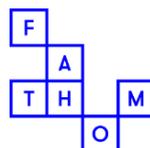


# CatchX Methodology Report



Photo by Susana Almeida



Luis Felipe Patino Velasquez,  
Wim Clymans

9/11/2018

# Table of Contents

<b>1. Introduction .....</b>	<b>3</b>
<b>2. Catchment Explorer Data Pre-processing .....</b>	<b>4</b>
2.1 <i>Software</i> .....	4
2.2 <i>Datasets</i> .....	4
2.3 <i>Data Processing</i> .....	5
<b>3 Online Set-up .....</b>	<b>11</b>
<b>4 Graphical representation.....</b>	<b>0</b>

# 1. Introduction

The CatchX web-platform has been designed by the University of Leeds and Earthwatch Europe to allow universal access to cutting-edge scientific data in any river catchment globally. The most appropriate datasets currently available have been selected and processed into 57,646 catchments. The platform allows users, without needing specialist skills, to interrogate the datasets in their catchment to visualise water balance information, explore trends and compare with other catchments. <https://earthwatch.org.uk/get-involved/projects-activities/catchx>

The project was funded by the UK Natural Environment Research Council (Grant number NE/P016812/1).

There are five reports in all covering the CatchX project:

- (1) The **Final Report** covers all aspects of the project in summary.
- (2) The **Methodology Report** describes the methods used to create the platform.
- (3) The **Data Selection Report** describes the review and selection procedure of datasets.
- (4) The **Workshop Report** describes the workshop held on user needs in South Africa.
- (5) The **Data Testing Report** describes testing of CatchX data for South Africa.

This **Methodology Report** aims to provide the reader with an overview of the different software and methods used during the development of the CatchX platform. The platform was initially developed during the pilot stage of the project (covering South Africa) and has since been scaled globally. The functionalities and design of the pilot stage have been based on user requirement scoping (**Final Report**) and evaluated during an interactive workshop in South Africa (**Workshop Report**).

All code created as part of the project, including data processing and front end development has been stored in an online repository in GitHub. Where possible the project's code was divided into different scripts corresponding to each of the sections within this report. Access to the different scripts can be achieved by contacting the project's team using the contact form online.

## 2. Catchment Explorer Data Pre-processing

### 2.1 Software

Software	Description
Python v 2.7.14	Programming Language
PostgreSQL v 9.6	Object-relational database system
PostGIS v 2.4.4	Spatial database extender for PostgreSQL
ArcGIS Desktop v 10.6	Geographic Information System
QGIS Desktop v 2.18.9	Geographic Information System

Table 1 Software used for pre-processing

### 2.2 Datasets

Dataset	Links	Variables	Geographic and Temporal Scale	Format
Open Street Map	<a href="#">OSM</a>	Waterway	Continent	.pbf
Earth2Observe	<a href="#">earth2observe</a>	Rain	Global – Daily	.nc
		Snow	Global – Daily	.nc
		Total Evapotranspiration (mean, min and max)	Global - Monthly	.nc
		Runoff (mean, min and max)	Global - Monthly	.nc
		Temperature	Global – Daily	.nc
Climate Change Initiative	<a href="#">CCI</a>	Land Cover	Global - Yearly	.tiff
HydroBasin	<a href="#">Hydrosheds</a>	Basin contours	Continent - NA	.shp

Table 2 Dataset pre-processing – links accessed November 08<sup>th</sup> 2018

## 2.3 Data Processing

The tools developed as part of this project use python (v2.7.14) as scripting language in combination with SQL statements when interacting with the project's database (PostgreSQL v 9.6).

### *Open Street Map*

---

1. **Extract OSM river data:** the software used to extract the river data from the original OSM dataset was 'osmosis' and the query for the software is:

```
osmosis --read-pbf-fast fileName.osm.pbf --wkv
keyValueList="waterway.river,waterway.stream,waterway.ditch" --used-node --
write-xml NewFileName.osm
```

2. **Create OSM river data shapefile:** using QGIS the output from 'osmosis' can be exported as shapefile – This step is necessary in order to run the script that adds the name of the river to the catchment
3. Adding the river name to catchments (HydroBASINS Standard – without lakes - level 7)
  - a. Script Logic
    - i. Create waterbody dataset identifying waterbody types that have been defined as *river* or *stream*
    - ii. In the new dataset modify the Name given to the river by removing the word '*river*' needed as this can generate some conflicts. i.e. River Breede will be changed to only Breede
    - iii. Dissolve all river/stream segments where the name matches, generating a unique line representing the water course for each name rather than having a set of multiple polylines
    - iv. Intersect all watercourses within each catchment.
    - v. Established the longest water course within each catchment by looking at the length attribute and assign the *OSM river name* to it.

*NB. This script must be manually updated when dealing with the Spanish language as the word used changes from 'river' to 'rio'. This was not the case when looking at datasets around Asia, Africa, Europe and North America*

4. Rainfall - Snowfall- Temperature – Evapotranspiration and Runoff

- a. Rainfall, Snowfall and Temperature files correspond to the daily dataset, whilst Evapotranspiration and Runoff files correspond to the monthly dataset

**Daily dataset:**

Rainf\_daily\_MSWEPEP\_025\_199001.nc

**Monthly dataset:**

e2o\_ecmwf\_wrr2\_glob15\_mon\_Runoff\_1980-2014.nc

- b. Each dataset was added to PostgreSQL using the following scripts

- i. Uploading data to PostgreSQL –

- 1. Script Logic – Rainfall, Snowfall and Temperature (daily)

- a. Open *nc file* and transform to pandas data-frame
    - b. Create monthly data-frame by grouping daily data using Latitude (Lat) and Longitude (Lon) for each point – **(Snowfall and Rainfall values were summed over the month whilst Temp values were averaged over the month)**
    - c. Create new time stamp for the data-frame using the format 01/MM/YYYY
    - d. Upload yearly data-frame to PostgreSQL

- 2. Script Logic – Evapotranspiration and Runoff (monthly)

- a. Open *nc file* and transform to pandas data-frame
    - b. If NaN values exist remove row from data-frame
    - c. Create data-frame between 1990 and 2014
    - d. Upload yearly data-frame to PostgreSQL

ii. Edit Tables in PostgreSQL by adding geometry and SRID

1. Script Logic

a. For each year create a geometry column using point Lat and Lon defining the coordinate system as WGS84 – SRID 4326

b. SQL query:

```
ALTER TABLE TableName
```

```
ADD COLUMN geom geometry(Point,4326);
```

```
UPDATE TableName
```

```
SET geom = ST_SetSRID(ST_MakePoint(lon, lat), 4326);
```

```
ALTER TABLE TableName
```

```
DROP COLUMN index
```

iii. Create yearly Dataset with weighted values of the variable

1. Script Logic. This process describes the methodology used for each individual catchment

a. Identify the Thiessen grids that intersect the catchment

b. For each Thiessen grid identify the surface area that lies within the catchment

c. For each of the Thiessen grid areas within the catchment calculate a weighted value by using the following formula:

$$\text{Variable Value} \times \left( \frac{\text{Thiessen grid area inside catchment}}{\text{Catchment area}} \right)$$

*Variable value = Rainfall, Snowfall, Temperature,  
Evapotranspiration and Runoff*

d. Sum or Average the weighted values to obtain the final weighted value for the catchment

iv. Create the Final dataset

1. This corresponds to the 25y monthly data. The final table is created by joining all the yearly tables using SQL in PostgreSQL

5. Each yearly Land Cover dataset was added to PostgreSQL using the following scripts:
  - a. LandCover data to CSV file.
    - i. Script Logic. This process describes the methodology used for each yearly dataset
      1. Using *Arcpy* the total areas of each Land Cover classification within each individual catchment were calculated. The tool used for this process was *Tabulate Area*; more information regarding the tool can be found here:  
<http://desktop.arcgis.com/en/arcmap/latest/tools/spatial-analyst-toolbox/tabulate-area.htm>
      2. The result of the tool was then transposed and converted to CSV
      3. The final output of the script is a set of yearly CSV files
  - b. CSV file to PostgreSQL.
    - i. Script Logic.
      1. Using a SQL query each CSV file representing yearly data was uploaded to PostgreSQL

*NB. The data display in the online platform corresponds to the TIFF file for the year 2015 downloaded from the Climate Change Initiative website.*

**1. Water balance data and Land cover**

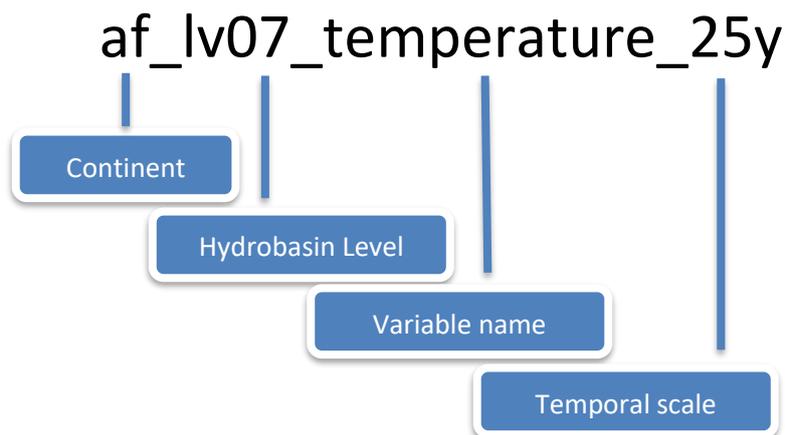
Dataset is the result of the script in step **4.b.iv. Create the Final Dataset in the Earth2Observe** section for precipitation, temperature, runoff and evapotranspiration. The dataset for land cover is the result of the script in step **5.b. CSV file to PostgreSQL in the Climate Change Initiative – Land Cover data** section.

These datasets provide the information for the graphical visualisation and the reporting tool in the CatchX platform.

Variable	Dataset name
<b>Precipitation</b>	af_lv07_precip_25y
<b>Temperature</b>	af_lv07_temperature_25y
<b>Evapotranspiration</b>	af_lv07_evapotran_25y
<b>Runoff</b>	af_lv07_runoff_25y
<b>Land cover</b>	af_lv07_land_cover_24y

*Table 3 Final output name convention*

The following is a description of the name convention:



NB. Within the name convention “af” refers to the dataset for Africa, within the project datasets for Antarctica – ar, Asia – as, Australia – au, Europe – eu, Greenland -gr, North America – na, South America – sa, Siberia -si, were also created. This classification echoes the Hydrobasin data classification.

## 2. Web map visualisation

This uses the dataset containing the monthly average for a 25 year’s period for each catchment globally for precipitation, temperature, evapotranspiration (mean, min, max) and runoff (mean, min, max). Each average was calculated individually using SQL and the outputs were put together using GIS to produce a final shapefile. This dataset was converted to a web mapping service which is use in the CatchX mapping platform.

SQL query example for Africa:

```
CREATE TABLE af_lv07_precip_25y_avg AS
SELECT
hybas_id,geom,
AVG(weighted_precip) as rain_avg, AVG(weighted_snow) as snow_avg
FROM af_lv07_precip_25y
GROUP BY hybas_id,geom
```

### 3 Online Set-up

The backend framework was developed over a pre-existent ESRI ArcGIS Enterprise cloud GIS solution deployed under a Microsoft Windows Azure cloud environment<sup>1</sup> using an IIS Server on Windows Server 2012 R2 Datacenter.

The data used by the platform are stored in a PostgreSQL relational database installed on the server as it allows the storage of both spatial and non-spatial data (PostgreSQL, 2017, PostgreSQL Documentation. From. <http://www.postgresql.com>). Access to the database is coded using PHP (recursive acronym for PHP, a scripting language well suited for web development as the code is executed on the server side).

Development of the user interface and visualisation render is based in Bootstrap for user interface components and responsive layout, JQuery/JavaScript for DOM manipulation, AJAX for communication and event handling and Highcharts under the non-commercial licence for data visualisation.

---

<sup>1</sup> ArcGIS Enterprise on Microsoft Azure - <http://enterprise.arcgis.com/en/server/latest/cloud/azure/overview-arcgis-server-on-microsoft-azure.htm>,2018

## 4 Graphical representation

