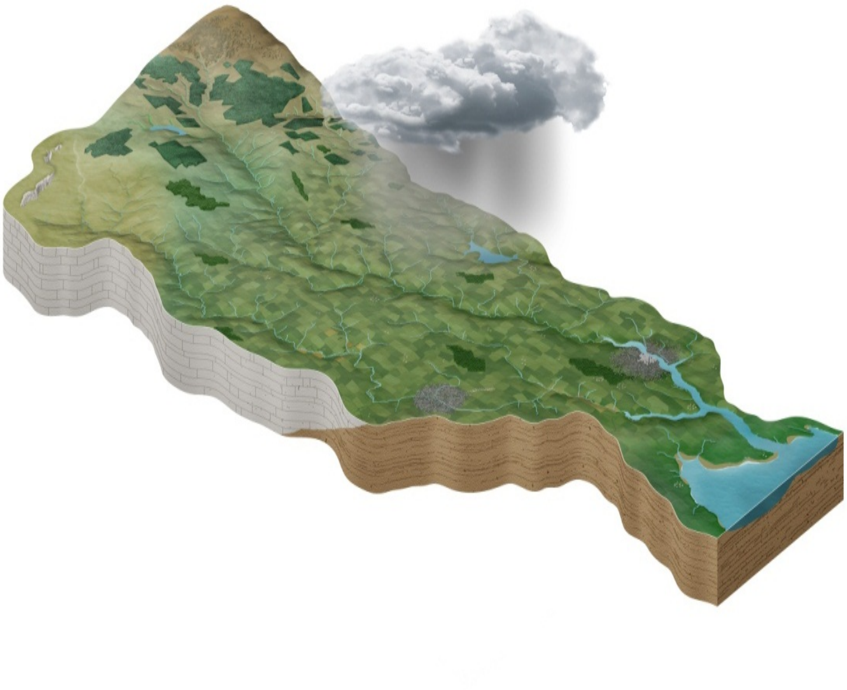
The Nutrient Source Loading Apportionment Model (SLAM) Framework for Ireland

Technical Documentation



Eva M Mockler, Thibault Hallouin

June 2020

Version no. 3.0

Version Number	Authors	Changes	Date
V1	Eva M Mockler	Model v1 document	2016
V2	Eva M Mockler	Updated for Model v2.06	March 2017
V3	Thibault Hallouin	Updated for Model v3.0.0dev	June 2020

epaCatchments
Healthy • Resilient • Productive
Valued water resources supporting vibrant communities

Table of Contents

1 Introduction

Excessive nitrogen and phosphorus in water and the air can cause health problems, damage the environment and impact the economy. Source apportionment is used to estimate the nutrient load from various sectors entering waterbodies, following attenuation or treatment. This document outlines the technical details of the **Source Loading Apportionment Model (SLAM) Framework**. This model estimates the **phosphorus (P) and nitrogen (N) loads** that reach a river from all major sources in a sub-catchment e.g. agriculture, forestry and industry.

The objective of the SLAM toolbox is to support catchment managers by providing scientifically robust evidence base to back-up decision making in relation to reducing nutrient pollution. This model was developed by the EPA CatchmentTools Project (2014-2017), and builds on previous research from the EPA Pathways Project which produced both numerical (Mockler et al., 2014; Mockler et al., 2016) and GIS-based water quality models for Irish catchments (Archbold et al., 2016).

The general method of source-orientated load apportionment models is:

1. Calculate initial nutrient loads from sector.
2. Reduced load by a factor, where required, to account for treatment (e.g. urban wastewater) or attenuation in the environment (e.g. diffuse agricultural sources).
3. Compare predicted annual in-stream load with annual load calculated from measurements, where available.

As detailed risk assessments are not feasible at national scale, simple tools can be used to prioritise resources for catchment management. The average annual model results produced by the SLAM are used in combination with monitoring data and local knowledge to highlight the likely sources of issues within a sub-catchment (Daly et al., 2016). Additional detailed assessments may be required in some cases where, for example, seasonality or local conditions have a potential ecological impact.

1.1 Model Software Selection

An initial scoping exercise in 2014 identified that the load apportionment model proposed by the Irish EPA was required to;

(i) produce national results using available datasets,
(ii) include a user-friendly interface,
(iii) facilitate continuous development, and
(iv) be available to the EPA after only one year of development.

After testing several options, the optimal software solution was identified as **ArcGIS toolbox** developed using ModelBuilder. ArcGIS is the most common GIS software used by the relevant stakeholders, and provides a user-friendly interface with limited development. Using this software, sample results were easily produced which were used to elicit feedback from stakeholders and hence to ensure that all of the requirements of the stakeholders were met. The first and second versions of the SLAM were developed using ModelBuilder, while the third version of the SLAM was developed using Python scripts to create a Python toolbox. Any version comes as an ArcGIS toolbox.

The third version was developed using Python scripts in order to facilitate future uses of the SLAM for scenario analyses directly from a Python script, instead of using dialog boxes. However, this does not compromise on the use of the SLAM as in the first two versions, and the SLAM version 3 can still be fully operated from ArcMap and dialog boxes. A 'python toolbox' was preferred over a 'custom toolbox' (i.e. that can create script tools) because it does not require the use of any dialog box to set up new tools or update existing tools in the toolbox, everything is defined and changed in the python code. This improves the maintainability and the transparency of the toolbox, while also allowing for the geoprocessing section of each tool to be callable from within Python.

1.2 Model Development

The SLAM toolbox was designed with individual sub-models for each sector to facilitate engagement with relevant stakeholders and scientific experts. For example, while the septic tank system sub-model was in development, several alternative options were tested and compared within the toolbox. The alternative results that these models produced were presented to stakeholders, to highlight the impacts of model improvements on the overall load apportionment results.

Further development is planned for the forestry, peatlands, diffuse urban and lake retention sub-models to incorporate recent research findings and higher resolution datasets.

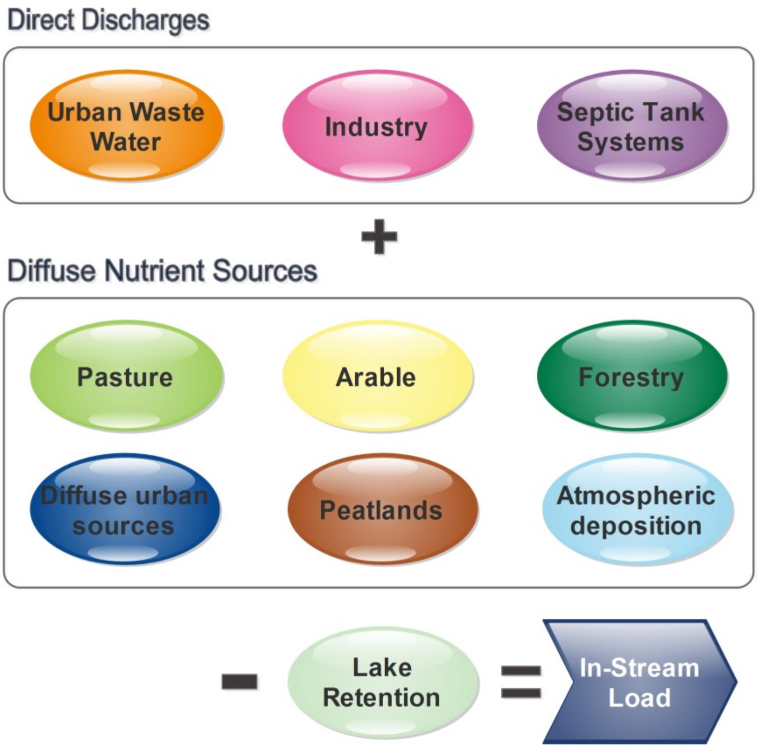


Figure 1. Sub-models of the Source Load Apportionment Model (SLAM) Framework.

2 Model Overview

The SLAM has been developed as an ArcGIS toolbox that uses available Irish data and models to quantify annual nutrient losses from both point discharges from urban wastewater, industry and septic tank systems, and diffuse sources including agriculture, forestry, etc. (Figure 1). The total annual nutrient load at the outlet of each sub-catchment (L_j) is calculated as;

$$L_j = \left(Point_j + Diffuse_j\right) * (1 - Lake_j) \qquad (1)$$

where,

$Point_j$= sum of nutrient loads discharged from point sources

$Diffuse_j$= sum of diffuse nutrient losses

$Lake_j$= estimated nutrient lake retention factor

An outline of the calculation methods for each sector and primary data sources are given in Table 1. Alternative data can replace the primary data sources, once the field mapping in the toolbox is updated. This facilitates the updating of pressure datasets e.g. with new emission values from annual environmental reports, and also enables modelling of historical data. For example, CORINE datasets from 2000 and 2006 can be combined with historical census data to model the sources of long term nutrient monitoring data. Ní Longphuirt et al. (2016) used the SLAM toolbox in this manner to evaluate the changes in nutrient sources over a ten-year period in 17 catchments in Ireland.

2.1 Direct Discharges

For direct discharges, the main data source is from monitoring of nutrients in effluent discharged by individual municipal or industrial facilities. Annual environmental performance reporting provides these estimates of annual nutrient loads which are spatially associated with the point of discharge. Where monitoring data is not available, loads are estimated from the population equivalent (PE) of the facility. For the septic tank systems sub-model, results of the SANICOSE model (Gill and Mockler, submitted) are aggregated up to the relevant water body or sub-catchment scale.

2.2 Diffuse Nutrient Sources

Ireland is predominantly covered by agricultural land, with significant areas of forestry and wetlands (Table 2). The 2012 CORINE land cover data (Lydon and Smith, 2014) is used in the SLAM toolbox to estimate nutrient emissions to water from areas of forestry, peatlands and artificial surfaces.

The pressure information for diffuse agriculture is mainly from the Land-Parcel Identification System (LPIS) provided by the Department of Agriculture, Food and the Marine (DAFM). The pasture and arable sub-models aggregate the relevant results from the Catchment Characterisation Tool (CCT) (Archbold et al., 2016 and reference therein). Both the SANICOSE (for septic tank systems) and the CCT models produce simple estimates of nutrient loads in groundwater, which the SLAM toolbox combines to give an indication of the percentage of nutrients that are delivered to surface waters via groundwater pathways. See Mockler et al. (submitted) for further details on the calculation methods for each sub-model.

Table 2. CORINE 2012 land cover by area and percentage area of Ireland.

CORINE 2012	*Area (km^2)*	*%*
Artificial Surfaces	1,747	2.5
Agricultural Areas	47,879	68.1
Forestry and semi-natural areas	8,073	11.5
Wetlands	11,069	15.8
Water	1,502	2.1
Total	70,271	100

Table 1. Data sources and calculation methods for sub-models of the SLAM framework.

Sub-model	Calculation Methods	Primary Data Sources
Waste water discharges	Annual nutrient load emissions reported to EPA for 2014 OR calculation based on 2014 PE and assumed figures for nutrient production and treatment efficiency.	2014 Annual Environmental performance Reports OR 2014 Population Equivalent (PE)
Diffuse urban sources*	Export coefficient model based on land cover	2012 CORINE land cover
Industrial discharges	Three year average of annual reported nutrient emissions OR calculation based on 25% of licenced limits	2011-2013 PRTR database OR EPA licenced limits
Septic Tank Systems	Export coefficient model (SANICOSE) with 3 transport pathways: 1. Inadequate percolation 2. Near surface (subsoils) 3. Groundwater	Non-sewered house dataset, surface water bodies (EPA, OSi) and catchment characteristics including aquifer vulnerability and subsoil permeability (from Geological Survey of Ireland)
Pasture **(diffuse agriculture)**	Export coefficient model (CCT) with 2 transport pathways: 1. Near surface (soils/ subsoils) 2. Groundwater (sub surface pathways)	2012 LPIS (DAFM), Good Agricultural Practices (GAP) Regulations (S.I. 31, 2014) and catchment characteristics including soil drainage properties and depth to bedrock.
Arable **(diffuse agriculture)**	Export coefficient model (CCT) with 2 transport pathways (same as Pasture sub-model).	2012 LPIS (DAFM), fertilizer application rates (Lalor et al., 2010) and catchment characteristics
Forestry*	Export coefficient model based on land cover	2012 CORINE land cover
Peatlands*	Export coefficient model based on land cover	2012 CORINE land cover
Atmospheric deposition	Direct deposition on lakes	Nitrogen deposition from map from Henry and Aherne (2014) & Lakes map (EPA)

* Sub-model structure under review.

3 SLAM Toolbox Interface

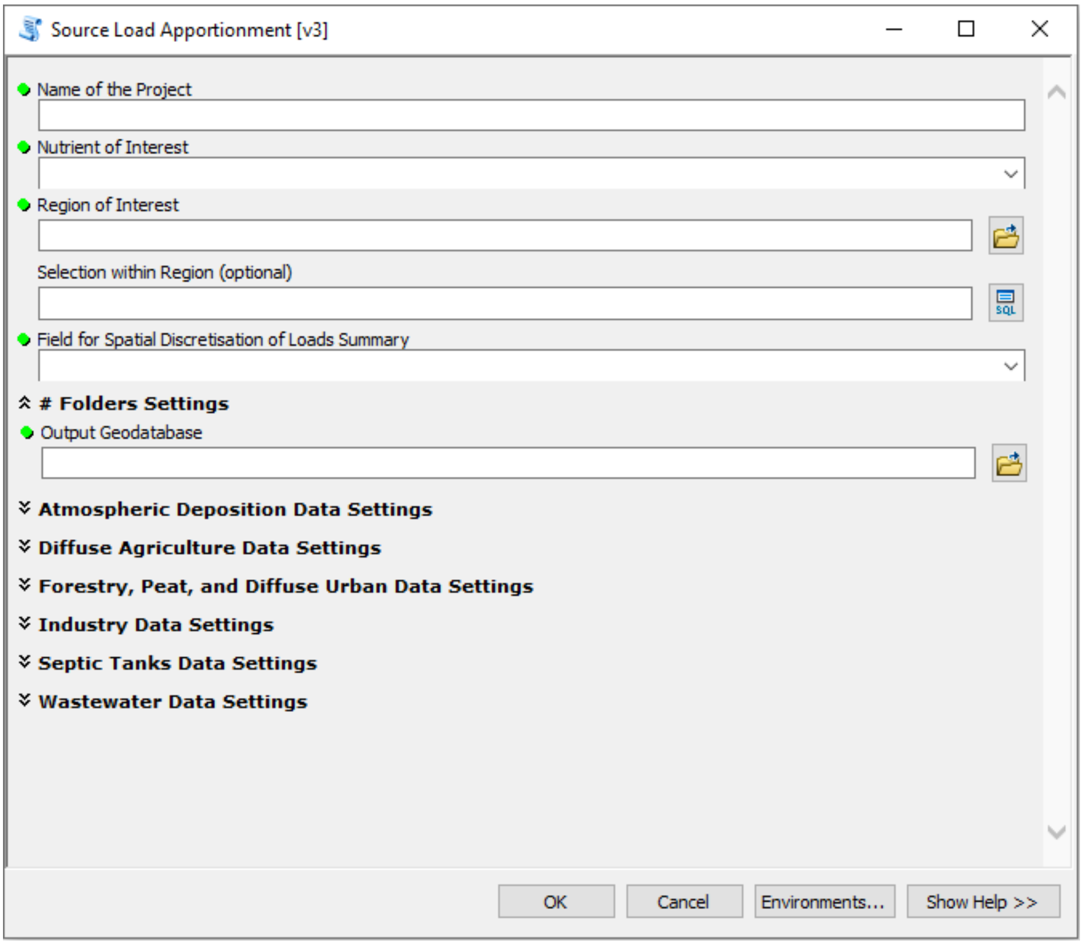
The SLAM toolbox interface is opened through the ArcToolbox panel in ArcGIS (Figure 2). Firstly, the user is required to choose a name for their modelling project. Then, the user is required to indicate whether nitrogen (N) or phosphorus (P) is to be modelled. Users can select any polygon feature class (or shapefile) in the Field 'Region of Interest' to represent the catchments, and results will be reported using the 'Field for Spatial Discretisation' in the attribute table. In addition, a SQL expression can be provided in the Field 'Selection within Region' to indicate a sub-set of polygons in the catchment shapefile.

Users can then select the preferred datasets for the sub-models in the corresponding expandable window shades 'Data Settings', including;

- the CORINE land cover data,
- the SANICOSE and CCT results,
- wastewater treatment plants shapefile,
- industrial discharges shapefiles, and
- parameter tables.

The parameter tables are provided in Excel spreadsheets, and can be easily edited and updated by users.

The SLAM results are provided in feature tables and feature classes, and are saved to the geodatabase indicated by the 'Output Geodatabase' in the expandable window shade '# Folders Settings', using predefined names including a customisable prefix specified in the 'Name of the Project' field.



Source Load Apportionment [v3]

Name of the Project

Nutrient of Interest

Region of Interest

Selection within Region (optional)

Field for Spatial Discretisation of Loads Summary

# Folders Settings

Output Geodatabase

Atmospheric Deposition Data Settings

Diffuse Agriculture Data Settings

Forestry, Peat, and Diffuse Urban Data Settings

Industry Data Settings

Septic Tanks Data Settings

Wastewater Data Settings

Figure 2. Input screen for SLAM ArcGIS toolbox.

Results are displayed spatially using standard ArcGIS functions to produce pie charts of sources of nutrients (Figure 3). In addition, tabular results are easily exported into other software.

4 SLAM Toolbox Structure

4.1 Python Toolbox Overview

The ‘Load Apportionment’ version 3 is the main tool of the SLAM Framework (toolbox) that links together all the sector sub-models tools. This main tool also specifies the input and output file names and locations, as well as the sub-model data inputs. Summary Statistics tools are used to aggregate the values calculated by the sub-models for each of the specified catchment polygons.

List of General Parameters:

Name	Description	Type	More information
Name of the Project	Name that will be used to identify the outputs (used as their prefix)	String	-
Nutrient of Interest	Choice for the nutrient whose sources will be calculated	String	either “Nitrogen (N)” or “Phosphorus(P)”
Region of Interest	Feature class containing polygons used to delineate the study region (and optional sub-regions)	Feature Class	-
Selection within Region (optional)	Query to only work on a sub-region included in the Region of Interest	SQL Query	-
Field for Spatial Discretisation of Loads Summary	Name of the field in the feature class for the Region of Interest to be used as the spatial level at which to summarise the source loads	Field	-
Output Geodatabase	Geodatabase where all output feature classes will be stored	Geodatabase	default: SLAMpy\out\output.gdb

In addition to these general parameters, the parameters specific to each sub-model presented in the following sub-sections are also required in the ‘Data Option’ expandable window shades. In turn, these general parameter are also required if the sub-models are used as stand-alone tools (except for the ‘Field for Spatial Discretisation’ that is not required because summary statistics are only done in the Load Apportionment model).

Note: All input datasets used must be in the same geographic coordinate system to be correctly related to the selected region of interest.

If the Load Apportionment model needs to be re-run only because a few sub-models are to be changed, it is possible for the user to reuse existing output for the tools that do not need to be re-run. This can be done in each ‘Data Option’ expandable window shade for the relevant sub-models.

All of the sub-models are described in the subsequent sections.

In the description of each sub-model, %NorP% is often used to specify that it is either N if Nitrogen is selected as nutrient of interest, or P if it is Phosphorus.

Moreover, when factors from external spreadsheets are required, the name of the factors extracted from the spreadsheet are specified in curly braces in the Python expressions, e.g. {FactorName} to refer to a factor named FactorName in the spreadsheet.

Output name(s): “Name of the Project”_%NorP%_Loads_Summary

4.2 Diffuse Agriculture

4.2.1 Diffuse Agriculture Version 1

The inputs for the diffuse agriculture tool is the CSO Agricultural Census containing the area of the different types of crops for arable loads and the area of pasture and the herds of animals for pasture. These figures are combined with estimated nutrient loads per hectare/per animal to determine the nutrient export loads.

List of Specific Parameters:

Name	Description	Type	More information
CSO Census Electoral Divisions Data [1]	Feature class containing the agricultural census data at the electoral division level	Feature Class	default: SLAMpy\in\input.gdb\T2010_ED_Agri4 *
Crop Factors for Nitrogen (N) [2]	Spreadsheet containing the export factors from arable land for N	Table	default: SLAMpy\in\LAM_Factors.xlsx\Crop_N$
Crop Factors for Phosphorus (P) [2]	Spreadsheet containing the export factors from arable land for P	Table	default: SLAMpy\in\LAM_Factors.xlsx\Crop_P$
Livestock Factors for Nitrogen (N) [3]	Spreadsheet containing the export factors from pastureland for N	Table	default: SLAMpy\in\LAM_Factors.xlsx\Livestock_N$
Livestock Factors for Phosphorus (P) [3]	Spreadsheet containing the export factors from pastureland for P	Table	default: SLAMpy\in\LAM_Factors.xlsx\Livestock_P$
[1] **must contain fields:** total_cere, other_crop, potatoes, bulls, dairy_cows, suckler_co, cattle_m_1, cattle_f_1, cattle_m_2, cattle_f_2, cattle_m_3, cattle_f_3, cattle_m_4, cattle_f_4, dairyheife, otherheife, total_shee, horses, Hay, Pasture, Silage [2] **must contain fields:** FactorName, WinterWheat, SpringWheat, WinterBarley, SpringBarley, WinterOats, SpringOats, CerealOther, Potatoes, SugarBeet, OtherCrops, Pasture, ExportFactor [3] **must contain fields:** bulls, dairy_cows, other_cattle, total_cattle, cattle_m_4, cattle_m_3, cattle_m_2, cattle_m_1, total_sheep, horses, ExportFactor * can also be readily used with T1990_ED_Agri4 and T2000_ED_Agri4			

The arable values are calculated as follows:

Output name(s): "Name of the Project"_%NorP%_Arable

Field(s) added in output: Area_ha2, Arab_calc, Arab1calc

Area_ha2 field is calculated as:

- Expression: !shape.area@hectares!
- Expression Type: PYTHON_9.3

Arab_calc field is calculated as:

- Expression: (!total_cere! * {CerealOther} + !other_crop! * {OtherCrops} + !potatoes! * {Potatoes}) * {ExportFactor} / !Area_ha!
- Expression Type: PYTHON_9.3

Arab1calc field is calculated as:

- Expression: !Arab_calc! * !Area_ha2!
- Expression Type: PYTHON_9.3

Note: The Agricultural Census data is available for several reporting periods (e.g. reported in 1991, 2000, 2010), but the oldest reports do not provide details on the type of cereals (e.g. winter/spring barley, winter/spring wheat, etc.). This is why the detailed factors available for these cereals in the factors spreadsheet are not used, and they are all regrouped under total cereals using the factor for other cereals.

The pasture values are calculated as follows:

Output name(s): “Name of the Project”_%NorP%_Pasture

Field(s) added in output: Area_ha2, Past_calc, Past1calc

Area_ha2 field is calculated as:

- Expression: !shape.area@hectares!
- Expression Type: PYTHON_9.3

Past_calc field is calculated as:

- Expression: {ExportFactor} * (!bulls! * {bulls} + !dairy_cows! * {dairy_cows} + !suckler_co! * {other_cattle} + (!cattle_m_1! + !cattle_f_1!) * {cattle_m_1} + (!cattle_m_2! + !cattle_f_2!) * {cattle_m_2} + (!cattle_m_3! + !cattle_f_3! + !cattle_m_4! + !cattle_f_4! + !dairyheife! + !otherheife!) * {cattle_m_3} + !total_shee! * {total_sheep} + !horses! * {horses} + (!Hay! + !Pasture! + !Silage!)* {Pasture}) / !Area_ha!
- Expression Type: PYTHON_9.3

Past1calc field is calculated as:

- Expression: !Past_calc! * !Area_ha2!
- Expression Type: PYTHON_9.3

4.2.2 Diffuse Agriculture Version 2

The input shapefiles to the Diffuse Agriculture model are output from the Catchment Characterisation Tool (CCT), separated in to arable and pasture. Hence, this module identifies the N or P model output from the CCT output for both groundwater (!%NorP%SwFromGw!) and total emissions (!%NorP%TotaltoSW!) for the required catchment polygons.

List of Parameters:

Name	Description	Type	More information
PIP Maps Data for Arable [1]	Feature class containing the N and P export loads to surface waters for arable (based on LPIS or Census 2010 data) – specially pre-processed for SLAM (see section 5.1)	Feature Class	default: SLAMpy\in\input.gdb\PathwaysCCT_IRL_Arable_LPIS
PIP Maps Data for Pasture [1]	Feature class containing the N and P export loads to surface waters for pasture (based on LPIS or Census 2010 data) – specially pre-processed for SLAM (see section 5.1)	Feature Class	default: SLAMpy\in\input.gdb\PathwaysCCT_IRL_Pasture_LPIS
[1] **must contain fields:** nSwFromGw, pSwFromGw, nTotaltoSWreceptor, pTotaltoSWreceptor			

The arable values are calculated as follows:

Output name(s): “Name of the Project”_%NorP%_Arable

Field(s) added in output: Area_ha, GWArab2calc, Arab2calc

Area_ha field is the annual value for arable contributed via groundwater, calculate as:

- Expression: !shape.area@hectares!
- Expression Type: PYTHON_9.3

GWArab2calc field is calculated as:

- Expression: !%norp%SwFromGw! * !Area_ha!
- Expression Type: PYTHON_9.3

Arab2calc field is the annual value for arable contributed to surface waters, calculate as:

- Expression: !%norp%TotaltoSWreceptor! * !Area_ha!
- Expression Type: PYTHON_9.3

The pasture values are calculated as follows:

Output name(s): "Name of the Project"_%NorP%_Pasture

Field(s) added in output: Area_ha, GWPast2calc, Past2calc
Area_ha field is calculated as:

- Expression: !shape.area@hectares!
- Expression Type: PYTHON_9.3

GWPast2calc field is the annual value for pasture contributed via groundwater, calculate as:

- Expression: !%norp%SwFromGw! * !Area_ha!
- Expression Type: PYTHON_9.3

Past2calc field is the annual value for pasture contributed to surface waters, calculate as:

- Expression: !%norp%TotaltoSWreceptor! * !Area_ha!
- Expression Type: PYTHON_9.3

Note: GWArab2cal and GWPast2calc are already included in Arab2calc and Past2calc, respectively. This is why the post-processing tool is only using Arab2calc and Past2calc for the total loads for arable and pasture, respectively.

4.3 Wastewater Discharges

The wastewater discharges (or agglomeration) module calculates the emissions from wastewater treatment plants (WWTPs) and Storm Water Overflows (SWOs, aka combined sewer overflow; CSO) using information reported in the annual environmental reports (AERs) where available, and otherwise make estimates using the best available information on the population equivalents (PE), influent concentrations, and flow rates. The first version of the tool is using the location of the WWTP whereas the second version of the tool is using the location of the discharge point to associate effluents to a given polygon. Moreover, the second version of the tool used input data more pre-processed than the first version so that this results in a simpler algorithm for the tool.

4.3.1 Wastewater Discharges Version 1

List of Parameters:

Name	Description	Type	More information
Agglomerations Discharges Data [1]	Feature class containing the WWTP information, including the location of the treatment plants as points, the treatment levels, the population equivalent, and possibly AER information – specially pre-processed for SLAM (see section 5.2.4)	Feature Class	default: SLAMpy\in\input.gdb\LAM_Agglom_Nov15
WWTP Factors for Nitrogen (N) [2]	Spreadsheet containing the treatment efficiency factors for the different levels of waste water treatment for N	Table	default: SLAMpy\in\LAM_Factors.xlsx\UWWTP_N$
WWTP Factors for Phosphorus (P) [2]	Spreadsheet containing the treatment efficiency factors for the different levels of waste water treatment for P	Table	default: SLAMpy\in\LAM_Factors.xlsx\UWWTP_P$

[1] **must contain fields:** AER14_PE, LEMA_PE, TreatmentL, N_WWTP_AER, P_WWTP_AER, N_SWO_AER, P_SWO_AER, PE, LOSS_perce
[2] **must contain fields:** raw, prelim, primary, second, tertN, tertNP, tertP, POPfactor

Output name(s): “Name of the Project”_%NorP%_Wastewater

Field(s) added in output: PE_calc, Treat_Fact, PEqWast1calc, PEqSWOWast1calc, AERWast1calc, AERSWOWast1calc

PE_calc field is the annual population equivalent (PE) for the WWTP discharge, calculate as:

- Expression: factor(float(!AER14_PE!), float(!LEMA_PE!))
- Expression Type: PYTHON_9.3
- Code Block:

```
def factor(aer14, lema):
        if aer14 > 1:
           return aer14
        elif code == '112':
           return lema
```

Treat_Fact field is the treatment factor for the WWTP discharge, calculate as:

- Expression: factor(!TreatmentL!)
- Expression Type: PYTHON_9.3
- Code Block:

```
def factor(treatment):
        if treatment == '0 - No Treatment':
           return {raw}
        elif treatment == '0 - Preliminary Treatment':
           return {prelim}
        elif treatment == '1 - Primary Treatment':
           return {primary}
        elif treatment == '2 - Secondary Treatment':
           return {second}
        elif treatment == '3N - Tertiary N Removal':
           return {tertN}
        elif treatment == '3NP - Tertiary N&P Removal':
           return {tertNP}
        elif treatment == '3P - Tertiary P Removal':
           return {tertP}
        elif treatment == 'Secondary':
           return {second}
        else:
           return {primary}
```

PEqWast1calc field is the annual load for the WWTP discharge, calculate as:

- Expression: value(float(!%NorP%_WWTP_AER!), float(!PE_calc!), float(!Treat_Fact!))
- Expression Type: PYTHON_9.3
- Code Block:

```
def value(wwtp_aer, pe_calc, treat_fact):
        if wwtp_aer > 1:
           return 0
        else:
           return pe_calc * treat_fact *({POPfactor} * 365 / 1000)
```

PEqSWOWast1calc field is the annual Storm Water Overflow (SWO) load for the WWTP agglomeration, calculated as:

- Expression: value(float(!%NorP%_WWTP_AER!), float(!PE!), float(!LOSS_perce!))
- Expression Type: PYTHON_9.3

- Code Block:

```
def value(wwtp_aer, pe, loss_perce):
    if swo_aer < 0.1:
        return ((pe * ({POPfactor} * 365 / 1000)) / 1 - loss_perce) * loss_perce
    else:
        return 0
```

AERWast1calc field is the annual load for the WWTP discharge as specified in the annual environmental report (AER), calculated as:

- Expression: !%NorP%_WWTP_AER!
- Expression Type: PYTHON_9.3

AERSWOWast1calc field is the annual Storm Water Overflow (SWO) load for the WWTP agglomeration as specified in the annual environmental report (AER), calculated as:

- Expression: !%NorP%_SWO_AER!
- Expression Type: PYTHON_9.3

4.3.2 Wastewater Discharges Version 2

List of Parameters:

Name	Description	Type	More information
Agglomerations Discharges Data [1]	Feature class containing the WWTP information, including the location of the treatment plant outlets as points, the treatment levels, the population equivalent, and possibly AER information – specially pre-processed for SLAM (see section 5.2.3)	Feature Class	default: SLAMpy\in\input.gdb\SLAM_Agglom15_March17_IsMain
[1] **must contain fields:** TN_SWO, TP_SWO, PointTN, PointTP			

Output name(s): "Name of the Project"_%NorP%_Wastewater

Field(s) added in output: SWOWast2calc, Wast2calc

SWOWast2calc field is the load for the agglomeration discharge location, calculated as:

- Expression: !T%NorP%_SWO!
- Expression Type: PYTHON_9.3

Wast2calc is calculated as:

- Expression: !PointT%NorP%!
- Expression Type: PYTHON_9.3

4.3.3 Wastewater Discharges Version 3

List of Parameters:

Name	Description	Type	More information
Agglomerations Discharges Data [1]	Feature class containing the WWTP information, including the location of the treatment plant outlets as points – specially pre-processed for SLAM (see section 5.2.2)	Feature Class	default: SLAMpy\in\input.gdb\UWW_EmissionPointData_2016
[1] **must contain fields:** TN2016Kgyr, TP2016Kgyr			

Output name(s): "Name of the Project"_%NorP%_Wastewater

Field(s) added in output: SWOWast3calc, Wast3calc

Wast3calc is calculated as:

- Expression: !T%NorP%2016Kgyr!
- Expression Type: PYTHON_9.3

4.4 Industrial Discharges

4.4.1 Industrial Discharges Version 1

Not implemented anymore.

4.4.2 Industrial Discharges Version 2

The two datasets of industrial discharges are required by this module relate to (1) section 4 licences, and (2) IPC licences, in addition to the catchment polygons.

List of Parameters:

Name	Description	Type	More information
IPC Licences Data [1]	Feature class containing the data for the IPC (Integrated Pollution Control) licensed industries – specially pre-processed for SLAM (see section 5.2.6)	Feature Class	default: SLAMpy\in\input.gdb\IPPC_Loads_LAM2
Section 4 Licences Data [2]	Feature class containing the data for the Section licensed industries – specially pre-processed for SLAM (see section 5.2.5)	Feature Class	default: SLAMpy\in\input.gdb\Section4Discharges_D07_IsMain
[1] **must contain fields:** N_2012_LAM, P_2012_LAM [2] **must contain fields:** Flow__m3_d, Discharge_, TON_ELV, TN_ELV, NO3_ELV, NH3_ELV, NH4_ELV, NO2_ELV, TP_ELV, PO4_ELV			

For Section 4 Licences:
Calculate values from Section 4 Industrial discharges data using flows and max licence values.

Output name(s): “Name of the Project”_%NorP%_IndustrySect4

Field(s) added in output: Sect4_Flow, Sect4_ELV, S4Ind2calc

Sect4_Flow field is the annual flow value for the discharge, calculate as:

- Expression: flow(float(!Flow__m3_d!), float(!Discharge_!))
- Expression Type: PYTHON_9.3
- Code Block:

```
def factor(code, area):
    if flow_m3_d > 0:
        return flow_m3_d
    else:
        return discharge
```

Sect4_ELV field value is the highest specified licence limit for N or P species, calculated as:

- Expression: elv(float(!TON_ELV!), float(!TN_ELV!), float(!NO3_ELV!), float(!NH3_ELV!), float(!NH4_ELV!), float(!NO2_ELV!), float(!TP_ELV!), float(!PO4_ELV!))
- Expression Type: PYTHON_9.3
- Code Block:

```
def elv(ton, tn, no3, nh3, nh4, no2, tp, po4):
        nutrient = '%NorP%'
        if nutrient == 'N':
           return float(max([ton, tn, no3, nh3, nh4, no2]))
        else:
           return float(max([tp, po4]))
```

S4Ind2calc field is 25% of the product of the flow and load, calculated as:

- Expression: !Sect4_ELV! * 0.25 * !Sect4_Flow! * 0.365
- Expression Type: PYTHON_9.3

For IPC licences:
Calculate values from IPC data set:

Output name(s): “Name of the Project”_%NorP%_IndustryIPC

Field(s) added in output: IPInd2calc

IPInd2calc field is the 3-year average value from the pre-processed data, calculated as:

- Expression: !%NorP%_2012_LAM!
- Expression Type: PYTHON_9.3

4.5 Septic Tank Systems

4.5.1 Septic Tank Systems Version 1

Not implemented anymore.

4.5.2 Septic Tank Systems Version 2

The input to the Septic Tank Systems model is the output from the SANICOSE model. Hence, this module identifies the N or P model output from the SANICOSE model for both groundwater (GW) and total emissions ([Total_%NorP%_2c]) for the required catchment polygons.

List of Parameters:

Name	Description	Type	More information
Domestic Septic Tanks Data [1]	Feature class containing the data generated by the SANICOSE model	Feature Class	default: SLAMpy\in\input.gdb\SepticTankSystems_Load Model17
[1] **must contain fields:** GW_N_2c, GW_P_2c, Total_N_2c, Total_P_2c			

Output name(s): “Name of the Project”_%NorP%_SepticTanks

Field(s) added in output: GWSept2calc, Sept2calc

GWSept2calc is the groundwater value Field, calculated as:

- Expression: !GW_%NorP%_2c!
- Expression Type: PYTHON_9.3

Sept2calc is the total value Field, calculated as:

- Expression: !Total_%NorP%_2c!
- Expression Type: PYTHON_9.3

Note: GWSept2calc is already included in Sept2calc. This is why the post-processing tool is using Sept2calc only for the total load from septic tank systems.

4.6 Diffuse Urban Emissions

4.6.1 Diffuse Urban Emissions Version 1

The Diffuse Urban module inputs are the catchments, CORINE, and the SLAM factors tables.

List of Parameters:

Name	Description	Type	More information
Corine Land Cover Data	Feature class containing the Corine land cover dataset	Feature Class	default: SLAMpy\in\input.gdb\ CLC18_IE
Field for Land Cover Code	Name of the field in the Corine Land Cover Data that contains the code used to distinguish land cover types	Field	default: CODE_18
Land Cover Factors for Nitrogen (N) [1]	Spreadsheet containing the export factors from the different land cover types for N	Table	default: SLAMpy\in\LAM_Factors .xlsx\Corine_N$
Land Cover Factors for Phosphorus (P) [1]	Spreadsheet containing the export factors from the different land cover types for P	Table	default: SLAMpy\in\LAM_Factors .xlsx\Corine_P$
[1] **must contain fields:** c111, c112, c121, c122, c133, c141, c142			

Output name(s): “Name of the Project”_%NorP%_Urban

Field(s) added in output: Area_ha, Urb1calc

Area_ha field is calculated as:

- Expression: !shape.area@hectares!
- Expression Type: PYTHON_9.3

Urb1calc field is the output value, calculated as:

- Expression: factor(!CODE_12!, float(!Area_ha!))
- Expression Type: PYTHON_9.3
- Code Block:

```
def factor(code, area):
    if code == '111':
        return {c111} * area
    elif code == '112':
        return {c112} * area
    elif code == '121':
        return {c121} * area
    elif code == '122':
        return {c122} * area
    elif code == '133':
        return {c133} * area
    elif code == '141':
        return {c141} * area
    elif code == '142':
        return {c142} * area
    else:
        return 0.0
```

4.7 Forestry

4.7.1 Forestry Version 1

The forestry module inputs are the catchments, CORINE, and the SLAM factors tables.

List of Parameters:

Name	Description	Type	More information
Corine Land Cover Data	Feature class containing the Corine land cover dataset	Feature Class	default: SLAMpy\in\input.gdb\ CLC18_IE *
Field for Land Cover Code	Name of the field in the Corine Land Cover Data that contains the code used to distinguish land cover types	Field	default: CODE_18 *
Land Cover Factors for Nitrogen (N) [1]	Spreadsheet containing the export factors from the different land cover types for N	Table	default: SLAMpy\in\LAM_Factors .xlsx\Corine_N$
Land Cover Factors for Phosphorus (P) [1]	Spreadsheet containing the export factors from the different land cover types for P	Table	default: SLAMpy\in\LAM_Factors .xlsx\Corine_P$
[1] **must contain fields:** c311, c312, c313, c324 * can also be readily used with clc12_IE and CODE_12, clc06_IE and CODE_06, clc00_IE and CODE_00, and clc90 and CODE_90 (all available at http://gis.epa.ie/GetData/Download)			

Output name(s): “Name of the Project”_%NorP%_Forestry

Field(s) added in output: Area_ha, For1calc

Area_ha field is calculated as:

- Expression: !shape.area@hectares!
- Expression Type: PYTHON_9.3

For1calc field is the output value, calculated as:

- Expression: factor(!CODE_12!, float(!Area_ha!))
- Expression Type: PYTHON_9.3
- Code Block:

```
def factor(code, area):
    if code == '311':
        return {c311} * area
    elif code == '312':
        return {c312} * area
    elif code == '313':
        return {c313} * area
    elif code == '324':
        return {c324} * area
    else:
        return 0.0
```

4.8 Peatlands

4.8.1 Peatlands Version 1

The peatlands module inputs are the catchments, CORINE, and the SLAM factors tables.

List of Parameters:

Name	Description	Type	More information
Corine Land Cover Data	Feature class containing the Corine land cover dataset	Feature Class	default: SLAMpy\in\input.gdb\CLC18_IE
Field for Land Cover Code	Name of the field in the Corine Land Cover Data that contains the code used to distinguish land cover types	Field	default: CODE_18
Land Cover Factors for Nitrogen (N) [1]	Spreadsheet containing the export factors from the different land cover types for N	Table	default: SLAMpy\in\LAM_Factors.xlsx\Corine_N$
Land Cover Factors for Phosphorus (P) [1]	Spreadsheet containing the export factors from the different land cover types for P	Table	default: SLAMpy\in\LAM_Factors.xlsx\Corine_P$
[1] **must contain fields:** c411, c412			

Output name(s): “Name of the Project”_%NorP%_Peat

Field(s) added in output: Area_ha, Peat1calc

Area_ha field is calculated as:

- Expression: !shape.area@hectares!
- Expression Type: PYTHON_9.3

Peat1calc field is the output value, calculated as:

- Expression: factor(!CODE_12!, float(!Area_ha!))
- Expression Type: PYTHON_9.3
- Code Block:

```
def factor(code, area):
    if code == '411':
        return {c411} * area
    elif code == '412':
        return {c412} * area
    else:
        return 0.0
```

4.9 Atmospheric Deposition

4.9.1 Atmospheric Deposition Version 1

The Atmospheric Deposition module inputs are the catchments and the pre-processed shapefile.

List of Parameters:

Name	Description	Type	More information
Data for Atmospheric Deposition [1]	Feature class containing the N and P atmospheric deposition on lakes – specially pre-processed for SLAM (see section 5.3)	Feature Class	default: SLAMpy\in\input.gdb\AtmosDep_Lakes
[1] **must contain fields:** N_Dep_tot, P_Dep_tot			

Output name(s): “Name of the Project”_%NorP%_AtmDepo

Field(s) added in output: Area_ha, AtmRate, Atm2calc

Area_ha field is calculated as:

- Expression: !shape.area@hectares!
- Expression Type: PYTHON_9.3

AtmRate field is the rate of atmospheric deposition, calculated as:

- Expression: !%NorP%_Dep_tot!
- Expression Type: PYTHON_9.3

Atm2calc field is the output value, calculated as:

- Expression: !AtmRate! * !Area_ha!
- Expression Type: PYTHON_9.3

4.10 Post-Processing Tool

4.10.1 Post-Processing Tool Version 2

The Post-processing add-on tool is used to sum up some output fields to facilitate display. Since the naming of the outputs is standardised as specified in the previous sub-sections, the only information required by the post-processing tool to retrieve the output feature class are the name of the project, the nutrient of interest, and the location of geodatabase.

Note: each new version of the SLAM comes with a new association of sub-models that may have different output fields compared with previous versions. This is why each new version of the overall SLAM comes with its own customised post-processing tool.

List of Parameters:

Name	Description	Type	More information
Name of the Project	Name that will be used to identify the outputs (used as their prefix)	String	-
Nutrient of Interest	Choice for the nutrient whose sources will be calculated	String	either “Nitrogen (N)” or “Phosphorus(P)”
Output Geodatabase	Geodatabase where all output feature classes will be stored	Geodatabase	default: SLAMpy\out\output.gdb

Output name(s): Only adding fields in existing feature class named “Name of the Project”_%NorP%_Loads_Summary

Add Fields: Wastewater, Industry, Diffuse_Urban, Septic_Tank_Systems, Pasture, Arable, Forestry, Peatlands, Lake_Deposition, TotalDiffuse, TotalPoint, Total, TotalHa, PercentGW, PercentPoint, PercentPasture

Wastewater is calculated as:

- Expression: !SUM_CSOWast2calc! + !SUM_Wast2calc!
- Expression Type: PYTHON_9.3

Industry is calculated as:

- Expression: !SUM_IPInd2calc! + !SUM_S4Ind2calc!
- Expression Type: PYTHON_9.3

Diffuse_Urban is calculated as:

- Expression: !SUM_Urb1calc!
- Expression Type: PYTHON_9.3

Septic_Tank_Systems is calculated as:

- Expression:!SUM_Sept2calc!
- Expression Type: PYTHON_9.3

Pasture is calculated as:

- Expression:!SUM_Past2calc!
- Expression Type: PYTHON_9.3

Arable is calculated as:

- Expression: !SUM_GWArab2calc! + !SUM_Arab2calc!
- Expression Type: PYTHON_9.3

Forestry is calculated as:

- Expression: !SUM_For1calc!
- Expression Type: PYTHON_9.3

Peatlands is calculated as:

- Expression: !SUM_Peat1calc!
- Expression Type: PYTHON_9.3

Lake_Deposition is calculated as:

- Expression: !SUM_Atm2calc!
  Expression Type: PYTHON_9.3

TotalDiffuse is calculated as:

- Expression: !Diffuse_Urban! + !Pasture! + !Arable! + !Forestry! + !Peatlands! + !Lake_Deposition!
- Expression Type: PYTHON_9.3

TotalPoint is calculated as:

- Expression: !Wastewater! + !Industry! + !Septic_Tank_Systems!
- Expression Type: PYTHON_9.3

Total is calculated as:

- Expression: !TotalDiffuse! + !TotalPoint!
- Expression Type: PYTHON_9.3

TotalHa is calculated as:

- Expression: !Total! / (!Area_km2! * 100)
- Expression Type: PYTHON_9.3

PercentGW is calculated as:

- Expression: (!SUM_GWSept2calc! + !SUM_GWPast2calc! + !SUM_GWArab2calc!) / !Total! *100 Expression Type: PYTHON_9.3

PercentPoint is calculated as:

- Expression: !TotalPoint! / !Total! *100
- Expression Type: PYTHON_9.3

PercentPasture is calculated as:

- Expression: !Pasture! / !Total! * 100
- Expression Type: PYTHON_9.3

4.10.2 Post-processing Tool Version 3

Changes compared to Version 2 are:

Wastewater is calculated as:

- Expression: !SUM_Wast3calc!
- Expression Type: PYTHON_9.3

All other fields are calculated as in Version 2, please refer to section 4.10.1.

4.11 Lake Retention

Lake retention is calculated as a post-processing step in Excel, using the SLAM_Catchment_Template.xlxs spreadsheet.

The number of downstream lakes has been assigned to each RiverWaterBody as follows:

1. Using lake catchment boundaries, filter out lakes with
    a. Lake area less than e.g. 1000ha and
    b. catchment areas less than e.g. 50km^2
2. Join Sub-catchments with lake catchment boundaries using 'Join field' one-to-many on LWB_CD ** *This is enough for first draft lake attenuation model***
    a. *OR spatial join with 'Centre Contains' and one-to-one****

3. Join Sub-catchments with lake catchment boundaries using 'Join field' one-to-one on LWB_CD
4. Use Tony's FME script to join lists of LWB_CD in individual fields
    a. Addition parameters of lake depth, alkalinity lake area etc. can be used in future attenuation model to be developed.

5 Data Pre-processing for SLAM Toolbox Inputs

5.1 Agriculture

The output from the Catchment Characterisation Tool (CCT) is processed into two input file for the agricultural model. These inputs may require updating if the CCT output is regenerated.

Pre-processed Input: PathwaysCCT_IRL_Arable_LPIS

Pre-processed Input: PathwaysCCT_IRL_Pasture_LPIS

The pre-processed Input files for arable (PathwaysCCT_IRL_Arable_LPIS) and pasture (PathwaysCCT_IRL_Pasture_LPIS) are calculated from the Pathways CCT Unit map, as follows:

1. Join spatial data with required attributes using 'Join Field' with:
    a. Landuse
    b. LoadingSource
    c. nTotaltoSW
    d. nGWtoSW
    e. pTotaltoSW
    f. pGWtoSW
2. Select "Landuse" LIKE 'Pasture' to generate PathwaysCCT_IRL_Pasture_LPIS
3. Select "Landuse" LIKE 'Arable' to generate PathwaysCCT_IRL_Arable_LPIS

5.2 Direct Discharge Data

5.2.1 LEMA Facilities

Irish Water provide unstructured data on emissions from wastewater discharges, in the form of a national (2013, 2014) or regional (2015) spreadsheets, which has changed format each year. Hence, the following pre-processing outline is a guideline only based on 2014/2015, and may change if the data format changes.

Some data issues:

- There may be duplication of points due to revision of license and hence code.
- The SWO data does not link with the agglom code.
- Numerical fields contain text.
- Erroneous numerical values.
- Unit inconsistencies in numerical fields.
- The method of determining PE has changed from 2014 to 2015.

5.2.2 Agglomeration data for 2016/2017

The structure of the 2016 and 2017 AER data provided by IW, differed from previous years and from each other. The following data was extracted from the AERs and formatted to give a consistent input:

- Agglomeration Name and Code
- Treatment Applied
- TN and TP Effluent Emissions and Inputs in Kg/yr
- Current Loading (PE)
  Losses associated with SWOs

All processing was performed via an R script (AERProcessingV2.R) which takes the input data and outputs TN and TP loads (Kg/yr) for each primary emission point and storm water overflow (SW) associated with an agglomeration.

Facility details were extracted from CRM in November 2017 – this forms the list of agglomerations for which estimates were either obtained or calculated as not all facilities (Certified Authorisations) were reported in the AER.

Emission points associated with agglomerations were also extracted at this time – this provided counts of each type of emission point at a facility and its coordinates.

FacilityData for processing is obtained by joining the AER data and the emission point data to Facility details and identifying the population served by the plant (PE) to use (AER data, otherwise licence data) and Treatment at the plant (AER data otherwise licence data). The treatment factors for N and P are also associated with each plant based on the treatment at the plant.

Following fields are calculated:

Field	Calculation	Comment
LOSSFraction	Use Loss Fraction calculated from AER data (see comment) if > 0.03 and < 0.9, otherwise (or if empty) use default value of 0.03.	AER data in 2017 had additional detail on SWOs and this information was used to calculate a loss fraction using one of the following approaches depending on provided data (%Loss, difference between loads collected by network and entering plant, volume emitted by SWOs as a % of total hydraulic flow, default) - The restriction on 0.9 was selected as the AERs for some plants without treatment reported loss at SWOs as 100% (loss fraction = 1) – these losses occur at the emission point. Some facilities with a Loss fraction >0.3 may be a result of misreporting treatment factors – these were investigated further (see below).
TNGenerated	PE * 12 * 365/1000	Calculates the Kg/year of Nitrogen produced by an individual
TNcalcPE	TN * N Treatment Factor	The Nitrogen treatment factors are based on the treatment in place at the plant and are specified in an input file
TNcalc	The TN Effluent emission value (kg/yr) is used unless not reported in which case TNcalcPE is used	Identifies how much Nitrogen was emitted at the emission points associated with the plant
TN_SWO	TNGenerated/(1-LOSSFraction) *LOSSFraction	Estimates the amount of Nitrogen lost in sewer network as a result of overflows based on the LOSSFraction and the amount estimated to be generated.
TN_Agglom	TNcalc+TN_SWO	Total amount of Nitrogen emitted by the agglomeration via the sewage network
TN_EmissionPoint	TNcalc/Number of Emission points	Spreads the amount of Nitrogen emitted at emission points over all the emission points in the agglomeration – typically there is only one but for cases where multiple, this may be a simplification i.e. if one is much smaller than the other
TN_SWOPoint	TN_SWO/# of Stormwater points	Assigns the estimated amount of Nitrogen lost (Kg/year) across the SWO's in the agglomeration – is a simplification as some SWO's account for much more than others (pumping stations and those at the plant) and not all SWO's in a network are identified as emission points and so are not counted. If there are no SWOs associated with the facility, this is set to zero.

These fields are replicated for Phosphorous (TP).

This data is then linked to emission points via agglomeration code and emission point type and output as a csv file EmissionPointData<YYYY>.

Following Assumptions were made:

- Treatment in 2016 was applied to 2017 agglomerations as AER2017 did not provide this information in an easily extractable manner
- In cases where more than one WWTP was associated with an agglomeration, the PE, emissions and effluents were summed for each plant.
- SWO losses shared over all SWO's however certain SWO's are typically associated with the majority of losses in an agglomeration i.e. pump houses.
- Treatment factors applied based on treatment tend to overestimate emissions of N from secondary and tertiary P treatment, whilst underestimating that from Primary treatment. Treatment factors for P match the experimental data quite well.

5.2.3 Agglomeration data for 2015

Pre-processed Input: SLAM_Agglom_March2017

Join LEMA_Emission_Points_20160818 with AER spreadsheet on EmissionID field.

Spatial join with MainChannel_Estuary500.shp -> To generate is main channel field (**IsMainChan**)

IsMainChan output value is calculated as:

- Expression: factor
- Expression Type: VB
- Code Block:

```
If [IsMainChan] = "Y" Then
 factor = "Y"
Else
 factor = "N"
End If
```

5.2.4 Agglomeration data for 2014

The following (now superseded) uses data relating to 2014 emissions:

Pre-processed Input: LAM_UrbanWasteWaterDec15.shp

Data from vector.SDE.LEMA_Facilities. See spreadsheet for more details.

- For Urban waste water, Select Category = 'Urban Waste Water'
- Link spatial data with AER collated spreadsheet
- Re-calculate LEMA_PE = [Agglomerat]

5.2.5 Section 4 Facilities

This data was a deliverable of the 'abstractions and discharges project', and required minimal pre-processing. This original data set is not maintained and hence no updates to this input shapefile are envisioned.

Pre-processed Input: Sect4_Disch_F01_IsMain.shp

ELV and discharge data from Abstractions & Discharges database

Extract from gdb to shapefile

Spatial join with MainChannel_Estuary500.shp -> To generate is main channel field

Tidy up with attribute table code (to fill in “N”)

5.2.6 IPC Facilities

The IPC facilities input data could be updated annually.

Pre-processed Input: IPPC_Loads_LAM2.shp

For Industry, Select NOT Category = 'Urban Waste Water'. See spreadsheet for more details.

- Get PRTR database extract
- Link with spreadsheet with PRTR loads and licenced limits
- Spatial join with MainChannel_Estuary500.shp -> To generate is main channel field (**IsMainChan**)
    - Tidy up with attribute table code:

IsMainChan output value is calculated as:

- Expression: factor
- Expression Type: VB
- Code Block:

```
If [IsMainChan] = "Y" Then
 factor = "Y"
Else
 factor = "N"
End If
```

5.3 Atmospheric Deposition (Lakes)

This pre-processed data will not require updating.

Pre-processed Input: AtmosDep_Lakes.shp

6 Example Output: Suir Catchment

Results for the Suir catchment are described in Mockler et al. (2016), indicating that diffuse losses of nutrients from pasture are the dominant source of nitrogen and both wastewater discharges and losses from pasture are dominant for phosphorus. The overall annual phosphorus emissions are dominated by a large municipal discharge at the mouth of the catchment (Figure 3).

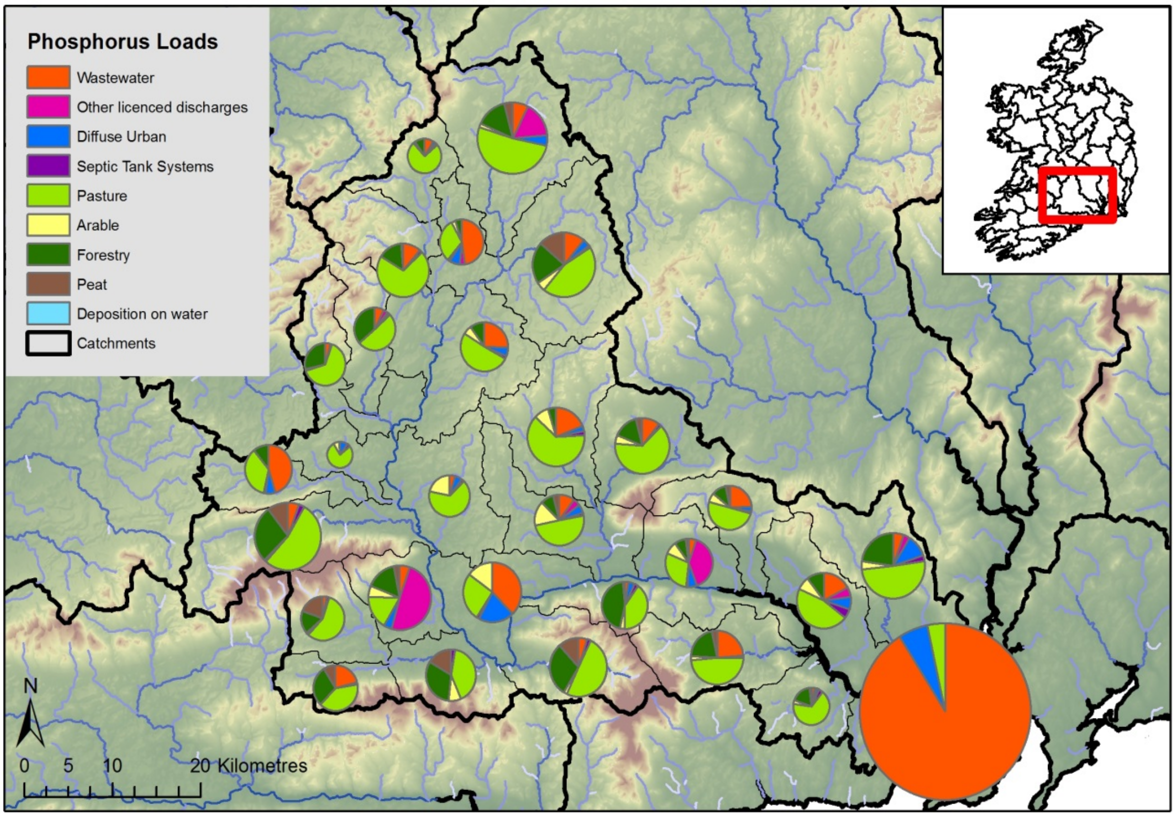


Figure 3. Phosphorus load apportionment results for the Suir catchment (size of pie indicates relative contribution of annual loads from each sub-catchment).

7 Conclusions

The SLAM framework has been developed to produce nutrient load apportionment in Irish catchments based on the best available national models and data. The framework was developed as an ArcGIS toolbox using ModelBuilder, with individual sub-models for each sector. This design was chosen because of the possibility for contributions to model development from a broad range of scientists, without the need for experience in programming.

The results presented here have been reproduced at national level and are incorporated into the Irish EPAs WFD characterisation process. In addition to the identification of significant pressures, the SLAM toolbox can be used to assess scenarios of changes in pressures and mitigation measures, thereby supporting decision making in the water domain.

Using local knowledge in conjunction with scientific evidence, such as provided by the SLAM, effective, targeted measures can be identified to reduce nutrient losses to water. This will potentially lead to improvements in the health of our aquatic environment and valuable ecosystem services for citizens.

7.1 Future SLAM developments

The SLAM Framework was designed to facilitate incorporation of data and model updates. Some of the key on-going/future developments for the sector calculations include:

- New hydrology flow path model for diffuse pollution
- Assessment of uncertainties in load estimates due to data and methods,
- Comparisons of nutrient loads in different years,
- Scenario analysis and assessment of implementing measures, and
- Estimated load source apportionment for a critical time period i.e. summer flows.

Acknowledgments

The author acknowledges funding for the *CatchmentTools* Project from the Irish Environmental Protection Agency (project ref. 2013-W-FS-14) on behalf of the Department of the Environment, Community and Local Government. We thank EPA laboratory, informatics and scientific staff, and OPW for providing national datasets integral to this research. The author would particularly like to acknowledge members of the EPA Catchment Science and Management Unit.

References

Mockler, E.M., Deakin, J., Archbold, M., Daly, D., Bruen, M., 2016. Nutrient Load Apportionment to Support the Identification of Appropriate Water Framework Directive Measures. Biology and Environment 116B(3) 245-263. http://dx.doi.org.10.3318/BIOE.2016.22

Ní Longphuirt, S., Mockler, E., O'Boyle, S., Wynne, C., Stengel, D.B., 2016. Linking changes in nutrient load source apportionment to estuarine responses: An Irish perspective. Biology and Environment 116B(3) 295-311. https://doi.org/10.3318/bioe.2016.21