STE General Workflow

Technical Guide

Elephant Movement Ecology — Methodology & Calculation Reference

Version 1.1 · Generated July 21, 2026

1. Overview

The **STE General Workflow** is a comprehensive data-to-report pipeline purpose-built for analysing African elephant (*Loxodonta africana*) movement ecology. It ingests GPS telemetry from **EarthRanger**, derives seasonal boundaries from **Google Earth Engine** NDVI time-series, performs a rich sequence of spatial computations, and delivers **14 interactive map products**, a protection status chart, seasonal CSV data, an interactive dashboard, and a print-ready Word report — all at the subject-group level.

The workflow is implemented as an **Ecoscope Workflow** (YAML spec) and executes within the ecoscope-workflows runtime. Tasks are chained via data references (${{ workflow.<id>.return }}); the engine resolves execution order automatically. Fan-out iteration over individual subjects uses the mapvalues directive.

> **Note:** The trajectory segment filter defaults (max speed 9 knots, max gap 6 h) are tuned for elephant movement. All thresholds are user-configurable at run time through the workflow configuration form.

The workflow produces the following outputs per run:

Category	Outputs
Movement maps	Movement Tracks (current + previous), Subject Tracks
Home range maps	Overall ETD, Filtered 99th percentile, Protected areas ETD, Unprotected areas ETD, Seasonal HR per subject
Habitat use maps	Recursion Events raster, Dry season speed raster, Wet season speed raster
Activity maps	Time of Day Dominance grid, Night Fixes grid
Event maps	Elephant Sightings / Collaring Events
Charts	Protection Status bar chart
Data files	Relocations, trajectories, ETD polygons, seasonal windows CSV (GeoParquet + CSV)
Report	Word document: cover page + general section; Interactive dashboard

2. Dependencies & Prerequisites

2.1 EarthRanger Connection

All telemetry is sourced from an **EarthRanger** instance (set_er_connection). The workflow calls get_subjectgroup_observations twice — once for the current analysis period and once for the comparison (previous) period. Both calls use filter: clean, discarding any observation already flagged as junk before the data reaches the workflow. include_details and include_subjectsource_details are both set to false to minimise payload size.

- The connection handle is passed as client to every EarthRanger fetch task.
- EarthRanger events are fetched separately via get_events using the same client.

2.2 Google Earth Engine Connection

A **Google Earth Engine (GEE)** service account is required for seasonal analysis. The task custom_determine_season_windows queries the GEE NDVI time-series for the region of interest (derived from the ETD raster extent) to classify the analysis period into wet and dry seasons. The GEE project is supplied by the user

via set_gee_connection.

2.3 Subject Group

The user specifies a **Subject Group Name** (default: Elephants) that must match exactly — case-sensitively — a subject group in the connected EarthRanger instance. All subjects within that group are fetched for the configured time range. The grouper is fixed to subject_name, so outputs are produced at the group level (not per individual).

> **Note:** Unlike the STE Mapbook workflow, the General workflow does not split outputs per subject for most maps — it generates one map per map type across the entire group. Only seasonal home range maps are fanned out per subject.

2.4 Map Overlay & landDx Geodatabase

The **Map Overlay** step (select_map_overlay) selects the protected-area layer shared by every map, and doubles as the basis for the protected/unprotected split (Section 4.6) and the Protection Status bar chart (Section 5). Only protected-area-style layers are supported. Three sources are available:

- **LandDx (default)** — the standard protected-area GeoPackage, downloaded and styled automatically; no input required.
- **EarthRanger Spatial Feature** — fetches one or more named feature sets from the connected EarthRanger instance.
- **Custom Local File** — one or more local GeoJSON / GeoPackage / GeoParquet files, each becoming its own styled layer.

When **LandDx** is selected, the geodatabase (landDx.gpkg, layer landDx_polygons) is downloaded from a Dropbox URL by default (a local path may be supplied instead), then filtered to three land-use types with only three columns retained (type, name, geometry).

Land-use Type	Fill / Line Colour	Opacity
Community Conservancy	RGB(166, 182, 151) — #a6b697	17.5 %
National Reserve	RGB(136, 167, 142) — #88a78e	17.5 %
National Park	RGB(17, 86, 49) — #115631	17.5 %

A text-label layer (centroid-anchored, Arial, 1 000 m base size, 40–75 px screen clamp) renders protected-area names. The database is also split by type and the geometry type is annotated before DeckGL layer creation.

2.5 Base Map Tile Layers

Every map uses two stacked tile services, configured through the **Configure basemap layers** step (set_base_maps_pydeck). The URL, opacity, and max zoom of each layer are pre-filled with the defaults below but are user-editable:

Layer	URL (abbreviated)	Opacity	Max zoom
ESRI World Hillshade	arcgisonline.com/…/World_Hillshade/…/{z}/{y}/{x}	0.995	20
ESRI World Boundaries & Places	arcgisonline.com/…/World_Boundaries_and_Places_Alternate/…/{z}/{y}/{x}	0.35	20

3. Data Ingestion Pipeline

3.1 Observations → Relocations

process_relocations converts raw EarthRanger observations into a standardised relocations GeoDataFrame. The following columns are extracted and retained:

Column	Source field	Description
groupby_col	internal	Subject identifier for grouping
fixtime	observation timestamp	UTC datetime of the GPS fix
junk_status	EarthRanger flag	True if fix is marked junk
geometry	lat/lon	Point geometry (WGS 84)
extra__subject__name	subject.name	Elephant display name
extra__subject__hex	subject.hex_color	Subject colour used for per-subject maps
extra__subject__sex	subject.sex	Subject sex (M / F / unknown)
extra__created_at	observation.created_at	Record creation timestamp
extra__subject__subject_subtype	subject.subject_subtype	Subject subtype (e.g. elephant)

Three known-bad coordinate pairs are filtered out unconditionally:

- (180.0, 90.0) — boundary sentinel
- (0.0, 0.0) — null-island artefact
- (1.0, 1.0) — common default / test value

3.2 Day / Night Annotation

Each relocation is labelled as day or night by classify_is_night. Solar elevation is calculated at the fix's location and UTC timestamp. A fix is classified as night (is_night = True) when the sun is below the horizon. This boolean is later used as the colour key for the Time of Day Dominance and Night Fixes maps.

3.3 Trajectory Segment Filter

custom_trajectory_segment_filter builds a reusable filter object that is applied to *both* the current and previous period trajectories, ensuring methodological consistency across periods.

Parameter	Default	Unit	Purpose
min_length_meters	0.001	m	Remove sub-centimetre GPS noise
max_length_meters	5 000	m	Remove implausible long jumps
min_time_secs	1	s	Remove zero-duration segments
max_time_secs	21 600	s	Remove gaps > 6 h (fix dropout)
min_speed_kmhr	0.01	km/h	Remove near-stationary fixes

Parameter	Default	Unit	Purpose
max_speed_kmhr	9	km/h	Remove biologically impossible speeds for elephants

3.4 Relocations → Trajectories

relocations_to_trajectory connects consecutive fixes for the same subject into linear trajectory segments. Each segment records:

- **dist_meters** — straight-line distance between the two fixes
- **speed_kmhr** — dist_meters ÷ elapsed time × 3.6
- **segment_start / segment_end** — timestamps of the bounding fixes
- **geometry** — LineString in WGS 84

A temporal index is then added (add_temporal_index) keyed on segment_start and grouped by subject name. Column names are normalised via map_columns (e.g. extra__name → subject_name, extra__hex → hex_color).

3.5 NDVI Seasonal Window Detection

custom_determine_season_windows queries the Google Earth Engine NDVI time-series for the region of interest (the ETD raster extent). Wet and dry season windows are identified from the NDVI phenology curve: periods where NDVI exceeds a threshold are classified as *wet*; periods below are classified as *dry*. The function returns a DataFrame of dated season intervals, which is persisted as seasonal_windows.csv for auditability.

> **Note:** Season boundaries are data-driven — they reflect the actual green-up and senescence pattern for the specific landscape and year, rather than fixed calendar dates. This is critical in East African ecosystems where rainfall timing varies considerably year to year.

3.6 Season Labelling

create_seasonal_labels joins the season windows onto the trajectory GeoDataFrame by timestamp, appending a season column ('wet' or 'dry') to each segment. All subsequent raster and home range analyses that are season-specific operate on this labelled GeoDataFrame.

3.7 Previous Period

The previous period is determined by determine_previous_period. The same trajectory segment filter is applied to the previous period trajectories. A duration_status column is appended — 'Previous tracks' — before merging with current period trajectories for the Movement Tracks Map.

3.8 Speed Classification

apply_classification bins the speed_kmhr column using **Equal Interval** with **k = 6**. Labels are formatted to one decimal place with the suffix km/h. The resulting speed_bins column is used on the speed-coloured layers in the Subject Tracks context.

4. Map Outputs — Methodology

4.1 Movement Tracks Map

Overlays current and previous period trajectories for the entire subject group, allowing rangers to identify range shifts across time.

Dual-period construction

After independent trajectory builds for both periods, a duration_status column is appended to each ('Current tracks' or 'Previous tracks'). The two GeoDataFrames are merged with merge_multiple_df and colours are assigned by assign_duration_colors / modify_status_colors. Trajectories are sorted descending by duration_status so current tracks always render on top.

Retained columns

Before layer creation, filter_movement_cols retains only: duration_status, duration_status_colors, is_night, geometry.

Line styling

Property	Value
Width	2.85 px (min 2, max 8, screen-space)
Opacity	55 %
Cap / Join	Rounded
Legend sort	Ascending by duration_status

4.2 Subject Tracks Map

Trajectories coloured by individual subject using each animal's own EarthRanger hex colour, providing a per-animal view of space use without the time-comparison overlay of the Movement Tracks Map.

convert_hex_to_rgba converts the hex_color string field to an RGBA list (rgba_colors) compatible with DeckGL's get_color property. The legend labels subjects by subject_name.

4.3 Elephant Sightings / Collaring Events Map

get_events fetches EarthRanger events for the analysis period. Events include collaring records and elephant sightings logged in the field. Only events with a point geometry are included (include_null_geometry: false).

Columns retained per event:

- id, time, event_type, event_category, reported_by, serial_number, geometry, created_at, event_details

Scatter layer styling

Property	Value
Fill colour	RGB(85, 107, 47) — #556b2f (dark olive green)
Line colour	RGBA(0, 0, 0, 200) (near-black outline)
Line width	0.55
Radius	3.55
Opacity	75 %

4.4 Overall Home Range Map (ETD)

The primary space-use estimator. Elliptical Time Density (ETD) weights space use by the time spent in each area, producing ecologically realistic home range contours.

ETD parameters — calculate_elliptical_time_density

Parameter	Value	Meaning
CRS	ESRI:53042	World Azimuthal Equidistant — equal-area, minimises distortion across Africa
cell_size	Auto-scale	Derived automatically from data extent and trajectory density
max_speed_factor	1.05	Segments with speed > 1.05 × median are down-weighted
expansion_factor	1.3	Raster extent expanded 30 % beyond data envelope
band_count	1	Single density band output
nodata_value	NaN	Cells outside probability mass are transparent
percentiles	50, 60, 70, 80, 90, 95, 99.9	Contour polygons at these probability thresholds

After computation the raster is reprojected to **EPSG:4326**. Contour polygons are coloured using the **RdYlGn** diverging colormap — inner core (50th percentile) in red, outer extent (99.9th) in green.

4.5 Filtered 99th Percentile Home Range Map

A simplified view retaining only the 99th percentile contour from the overall ETD, displayed in a single chocolate tone for clean cartographic presentation.

filter_df_values selects rows where percentile ≥ 99.0. The layer uses a fixed fill colour of **RGB(210, 105, 30) — #d2691e** (chocolate) at 55 % opacity, with a static legend entry labelled *99th percentile*.

4.6 Protected and Unprotected Areas Home Range Maps

Two additional ETD maps show how much time elephants spend inside versus outside protected land, supporting ranger management decisions.

Protection status determination

generate_protected_column performs a spatial join between the trajectory GeoDataFrame and the filtered landDx polygons (column: type). Each trajectory segment is labelled protection_status = 'Protected' if its centroid falls within a Community Conservancy, National Reserve, or National Park polygon; otherwise 'Unprotected'.

Two separate ETD calculations are then run — one on the protected subset and one on the unprotected subset — using identical parameters to the overall ETD (percentiles 50–99.9 %, RdYlGn colormap). The resulting GeoDataFrames are persisted as:

- protected_areas_home_range_etd.geoparquet
- unprotected_areas_home_range_etd.geoparquet

4.7 Seasonal Home Range Maps (per subject)

The only outputs fanned out *per subject* via mapvalues. Shows how each individual's space use shifts between wet and dry seasons.

Fan-out pipeline

split_groups partitions the season-labelled trajectory GeoDataFrame by the subject_name grouper, producing one GeoDataFrame per subject. calculate_seasonal_home_range then groups each subject's trajectories by season and computes an ETD for each season, extracting only the **99.9th percentile** contour. Cell size is auto-scaled independently for each subject–season combination.

Colour assignment

assign_season_colors maps season labels to a fixed palette (wet = blue family, dry = orange/brown family). The season column is first converted to string to ensure consistent matching. Polygons are rendered as filled, semi-transparent GeoJSON layers (55 % opacity, 0.35 px line width) so both seasons are visible when overlapping.

View state and map layers are zipped per subject via zip_groupbykey before being passed to draw_map. Output filenames are suffixed with the unique subject name retrieved by dataframe_column_first_unique_str.

4.8 Recursion Events Raster

The recursion raster identifies landscape cells that elephants return to repeatedly — key resource hotspots such as waterholes, mineral licks, and seasonal foraging areas.

Ecograph raster parameters

Parameter	Value	Effect
step_length	User-set via Raster maps → Step Length / Grid Size (default 2 000 m)	Trajectories are resampled to this step length before aggregation
movement_covariate	null	No speed metric — the count of revisits (network weight) is aggregated
interpolation	mean	Cell values averaged across contributing steps
network_metric	weight	Ecograph network edge weight — proportional to revisit frequency
radius	2	Each cell incorporates contributions from its 2-ring neighbourhood
dist_col	dist_meters	Source column for step-length weighting
tortuosity_length	3	Path sinuosity computed over 3-step windows
filename	weighted_raster	Output raster filename prefix

Note: The **Step Length / Grid Size** field in the **Raster maps** step of the workflow form sets a single grid size shared by this Recursion Events raster *and* both the Dry and Wet Season Mean Speed rasters (Sections 4.9–4.10) — it is not fixed per raster.

Classification and colouring

Raw per-cell values are classified using **Natural Breaks (Jenks)** with **k = 6**. Natural breaks minimises within-class variance — well-suited to the right-skewed distribution of revisit counts. The six-colour green-to-red ramp is applied:

Bin	Hex	Perception
1 (lowest)	#1a9850	Dark green — rarely visited
2	#91cf60	Light green

Bin	Hex	Perception
3	#d9ef8b	Yellow-green
4	#fee08b	Yellow
5	#fc8d59	Orange
6 (highest)	#d73027	Red — high-frequency recursion area

4.9 Dry Season Mean Speed Raster

Reveals landscape-scale movement intensity during the dry season — typically characterised by longer, faster directional travel to water.

Season filtering

filter_df selects rows where season == 'dry' from the season-labelled trajectory GeoDataFrame.

Ecograph raster parameters

Parameter	Value	Effect
step_length	Shared with Recursion Events raster (default 2 000 m, user-set)	Trajectories resampled to this step length before aggregation
movement_covariate	speed	Metric being aggregated is speed (km/h)
interpolation	mean	Per-cell values are mean speed across contributing steps
radius	2	Neighbourhood smoothing over 2-ring radius
network_metric	null	No network metric — pure speed raster

Post-processing

After feature extraction, the GeoDataFrame is reprojected to **EPSG:4326** and sorted ascending by value. Natural Breaks classification (**k = 6**) produces value_bins. map_values_with_unit formats bin labels to one decimal place in km/h. The same six-colour green-to-red ramp used for the recursion raster is applied.

4.10 Wet Season Mean Speed Raster

Identical pipeline to the dry season speed raster (Section 4.9) but filtering for season == 'wet'. Wet season movement typically shows slower, more exploratory patterns as resource availability is higher.

4.11 Time of Day Dominance Map

A grid-based map showing whether each landscape cell is used predominantly during the day or at night — revealing temporal habitat partitioning.

Meshgrid construction

create_meshgrid generates a regular spatial grid over the extent of the annotated relocations (CRS ESRI:53042, auto-scaled cell size, intersecting_only: false). The same meshgrid is reused for the Night Fixes map (Section 4.12).

Dominance calculation

get_day_night_dominance assigns each grid cell a boolean is_night_majority — True if more than half of fixes within the cell have is_night = True.

Colour assignment

is_night_majority	Colour	Hex
False (day)	Cornflower blue	#6495ed
True (night)	Dark navy blue	#00008b

Cells are sorted descending by is_night_majority so night cells (True) render last and appear on top where they overlap with day cells.

4.12 Night Fixes Map

Highlights cells where a large majority of GPS fixes are nocturnal — identifying areas of consistent night-time habitat use.

get_grid_night_fixes operates on the same meshgrid and annotated relocations as Section 4.11. A cell is labelled night_activity = True when the proportion of nocturnal fixes meets or exceeds the **threshold of 0.65** (65 %).

night_activity	Colour	Hex
False	Cornflower blue	#6495ed
True (≥ 65 %)	Dark navy blue	#00008b

5. Protection Status Analysis

In addition to the home range maps (Section 4.6), the workflow generates a **Protection Status Bar Chart** that summarises the proportion of trajectory fixes occurring inside versus outside protected land across all subjects and the full analysis period.

plot_fix_protection_status receives the full protection_status-annotated GeoDataFrame (from generate_protected_column) and renders an interactive bar chart grouped by protection status. The chart is persisted as both HTML and a 2× PNG screenshot.

6. Word Report & Dashboard

6.1 Cover Page

The cover page template (general_ele_cover_page.docx) is downloaded automatically from Dropbox and populated by create_mapbook_ctx_cover with:

- Total unique subject count (derived from dataframe_column_nunique on subject_name)
- Analysis time range (since / until)
- Prepared by: *Ecoscope*
- Organisation logo (user-supplied; fetched via get_file_path / Report logo task group)

6.2 General Report Section

The per-run section template (general_elephant_grouper_page.docx) is downloaded from Dropbox and rendered by general_template_context. Map image boxes are **5.34 × 3.12 inches**. Screenshots are captured at 2× device scale factor with a 40 000 ms tile-load wait.

6.3 Document Merge

merge_mapbook_files concatenates the cover page document and the general section into a single Word file, saved as the final combined report.

6.4 Interactive Dashboard

gather_dashboard assembles the dashboard from workflow metadata (details, time range, groupers). The widgets list is empty for this workflow — the dashboard provides a summary view of run parameters rather than embedded map widgets.

7. Output Files

All outputs are written to the directory defined by ECOSCOPE_WORKFLOWS_RESULTS.

File	Format	Content
relocations.geoparquet	GeoParquet	Current period GPS fixes with day/night labels
previous_period_relocations.geoparquet	GeoParquet	Previous period GPS fixes
trajectories.geoparquet	GeoParquet	Current period segments with season labels
previous_period_trajectories.geoparquet	GeoParquet	Previous period segments
home_range_etd.geoparquet	GeoParquet	Overall ETD contours (50–99.9th percentiles)
protected_areas_home_range_etd.geoparquet	GeoParquet	ETD for fixes inside protected land
unprotected_areas_home_range_etd.geoparquet	GeoParquet	ETD for fixes outside protected land
_seasonal_etd.geoparquet	GeoParquet	Seasonal ETD polygons per subject
seasonal_windows.csv	CSV	NDVI-derived wet/dry season date ranges
weighted_raster.*	Raster	Recursion ecograph raster file
movement_tracks.html / .png	HTML / PNG	Movement Tracks map (current + previous period)
subject_tracks.html / .png	HTML / PNG	Subject Tracks map (coloured per individual)
collared_points.html / .png	HTML / PNG	Elephant Sightings / Collaring Events map
overall_homerange.html / .png	HTML / PNG	Overall ETD Home Range map
filtered_homerange.html / .png	HTML / PNG	Filtered 99th percentile Home Range map
protected_areas.html / .png	HTML / PNG	Protected Areas Home Range map
unprotected_areas.html / .png	HTML / PNG	Unprotected Areas Home Range map

File	Format	Content
_seasonal_homerange.html / .png	HTML / PNG	Seasonal Home Range per subject
recursion_events.html / .png	HTML / PNG	Recursion Events raster map
dry_mean_speed_raster_map.html / .png	HTML / PNG	Dry Season Mean Speed raster map
wet_mean_speed_raster_map.html / .png	HTML / PNG	Wet Season Mean Speed raster map
time_of_day_dominance.html / .png	HTML / PNG	Time of Day Dominance grid map
night_fixes.html / .png	HTML / PNG	Night Fixes grid map
protection_status_bar.html / .png	HTML / PNG	Protection Status bar chart
general_cover_page.docx	Word	Report cover page
general.docx	Word	General per-subject report section
.docx	Word	Final combined Word report

8. Workflow Execution Logic

8.1 Grouping Strategy

The grouper is fixed to subject_name. Most maps operate at the group level (all subjects combined). Only the seasonal home range maps use mapvalues to fan out per subject — splitting the trajectory GeoDataFrame with split_groups and processing each subject independently.

8.2 Skip Conditions

Every task carries two default skip conditions via task-instance-defaults:

- **any_is_empty_df** — if any input DataFrame is empty, the task and all its dependants are skipped rather than raising an error.
- **any_dependency_skipped** — if an upstream task was skipped, all downstream tasks are also skipped automatically.

The get_events_data task carries an explicit skipif block with both conditions, ensuring that if no events are returned from EarthRanger the collaring events map pipeline is bypassed cleanly.

8.3 Previous Period Logic

The previous period is determined by flexible_previous_period. In every mode, the comparison period's **end date** is fixed to the current time range's start date (so the two periods never overlap) — only the start date calculation differs. There are three modes:

Mode	Behaviour
Preset	Choose a common lookback — Same as current period, 1 month back, 3 months back, 6 months back, or 1 year back (default: 1 month back)
Calendar	Manually pick the exact start date for the comparison period

Mode	Behaviour
Custom	Enter a Years / Months / Weeks / Days offset to count backward from the current time range's start date (default: 1 month back)

8.4 mapvalues Fan-out (Seasonal Home Range)

The seasonal home range pipeline uses mapvalues to iterate over the list produced by split_groups. Each subsequent task in the chain (calculate_seasonal_home_range, convert_to_str, assign_season_colors, reproject_gdf, filter_df_cols, create_geojson_layer, view_state_deck_gdf, draw_map, html_to_png) receives the argvalues list from its upstream task, processing one subject per invocation. zip_groupbykey aligns the per-subject map layers with their corresponding view states before the final draw_map call.

9. Software Versions

As of the migration to the **ecoscope-platform** task/compiler runtime, the workflow's requirements are:

Package	Version	Role
ecoscope-platform	>=2.15.0, <2.16.0	Core task library and workflow engine (replaces ecoscope-workflows-core / -ext-ecoscope / -ext-big-life)
ecoscope-workflows-ext-custom	0.1.0rc14.*	Custom STE utility tasks (protection status, template context)
ecoscope-workflows-ext-ste	0.0.0rc1.*	STE-specific tasks (seasonal analysis, day/night, recursion)
pydeck	0.9.2	Renders the interactive DeckGL maps
opentelemetry-sdk	>=1.20.0, <2.0.0	Workflow run telemetry/observability instrumentation

Note: ecoscope-workflows-ext-ecoscope and ecoscope-workflows-ext-big-life are no longer direct dependencies — their task coverage now lives in ecoscope-platform.

ecoscope-platform is pulled from the ecoscope-workflows prefix.dev channel; the two ecoscope-workflows-ext-* packages from the ecoscope-workflows-custom channel; pydeck and opentelemetry-sdk from conda-forge. The runtime environment is managed by **pixi**.