XXVI Simpósio Brasileiro de Recursos Hídricos
23 a 28 de novembro de 2025 - Vitória - ES

ABRHidro
Associação Brasileira de Recursos Hídricos

XXVI SIMPÓSIO BRASILEIRO DE RECURSOS HÍDRICOS

EXTRACTING, STRUCTURING AND ORGANIZING SPATIAL, PHYSICAL AND HYDROLOGICAL DATA FOR MODELING: AN APPROACH USING RSTUDIO

Ana Laura Pascucci de Oliveira [1]; Simone Andrea Furegatti [2]& Graziele Ruas [3]

Abstract: Preparing spatial data for hydrological modeling is a complex task, especially when it involves multiple contributing areas and urban contexts, where land use intensifies spatial variability. This study presents an automated routine developed in R to extract, structure, and organize spatial, physical, and hydrological information needed for modeling in software such as Storm Water Management Model (SWMM). The routine helps optimize data pre-processing; nevertheless, critical evaluation of input data remains an essential step to ensure the quality and consistency of the analyses. Thus, the proposed approach offers an accessible, replicable, and adaptable tool for different datasets and contexts, designed to support the pre-processing of hydrological models, with the ability to inform detailed hydrological analyses and intervention planning.

Resumo: A preparação de dados espaciais para modelagem hidrológica é uma etapa complexa, especialmente quando envolve múltiplas áreas de contribuição e em contextos urbanos, nos quais a ocupação do solo intensifica a variabilidade espacial. Este estudo apresenta uma rotina automatizada desenvolvida em R para extrair, estruturar e organizar informações espaciais, físicas e hidrológicas necessárias à modelagem em softwares como o SWMM. A rotina contribui para otimizar o pré-processamento de dados; ainda assim, a avaliação crítica dos insumos permanece uma etapa essencial, garantindo a qualidade e a consistência das análises. Portanto, a abordagem proposta apresenta uma ferramenta acessível, replicável e adaptável a diferentes bases de dados e contextos, destinada ao pré-processamento de dados para modelagem hidrológica, com capacidade de subsidiar análises hidrológicas detalhadas e o planejamento de intervenções.

Keywords – Hydrological Model, Data pre-processing, R programming.

1. INTRODUCTION

With the advance of urban occupation, the management of water resources has become an increasing challenge, both in terms of directing water flows and maintaining the quality of water courses. The history of occupation and the indiscriminate transformation of space highlight the impacts on natural resources and the difficulties in integrating environmental and urban aspects in territorial planning processes (Peixoto, 2006). As a consequence of this disarticulation, water security is compromised, resulting in environmental, social, and economic impacts in occupied areas (Bao and Fang, 2008). Thus, the balance between urban development and the conservation of water resources constitutes a fundamental basis for effective management of water resources as well

[1]) Universidade Estadual Paulista (UNESP), Faculdade de Engenharia, Bauru, 17033-360, Brasil, ana.pascucci@unesp.br
[2]) Universidade Estadual Paulista (UNESP), Faculdade de Engenharia, Bauru, 17033-360, Brasil, simone.furegatti@unesp.br
[3])Universidade Estadual Paulista (UNESP), Faculdade de Engenharia, Bauru, 17033-360, Brasil, graziele.ruas@unesp.br

as a strategy to minimize the negative impacts of urbanization and achieve sustainable urban development (Peixoto, 2006; Chen et al., 2024). In this context, the watershed is consolidated as the planning and management unit, being a well-defined physical unit representative of hydrological processes (Aher et al., 2014; Vilaça et al., 2009).

In order to support urban water management through the adoption of public policies that integrate public spaces with sustainable stormwater management measures — including runoff forecasting, structures sizing, and flood mitigation — simulation models are currently used to represent hydrological processes in urban watersheds (Tucci, 2007). Computational numerical models such as the Storm Water Management Model (SWMM) allow simulation of the hydrological and hydraulic behavior of urban watersheds, serving as a convenient tool for drainage system planning and for the implementation of Nature-based Solutions (NbS). These solutions demonstrate effectiveness in reducing surface runoff and contribute to increasing biodiversity, strengthening climate resilience, recharging aquifers, improving water quality, and enhancing urban spaces (Woods Ballard et al., 2015; El Baida, Chourak & Boushaba, 2025; Tsatsou, Frantzeskaki, and Malamis, 2023).

Adequate representation of hydrological dynamics and the effects of NbS on the watershed in models such as SWMM demands detailed characterization of watersheds or drainage areas, with spatial and physical data such as land cover, slope, etc (Niemi et al., 2019). Distributed models that incorporate spatial variability provide a more realistic representation of urban hydrological processes (Nguyen et al., 2016). Despite advances in hydrological modeling, the use of models with detailed input data often presents challenges, especially due to the volume and complexity of the information, particularly when working with multiple areas. In such cases, the process of data acquisition and organization requires greater care to avoid errors and ensure model quality.

Due to the complexity of preprocessing input data for hydrological models, implementing automated computational routines for this process constitutes an important strategy to reduce preparation time, minimize errors, and ensure data traceability. In this regard, the present study proposes a routine within the RStudio environment, using the R language, with the aim to facilitate the extraction, structuring, and organization of spatial and physical data necessary for hydrological modeling in software. The proposed routine seeks to optimize the workflow and provide an accessible, replicable, and efficient tool for researchers and professionals in water resources management.

2. MATERIAL AND METHODS

2.1. General structure of the computer routine

The routine consists of a set of modular scripts, each dedicated to a specific task within the hydrological modeling preprocessing workflow. Each script follows a standardized structure: loading packages, setting the working directory, importing input data with defined formats and coordinate reference systems (CRS), performing targeted operations (e.g., attribute extraction, topological analyses), and finally visualizing and exporting the results. These scripts are designed to be used as a complete routine or individually, as long as the necessary inputs, which also include the outputs of the previous processes, are available. Figure 1 presents the workflow diagram outlining the overall logic.

Figure 1 – Workflow diagram.

Table 1 presents the packages and functions commonly used across the scripts, grouped by processing steps such as setting the working directory, data import, CRS definition, result visualization, and export. The following subsections describe each script individually, according to its specific role within the modular preprocessing routine.

Table 1 – Packages and functions common to the scripts.

Packages	Functions	Description
Base R	*setwd()*; *dirname()*	Defines the working directory.
sf	*st_read()*; *st_crs()*; *st_transform()*; *st_as_sf()*; *st_write()*; *st_drop_geometry()*; *st_make_valid ()*; *st_is_empty()*	Import data; transform CRS; convert to sf; export GIS formats; remove geometry; ensure topologically correct geometries; check empty geometries.
writexl	*write_xlsx()*	Export dataframes to Excel files (.xlsx).
raster	*raster(); writeRaster()*	Import and export raster data.
ggplot2	*ggplot()*; *geom_sf()*; *theme_minimal()*; *aes()*; *labs()*	Create graphic; plot spatial data; minimalist theme; aesthetic mapping; add titles and subtitles.

2.2. Input Data

The input data required to run the routine includes spatial and physical information about the watershed. Table 2 provides a detailed summary of all input data required for the routine, specifying the type of information, a brief description of each item, and the respective formats accepted for processing. In addition, some of the outputs generated by the processing of the modular steps are also inputs for subsequent steps in the routine.

Table 2 – Input data, description and file format.

Input Data File	Description	File Format
Digital Terrain Model (DTM)	Terrain surface elevation	Raster (.tif)
Stream Network	Lines representing the watercourses.	Vector - Line (.shp)
Land Cover	Classification of land cover.	Raster (.tif)
Areas of Hydrological Interest	Areas for sub-basin delimitation, as NbS areas.	Vector - Polygon (.shp)

3. DEVELOPMENT AND APPLICATION OF THE ROUTINE

3.1. Outlets

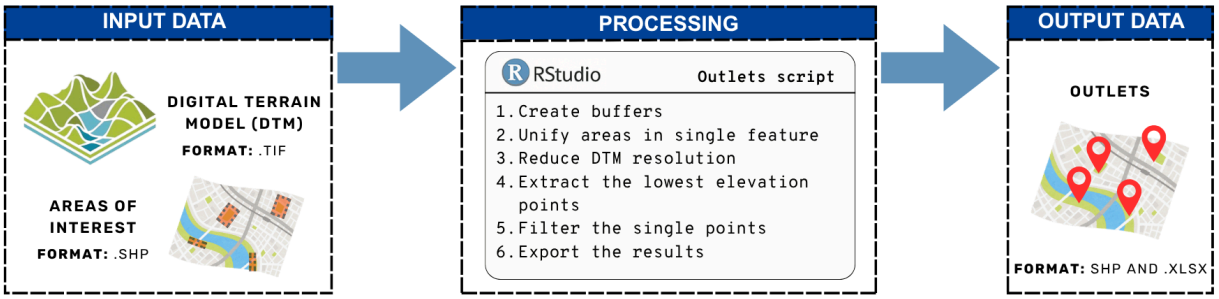
The identification of preferential outlet points within hydrological areas of interest is structured through an RStudio script using the packages *sf*, *raster*, *dplyr*, *ggplot2*, and *writexl*. The main objective of the code is to extract, within each area, the points with the lowest elevation, which represent outlets, contributing to the delineation of drainage areas. The spatial data processing begins with the application of the *st_buffer()* function to create buffers around the

geometries of interest, ensuring that narrow or linear features are properly represented. Next, a column is added to identify the type of area of interest for each simple feature object, and the layers are merged into a single feature using *bind_rows()* from *dplyr* package.

To optimize computational performance and processing complexity, the spatial resolution of the Digital Terrain Model (DTM) is adjusted using the *aggregate()* function from the *raster* package, which aggregates pixels by increasing their size according to the aggregation factor configured. Next, a list is created to store the lowest elevation points in each polygon of interest, with the number of points to be extracted predefined. The main processing involves an iterative loop controlled by the *nrow()* function, which iterates for each area of interest individually. For each polygon, the *mask()* function of the raster package is applied to crop the DTM, isolating the elevation values within that spatial extent.

Non-null elevation values are extracted using the *values()* and *st_zm()* functions (from the *raster* and *sf* packages, respectively) to check for sufficient data availability. When data is present, the indices of the "n" lowest elevation points are identified using the *order()* function. These indices are converted into geographic coordinates with *xyFromCell()*, transforming raster pixels into XY points in the DTM's CRS. To avoid duplicates, the *unique()* function is applied, prioritizing points with the lowest elevations, ensuring a reliable spatial representation of the outlet points. Each point is then converted into a spatial object with attributes (area ID, area type, point sequence number, and elevation value) and stored in a list. Finally, all points are merged into a single shapefile and a summary table for further analysis. Figure 2 shows the illustrated flowchart.

Figure 2 – Illustrated flowchart with input data, Outlets script processing steps and output data.



3.2. Drainage Areas

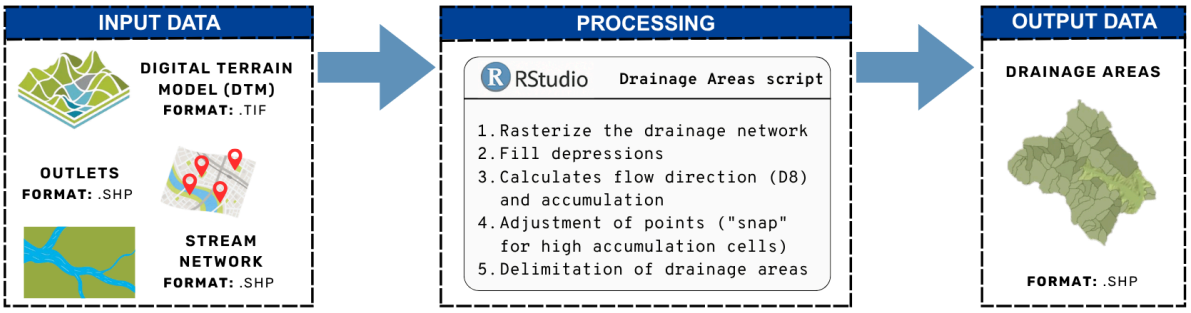
The delineation of the drainage areas associated with the output points is carried out using the *terra*, *sf*, *dplyr* and *whitebox* packages. Initially, the vector network of watercourses is rasterized using the *rasterize()* function to generate a mask in which the pixels of the watercourses are assigned a value of 1. These pixels are subsequently defined as NA in the modified DTM to ensure that surface runoff is adhered to the fluvial topology and to avoid trespassing into the channels of existing watercourses.

Hydrological pre-processing begins by applying the *wbt_fill_depressions()* function to correct the spurious depressions present in the DTM. The flow direction is calculated using *wbt_d8_pointer()*, which assigns each raster cell the neighboring cell that receives the flow. Flow accumulation is then calculated using *wbt_d8_flow_accumulation()*, quantifying the volume of flow accumulated per pixel. The output points are refined using *wbt_snap_pour_points()*, which adjusts their positions to the adjacent cells with the greatest flow accumulation within a predetermined threshold distance. Subsequently, drainage areas are delineated using *wbt_drainage_areas()*, which

identifies the raster cells that contribute flow to each outlet, thus defining discrete catchment polygons.

These raster drainage areas are converted to vector polygons using *as.polygons()* and subsequently transformed into simple feature objects via *st_as_sf()*. A spatial join is conducted with *st_join()* to associate each polygon with its corresponding outlet point, allowing identifiers to be assigned and facilitating further analyses. Finally, polygons related to the same outlet are aggregated and consolidated using *group_by()* and *summarise()* functions from the *dplyr* package. The resulting output comprises a set of validated vector polygons representing the delineated drainage areas for each hydrological outlet. Figure 3 shows the illustrated flowchart.

Figure 3 – Illustrated flowchart with input data, Drainage Area script processing steps and output data.



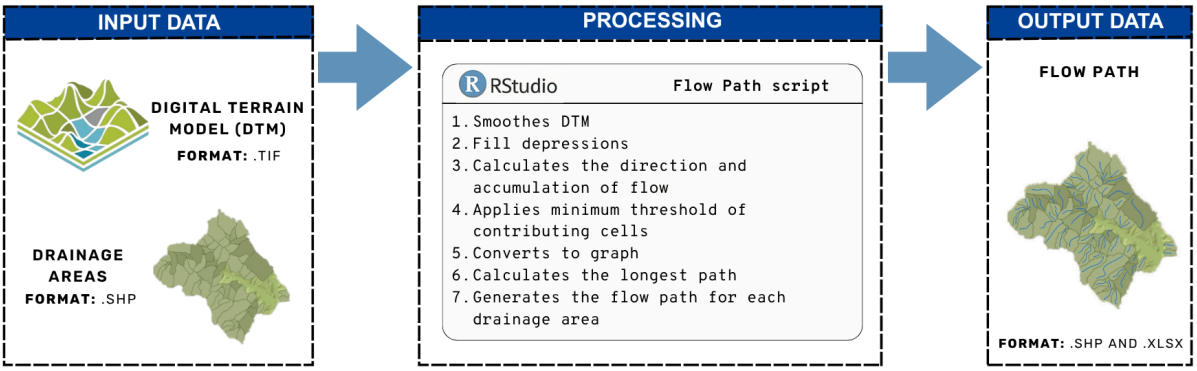
3.3. Flow Path

The developed script employs the packages *whitebox*, *terra*, *sf*, *igraph*, *dplyr*, *purrr*, *ggplot2*, and *writexl* to extract and analyze the flow paths, as well as identify the longest preferential flow paths within each drainage area polygon. The processing begins with reading the spatial data: DTM and drainage areas. The function *process_drainage_area()* is designed to execute the processing and vectorization steps for each drainage area. The function starts by selecting a drainage area based on the Name field and uses its geometry to crop the DTM using the *crop()* and *mask()*. The model is smoothed using *wbt_gaussian_filter()* to reduce noise and depressions are filled using *wbt_fill_depressions()* to ensure flow continuity

From the filled DTM, flow direction (*wbt_d8_pointer()*) and flow accumulation (*wbt_d8_flow_accumulation()*) files are generated. Based on this data, the raster drainage network is extracted through the *wbt_extract_streams()* function, where a minimum contribution threshold can be set to define waterways, controlling the sensitivity of identification. The rasterized drainage network is then converted into linear vector features with the *wbt_raster_streams_to_vector()* function. Next, to identify the longest path within the network, a graph structure is built using the *igraph* package. Each network segment is converted into a graph edge using the functions *map()*, *paste()*, and *tibble()*. From the graph generated by *graph_from_data_frame()*, a search is performed for the longest path, using the *shortest_paths()* function with negative weights.

The segments comprising the main flow path are extracted from the original network using *semi_join()* and *mutate()* to identify matching vertex pairs (from-to). The final result is saved as a simple feature, named according to the corresponding drainage area, for export. After running the *process_drainage_area()* function, results are returned as a list. This function is applied iteratively to all drainage areas using *purrr's map()*. The consolidated results are combined into a data frame with *map_dfr()* and exported as a spreadsheet and shapefile. Figure 4 shows the illustrated flowchart.

Figure 4 – Illustrated flowchart with input data, Flow Path script processing steps and output data.



3.4. Equivalent slope

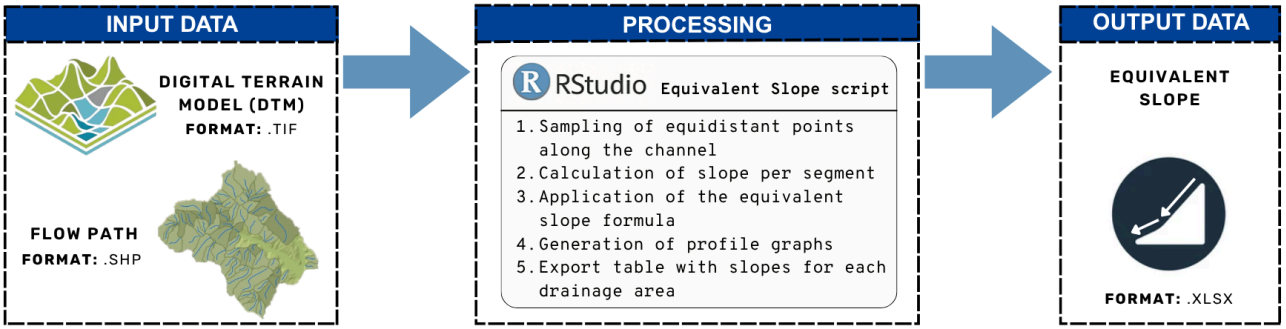
The script aims to calculate the equivalent slope along the longitudinal profiles of the flow paths. A Digital Terrain Model (DTM) and a vector layer of the flow paths are used as input data. Processing is carried out using the packages *terra*, *sf*, *ggplot2*, *dplyr*, *purrr*, and *writexl*. Initially, the script performs data import and verification — a step that ensures the correct association between each geometry and its respective identifier. It then defines the function *sample_continuous_points()*, which is responsible for generating elevation sampling points along each flow path, with a regular spacing defined by the user, as well as including the residual point corresponding to the final segment leftover after this regular division. Then, the function *process_flow_path()* uses the sampled points to extract the corresponding elevations for each flow path from the DTM. Based on this information, the slope of each segment (Di) between consecutive point pairs is calculated using the *base R diff()* function. With these data, the equivalent slope of the main channels is calculated according to equation (1) (DAEE, 2005).

$$I_{eq} = \left(\frac{\Sigma L_i}{\Sigma \left(\frac{L_i}{\sqrt{D_i}} \right)} \right)^2 \quad \text{where } D_i = \frac{\Delta Hi}{Li} \qquad (1)$$

Deq = equivalent slope (m/m); Li = horizontal length of each segment i (m); Di = slope of each segment i (m/m); ΔHi = elevation difference between the ends of each segment i (m).

With the functions defined, the script iterates over all the flow paths, applying the processing function to each one. In the final step, the script organizes and exports the data in a summary table containing, for each channel, its name, total length in meters, and equivalent slope as a percentage. Additionally, longitudinal profile graphs are generated, showing the variation in elevation along each flow path, with distance and elevation indications. Figure 5 shows the illustrated flowchart.

Figure 5 – Illustrated flowchart with input data, Equivalent Slope script processing steps and output data.

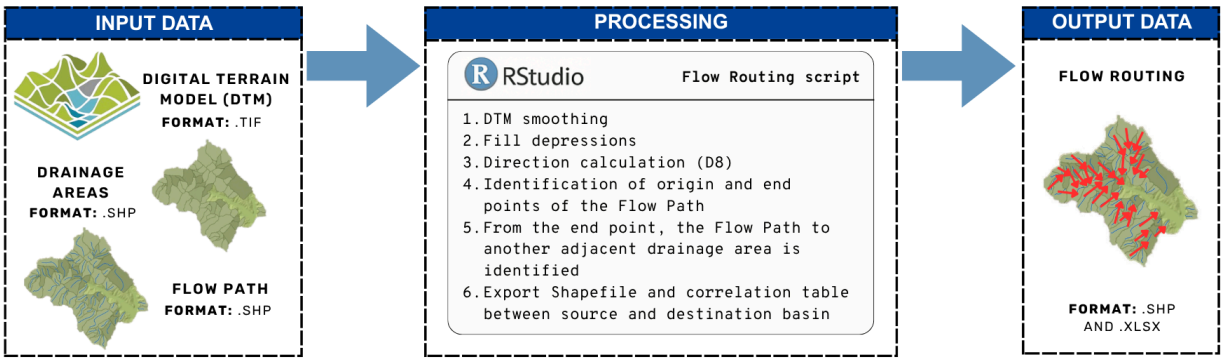


3.5. Flow routing

To characterize the hydrological connections between drainage areas through runoff, the "Flow Routing" script uses the *sf*, *dplyr*, *whitebox*, *ggplot2*, and *writexl* packages. Processing begins with the loading of the files: DTM, flow path vector, and drainage areas. The subsequent steps consist of smoothing the DTM (*wbt_gaussian_filter()*), filling depressions (*wbt_fill_depressions()*), and determining flow directions (*wbt_d8_pointer()*), are carried out in a similar manner to the "Flow Path" script.

The *get_extreme_points()* function identifies the origin and destination points of the flow paths (highest and lowest elevation, respectively) by extracting the coordinates and elevation values from the DTM. From the identified end point, the *find_destination_drainage_area()* function simulates the water path cell-by-cell on the flow direction raster, moving in the direction indicated by the D8 code. The D8 model codes the water movement toward one of the eight neighboring cells using discrete values: 1 for east, 2 for southeast, 4 for south, 8 for southwest, 16 for west, 32 for northwest, 64 for north, and 128 for northeast. Subsequently, the script goes through a sequence of conditionals to identify and update the position of the current cell, progressing until the path reaches a cell that intersects the drainage area receiving the flow. The functions described are applied to each flow path and the resulting information is associated with the corresponding drainage areas and converted into shapefile and spreadsheet format. Figure 6 shows the illustrated flowchart.

Figure 6 – Illustrated flowchart with input data, Flow Routing script processing steps and output data.



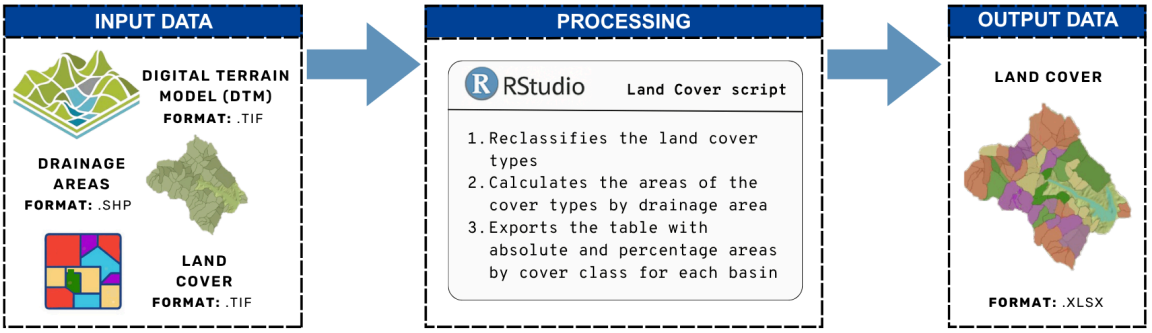
3.6. Land cover

The land cover analysis script for each drainage basin employs the *terra*, *sf*, *dplyr*, and *writexl* packages. The input data include a land cover raster and a shapefile of the drainage areas. Merging the drainage areas using *st_union()* from the *sf* package creates a single continuous polygon covering the entire basin, which is then used to crop the raster with the *crop()* and *mask()* functions. Next, the raster is reclassified using the *classify()* function, based on a matrix (reclass_matrix) that groups different land use classes into broader categories. In the current configuration, the categories include: Urban Area (1) – intensive anthropogenic uses such as cities, agriculture, and mining; Barren or Non-Vegetated Areas (2) – undefined or naturally non-vegetated areas; and Green Area (3) – native vegetation formations or reforested areas. These categories can be adjusted as necessary, depending on the analysis objectives or different land use and land cover classifications. The reclassified raster is then converted to vector format using *as.polygons()*, with the value column renamed to "Classes" using *rename()*.

The intersection between the land cover polygons and the drainage areas is performed with *st_intersection()*, and for each polygon intersected, the area in square meters is calculated using *st_area()*. A new column (LAND_USE_TYPE) is created using *case_when()* to translate the

reclassified values into meaningful labels. The *group_by()* and *summarise()* functions from the *dplyr* package are used to calculate the total area of each land use class per drainage basin. Then the percentage of each class within its respective basin is calculated and stored in the percent column. As a verification step, the percentage totals per basin are summed to ensure they approximately equal 100%, to identify possible data inconsistencies. Finally, the results are exported as a table. The reclassified raster can also be saved to disk using *writeRaster()*. The land cover categories are visualized on a map using *plot()*, with a color palette that differentiates the three categories. Figure 7 shows the illustrated flowchart.

Figure 7 – Illustrated flowchart with input data, Land Cover script processing steps and output data.



3.7. Results of applying the routine modules

To validate the proposed methodology, an urban sub-basin of the Água Comprida stream, located in the municipality of Bauru (SP), was selected to apply the routine. Figure 8 shows the result of the automatic generation of outlets, drainage areas and flow paths for the areas of interest in the sub-basin. Figure 9 illustrates the longitudinal profile generated by the "Equivalent Slope" module, with the respective flow path length information and the result of the slope calculation.

Figure 8 – Illustrative result of the application of the "Outlets", "Drainage Areas" and "Flow Path" scripts.

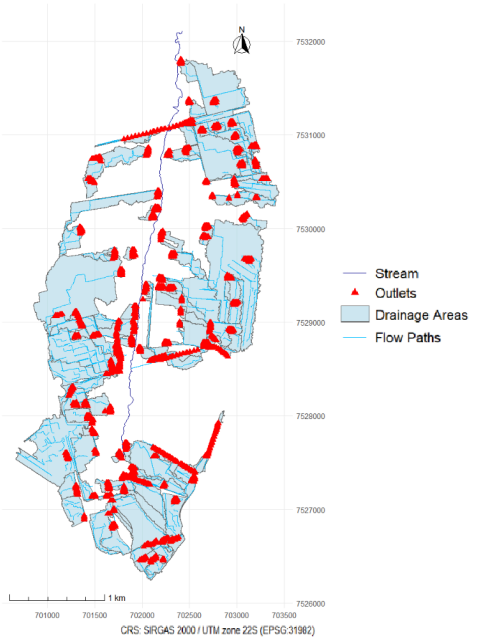
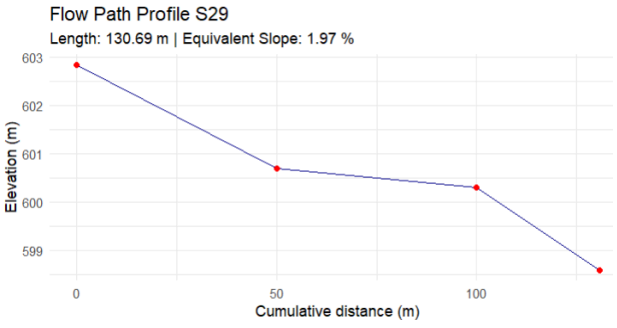


Figure 9 – Illustrative result of the practical application of the "Equivalent Slope" script.



4. DISCUSSION

The routine integrated tools for spatial analysis, hydrological processing, and data manipulation within the R environment, with the aim of reducing the time required to extract and organize spatial, physical, and hydrological information from areas of interest. The automation of these steps represents a significant advancement, especially in hydrological studies involving multiple sub-basins or large territorial extensions. The routine's modular structure allows easy parameter customization and adaptation to different geographic contexts, data sources, and analytical objectives, according to the user's data, goals, and scale of work. This flexibility considerably expands the routine's potential applications, making it suitable for both individual case studies and comparative analyses across different scenarios or regions.

However, it is important to highlight that the quality of the results generated depends directly on the quality of the input data. The spatial resolution of the Digital Terrain Model (DTM) and the quality and detail of the land use and land cover classification are critical factors that can significantly influence the accuracy of the derived variables. Therefore, although the routine provides a systematic and replicable process for data preprocessing, it does not replace the critical evaluation of the inputs used. Users should check the suitability of the data for the requirements of the hydrological model or the intended analysis.

5. CONCLUDING REMARKS

The automated routine developed in this study represents a practical advancement in supporting hydrological modeling, especially in contexts that require processing large volumes of spatial data. Its implementation in R, using widely adopted packages such as *sf*, *terra*, and *ggplot2*, ensures accessibility and ease of adaptation to different projects.By enabling the systematic extraction of variables such as elevation, slope, and land use, the tool helps optimize the workflow of researchers and technicians, promoting greater efficiency, reproducibility, and reliability in the preparation of input data. Future perspectives could include the incorporation of machine learning methods to enhance land use classification and the prediction of hydrological variables, as well as the development of graphical user interfaces to make the routine more accessible to users with varying levels of programming expertise. Additionally, integration with real-time data, such as hydrometeorological sensors and updated satellite imagery, could increase the routine's applicability in dynamic environmental monitoring and management.

6. DATA AVAILABILITY STATEMENT

All codes generated and used during the study are available in a repository online: https://github.com/anapascucci/preprocessingroutine.git.

REFERENCES

BAO, C.; FANG, C.-L. (2008). "Analysis of spatial and temporal variation of water resources on the intensity of urbanization constraints in arid regions". Journal of Geographical Sciences, 63, pp. 1140–1150.

CHEN, Y.; ZHONG, S.; LIANG, X.; LI, Y.; CHENG, J.; CAO, Y. (2024). "The Relationship between Urbanization and the Water Environment in the Chengdu-Chongqing Urban Agglomeration". Land, 13(7), 1054.

EL BAIDA, M.; CHOURAK, M.; BOUSHABA, F. (2025). "Flood Mitigation and Water Resource Preservation: Hydrodynamic and SWMM Simulations of Nature-Based Solutions under Climate Change". Water Resources Management, 39(3), pp. 1149-1176.

NGUYEN, P.; THORSTENSEN, A.; SOROOSHIAN, S.; HSU, K.; AGHAKOUCHAK, A.; SANDERS, B.; KOREN, V.; CUI, Z.; SMITH, M. (2016). "A high resolution coupled hydrologic–hydraulic model (HiResFlood-UCI) for flash flood modeling". Journal of Hydrology, 541(A), pp. 401-420.

NIEMI, T. J.; KOKKONEN, T.; SILLANPÄÄ, N.; SETÄLÄ, H.; KOIVUSALO, H. (2019). "Automated Urban Rainfall–Runoff Model Generation with Detailed Land Cover and Flow Routing". Journal of Hydrologic Engineering, 24(5), 04019011.

PEIXOTO, M. C. D. (2006). "Expansão urbana e proteção ambiental: Um estudo a partir do caso de Nova Lima/MG", in COSTA, H. S. de M. (Org.), Novas periferias metropolitanas: A expansão metropolitana em Belo Horizonte: Dinâmica e especificidades no Eixo Sul, Belo Horizonte: C/Arte, pp. [inserir páginas].

POSIT Team. (2024). RStudio: Integrated Development Environment for R. Posit Software, PBC, Boston, MA. URL http://www.posit.co/

QUININO, U. C.; CAMPOS, L. F.; GADELHA, C. L. (2000). "Avaliação da qualidade das águas subterrâneas na bacia do rio Gramame no Estado da Paraíba", in Anais do V Simpósio de Recursos Hídricos do Nordeste, Natal, Nov. 2000, vol. 1, pp. 162-176.

SÃO PAULO (Estado). Secretaria de Estado de Energia, Recursos Hídricos e Saneamento. Departamento de Águas e Energia Elétrica – DAEE. (2005). Guia prático para projetos de pequenas obras hidráulicas. São Paulo: DAEE, Capítulo 3: "Determinação da vazão de projeto".

TSATSOU, A.; FRANTZESKAKI, N.; MALAMIS, S. (2023). "Nature-based solutions for circular urban water systems: A scoping literature review and a proposal for urban design and planning". Journal of Cleaner Production, 394, 136325.

TUCCI, C. E. M. (2007). Inundações urbanas. Ed. ABRH-RHAMA, 393 p.

VILAÇA, M. F. et al. (2009). "Bacia hidrográfica como unidade de planejamento e gestão: o estudo de caso do ribeirão Conquista no município de Itaguara – MG", in XIII Simpósio Brasileiro de Geografia Física Aplicada, Viçosa, MG.

WOODS-BALLARD, B.; WILSON, S.; UDALE-CLARKE, H.; ILLMAN, S.; ASHLEY, R.; KELLAGHER, R. (2015). The SUDS manual. Londres: Ciria.

ACKNOWLEDGMENTS - The author thanks CAPES for the academic master's scholarship (process: 88887.962863/2024-00), and express their gratitude to Graziele Ruas and Simone Andrea Furegatti for their guidance during this research.