_burgwald,
  y         = grid,
  what      = c("lsm_l_ent", "lsm_l_relmutinf"),
  level     = "landscape"
)

# Wide format: grid_id | ent | relmutinf
lsm_grid_it_wide <- lsm_grid_it |>
  dplyr::select(grid_id = plot_id, metric, value) |>
  tidyr::pivot_wider(names_from = metric, values_from = value)

grid_it_sf <- grid |>
  dplyr::left_join(lsm_grid_it_wide, by = "grid_id")
```

Cluster cells into IT-based strata

Concept:

- Use (H, U) to cluster grid cells into **k structural strata**.
- Each stratum is a region with similar composition and configuration.
- We will later select representative cells per stratum as station candidates.

```r
# Extract numeric metrics
it_clust_df <- grid_it_sf |>
  sf::st_drop_geometry() |>
  dplyr::select(grid_id, ent, relmutinf) |>
  dplyr::filter(!is.na(ent), !is.na(relmutinf))

# Standardise to z-scores
it_clust_scaled <- it_clust_df |>
  dplyr::mutate(
    ent_z       = as.numeric(scale(ent)),
    relmutinf_z = as.numeric(scale(relmutinf))
  )

# Number of strata
```

```r
k_strata <- 5

set.seed(123)
km_it <- stats::kmeans(
  it_clust_scaled[, c("ent_z", "relmutinf_z")],
  centers = k_strata,
  nstart  = 50
)

it_clust_scaled$stratum_id <- km_it$cluster

# Attach strata to grid sf
grid_it_strata_sf <- grid_it_sf |>
  dplyr::left_join(
    it_clust_scaled |> dplyr::select(grid_id, stratum_id),
    by = "grid_id"
  )
```

Select multiple representative cells per stratum

We now choose several "typical" cells per stratum (here: 4 each → 20 candidates total):

```r
n_per_stratum <- 4

# Stratum centroids in z-space
strata_centroids <- it_clust_scaled |>
  dplyr::group_by(stratum_id) |>
  dplyr::summarise(
    ent_z_mean       = mean(ent_z, na.rm = TRUE),
    relmutinf_z_mean = mean(relmutinf_z, na.rm = TRUE),
    .groups = "drop"
  )

# Distance to stratum centroid
it_clust_scaled <- it_clust_scaled |>
  dplyr::left_join(strata_centroids, by = "stratum_id") |>
  dplyr::mutate(
    dist_to_center = sqrt(
      (ent_z       - ent_z_mean)^2 +
      (relmutinf_z - relmutinf_z_mean)^2
    )
```

```r
  )

# n nearest cells per stratum
rep_cells <- it_clust_scaled |>
  dplyr::group_by(stratum_id) |>
  dplyr::slice_min(order_by = dist_to_center,
                   n = n_per_stratum,
                   with_ties = FALSE) |>
  dplyr::ungroup()

# Attach geometries
rep_cells_sf <- grid_it_strata_sf |>
  dplyr::inner_join(
    rep_cells |>
      dplyr::select(
        grid_id,
        stratum_id,
        ent,
        relmutinf,
        ent_z,
        relmutinf_z
      ),
    by = c("grid_id", "stratum_id")
  )
```

Compute station-candidate centroids

```r
station_candidates_sf <- rep_cells_sf |>
  sf::st_centroid() |>
  dplyr::mutate(
    station_id = dplyr::row_number()
  ) |>
  dplyr::select(
    station_id,
    stratum_id,
    grid_id,
    ent,
    relmutinf,
    ent_z,
    relmutinf_z
```

```
    # geometry is kept implicitly by sf
  )
```

Pattern-based signatures (motif COVE)

Concept:

- motif calculates **co-occurrence signatures** on moving windows.
- Here: window size 25×25 cells (~2.5 km at 100 m resolution).
- Each window yields a probability distribution of adjacency patterns.
- Distances between signatures (e.g. Jensen–Shannon) define **pattern types**.

```
lsp_cove_burgwald <- lsp_signature(
  x             = lc_burgwald,
  type          = "cove",
  window        = 25,
  normalization = "pdf"
)

dist_mat <- lsp_to_dist(
  x        = lsp_cove_burgwald,
  dist_fun = "jensen-shannon"
)

k_pattern <- 6
set.seed(123)
hc <- hclust(dist_mat, method = "ward.D2")
pattern_ids <- cutree(hc, k = k_pattern)

lsp_cove_burgwald$pattern_cluster <- pattern_ids

lsp_pattern_sf <- lsp_add_sf(lsp_cove_burgwald)

# Reduce list-columns for plotting / joins
geom_col <- attr(lsp_pattern_sf, "sf_column")
lsp_pattern_sf_plot <- lsp_pattern_sf[, c("pattern_cluster", geom_col)]
lsp_pattern_sf_plot$pattern_cluster <- as.factor(lsp_pattern_sf_plot$pattern_cluster)
```

Attach pattern types to station candidates

```
station_with_pattern_sf <- sf::st_join(
  station_candidates_sf,
  lsp_pattern_sf_plot,
  join = sf::st_intersects,
  left = TRUE
)
```

Visualisation (optional)

```
mapview(grid_it_strata_sf, zcol = "stratum_id")
mapview(station_with_pattern_sf, zcol = "stratum_id", cex = 4)
mapview(lsp_pattern_sf_plot, zcol = "pattern_cluster")
```

Outlook: combining IT-strata with classical physiographic stratification

The IT-based stratification identifies **landscape structural regimes**, but it should be combined with more traditional physiographic factors.

Future integration should combine IT strata with:

- **Elevation bands** → capture orographic rainfall gradients
- **Slope and aspect classes** → control exposure, interception, wind undercatch
- **Geological units / parent material** → influence stormflow generation processes
- **Hydrotopes / representative hillslopes** → link rainfall structure to runoff response
- **Forest structural types** (deciduous, coniferous, mixed, stand age) → modify canopy storage and interception patterns
- **Cold-air drainage pathways** → relevant for radiation fog and winter precipitation phase transitions
- **Convective storm tracks / radar climatology** → areas repeatedly hit by convective cells
- **Accessibility and operational constraints** → maintenance, vandalism, telemetry, power supply

The final Burgwald network should merge:

1. **IT strata** – structural representativeness
2. **Physiographic strata** – process-based representativeness
3. **Information-gain optimisation** – cost-efficient densification

4. **Hydrological validation** – functional adequacy

Together, these layers define a (assumingly) scientifically defensible, resource-efficient, and hydrologically meaningful precipitation-monitoring network for the Burgwald.