

# ***BestiaPop* - A Python package to automatically generate and visualise gridded climate data for crop model applications**

Jonathan J. Ojeda<sup>1</sup>, Diego Perez<sup>2</sup>

<sup>1</sup> Regrow Ag, Brisbane, QLD 4101, Email: [jonathan@regrow.ag](mailto:jonathan@regrow.ag), [www.jojeda.com](http://www.jojeda.com)

<sup>2</sup> Independent Cyber Security Researcher & Data Scientist, Brisbane, QLD 4101, Email: [darkquasar7@gmail.com](mailto:darkquasar7@gmail.com)

## **Abstract**

Climate data is an essential input for crop models to predict crop growth and development using site-specific (point) or gridded climate data. While point data is usually available in a readily accessible format, gridded data is often stored in NetCDF files which are difficult to convert to an input file readable by the Agricultural Production Systems sIMulator (APSIM) or other crop models such as the Decision Support System for Agrotechnology Transfer (DSSAT). We developed *BestiaPop*, a Python package which allows model users to automatically download gridded climate data in an APSIM and DSSAT format from an Australian (Scientific Information for Land Owners; SILO) and global (NASA- Prediction Of Worldwide Energy Resource; NASA-POWER) climate data source. The package offers the possibility to select a range of grids (0.05° resolution) and years producing files with daily climate data in different formats (CSV, MET, WTH). We (i) compared the *Bestiapop* performance to download CSV, MET and WTH files using multiprocessing and (ii) tested the performance of the package to generate climate data across areas suitable for potato (*Solanum tuberosum* L.) in Tasmania, Australia. A total of 1724 climate files across 20 years (1991-2020) were automatically downloaded and the spatio-temporal variability of climate inputs was mapped. The case study reveals that implementing *BestiaPop* is a useful and efficient tool to automatically download gridded climate data in an APSIM format and could be extended to other crop models and regions across the world.

## **Keywords**

Python, weather data, data science, APSIM, model inputs, MET file

## **Introduction**

Crop models need climate data to predict crop growth and development. The spatial resolution of these datasets ranges from site-specific (point) or gridded climate data at different spatial and temporal resolutions. A previous study integrated, with crop models (Sparks, 2018), an automatic tool to download point climate data from the NASA - Prediction Of Worldwide Energy Resource (NASA-POWER) API in the R programming language. For Australia, point climate data is currently available in an Agricultural Production Systems sIMulator (APSIM) format, however, gridded climate data is stored in very large NetCDF files which are difficult to convert to an input file readable by APSIM or other crop models. Although some APSIM studies used gridded climate data across Australia to simulate crop yield (Ojeda et al., 2020; 2021; Zhao et al., 2013) and irrigation water requirements (Ojeda et al., 2020; 2021), there is no available tool to download, visualise and format gridded climate data in an APSIM format automatically for thousands of climate files in the cloud with minimum hardware requirements. Here we present the development of *BestiaPop*, a modular Python application that can add new connectors (data sources) and producers (climate data formats such as CSV, MET, WTH, etc.). This package has the potential to be used in conjunction with data analysis tools of the Python ecosystem as a data pre-processor, therefore considerably speeding the data exploration and pre-parsing phases of crop modelling studies. The ease of access to data produced by this tool using Project Binder in the cloud democratizes access to climate data around the world without relying on proprietary software. In this paper, we tested (i) the *Bestiapop* performance to download CSV, MET and WTH files using parallel processing and (ii) the use of the package to automatically download SILO climate data for modelling purposes across the areas suitable for potato (*Solanum tuberosum* L.) in Tasmania, Australia.

## **Methods**

### *Package functionality*

*BestiaPop* uses a selection of Python packages that make extracting climate data from large files in the Scientific Information for Land Owners (SILO) and NASA-POWER cloud a fast and flexible process. The program is built with modularity in mind and implements backend modules (connectors and producers) to achieve its objectives. Connectors reach out to climate data source APIs and only request specific datasets thus avoiding the need to download whole climate files for data exploration. We implement *xarray* instead of the slower *NetCD4* package to access multi-dimensional arrays (latitude, longitude and day of the year) inside NetCDF files.

Finally, data processing and conversion to CSV, MET and WTH files is performed using *Pandas* and parallel processing allowing slower systems like end-user laptops to run *BestiaPop* smoothly. Although *Bestiapop* can be easily implemented through the command line, the package can also be executed in the cloud for free with a single click using Project Binder which turns the *BestiaPop* GitHub repo into a collection of interactive Jupyter notebooks. Binder opens those notebooks in an executable environment, making the code and data produced by it immediately reproducible by anyone, anywhere without the need to download or install any software or dependencies.

#### *BestiaPop* performance using SILO and NASA-POWER data sources

We used an AMD Ryzen Threadripper 2990WX 32-Core Processor (128 GB of physical memory) to download 20 latitude × 20 longitude combinations from SILO (400 files at 0.05° resolution) through the command line interface. The same latitude-longitude combinations were retrieved from NASA-POWER, however it generated only 9 files at 0.5° resolution due to the nature of its data resolution. Runs were performed for a 5-year period to generate MET, WTH and CSV files with the parallel computing (PaCo) function of the package activated and deactivated. We calculated the total workload time to generate all files [*Total Time (seconds)*], a single file [*Time/File (s)*] and the time to generate a single year of daily data [*Time/Year (seconds)*]. We also estimated the efficiency of the parallel computing function, i.e. how many times faster was *BestiaPop* using PaCo activated [*PaCo Efficiency (times)*].

#### Case study using SILO data source

*Bestiapop* has the option to choose the data source (SILO or NASA-POWER). In this case study, we used only the SILO dataset across the areas suitable for potato in Tasmania, Australia. SILO's gridded climate data (Jeffrey et al., 2001) are freely available for download via a web interface (<https://www.longpaddock.qld.gov.au/silo/gridded-data/>) and contains continuous daily climate data for Australia from 1889 to present. The grid spans 112°E to 154°E, 10°S to 44°S with resolution 0.05° latitude by 0.05° longitude (approximately 5 km × 5 km). We applied *BestiaPop* to automatically download daily solar radiation, rainfall, maximum and minimum temperature in an APSIM input format (MET file) across Tasmania (n=8064 grids) using a Dell Latitude 5411, Intel Core-i7 6 cores (32 GB of physical memory). Then, the grids within areas suitable for potato in Tasmania (<https://dpiwwe.tas.gov.au/agriculture/investing-in-irrigation/enterprise-suitability-toolkit/enterprise-suitability-maps>) were selected (n=1724 grids) inputting a list of latitude and longitude combinations hosted in the GitHub repository. Average of sum of solar radiation, sum of rainfall, mean daily minimum temperature and mean daily maximum temperature for the period 1991-2020 were calculated and mapped across the areas suitable for potato in Tasmania (Ojeda et al., 2020; 2021). For this functionality *Bestiapop* uses *cartopy* and *matplotlib* Python packages. The complete Jupyter notebook example of the *Bestiapop* application using the Project Binder for this region is hosted in the following link: <https://mybinder.org/v2/gh/JJguri/bestiapop/HEAD?filepath=sample-data%2FExampleMapsTasmania.ipynb>.

```
# Define variables to pass in to Bestiapop

#You need to host your Lat/Lon List in some web server like GitHub and then provide the url. This example is using a csv host
from csv import reader
url = "https://raw.githubusercontent.com/JJguri/bestiapop/master/sample-data/bpop_tassie_example.csv" #specify url
df = pd.read_csv(url)
coordinate_list = df.values.tolist()
action='generate-climate-file' #specify action
data_source='silo' #specify data source
year_range='1991-2020' #specify range of years
climate_variables=['radiation', 'max_temp', 'min_temp', 'daily_rain'] #specify climate variables
output_type='dataframe'
output_path=None
```

**Figure 1.** Screen shoot of *Bestiapop* input parameters through the Project Binder interface for an example to download and visualise climate data across the areas suitable for potato in Tasmania, Australia. Inputs are (i) *url*, list of latitude and longitude combinations, (ii) *action*, generate the climate file, (iii) *data source*, SILO, (iv) *year range*, from 1991 to 2020, (v) *climate variables*, solar radiation, maximum temperature, minimum temperature and rainfall and (vi) *output type*, Pandas dataframe which then is used to create the maps.

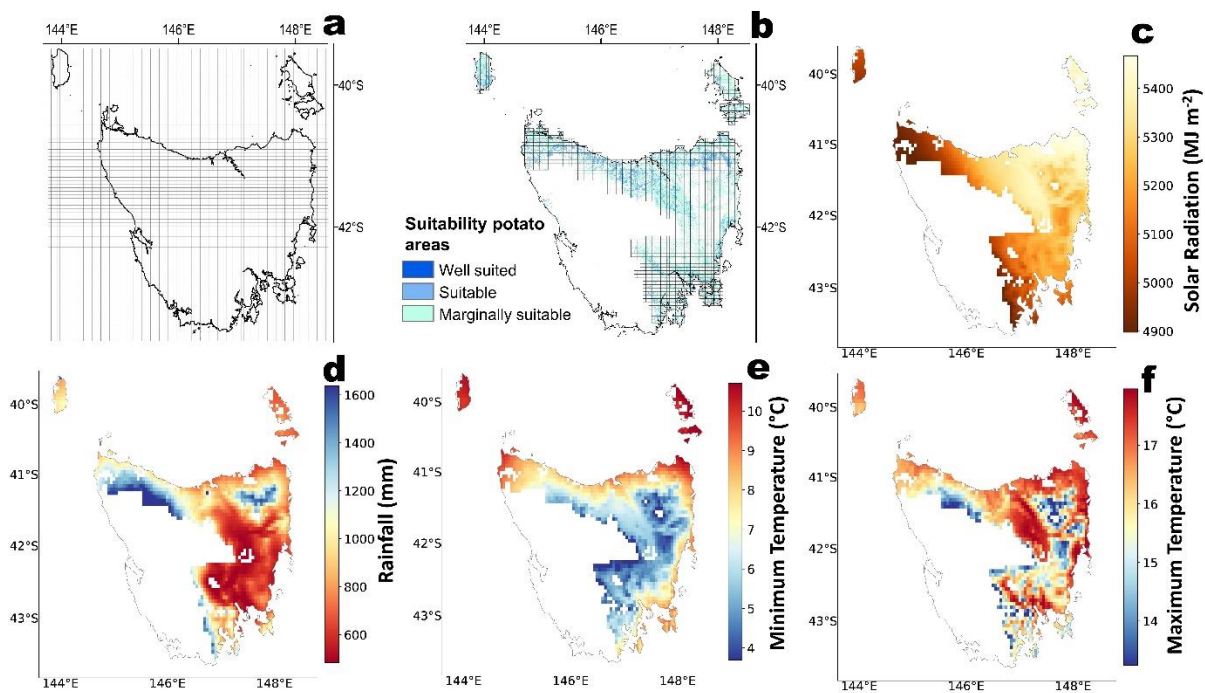
## Results

*Bestiapop* was 11, 9.5 and 9.3 times faster to generate a MET, WTH and CSV file when PaCo was activated (Table 1). The total time needed to generate a MET file using NASA-POWER was 110% faster than using SILO with PaCo activated due to the differences in data resolution between data sources (0.05° vs 0.5° for SILO and NASA-POWER, respectively) for the same study area (Table 1). We found that when the number of files is reduced (e.g. 9 files in this example), the benefits of parallel processing applied by *Bestiapop* are not tangible. However, when the amount of latitude-longitude combinations increased (e.g. 400 files), the use of the parallel processing function is recommended due to its increase considerably the time efficiency (Table 1).

**Table 1. Descriptive performance indicators for *BestiaPop* implemented through command line.**

Data Source	Parallel Computing (PaCo)	File Type	Number of Files	Years	Total Time (s)	Time/ File (s)	Time/ Year (s)	PaCo Efficiency (times)
SILO	deactivated	MET	400	5	1571	3.93	0.79	
		WTH	400	5	1450	3.63	0.73	
		CSV	400	5	1503	3.76	0.75	
	activated	MET	400	5	149	0.37	0.07	10.5
		WTH	400	5	152	0.38	0.08	9.5
		CSV	400	5	161	0.40	0.08	9.3
NASA-POWER	deactivated	MET	9	5	41	4.56	0.91	
		WTH	9	5	38	4.22	0.84	
		CSV	9	5	41	4.56	0.91	
	activated	MET	9	5	71	7.89	1.58	0.6
		WTH	9	5	84	9.33	1.87	0.5
		CSV	9	5	94	10.44	2.09	0.4

The time required to manually create a long-term MET file from SILO gridded climate data could increase up to several hours and days due to data stored in separated yearly files. *BestiaPop* was able to download daily solar radiation, rainfall, minimum and maximum temperature for 10,958 days (30 years) at a rate of 3.5 seconds per 0.05° grid using the Project Binder interface. The package automatically generated a Pandas dataframe (18,891,592 rows), yearly statistics by grid cell, data visualisation in maps and the corresponding MET files for the suitable potato areas across Tasmania (n=1724; Fig. 2) in a few minutes using a medium-range laptop. The spatial variability of climate data across the areas suitable for potato in Tasmania, Australia is shown in Figure 2.



**Figure 2.** Maps of 0.05° climate grid cells across (a) Tasmania and (b) areas suitable for potato in Tasmania (blue scale). (c) Spatial variability of sum of solar radiation, (d) sum of rainfall, (e) daily minimum temperature and (f) daily maximum temperature across areas suitable for potato in Tasmania. Values shown in panels c, d, e and f are the means of the climate variable for 30 years (1991-2020).

## Conclusion

Our analysis reveals that the implemented *BestiaPop* is a useful and efficient tool to automatically download a massive amount of gridded climate data across Tasmania in an APSIM format in a shorter period of time than conventional methods and could be potentially extended to other crop models and regions in Australia (using SILO) and the world (using NASA-POWER). This study highlights the need to create more efficient and easy-to-use tools for (i) climate input data visualisation for pre-processing and (ii) automation of input data generation for crop models. The code is publicly available in a GitHub repository (<https://github.com/JJguri/bestiapop>). For further information please contact the corresponding author. While *BestiaPop* does not redistribute any of the SILO data, users are encouraged to please refer to the reference guidelines available at, <https://www.longpaddock.qld.gov.au/silo/about/publications-references/>.

## References

- Jeffrey SJ, et al. (2001). Using spatial interpolation to construct a comprehensive archive of Australian climate data. *Environmental Modelling and Software*. 16, 309–330. (doi.org/10.1016/S1364-8152(01)00008-1)
- Ojeda JJ, et al. (2020). Effects of soil- and climate data aggregation on simulated potato yield and irrigation water requirement. *Science of the Total Environment*. 710, 135589. (doi:10.1016/j.scitotenv.2019.135589)
- Ojeda JJ, et al. (2021). Implications of data aggregation method on crop model outputs – The case of irrigated potato systems in Tasmania, Australia. *European Journal of Agronomy*. 126, 126276. (doi.org/10.1016/j.eja.2021.126276)
- Sparks AH, (2018). nasapower: a NASA POWER global meteorology, surface solar energy and climatology data client for R. (doi:10.21105/joss.01035)
- Van Rossum G, (1995). Python tutorial, Technical Report CS-R9526, Centrum voor Wiskunde en Informatica (CWI), Amsterdam.
- Zhao G, et al. (2013). Large-scale, high-resolution agricultural systems modeling using a hybrid approach combining grid computing and parallel processing. *Environmental Modelling and Software*. 41, 231–238. (doi.org/10.1016/J.ENVSOFT.2012.08.007)