Supporting **I**mplementation of **M**aritime Spatial Planning in the **Celt**ic Seas

Component 1: Supporting implementation of MSP

Sub-component: C1.2.4 Approaches to MSP Implementation

Deliverable 11(b): Case Study 2 – Assessment of Cumulative Effects in Marine Spatial Planning: Irish Sea Pilot Project Methodology

European Commission
Directorate-General for Maritime Affairs and Fisheries
Grant Agreement No. EASME/EMFF/2014/1.2.1.5/3/SI2.719473 MSP Lot 3

SIMCelt partners

Deliverable ID.:	SIMCelt-C1.2.4-D11
Deliverable Lead Partner:	**Marine Institute**
Due Date of Deliverable:	29/09/17
Completion Date of Deliverable:	29/09/17
Start Date of Project:	29/12/15
Duration:	26 Months
Version:	1
EU Grant Call:	Action on Maritime Spatial Planning in the Northern European Atlantic
EU Project Officer:	David San Miguel Esteban

Dissemination Level		
PU	Public	
PP	Restricted to a group specified by the consortium (including the Commission services)	
RE	Restricted to other programme participants (including the Commission services)	
CO	Confidential, only for members of the consortium (Including the Commission services)	

Document Information

Deliverable Title	D11(b) Assessment of Cumulative Effects in Marine Spatial Planning: Irish Sea Pilot Project Methodology
Coordinator	Caitríona Nic Aonghusa, Marine Institute
Authors	Caitríona Nic Aonghusa, Marine Institute Michelle Riordan and Seth Girvin, Compass Informatics
Recommended Citation	Nic Aonghusa, C., Riordan, M. and Girvin, S. 2017 Deliverable 11(b): Assessment of Cumulative Effects in Marine Spatial Planning: Irish Sea Pilot Project Methodology. EU Project Grant No.: EASME/EMFF/2014/1.2.1.5/3/SI2.719473 MSP Lot 3. Supporting Implementation of Maritime Spatial Planning in the Celtic Seas (SIMCelt). University College Cork. 71pp.

Version History

Date	Document Version	Reviewer	Revision
20th June 2017	Ver 1.0	Michelle Riordan/ Seth Girvin	
27th June 2017	Ver 1.1	Caitriona Nic Aonghusa	
24th March 2018	Final	Caitriona Nic Aonghusa	

Table of Contents

Contents

About SIMCelt

SIMCelt - Supporting Implementation of Maritime Spatial Planning in the Celtic Seas is a two-year €1.8 million project co-financed by DG Mare and focussed on promoting the development of transnational cooperation to support the implementation of EU Directive 2014/89/EU in the Celtic Seas. Led by University College Cork, the project consortium comprises both planners and researchers from seven partner institutes representing a mix of governmental authorities and academic institutes from Ireland, France and the UK. This consortium is particularly interested in developing meaningful cooperation between neighbouring Member States to support implementation of spatially coherent plans across transboundary zones of the Celtic Seas, building on previous work and leveraging new opportunities to identify and share best practice on technical, scientific and social aspects of transboundary MSP.

.

Summary

This report details the methodology applied for Cumulative Effects Assessment as part of SIMCelt Case Study #2 *Cumulative Effects Assessment in Marine Spatial Planning*. A pilot project was undertaken in the NW Irish Sea. This document details the spatial analyses undertaken and summarises findings and recommendations for consideration in future work.

The aim of this SIMCelt Case Study was to develop an approach to Cumulative Effects Assessment for Maritime Spatial Planning. The case study was a demonstrator project and focused on a select few activities and environmental parameters. We carried out a Cumulative Effects Assessment of a single pressure from multiple activities, i.e. a cumulative effects assessment of seabed disturbance caused by fishing, shipping and aquaculture. Good quality spatial and temporal data exist for these activities. There are also other pressures and activities but the data do not currently exist for inclusion in the assessment.

Part 1

Overview and Findings

1.1 Introduction

The main purpose of MSP is to promote sustainable development and to identify the utilisation of marine space for different uses. Directive 2014/89/EU establishing a Framework for Maritime Spatial Planning (MSP) states that an Ecosystem-Based Approach should be applied to the implementation of the Directive. The framework provides for consistent transparent sustainable and evidence-based decision making. Part of this is to manage spatial uses and conflicts in marine areas, therefore, addressing Cumulative Effects is an essential part of this process. This case study assesses if and how this could be undertaken.

The aim of this SIMCelt Case Study was to develop an approach to Cumulative Effects Assessment for Maritime Spatial Planning. The case study was a demonstrator project and focused on a select few activities and environmental parameters. We carried out a Cumulative Effects Assessment of a single pressure from multiple activities, i.e. a cumulative effects assessment of seabed disturbance caused by fishing, shipping and aquaculture. Good quality spatial and temporal data exist for these activities. There are also other pressures and activities but the data do not currently exist for inclusion in the assessment.

Cumulative effects assessment (CEA) is defined as a systematic procedure for identifying and evaluating the significance of effects from multiple human activities. It can also provide an estimate of the overall expected impact in order to inform management decisions. Analysis of the causes, pathways of exposure and consequences of these effects on ecosystem components is an essential and integral part of the process. (OSPAR, 2017). Many policy documents on MSP refer to cumulative 'impacts' or 'effects'([1]) in some way, however there is a need to consider in further detail how CEA can be integrated into the MSP process (MMO, 2014).The purpose of this component was to carry out Spatial Analysis to investigate Cumulative Effects in the Irish Sea.

As a Pilot Project, it was decided to define a process and complete an analysis for a single pressure (Abrasion), in order to demonstrate the overall approach that can be applied to a broader set of pressures in the future.

Abrasion was considered the most appropriate pressure to pilot as it has the most complete suite of data available for both the Irish Sea. Using the ODEMM tool the following human activities were prioritised as a cause of Abrasion pressure and were included in the pilot are: Aquaculture, Beam Trawling, Bottom Trawling, Dredging and Shipping.

---

[1] *The term 'Cumulative Impacts' and 'Cumulative Effects are often used interchangeably, in this report we use 'Cumulative Effects' as standard*

A study area in the relatively busy Irish Sea was selected to assess how a cumulative effects assessment methodology could be applied within the MSP process (Figure 1). The area is in the NW of the Irish Sea, extending from Dublin Bay to Carlingford Lough on the East coast of Ireland. It was selected based on the availability of data, existing activities and consideration of transboundary issues. Carlingford Lough is located between the Republic of Ireland and Northern Ireland, though there is no officially agreed legal boundary. The study area incorporates the most productive fishing grounds in the Irish Sea. Shipping activity is primarily import and export of goods or passenger transport (ferry and cruise) and includes containers, liquid and dry bulk, Ro-Ro units and other general cargo. Dublin Port is the primary port. Total throughput in 2016 was 34.9m tonnes, 1,814,089 passengers travelled through the port and 109 cruise vessels visited the port.

Initially only data for the Irish Sea was included in the model, as UK abrasion data is available mostly in the form of webservices (the format of which means it cannot be used in the spatial analysis model, however it is expected that the original files from which these webservices are provided can be used in a re-run of the model)

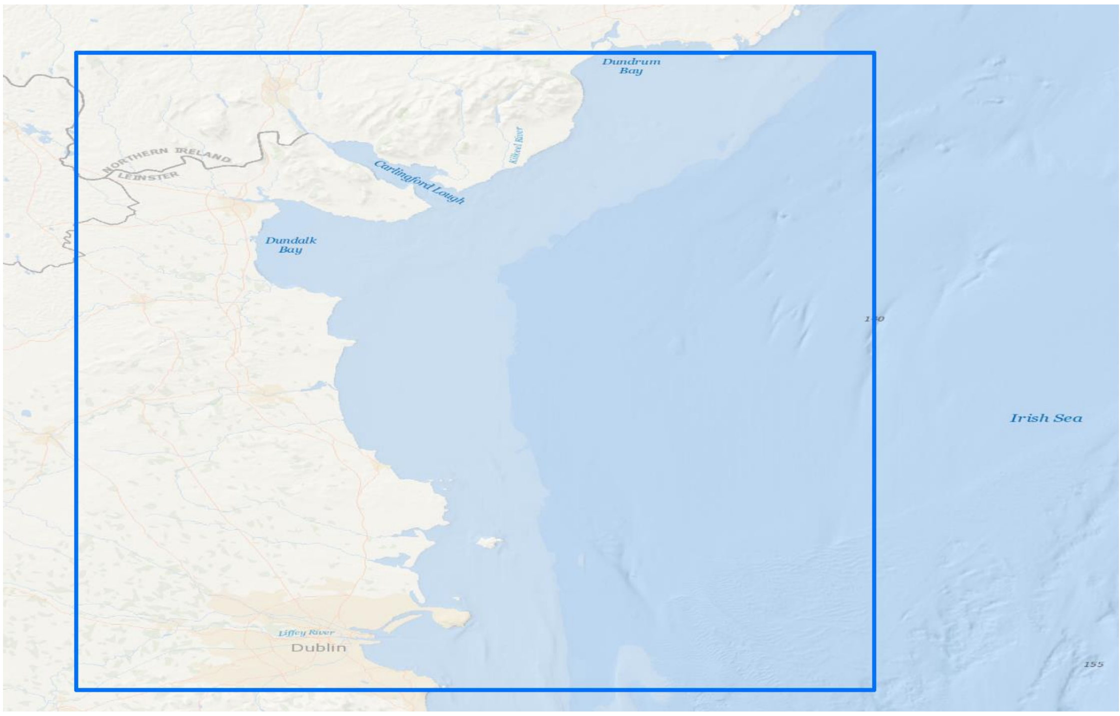


Figure 1 Irish Sea Pilot Study Area

1.2 Spatial Analysis Tasks

a. Projections and Cell Sizes

Finding: Irish and UK datasets relevant to cumulative effect analysis are stored in a variety of projection systems.

The majority of Marine Institute data is stored in WGS 1984 Coordinate Reference System (EPSG:4326) because station locations are in longitude latitude measurement units. There is some projected data from the INFOMAR programme in UTM Zone 29N.

The Irish Sea has extents outside of the ITM projection, and future study areas may not fit exactly into UTM Zone 29N, for this reason it was decided to keep the modelling data in WGS84 (Lat/Long); as this coordinate system is based on a sphere rather than projected (flat), cell sizes will not have equal length and width values.

The cell sizes chosen for the analysis are 0.002 degrees (Cell height of 222.6m and Cell width of 134m). This can be changed easily within the script provided to the Marine Institute. The main reason this cell size would be increased is to anonymise the sensitive data further and this could be requested by some data providers.

Recommendation

For any future analysis to work, it is necessary to ensure all datasets are in a standard projection system. WGS 84 is the recommended standard projection system to apply to all datasets.

b. Pre-Processing of Data

Finding: there is a great wealth and variety of datasets with the potential for inclusion in cumulative effects assessment. However, in order to include them in any spatial analysis model each dataset requires individual assessment and processing to ensure it is appropriate to the specific requirements of the spatial analysis.

The following pre-processing procedures were undertaken before the script could be run on the data.

- Aquaculture data – ROI Aquaculture Data and NI Aquaculture data was merged together.
- Shipping Data – AIS shipping data that was within 5m from the seafloor was selected and all other shipping data was excluded. To find ships that were greater than 5m from the sea floor, bathymetry data was used to give each ship a Sea Depth. The AIS Shipping Data has a Draught column so a new field was added to the data and the Sea Depth was subtracted from the Draught to give the distance from the ship hull to the sea bed. All Ships with a draught within 5m of the sea bed were selected and exported as a new layer – This layer is used in the analysis.

A python script was created to make this process easier and faster and is available to re-run in future analyses. Manually doing this process is very slow as the AIS dataset is very large.

Recommendation:

Each dataset should be reviewed individually to ensure it is fit for purpose. Where required and possible, processing can be automated and carried out as a scripted batch of work to save time.

1.3 Spatial Dataset Modelling

Finding: datasets representing activities can be related to one or multiple pressures. This relationship is defined through specific research methodologies. A spatial analysis model focusing upon the pressure, can combine multiple activities to give a visual representation of the cumulative effect of the activities for end users, specifically those involved in the planning process.

A Python based model was developed for spatial analysis, this model carries out the following steps:

- All vector datasets are converted to grid based raster files with presence/absence values or value counts for each activity assigned to each raster grid cell
- A weighting to indicate the significance of its relationship to the pressure was then applied to each activity. The weightings used were based upon the Impact Risk Scores from the ODEMM tool (see Table 1 below).
- The activities and their weightings are combined via a series of calculations to present an overall representation of the cumulative effect of Abrasion (**Error! Reference source not found.**).

Note: Using the tools provided, weightings in the script can be easily modified by the Marine Institute.

Table 1. Abrasion Activity weightings for each activity taken from ODEMM.

Abrasion Activity	Impact Risk Score applied
Aquaculture	0.0018624
Beam Trawling	0.27326
Bottom Trawling	0.608
Dredging	0.0423
Shipping	0.000908

When the weightings are applied to each activity raster a new raster dataset is produced indicating where that activity has caused abrasion. All rasters are then combined arithmetically to give an overall representation of the combined Abrasion pressure from the associated activities.

The formula to calculate overall Abrasion is as follows:

Model abbreviations:

- CV = Cell Value
- IR = Impact Risk

Formula for Overall Abrasion Score

- AquacultureCV - will be either 1 or 0 dependent on the presence of aquaculture.
- BeamTrawlingCV - will be a count of all recordings of this fishing type for a cell, for a particular year (in this case 2016 data).
- BottomTrawlingCV - as above.
- DredgingCV - as above.
- ShippingCV - count of all recordings of boats in a cell.

(AquacultureIR * AquacultureCV) + (BeamTrawlingIR * BeamTrawlingCV) + (BottomTrawlingIR * BottomTrawlingCV) + (DredgingIR * DredgingCV) + (ShippingIR * ShippingCV)

Figure 2 Abrasion Formula

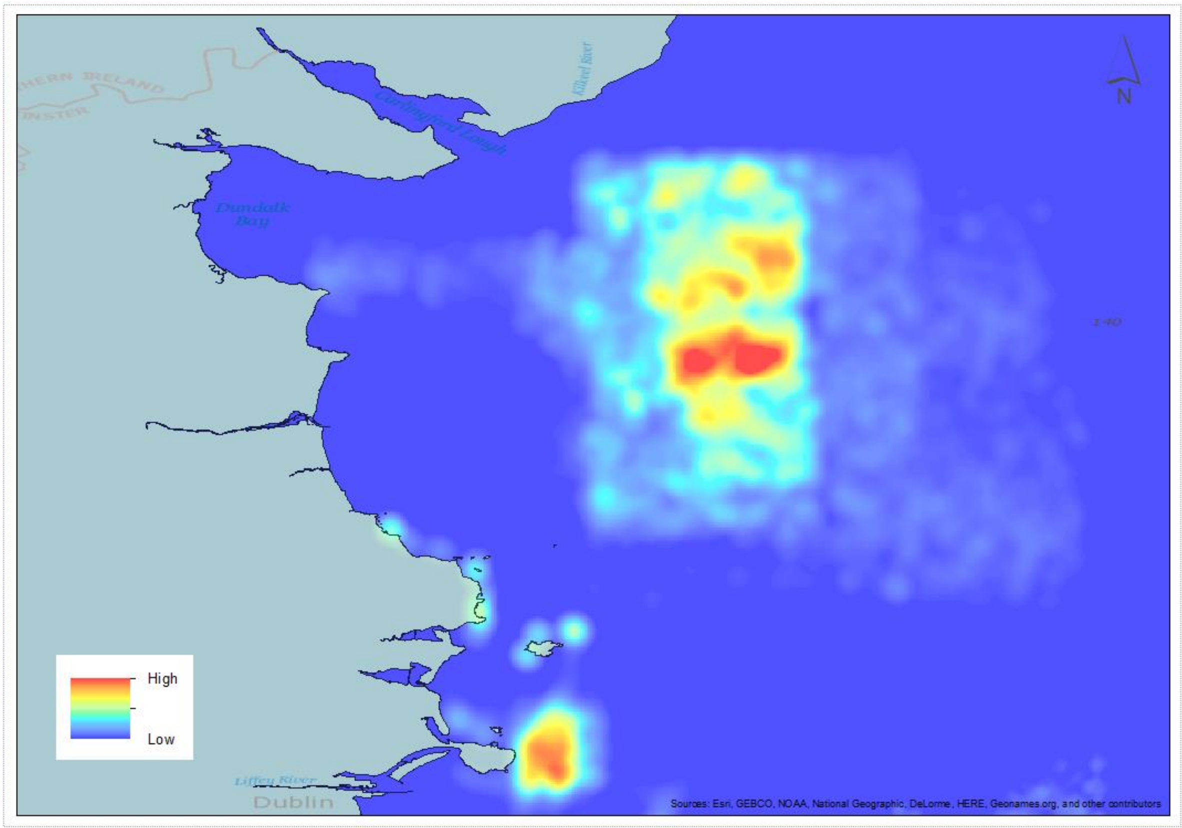


*Figure 3 Overall Abrasion Raster*** based upon Mock Data*

Figure 3, which is based upon mock data, illustrates the areas of the pilot study where activities relating to abrasion (Aquaculture, Beam Trawling, Bottom Trawling, Dredging and Shipping) have taken place, as per the data collected. Areas illustrated in warmer, red tones illustrates a higher density of activities, than the cooler blue areas. Those areas in darker blue have no recorded abrasion activity within the datasets.

The rasters illustrating the activities can then be combined with other raster images representing sensitive receptors to represent the risk posed by certain activities. As part of the pilot study, a map was prepared with values indicating the sensitivity of the habitats to abrasion. These sensitivity values were based upon the ODEMM tool. The habitats sensitivity was then combined with the Abrasion map to illustrate the risk of habitats to abrasion is taking place.

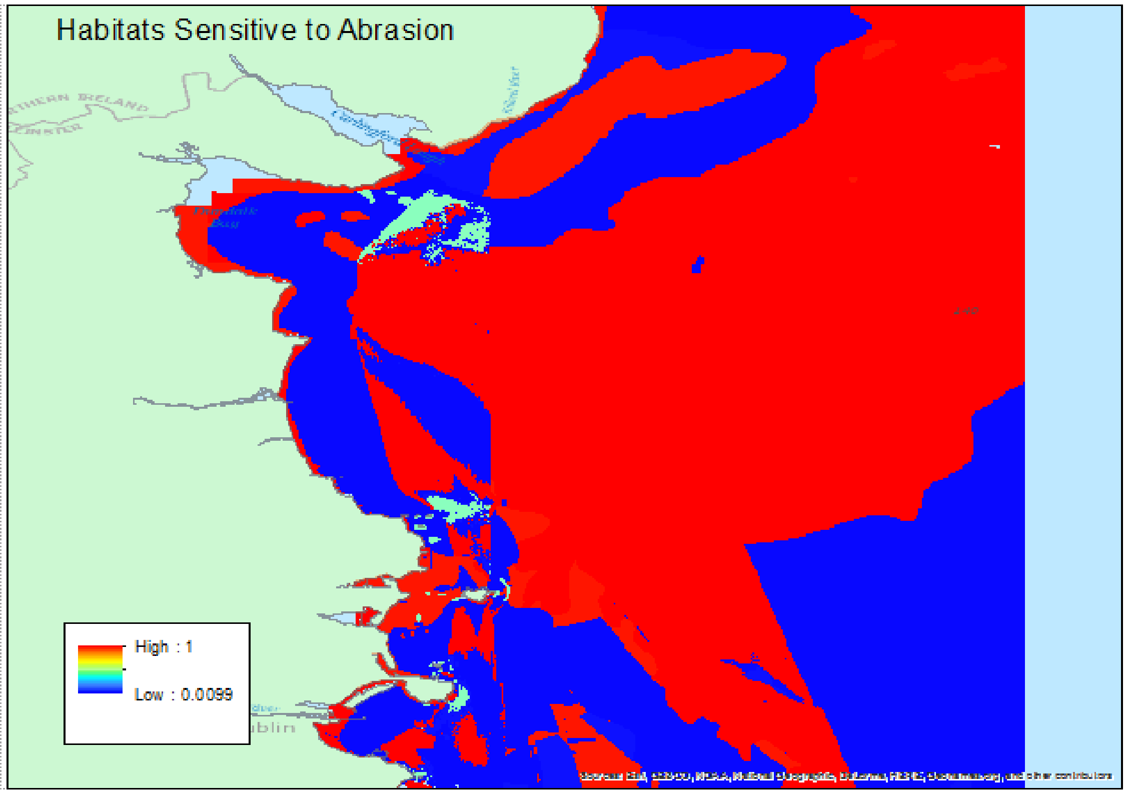


Figure 4 Habitats Map - Sensitivity to Abrasion

The habitat grid contained a cell value with its risk weighting, then the simple multiplication (below) was used;

Formula for Habitat Impact;

(AbrasionCV * HabitatCV)

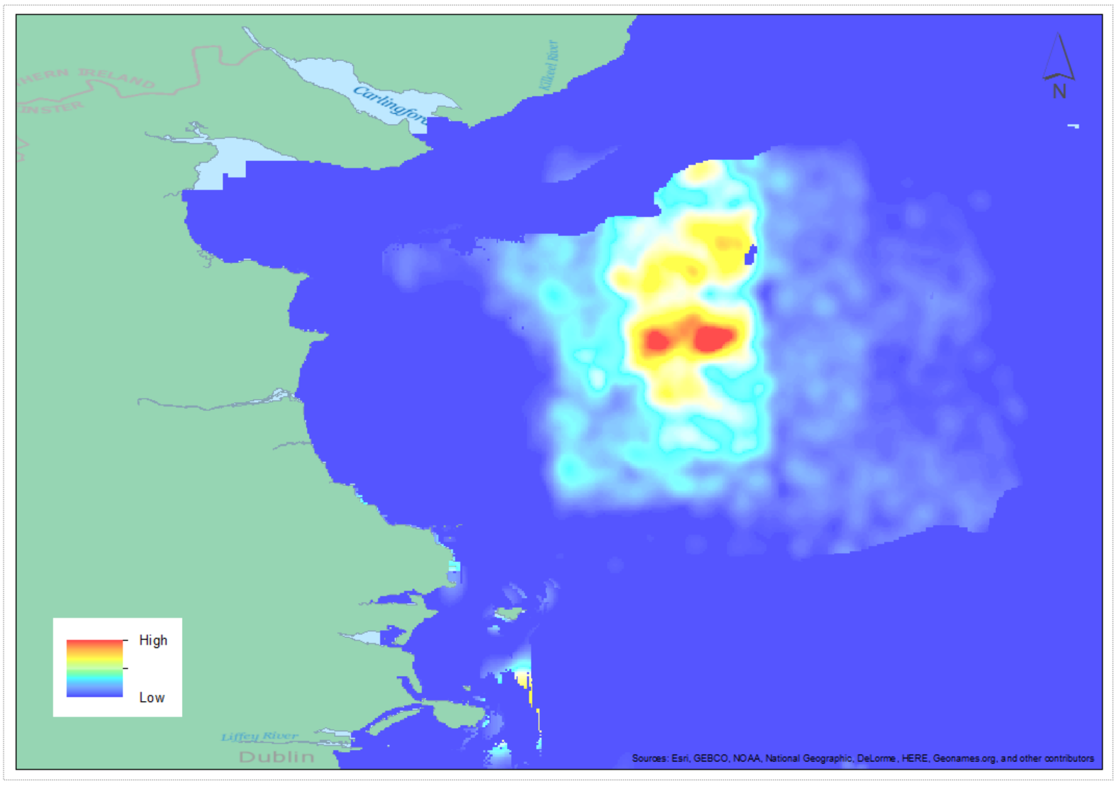


*Figure 5 Overall Habitat Impact raster. ***Based upon mock data*

From the output rasters of this model, we understand that abrasion has the potential to be caused by a number of activities with various degrees of significance. Activities were included in the model as different GIS layers. These layers were weighted according to the potential significance upon the pressure.

Each layer has an effect on abrasion, which is best solved using a raster-based approach.

All datasets used in the model were converted to Raster (grid based) files. This approach has the following advantages:

1) Weightings can be created for each activity type at the grid cell level (See ODEMM Pressure Assessment Tool below for more information on weightings applied in this Analysis)

2) Sensitive data can be anonymised, and its resolution reduced to allow for publishing.

Recommendation:

Each of the technical steps used in generating the spatial analysis model for Abrasion can be reused for other pressures. When re-applying the model, weightings will have to be sourced from ODEMM, or a similar suitable source

1.4 Spatial Analysis Techniques

The following GIS Spatial Analysis Techniques were employed as part of the overall analysis:

a. Point to Raster (Conversion)

Licensing Information - Requires Spatial Analyst or 3D Analyst, or an ArcInfo license.

This was used on Shipping and the 3 fishing layers (Beam Trawling, Bottom Trawling and Dredging). This tool takes an input point dataset and creates a new raster with each cell containing the count of points within the specified region.

This tool takes an input point dataset:

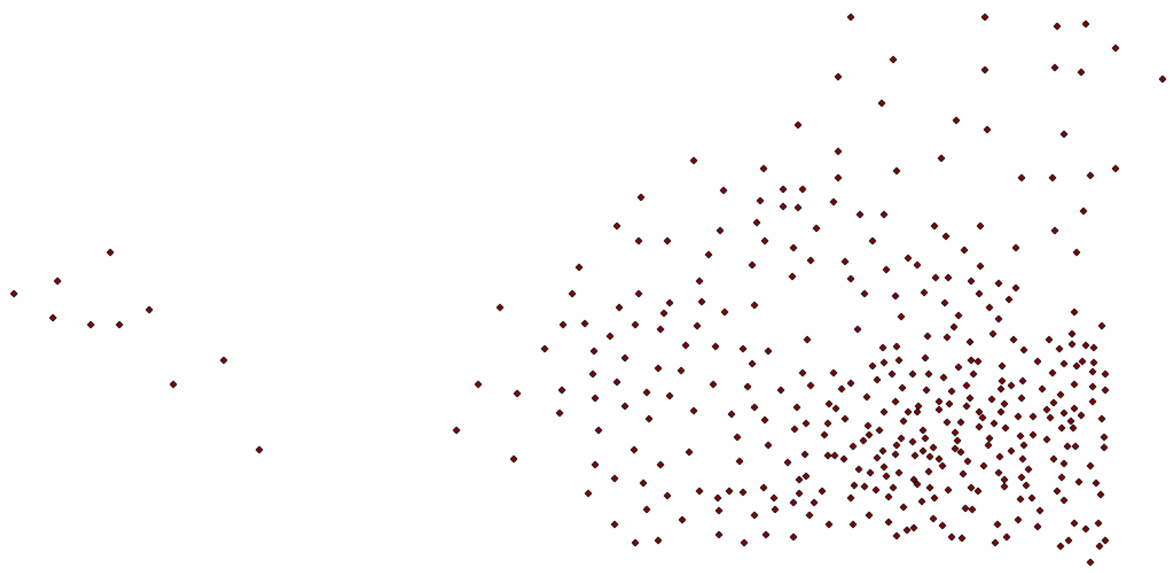


*Figure 6 - Input point dataset above is Bottom Trawling. ***The data used to generate this raster was mock data.*

And creates a new raster grid with each cell containing the count of points within the region:

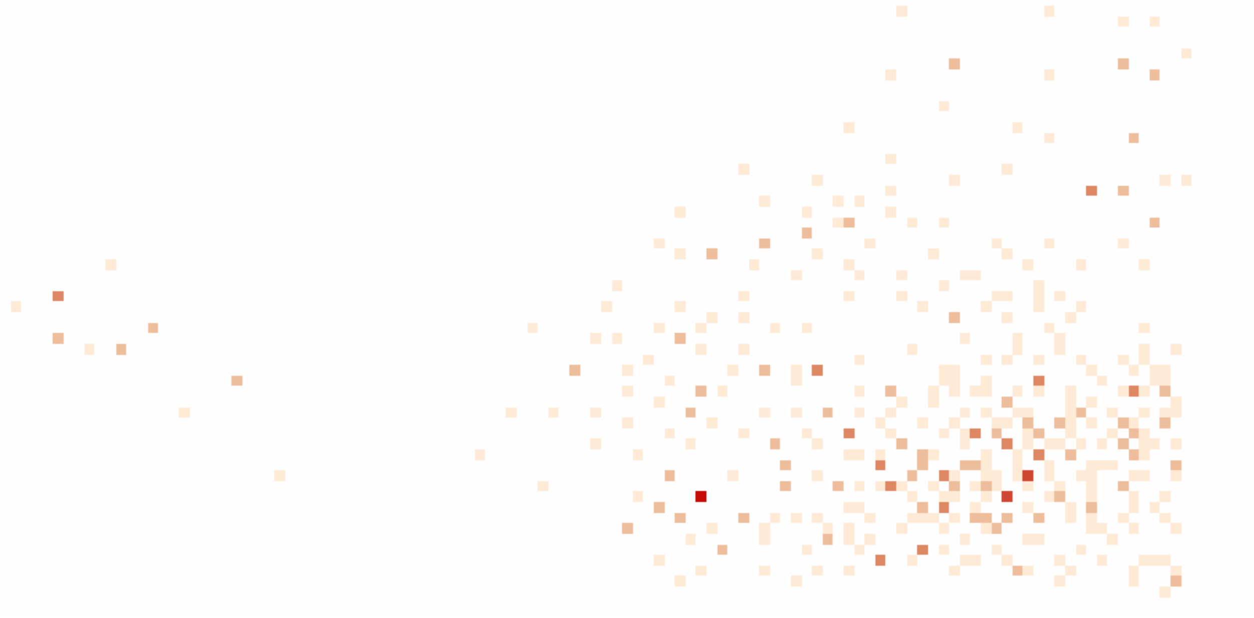


*Figure 7 New raster grid created from the Bottom Trawling data. ***The data used to generate this raster was mock data.*

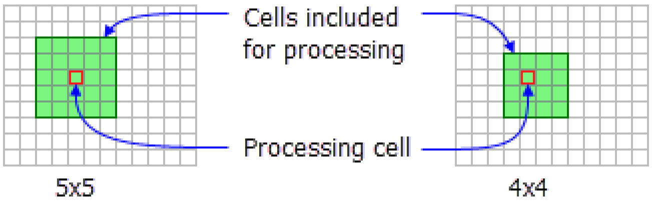
b. Point Density / Kernel Point Density

An alternative to the Point to Raster tool above is the Point Density tool]. This calculates a magnitude-per-unit area from point features that fall within a neighbourhood around each cell. Note the values on the output raster will always be floating point.

An example of the output for the sample data (Bottom Trawling) above is shown below:

*Figure 8 New raster grid created from the Bottom Trawling data. ***The data used to generate this raster was mock data.*

It was decided this approach gives a more accurate output. The input fishing points represent boats of a certain speed, but the abrasion damage is a continuous line between these points.



A rectangular neighbourhood will be used with a cell size of 3x3.

When added together, the population values of the cells equal the sum of the population of the original point layer. In a simple density calculation, points that fall within the search area are summed, then divided by the search area size to get each cell's density value.

c. Kernel Density

The Kernel Density tool calculates the density of features in a neighbourhood around those features

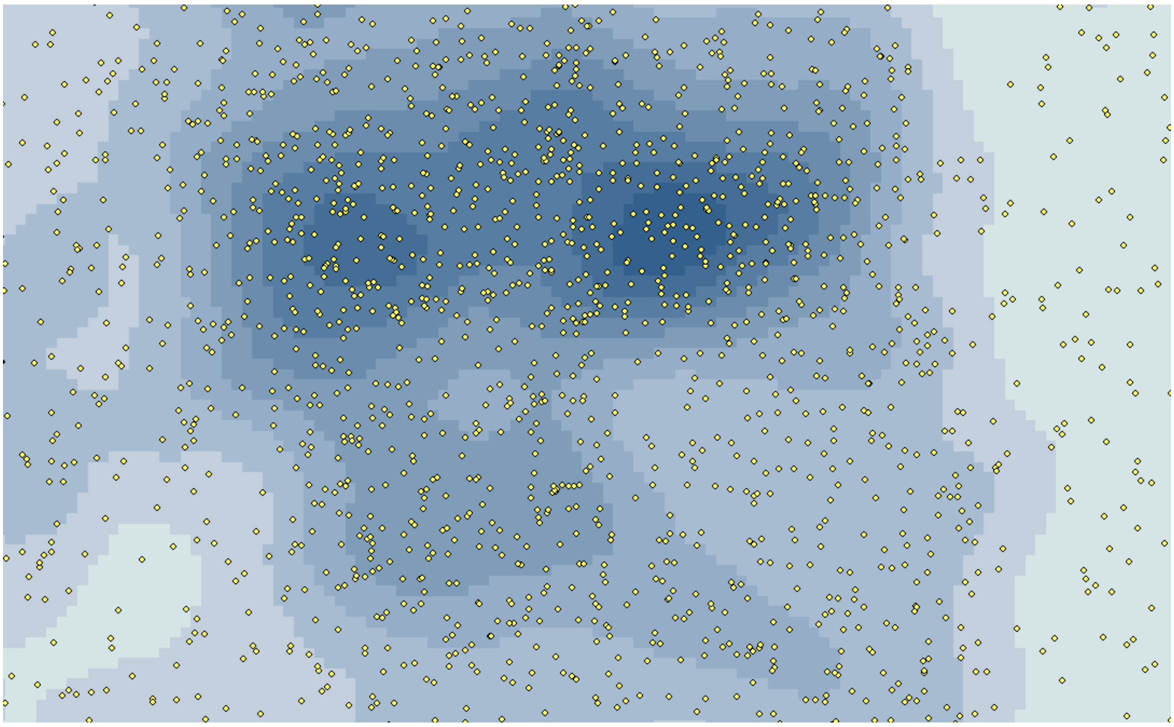


*Figure 9 Kernel Density output created from the Bottom Trawling data. ***The data used to generate this raster was mock data.*

d. Presence/Absence

Presence/Absence was included in the overall script to allow the planner to see where activities overlap.

This analysis was undertaken to determine if this simpler process could be used instead of the detailed analysis above. The output of this analysis would show a planner where there were 0 activities, 1 activity, 2 activities in the same location, 3 activities in the same location etc.

For this analysis, all data were changed to rasters. These all had 0.002 cell size like in the analysis above. These rasters were then reclassified to simply contain presences values. From these presence/absence grids for each activity an overall output for Abrasion was created (see image below using mock data).

The advantage of this analysis is that it is simple for planners to do and they can see where there are multiple activities. However, weightings are not included in this analysis so all activities are considered to have the same impact.

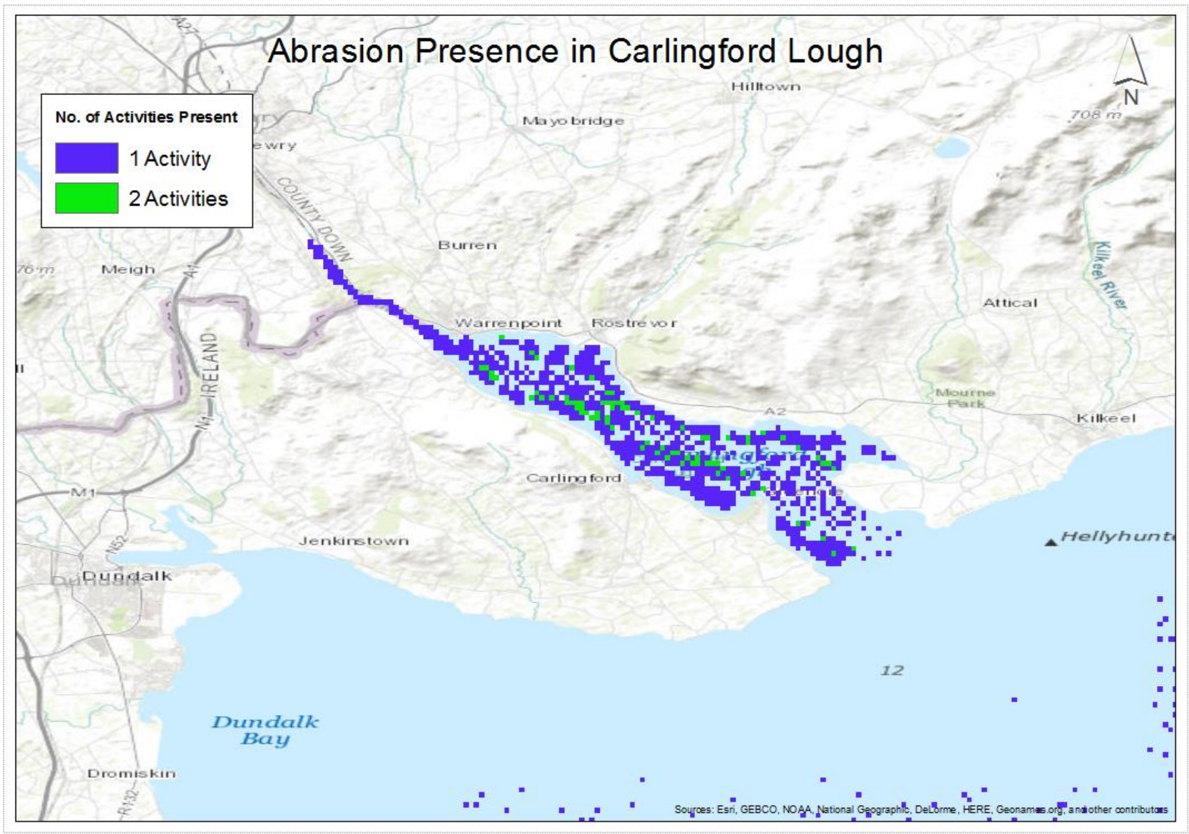


*Figure 10 Abrasion Presence in Carlingford Lough. ***The data used to generate this raster was mock data.*

e. ODEMM Pressure Assessment Tool

The ODEMM Pressure Assessment tool was used firstly to give a weighting to each activity and the output of this provided a raster for each activity showing where they caused abrasion. This tool accesses and ranks the threat associated with any particular sector/pressure combination on any ecological component. This tool was used firstly to apply an impact risk score to each activity, the output of this was a raster for each activity showing where that activity caused pressure. The raster calculator tool was then used to sum all of these rasters to provide an overall Abrasion raster. This raster showed where Abrasion occurs in the chosen study area.

In order to find how Abrasion impacts habitats, the ODEMM tool was used again to give an Abrasion Max impact score for each habitat in the study area. This resulted in a raster showing how Abrasion impacts habits in the study area.

1.5 Outputs from the Analysis

The following datasets have been prepared and are available to the Marine Institute:

Raster outputs:

abrasion.tif - overall abrasion pressure, created from all input rasters

abrasion_presence.tif - a simple grid that displays the sum of abrasion pressures in each cell (a count of features)

aquaculture_presence.tif - a raster containing 1s and 0s to indicate aquaculture presence

beam_trawl_kernel_density.tif - a raster showing Beam Trawl activity using kernel density

beam_trawl_point_density.tif - a raster showing Beam Trawl activity using point density

beam_trawl_presence.tif - a raster showing 1s and 0s to indicate Beam Trawl activity

bottom_trawl_kernel_density.tif - a raster showing Bottom Trawl activity using kernel density

bottom_trawl_point_density.tif - a raster showing Bottom Trawl activity using point density

bottom_trawl_presence.tif - a raster showing 1s and 0s to indicate Bottom Trawl activity

dredging_trawl_kernel_density.tif - a raster showing Dredging activity using kernel density

dredging_trawl_point_density.tif - a raster showing Dredging activity using point density

dredging_trawl_presence.tif - a raster showing 1s and 0s to indicate Dredging activity

habitat_weightings.tif - a raster showing Habitat abrasion weightings

habitat_risk.tif - raster showing total abrasion impact taking into account habitats - this is the final results grid

shipping_trawl_kernel_density.tif - a raster showing Shipping activity using kernel density

shipping_trawl_point_density.tif - a raster showing Shipping activity using point density

shipping_trawl_presence.tif - a raster showing 1s and 0s to indicate Shipping activity

From the above outputs the main two outputs needed are:

abrasion.tif - overall abrasion pressure, created from all input rasters

habitat_risk.tif - raster showing total abrasion impact taking into account habitats - this is the final results grid

1.6 Additional Recommendations for future Analysis

In order to complete the analysis for the remaining pressures for all of the Irish Sea it is recommended that the following steps are taken.

1. *Data Gap evaluation*

- Identify what data is needed and where to get it i.e. how to obtain shapefiles for the available Web services. Time will have to be taken to contact owners of these web services to see if they can provide the users with Shapefiles.
- In this Pilot, gaps have been identified in data such as the Habitat and Bathymetry data that was used in the analysis, no data existed for some areas along the Irish coast. It is important to identify these data gaps before any analysis is under taken.

2. *New location to save data*

- As new data is being sourced and received for the remaining pressures through the Data gap evaluation task it is important that a new data folder is created so that only data that is going to be used in the analysis is saved here.
- Only data that has been approved for use in the analysis should be saved in this folder. As the Marine Institute has a lot of data for the same activity, choosing one dataset for each activity will ensure the correct/approved data is being used in the analysis.
- Data saved in MI databases that are needed for the analysis should also be exported and saved here. Saving all the required data in one location will not only save time when adding data, it will reduce the likelihood of data links been broken.

3. Recommendations on Sourcing Data for Activities

- Tourism/Recreation – Contact Fáilte Ireland for Irish data and Visit Britain for UK data. Fáilte Ireland have Activities data available on their website which have locations of outdoor pursuit centres, they don’t provide visitor numbers with these but Fáilte Ireland might have these available internally.

Part 2

General Analysis Approach

2. General Analysis Approach

This section of the report details design choices and decisions to implement GIS modelling of abrasion in the Irish Sea.

2.1. Projections and Cell Sizes

Current map projections used within the Marine Institute are summarised as follows:

The majority of Marine Institute data is stored in WGS 1984 Coordinate Reference System (EPSG:4326) because station locations are in longitude latitude measurement units. There is some projected data from our INFOMAR programme in UTM Zone 29N, however, for mapping purposes we would tend to plot in other projection systems such as Web Mercator and Irish Transverse Mercator (close to shoreline small area maps).

The Irish Sea has extents outside of the ITM projection, and future study areas may not exactly into UTM Zone 29N, for this reason it was decided to keep the modelling data in WGS84 (Lat/Long). As this coordinate system is based on a sphere rather than projected (flat), cell sizes will not have equal length and width values.

The cell sizes chosen the analysis are 0.002 degrees. For the study are as below on the ground measurements for each cell are as follows [1] :

- Latitude of bottom-left corner of study area: 53.2678
- Length of a Degree of Latitude: 111,291m
- Length of a Degree of Longitude: 66,721m
- Using a raster cell size of 0.002 of a degree
  - Cell height of 222.6m
  - Cell width of 134m

2.2. Modelling

The abrasion modelling requires several input layers. Each layer has an effect on abrasion, which is best solved using a raster-based approach.

This has the following advantages:

- Weightings can be created for each cell
- Sensitive data can be anonymised, and its resolution reduced to allow for publishing

An alternative (or additional) option is to display maps with the data. Looking at data layered on top of each other will show where give a visual indication of the cumulative effects.

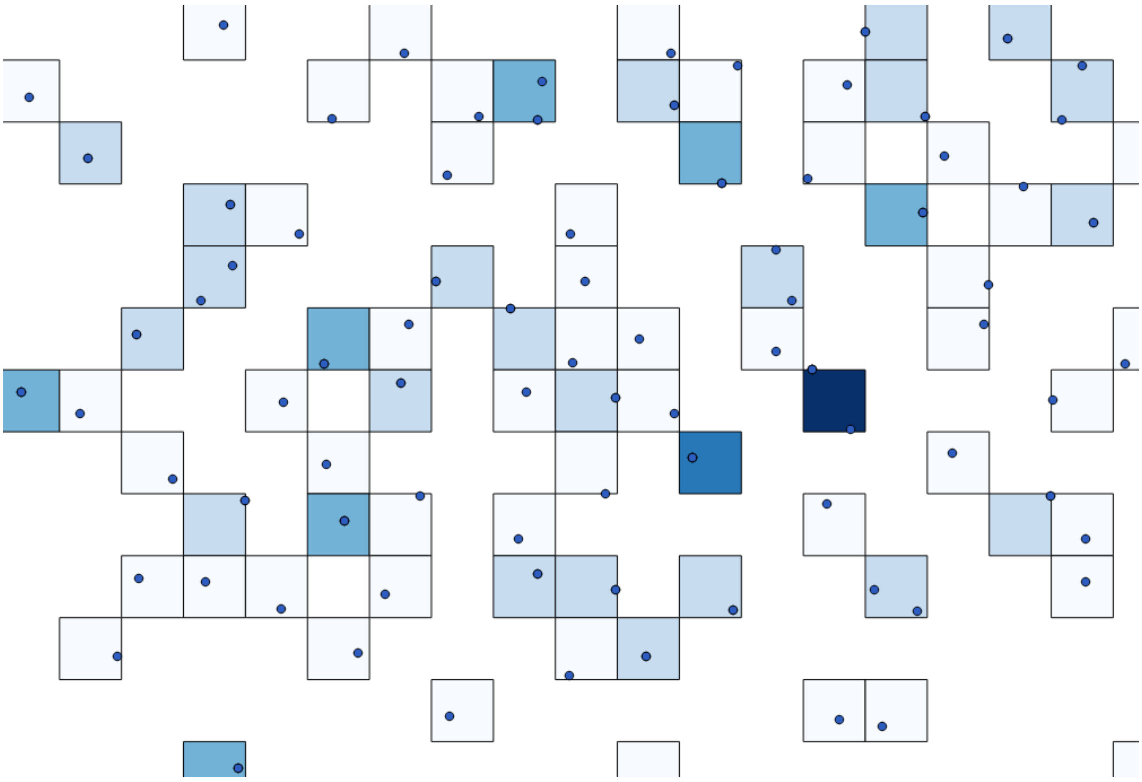
2.3. Software

Several different options for running the raster analysis were assessed. The criteria for selecting the software were:

- Needs to be easy for MI staff to adjust model parameters and rerun
- Should be repeatable (and therefore scriptable)

2.3.1. QGIS

- Lacks a tool to calculate number of points in a grid cell. Workarounds involve creating vector grids, running a spatial join, and converting the results to a raster, but this approach is not stable or fast when working with the study area



2.3.2. GRASS

https://grasswiki.osgeo.org/wiki/Count_points_in_raster_cells It is also possible to use the v.to.rast.* GRASS modules v.to.rast - Converts (rasterize) a vector map into a raster map.

https://gis.stackexchange.com/questions/54949/how-to-evaluate-raster-calculator-expressions-from-the-console

https://gis.stackexchange.com/questions/206004/number-inputs-as-raster-calculator-variables-in-qgis-modeler

2.3.3. ArcGIS

- A drawback is that *"the Raster Calculator tool is intended for use in the application only as a GP tool dialog box or in ModelBuilder. It is not intended for use in scripting and is not available in the ArcPy Spatial Analyst module"*. For this reason GDAL may be used to run the final raster calculations.

2.3.4. GDAL

2.4. Study Area

```
gdal_rasterize -a ID -ot Int16 -of GTiff -tr 0.002 0.002 -a_nodata 0 -co
COMPRESS=DEFLATE -co PREDICTOR=1 -co ZLEVEL=6

-l study_area C:/Data/MI/study_area.shp "[temporary file]"
```

2.5. Model

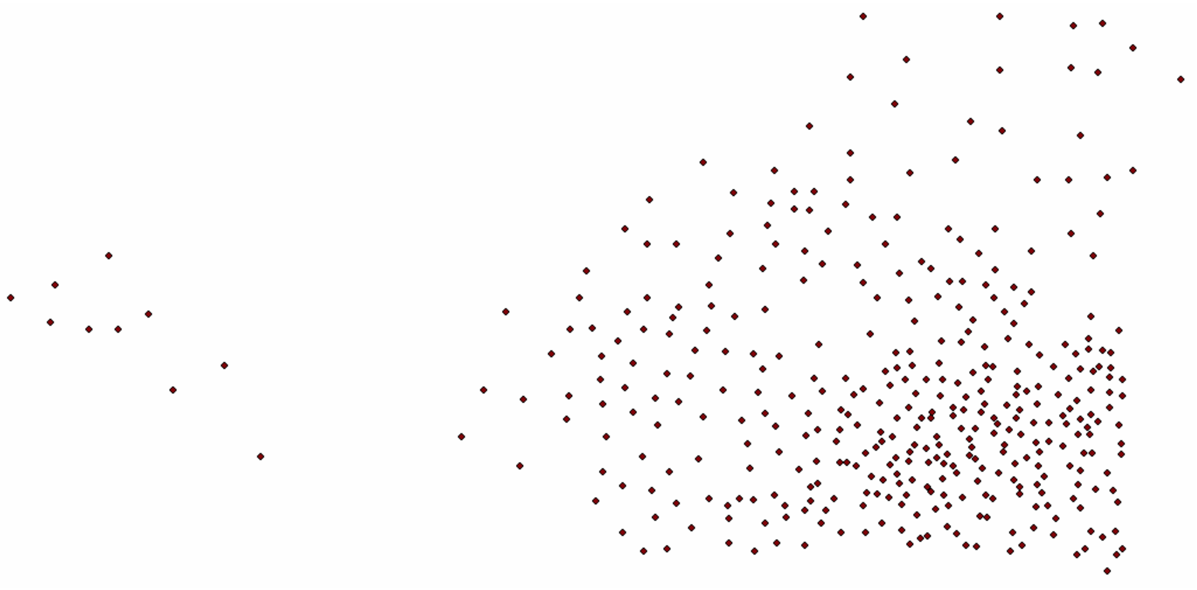
The model uses a number of tools from the ArcGIS Toolbox. It also requires the Spatial Analyst Extension.

Cell values can either be Yes/No to indicate a presence, or they can contain a number from within a range e.g. to indicate frequency.

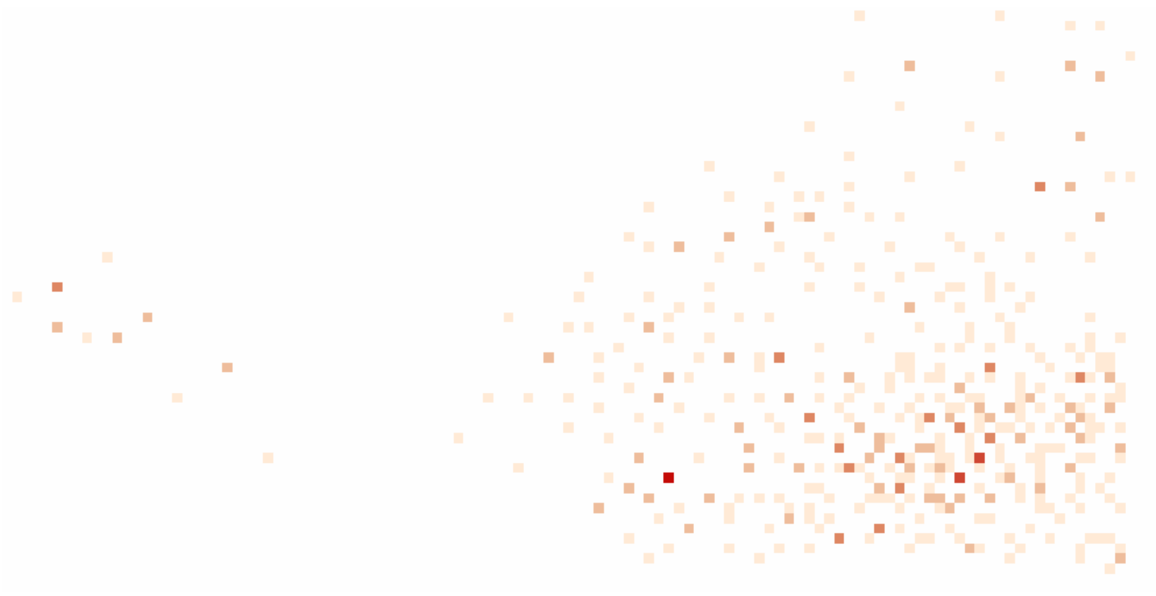
2.5.1. Point to Raster (Conversion)

Licensing Information - Requires Spatial Analyst or 3D Analyst, or an ArcInfo license.

This tool takes an input point dataset:



And creates a new raster grid with each cell containing the count of points within the region:



2.5.2. Point Density

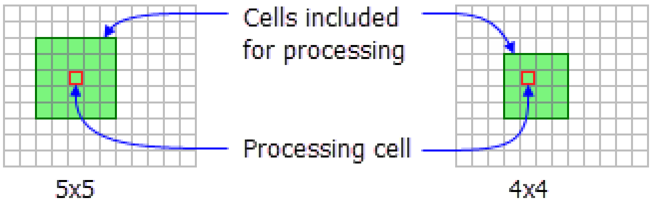
An alternative to the Point to Raster tool above is the Point Density tool [2]. This calculates a magnitude-per-unit area from point features that fall within a neighbourhood around each cell. See the tool documentation for details

For further details on point density see [3].

Note the values on the output raster will always be floating point.

An example of the output for the sample data above is shown below:

It was decided this approach gives a more accurate output. The input fishing points represent boats of a certain speed, but the abrasion damage is a continuous line between these points.



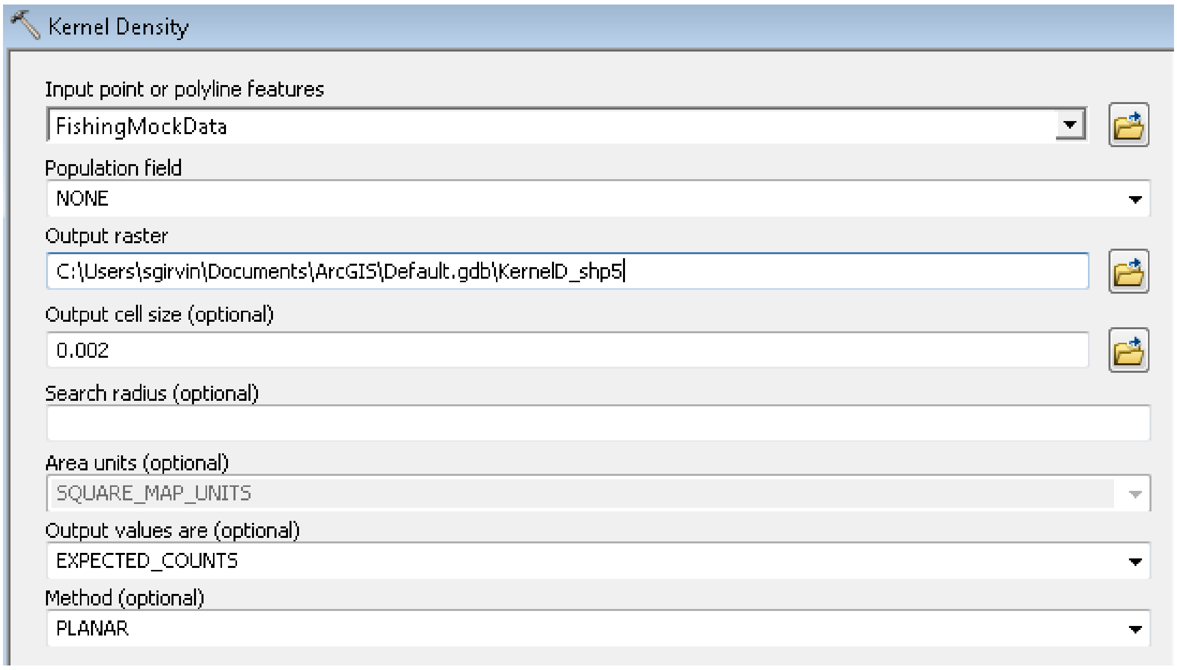
A rectangular neighbourhood will be used with a cell size of 3x3. [4]

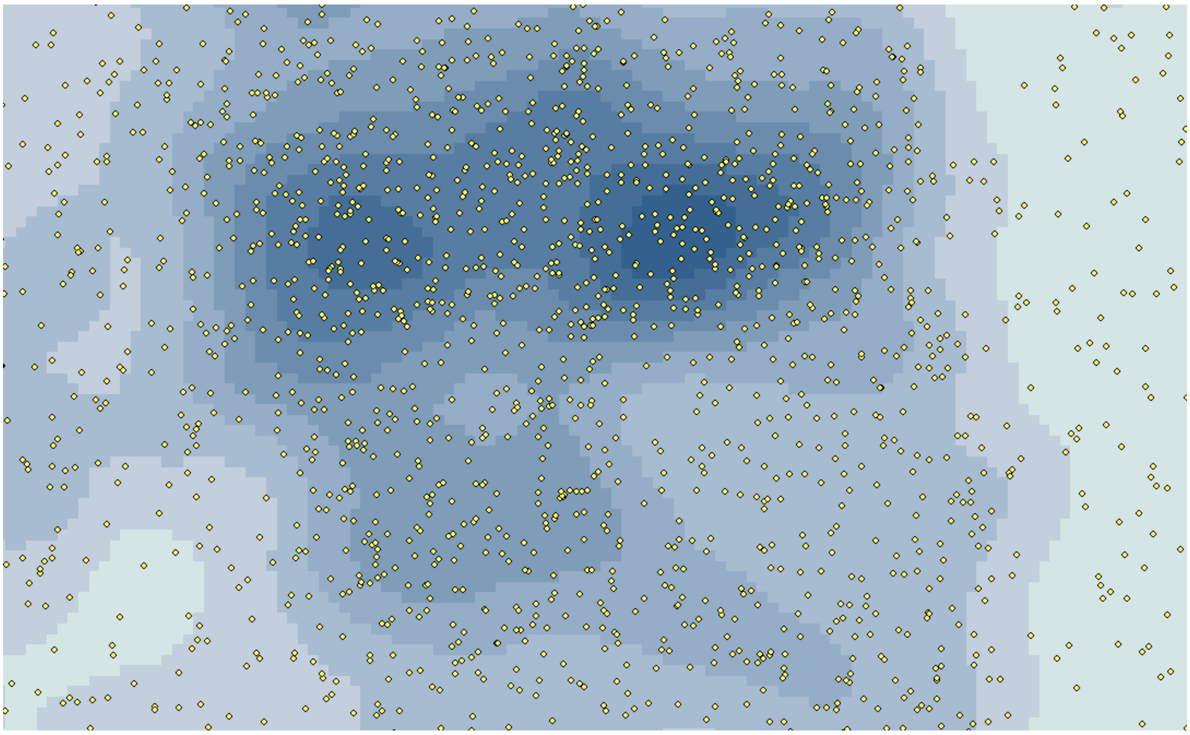
```
arcpy.gp.PointDensity_sa("FishingMockData", "NONE",
"D:/Data/MI/PointDe_shp11", "0.002", "Rectangle 3 3 CELL",
"SQUARE_MAP_UNITS")
```

When added together, the population values of the cells equal the sum of the population of the original point layer. [5] In a simple density calculation, points that fall within the search area are summed, and then divided by the search area size to get each cell's density value. [6]

2.5.3. Kernel Point Density

See https://desktop.arcgis.com/en/arcmap/latest/tools/spatial-analyst-toolbox/kernel-density.htm





Can't set method to Geodesic Kernel density cannot set error 000864 Works in Python however! And can then run and reopen.

```
arcpy.gp.KernelDensity_sa("FishingMockData", "NONE",
"D:/Data/MI/KernelD_shp5", "0.002", "", "SQUARE_MAP_UNITS",
"EXPECTED_COUNTS", "PLANAR")

# works in command prompt!

arcpy.gp.KernelDensity_sa("FishingMockData", "NONE",
"D:/Data/MI/KernelD_shp5", "0.002", "", "SQUARE_MAP_UNITS",
"EXPECTED_COUNTS", "GEODESIC")
```

2.6. Raster Calculator

Once all data has been prepared the Raster Calculator can be used to produce a new raster based on the values and weightings of the inputs.

Map Algebra - http://desktop.arcgis.com/en/arcmap/latest/extensions/spatial-analyst/map-algebra/a-quick-tour-of-using-map-algebra.htm

To open the calculator use Spatial Analyst Tools > Map Algebra > Raster Calculator

The Python code generated by running the tools manually is below:

```
# Replace a layer/table view name with a path to a dataset (which can be a
layer file) or create the layer/table view within the script

# The following inputs are layers or table views: "FishingMockData"



arcpy.PointToRaster_conversion(in_features="FishingMockData",
value_field="FID",
out_rasterdataset="C:/Users/sgirvin/Documents/ArcGIS/Default.gdb/FishingMoc
kData_PointToRaste", cell_assignment="COUNT", priority_field="NONE",
cellsize="0.002")



# Replace a layer/table view name with a path to a dataset (which can be a
layer file) or create the layer/table view within the script

# The following inputs are layers or table views: "FishingMockData"

arcpy.gp.PointDensity_sa("FishingMockData", "NONE",
"C:/Users/sgirvin/Documents/ArcGIS/Default.gdb/PointDe_shp7b", "0.002",
"Circle 2 CELL", "SQUARE_MAP_UNITS")



arcpy.gp.RasterCalculator_sa(""""FishingMockData_PointToRaste" *
"PointDe_shp7b"""",
"C:/Users/sgirvin/Documents/ArcGIS/Default.gdb/rastercalc")

gdal_rasterize -a FREQ -te -6.64388 53.2678 -5.43035 54.2346 -tr 0.002
0.002 -l grid_fish_count C:/Data/MI/grid_fish_count.shp
C:/Data/MI/test_freq3.tif

gdal_rasterize -a NUMPOINTS -ot Float32 -of GTiff -tr 0.002 0.002 TFW=YES -
co COMPRESS=DEFLATE -co PREDICTOR=1 -co ZLEVEL=6 "te -6.64388 53.2678 -
5.43035 54.2346" -l grid_fish_count
C:\Users\SG\.qgis2\\processing\outputs\grid_fish_count.shp
"C:\Data\MI\fish.tif"

gdal_rasterize -a NUMPOINTS -ot Float32 -of GTiff -tr 0.002 0.002 -te -
```

```
6.64388 53.2678 -5.43035 54.2346
C:\Users\SG\.qgis2\processing\outputs\grid_fish_count.shp
"C:\Data\MI\fish.tif"
```

Footnotes

[1] Calculated using http://msi.nga.mil/msisitecontent/staticfiles/calculators/degree.html

[2] See https://gis.stackexchange.com/questions/14593/converting-points-to-raster-in-arcgis

[3] See https://desktop.arcgis.com/en/arcmap/latest/tools/spatial-analyst-toolbox/how-point-density-works.htm

[4] See https://desktop.arcgis.com/en/arcmap/latest/analyze/arcpy-spatial-analyst/nbrrectangle-class.htm

[5] See https://desktop.arcgis.com/en/arcmap/latest/tools/spatial-analyst-toolbox/understanding-density-analysis.htm

[6] See https://desktop.arcgis.com/en/arcmap/latest/tools/spatial-analyst-toolbox/differences-between-point-line-and-kernel-density.htm

PART 3

Marine Abrasion Modelling

3. Marine Abrasion Modelling

3.1. Previous Studies

From *"Further development of marine pressure datalayers and ensuring the socio-economic data and datalayers are developed for use in the planning of marine protected area networks (MB0106)"* **[1]**

Within the GIS, activities exerting the same pressure on the marine environment have been merged into a single pressure layer and a summary grid with a cell size of 0.05 by 0.05 degrees has been superimposed. The spatial resolution of the summary grid (pressure grid) is a compromise between the spatial resolution of the data used and the required computation time at the scale of the study area. A series of five pressure layers (abrasion, extraction, obstruction, siltation and smothering) were developed based on the proportion of a grid cell affected by the footprint and/or intensity of all the human activities exerting the same pressure. For the pressures abrasion and extraction the average pressure per grid cell was calculated for the respective period of time considered; see detailed description in Table 1.

The abrasion pressure layer is derived from the rasters of hours fished for the demersal trawled gears and dredges (OT, OTB, OTT, PTB, TBB, TBN, TX, DRB and HMD). Average values for 2006 and 2007 for each gear are summarised.

Output - a standardised version of the Abrasion pressure layer scaled to a range of 0 - 1 by means of a linear scale transformation: (value - min) / range where 0 indicates low levels of abrasion and 1 indicates high levels of abrasion.

See also:

- *HELCOM request on pressure from fishing activity (based on VMS/logbook data) in the HELCOM area relating to both seafloor integrity and management of HELCOM MPAs* **[2]**

3.2. Activity Weightings

These are based on the output of the ODEMM tool and when plotted in a graph look like reasonable values that can be backed up with evidence. They include both the total risk score and the impact risk score, either of these could be used. There is also the relative contribution of total risk and impact risk, i.e. relative to all other activities and pressures.

Abrasion	Total Risk	Impact Risk	TR relative contribution	IR relative contribution
Aquaculture	0.00000108744	0.0018624	0.000123336	0.003042541
Beam Trawling	0.000163956	0.27326	0.018595711	0.446415721
Bottom Trawling	0.0003648	0.608	0.041375218	0.993269261
Dredging	0.00002538	0.0423	0.002878572	0.069104095
Shipping	0.0000005448	0.000908	0.0000617906	0.001483369
Tourism/Recreation	0.00000205112	0.0040352	0.000232636	0.006592171

3.3. Habitat Weightings

The **Max** field is used to weight a habitat's susceptibility to abrasion.

Code	Max	Average	Name
1	0.0269449		Shallow sublittoral coarse sediment
2	0.2479	0.0270325	Shallow sublittoral mixed sediment
3	1	0.1012645	Shallow sublittoral mud
4	0.1221	0.0146625	Shallow sublittoral rock and biogenic reef
5	0.0099	0.0099	Shallow sublittoral sand
6	1	0.1012372	Shelf sublittoral mixed sediment
7	1	0.1002472	Shelf sublittoral mud
8	0.67	0.1693	Shelf sublittoral rock and biogenic reef
9	1	0.111385778	Shelf sublittoral sand

3.4. Datasets

Fishing dataset. This is a point dataset of boat locations and their fishing types.

- Beam Trawling
- Bottom Trawling
- Dredging
- Shipping - sipping lines. The dataset has a distance from the bottom of the boat to the sea floor.
- Aquaculture - this is a polygon dataset.

3.5. Calculations

Model Assumptions:

- Aquaculture cells will have a value of 1 or 0. Or do different types of aquaculture have different weightings?
- A habitat type is uniformly affected by all types of abrasion. E.g. if a habitat is not immune to one type of abrasion, but heavily affected by another. If this is the case then an overall abrasion value can be applied to all habitats to assess risk.
- Require weightings for a habitat. How at risk is each habitat to abrasion.
- Tourism and recreation will not be taken into account for this initial model.
- Is there a minimum frequency counts for fishing before it has an effect, or should any instance have an impact?
- Frequency of trawling - counts for a year, or pick totals for a month?

Model abbreviations:

- CV - Cell Value
- IR - Impact Risk

3.5.1. Formula for Overall Abrasion Score

- AquacultureCV - will be either 1 or 0 dependent on the presence of aquaculture.
- BeamTrawlingCV - will be a count of all recordings of this fishing type for a cell, for a particular year (in this case 2016 data).
- BottomTrawlingCV - as above.
- DredgingCV - as above.
- ShippingCV - count of all recordings of boats in this cell for a year? Only boats within a certain distance to the seabed are included?

```
(AquacultureIR * AquacultureCV) + (BeamTrawlingIR *
BeamTrawlingCV) + (BottomTrawlingIR * BottomTrawlingCV) +
(DredgingIR * DredgingCV) + (ShippingIR * ShippingCV)
```

Example cell calculation, where there is no aquaculture, 2 instances of beam trawling, 0 instances of bottom trawling, 1 instance of dredging, and 3 recordings of shipping:

```
(0.0018624 * 0) + (0.27326 * 2) + (0.608 * 0) + (0.0423 * 1) +
(0.000908 * 3)

= 0 + 0.54652 + 0 + 0.0423 + 0.002724

= 0.591544
```

3.5.2. Formula for Habitat Impact

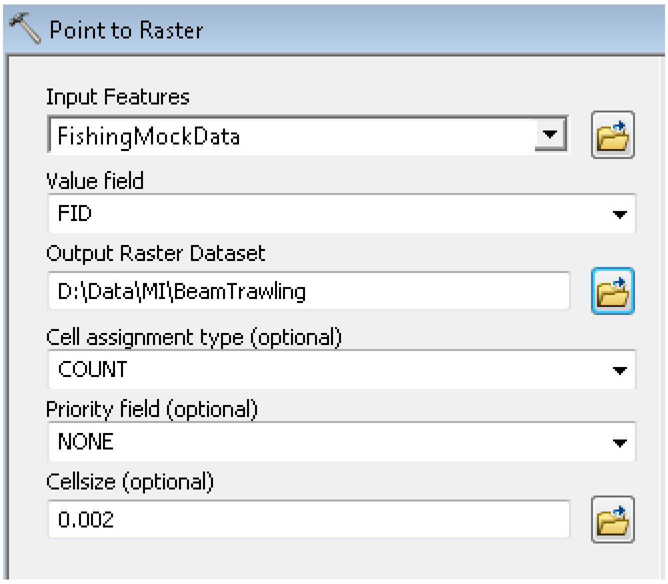
Once a new raster containing cell values with overall abrasion values has been created, this can be used with a raster grid of habitats to calculate which areas are at risk. For example a high-risk habitat with no abrasion activities won't be at risk.

The habitat grid could contain a cell value with its risk weighting, then a simple multiplication could be used as below:

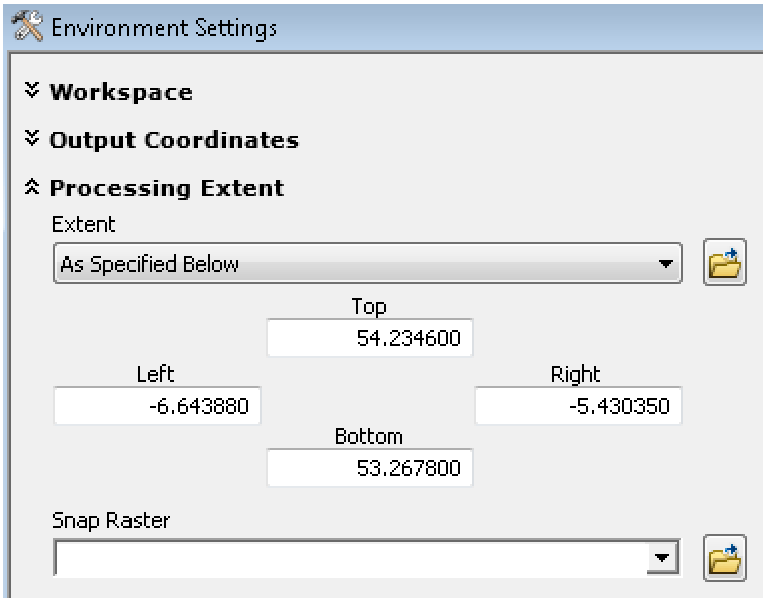
```
(AbrasionCV * HabitatCV)
```

3.6. Model Implementation

- All outputs should be in standalone GeoTIFF files (rather than in FileGeodatabases). This makes them easier to move between different tools and systems.
- All raster should be created using the same extents.

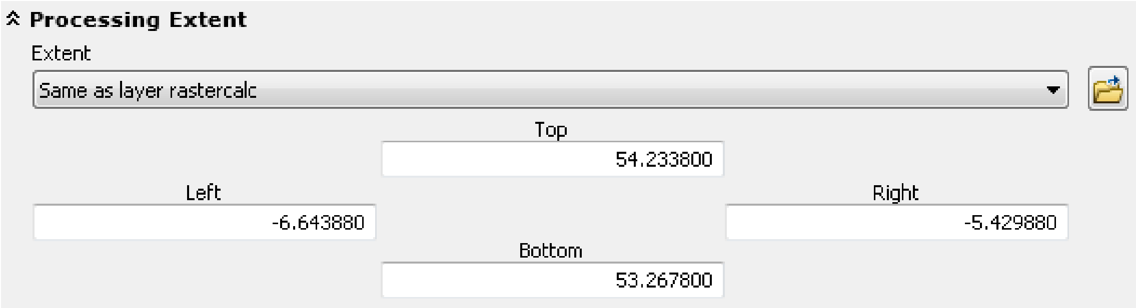


To set the Extent, click the Environment button and set the Process Extent options as below:



The extent values are: -6.64388 53.2678 -5.43035 54.2346 Once you have set these for one output you can then set any future tools to use the extent of this layer:

Cell size can also be set in the environment. See [3].

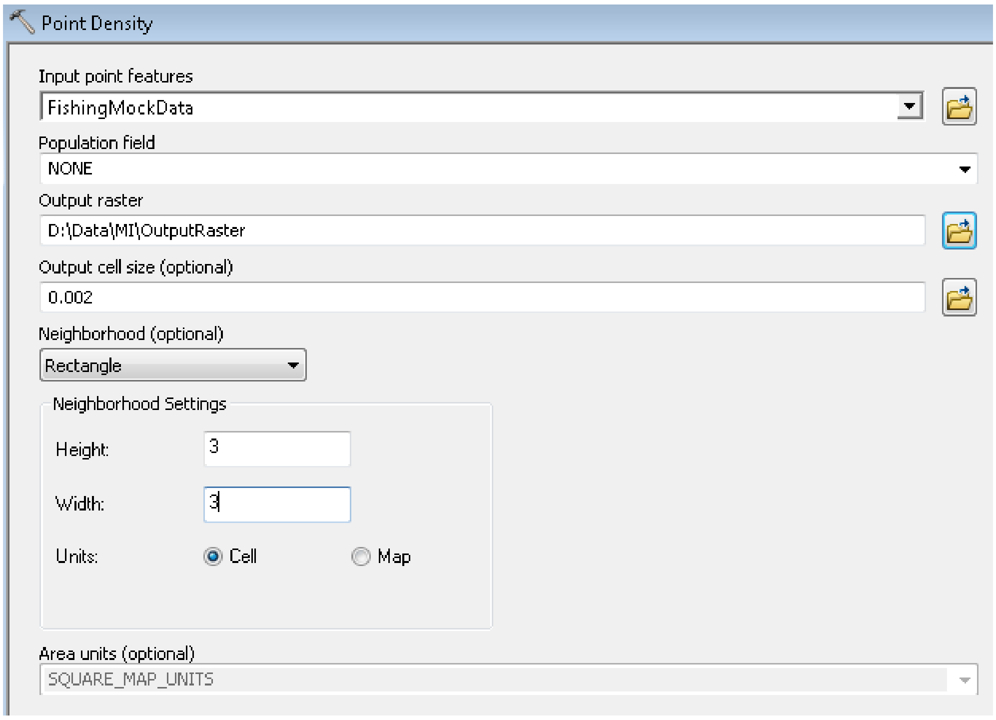


Steps:

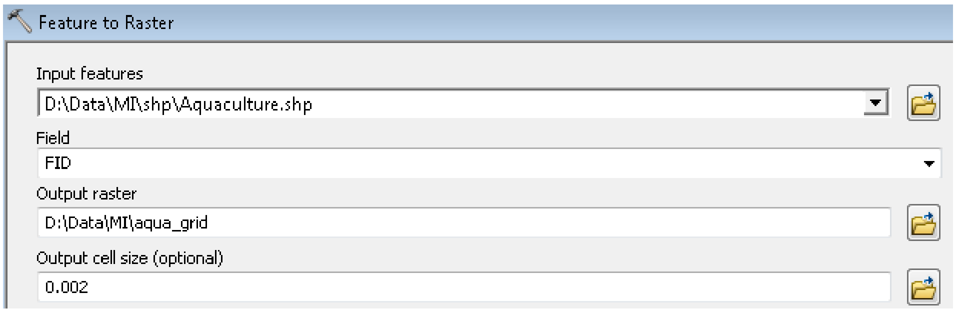
1. Take input fishing dataset (point shapefile)

2. Select all points in turn for each of the three types causing abrasion (Beam Trawling, Bottom Trawling, Dredging)

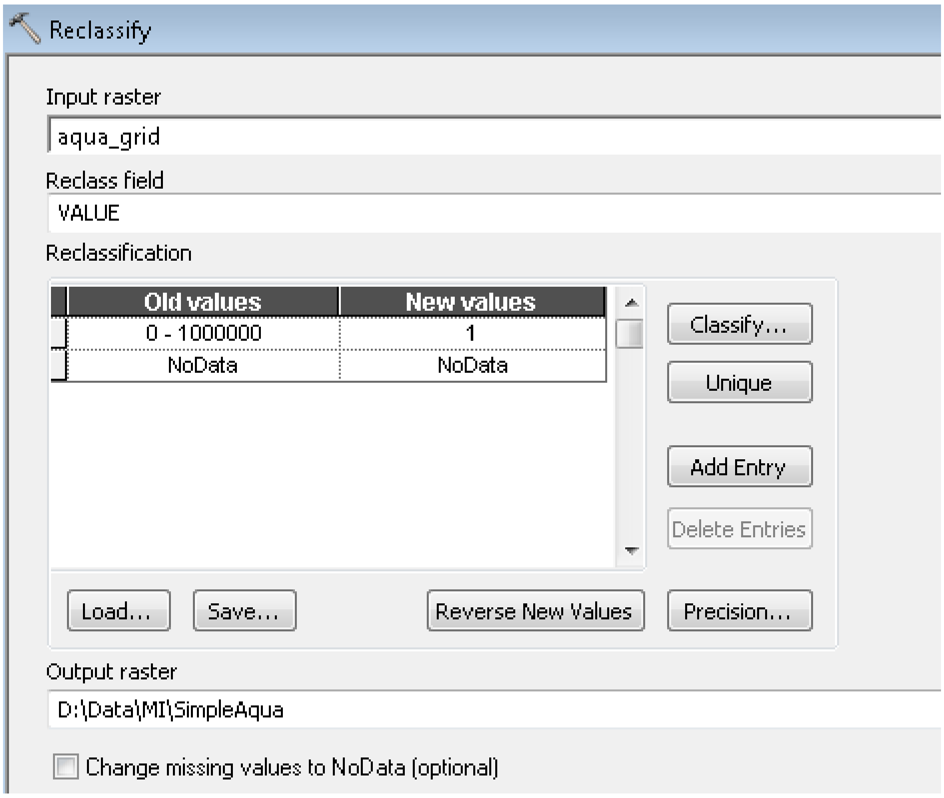
3. Run the Point to Raster tool for each of these, making sure the extent is set to the study area. **Alternative** use the Spatial Analyst Tools > Density > Point Density Tool



4. Create a new Aquaculture raster using the Feature to Raster tool found in ArcToolbox >> Conversion Tools >> To Raster >> Feature to Raster. The FID can be used as the cell values aren't important, just whether or not aquaculture is present.



5. Next we can Reclassify the Aquaculture grid to simply contain a 1 or 0 value to indicate presence:



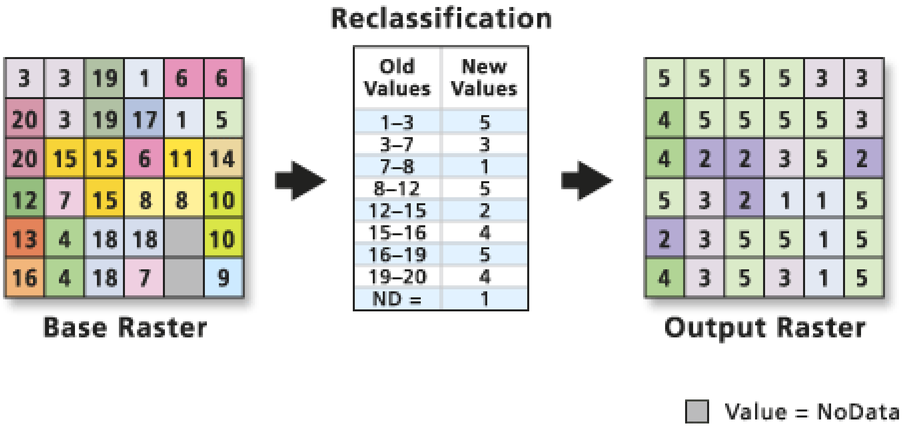
6. Create a new Shipping raster using the above tool and settings. Again the field used is not currently important. We can reclassify this as above.

7. Create a new Habitat raster using the Polygon to Raster tool found in ArcToolbox >> Conversion Tools >> To Raster >> Polygon to Raster. In this case the cell values are important - it needs to be set to the habitat code/id so this can be used in the calculations.

At the end of the process there should be the following GeoTiffs, all with the same cell size and extent:

- BeamTrawling.tif
- BottomTrawling.tif
- Dredging.tif
- Aquaculture.tif
- AquaculturePresence.tif - as above but reclassified. There may be a limit on characters for dataset names so we may need to choose a new convention.
- Shipping.tif
- ShippingPresence.tif - as above but reclassified
- Habitats.tif

Note - we may want to save any reclassification of habitat types in a table to then use the Reclass by Table tool.



Footnotes

[1] See Further development of marine pressure datalayers and ensuring the socio-economic data and datalayers are developed for use in the planning of marine protected area networks (MB0106) (mb0106_9436_FRP.pdf). Related documents at http://sciencesearch.defra.gov.uk/Default.aspx?Menu=Menu&Module=More&Location=None&Completed=0&ProjectID=16415

[2] https://www.ices.dk/sites/pub/Publication%20Reports/Advice/2015/Special_Requests/HELCOM_pressure_from_fishing_activity.pdf

[3] https://desktop.arcgis.com/en/arcmap/latest/tools/environments/cell-size.htm

PART 4 Abrasion Pressures Script

4. Abrasion Pressures Script

4.1. Setup

1. Create a folder with all the required shapefile inputs.
2. Copy the (Python script below) and save as abrasion.py to your PC (any location is fine).
3. Open the script in IDLE.
4. Update the ROOT_FOLDER path in the script.
5. Press F5 to run.

The script below assumes a folder structure as follows:

- Root folder
  - /output - a folder to store all the outputs generated by the script - this will automatically be created by the script if it does not already exist. This folder can be safely deleted.

The settings below can also be modified. It is recommended the conventions in the script are used for naming the shapefiles. In this case only the ROOT_FOLDER parameter should need updating to run the script.

4.1.1. Input Data

This section points to the names of the shapefiles required to run the script. They should be stored in the ROOT_FOLDER. Note the *r* is added before the ROOT_FOLDER path (indicating a raw string), so that the backslashes in the path are read correctly by Python.

All input shapefiles should be in the WGS84 projection (EPSG:4326).

```
EXTENT = '-6.574271 53.257102 -5.360746 54.22389'

ROOT_FOLDER = r'D:\Data\MI\Abrasion'
```

```
FISHING_TYPE_FIELD = 'gear_group' # the field name containing the fishing types (e.g. 'Dredging', 'Beam Trawl', 'Bottom Trawls')

FISHING_SHP = 'FishingData.shp' # the shapefile containing the fishing points

AQUACLUTURE_SHP = 'Aquaculture.shp' # the shapefile of Aquaculture polygons

SHIPPING_SHP = 'Shipping.shp' # the shapefile of shipping points

HABITAT_SHP = 'Habitats_StudyArea.shp' # shapefile of Marine habitats

HABITAT_CODE_FIELD = 'Code' # field containing the habitat type ID, used in the mappings file



# create a dictionary to store rasters and their associated settings

# key names will be used for output tiffs

# impact settings can be changed below



RASTERS = {

    'bottom_trawl': {'impact_risk': 0.608, 'field_val': 'Bottom trawls', 'src': FISHING_SHP}, # Note - there are also 'Bottom Trawl' records in the shapefile

    'beam_trawl': {'impact_risk': 0.27326, 'field_val': 'Beam trawl', 'src': FISHING_SHP},

    'dredging': {'impact_risk': 0.0423, 'field_val': 'Dredge', 'src': FISHING_SHP},

    'aquaculture': {'impact_risk': 0.0018624, 'src': AQUACLUTURE_SHP},

    'shipping': {'impact_risk': 0.000908, 'src': SHIPPING_SHP}

}
```

```
HABITAT_WEIGHTINGS = [
    ['Code', 'Max_', 'Average', 'Name'], # Max is a reserved word so add an
underscore
    [1, 0.0269449, None, 'Shallow sublittoral coarse sediment'],
    [2, 0.2479, 0.0270325, 'Shallow sublittoral mixed sediment'],
    [3, 1, 0.1012645, 'Shallow sublittoral mud'],
    [4, 0.1221, 0.0146625, 'Shallow sublittoral rock and biogenic reef'],
    [5, 0.0099, 0.0099, 'Shallow sublittoral sand'],
    [6, 1, 0.1012372, 'Shelf sublittoral mixed sediment'],
    [7, 1, 0.1002472, 'Shelf sublittoral mud'],
    [8, 0.67, 0.1693, 'Shelf sublittoral rock and biogenic reef'],
    [9, 1, 0.111385778, 'Shelf sublittoral sand']
]
```

4.1.2. Environment Settings

The variables below can be changed to run the script on a different study area, or to use a different cell size for the analysis.

```
# the extent of the study area in long/lat  - min_long, min_lat, max_long,
max_lat
arcpy.env.extent = EXTENT
arcpy.env.cellSize = 0.002 # the cell size in degrees
```

4.2. Script Outputs

- The script automatically replaces any existing datasets in the output folder (unless these files are in use - see Possible Errors section below).
- All intermediate rasters and shapefiles are saved to the output folder, to help verify results

Raster outputs:

- abrasion.tif - overall abrasion pressure, created from all input rasters
- abrasion_presence.tif - a simple grid that displays the sum of abrasion pressures in each cell (a count of features)
- aquaculture_presence.tif - a raster containing 1s and 0s to indicate aquaculture presence
- beam_trawl_kernel_density.tif - a raster showing Beam Trawl activity using kernel density
- beam_trawl_point_density.tif - a raster showing Beam Trawl activity using point density
- beam_trawl_presence.tif - a raster showing 1s and 0s to indicate Beam Trawl activity
- bottom_trawl_kernel_density.tif - a raster showing Bottom Trawl activity using kernel density
- bottom_trawl_point_density.tif - a raster showing Bottom Trawl activity using point density
- bottom_trawl_presence.tif - a raster showing 1s and 0s to indicate Bottom Trawl activity
- dredging_trawl_kernel_density.tif - a raster showing Dredging activity using kernel density
- dredging_trawl_point_density.tif - a raster showing Dredging activity using point density

- dredging_trawl_presence.tif - a raster showing 1s and 0s to indicate Dredging activity
- habitat_weightings.tif - a raster showing Habitat abrastion weightings
- **habitat_risk.tif** - raster showing total abrasion impact taking into account habitats - this is the final results grid
- shipping_trawl_kernel_density.tif - a raster showing Shipping activity using kernel density
- shipping_trawl_point_density.tif - a raster showing Shipping activity using point density
- shipping_trawl_presence.tif - a raster showing 1s and 0s to indicate Shipping activity

Shapefile outputs:

- beam_trawl.shp - Beam Trawl points
- bottom_trawl.shp - Bottom Trawl points
- dredging.shp - Dredging points
- habitat_weightings.shp - the Habitats shapefile joined to the CSV of weightings

When run successfully you should see output similar to below:

```
Creating bottom_trawl.shp

Creating beam_trawl.shp

Creating dredging.shp

Creating D:\Data\MI\Abrasion\output\shipping_presence.tif

Creating D:\Data\MI\Abrasion\output\dredging_presence.tif

Creating D:\Data\MI\Abrasion\output\beam_trawl_presence.tif

Creating D:\Data\MI\Abrasion\output\aquaculture_presence.tif
```

```
Creating D:\Data\MI\Abrasion\output\bottom_trawl_presence.tif

Creating D:\Data\MI\Abrasion\output\shipping_kernel_density.tif

Creating D:\Data\MI\Abrasion\output\shipping_point_density.tif

Creating D:\Data\MI\Abrasion\output\dredging_kernel_density.tif

Creating D:\Data\MI\Abrasion\output\dredging_point_density.tif

Creating D:\Data\MI\Abrasion\output\beam_trawl_kernel_density.tif

Creating D:\Data\MI\Abrasion\output\beam_trawl_point_density.tif

Creating D:\Data\MI\Abrasion\output\bottom_trawl_kernel_density.tif

Creating D:\Data\MI\Abrasion\output\bottom_trawl_point_density.tif

Creating D:\Data\MI\Abrasion\output\abrasion_presence.tif

Creating D:\Data\MI\Abrasion\output\abrasion.tif

Creating habitat_weightings.shp

Creating D:\Data\MI\Abrasion\output\habitat_weightings.tif

Creating D:\Data\MI\Abrasion\output\habitat_risk.tif

Done!
```

4.3. Possible Errors

```
ExecuteError: ERROR 999999: Error executing function.

Failed to delete raster dataset

Failed to execute (CopyRaster).
```

This indicates that a new raster cannot be created in the output folder, as there is already a file with the same name that cannot be automatically overwritten. Check that the raster is not opened in ArcMap and if this is not the case try and delete all files in the output folder manually.

```
ExecuteError: ERROR 000358: Invalid expression

Failed to execute (SelectLayerByAttribute).
```

This error is likely caused by using a non-existent field name for the FISHING_TYPE_FIELD setting. Check this field exists in the FISHING_SHP shapefile.

```
ExecuteError: ERROR 000875: Output raster: aquaculture_values.tif's
workspace is an invalid output workspace.

ERROR 000581: Invalid parameters.

Failed to execute (FeatureToRaster).
```

This error normally indicates the user running the script does not have permission to save files to the ROOT_FOLDER.

```
arcpy.CopyRaster_management(raster, output_name, format="TIFF")

TypeError: CopyRaster() got an unexpected keyword argument 'format'
```

This occurs as the format option is not available in ArcGIS 10.3 or earlier. It is recommended to upgrade to ArcGIS 10.4 or higher.

```
ExecuteError: ERROR 000358: Invalid expression

Failed to execute (SelectLayerByAttribute).

import arcpy error
```

If you have multiple versions of Python on your machine then opening IDLE may open a Python installation without arcpy. To ensure you are using the correct IDLE you can open it from the command line as follows (for 10.4):

```
C:\Python27\ArcGIS10.4\pythonw.exe
"C:\Python27\ArcGIS10.4\Lib\idlelib\idle.pyw"
```

4.4. Script

```
"""

Tested with ArcGIS 10.3 and 10.4 / Python 2.7.10

The script requires an ArcGIS Spatial Analyst license



Author: Seth Girvin - sgirvin@compass.ie

Last Modified: 29/05/2017

"""

import arcpy

from arcpy.sa import KernelDensity, PointDensity, Raster, Reclassify,
RemapRange, Con, IsNull, NbrRectangle

import os

import csv



# check out the ArcGIS Spatial Analyst extension license

arcpy.CheckOutExtension('Spatial')



# input dataset settings
```

```
EXTENT = '-6.574271 53.257102 -5.360746 54.22389'

ROOT_FOLDER = r'D:\Data\MI\Abrasion'

FISHING_TYPE_FIELD = 'gear_group' # the field name containing the fishing
types (e.g. 'Dredging', 'Beam Trawl', 'Bottom Trawls')

FISHING_SHP = 'FishingData.shp' # the shapefile containing the fishing
points

AQUACLUTURE_SHP = 'Aquaculture.shp' # the shapefile of Aquaculture polygons

SHIPPING_SHP = 'Shipping.shp' # the shapefile of shipping points

HABITAT_SHP = 'Habitats_StudyArea.shp' # shapefile of Marine habitats

HABITAT_CODE_FIELD = 'Code' # field containing the habitat type ID, used in
the mappings file



# create a dictionary to store rasters and their associated settings

# key names will be used for output tiffs

# impact settings can be changed below



RASTERS = {

    'bottom_trawl': {'impact_risk': 0.608, 'field_val': 'Bottom trawls',
'src': FISHING_SHP}, # Note - there are also 'Bottom Trawl' records in the
shapefile

    'beam_trawl': {'impact_risk': 0.27326, 'field_val': 'Beam trawl',
'src': FISHING_SHP},

    'dredging': {'impact_risk': 0.0423, 'field_val': 'Dredge', 'src':
FISHING_SHP},

    'aquaculture': {'impact_risk': 0.0018624, 'src': AQUACLUTURE_SHP},
```

```
    'shipping': {'impact_risk': 0.000908, 'src': SHIPPING_SHP}

}



HABITAT_WEIGHTINGS = [

    ['Code', 'Max_', 'Average', 'Name'], # Max is a reserved word so add an
underscore

    [1, 0.0269449, None, 'Shallow sublittoral coarse sediment'],

    [2, 0.2479, 0.0270325, 'Shallow sublittoral mixed sediment'],

    [3, 1, 0.1012645, 'Shallow sublittoral mud'],

    [4, 0.1221, 0.0146625, 'Shallow sublittoral rock and biogenic reef'],

    [5, 0.0099, 0.0099, 'Shallow sublittoral sand'],

    [6, 1, 0.1012372, 'Shelf sublittoral mixed sediment'],

    [7, 1, 0.1002472, 'Shelf sublittoral mud'],

    [8, 0.67, 0.1693, 'Shelf sublittoral rock and biogenic reef'],

    [9, 1, 0.111385778, 'Shelf sublittoral sand']

]

# end of input dataset settings



# environment settings

# the extent of the study area in long/lat  - min_long, min_lat, max_long,
max_lat

arcpy.env.extent = EXTENT
```

```
arcpy.env.cellSize = 0.002 # the cell size in degrees

# end of environment settings



arcpy.env.workspace = os.path.join(ROOT_FOLDER, 'output')  # the folder
containing all input datasets (and all rasters will be created here)

arcpy.env.overwriteOutput = True # always overwrite any datasets previously
created by the script



def save_raster(raster, output_name):

    """

    Save a raster dataset as a .tif file

    """

    output_name += '.tif'

    print('Creating %s' % os.path.join(arcpy.env.workspace, output_name))

    #arcpy.CopyRaster_management(raster, output_name, format='TIFF') # 10.4
only

    arcpy.CopyRaster_management(raster, output_name) # for ArcGIS 10.3

    return output_name



def save_shapefile(layer_name, output_name):

    """

    Save a layer to a shapefile

    # http://desktop.arcgis.com/en/arcmap/latest/tools/data-management-
```

```
toolbox/copy-features.htm

    """

    out_fc = output_name + '.shp'

    print("Creating %s" % out_fc)

    arcpy.env.qualifiedFieldNames = False # do not save the table name as
part of the field name for the join

    arcpy.CopyFeatures_management(layer_name, out_fc)

    arcpy.env.qualifiedFieldNames = True

    return out_fc



def save_csv(lst, output_name):

    """

    Save a list of values to a CSV file

    """

    with open(output_name, 'wb') as f:

        wr = csv.writer(f, quoting=csv.QUOTE_NONE)

        wr.writerow(lst[0]) # header

        wr.writerows(lst[1:]) # records

    return output_name



def create_kernel_density_raster(layer_name, output_name):

    """
```

```
    Create a Kernel Density raster from the input layer, and save to a .tif

    See https://desktop.arcgis.com/en/arcmap/latest/tools/spatial-analyst-
toolbox/kernel-density.htm for tool details

    """



    out_raster = KernelDensity(layer_name, population_field='NONE',
area_unit_scale_factor='SQUARE_MAP_UNITS',

                                     out_cell_values='EXPECTED_COUNTS',
method='PLANAR') #cannot use GEODESIC setting



    return save_raster(out_raster, output_name)



def create_point_density_raster(layer_name, output_name):

    """

    Create a Point Density raster from the input layer, and save to a .tif

    https://desktop.arcgis.com/en/arcmap/latest/tools/spatial-analyst-
toolbox/point-density.htm

    """



    neighborhood = NbrRectangle(3, 3, "CELL")



    out_raster = PointDensity(layer_name, population_field="NONE",
neighborhood=neighborhood, area_unit_scale_factor="SQUARE_MAP_UNITS")

    return save_raster(out_raster, output_name)
```

```
def create_filtered_fc(output_name, original_fc, filter_field,
filter_field_val):

    """

    Select relevant features from a feature class, and save to a new one

    See http://desktop.arcgis.com/en/arcmap/latest/tools/data-management-
toolbox/select-layer-by-attribute.htm for selection tool details

    """



    # Make a layer from the feature class

    layer_name = 'full_lyr'

    arcpy.MakeFeatureLayer_management(original_fc, layer_name)



    where_clause = "\"%s\" = '%s'" % (filter_field, filter_field_val)



    try:

        arcpy.SelectLayerByAttribute_management(layer_name,
selection_type='NEW_SELECTION', where_clause=where_clause)

    except Exception as ex:

        print("Is the following WHERE clause correct? %s" % where_clause)

        raise
```

```
    return save_shapefile(layer_name, output_name)



def calculate_habitat_risk(abrasion_raster, habitat_risk_raster):



    # created a combined raster using Map Algebra

    combined_raster = (Raster(abrasion_raster) *
Raster(habitat_risk_raster))

    return save_raster(combined_raster, 'habitat_risk')



def create_abrasion_habitat_grid(habitat_shp, habitat_weightings,
habitat_code_field):

    """

    weighting_field is the name of the field to use from the CSV file

    """



    join_field = "Code"

    weighting_field = "Max_"



    habitat_weightings_csv = os.path.join(arcpy.env.workspace,
"habitat_weightings.csv")



    save_csv(habitat_weightings, habitat_weightings_csv)
```

```
    # Make a layer from the feature class

    layer_name = 'habitat_lyr'

    arcpy.MakeFeatureLayer_management(habitat_shp, layer_name)



    # Join to the CSV file

    arcpy.AddJoin_management(in_layer_or_view=layer_name,
in_field=habitat_code_field,

                             join_table=habitat_weightings_csv,
join_field=join_field, join_type='KEEP_ALL')



    output_name = "habitat_weightings"

    out_fc = save_shapefile(layer_name, output_name)



    output_name += '.tif'

    # http://desktop.arcgis.com/en/arcmap/latest/tools/conversion-
toolbox/feature-to-raster.htm

    print('Creating %s' % os.path.join(arcpy.env.workspace, output_name))

    arcpy.FeatureToRaster_conversion(out_fc, weighting_field, output_name)

    return output_name



def check_rasters_exist(rasters, key):

    # check each pressure has an associated raster

    for k, v in rasters.items():
```

```
        if key in v:

            try:

                fn = os.path.join(arcpy.env.workspace, v[key])

                assert(os.path.isfile(fn))

            except AssertionError as ex:

                print('%s does not have a raster at %s' % (k, fn))


def calculate_combined_pressures(rasters):

    """

    Create an overall raster taking into account all the different

    abrasion pressures, and save this to a .tif file

    """



    # get the rasters as variables



    bt = rasters['bottom_trawl']

    bmt = rasters['beam_trawl']

    d = rasters['dredging']

    aq = rasters['aquaculture'] # for aquaculture simple 1/0 presence
values are used

    s = rasters['shipping']
```

```
    # created a combined raster using Map Algebra



    combined_raster = (

                      (Raster(bt['kernel_density_raster']) *
bt['impact_risk']) \

                    + (Raster(bmt['kernel_density_raster']) *
bmt['impact_risk']) \

                    + (Raster(d['kernel_density_raster']) *
d['impact_risk']) \

                    + (Raster(aq['presence_raster']) * aq['impact_risk']) \

                    + (Raster(s['kernel_density_raster']) *
s['impact_risk'])

                    )



    return save_raster(combined_raster, 'abrasion')



def calculate_combined_presence(rasters):

    """

    Create a simple overall raster based on if a feature is present (1) or
not (0)

    """



    # get the rasters as variables
```

```
    bt = rasters['bottom_trawl']

    bmt = rasters['beam_trawl']

    d = rasters['dredging']

    aq = rasters['aquaculture']

    s = rasters['shipping']



    # created a combined raster using Map Algebra



    combined_raster = (Raster(bt['presence_raster']) +
Raster(bmt['presence_raster']) + Raster(d['presence_raster']) \

                     + Raster(aq['presence_raster']) +
Raster(s['presence_raster']))



    return save_raster(combined_raster, 'abrasion_presence')



def create_present_absent_grid(shp, output_name):

    """

    Convert the polygon vector layer to a raster

    Then assign either a 1 or 0 to the raster dependent on if a feature is
present

    (0 no feature, 1 feature is present)

    """

    raster_output_name = output_name + '_presence.tif'
```

```
    # http://desktop.arcgis.com/en/arcmap/latest/tools/conversion-toolbox/feature-to-raster.htm

    arcpy.FeatureToRaster_conversion(shp, 'FID', raster_output_name)



    # conditional raster

    #https://desktop.arcgis.com/en/arcmap/latest/tools/spatial-analyst-toolbox/con-.htm

    presence_raster = Con(IsNull(raster_output_name), 0, 1)



    output_name = save_raster(presence_raster, output_name)



    # delete the intermediary raster (cells simply display FIDs) - this must be run after the presence_raster has been saved

    # or the script crashes

    arcpy.Delete_management(raster_output_name)



    return output_name



def check_folders(root):

    """

    Check the root folder exists, and create a new output subfolder

    if it does not already exist

    """
```

```
    try:
        assert(os.path.isdir(root))
    except AssertionError:
        print('The root folder %s does not exist' % root)
        raise

    # create output folder
    ws = arcpy.env.workspace
    if not os.path.isdir(ws):
        os.makedirs(ws)

def check_input(sf):
    """
    Check that the input shapefiles are in the root folder
    """
    root_folder = os.path.dirname(arcpy.env.workspace)
    sf = os.path.join(root_folder, sf)

    try:
        assert(os.path.isfile(sf))
    except AssertionError:
```

```
        print('The input shapefile %s does not exist' % sf)

        raise



    return sf



def main():



    root = ROOT_FOLDER

    check_folders(root)



    # set globals to locals so easier to read from a config file if 
necessary

    fishing_type_field = FISHING_TYPE_FIELD

    habitat_shp = check_input(HABITAT_SHP)

    habitat_code_field = HABITAT_CODE_FIELD

    habitat_weightings = HABITAT_WEIGHTINGS



    rasters = RASTERS

    # create new filtered feature classes for fishing types



    fishing_layers = ['bottom_trawl', 'beam_trawl', 'dredging']
```

```
    # update the source shapefile for each fishing type to be a filtered
shapefile

    for k in fishing_layers:

        fl = rasters[k]

        filtered_shp = create_filtered_fc(k, fl['src'], fishing_type_field,
fl['field_val'])

        fl['src'] = filtered_shp



    # create present/absent grids for all layers

    all_shps = []

    for k, v in rasters.items():

        v['presence_raster'] = create_present_absent_grid(v['src'], k +
'_presence')



    # create kernel density grids for all layers

    for k, v in rasters.items():

        if k != 'aquaculture': # no raster required for Aquaculture which
is a polygon layer (Kernel Density is for points/lines only)

            v['kernel_density_raster'] =
create_kernel_density_raster(v['src'], k + "_kernel_density")

            v['point_density_raster'] =
create_point_density_raster(v['src'], k + "_point_density")



    # validate all rasters are created
```

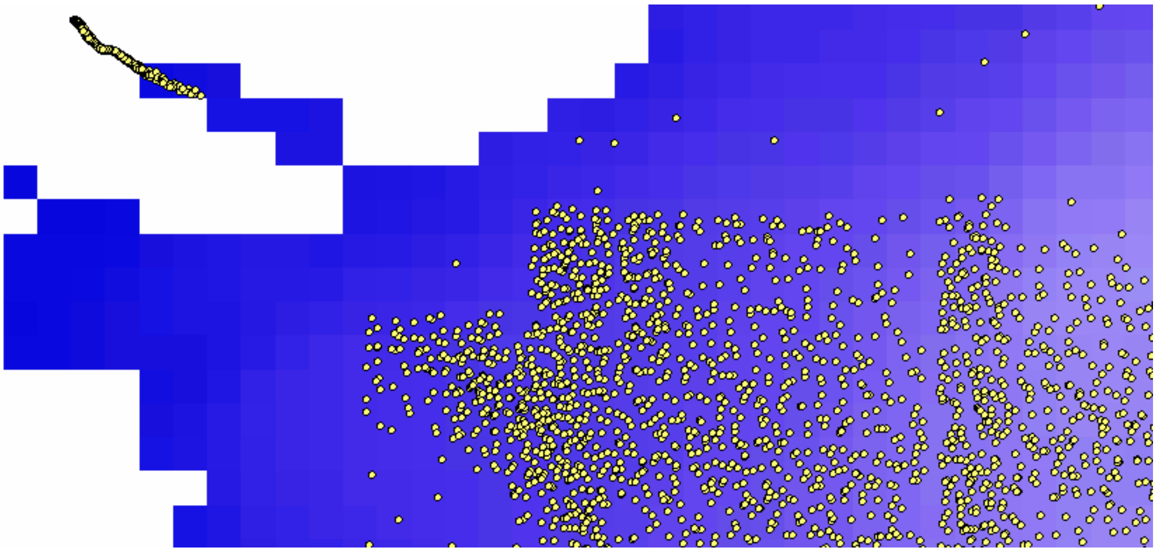
```
    check_rasters_exist(rasters, 'presence_raster')

    check_rasters_exist(rasters, 'kernel_density_raster')

    check_rasters_exist(rasters, 'point_density_raster')



    # create an overall presence grid

    calculate_combined_presence(rasters)



    # create an overall pressure grid

    abrasion_raster = calculate_combined_pressures(rasters)



    # create habitat risk grid

    habitat_grid = create_abrasion_habitat_grid(habitat_shp,
habitat_weightings, habitat_code_field)



    # combine with abrasion pressures to give actual risk

    calculate_habitat_risk(abrasion_raster, habitat_grid)



main()

print('Done!')
```

Part 4 Shipping Geoprocessing Script

4. Shipping Geoprocessing Script

4.1. Introduction

This script takes a shapefile of recorded shipping locations, extracts the seabed depth at these points, and then outputs a final set of points where the bottom of the boat is within 5m of the seabed. This dataset can then be used as part of the overall abrasion modelling.



4.2. Setup

1. Create a folder with all the required shapefile inputs.
2. Copy the (Python script below) and save as shipping_processing.py to your PC (any location is fine).
3. Open the script in IDLE.
4. Update the ROOT_FOLDER path in the script.
5. Press F5 to run.

The script below assumes a folder structure as follows:

- Root folder
    - /output - a folder to store all the outputs generated by the script - this will automatically be created by the script if it does not already exist. This folder can be safely deleted.

The settings below can also be modified. It is recommended the conventions in the script are used for naming the shapefiles. In this case only the ROOT_FOLDER parameter should need updating to run the script.

4.2.1. Input Data

This section points to the names of the shapefiles required to run the script. They should be stored in the ROOT_FOLDER. Note the *r* is added before the ROOT_FOLDER path (indicating a raw string), so that the backslashes in the path are read correctly by Python.

All input datasets should be in the WGS84 projection (EPSG:4326).

```
ROOT_FOLDER = r'D:\Data\MI\Shipping'

ALL_SHIPPPING_PTS = "All_Shipping.shp" # shapefile of shipping points - this must contain a "Draught" field

BATHYMETRY_TIF = "CelticSeasBathymetry1.tif" # raster with values representing sea depths
```

4.3. Script Outputs

Notes:

- The script automatically replaces any existing datasets in the output folder (unless these files are in use - see Possible Errors section below).
- All intermediate shapefiles are saved to the output folder, to help verify results

Shapefile outputs:

- shipping_depths_all.shp - all shipping points and their distance between the bottom of the boat to the seabed
- shipping_depths_shallow.shp - all recorded shipping points with the bottom of the boat within 5m of the seabed

When run successfully you should see output similar to below:

```
Creating D:\Data\MI\Shipping\output\shipping_depths_all.shp

Creating D:\Data\MI\Shipping\output\shipping_depths_shallow.shp
```

```
Done!
```

4.4. Possible Errors

```
ExecuteError: ERROR 999999: Error executing function.

Cannot acquire a schema lock because of an existing lock.

Failed to execute (ExtractValuesToPoints).
```

This indicates that a new raster cannot be created in the output folder, as there is already a file with the same name that cannot be automatically overwritten. Check that the raster is not opened in ArcMap and if this is not the case try and delete all files in the output folder manually.

4.5. Script

```
"""

This script takes a shapefile of recorded shipping locations, extracts the seabed depth at these points, and

then outputs a final set of points where the bottom of the boat is within 5m of the seabed.



Tested with ArcGIS 10.3 and 10.4 / Python 2.7.10

The script requires an ArcGIS Spatial Analyst license



Author: Seth Girvin - sgirvin@compass.ie

Last Modified: 09/06/2017

"""
```

```
import arcpy, os


# input dataset settings

ROOT_FOLDER = r'D:\Data\MI\Shipping'

ALL_SHIPPPING_PTS = "All_Shipping.shp" # shapefile of shipping points - this must contain a "Draught" field

BATHYMETRY_TIF = "CelticSeasBathymetry1.tif" # raster with values representing sea depths

# end of input dataset settings


# check out the ArcGIS Spatial Analyst extension license

arcpy.CheckOutExtension('Spatial')

arcpy.env.workspace = os.path.join(ROOT_FOLDER, 'output')

arcpy.env.overwriteOutput = True # always overwrite any datasets previously created by the script


def check_folders(root):

    """

    Check the root folder exists, and create a new output subfolder

    if it does not already exist

    """

    try:
```

```
        assert(os.path.isdir(root))

    except AssertionError:

        print('The root folder %s does not exist' % root)

        raise



    # create output folder

    ws = arcpy.env.workspace

    if not os.path.isdir(ws):

        os.makedirs(ws)



def check_input(sf):

    """

    Check that the input shapefiles are in the root folder

    """

    root_folder = os.path.dirname(arcpy.env.workspace)

    sf = os.path.join(root_folder, sf)



    try:

        assert(os.path.isfile(sf))

    except AssertionError:

        print('The input dataset %s does not exist' % sf)
```

```
        raise

    return sf

def run(root, shipping_points_shp, bathymetry_tif):

    check_folders(root)
    check_input(shipping_points_shp)
    check_input(bathymetry_tif)

    # extract Values to Points
    # http://desktop.arcgis.com/en/arcmap/latest/tools/spatial-analyst-
toolbox/extract-values-to-points.htm

    all_points = os.path.join(arcpy.env.workspace,
"shipping_depths_all.shp")
    print("Creating %s" % all_points)
    arcpy.sa.ExtractValuesToPoints(in_point_features=shipping_points_shp,
in_raster=bathymetry_tif,
                                   out_point_features=all_points,
interpolate_values="NONE", add_attributes="VALUE_ONLY")

    # add a new Clearance field to the shapefile
```

```
    # http://desktop.arcgis.com/en/arcmap/latest/tools/data-management-toolbox/add-field.htm

    arcpy.AddField_management(in_table=all_points, field_name="Clearance", field_type="DOUBLE", field_is_nullable="NULLABLE",

                              field_is_required="NON_REQUIRED")



    # set the Clearance field value to the sea depth + draught

    # http://desktop.arcgis.com/en/arcmap/latest/tools/data-management-toolbox/calculate-field.htm



    expression = "!RASTERVALU! + !Draught!"

    arcpy.CalculateField_management(in_table=all_points, field="Clearance", expression=expression , expression_type="PYTHON")



    # select all points that have a clearance greater than -5m

    # http://desktop.arcgis.com/en/arcmap/latest/tools/analysis-toolbox/select.htm

    selected_points = os.path.join(arcpy.env.workspace, "shipping_depths_shallow.shp")

    where_clause = '"Clearance" > -5'

    print("Creating %s" % selected_points)

    arcpy.Select_analysis(in_features=all_points, out_feature_class=selected_points, where_clause=where_clause)

    return selected_points
```

```
if __name__ == "__main__":

    all_shipping_pts = ALL_SHIPPPING_PTS

    bathymetry = BATHYMETRY_TIF

    root = ROOT_FOLDER

    run(root, all_shipping_pts, bathymetry)

    print("Done!")
```