

# Running Taverna Workflows within IPython Notebook

Alan Williams, Aleksandra Pawlik, Carole Goble, University of Manchester, UK

## Introduction

**IPython Notebook**, originally developed by Fernando Perez of University of Berkeley (where the core development team is currently based) is a browser-based **environment for interactive computing**.

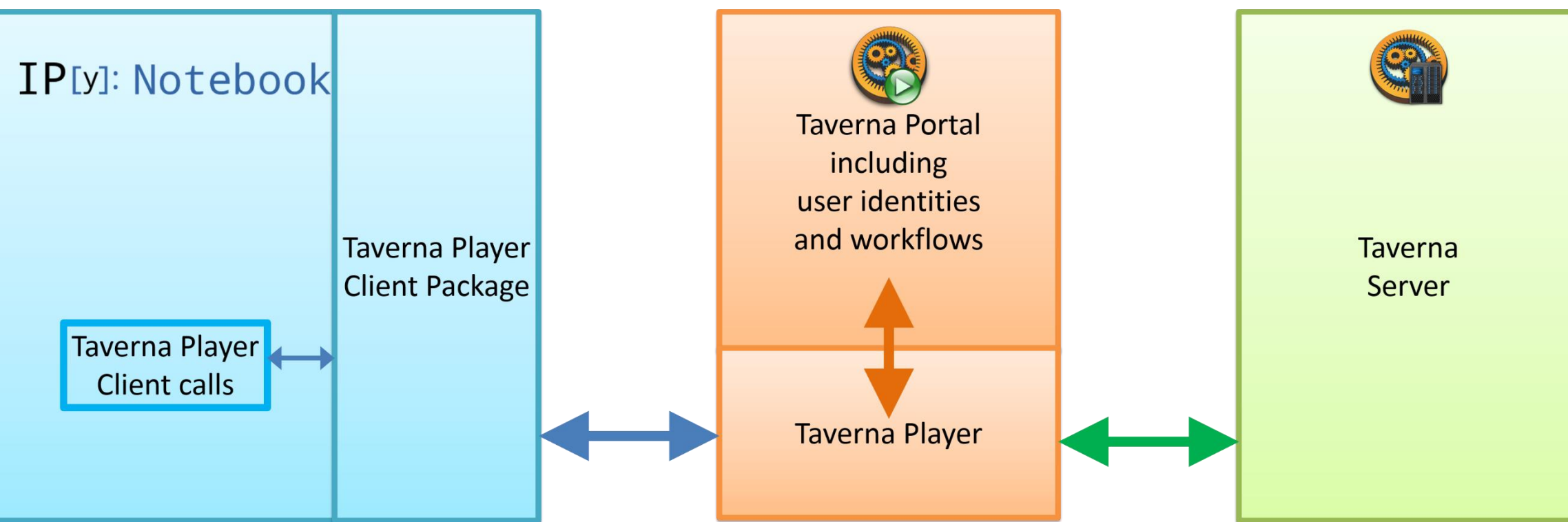
Users can **write, edit and re-run Python scripts**. IPython Notebook has support for interactive data visualization and report presentation. A notebook can be saved and shared. The code saved in notebooks can be modified and re-run using the same or different data. The record of a notebook "run" can be saved and displayed in a static Notebook Viewer.

IPython Notebooks with embedded workflows can be saved and shared; their static version can be viewed in the [IPython Notebook Viewer](#).

**Taverna** is a suite of open source tools that allow the design and execution of scientific workflows. The running of Taverna Workflows within IPython Notebook allows users to include existing **workflows as part of their interactive computing**. This provides IPython users with access to functionality that has been developed for a significant amount of time and to the large number of shared workflows. In addition, use of workflows provides access to functionality within workflows that may not be readily available within the Python environment.

For workflow developers, calling the workflows from IPython Notebook allows: **pre-processing of input** data, the **chaining of workflow** executions with data transformations in Python, and the presentation of results and **report generation** using IPython's extensive capabilities.

## Architecture



## Taverna Player

- Allows Taverna workflows to be run as part of a website like embedding a video from Youtube
- Uses [Taverna Portal](#) for user authorization
- Reads workflows from a [Taverna Portal](#)
- Runs workflows on a [Taverna Server](#)
- Has a [REST API](#)

## Taverna Player Client

- Uses [Taverna Player](#) and its Server to run workflows within an [IPython Notebook](#)
- Data passed from the Notebook to the executing Taverna Workflows
- Workflow run's requests for data answered within the Notebook using Taverna's [interaction service](#)
- Results retrieved from the run and fed back into the Notebook
- Available from the [PyPi](#) registry
- Tested with BioVeL workflows for data refinement and ecological niche modelling.

## Taverna Server

- Runs arbitrary Taverna 2 workflows
- Secure, isolated, user-specific workflow execution
- Manages data for separate runs
- Publishes interactions and provenance

## Flows

Flow	Client	Server
Authorize Client	Notebook -> Player -> Portal Portal -> Player -> Notebook	Start run
List workflows	Notebook -> Player -> Portal Portal -> Player -> Notebook	Interact with run Server -> Player -> Notebook Notebook -> Player -> Server
Get workflow and run template	Notebook -> Player -> Portal Portal -> Player -> Notebook	Return results Server -> Player -> Notebook

## Use Case

- Connect to Taverna Player
- Read data file
- Select random rows
- Run Ecological Niche Modelling
- Organize ENM outputs for analysis
- Run statistical stack
- Use ENM outputs for difference
- Run statistical difference
- Display projections and difference

## Use Case

The example notebook shows the running of a series of workflows for **Ecological Niche Modelling (ENM)**, using the [Taverna Player Client](#) and the [pandas](#) and [random](#) Python packages.

A file of csv data representing occurrences of species is read. 100 rows are selected at random for use in the Ecological Niche Modelling.

A model is created and projected for the present day and for 2050.

The results of the ENM are re-organized and then used as inputs to two other workflows: one to produce statistics and one to calculate the differences (climatic effects) between the projections.

## Client Creation

```
In [3]: from tavernaplayerclient import *
In [4]: c = Client('http://dev.at.biovel.eu', 'player', 'player')
In [5]: workflows = c.workflows
In [6]: for w in c.workflows: print w.identifier, w.title
```

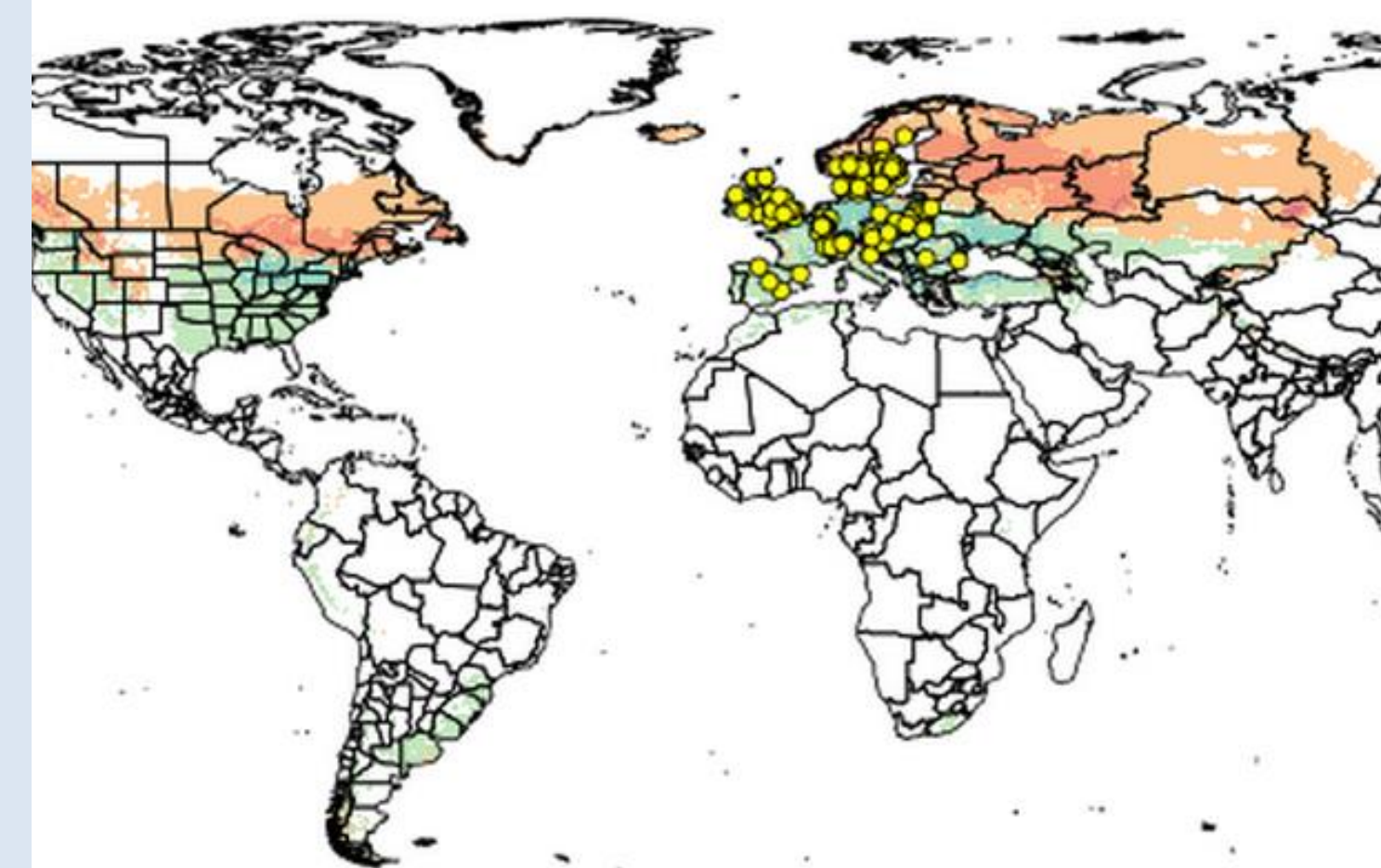
## Interaction with Run

```
In [20]: esw = c.get_workflow(20)
In [21]: esw_run_template = esw.run_template
In [22]: print esw.description
In [ ]: for k in esw_run_template.inputs.keys(): print k
In [23]: namelist = "now,2050"
In [24]: layerlist = projection_urls[0] + "," + projection_urls[1]
In [25]: esw.run("Example esw run",
                {"namelist": namelist,
                 "layerlist": layerlist,
                 "png_size": "800"})
```

## Organization of Data

```
1 Various output types
2 Ecological niche modelling
3 Select Model For Me with co
4 Define with PartitionFinder
5 Partitioning environmental
6 Partitioning environmental
7 Bioclim workflow with inte
8 A workflow
9 BioVeL ESW DIFF - ENM Stat
```

## Display of Results



## Future Work

- Collation of provenance
  - The provenance of the workflow runs will be linked with the history generated by IPython Notebook
- Creation of Research Objects
  - Combine the IPython Notebook, the workflows and the linked provenance within a "Research Object" - <http://www.researchobject.org/>
- Creation of documentation
  - Use IPython or dexty to generate reports about workflow runs
- Running the Taverna Player directly on the Taverna Server
  - The Taverna Player currently has to be within a Taverna Portal, or similar workflow registry.
  - In the future, Taverna Player will be extended to run directly against a Taverna Server
  - Will allow running of "arbitrary" workflows and allow them to be exposed in IPython Notebook
- Improved interaction
  - Allow the data entered in an interaction to come from values within IPython Notebook
- Teaching
  - Use the combination of IPython Notebook and Taverna to provide training materials

## Links

IPython and IPython Notebook [www.ipython.org](http://www.ipython.org)  
 The Taverna Player code on GitHub at [www.github.com/myGrid/taverna-player](https://github.com/myGrid/taverna-player)  
 The Taverna Player Client package at [www.pypi.python.org/pypi/tavernaPlayerClient](https://pypi.python.org/pypi/tavernaPlayerClient) and the code on GitHub at [www.github.com/myGrid/DataHackLeiden](https://github.com/myGrid/DataHackLeiden)  
 Taverna's homepage at [www.taverna.org.uk](http://www.taverna.org.uk)

## Acknowledgments

This work was enabled by the [BioVeL project](#), that has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 283359, and also enabled by the [Software Sustainability Institute](#) supported by the UK Engineering and Physical Sciences Research Council (EPSRC) through grant EP/H043160/1

Much of the work on the Taverna Player Client was carried out during the "Data enrichment hackathon" organized by [pro-Biosphere](#). Thanks are due to [Ross Mounce](#) of the University of Bath, UK and to [Youri Lamers](#) of Naturalis, Netherlands for their work on the client during the hackathon.

Thanks are also due to the [developers of IPython Notebook](#) for their help and advice during the hackathon.