Nature's integration in cities' hydrologies, ecologies and societies

D2.2/M2.2 Spatially explicit modeling framework

23/06/2023

Lead partner: NIOO-KNAW

Authors: Lisette de Senerpont Domis, Francis Dullemond and Sven Teurlincx

Prepared under Biodiversa and WaterJPI joint COFUND call on "Conservation and restoration of degraded ecosystems and their biodiversity, including a focus on aquatic systems"

Project full title:	**Nature's integration into cities' hydrologies, ecologies and societies**
Project duration:	01.04.2022 – 30.06.2025 (42 months)
Project coordinator:	McKenna Davis, Ecologic Institute
Deliverable title:	Spatially explicit modeling framework
Deliverable n°:	2.2
Nature of the deliverable:	Report
Dissemination level:	Confidential
Lead partner:	NIOO-KNAW
Citation:	Lisette de Senerpont Domis, Francis Dullemond and Sven Teurlincx (2025). Spatially explicit modeling framework NICHES Deliverable D2.1.
Due date of deliverable:	Month 42
Submission date:	Month 42

Reviewed by:

Name	Date
Matt Eckelman	27 06 2025

Deliverable status:

Version	Status	Date	Author(s)
2.0	Final	June 27 2025	Lisette de Senerpont Domis, Francis Dullemond and Sven Teurlincx

This project was funded through the 2020-2021 Biodiversa and Water JPI joint call for research proposals, under the BiodivRestore ERA-Net COFUND programme, and with the funding organisations: German Federal Ministry of Education and Research; Agencia Estatal de Investigación; Ministry of Agriculture, Nature and Food Quality of the Netherlands.

Acknowledgements

We would like to thank Svetlana Khromova for useful input for the use of BATT in the modeling framework.

Table of Contents

1 Preface

This background of this report lies in developing an up scalable, open-source approach that allows for quantitative assessment of the effectiveness of NBS on reducing pollution load of stormwater to receiving waters. Predicting possible future ES delivery is key in the urban waterscape in relation to sewage overflow events and extreme rainfall, as they are expected to change markedly in the future. Simultaneously, as combined sewage systems are technical systems, they offer the ability to be adapted in such a way that the impacts of CSOs on aquatic ecosystems and the ecosystems services they provide are mitigated.

2 Summary

NICHES advances scientific knowledge on restorative NBS through the application and testing of impact assessments, models and transitional governance models for improved urban drainage in five cases within and beyond Europe. The project hypothesizes and aims to demonstrate that sustainable transformations of cities based on restorative NBS which enhance water retention capacities in urban areas could widely mitigate impacts from combined sewers on aquatic ecosystems. As urban catchment is part of a multi-owner landscape with associated stakeholder conflicts linked through teleconnections and multi-scale governance structures, the involvement of diverse stakeholders and their values from the NICHES core cities is vital to co-design the impact assessment and ES module design and to ensure maximal applicability. This deliverable describes the development of a modeling framework that allows for assessment of the effectiveness of urban Nature Based Solutions on aquatic ecosystem services provisioning. In short, we build on an existing framework where ecosystem service delivery is determined based on threshold values of water quality and ecological variables (Seelen et al., 2022; Zhan et al., 2023). Rather than determining these variables from field-based measurements we retrieve them from an aquatic ecosystem model, PCLake+. To enable the evaluation of storm water best practices on receiving water, we forced PCLake+ with the BATT tracking tool developed by the US Environmental Protection AgencyBATT. This spreadsheet tool specifically estimates the removal of pollutants such as phosphorus, nitrogen, and sediment from stormwater. To allow for upscaling to a European scale, we used open data sources such as Corine Land Cover, the European Soil Database, the OpenStreetmap database and the HydroSHEDS database as an input for BATT. We validate this approach against Water Framework monitoring data from the Province of Zuid Holland, where the NICHES case study Rotterdam is located. This validation shows that our modeling framework performs reasonably well at capturing concentrations of dissolved oxygen and water transparency, whereas the simulation of concentrations of total nitrogen, total phosphorus and Chl-a needs improvement. Furthermore, our modeling exercise for 25 lakes in the Province of Zuid Holland also shows that – even in an unrealistic scenario where Nature Based Solutions were applied at every potential location- NBS have no significant effect in reducing the pollution load to receiving waters. Even though the current validation shows that there is room for improvement in further finetuning the modeling approach, it clearly shows the value of our approach lies that is up scalable to the European spatial context, open

source and allows for quantitative underpinning of the effectiveness of NBS on aquatic ecosystem services provisioning.

3 List of abbreviations

EU	European Union
ES	Ecosystem Service
CSO	Combined Sewage Overflow
AEM	Aquatic Ecosystem Model
WFD	EU Water Framework Directive
NBS	Nature Based Solution
BATT	Stormwater Best management practices Accounting and Tracking Tool developed by the EPA-USA
CLC	Corinne Land Cover
BMP	Best Management Practices
OSM	OpenStreetMap
ESDAC	European Soil Data Centre
BAU	Business as Usual scenario
NRMSE	Normalized Root Mean Square Error (by the mean)

4 Introduction

Healthy freshwater ecosystems can provide vital ecosystem services (ESs), but this capacity may be hampered due to water quality deterioration and climate change. In the urban waterscape combined sewer overflows (CSOs) form a direct threat to the quality of aquatic ecosystems and the species that inhabit them. Additionally, many of the services that the urban populace depends on (e.g., recreational fishing, swimming, carbon and nutrient retention) are threatened by loss of ecological quality caused by CSOs. CSO events are expected to increase with increasing intensity in rainfall due to climate change (van der Werf et al., 2023) and hence there is a need to understand how increased CSO events will impact both ecological functioning as well as service provisioning.

There is an urgent need to identify new solutions for reducing the impact of increased precipitation both on sewage systems and aquatic ecosystems. Nature-Based solutions (NBS) offer an alternative to the existing engineered stormwater management systems, having the potential to alleviate pressure during high rainfall events while also providing wider societal and environmental co-benefits, including increase in local biodiversity, enhance social well-being of residents and improving the aesthetics of the built environment (Chelli et al., 2025). Widespread uptake of NBS may be hampered due to lacking evidence of performance and co-benefits, approaches and targeted guidance that take the wider social-ecological-technological system (SETS) into account. NICHES aims to fill this gap by defining a holistic SETS framework for understanding restorative NBS for urban runoff mitigation and the resultant reduction of impacts on aquatic systems and resulting ecosystem services.

In the context of CSOs, there are characteristics of the sociological, ecological and technological urban waterscape systems that determine vulnerability to CSO events of receiving water bodies, its exposure to CSO events, and its capacity to adapt to CSO events (Figure 1). Ecosystem services provide a link between the ecological and the socio-technological system, and a quantitative understanding of ES provisioning under CSO events will provide us with a deeper understanding of SETS in the context of urban waterscapes. In the NICHES project, we focus on the ecosystem services that the aquatic system provides, which are described in more detail in D2.1.

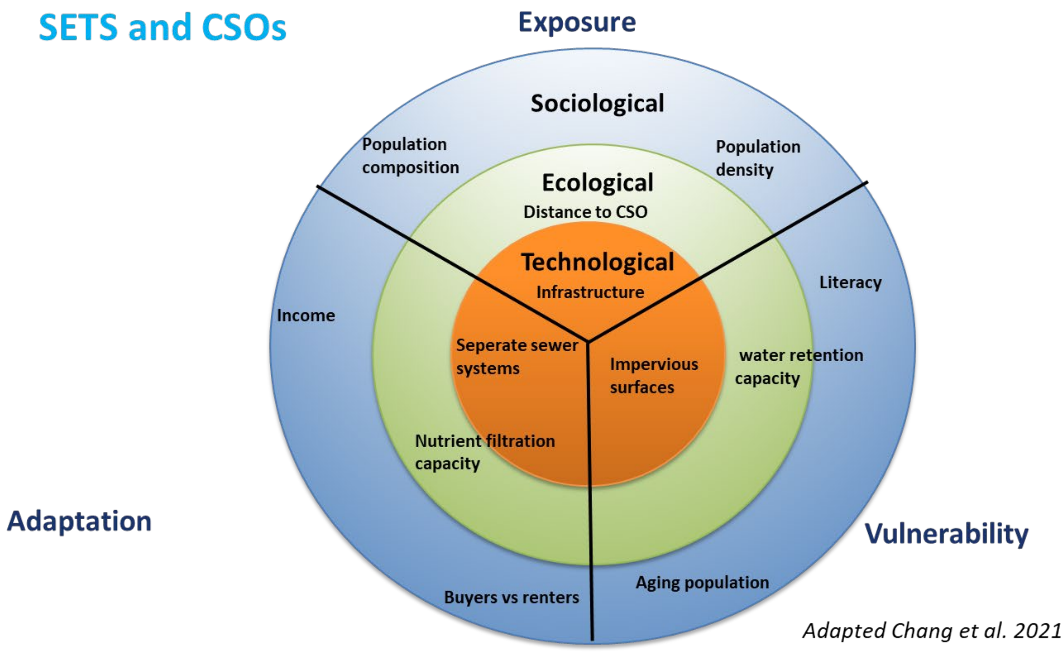

Figure 1 Scheme illustrating how the three systems of SETs (sociological, ecological and technological) relate to CSO events and how different aspects of the systems are influencing aspects present in a risk framework approach. Figure adapted based on (Chang et al., 2021). Increasing the nutrient filtration capacity of urban waterscapes through constructed wetlands and wadis could be viewed as a Nature Based Solution application.

4.1 Using aquatic ecosystem models to quantify stormwater impacts on receiving waters

Quantifying ecosystem services can be instrumental in recognizing the benefits humans receive from ecosystems, providing stronger arguments for ecological restoration (Grizzetti et al., 2019; Guerry et al., 2015). Conveying restoration impacts in terms of the loss or gain of ESs can facilitate effective communication of restoration outcomes to policy-makers and river basin authorities responsible for implementing restoration measures (Wortley et al., 2013).

While modeling terrestrial ecosystem services often focuses on mapping ESs provisioning through spatial variations of catchment attributes (e.g., land use, topography, lithology) (Nelson et al., 2009), the non-linear dynamics of water quantity and quality necessitate a more explicit consideration in aquatic ecosystem service modeling (Grizzetti et al., 2016).

There is increasing evidence that freshwater ecosystem services provisioning is closely linked to the ecological quality (or ecological state) of different aquatic environments, including shallow lakes (Janssen et al., 2021), deep lakes (Seelen et al., 2022), rivers, and coastal waters (Grizzetti et al., 2019). Based on data reported under the European Water Framework Directive (WFD), Grizzetti et al. (2019) demonstrated that higher provisioning of ESs is mostly correlated with more desirable ecological states (i.e., clear, submerged plant dominated waters), particularly for regulating services (e.g., water purification, erosion retention, flood protection) and cultural services (e.g., recreation). However, current modeling tools for water-related services primarily focus on water quantity (Grizzetti et al., 2016), with limited integration of services closely related to water quality (Keeler et al., 2012).

Water quality dynamics are mediated by complex interactions among a myriad of ecosystem processes, which are often oversimplified in large-scale modeling frameworks. For instance, one widely-used ecosystem service model, InVEST, simplifies by using nutrient loading as a proxy for determining the availability of lake-related ESs (Nelson et al., 2009; Polasky et al., 2011), assuming simple linear responses of ecosystems to nutrient loading. This approach contradicts the resistance theory of (Gómez-Baggethun & Ruiz-Pérez, 2011; Ibelings et al., 2007), which supports threshold-type ecosystem responses to pressures. Consequently, the assessment of management actions within InVEST often relies on variables collected at the landscape scale (Burkhard et al., 2012), which may be inaccurate due to the aforementioned nonlinear responses or ill-fitting when assessing the impacts of in-lake restoration measures (Lürling & Mucci, 2020). Keeler et al. (2012) proposed a conceptual framework linking ecological-related services with corresponding water quality variables based on a review of existing ES models, emphasizing the importance of this link in assessing management actions. Given the long-history of development of AEMs (Janssen et al., 2015), linking water quality variable outcomes of these models to ESs provisioning approaches is a logical next step to capture the full dynamics of how water quality dynamics impacts ESs. As an input, AEMS require nutrient loadings and a water budget, that can be derived from catchment or watershed models (Clopin et al., 2025).

5 Development of a spatially explicit modeling framework

The modeling workflow we adopted in D2.2 is displayed in Fig. 1 and is fully open source. We used the AEM PCLake+ coupled to an ecosystem services (ES) module as described in D2.1 to quantify aquatic ecosystem service provisioning. To allow for the evaluation of different best management practices on reducing the pollution load of storm water to receiving waters, we forced this AEM-ES with output from Stormwater Best management practices Accounting and Tracking Tool developed by the EPA-USA (BATT). In order to be able to upscale to the European Scale, we used open pan-European data sources such as HydroSHEDS, we used open data sources such as Corine Land Cover, the European Soil Database, the

OpenStreetmap database and the HydroSHEDS database as an input for BATT. Below we describe in detail the different components of the modeling flow.

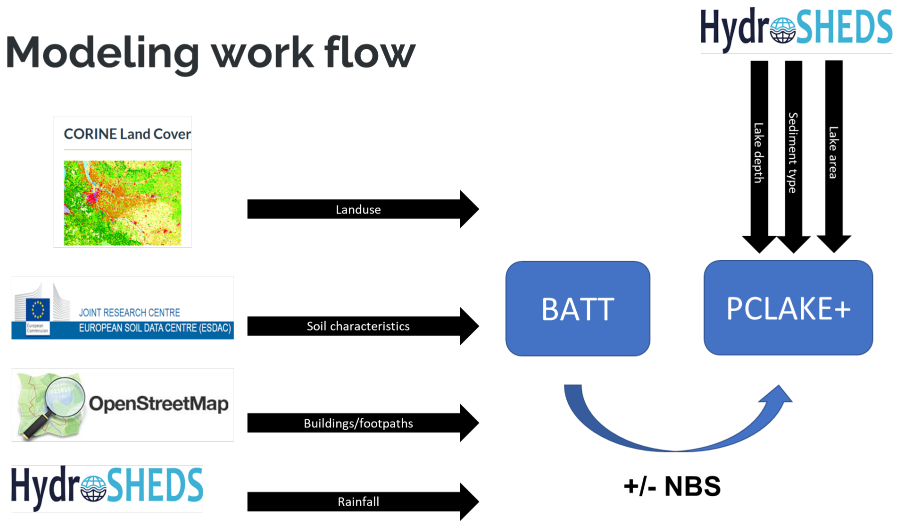

Figure 2: Modeling workflow adopted in D2.2, with open data sources Corine Land Cover, the European Soil Database, the OpenStreetMap as well as HydroSHEDS used as an input for BATT and PCLAKE+.

5.1 BATT

BATT (Best Management Practice Accounting and Tracking Tool) is a spreadsheet-based tool developed by the United States Environmental Protection Agency to support accounting, tracking, and reporting of pollutant load reductions, specifically focusing on nutrients (nitrogen and phosphorus) and sediment. It provides a user-friendly interface for documenting and quantifying the effectiveness of Best Management Practices (BMPs) implemented in a watershed or project area. Figure 3 displays the set-up of the tool.

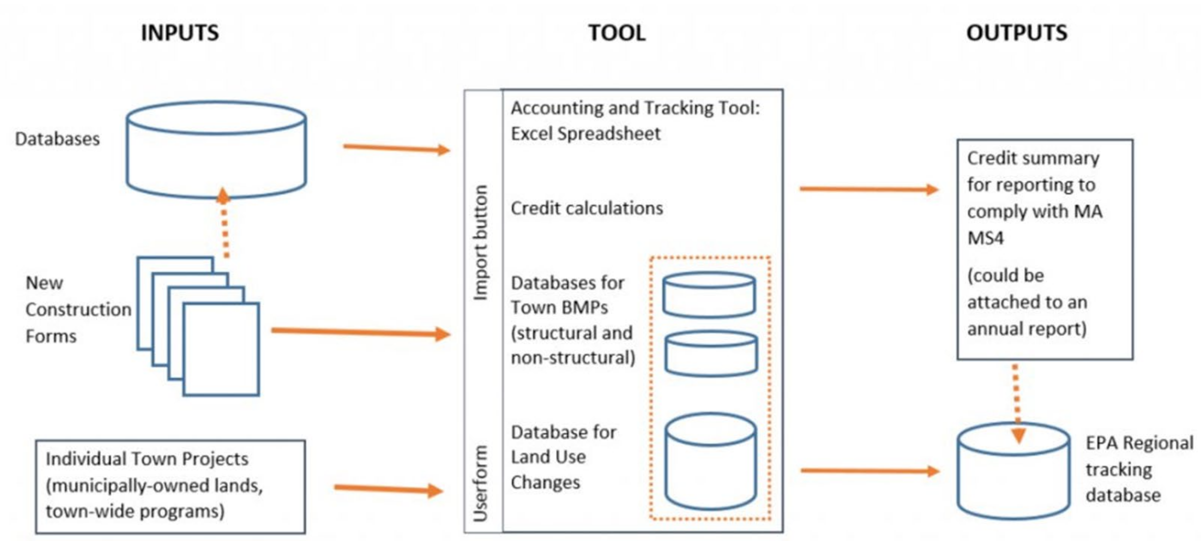

Figure 3: Set-up of the spreadsheet tool BATT

Rather than using the spreadsheet tool itself, we rebuilt the tool using the relationships and tables underlying the tool. BATT requires input regarding land cover (impervious vs non-impervious substrate), drainage area, soil characteristics (e.g., soil infiltration capacity, and the characteristics of the NBS (cf. structural BMPs in BATT). Below we describe in more detail what type of input data we used for BATT.

5.1.1 ESDAC Soil Database

The ESDAC Soil Database is a comprehensive collection of soil-related data maintained by the European Soil Data Centre (ESDAC), which operates under the Joint Research Centre (JRC) of the European Commission. It serves as the central hub for soil data and information in Europe. We used the Topsoil physical properties for Europe database (based on LUCAS topsoil data) to acquire the Hydrological Soil Groups (HSG) required for BATT. This database (Ballabio et al., 2016) has a resolution of 500m and contains 7 soil property maps that have been derived using soil point data from the LUCAS 2009 soil survey (around 20,000 points) for EU-25, using hybrid approaches like regression kriging. The soil map was originally a raster file (GeoTIFF). It was converted to polygons using the *Raster to Polygon* tool of ARCGIS Pro. In the Annex (8.1) you can find the table where the BATT Hydrological Soil Groups are mapped on the ESDAC soiltypes.

5.1.2 CORINE land cover data

CORINE Land Cover (CLC) is a standardized land use/land cover (LULC) dataset developed by the European Environment Agency (EEA) as part of the CORINE (Coordination of Information on the Environment) program. It provides consistent and comparable land cover data across European countries for environmental analysis, spatial planning, and monitoring land change over time. It provides a pan-European CORINE Land Cover inventory for 44 thematic classes for the 2018 reference year. The dataset has a Minimum Mapping Unit (MMU) of 25 hectares (ha) for areal phenomena and a Minimum Mapping Width (MMW) of 100 m for linear phenomena and is available as vector and as 100 m raster data. We used the vector dataset (https://doi.org/10.2909/71c95a07-e296-44fc-b22b-415f42acfdf0) to map BATT land cover types on CORINE categories, enabling nutrient calculations based on the area of each land cover class. The BATT land use conversion table can be found in the annex (8.2). Furthermore, to acquire the required hydrological curve numbers, the BATT Land use/Landcover combinations were mapped on Corine Land Cover types. The runoff curve number is based on the area's hydrologic soil group, land use, treatment and hydrologic condition, and is an empirical parameter used in hydrology for predicting direct runoff or infiltration from rainfall excess.

5.1.3 OpenStreetMap data

We used OpenStreetMap data to define the spatial extent of the BATT Nature Based Solutions. OpenStreetMap (OSM) is a free, open map database, built by a community of volunteers who use GPS devices, aerial imagery, field surveys, and local knowledge to map everything from roads and rivers to buildings, parks, bike lanes, hospitals, and more.

The Overpass API was used to retrieve the relevant OSM data: https://overpass-api.de/api/interpreter. A bounding box was created based on each watershed (see 5.2.2 for

details on watershed delineation), which served as the spatial extent for querying relevant OSM features.
Requests were made to the Overpass API using specific key–value pairs to extract relevant datasets:

- Pathways:
  - Key: highway
  - Values: 'footway', 'living street', 'pedestrian', 'sidewalk', 'cycleway', 'motorway'
- Buildings:
  - Key: building
  - Values: 'residential', 'apartments', 'terrace', 'house', 'detached', 'annex', 'hotel', 'semidetached house', 'commercial', 'industrial', 'office', 'retail', 'supermarket', 'warehouse', 'college', 'government', 'university'

The retrieved OSM data was then spatially matched to the watershed (see 5.2.2. for details on delineation for watersheds) for further analysis. In this study we focused on two BATT NBS, i.e., grass swales (Fig. 4) and gravel wetlands. Grass swales convey runoff through an open channel vegetated with grass. The primary removal mechanism is infiltration as runoff flows are conveyed. In the NICHES project, we assumed that bioswales were implemented along each pathway as present in the OSM database within the watershed, which is an unrealistic scenario (Sarabi et al., 2020). Space constraints, property ownership complexities, lack of financial incentives, design standards and uncertainty of functionality and performance have been identified as important barriers for the large-scale implementation of urban NBS. We did this to assess the maximum pollution reduction that can be achieved through constructing grass swales.

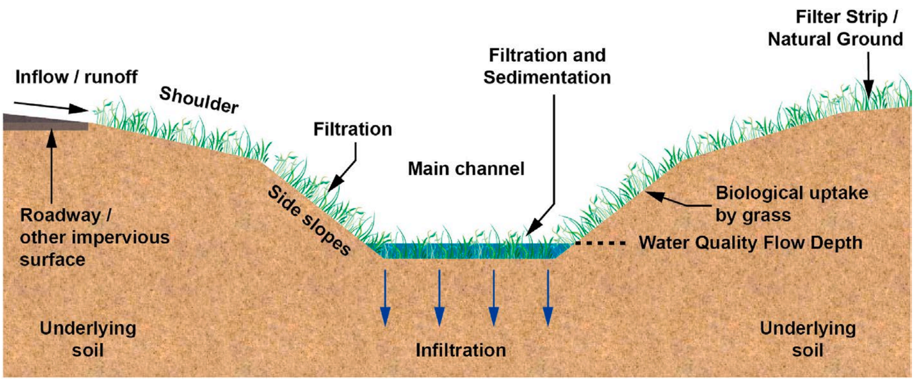

Figure 4: Typical grass swale cross-section and pollutant removal mechanism (Ekka et al., 2021)

The gravel wetlands are based on a design by the University of New Hampshire (UNH) Stormwater Center (UNHSC), see Fig. 5. Gravel wetlands provide a temporary surface ponding storage of runoff in a vegetated wetland cell that is eventually routed to an underlying saturated gravel internal storage reservoir (ISR) for nitrogen treatment. The outflow is controlled by an elevated orifice that has its invert elevation equal at the top of the ISR layer and provides a retention time of at least 24 hours. BATT assumes that 8 times the surface area of a grass swale will run off through the swale.

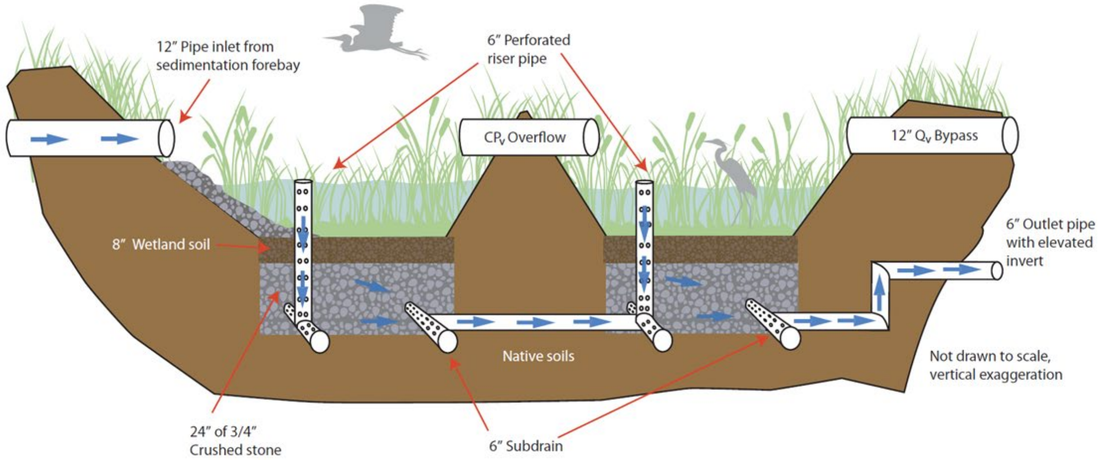

Figure 5: Gravel wetland design by UNH Stormwater Center (UNHSC) Source: https://scholars.unh.edu/cgi/viewcontent.cgi?article=1013&context=stormwater

For NICHES we assumed an unrealistic scenario where gravel wetlands were implemented along the length of each building present in the OSM database (i.e., the longest side of the building) within the watershed. We assumed the following surface dimensions of the gravel wetland, i.e., length of the building (m) x 0.5 (width of the gravel wetland). Furthermore, we assumed that the gravel layer was 0.1 m deep and had a soil porosity factor of 0.4. Similar to the implementation of grass swales, we ran this unrealistic scenario to assess the maximum pollution reduction that can be achieved through constructing gravel wetlands.

We contrasted both NBS with a "business as usual" (BAU) scenario, where BATT nutrient calculations were carried out in the absence of a nature-based solution implementation.

5.2 PCLake+-ES

PCLake+ is a process-based ecological model that was developed to simulate water quality and assess the trophic state of lakes based on ecological interactions (Janse, 2005, p. 200; Janssen et al., 2019). The model is a 0D model and assumes either a fully mixed water column connected to a sediment layer, or a two-layer water column differentiating between epilimnion and hypolimnion when water is stratified. It models nutrient cycling including nitrogen and phosphorus and a simple food web consisting of three functional groups of phytoplankton (cyanobacteria, green algae and diatoms), zooplankton, and fish. The model is widely used to assess effective management strategies for water bodies in the Netherlands and worldwide (Andersen et al., 2020; Janse et al., 2008; Wang et al., 2019). The model can capture well the state-shifts that can occur in inland waters, when nutrient loading forces a system to transition from a clear macrophyte-dominated state to a turbid phytoplankton-dominated state. In shallower systems, this state-shift is a step-change happening at a specific nutrient loading, defined as the critical nutrient loading. Importantly, due to a process called hysteresis, this step-change happens at a different transition point from clear to turbid, then from turbid to clear. As such the system can be in two alternative states. In deeper systems, however, this transition happens more gradually.

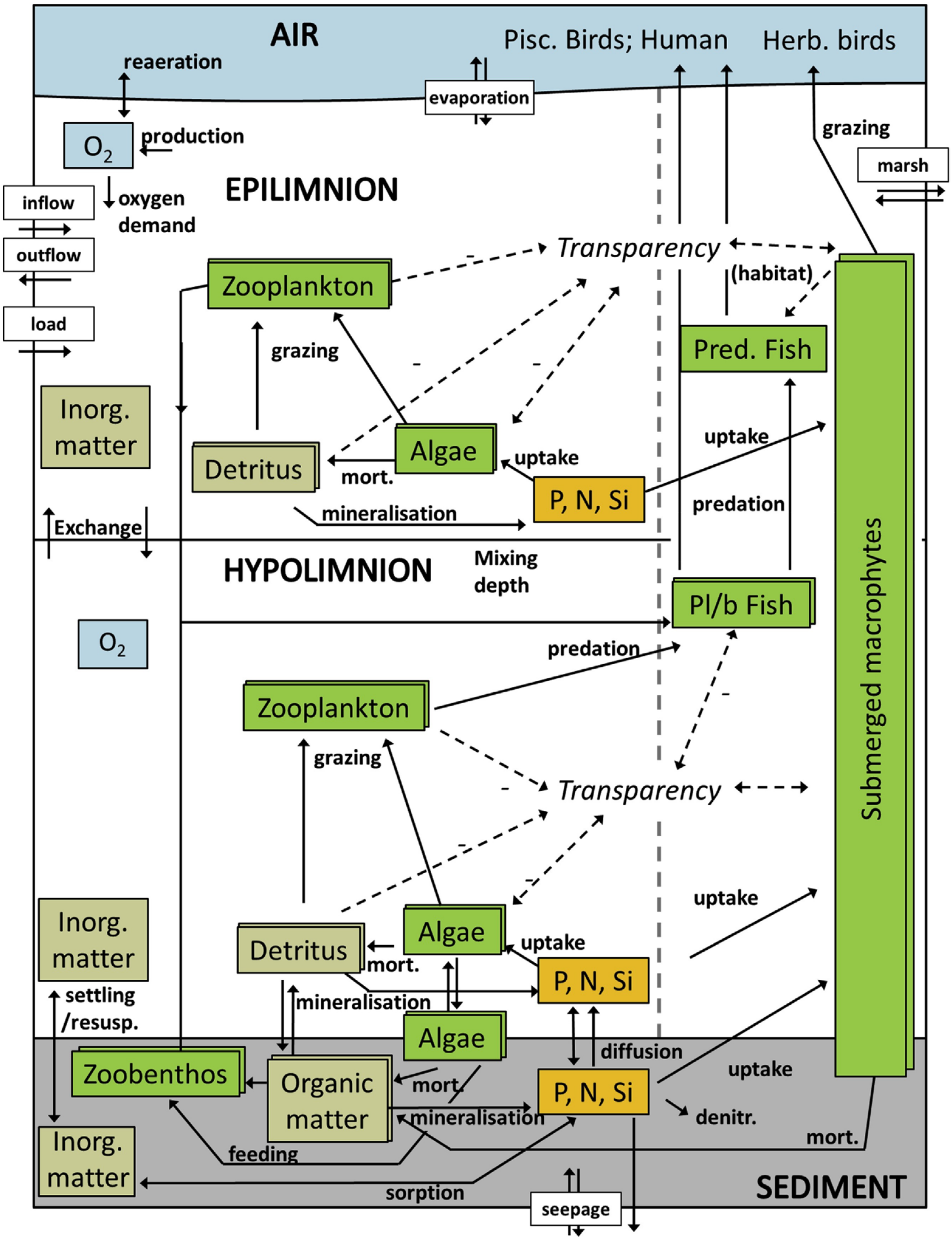

Figure 6: Schematization of a stratifying lake in PCLake+ with the water column divided in two layers: epilimnion and hypolimnion. For ease of comparison, this scheme of PCLake+ is designed similar to the scheme of the original PCLake model published in Janse (2005) and later updated by (Seelen et al., 2022; Zhan et al., 2023). An important addition to the original PCLake is the inclusion of a hypolimnion layer. Furthermore, all water state variables on the left side of the vertical grey dashed line were duplicated so that they are represented in both the epilimnion and hypolimnion; variables on the right side of the vertical grey dashed line were each captured in a single state variable for the entire water column

The model has also been used to estimate impacts on ecological and water quality of climate change and changing socio-economical scenarios (Mooij et al., 2007; Yang et al., 2022). Here, we expanded the model with a threshold-based ecosystem service delivery (Seelen et al., 2022; Zhan et al., 2023) based on its existing ecological outcomes. The expansion has been described in detail in NICHES D2.1. In short, we link ecosystem state indicators with ecosystem service provisioning through a threshold approach. The threshold values reflect the values that certain water quality parameter required to support the provision of a given service. In the ES module, the suitability of delivering each ES was expressed by an indicator function ranging between 0-1, with "1" representing a fully suitable provisioning, "0" representing an unsuitable provisioning, and values in between representing a moderate suitability.

5.2.1 Inputs to the model

To run our developed coupled AEM-ES PCLake+ model several input parameters are required. While PCLake+ has over 500 parameters, a large part of these parameters does not need to be changed by users as they result from the generic calibration of the model (Janse et al., 2010). Users are primarily required to define the boundary conditions of their own water system in terms of inflows (water and nutrients), climate and meteorology (precipitation, evaporation, irradiance) and lake properties such as depth, lake area and fetch (Figure 5). Water temperature can be estimated based on simple parameters defining variation around a mean temperature, or time series of water temperature from either measurements or physical lake models (e.g. Flake (Kirillin et al., 2011), General Lake Model (Hipsey et al., 2019)) can be used. The required inputs to the model closely align with the type of information available from climate models and the type of knowledge gathered in the construction of river basin management plans in the WFD.

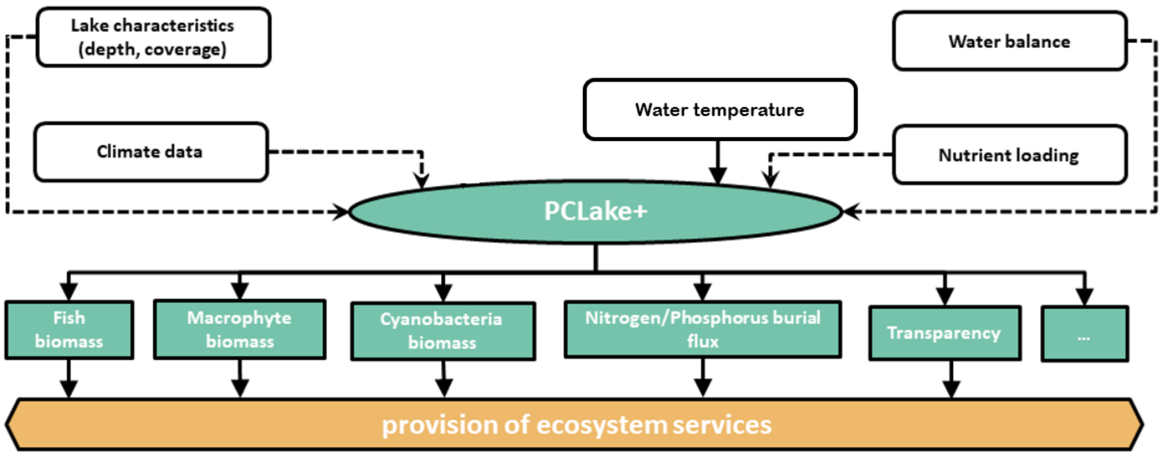

Figure 7: Model chain for ecosystem service modeling (Zhan et al., 2023). Rectangles denote state variables, ovals denote models, hexagon denotes ecosystem service module, rounded rectangles denote input data, solid arrows denote model input or output, dashed arrows denote data input. (PCLake+ in green, input in white, output in orange).

As an up scalable approach was a key requirement of our NICHES spatially explicit modeling framework, we used – in addition to the BATT output, open-source data on lake characteristics and climate data as well as on water balance were used. PCLake + has a large set of parameters (>250), making overfitting the model a risk when subjecting it to site-specific calibration when data is not abundantly present. Hence, we rely on the generic calibration for our study and only adjust boundary conditions of the lake, (i.e., depth, hydraulic and nutrient loads, climate forcing, wind fetch, etc.). Table 1 describes what databases were sourced to acquire the required PCLAKE input. Apart from BATT input, from which we derived nutrient runoff concentrations for the different NBS scenarios, we used the HydroSHEDS databases (see 5.2.2 for details).

Table 1: Overview of input parameters to PCLAKE+ES and their data sources.

Category	Parameters	Description	Source
Water balance	Qin	Water inflow to the lake	LakeATLAS
	Res_time	Residence time	LakeATLAS
	Depth_avg	Average Lake Depth	LakeATLAS
	Lake_area	Surface area of lake	LakeATLAS
Nutrient loading	fPLOAD_TOTAL	Total phosphorus loading to the lake	BATT, LakeATLAS
	fNLOAD_TOTAL	Total nitrogen loading to the lake	BATT, LakeATLAS
	fTLOAD_TOTAL	Total Suspended Solids Loading to the Lake	BATT, LakeATLAS
Climate data	tmp	Monthly temperatures	LakeATLAS
	Pour_lat	Latitude of Lake	LakeATLAS
	pre_mm_uyr	Yearly rainfall data	LakeATLAS
Lake characteristics	Sand_fraction	Fraction of sand in the lake sediment	LakeATLAS
	Clay_fraction	Fraction of clay in the lake sediment	LakeATLAS

5.2.2 HydroSHEDS databases

The HydroSHEDS database offers a suite of global digital data layers in support of hydro-ecological research and applications worldwide. Its various hydrographic data products include catchment boundaries, river networks, and lakes at multiple resolutions and scales. For developing the spatial explicit NICHES framework, we made use of the LakeATLAS database (Lehner et al., 2022), and the BasinATLAS (Linke et al., 2019). HydroATLAS has been created by compiling and re-formatting a wide range of hydro-environmental attributes derived from existing global datasets in a consistent and organized manner. The resulting data compendium offers attributes grouped in seven categories: hydrology; physiography; climate; land cover & use; soils & geology; and anthropogenic influences. For each of the sub-datasets, HydroATLAS contains 56 hydro-environmental variables, partitioned into 281 individual attributes.

We used the LakeATLAS to derive monthly temperatures, lake surface area, average lake depth, residence time, and watershed area. LakeATLAS aims to provide data on all global lakes with a surface area of at least 10 ha. The inflow to the lakes (Qin) was calculated based on the residence time and the lake depth (See Annex, 8.4 R-script Spatial Modeling Framework). We assumed that lakes maintained a constant water level with inflow equaling outflow. We used BasinATLAS to delineate the watershed area. This watershed was intersected with the Corine landcover data to allow for calculation of the nutrient loadings to the receiving water using BATT. To ensure that the size of a watershed area as recorded in the LakeATLAS was aligned with the size of the watershed calculated using the BasinATLAS, we performed an optimization by creating a buffer polygon around the BasinATLAS watershed polygon until the deviation between the two size values was 0.001.

6 Model validation and output

As the HydroSHEDS LakeATLAS contains ~70,000 lakes, and validation of our spatial modelling framework at such an extensive spatial scale was beyond the scope of the NICHES project, we first applied our spatial modelling framework to the area of Zuid Holland. This province, where the NICHES case Rotterdam is located, has a population of over 3.8 million as of January 2023 and a population density of about 1,410/km², making it the country's most populous province and one of the world's most densely populated areas. Whereas the LakeATLAS only contained 5 lakes in the larger Rotterdam area, the area of Zuid Holland contains 43 lakes. The provincial boundary of Zuid Holland was obtained from the CBS Provincie Actueel shapefile. Using an attribute query of ARC GIS Pro, “Zuid-Holland” was selected based on the statnaam field. The selected province was then exported as a separate feature layer using the *Export* Features tool of ARC GIS Pro. This polygon served as the basis for *intersect* and *clip* operations with other datasets in the subsequent analysis as described above (Section 5).

Using the set-up as described above we ran PCLAKE+ ES for a period of 30 yrs. (until equilibrium) and evaluated the impact of a business-as-usual scenario (BAU), a scenario with maximum implementation of grass swales, and a scenario with maximum implementation of gravel wetlands. We ran the model starting from a clear state, as well as from a turbid state. Of the 43 lakes in the LakeATLAS databases only 42 lakes could be modeled using the spatial modeling framework, as lakes that had a watershed area smaller than the lake surface, as well as coastal lakes that did have soil data for less than 90% of the watershed were automatically discarded (Fig. 8).

6.1 Model validation

We used the Water Framework Directive monitoring data (WFD) for the validation of our model output, using the selection of parameters suggested by Zhan et al. (2023). For each of the WFD monitoring locations within the Province of Zuid-Holland the monitoring data of the most recent WFD reporting year, i.e., 2023, were downloaded, and summer averages were calculated for water transparency (m), total nitrogen (mg/L), total phosphorus (mg/L), and dissolved oxygen (mg/L). In line with the definition of the WFD, the summer months covered April 1-September 30, and with monitoring typically taking place on a biweekly basis. As the input of PCLAKE was based on recent data as well (HydroSHEDS data 2019-2024), the time window of observed and predicted values align. We evaluated model performance, by calculating by normalizing the Root Means Square Error (NRMSE) by the mean of the observation. NRMSE values closer to zero represent better fitting models.

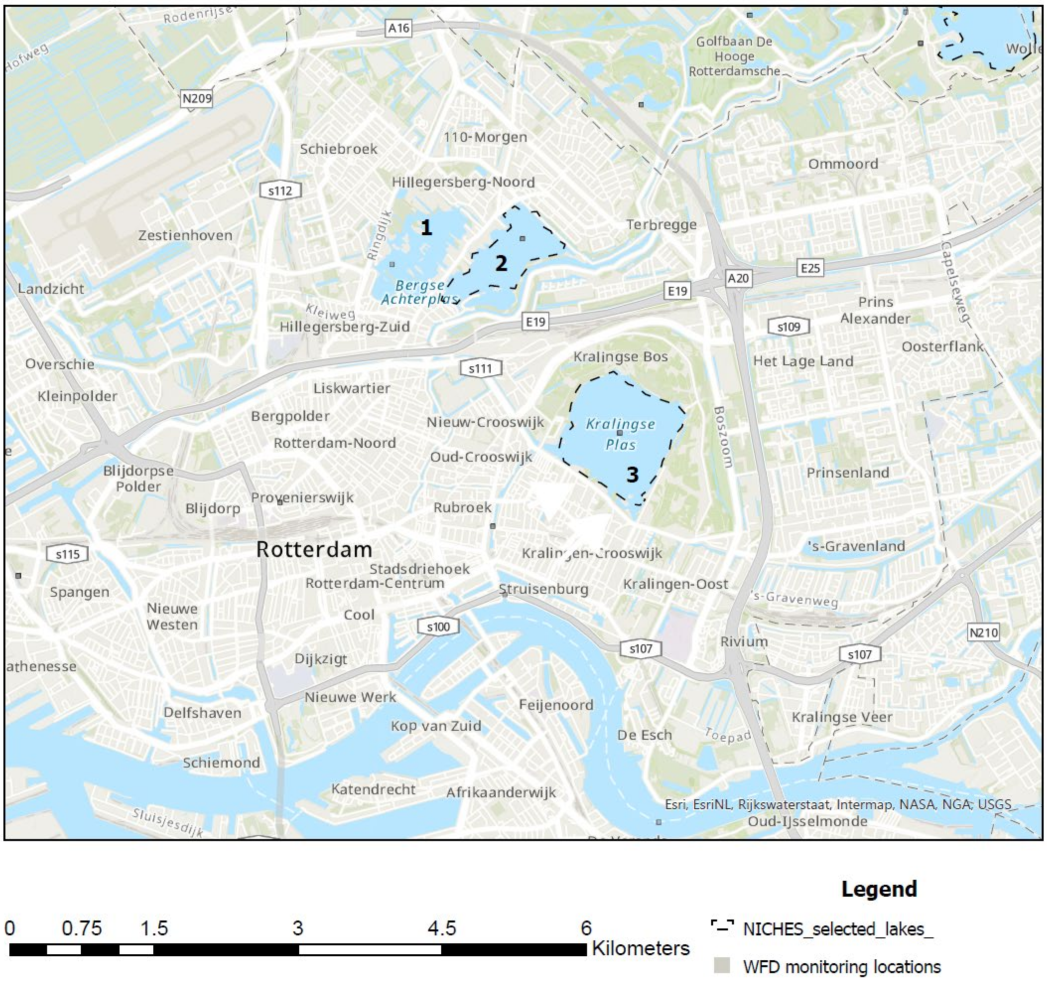

Figure 8: Detailed view on lakes not selected for modelling with the spatially explicit modelling framework (1), with LakeATLAS polygons only covering part of the water body (2) and LakeATLAS polygon aligned with waterbody (3).

For only 12 of the 42 modeled lakes, WFD data was available for transparency, and concentrations of dissolved oxygen, total nitrogen and total phosphorus. Below we show the fit of the observed vs. predicted starting the model from a turbid situation. Starting the model from a clear situation showed similar fits. In general, our spatial explicit modeling framework approximated the concentrations of total phosphorus well (NRMSE=0.01), total nitrogen (NRMSE=0.10), transparency (NRMSE=0.09) but showed a poor fit with observed Chl-A (NRMSE=1.84), and dissolved oxygen concentrations (NRMSE=0.46; Fig 9). Comparing the observed depth values with the values recorded in the LakeATLAS shows the potential culprit for the rather poor fits for concentrations of dissolved oxygen and Chl-A, with the LakeATLAS depth showing a poor fit (NRMSE=0.7) with the observed depth. As lake depth is crucial for calculating the water balance as well as the nutrient loadings, having poor estimate of depth will have a strong effect on model output.

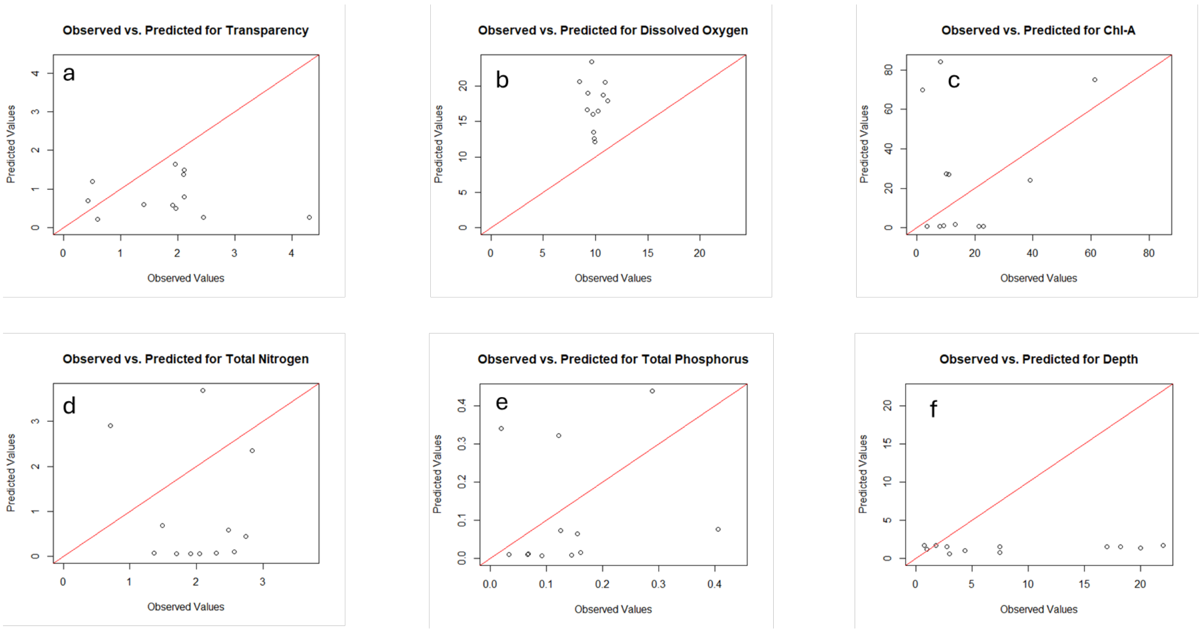

Figure 9: Observed vs predicted values for summer averages of transparency (a), dissolved oxygen concentrations (b), Chl-A concentration (c). total nitrogen concentration (d), total phosphorus concentration (e) and depth (f). Predicted depth values are the depth values as modelled in the LakeATLAS.

Using the observed depth of lakes as an input parameter to PCLake+ES improved the fit of most of the parameters, i.e., transparency (NRMSE=0.07), concentrations of total phosphorus (NRMSE=0.01) and total nitrogen (NRMSE=0.1), as well as Chl-A (NRMSE= 0.95), and dissolved oxygen (NRMSE=0.33; Fig. 10).

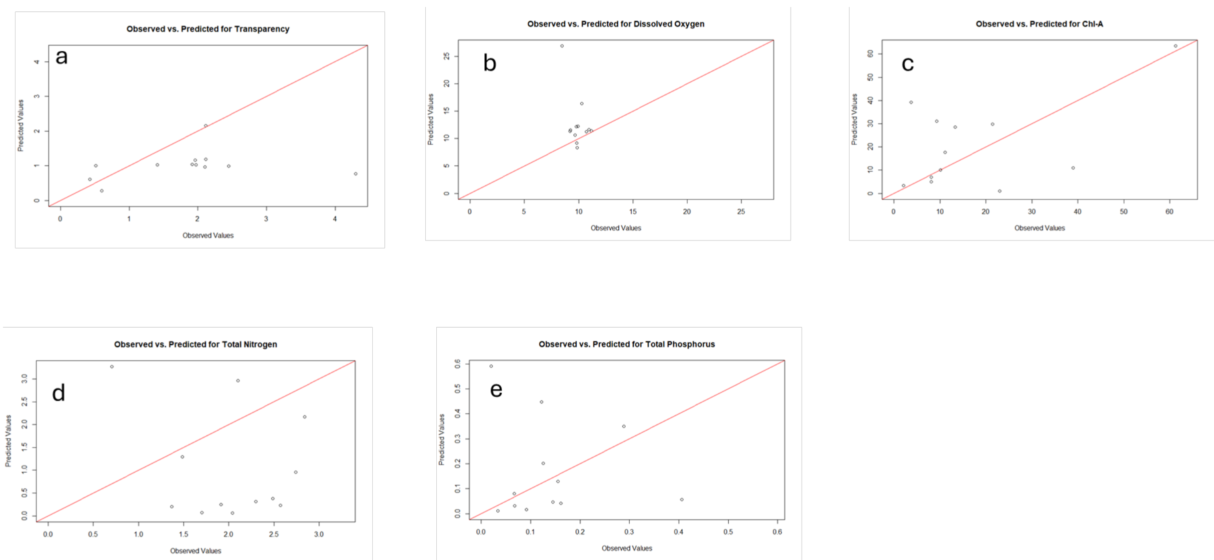

Figure 10: Observed vs predicted values for summer averages of transparency (a), dissolved oxygen concentrations (b), Chl-A concentration (c). total nitrogen concentration (d), total phosphorus concentration (e) with the model runs using observed depth as an input to PCLAKE+ES

Next steps in model validation should focus on improving the fits for Chl-A and dissolved oxygen, by including a larger validation set, i.e., WFD data (741 water bodies) of the

Netherlands, rather than the small dataset of Zuid-Holland, where observation outlier have a large impact on model performance evaluations.

6.2 Output

Due to the overall better model performance, we will show the modeling output for the PCLAKE+ ES runs which used the observed depth (based on WFD monitoring) as an input rather than the LakeATLAS modeled depth.

The Nature Based solutions were only able to marginally reduce the nitrogen and phosphorus loading to the receiving water body, see Table 2.

Table 2: Average nutrient loadings according to NBS scenario (value ± standard deviation)

	Phosphorus loading (g/m2/day)	**Nitrogen loading (g/m2/day)**
BAU	0.0092 (± 0.022)	0.059 (± 0. 130)
Grass swales	0.0086 (± 0.021)	0.054 (± 0.127)
Gravel wetlands	0.0088 (± 0.022)	0.055 (± 0.128)

Overall, our model results indicate that applying grass swales or gravel wetlands does not significantly (Fisher's exact test $P > 0.05$) improve the provisioning of regulating services for lakes in the Province of Zuid Holland as is evident for phosphorus sequestration (Fig. 11) or nitrogen sequestration (Fig. 12), both measures of the capacity of the receiving water body for nutrient burial in the sediment.

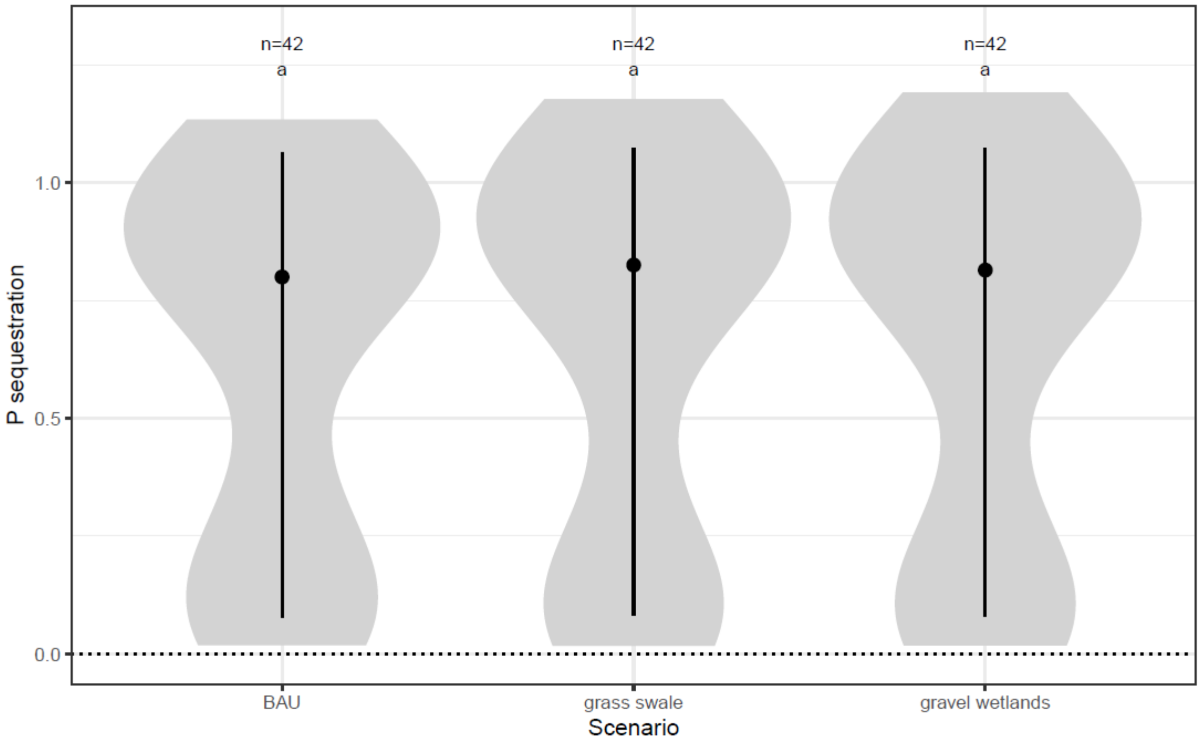

Figure 11: Effect of different NBS scenarios relative to BAU for phosphorus sequestration, a measure of phosphorus burial in the sediment of the receiving water body. The results for PCLAKE+ ES runs initiated at the turbid state are shown, with runs initiated at the clear state showing comparable results. The letters above the violin plots indicate the presence/absence of significant differences as tested by Fischer's exact test. Similar letters indicate that scenarios do not show significant differences.

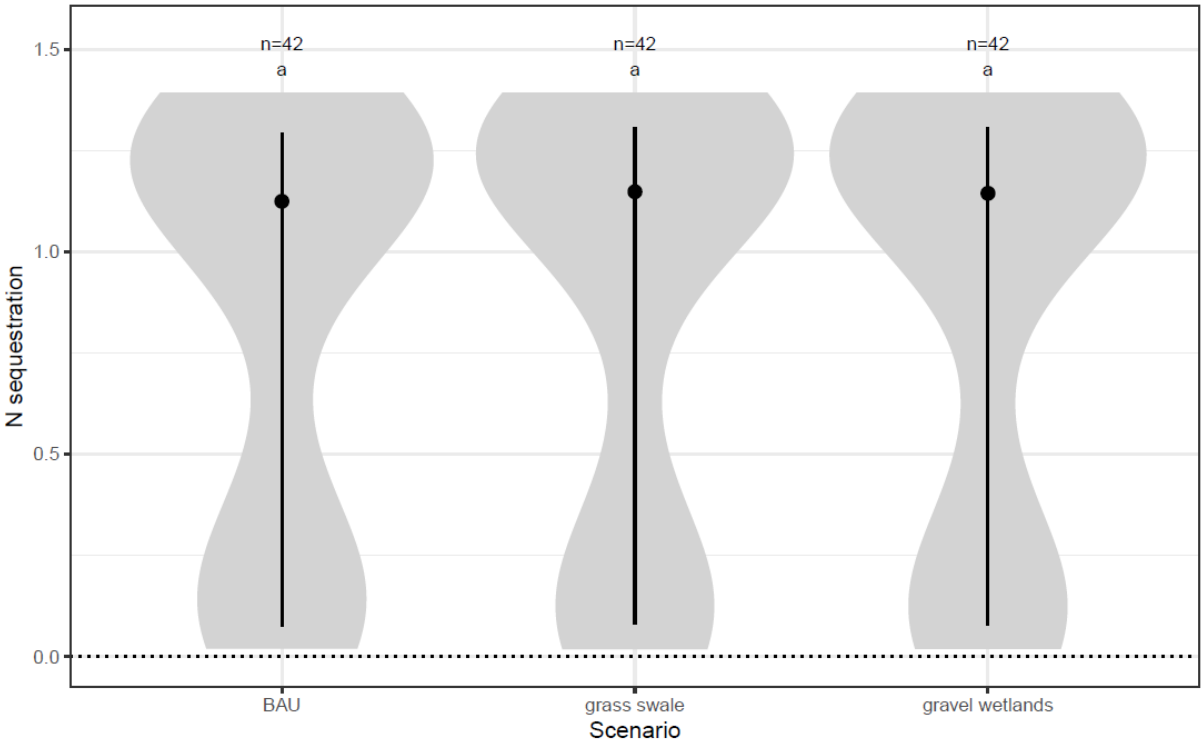

Figure 12: Effect of different NBS scenarios relative to BAU for nitrogen sequestration, a measure of nitrogen burial in the sediment of the receiving water body. The results for PCLAKE+ ES runs initiated at the turbid state are shown, with runs initiated at the clear state showing comparable results. The letters above the violin plots indicate the presence/absence of significant differences as tested by Fischer's exact test. Similar letters indicate that scenarios do not show significant differences.

Also, cultural services such as the potential for safe and nuisance free swimming (Fig. 13) bird watching (Fig. 14) or recreational fishing (Fig. 15) are not significantly affected by the implementation of grass swales and gravel wetlands (Fisher's exact test P >0.05).

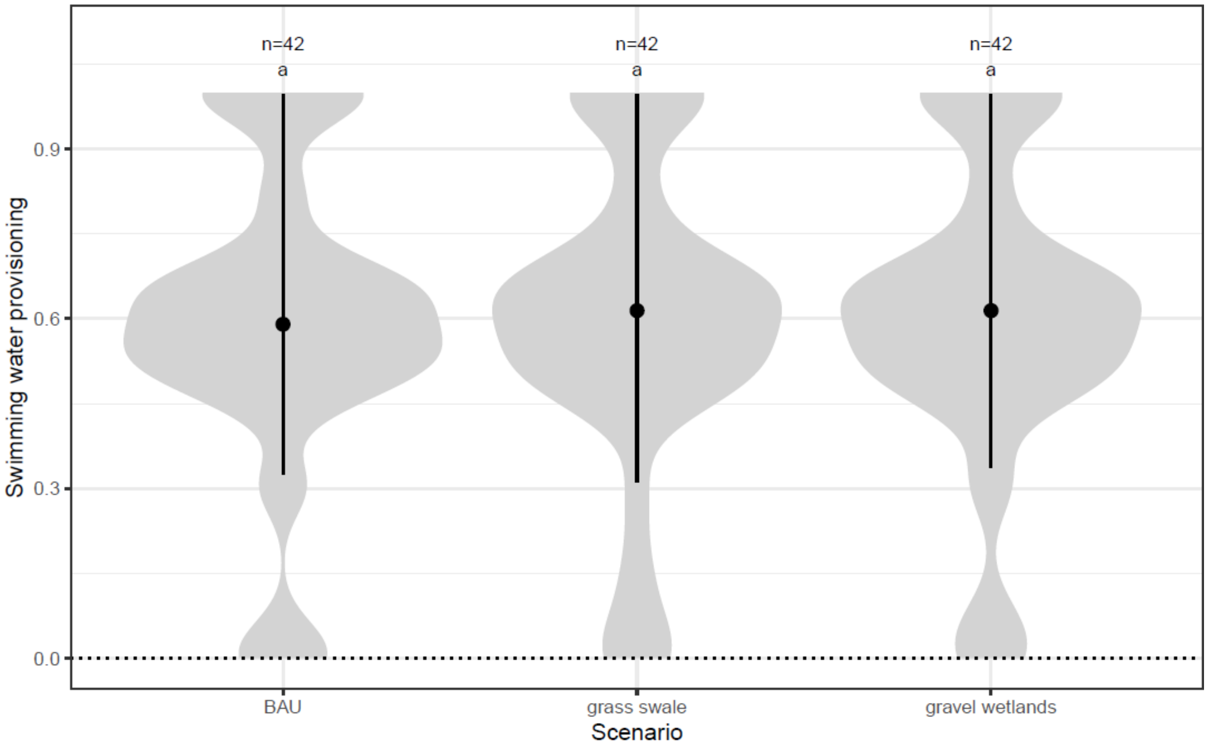

Figure 13: Effect of different NBS scenarios relative to BAU for swimming water provisioning of the receiving water body. The results for PCLAKE+ ES runs initiated at the turbid state are shown, with runs initiated at the clear state showing comparable results. The letters above the violin plots indicate the presence/absence of significant differences as tested by Fischer's exact test. Similar letters indicate that scenarios do not show significant differences.

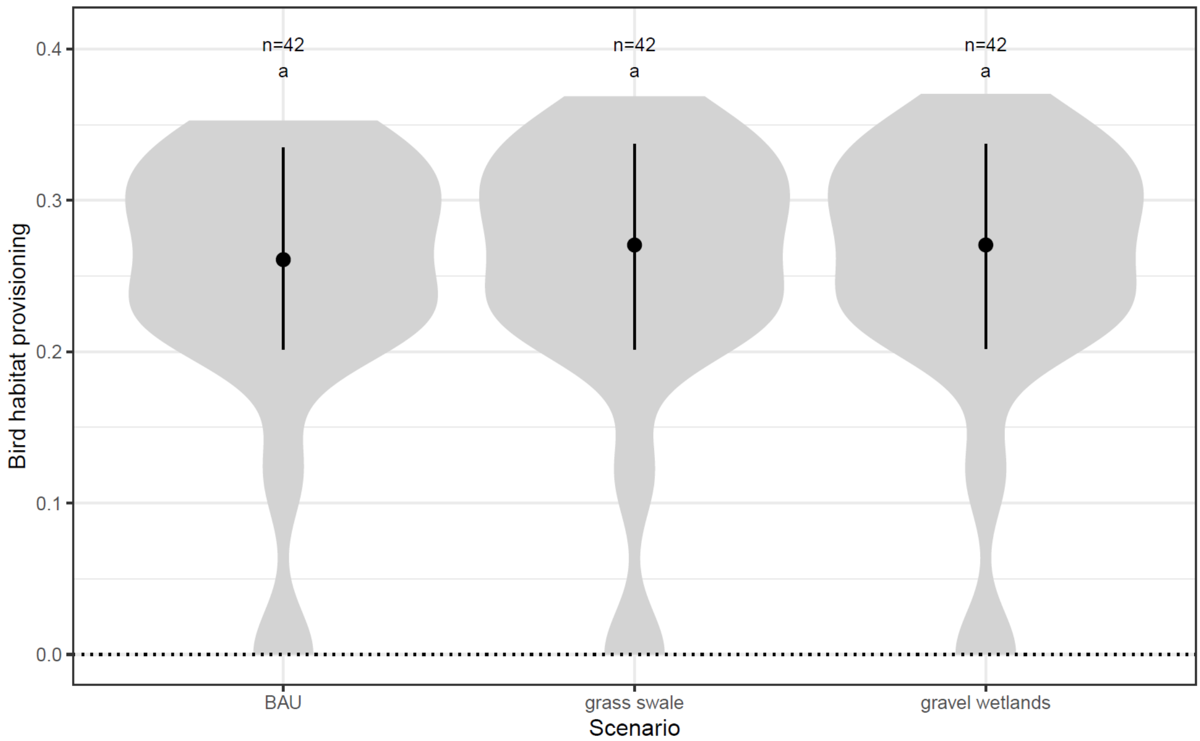

Figure 14: Effect of different NBS scenarios relative to BAU for bird habitat provisioning of the receiving water body. The results for PCLAKE+ ES runs initiated at the turbid state are shown, with runs initiated at the clear state showing comparable results. The letters above the violin plots indicate the presence/absence of significant differences as tested by Fischer's exact test. Similar letters indicate that scenarios do not show significant differences.

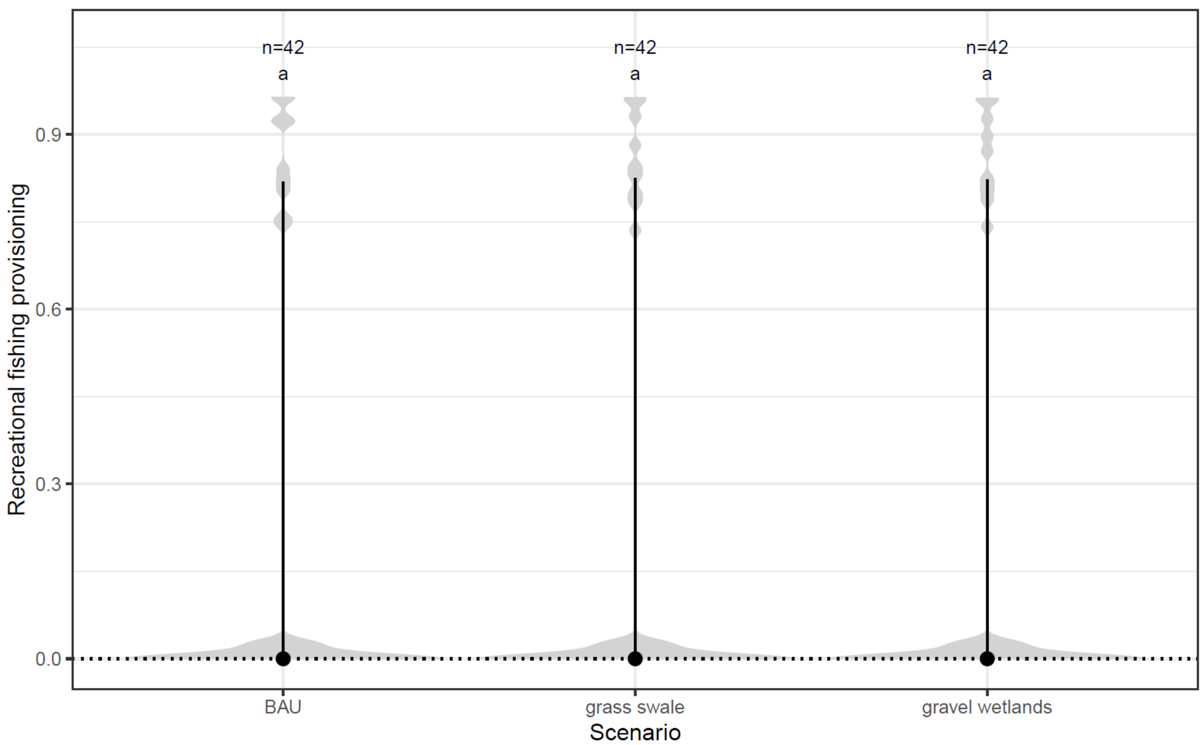

Figure 15: Effect of different NBS scenarios relative to BAU for recreational fishing provisioning of the receiving water body. The results for PCLAKE+ ES runs initiated at the turbid state are shown, with runs initiated at the clear state showing comparable results. The letters above the violin plots indicate the presence/absence of significant differences as tested by Fischer's exact test. Similar letters indicate that scenarios do not show significant differences.

Our results suggest that -with almost 100 % of the water bodies of Zuid Holland not reaching the [environmental targets for WFD,](#) these waters seem to be firmly locked in a turbid state. In these waters, diffuse pollution from agriculture and industry rather than sewages is most of the time the prime driver of water quality deterioration as is evident from the WFD reporting (Fig. 16).

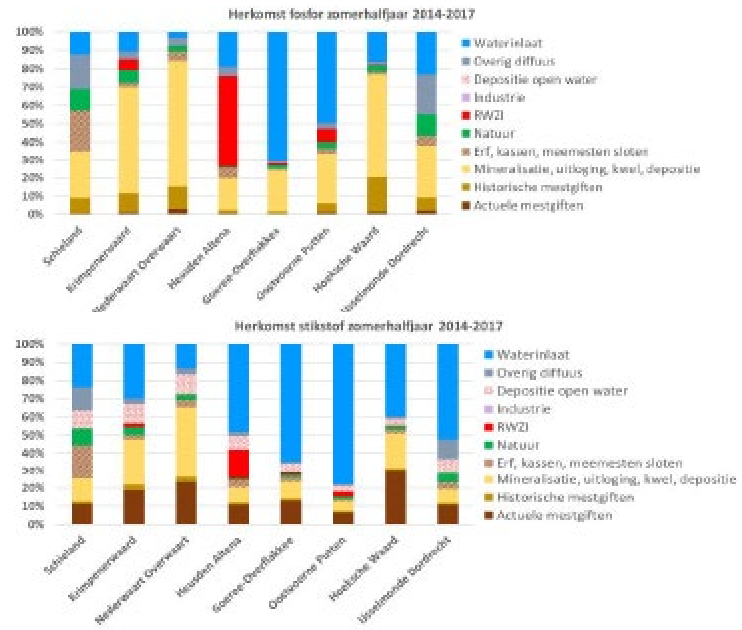

Figure 16: Nutrient pollution sources of different ZH waterbodies. In red the pollution originating from sewage systems. Source: https://www.zuid-holland.nl/actueel/nieuws/april-2025/tussenbalans-krw-provincie-zuid-holland-zet-extra/

Upscaling the spatially explicit modelling framework to areas in the Netherlands and beyond where sewage overflows play a larger role should more clearly underline what the potential of these NBS are for stormwater pollution reduction.

6.3 Future steps

Our modeling exercises show that there are several areas for improvement for our spatially explicit modeling framework:

1. Using observed depth rather than modelled depth as an input for PCLAKE+ES, drawing from the WISER database
2. Improving the delineation of water surface and watersheds to better align with water surfaces
3. Improved model validation using the WFD dataset for the entire Netherlands
4. Expanding the number of NBS modeling scenario's
5. Upscale to the European Scale for urban lakes through application of a filter representing the degree of urbanization.

7 References

Andersen, T. K., Nielsen, A., Jeppesen, E., Hu, F., Bolding, K., Liu, Z., Søndergaard, M., Johansson, L. S., & Trolle, D. (2020). Predicting ecosystem state changes in shallow lakes using an aquatic ecosystem model: Lake Hinge, Denmark, an example. *Ecological Applications*, *30*(7), e02160. https://doi.org/10.1002/eap.2160

Ballabio, C., Panagos, P., & Monatanarella, L. (2016). Mapping topsoil physical properties at European scale using the LUCAS database. *Geoderma*, *261*, 110–123. https://doi.org/10.1016/j.geoderma.2015.07.006

Burkhard, B., Kroll, F., Nedkov, S., & Müller, F. (2012). Mapping ecosystem service supply, demand and budgets. *Ecological Indicators*, *21*, 17–29. https://doi.org/10.1016/j.ecolind.2011.06.019

Chang, H., Pallathadka, A., Sauer, J., Grimm, N. B., Zimmerman, R., Cheng, C., Iwaniec, D. M., Kim, Y., Lloyd, R., McPhearson, T., Rosenzweig, B., Troxler, T., Welty, C., Brenner, R., & Herreros-Cantis, P. (2021). Assessment of urban flood vulnerability using the social-ecological-technological systems framework in six US cities. *Sustainable Cities and Society*, *68*, 102786. https://doi.org/10.1016/j.scs.2021.102786

Chelli, A., Brander, L., & Geneletti, D. (2025). Cost-Benefit analysis of urban nature-based solutions: A systematic review of approaches and scales with a focus on benefit valuation. *Ecosystem Services*, *71*, 101684. https://doi.org/10.1016/j.ecoser.2024.101684

Clopin, F., Micella, I., Mesman, J. P., Paule-Mercado, M. C., Amadori, M., Lin, S., de Senerpont Domis, L. N., & de Klein, J. J. M. (2025). Integrated models of nutrient dynamics in lake and reservoir watersheds: A systematic review and integrated modelling decision pathway. *Environmental Modelling & Software*, *185*, 106321. https://doi.org/10.1016/j.envsoft.2025.106321

Ekka, S. A., Rujner, H., Leonhardt, G., Blecken, G.-T., Viklander, M., & Hunt, W. F. (2021). Next generation swale design for stormwater runoff treatment: A comprehensive approach. *Journal of Environmental Management*, *279*, 111756. https://doi.org/10.1016/j.jenvman.2020.111756

Gómez-Baggethun, E., & Ruiz-Pérez, M. (2011). Economic valuation and the commodification of ecosystem services. *Progress in Physical Geography: Earth and Environment*, *35*(5), 613–628. https://doi.org/10.1177/0309133311421708

Grizzetti, B., Lanzanova, D., Liquete, C., Reynaud, A., & Cardoso, A. C. (2016). Assessing water ecosystem services for water resource management. *Environmental Science & Policy*, *61*, 194–203. https://doi.org/10.1016/j.envsci.2016.04.008

Grizzetti, B., Liquete, C., Pistocchi, A., Vigiak, O., Zulian, G., Bouraoui, F., De Roo, A., & Cardoso, A. C. (2019). Relationship between ecological condition and ecosystem services in European rivers, lakes and coastal waters. *Science of The Total Environment*, *671*, 452–465. https://doi.org/10.1016/j.scitotenv.2019.03.155

Guerry, A. D., Polasky, S., Lubchenco, J., Chaplin-Kramer, R., Daily, G. C., Griffin, R., Ruckelshaus, M., Bateman, I. J., Duraiappah, A., Elmqvist, T., Feldman, M. W., Folke, C., Hoekstra, J., Kareiva, P. M., Keeler, B. L., Li, S., McKenzie, E., Ouyang, Z., Reyers, B., … Vira, B. (2015). Natural capital and ecosystem services informing decisions: From promise to practice. *Proceedings of the National Academy of Sciences*, *112*(24), 7348–7355. https://doi.org/10.1073/pnas.1503751112

Hipsey, M. R., Bruce, L. C., Boon, C., Busch, B., Carey, C. C., Hamilton, D. P., Hanson, P. C., Read, J. S., de Sousa, E., Weber, M., & Winslow, L. A. (2019). A General Lake Model

(GLM 3.0) for linking with high-frequency sensor data from the Global Lake Ecological Observatory Network (GLEON). *Geoscientific Model Development*, *12*(1), 473–523. https://doi.org/10.5194/gmd-12-473-2019

Ibelings, B. W., Portielje, R., Lammens, E. H. R. R., Noordhuis, R., van den Berg, M. S., Joosse, W., & Meijer, M. L. (2007). Resilience of Alternative Stable States during the Recovery of Shallow Lakes from Eutrophication: Lake Veluwe as a Case Study. *Ecosystems*, *10*(1), 4–16. https://doi.org/10.1007/s10021-006-9009-4

Janse, J. H. (2005). *Model studies on the eutrophication of shallow lakes and ditches*. Wageningen University.

Janse, J. H., De Senerpont Domis, L. N., Scheffer, M., Lijklema, L., Van Liere, L., Klinge, M., & Mooij, W. M. (2008). Critical phosphorus loading of different types of shallow lakes and the consequences for management estimated with the ecosystem model PCLake. *Limnologica*, *38*(3), 203–219. https://doi.org/10.1016/j.limno.2008.06.001

Janse, J. H., Scheffer, M., Lijklema, L., Van Liere, L., Sloot, J. S., & Mooij, W. M. (2010). Estimating the critical phosphorus loading of shallow lakes with the ecosystem model PCLake: Sensitivity, calibration and uncertainty. *Ecological Modelling*, *221*(4), 654–665. https://doi.org/10.1016/j.ecolmodel.2009.07.023

Janssen, A. B. G., Arhonditsis, G. B., Beusen, A., Bolding, K., Bruce, L., Bruggeman, J., Couture, R.-M., Downing, A. S., Alex Elliott, J., Frassl, M. A., Gal, G., Gerla, D. J., Hipsey, M. R., Hu, F., Ives, S. C., Janse, J. H., Jeppesen, E., Jöhnk, K. D., Kneis, D., … Mooij, W. M. (2015). Exploring, exploiting and evolving diversity of aquatic ecosystem models: A community perspective. *Aquatic Ecology*, *49*(4), 513–548. https://doi.org/10.1007/s10452-015-9544-1

Janssen, A. B. G., Hilt, S., Kosten, S., de Klein, J. J. M., Paerl, H. W., & Van de Waal, D. B. (2021). Shifting states, shifting services: Linking regime shifts to changes in ecosystem services of shallow lakes. *Freshwater Biology*, *66*(1), 1–12. https://doi.org/10.1111/fwb.13582

Janssen, A. B. G., Teurlincx, S., Beusen, A. H. W., Huijbregts, M. A. J., Rost, J., Schipper, A. M., Seelen, L. M. S., Mooij, W. M., & Janse, J. H. (2019). PCLake+: A process-based ecological model to assess the trophic state of stratified and non-stratified freshwater lakes worldwide. *Ecological Modelling*, *396*, 23–32. https://doi.org/10.1016/j.ecolmodel.2019.01.006

Kirillin, G., Hochschild, J., Mironov, D., Terzhevik, A., Golosov, S., & Nützmann, G. (2011). FLake-Global: Online lake model with worldwide coverage. *Environmental Modelling & Software*, *26*(5), 683–684. https://doi.org/10.1016/j.envsoft.2010.12.004

Lehner, B., Messager, M. L., Korver, M. C., & Linke, S. (2022). Global hydro-environmental lake characteristics at high spatial resolution. *Scientific Data*, *9*(1), 351. https://doi.org/10.1038/s41597-022-01425-z

Linke, S., Lehner, B., Ouellet Dallaire, C., Ariwi, J., Grill, G., Anand, M., Beames, P., Burchard-Levine, V., Maxwell, S., Moidu, H., Tan, F., & Thieme, M. (2019). Global hydro-environmental sub-basin and river reach characteristics at high spatial resolution. *Scientific Data*, *6*(1), 283. https://doi.org/10.1038/s41597-019-0300-6

Lürling, M., & Mucci, M. (2020). Mitigating eutrophication nuisance: In-lake measures are becoming inevitable in eutrophic waters in the Netherlands. *Hydrobiologia*, *847*(21), 4447–4467. https://doi.org/10.1007/s10750-020-04297-9

Mooij, W. M., Janse, J. H., De Senerpont Domis, L. N., Hülsmann, S., & Ibelings, B. W. (2007). Predicting the effect of climate change on temperate shallow lakes with the ecosystem model PCLake. In R. D. Gulati, E. Lammens, N. De Pauw, & E. Van Donk (Eds.), *Shallow Lakes in a Changing World* (pp. 443–454). Springer Netherlands. https://doi.org/10.1007/978-1-4020-6399-2_40

Nelson, E., Mendoza, G., Regetz, J., Polasky, S., Tallis, H., Cameron, Dr., Chan, K. M., Daily, G. C., Goldstein, J., Kareiva, P. M., Lonsdorf, E., Naidoo, R., Ricketts, T. H., & Shaw, Mr. (2009). Modeling multiple ecosystem services, biodiversity conservation, commodity production, and tradeoffs at landscape scales. *Frontiers in Ecology and the Environment*, *7*(1), 4–11. https://doi.org/10.1890/080023

Polasky, S., Nelson, E., Pennington, D., & Johnson, K. A. (2011). The Impact of Land-Use Change on Ecosystem Services, Biodiversity and Returns to Landowners: A Case Study in the State of Minnesota. *Environmental and Resource Economics*, *48*(2), 219–242. https://doi.org/10.1007/s10640-010-9407-0

Sarabi, S., Han, Q., Romme, A. G. L., de Vries, B., Valkenburg, R., & den Ouden, E. (2020). Uptake and implementation of Nature-Based Solutions: An analysis of barriers using Interpretive Structural Modeling. *Journal of Environmental Management*, *270*, 110749. https://doi.org/10.1016/j.jenvman.2020.110749

Seelen, L. M. S., Teurlincx, S., Armstrong, M. R., Lürling, M., van Donk, E., & de Senerpont Domis, L. N. (2022). Serving many masters at once: A framework for assessing ecosystem services delivered by quarry lakes. *Inland Waters*, *12*(1), 121–137. https://doi.org/10.1080/20442041.2021.1944765

van der Werf, J. A., Kapelan, Z., & Langeveld, J. (2023). Real-time control of combined sewer systems: Risks associated with uncertainties. *Journal of Hydrology*, *617*, 128900. https://doi.org/10.1016/j.jhydrol.2022.128900

Wang, M., Strokal, M., Burek, P., Kroeze, C., Ma, L., & Janssen, A. B. G. (2019). Excess nutrient loads to Lake Taihu: Opportunities for nutrient reduction. *Science of The Total Environment*, *664*, 865–873. https://doi.org/10.1016/j.scitotenv.2019.02.051

Wortley, L., Hero, J.-M., & Howes, M. (2013). Evaluating Ecological Restoration Success: A Review of the Literature. *Restoration Ecology*, *21*(5), 537–543. https://doi.org/10.1111/rec.12028

Yang, J., Strokal, M., Kroeze, C., Ma, L., Bai, Z., Teurlincx, S., & Janssen, A. B. G. (2022). What is the pollution limit? Comparing nutrient loads with thresholds to improve water quality in Lake Baiyangdian. *Science of The Total Environment*, *807*, 150710. https://doi.org/10.1016/j.scitotenv.2021.150710

Zhan, Q., Domis, L. N. de S., Lürling, M., Marcé, R., Heuts, T. S., & Teurlincx, S. (2023). Process-based modeling for ecosystem service provisioning: Non-linear responses to restoration efforts in a quarry lake under climate change. *Journal of Environmental Management*, *348*, 119163. https://doi.org/10.1016/j.jenvman.2023.119163

http://niches-project.eu/

Project partners

NICHES is made possible with the support of:

8 Annex

8.1 BATT land use conversion table

Table 3: BATT land use conversion table and the associated nutrient loadings expressed in lb/acre/year. I stands for impervious land use, and P stands for pervious land use.

BATT_Land_cover	Type_of_LC	Corine_Land_Cover	Phosphorous_lb_ac_yr	Nitrogen_lb_ac_yr	TSS_lb_ac_yr
Highways	I	Road and rail networks and associated land	1.34	10.17	1480.13
Highways	I	Airports	1.34	10.17	1480.13
Agriculture	P	Arable land	0.45	2.59	29.44
Agriculture	P	Annual crops associated with permanent crops	0.45	2.59	29.44
Agriculture	P	Complex cultivation patterns	0.45	2.59	29.44
Agriculture	P	Land principally occupied by agriculture, with significant areas of natural vegetation	0.45	2.59	29.44
Agriculture	P	Non-irrigated arable land	0.45	2.59	29.44
Agriculture	P	Permanently irrigated arable land	0.45	2.59	29.44
Agriculture	P	Rice fields	0.45	2.59	29.44
Agriculture	P	Agro-forestry areas	0.45	2.59	29.44
Agriculture	P	Vineyards	0.45	2.59	29.44
Agriculture	P	Fruit tree and berry plantations	0.45	2.59	29.44
Agriculture	P	Olive groves	0.45	2.59	29.44
Agriculture	P	Pastures	0.45	2.59	29.44
Agriculture	P	Heterogeneous agricultural areas	0.45	2.59	29.44
Agriculture	I	NA	1.52	11.33	649.51
Commercial	I	Port area’s	1.78	15.08	377.39
Commercial	P	NA	NA	NA	NA
High Density Residential	I	Continuous urban fabric	2.32	14.1	438.95
High Density Residential	P	NA	NA	NA	NA
Middle Density Residential	I	Discontinuous urban fabric	1.96	14.1	438.95
Middle Density Residential	P	NA	NA	NA	NA
Low Density Residential (single family)	I	NA	1.52	14.1	438.95
Low Density Residential (single family)	P	NA	NA	NA	NA
Open land	I	Green urban areas	1.52	11.33	649.51
Open land	I	Sport and leisure facilities	1.52	11.33	649.51
Open land	I	Bare rock	1.52	11.33	649.51
Open land	I	NA	1.52	11.33	649.51
Open land	P	Natural grasslands	NA	NA	NA

Open land	P	Beaches, dunes, sands	NA	NA	NA
Open land	P	Sparsely vegetated areas	NA	NA	NA
Open land	P	Burnt areas	NA	NA	NA
Industrial	I	Airports	1.78	15.08	377.39
Industrial	P	Mineral extraction sites	1.78	15.08	377.39
Industrial	P	Dump sites	1.78	15.08	377.39
Industrial	P	Construction sites	1.78	15.08	377.39
Industrial	P	Industrial or commercial units	1.78	15.08	377.39
Forest	P	Broad-leaved forest	0.12	0.54	29.44
Forest	P	Coniferous forest	0.12	0.54	29.44
Forest	P	Mixed forest	0.12	0.54	29.44
Forest	P	Moors and heathland	0.12	0.54	29.44
Forest	P	Transitional woodland-shrub	0.12	0.54	29.44
Forest	P	Sclerophyllous vegetation	0.12	0.54	29.44
Forest	P	Inland marshes	0.12	0.54	29.44
Forest	P	Peatbogs	0.12	0.54	29.44
Forest	I	NA	1.52	11.33	649.51
Water	P	Glaciers and perpetual snow	0.03	0.27	7.14
Water	P	Salt marshes	0.03	0.27	7.14
Water	P	Salines	0.03	0.27	7.14
Water	P	Intertidal flats	0.03	0.27	7.14
Water	P	Water courses	0.03	0.27	7.14
Water	P	Water bodies	0.03	0.27	7.14
Water	P	Coastal lagoons	0.03	0.27	7.14
Water	P	Estuaries	0.03	0.27	7.14
Water	P	Sea and ocean	0.03	0.27	7.14

8.2 BATT Hydrological Soil group Conversion table

Table 4: BATT Hydrological Soil Group Conversion table using the soil codes of the ESDAC topsoil properties database

Soil_Type	Description	Soil_Code
A	Sand	10
A	Loamy_Sand	11
A	Sandy_Loam	12
B	Loam	9
C	Sandy_Clay-Loam	5
D	Clay_Loam	6
D	Silty_Clay-Loam	3
D	Sandy_Clay	4
D	Silty_Clay	2
D	Clay	1
B	Silt	7
B	Silt-Loam	8

8.3 BATT Runoff curve number conversion table

Table 5: BATT Runnoff curve number conversion table, where Land Use/Land Cover combinations are mapped on Corine Land Cover categories to retrieve hydrological curve numbers.

Land_use	Land_cover	Corine_Landcover	A	B	C	D
Cultivated	Straight row	Annual crops associated with permanent crops	76	86	90	93
Cultivated	Straight row	Complex cultivation patterns	76	86	90	93
Cultivated	Straight row	Land principally occupied by agriculture, with significant areas of natural vegetation	76	86	90	93
Cultivated	Straight row	Non-irrigated arable land	76	86	90	93
Cultivated	Straight row	Permanently irrigated arable land	76	86	90	93
Cultivated	Contoured_poor		70	79	84	88
Cultivated	Contoured_good		65	75	82	86
Cultivated	Con_terr_poor		66	74	80	82
Cultivated	Con_terr_good		62	71	77	81
Cultivated	Bunded_poor		67	75	81	83
Cultivated	Bunded_good		59	69	76	79
Cultivated	Paddy	Rice fields	95	95	95	95
Orchards	Understory	Agro-forestry areas	39	53	67	71
Orchards	No_understory	Vineyards	41	55	69	73
Orchards	No_understory	Fruit tree and berry plantations	41	55	69	73
Orchards	No_understory	Olive groves	41	55	69	73
Forest	Dense	Broad-leaved forest	26	40	58	61
Forest	Dense	Coniferous forest	26	40	58	61
Forest	Dense	Mixed forest	26	40	58	61
Forest	Open	Transitional woodland/shrub	28	44	60	64
Forest	Open	Inland marshes	28	44	60	64
Forest	Open	Peatbogs	28	44	60	64
Forest	Scrub	Moors and heathland	33	47	64	67
Pasture	Poor		68	79	86	89
Pasture	Fair	Pastures	49	69	79	84
Pasture	Good	Natural grasslands	39	61	74	80
Wasteland		Sclerophyllous vegetation	71	80	85	88
Wasteland		Sparsely vegetated areas	71	80	85	88
Wasteland		Dump sites	71	80	85	88
Wasteland		Beaches, dunes, sands	71	80	85	88
Wasteland		Burnt areas	71	80	85	88
Dirt_road			73	83	88	90
Hard_surface		Bare rock	77	86	91	93
Open_space	Good	Green urban areas	39	61	74	80
Open_space	Fair	Sport and leisure facilities	49	69	79	84
Commercial		Continuous urban fabric	89	92	94	95

Commercial		Port areas	89	92	94	95
Industrial		Industrial or commercial units	81	88	91	93
Residential		Discontinuous urban fabric	77	85	90	92
Paved		Road and rail networks and associated land	98	98	98	98
Paved		Airports	98	98	98	98
Gravel_street		Mineral extraction sites	76	85	89	91
Gravel_street		Construction sites	76	85	89	91
Dirt_street			72	82	87	89

8.4 Spatially explicit modeling framework script for all lakes in EU

```
##===Script for NICHES analyses on all lakes in EU===

#needed datasets

#-CORINE LAND COVER

#--Land cover classes

#-LAKEATLAS

#--Temperature monthly

#--Residence time

#--Average lake depth

#--Latitude

#--Lake area

#--Watershed area

#-BASINATLAS

#--Basin shape delineation for land cover extraction

#-SOIL map

#--Soil characteristics to get hydrological runoff categories of BATT

#


rm(list=ls())

library(lubridate)
library(dplyr)
library(data.table)
library(stringr)
library(foreach)
library(doSNOW)
library(powerjoin)
library(ggplot2)
library(sf)
library(tidyverse, quietly=T)
library(osmdata)
library(rcompanion)
library(multcompView)
library(osmdata)
library(units)

options(scipen = 999)

nearZero <- 1E-28
days_of_summer <- expand.grid(seq(91, 274, 1), seq(0, 40))
```

```
summer_vec <- days_of_summer$Var1 + (365 * days_of_summer$Var2)


sNOW = str_replace_all(Sys.time(), "[[:punct:]]", "")

sFOLDER = file.path("C:","Users","SvenT","OneDrive -
NIOO","Documents","NICHES","ArcGIS","Comple_shp_files")
sFOLDER2 = file.path("C:","Users","SvenT","OneDrive -
NIOO","Documents","NICHES","ArcGIS")
sFOLDER_Rotterdam =
file.path("C:","Users","FrancisD","Documents","NICHES","GIS")
sFOLDER_Cschijf =
file.path("C:","Users","FrancisD","Documents","NICHES","GIS","final_shp
2")
sFOLDER_Corine =
file.path("C:","Users","FrancisD","Documents","NICHES","u2018_clc2018_v
2020_20u1_fgdb","u2018_clc2018_v2020_20u1_fgdb","DATA","U2018_CLC2018_V
2020_20u1.gdb")


shapeLAKES <- read_sf(dsn = sFOLDER_Rotterdam, layer =
"Lakes_Rotterdam")

shapeBASINS <- read_sf(dsn = sFOLDER_Cschijf, layer =
"BasinATLAS_Europe")

shapeCORINE <- read_sf(dsn = sFOLDER_Rotterdam, layer =
"Corine_ZuidHolland")

shapeURBAN <- read_sf(dsn = sFOLDER_Cschijf, layer =
"Urban_Europe_final")

shapeSOILS <- read_sf(dsn = sFOLDER_Cschijf, layer = "Soil_map_Europe")
colnames(shapeSOILS)[colnames(shapeSOILS) == "gridcode"] <- "Soil_Code"

shapeLAKES = st_make_valid(shapeLAKES)

shapeLAKES = st_transform(shapeLAKES, crs=st_crs(shapeCORINE))

shapeBASINS=st_make_valid(shapeBASINS)

shapeBASINS = st_transform(shapeBASINS, crs=st_crs(shapeCORINE))

shapeSOILS=st_make_valid(shapeSOILS)

shapeSOILS = st_transform(shapeSOILS, crs=st_crs(shapeCORINE))

shapeURBAN = st_make_valid(shapeURBAN)

shapeURBAN = st_transform(shapeURBAN, crs=st_crs(shapeCORINE))

shapeCORINE <- shapeCORINE[st_geometry_type(shapeCORINE) !=
"MULTISURFACE", ]
shapeCORINE=st_make_valid(shapeCORINE)


#acceptable difference between calculated watershed area and watershed
area of HYDROLAKES (in fraction)
```

```
fTHRESH_WSHD = 0.001
fBUFF_CHANGE_FRAC = 0.05 #change in buffer width (fraction of previous
buffer)

#acceptable difference between soil map and watershed (0.5 means at
least half the defined watershed has to have valid land use mapping)
fTHRESH_SOIL = 0.5

sf::sf_use_s2(TRUE)

#Corine landcover legend
CLC_legend <-
fread(file.path("C:","Users","FrancisD","Documents","NICHES","u2018_clc
2018_v2020_20u1_fgdb","u2018_clc2018_v2020_20u1_fgdb","Legend","CLC_leg
end.csv"))
colnames(CLC_legend)[colnames(CLC_legend) == "CLC_CODE"] <- "Code_18"
colnames(CLC_legend)[colnames(CLC_legend) == "LABEL3"] <-
"Corine_Landcover"

#BATT to Corine landcover conversion table
Conversion <-
fread(file.path("C:","Users","FrancisD","Documents","NICHES","BATT",
"BATT_Conversion_Table.csv"))
colnames(Conversion)[colnames(Conversion) == "Corine_Land_Cover"] <-
"Corine_Landcover"
colnames(Conversion)[colnames(Conversion) == "BATT_Land_cover"] <-
"BATT_Landcover"

#Curve number table
Curve_Number <-
fread(file.path("C:","Users","FrancisD","Documents","NICHES","BATT",
"Curve_Number.csv"))

#Runoff coefficient table
Runoff_Coefficient <-
fread(file.path("C:","Users","FrancisD","Documents","NICHES","BATT",
"Runoff_Coefficient.csv"))

#Soil types table
Soil_types <-
fread(file.path("C:","Users","FrancisD","Documents","NICHES","BATT",
"Soil_types.csv"))

#Nutrient loads per soil type table
Loading <-
fread(file.path("C:","Users","FrancisD","Documents","NICHES","BATT",
"Loading.csv"))

#Table of pathway types and their corresponding widths
Pathway_width <-
fread(file.path("C:","Users","FrancisD","Documents","NICHES","BATT",
"Pathways.csv"))
colnames(Pathway_width)[colnames(Pathway_width) == "Path"] <- "highway"


# Optional for spatial filter: extract lakes within a 5 km radius of
urban centers
```

```
# colnames(shapeURBAN)[colnames(shapeURBAN) == "gridcode"] <-
"Urban_class"
# UrbanClass30 <- shapeURBAN %>% filter(Urban_class == "30")

#Set url osm data
set_overpass_url("https://overpass-api.de/api/interpreter")

#PCLake+ initialization code----
dirHome <- "C:/Users/FrancisD/Documents/PCLAKE/PCModel-master/PCModel-
master/Licence_agreement/I_accept/"     # location of the PCModel1350
folder
dirShell <- file.path(dirHome, "PCModel1350", "PCModel", "3.00",
"Models", "PCLake+", "6.13.16", "PCShell")
dirCpp_root <- file.path(dirHome, "PCModel1350", "PCModel", "3.00",
"Frameworks", "Osiris", "3.01", "PCLake_plus")
nameWorkCase <- "PCLake_plus_NICHES_BATT"
fileDATM <- file.path(dirHome, "PCModel1350", "PCModel", "3.00",
"Models", "PCLake+", "6.13.16",
"PL613162PLUS_ESs_NICHES_BATTPCLake_20250320.xls")

## load all the functions
source(file.path(dirShell, "scripts", "R_system", "functions.R"))
      #load base functions by Luuk van Gerven (2012-2016)
source(file.path(dirShell, "scripts", "R_system",
"functions_PCLake.R"))

## 1. Making folder structure
PCModelWorkCaseSetup(dirSHELL = dirShell,
   dirCPP_ROOT = dirCpp_root,
   nameWORKCASE = nameWorkCase)

## 2. Load file
lDATM_SETTINGS <- PCModelReadDATMFile_PCLakePlus(fileXLS = fileDATM,
       locDATM = "excel",
       locFORCING = "excel",
       readAllForcings = F)


if (exists("PC_Lake")==TRUE){
 rm("PC_Lake")
}



#Stappenplan

#1: Select basins from HYDROATLAS which intersect with a given lake----

for (nLAKE in 1:nrow(shapeLAKES)){

 #select lake
 pLAKE = shapeLAKES[nLAKE,]
 pLAKE <- st_transform(pLAKE, st_crs(shapeLAKES))

 # Optional filter: select lakes located within 5,000 meters of urban
centers
 # This step identifies lakes in close proximity to urban areas based
on a buffer distance
```

```
# #Get Lakes in urban areas
# Urban_buffer <- st_buffer(UrbanClass30, dist = 5000)
#
# #Intersecting lakes
# Urban_buffer <- st_transform(UrbanClass30, st_crs(pLAKE))
# vUrban_Overlap = st_intersects(Urban_buffer, pLAKE,sparse=FALSE)
# pUrban_Class = Urban_buffer[vUrban_Overlap,]
#
#
# if (nrow(pUrban_Class) != 0){

#get basins that intersect with the lake polygon
vBASINS_OVERLAP = st_intersects(shapeBASINS, pLAKE,sparse=FALSE)

pBASINS = shapeBASINS[vBASINS_OVERLAP,]

#buffer the lake so that it roughly aproximates its watershed

# Calculate initial buffer width
#note the watershed area of hydroATLAS does not include the lake
surface

fBUFFER <- ((sqrt((pLAKE$Wshd_area+pLAKE$Lake_area) / pi)) -
(sqrt(pLAKE$Lake_area / pi))) *1000 #in m

#if(fBUFFER <0){fBUFFER =sqrt(pLAKE$Lake_area / pi)*1000}

# Create the buffered lake geometry
pLAKE_BUF <- st_buffer(pLAKE, dist = fBUFFER)

# Cut the buffered lake by the selected basin polygons
pLAKE_WSHD <- st_intersection(pLAKE_BUF,pBASINS)

fAREA_WATERSHED = drop_units(sum(st_area(pLAKE_WSHD))/1000000)

#fraction of calculated watershed area relative to the desired
(hydrolakes) watershed size
fAREAWSHD_DIV = 1.0-
(min(as.numeric(pLAKE$Wshd_area)+as.numeric(pLAKE$Lake_area),as.numeric
(fAREA_WATERSHED))/max(as.numeric(pLAKE$Wshd_area)+as.numeric(pLAKE$Lak
e_area),as.numeric(fAREA_WATERSHED)))

#basin max size
fAREA_BASINS = sum(as.numeric(st_area(pBASINS)))/1000000

#while loop to optimize watershed area
while(fAREAWSHD_DIV > fTHRESH_WSHD ){
#break if the watershed area needs to increase, but there is no more
area of basins left
if((round(as.numeric(fAREA_WATERSHED),2) ==
round(as.numeric(fAREA_BASINS),2)) & (as.numeric(pLAKE$Wshd_area) >
as.numeric(fAREA_WATERSHED))){
pLAKE$Oversized <- 1 # mark as oversized
break()
}
```

```
 if(as.numeric(pLAKE$Wshd_area)+as.numeric(pLAKE$Lake_area) <
as.numeric(fAREA_WATERSHED)){
 print(paste("decrease: ", fAREAWSHD_DIV))
 #decrease buffer size

 #buffer the lake so that it roughly aproximates its watershed

 # set buffer width based on previous buffer width
 fBUFFER <- fBUFFER*(1-fBUFF_CHANGE_FRAC)#in m

 # Create the buffered lake geometry
 pLAKE_BUF <- st_buffer(pLAKE, dist = fBUFFER)

 # Cut the buffered lake by the selected basin polygons
 pLAKE_WSHD <- st_intersection(pLAKE_BUF,pBASINS)

 fAREA_WATERSHED = sum(as.numeric(st_area(pLAKE_WSHD)))/1000000

 #fraction of calculated watershed area relative to the desired
(hydrolakes) watershed size
 fAREAWSHD_DIV = 1.0-
(min(as.numeric(pLAKE$Wshd_area)+as.numeric(pLAKE$Lake_area),as.numeric
(fAREA_WATERSHED))/max(as.numeric(pLAKE$Wshd_area)+as.numeric(pLAKE$Lak
e_area),as.numeric(fAREA_WATERSHED)))

 }else if(as.numeric(pLAKE$Wshd_area)+as.numeric(pLAKE$Lake_area) >
as.numeric(fAREA_WATERSHED)){
 print(paste("increase: ", fAREAWSHD_DIV, "buffer size: ",fBUFFER))
 #increase buffer size

 #buffer the lake so that it roughly aproximates its watershed

 # set buffer width based on previous buffer width
 fBUFFER <- fBUFFER*(1+fBUFF_CHANGE_FRAC)#in m

 # Create the buffered lake geometry
 pLAKE_BUF <- st_buffer(pLAKE, dist = fBUFFER)

 # Cut the buffered lake by the selected basin polygons
 pLAKE_WSHD <- st_intersection(pLAKE_BUF,pBASINS)

 fAREA_WATERSHED = sum(as.numeric(st_area(pLAKE_WSHD)))/1000000

 #fraction of calculated watershed area relative to the desired
(hydrolakes) watershed size
 fAREAWSHD_DIV = 1.0-
(min(as.numeric(pLAKE$Wshd_area)+as.numeric(pLAKE$Lake_area),as.numeric
(fAREA_WATERSHED))/max(as.numeric(pLAKE$Wshd_area)+as.numeric(pLAKE$Lak
e_area),as.numeric(fAREA_WATERSHED)))

 }else{
 print(paste("break: ", fAREAWSHD_DIV))
 break()

 }

 }
```

```
 pLAKE_WSHD=st_union(pLAKE_WSHD)

 #Check if watershed and soil map match
 pSOIL_WSHD = st_intersection(shapeSOILS, pLAKE_WSHD)

 WSHD_area <- drop_units(st_area(pLAKE_WSHD))
 Soil_area = drop_units(sum(st_area(pSOIL_WSHD)))

 #calculate if there is soil map fill in the WSHD, we say if there is
less then 50% fill we do not move forward with said lake
 if ((WSHD_area - Soil_area)/WSHD_area<fTHRESH_SOIL){

 #Overlappende polygonen tussen shapeCorine en pLAKE_WSHD vinden
 Corine_intersects <- st_intersects(shapeCORINE, pLAKE_WSHD, sparse =
FALSE)
 shapeCORINE_crop <- shapeCORINE[apply(Corine_intersects, 1, any), ]

 #Intersection uitvoeren
 pLANDCOVER_WSHD <- st_intersection(shapeCORINE_crop, pLAKE_WSHD)


 #build
 st_area(pLANDCOVER_WSHD)


 #Implement BATT

 #Make empty BATT df
 BATT <- data.frame()

 #Add Soil map to Corine landcover
 ##Check if both shapefiles have a valid CRS
 pLANDCOVER_WSHD <- st_transform(pLANDCOVER_WSHD, st_crs(shapeSOILS))
 Landcover <- st_transform(pLANDCOVER_WSHD, st_crs(shapeSOILS))


 ##Spatial join between land use and soil types
 landuse_soil <- st_join(Landcover, shapeSOILS, join = st_intersects)

 landuse_soil$area_new =
(as.numeric(unlist(st_area(landuse_soil)))/10000)*2.47105381

 landuse_soil$Area_Ac <- (landuse_soil$Area_Ha * 2.47105381)
 BATT <- landuse_soil %>%
 group_by(Code_18, Soil_Code) %>%
 summarise(Total_Area_Ac = sum(area_new))

 #Merge BATT and CLC_Legend by Code_18
 BATT <- merge(BATT, CLC_legend[,c("Code_18", "Corine_Landcover")], by
= "Code_18", all.x = TRUE)

 #Add hydrological soil typing
 BATT <- merge(BATT, Soil_types[,c("Soil_Code", "Soil_Type")], by =
"Soil_Code", all.x = TRUE)

 #Add Runoff coefficient
 BATT <- merge(BATT, Runoff_Coefficient[,c("Runoff_coefficient",
"Corine_Landcover")], by = "Corine_Landcover", all.x = TRUE)
```

```
 #Add Curve number
 ##Create temporary df
 Temp_BATT <- BATT %>%
 left_join(Curve_Number, by = "Corine_Landcover")

 ##Create a new column based on Soil_Type to get the corresponding
value from columns A, B, C, D
 Temp_BATT <- Temp_BATT %>%
 mutate("Curve_Number" = case_when(
 Soil_Type == "A" ~ A,
 Soil_Type == "B" ~ B,
 Soil_Type == "C" ~ C,
 Soil_Type == "D" ~ D
 ))
 #Merge
 BATT$Curve_Number <- Temp_BATT$Curve_Number


 #Merge BATT with Conversion table
 BATT <- merge(BATT,
Conversion[,c("BATT_Landcover","Type_of_LC","Corine_Landcover","Phospho
rous_lb_ac_yr","Nitrogen_lb_ac_yr","TSS_lb_ac_yr")], by =
c("Corine_Landcover"), all.x = TRUE)


 #Calculate P & N loading, Potential max retention, Direct surface
runoff
 ##Calculate P, N, TSS if soil is Pervious and NA
 BATT <- BATT %>%
 left_join(Loading, by = "Soil_Type") %>%
 mutate(
 Phosphorous_lb_ac_yr = ifelse(is.na(Phosphorous_lb_ac_yr.x),
Phosphorous_lb_ac_yr.y, Phosphorous_lb_ac_yr.x),
 Nitrogen_lb_ac_yr = ifelse(is.na(Nitrogen_lb_ac_yr.x),
Nitrogen_lb_ac_yr.y, Nitrogen_lb_ac_yr.x),
 TSS_lb_ac_yr = ifelse(is.na(TSS_lb_ac_yr.x), TSS_lb_ac_yr.y,
TSS_lb_ac_yr.x)
 ) %>%
 select(-ends_with(".x"), -ends_with(".y"))

 BATT <- BATT %>%
 select(-contains(".y"))


 #kg/km2/yr
 BATT$Phosphorous_kg_km2_yr = BATT$Phosphorous_lb_ac_yr*112.09
 BATT$Nitrogen_kg_km2_yr = BATT$Nitrogen_lb_ac_yr*112.09
 BATT$TSS_kg_km2_yr = BATT$TSS_lb_ac_yr*112.09
 BATT$Total_Area_km2 = BATT$Total_Area_Ac*0.004046860338725
 BATT$Land2Lake_Area = BATT$Total_Area_km2/ pLAKE$Lake_area

 # Calculate potential maximum retention based on Curve Number method
 BATT$Potential_maximum_retention <- (25400/BATT$Curve_Number)-254

 # Estimate direct surface runoff
```

```
 BATT$Direct_surface_runoff <- ((pLAKE$pre_mm_uyr-
0.2*BATT$Potential_maximum_retention)^2)/(pLAKE$pre_mm_uyr+0.8*BATT$Pot
ential_maximum_retention)

 # Convert nutrient and sediment loads from per kmÂ² to total yearly
load towards the lake
 BATT$Phosphorous_kg_yr <- BATT$Phosphorous_kg_km2_yr *
BATT$Total_Area_km2 / BATT$Land2Lake_Area
 BATT$Nitrogen_kg_yr <- BATT$Nitrogen_kg_km2_yr * BATT$Total_Area_km2 /
BATT$Land2Lake_Area
 BATT$TSS_kg_yr <- BATT$TSS_kg_km2_yr * BATT$Total_Area_km2 /
BATT$Land2Lake_Area

 #calculate Qin based on hydrolakes residence time
 #residence time in days
 #average depth in m *1000 for mm
 fQIN_HYDROLAKES = (pLAKE$Depth_avg*1000)/pLAKE$Res_time #mm/day

 #calculate concentrations of runoff according to BATT
 ##Phosphorous
 BATT$Phosphorous_g_m2_day =
(BATT$Phosphorous_kg_yr*1000/365)/(pLAKE$Lake_area*1000000)
 BATT$Phosphorous_mg_L = (BATT$Phosphorous_g_m2_day /
(BATT$Direct_surface_runoff/365))*1000

 ##Nitrogen
 BATT$Nitrogen_g_m2_day =
(BATT$Nitrogen_kg_yr*1000/365)/(pLAKE$Lake_area*1000000)
 BATT$Nitrogen_mg_L = (BATT$Nitrogen_g_m2_day /
(BATT$Direct_surface_runoff/365))*1000

 ##TSS
 BATT$TSS_g_m2_day =
(BATT$TSS_kg_yr*1000/365)/(pLAKE$Lake_area*1000000)
 BATT$TSS_mg_L = (BATT$TSS_g_m2_day /
(BATT$Direct_surface_runoff/365))*1000

 #weighted average concentration based on area
 library(stats)
 fP_CONC_WA= weighted.mean( BATT$Phosphorous_mg_L, BATT$Total_Area_Ac,
na.rm = TRUE )
 fN_CONC_WA= weighted.mean( BATT$Nitrogen_mg_L, BATT$Total_Area_Ac,
na.rm = TRUE )
 fT_CONC_WA= weighted.mean( BATT$TSS_mg_L, BATT$Total_Area_Ac, na.rm =
TRUE )

 #Total
 fTotal_Phosphorous = sum(BATT$Phosphorous_kg_yr)
 fTotal_Nitrogen = sum(BATT$Nitrogen_kg_yr)
 fTotal_TSS = sum(BATT$TSS_kg_yr)

 # Estimate total nutrient and sediment loads to the lake (g/m2/day)
 fPLOAD_TOTAL = (fP_CONC_WA * fQIN_HYDROLAKES)/1000
 fNLOAD_TOTAL = fN_CONC_WA * (fQIN_HYDROLAKES/1000)
 fTLOAD_TOTAL = fT_CONC_WA * (fQIN_HYDROLAKES/1000)
```

```
 #BATT_BMP
 ##BMP Grass swale
 # Select relevant columns including land cover type, area, nutrient
loads, and runoff
 BATT_BMP_gs <-
BATT[,c("BATT_Landcover","Corine_Landcover","Type_of_LC","Total_Area_km
2", "Phosphorous_kg_yr","Nitrogen_kg_yr","TSS_kg_yr",
"Direct_surface_runoff", "geometry")]
 #"geometry",
 BATT_BMP_gs = st_make_valid(BATT_BMP_gs)
 BATT_BMP_gs = st_transform(BATT_BMP_gs, crs=st_crs(shapeCORINE))

 #Change crs from pLAKE_WSHD to wgs84 for osmdata
 pLAKE_WSHD_wgs84 <- st_transform(pLAKE_WSHD, crs = 4326)

 #Create boundingbox from pLAKE_WSHD
 bbox <- st_bbox(pLAKE_WSHD_wgs84)

 # Query OpenStreetMap (OSM) data for pathway-related features within
the study area
 pathways <- opq(bbox = c(bbox["xmin"], bbox["ymin"], bbox["xmax"],
bbox["ymax"])) %>%
 add_osm_feature(
 key = 'highway',
 value = c('footway', 'living_street', 'pedestrian', 'sidewalk',
'cycleway', 'motorway')
 ) %>%
 osmdata_sf()

 # Reproject OSM pathway data to match the CRS of the CORINE land cover
layer
 # Check if osm_lines exist, otherwise create an empty sf collection
with the correct CRS
 if (is.null(pathways$osm_lines) || nrow(pathways$osm_lines) == 0) {
 # No pathways found -> GS values equal to total values
 fPLOAD_TOTAL_GS <- fPLOAD_TOTAL
 fNLOAD_TOTAL_GS <- fNLOAD_TOTAL
 fTLOAD_TOTAL_GS <- fTLOAD_TOTAL
 BATT_BMP_gs$Runoff_reduced <- BATT$Direct_surface_runoff
 } else {

 pathways$osm_lines <- st_transform(pathways$osm_lines, crs =
st_crs(shapeCORINE))

 # Perform spatial intersection to extract overlapping areas between
BATT land areas and pathway lines
 Pathways_intersect <- st_intersection(BATT_BMP_gs, pathways$osm_lines)

 # Filter for right land use
 Landcover_list <- list("High Density Residential","Middle Density
Residential","Low Density Residential (single family)","Highways",
"Commercial","Industrial")
 Pathways_filtered<- Pathways_intersect %>%
 filter(BATT_Landcover %in% Landcover_list & Type_of_LC == "I")

 # Get length and width of pathways
 Pathways_filtered$Length <- st_length(Pathways_filtered)
```

```
 Pathways_filtered <- merge(Pathways_filtered, Pathway_width, by =
"highway", all.x = TRUE)

 # Calculate storage capacity
 Pathways_filtered$DSV_storage <- 0.5 * Pathways_filtered$Length * 0.3
#per pathway, m3
 BATT_BMP_gs$DSV_capacity = ifelse(BATT_BMP_gs$BATT_Landcover %in%
Pathways_filtered$BATT_Landcover,
(sum(Pathways_filtered$DSV_storage)*1000)/(sum(Pathways_filtered$Length
*0.5)), NA) #total capacity per land cover


 #Grass swale area
 # Estimate the potential area for grass swales alongside selected
pathways
 # Assumes a 0.5 meter width of green infrastructure per meter of
pathway length
 Pathways_filtered$Green_area = 0.5 * Pathways_filtered$Length

 # Summarize total bioswale (green area) surface in square meters
 Green_area <- data.frame(
 Bioswale_total_area_m2 = sum(Pathways_filtered$Green_area)
 )

 # Estimate total water storage capacity of the bioswales (liters per
mÂ²)
 Green_area$Bioswale_storage_L_m2 = sum(Pathways_filtered$DSV_storage,
na.rm = TRUE)

 #Calculate reduced surface runoff due to green infrastructure
 BATT_BMP_gs <- BATT_BMP_gs %>%
 mutate(Runoff_reduced =
ifelse(!is.na(DSV_capacity),(Direct_surface_runoff-
DSV_capacity),Direct_surface_runoff))

 # Calculate the ratio between contributing land area and lake area
 BATT_BMP_gs$ratio <-BATT_BMP_gs$Total_Area_km2/ pLAKE$Lake_area

 # Adjust surface runoff based on reduced runoff and land-to-lake ratio
 BATT_BMP_gs$Surface_runoff_area <- BATT_BMP_gs$Runoff_reduced *
BATT_BMP_gs$ratio

 # Calculate the fraction of runoff that remains
 BATT_BMP_gs$runoff_reduction_fraction =
BATT_BMP_gs$Runoff_reduced/BATT_BMP_gs$Direct_surface_runoff

 # Estimate the adjusted total inflow (Qin) to the lake
 fQIN_HYDROLAKES_gs =
weighted.mean(BATT_BMP_gs$runoff_reduction_fraction* fQIN_HYDROLAKES,
BATT_BMP_gs$Total_Area_km2 , na.rm = TRUE )


 ##Nutrient reduction
 #Sum area per landcover per pathway
 Pathways_filtered$area <-
Pathways_filtered$Length*Pathways_filtered$Width #m2
 fTotal_A_pathway= sum(Pathways_filtered$area)/1000000 #km2
```

```
 #Calculate area (fraction) of pathway in land use cat
 #apply a factor for runoff through grass swales of 8:1 (8 times the
area goes through the swale)
 fCONV_RUNOFF = 1#8/1
 BATT_BMP_gs$Perc_Area = ifelse(BATT_BMP_gs$BATT_Landcover %in%
Pathways_filtered$BATT_Landcover, min(BATT_BMP_gs$Total_Area_km2,
(fTotal_A_pathway*fCONV_RUNOFF)/BATT_BMP_gs$Total_Area_km2), NA)


 # Recalculate annual loads of phosphorus, nitrogen, and TSS based on
the treated area (Perc_Area), applying reduction factors to the treated
portion while keeping untreated loads unchanged
 #Phosphorous
 BATT_BMP_gs <- BATT_BMP_gs %>%
 mutate(Phosphorous_kg_yr_reduced =
ifelse(!is.na(Perc_Area),(Phosphorous_kg_yr*Perc_Area*0.64)+(Phosphorou
s_kg_yr*(1-Perc_Area)),Phosphorous_kg_yr))
 #Nitrogen
 BATT_BMP_gs <- BATT_BMP_gs %>%
 mutate(Nitrogen_kg_yr_reduced =
ifelse(!is.na(Perc_Area),(((Nitrogen_kg_yr/100*Perc_Area)*0.7687)+(Nitr
ogen_kg_yr/100*(100-Perc_Area))),Nitrogen_kg_yr))

 #TSS
 BATT_BMP_gs <- BATT_BMP_gs %>%
 mutate(TSS_kg_yr_reduced =
ifelse(!is.na(Perc_Area),(((TSS_kg_yr/100*Perc_Area)*0.1)+(TSS_kg_yr/10
0*(100-Perc_Area))),TSS_kg_yr))


 #calculate concentrations of runoff according to BATT
 ##Phosphorous
 BATT_BMP_gs$Phosphorous_g_m2_day =
(BATT_BMP_gs$Phosphorous_kg_yr_reduced*1000/365)/(pLAKE$Lake_area*10000
00)
 BATT_BMP_gs$Phosphorous_mg_L = (BATT_BMP_gs$Phosphorous_g_m2_day /
(BATT_BMP_gs$Runoff_reduced/365))*1000

 ##Nitrogen
 BATT_BMP_gs$Nitrogen_g_m2_day =
(BATT_BMP_gs$Nitrogen_kg_yr_reduced*1000/365)/(pLAKE$Lake_area*1000000)
 BATT_BMP_gs$Nitrogen_mg_L = (BATT_BMP_gs$Nitrogen_g_m2_day /
(BATT_BMP_gs$Runoff_reduced/365))*1000

 ##TSS
 BATT_BMP_gs$TSS_g_m2_day =
(BATT_BMP_gs$TSS_kg_yr_reduced*1000/365)/(pLAKE$Lake_area*1000000)
 BATT_BMP_gs$TSS_mg_L = (BATT_BMP_gs$TSS_g_m2_day /
(BATT_BMP_gs$Runoff_reduced/365))*1000

 #weighted average concentration based on area
 library(stats)
 fP_CONC_WA_GS= weighted.mean( BATT_BMP_gs$Phosphorous_mg_L,
BATT_BMP_gs$Total_Area_km2, na.rm = TRUE )
 fN_CONC_WA_GS= weighted.mean( BATT_BMP_gs$Nitrogen_mg_L,
BATT_BMP_gs$Total_Area_km2, na.rm = TRUE )
 fT_CONC_WA_GS= weighted.mean( BATT_BMP_gs$TSS_mg_L,
BATT_BMP_gs$Total_Area_km2, na.rm = TRUE )
```

```
 #Total
 fTotal_Phosphorous_Pathways =
sum(BATT_BMP_gs$Phosphorous_kg_yr_reduced)
 fTotal_Nitrogen_Pathways = sum(BATT_BMP_gs$Nitrogen_kg_yr_reduced)
 fTotal_TSS_Pathways = sum(BATT_BMP_gs$TSS_kg_yr_reduced)

 # Estimate total nutrient and sediment loads to the lake (g/m2/day)
 fPLOAD_TOTAL_GS = fP_CONC_WA * (fQIN_HYDROLAKES_gs/1000)
 fNLOAD_TOTAL_GS = fN_CONC_WA * (fQIN_HYDROLAKES_gs/1000)
 fTLOAD_TOTAL_GS = fT_CONC_WA * (fQIN_HYDROLAKES_gs/1000)

 }

 ##BMP gravel wetland
 # Select relevant columns including land cover type, area, nutrient
loads, and runoff
 BATT_BMP_gw <-
BATT[,c("BATT_Landcover","Corine_Landcover","Type_of_LC",
"Phosphorous_kg_yr","Nitrogen_kg_yr","TSS_kg_yr", "Total_Area_km2",
"Phosphorous_kg_km2_yr","Nitrogen_kg_km2_yr","TSS_kg_km2_yr",
"Direct_surface_runoff","geometry")]
 #"geometry",
 BATT_BMP_gw = st_make_valid(BATT_BMP_gw)
 BATT_BMP_gw = st_transform(BATT_BMP_gw, crs=st_crs(shapeCORINE))

 # Query OpenStreetMap (OSM) data for pathway-related features within
the study area
 ##note: use same bbox as OSM query grass swale
 buildings <- opq(bbox = c(bbox["xmin"], bbox["ymin"], bbox["xmax"],
bbox["ymax"])) %>%
 add_osm_feature(
 key = 'building',
 value =
c('residential','apartments','terrace','house','detached','annexe','hot
el','semidetached_house','commercial','industrial','office','retail','s
upermarket','warehouse','college','government','university')
 ) %>%
 osmdata_sf()

 if (is.null(buildings$osm_polygons) || nrow(buildings$osm_polygons) ==
0) {
 # No buildings found -> GW values equal to total values
 fPLOAD_TOTAL_GW <- fPLOAD_TOTAL
 fNLOAD_TOTAL_GW <- fNLOAD_TOTAL
 fTLOAD_TOTAL_GW <- fTLOAD_TOTAL
 BATT_BMP_gw$Runoff_reduced <- BATT$Direct_surface_runoff
 } else {


 # Transform building polygons to match CRS of reference layer and find
spatial intersections with BMP areas
 buildings$osm_polygons <- st_transform(buildings$osm_polygons , crs =
st_crs(shapeCORINE))

 Buildings_intersect <- st_intersection(BATT_BMP_gw,
buildings$osm_polygons )
```

```
 #Filter for right land use
 Landcover_list_buildings <- list("High Density Residential","Middle
Density Residential","Low Density Residential (single family)",
"Commercial","Industrial")
 filtered<- Buildings_intersect %>%
 filter(BATT_Landcover %in% Landcover_list_buildings & Type_of_LC ==
"I")

 #Get longest side of polygon (m)
 filtered$Longest_side <- lapply(filtered$geometry, function(geom) {
 # Convert the polygon to lines (boundary)
 boundary <- st_boundary(geom)

 # Extract the coordinates of the boundary (edges of the polygon)
 coords <- st_coordinates(boundary)

 # Calculate the Euclidean distance between consecutive points
 side_lengths <- sqrt(diff(coords[, 1])^2 + diff(coords[, 2])^2)

 return(max(side_lengths))
 })


 # Calculate storage volume for Gravel Wetlands (DSV_storage) based on
dimensions and porosity
 ##0.4 is the porosity
 filtered$DSV_storage <- sapply(filtered$Longest_side,
function(longest_side) {
 # DSV formula for each Longest_side
 return((0.5 * longest_side * 0.1) + (0.5 * longest_side * 0.1) + (0.5
* longest_side * 0.1 * 0.4))
 })

 #Wetland area
 filtered$Green_area <- sapply(filtered$Longest_side,
function(longest_side) {
 # DSV formula for each Longest_side
 return(0.5 * longest_side )
 })
 Green_area$Wadi_total_area_m2 = sum(unlist(filtered$Green_area))


 # Calculate the total length of all Longest_side values
 sumLength = sum(unlist(filtered$Longest_side))

 # Calculate DSV capacity
 BATT_BMP_gw$DSV_capacity = ifelse(BATT_BMP_gw$BATT_Landcover %in%
filtered$BATT_Landcover,(sum(filtered$DSV_storage)*1000)/sum(sumLength*
0.5), NA)

 #Calculate reduced surface runoff due to green infrastructure
 BATT_BMP_gw <- BATT_BMP_gw %>%
 mutate(Runoff_reduced =
ifelse(!is.na(DSV_capacity),(Direct_surface_runoff-
DSV_capacity),Direct_surface_runoff))

 # Calculate the ratio between contributing land area and lake area
```

```
 BATT_BMP_gw$ratio <- BATT_BMP_gw$Total_Area_km2/pLAKE$Lake_area

 # Adjust surface runoff based on reduced runoff and land-to-lake ratio
 BATT_BMP_gw$Surface_runoff_area <- BATT_BMP_gw$Runoff_reduced *
BATT_BMP_gw$ratio

 # Calculate the fraction of runoff that remains
 BATT_BMP_gw$runoff_reduction_fraction =
BATT_BMP_gw$Runoff_reduced/BATT_BMP_gw$Direct_surface_runoff

 # Estimate the adjusted total inflow (Qin) to the lake
 fQIN_HYDROLAKES_gw =
weighted.mean(BATT_BMP_gw$runoff_reduction_fraction* fQIN_HYDROLAKES,
BATT_BMP_gw$Total_Area_km2 , na.rm = TRUE )

 # Estimate total water storage capacity of the gravel wetlands (liters
per mÂ²)
 Green_area$Wadi_storage_L_m2 = sum(BATT_BMP_gw$DSV_capacity, na.rm =
TRUE)


 ##Nutrient reduction
 #Sum area per landcover per building
 filtered$Building_area <- st_area(filtered$geometry)
 fTotal_A_building = sum(filtered$Building_area)/1000000 #km2

 #Calculate reduced nutrient
 BATT_BMP_gw$Perc_Area = ifelse(BATT_BMP_gw$BATT_Landcover %in%
filtered$BATT_Landcover, 100/BATT_BMP_gw$Total_Area_km2 *
fTotal_A_building, NA)

 # Recalculate annual loads of phosphorus, nitrogen, and TSS based on
the treated area (Perc_Area), applying reduction factors to the treated
portion while keeping untreated loads
 #Phosphorous
 BATT_BMP_gw <- BATT_BMP_gw %>%
 mutate(Phosphorous_kg_yr_reduced =
ifelse(!is.na(Perc_Area),(((Phosphorous_kg_yr/100*Perc_Area)*0.34)+(Pho
sphorous_kg_yr/100*(100-Perc_Area))),Phosphorous_kg_yr))
 #Nitrogen
 BATT_BMP_gw <- BATT_BMP_gw %>%
 mutate(Nitrogen_kg_yr_reduced =
ifelse(!is.na(Perc_Area),(((Nitrogen_kg_yr/100*Perc_Area)*0.21)+(Nitrog
en_kg_yr/100*(100-Perc_Area))),Nitrogen_kg_yr))

 #TSS
 BATT_BMP_gw <- BATT_BMP_gw %>%
 mutate(TSS_kg_yr_reduced =
ifelse(!is.na(Perc_Area),(((TSS_kg_yr/100*Perc_Area)*0.01)+(TSS_kg_yr/1
00*(100-Perc_Area))),TSS_kg_yr))



 #calculate concentrations of runoff according to BATT
 ##Phosphorous
 BATT_BMP_gw$Phosphorous_g_m2_day =
(BATT_BMP_gw$Phosphorous_kg_yr_reduced*1000/365)/(pLAKE$Lake_area*10000
00)
```

```
 BATT_BMP_gw$Phosphorous_mg_L = (BATT_BMP_gw$Phosphorous_g_m2_day /
(BATT_BMP_gw$Runoff_reduced/365))*1000

 ##Nitrogen
 BATT_BMP_gw$Nitrogen_g_m2_day =
(BATT_BMP_gw$Nitrogen_kg_yr_reduced*1000/365)/(pLAKE$Lake_area*1000000)
 BATT_BMP_gw$Nitrogen_mg_L = (BATT_BMP_gw$Nitrogen_g_m2_day /
(BATT_BMP_gw$Runoff_reduced/365))*1000

 ##TSS
 BATT_BMP_gw$TSS_g_m2_day =
(BATT_BMP_gw$TSS_kg_yr*1000/365)/(pLAKE$Lake_area*1000000)
 BATT_BMP_gw$TSS_mg_L = (BATT_BMP_gw$TSS_kg_yr_reduced /
(BATT_BMP_gw$Runoff_reduced/365))*1000

 #weighted average concentration based on area
 library(stats)
 fP_CONC_WA_GW= weighted.mean( BATT_BMP_gw$Phosphorous_mg_L,
BATT_BMP_gw$Total_Area_km2, na.rm = TRUE )
 fN_CONC_WA_GW= weighted.mean( BATT_BMP_gw$Nitrogen_mg_L,
BATT_BMP_gw$Total_Area_km2, na.rm = TRUE )
 fT_CONC_WA_GW= weighted.mean( BATT_BMP_gw$TSS_mg_L,
BATT_BMP_gw$Total_Area_km2, na.rm = TRUE )


 #Total
 fTotal_Phosphorous_Buildings =
sum(BATT_BMP_gw$Phosphorous_kg_yr_reduced)
 fTotal_Nitrogen_Buildings = sum(BATT_BMP_gw$Nitrogen_kg_yr_reduced)
 fTotal_TSS_Buildings = sum(BATT_BMP_gw$TSS_kg_yr_reduced)

 # Estimate total nutrient and sediment loads to the lake (g/m2/day)
 fPLOAD_TOTAL_GW = fP_CONC_WA * (fQIN_HYDROLAKES_gw/1000)
 fNLOAD_TOTAL_GW = fN_CONC_WA * (fQIN_HYDROLAKES_gw/1000)
 fTLOAD_TOTAL_GW = fT_CONC_WA * (fQIN_HYDROLAKES_gw/1000)

 }

 #calculate temperature based on hydrolakes monthly temperatures

 ##note: we run two years of water temperature to get rid of the
influence of initial temperatures at T=1 which we do not know

 #Extract monthly average air temperatures from pLAKE
 vTEMP_AIR =
c(unlist(pLAKE[,c("tmp_dc_l12")])[1]/10,unlist(pLAKE[,c("tmp_dc_l01","t
mp_dc_l02","tmp_dc_l03","tmp_dc_l04","tmp_dc_l05","tmp_dc_l06","tmp_dc_
l07","tmp_dc_l08","tmp_dc_l09","tmp_dc_l10","tmp_dc_l11","tmp_dc_l12")]
)[c(1:12)]/10,unlist(pLAKE[,c("tmp_dc_l12")])[1]/10)

 #Define the day of year corresponding to temperature measurements
 vDAYS_TEMP_AIR = c(1,16,45,75,105,136,166,197,228,258,289,319,350,365)

 #Duplicate temperature and day vectors to simulate two consecutive
years
 vTEMP_AIR = c(vTEMP_AIR,vTEMP_AIR)
 vDAYS_TEMP_AIR = c(vDAYS_TEMP_AIR, vDAYS_TEMP_AIR+365)
```

```
 #Combine days and temperatures into a data frame
 dfTEMP <- data.frame(day = vDAYS_TEMP_AIR,

    temp = vTEMP_AIR)

 library(dplyr)

 library(zoo)

 #Create a full daily time series for two years
 dfTEMP_YEAR = data.frame(day = seq(from=1, to=730, by = 1)) %>%

 full_join(dfTEMP, by = "day") %>%

 mutate(approx = na.approx(temp))


 #Constants for temperature model from Tjeukemeer calibration
 fH_CONS_MOOIJ   =     0.0269#0.0165 #h =     0.0083

 fF_CONS_MOOIJ   =     0.0100#0.0109 #f =     0.0072

 fG_CONS_MOOIJ   =     0.0432#0.0271 #g =     0.0017

 #Initial water temperature value for simulation start (degrees
Celsius)
 fTEMP_WATER_INIT=     8.0

 #Initialize empty vector to store water temperature results
 vTEMP=c()

 for(fTIME in c(1:730)){

 fTEMP_AIR =     as.numeric(dfTEMP_YEAR[fTIME,"approx"])

 # Get the current day of the simulation
 fDAY      =     dfTEMP_YEAR[fTIME, "day"]

 # For the first timestep, set water temperature to the initial value;
for subsequent timesteps use the previous water temperature
 if(fTIME   == 1){

 fTEMP_WATER_TMIN1     =     fTEMP_WATER_INIT

 }else{

 fTEMP_WATER_TMIN1     =     fTEMP_WATER

 }

 # Calculate the new water temperature based on previous water
temperature, air temperature, and seasonal sinusoidal variation
 fTEMP_WATER           =     fTEMP_WATER_TMIN1+
      fH_CONS_MOOIJ*(fTEMP_AIR-
fTEMP_WATER_TMIN1)+fF_CONS_MOOIJ+fG_CONS_MOOIJ*sin(2*pi*((fDAY-
81)/365.25))

 vTEMP     =     c(vTEMP,fTEMP_WATER)
```

```
 }



 # Extract the last 365 days from the two-year temperature vector
 vTEMP_365 <- tail(vTEMP, 365)

 # Convert the extracted temperatures into a 1-row matrix
 Temp_matrix <- matrix(vTEMP_365, nrow = 1, ncol = 365)

 # Assign column names to the matrix representing each day of the year
 colnames(Temp_matrix) <- paste0("Tm_Day_", 1:365)

 #Save all elements
 if(exists("PC_Lake")==TRUE){
 PC_Lake <- rbind.data.frame(PC_Lake, data.frame(
 LakeID = pLAKE$Hylak_id,
 Runoff = (sum(BATT$Direct_surface_runoff,
na.rm=TRUE)+pLAKE$pre_mm_lyr)/365,
 LakeDepth = pLAKE$Depth_avg,
 LakeArea = pLAKE$Lake_area,
 Latitude = pLAKE$Pour_lat,
 Phosphorous = fPLOAD_TOTAL,
 Nitrogen = fNLOAD_TOTAL,
 TSS = fTLOAD_TOTAL,
 Runoff_gs = (sum(BATT_BMP_gs$Runoff_reduced,
na.rm=TRUE)+pLAKE$pre_mm_lyr)/365,
 Phosphorous_gs = fPLOAD_TOTAL_GS,
 Nitrogen_gs = fNLOAD_TOTAL_GS,
 TSS_gs = fTLOAD_TOTAL_GS,
 Runoff_gw = (sum(BATT_BMP_gw$Runoff_reduced,
na.rm=TRUE)+pLAKE$pre_mm_lyr)/365,
 Phosphorous_gw = fPLOAD_TOTAL_GW,
 Nitrogen_gw = fNLOAD_TOTAL_GW,
 TSS_gw = fTLOAD_TOTAL_GW,
 Clay_fraction = pLAKE$cly_pc_vav,
 Silt_fraction = pLAKE$slt_pc_vav,
 Sand_fraction = pLAKE$snd_pc_vav,
 Temp_matrix
 ))} else{
 PC_Lake <- data.frame(
  LakeID = pLAKE$Hylak_id,
  Runoff = (sum(BATT$Direct_surface_runoff,
na.rm=TRUE)+pLAKE$pre_mm_lyr)/365,
  LakeDepth = pLAKE$Depth_avg,
  LakeArea = pLAKE$Lake_area,
  Latitude = pLAKE$Pour_lat,
  Phosphorous = fPLOAD_TOTAL,
  Nitrogen = fNLOAD_TOTAL,
  TSS = fTLOAD_TOTAL,
  Runoff_gs = (sum(BATT_BMP_gs$Runoff_reduced,
na.rm=TRUE)+pLAKE$pre_mm_lyr)/365,
  Phosphorous_gs = fPLOAD_TOTAL_GS,
  Nitrogen_gs = fNLOAD_TOTAL_GS,
  TSS_gs = fTLOAD_TOTAL_GS,
```

```
  Runoff_gw = (sum(BATT_BMP_gw$Runoff_reduced,
na.rm=TRUE)+pLAKE$pre_mm_lyr)/365,
  Phosphorous_gw = fPLOAD_TOTAL_GW,
  Nitrogen_gw = fNLOAD_TOTAL_GW,
  TSS_gw = fTLOAD_TOTAL_GW,
  Clay_fraction = pLAKE$cly_pc_vav,
  Silt_fraction = pLAKE$slt_pc_vav,
  Sand_fraction = pLAKE$snd_pc_vav,
  Temp_matrix
 )
 }

 }
}#end for loop over lakes

fwrite(PC_Lake, file.path(file.path(dirShell, "work_cases",
nameWorkCase, "output"),paste("PCLake_input",".csv",sep="")))

#write selected lakes to shp
st_write(shapeLAKES[which(shapeLAKES$Hylak_id %in% PC_Lake$LakeID),],
file.path(file.path(dirShell, "work_cases", nameWorkCase,
"output"),paste("NICHES_selected_lakes",".shp",sep="")))

cbind.data.frame(LakeID=as.data.frame(shapeLAKES[which(shapeLAKES$Hylak
_id %in%
PC_Lake$LakeID),"Hylak_id"])[1],as.data.frame(st_coordinates(st_centroi
d(shapeLAKES[which(shapeLAKES$Hylak_id %in% PC_Lake$LakeID),]))))


#define final output data for PCLake and remove it if it exists
(overwrite on)
if (exists("dtOUT_AGG")==TRUE){
 rm("dtOUT_AGG")
}

##CLUSTER VARIANT ON EULER INTEGRATOR##----
library(doSNOW)
library(foreach)

dfCOMBS = expand.grid(lake_no = c(1:nrow(PC_Lake)), scen = c("BAU",
"NBS_bioswale", "NBS_wadi"))

#make a cluster for calculations
nTHREADS=11
snowCLUSTER <- makeCluster(nTHREADS)
clusterExport(snowCLUSTER, c())
registerDoSNOW(snowCLUSTER)
pb<-txtProgressBar(0,nrow(dfCOMBS),style=3)
progress<-function(n){
 setTxtProgressBar(pb,n)
}
opts<-list(progress=progress)

comb <- function(x, ...) {
 lapply(seq_along(x),
  function(i) c(x[[i]], lapply(list(...), function(y) y[[i]])))
}
```

```
## 3. Make and adjust cpp files
## - nRUN_SET determines which forcings are switched on
PCModelAdjustCPPfiles(dirSHELL = dirShell,
   nameWORKCASE = nameWorkCase,
   lDATM = lDATM_SETTINGS,
   nRUN_SET = 0)


## 4. Compile model
PCModelCompileModelWorkCase(dirSHELL = dirShell,
    nameWORKCASE = nameWorkCase)

lOUT <-
foreach(i=c(1:nrow(dfCOMBS)),.combine='rbind', .multicombine=TRUE, .pac
kages=c('data.table', 'plyr', "dplyr","stringr"),.export =
c("dirHome"),.options.snow=opts) %dopar% {

 source(file.path(dirShell, "scripts", "R_system", "functions.R")) ##
load base functions by Luuk van Gerven (2012-2016)
 source(file.path(dirShell, "scripts", "R_system",
"functions_PCLake.R"))
 nLAKE = dfCOMBS$lake_no[i]
 sSCEN = dfCOMBS$scen[i]
 PC_Lake_SEL = PC_Lake[nLAKE,]
 if(complete.cases(PC_Lake_SEL)==FALSE){

 }else{
 #set sediment based on sand and clay fractions, assuming if it is not
clay, sand, or clay/sand it has to be peat
 if(PC_Lake_SEL$Sand_fraction>33 & PC_Lake_SEL$Clay_fraction>33){
 sPCLAKE_SED_NAME ="clay_sand"
 }else if(PC_Lake_SEL$Sand_fraction>33){
 sPCLAKE_SED_NAME ="sand"
 }else if(PC_Lake_SEL$Clay_fraction>33){
 sPCLAKE_SED_NAME ="clay"
 }else{
 sPCLAKE_SED_NAME ="peat"
 }

 ## Optional: change sediment settings
 lDATM_SETTINGS$params <-
adjustSedimentParamSettings_inclBank(lDATM_SETTINGS$params, paramset =
2, sediment_type = sPCLAKE_SED_NAME)
 #set depth

lDATM_SETTINGS$params$sDefault0[which(rownames(lDATM_SETTINGS$params)==
'cDepthWInit0')] = PC_Lake_SEL$LakeDepth
 #set fetch

lDATM_SETTINGS$params$sDefault0[which(rownames(lDATM_SETTINGS$params)==
'cFetch')] = sqrt(PC_Lake_SEL$LakeArea*1000000)
 #set latitude

lDATM_SETTINGS$params$sDefault0[which(rownames(lDATM_SETTINGS$params)==
'cLAT')] = PC_Lake_SEL$Latitude
```

```
 vTEMP =
as.vector(unlist(PC_Lake_SEL[,str_detect(colnames(PC_Lake_SEL),"Tm_Day_
")]))

 if(sSCEN == "BAU"){
 #adjust forcings
 lDATM_SETTINGS$forcings$sDefault0$mPLoadEpi$value =
PC_Lake_SEL$Phosphorous
 lDATM_SETTINGS$forcings$sDefault0$mNLoadEpi$value
=PC_Lake_SEL$Nitrogen
 lDATM_SETTINGS$forcings$sDefault0$mTempEpi$value =
c(vTEMP[1],rep(vTEMP,30))
 lDATM_SETTINGS$forcings$sDefault0$mTempHyp$value = 11.0
 #Det load is based on the ND ratio of PCLake and the nitrogen load
into the system, with a maximum value of the total modelled (BATT) TSS
load
 lDATM_SETTINGS$forcings$sDefault0$mDLoadDetEpi$value
=min(PC_Lake_SEL$TSS,PC_Lake_SEL$Nitrogen /
lDATM_SETTINGS$params$sDefault0[which(rownames(lDATM_SETTINGS$params)==
'cNDDetIn')])
 #IM load is based on the TSS load minus the Detrital load, with a
minimum of 0.
 lDATM_SETTINGS$forcings$sDefault0$mDLoadIMEpi$value
=max(0,PC_Lake_SEL$TSS-
lDATM_SETTINGS$forcings$sDefault0$mDLoadDetEpi$value)

 #set QIn

lDATM_SETTINGS$params$sDefault0[which(rownames(lDATM_SETTINGS$params)==
'cQInEpi')] = PC_Lake_SEL$Runoff

 }else if(sSCEN == "NBS_bioswale"){
 #adjust forcings
 lDATM_SETTINGS$forcings$sDefault0$mPLoadEpi$value =
PC_Lake_SEL$Phosphorous_gs
 lDATM_SETTINGS$forcings$sDefault0$mNLoadEpi$value
=PC_Lake_SEL$Nitrogen_gs
 lDATM_SETTINGS$forcings$sDefault0$mTempEpi$value =
c(vTEMP[1],rep(vTEMP,30))
 lDATM_SETTINGS$forcings$sDefault0$mTempHyp$value = 11.0
 #Det load is based on the ND ratio of PCLake and the nitrogen load
into the system, with a maximum value of the total modelled (BATT) TSS
load
 lDATM_SETTINGS$forcings$sDefault0$mDLoadDetEpi$value
=min(PC_Lake_SEL$TSS_gs,PC_Lake_SEL$Nitrogen_gs /
lDATM_SETTINGS$params$sDefault0[which(rownames(lDATM_SETTINGS$params)==
'cNDDetIn')])
 #IM load is based on the TSS load minus the Detrital load, with a
minimum of 0.
 lDATM_SETTINGS$forcings$sDefault0$mDLoadIMEpi$value
=max(0,PC_Lake_SEL$TSS_gs-
lDATM_SETTINGS$forcings$sDefault0$mDLoadDetEpi$value)

 #set QIn

lDATM_SETTINGS$params$sDefault0[which(rownames(lDATM_SETTINGS$params)==
'cQInEpi')] = PC_Lake_SEL$Runoff_gs
```

```
 }else if(sSCEN == "NBS_wadi"){
 #adjust forcings
 lDATM_SETTINGS$forcings$sDefault0$mPLoadEpi$value =
PC_Lake_SEL$Phosphorous_gw
 lDATM_SETTINGS$forcings$sDefault0$mNLoadEpi$value
=PC_Lake_SEL$Nitrogen_gw
 lDATM_SETTINGS$forcings$sDefault0$mTempEpi$value =
c(vTEMP[1],rep(vTEMP,30))
 lDATM_SETTINGS$forcings$sDefault0$mTempHyp$value = 11.0
 #Det load is based on the ND ratio of PCLake and the nitrogen load
into the system, with a maximum value of the total modelled (BATT) TSS
load
 lDATM_SETTINGS$forcings$sDefault0$mDLoadDetEpi$value
=min(PC_Lake_SEL$TSS_gw,PC_Lake_SEL$Nitrogen_gw /
lDATM_SETTINGS$params$sDefault0[which(rownames(lDATM_SETTINGS$params)==
'cNDDetIn')])
 #IM load is based on the TSS load minus the Detrital load, with a
minimum of 0.
 lDATM_SETTINGS$forcings$sDefault0$mDLoadIMEpi$value
=max(0,PC_Lake_SEL$TSS_gw-
lDATM_SETTINGS$forcings$sDefault0$mDLoadDetEpi$value)

 #set QIn

lDATM_SETTINGS$params$sDefault0[which(rownames(lDATM_SETTINGS$params)==
'cQInEpi')] = PC_Lake_SEL$Runoff_gw
 }
 ## 5. Initialize model
 ## - make all initial states according to the run settings
 InitStates <- PCModelInitializeModel(lDATM = lDATM_SETTINGS,
      dirSHELL = dirShell,
      nameWORKCASE = nameWorkCase)
 ## 6. run one model
 ## - Error catching on run_state & restart (if run_state = 0 & you use
restart should you be able to do so?)
 PCModel_run01 <- PCModelSingleRun(lDATM = lDATM_SETTINGS,
     nRUN_SET = 0,
     dfSTATES = InitStates,
     integrator_method = "euler",
     dirSHELL = dirShell,
     nameWORKCASE = nameWorkCase)

 ## - Error catching on run_state & restart (if run_state = 0 & you use
restart should you be able to do so?)
 PCModel_run02 <- PCModelSingleRun(lDATM = lDATM_SETTINGS,
     nRUN_SET = 1,
     dfSTATES = InitStates,
     integrator_method = "euler",
     dirSHELL = dirShell,
     nameWORKCASE = nameWorkCase)
 # Define output
 temp_res <- PCModel_run01

 temp_res$period <- "winter"
 temp_res[temp_res$time %in% summer_vec, "period"] <- "summer"

 last_year_res = temp_res[c((nrow(temp_res)-365):nrow(temp_res)),]
 dtOUT_SUM <- last_year_res %>%
```

```
 group_by(period) %>%
 summarise(across(where(is.numeric), ~ mean(.x, na.rm = TRUE)))
 dtOUT_SUM$initstate="turbid"
 dtOUT_SUM$scen=sSCEN
 dtOUT_SUM$LakeID = PC_Lake_SEL$LakeID
 dtOUT_SUM$bioswale_area = Green_area$Bioswale_total_area_m2
 dtOUT_SUM$wadi_area = Green_area$Wadi_total_area_m2
 dtOUT_SUM$Bioswale_storage = Green_area$Bioswale_storage_L_m2
 dtOUT_SUM$wadi_storage = Green_area$Wadi_storage_L_m2

 if (exists("dtOUT_AGG")==TRUE){
 dtOUT_AGG = rbind(dtOUT_AGG,dtOUT_SUM)
 }else{
 dtOUT_AGG = dtOUT_SUM
 }

 # Define output
 temp_res <- PCModel_run02

 temp_res$period <- "winter"
 temp_res[temp_res$time %in% summer_vec, "period"] <- "summer"

 last_year_res2 = temp_res[c((nrow(temp_res)-365):nrow(temp_res)),]
 dtOUT_SUM2 <- last_year_res2 %>%
 group_by(period) %>%
 summarise(across(where(is.numeric), ~ mean(.x, na.rm = TRUE)))
 dtOUT_SUM2$initstate="clear"
 dtOUT_SUM2$scen=sSCEN
 dtOUT_SUM2$LakeID = PC_Lake_SEL$LakeID
 dtOUT_SUM2$bioswale_area = Green_area$Bioswale_total_area_m2
 dtOUT_SUM2$wadi_area = Green_area$Wadi_total_area_m2
 dtOUT_SUM2$Bioswale_storage = Green_area$Bioswale_storage_L_m2
 dtOUT_SUM2$wadi_storage = Green_area$Wadi_storage_L_m2

 dtOUT_AGG = rbind(dtOUT_SUM,dtOUT_SUM2)


 data.table::fwrite(cbind.data.frame(last_year_res,last_year_res2),
file = file.path(dirShell, "work_cases", nameWorkCase,
"output","single_runs", paste0(PC_Lake_SEL$LakeID,"_", sSCEN,".txt")))

 return(dtOUT_AGG)
 }
}
stopCluster(snowCLUSTER)

fwrite(lOUT, file.path(dirShell, "work_cases", nameWorkCase, "output",
paste0("NICHES_AVG_all_runs",".csv")))


lOUT = fread(file.path(dirShell, "work_cases", nameWorkCase, "output",
paste0("NICHES_AVG_all_runs",".csv")))

##PLOTTING AND ANALYSIS BETWEEN SCENARIOS##----
dfPLOT = as.data.frame(lOUT[which(lOUT$period =="summer"),])
vVARS_PLOT <- c("oChlaEpi","aDVeg", "oPO4WEpi", "oO2WEpi",
"oChlaBlueEpi", "rPSeq", "rNSeq", "aESSwimming","aESBird","aESFish" )
#"aESIrrigation","aESThatching" -> werken niet
```

```
#cycle through the different variables
for(sVAR in vVARS_PLOT){
 for(sINIT_STATE in c("clear","turbid")){

 #sVAR="aDVeg"#debug
 # sVAR="EKR_MFT"#
 #declare axis label
 #sYLABEL = vYLABELS[which(sVAR == unique(dfPLOT$variable))]


 #do a permutation based (Fischer's exact test) multiple comparison
test with false discovery rate correction
 dtPERM_TEST_OUT=as.data.table(data.frame(matrix(NA,0,3)))
 colnames(dtPERM_TEST_OUT)=c("scen","total_count","sig_let")

 dfPLOT_SEL=dfPLOT[which(dfPLOT$initstate==sINIT_STATE),]
 dfPLOT_SEL=dfPLOT_SEL[,c("scen",sVAR)]

 colnames(dfPLOT_SEL)=c('scen', 'values')

 dfPLOT_SEL$scen=as.character(dfPLOT_SEL$scen)
 dfPERM_TEST_MULT = rcompanion::pairwisePermutationTest(value ~ scen,
        data = dfPLOT_SEL,
        method="BH")
 vPERM_TEST_ADJP = dfPERM_TEST_MULT$p.adjust
 #remove any NaNs resulting from insufficient sample size with 1
 vPERM_TEST_ADJP[is.na(vPERM_TEST_ADJP)]=1.0

 names(vPERM_TEST_ADJP)= str_replace(dfPERM_TEST_MULT$Comparison, " =
0", "")
 names(vPERM_TEST_ADJP)= str_replace(names(vPERM_TEST_ADJP), " - ", "-
")
 vPERM_TEST_LET = multcompLetters(vPERM_TEST_ADJP)

 #compute counts per EST
 dtCOUNT_N <- dfPLOT_SEL %>% group_by(scen) %>%
 summarise(total_count=n(), .groups = 'drop')
 #add significance letters to the data table
 dtCOUNT_N$sig_let = vPERM_TEST_LET$Letters
 #remove all significances with less then 4 samples as they are
spurious:
 dtCOUNT_N$sig_let[dtCOUNT_N$total_count<=3]=""
 dtPERM_TEST_OUT=rbind(dtPERM_TEST_OUT,dtCOUNT_N)



 dfPLOT_SEL$scen = factor(dfPLOT_SEL$scen, levels = c("BAU",
"NBS_bioswale", "NBS_wadi"))


 #make the plot
 pdf(file.path(file.path(dirShell, "work_cases", nameWorkCase,
"output","plots"),paste("PCLAKENBATT_RDAM_",sVAR,"_",sINIT_STATE,".pdf"
,sep="")),width=8, height=5, pointsize=14, useDingbats=FALSE)
 print(
 ggplot(data=dfPLOT_SEL, aes(x=scen, y=values))+
 geom_violin(fill="lightgrey", color=NA)+
```

```
 geom_text(data = dtPERM_TEST_OUT, aes(label=paste0("n=",total_count,
"\n", sig_let),x=scen,
y=max(dfPLOT_SEL$values,na.rm=TRUE)+max(dfPLOT_SEL$values,na.rm=TRUE)*0
.1), position=position_dodge2(0.75), vjust=1.0, size=3) +
 geom_hline(linetype="dotted", color='black', aes(yintercept=0.0))+

scale_x_discrete(labels=str_replace_all(as.character(levels(dfPLOT_SEL$
scen)),"_","\n"))+
 #ylim(0,2.5)+
 #geom_dotplot(binaxis='y', stackdir='center', dotsize=0.1)+
 #geom_jitter(shape=16, position=position_jitter(0.2))+
 stat_summary(fun.data = median_hilow, fun.args=list(conf.int = .75),
geom = "pointrange")+ #using a method that shows median and 75%
quantiles (i.e. acts like a boxpot in terms of info)
 ylab(sVAR)+
 xlab("Scenario")+
 theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
 theme_bw()
 )
 dev.off()
 }
}
# else{ # else{ TRUE
# print("This data frame is empty")
# }
#} #Sluiting if statement 2######
#} #sluiting if statement 1#####
#}

# #Write PCLake output to csv file
# write.csv(PC_Lake, "PC_Lake_output.csv", row.names = FALSE)
#
# #Write lake results (oversize watersheds) to csv file
# lake_results_df <- do.call(rbind, lake_results_list)
# write.csv(lake_results_df, "lake_results_summary.csv", row.names =
FALSE)







#2: Create buffer around lake that captures the watershed area----


#3: Cut polygon of buffered lake by polygon of basin and check surface
area of cut buffered lake polygon---

#IF surface < watershed: repeat 2 and 3 with larger buffer (+10%)

#IF surface > watershed: repeat 2 and 3 with larger buffer (-10%)

#IF surface = watershed OR within 0.1% of watershed area: save polygon
as lake watershed
```

```
#4: extract land cover (CORINE) in watershed buffer

#5: Run BATT without NBS

#6: Run BATT with NBS

#7: Run Temperature model (Mooij et al)

#8: Run PCLake+ four times: clear-no_NBS, clear-NBS, turbid-no_NBS,
turbid-NBS
```