ALGOPIPER

User Guide

A visual tool to create modular pipelines
from AlgoRun algorithms

AlgoPiper is a web-based interface to create, run, integrate and share pipelines from AlgoRun published algorithms. This document offers a tour through AlgoPiper interface and how to use it to create, run, integrate and share pipelines.

1 Introduction

AlgoPiper is a web-based software tool that offers a graphical user interface to create, run, integrate and share pipelines consisting of algorithms packaged with AlgoRun[1]. This document provides a tour through AlgoPiper interface and how to use it to create computational pipelines, run them on a given dataset, integrate them with other software tools using a RESTful API[2], and share them on AlgoPiper website[3]. Alternatively, learn to use AlgoPiper through an interactive step-by-step tutorial by navigating to: http://algopiper.org/try.

It is highly recommended to use the tool from AlgoPiper website. However, the document shows how to install and run AlgoPiper software on a local environment.

2 User Interface

AlgoPiper is a web-based interface that looks like the below picture.

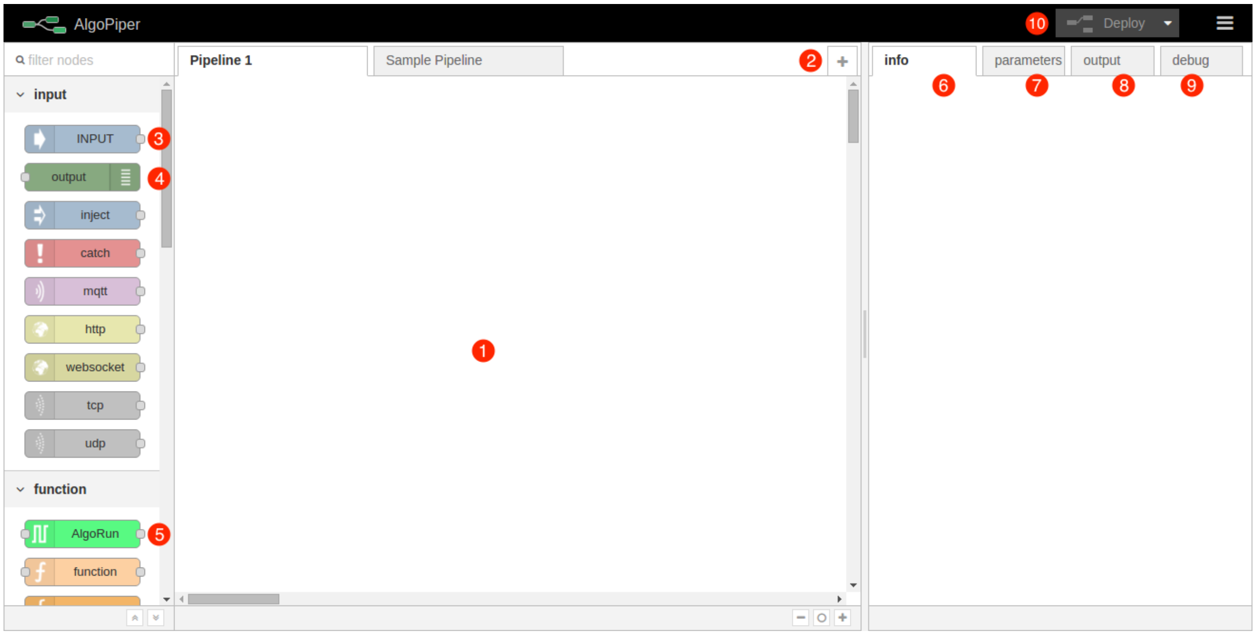


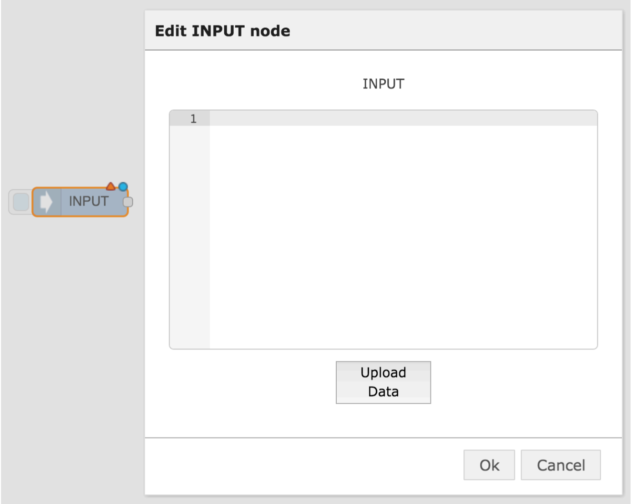
Figure 1 - AlgoPiper Interface. (1) Main workspace to create a pipeline. (2) Pipelines are arranged into tabs. Click '+' to add more pipelines. (3) Input node: represents input data to the pipeline. (4) Output node: is used to preview the output of the pipeline. (5) AlgoRun node: represents individual algorithms packaged using AlgoRun. (6) A tab to present detailed information about a selected node. (7) A tab to modify AlgoRun node parameters <algorithm parameters>. (8) A tab to present results from the output node. (9) A tab to present log information from debug node. (10) Deploy button to build a pipeline after creation.

[1] AlgoRun website: http://algorun.org
[2] RESTful Application Programming Interface: https://en.wikipedia.org/wiki/Representational_state_transfer
[3] AlgoPiper website: http://algopiper.org

3 Create Pipelines

To create a pipeline, drag and drop nodes from the left-side bar. A basic pipeline contains the following nodes:

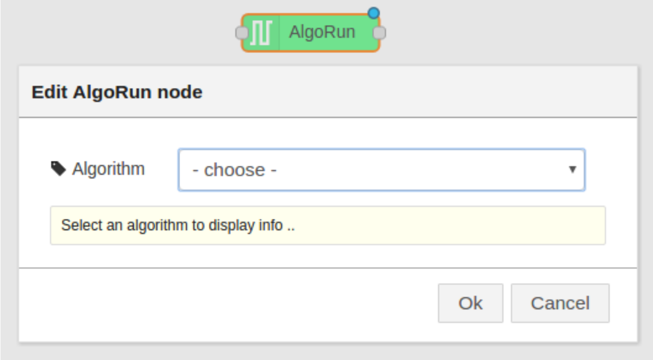
- **Input** node: is used to upload data to the pipeline. Drag an input node to the main workspace. Double click on the node to type data in the input dialog or upload a file.



Edit INPUT node

Figure 2 - Input node dialog

- **AlgoRun** node(s): each AlgoRun node represents an algorithm ready to be run on a given input. Drag an AlgoRun node into the main workspace. Double click on the node to select an algorithm from the list.



Edit AlgoRun node

Algorithm: - choose -

Select an algorithm to display info ..

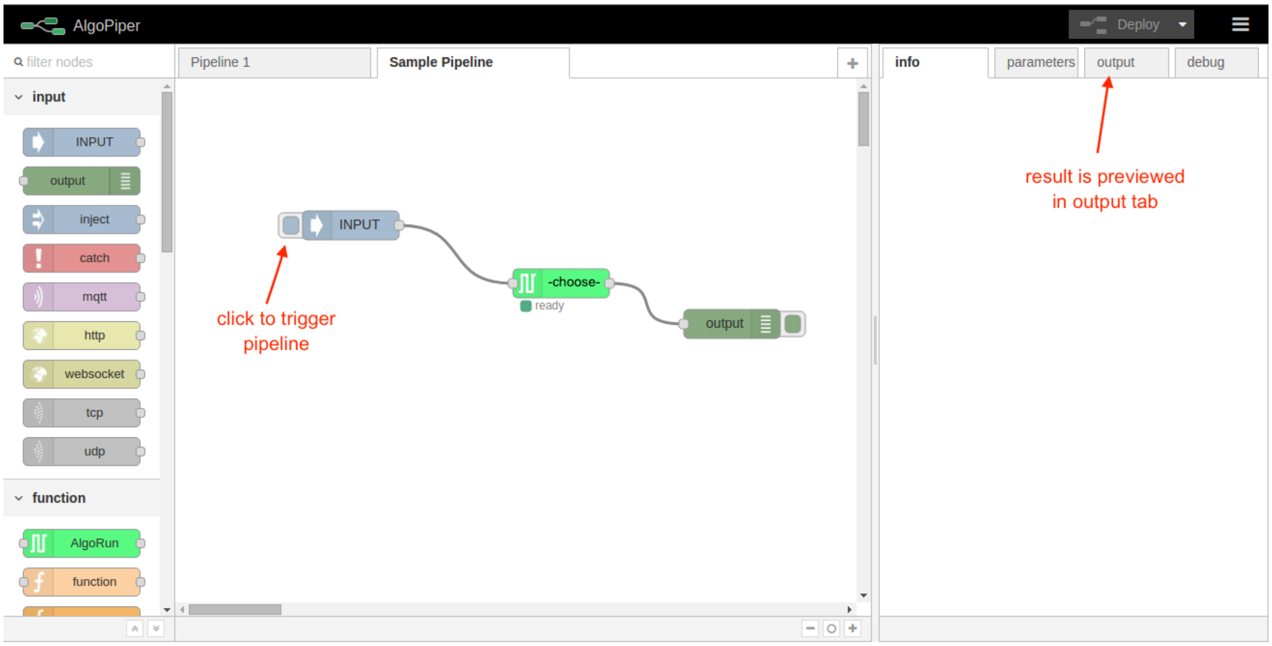
Figure 3 - AlgoRun node dialog

  - **Info tab**: shows algorithm description, input and output formats, and a reference to the algorithm page.
  - **Parameter tab**: shows algorithm parameters that can be changed dynamically.
- **Output** node: previews the output from the pipeline in the output tab on the right.

Connect the input node to the AlgoRun node by clicking on the small dot to the right of the input node to the small dot to the left of the AlgoRun node. Similarly, connect the small dot to the right of the AlgoRun node to the small dot the left of the output node. After creating the pipeline, hit deploy on the top-right corner.

4 Run Pipelines

To run a pipeline, click on the small button on the left of the Input node. This will trigger the pipeline by passing the input to the next node (after the Input) in the pipeline. Output appears in the output tab on the right sidebar.



click to trigger pipeline

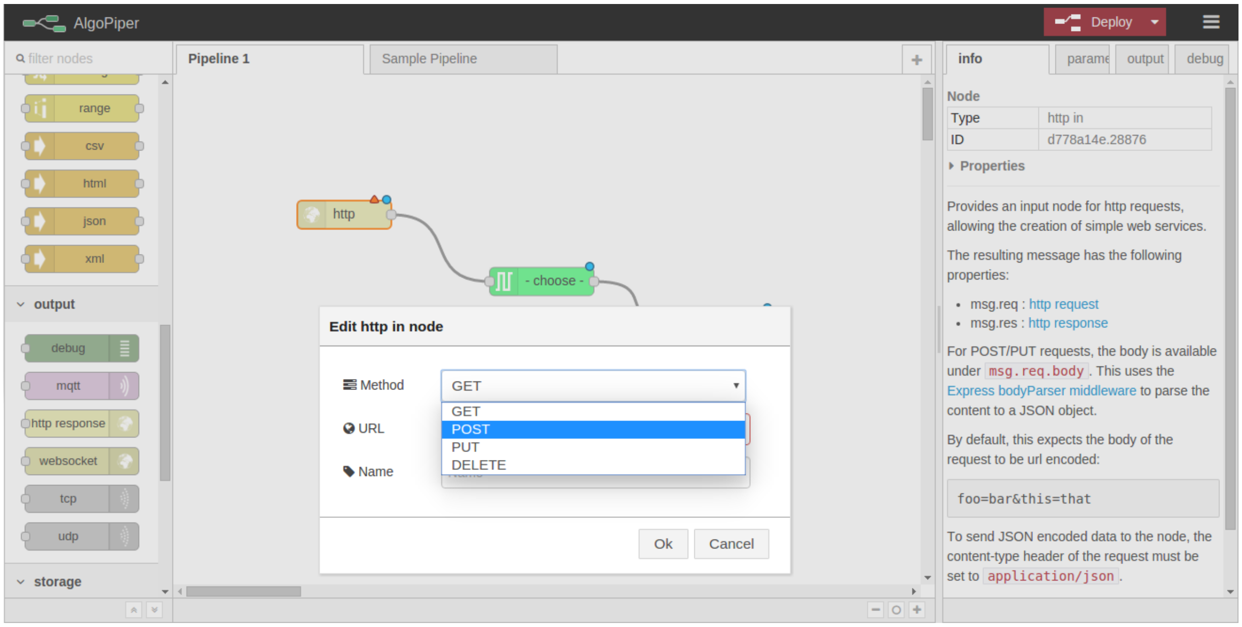
result is previewed in output tab

Figure 4 - Run a pipeline and preview output

Note: If you cannot find the algorithm you are looking for, follow the guide on http://algorun.org/documenttion to package it into an AlgoRun container.
Submit the packaged algorithm to AlgoRun website http://algorun.org/submit-algorithm and it will automatically appear in the ‘choose’ list of the AlgoRun node.

5 Integrate Pipelines

After creating and testing the pipeline on some arbitrary data, integrate it into other software tools by adding an HTTP endpoint node before the first node in the pipeline (replacing the Input node) and an HTTP response node after the last node in the pipeline (replacing the output node). See the below images for an example.



Node

Type	http in
ID	d778a14e.28876

Provides an input node for http requests, allowing the creation of simple web services.

The resulting message has the following properties:

- msg.req : http request
- msg.res : http response

For POST/PUT requests, the body is available under `msg.req.body`. This uses the Express bodyParser middleware to parse the content to a JSON object.

By default, this expects the body of the request to be url encoded:

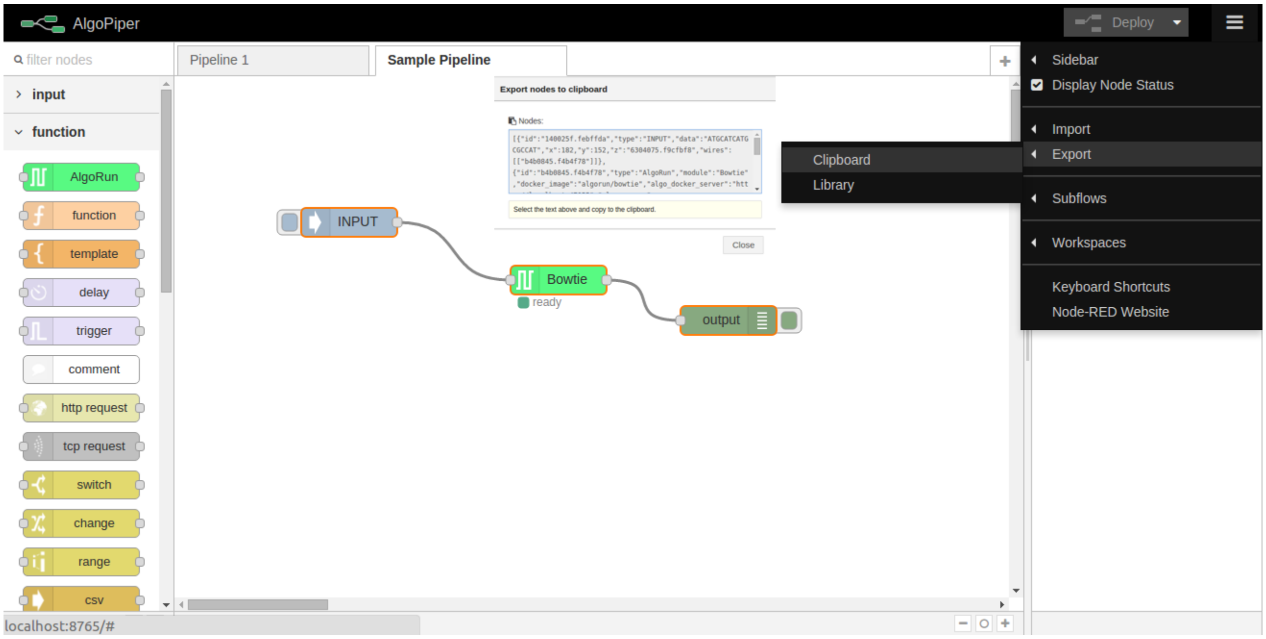
```
foo=bar&this=that
```

To send JSON encoded data to the node, the content-type header of the request must be set to `application/json`.

Figure 5 - Integrating a pipeline by adding an HTTP endpoint

6 Share Pipelines

Select all pipeline nodes in the main workspace, click on the top-right menu and choose to export to clipboard. The pipeline is exported into a JSON[4] format.



Export nodes to clipboard

Nodes:

```
[{"id":"140025f.febffda","type":"INPUT","data":"ATGCATCATG
CGCCAT","x":182,"y":152,"z":"6304075.f9cfbf8","wires":
[["b4b0845.f4b4f78"]]},
{"id":"b4b0845.f4b4f78","type":"AlgoRun","module":"Bowtie"
,"docker_image":"algorun/bowtie","algo_docker_server":"htt
```

Select the text above and copy to the clipboard.

Close

Figure 6 - Export a pipeline for sharing

If you built your pipeline with AlgoPiper and want to share it publicly, do not hesitate to submit it for listing on the AlgoPiper website. The AlgoPiper website serves as a repository for all computational pipelines that were exported from AlgoPiper: http://algopiper.org

To submit your pipeline for listing, fill the form located at http://algopiper.org/submit-pipeline

7 Examples

For complete pipelining examples, please refer to Additional File 2.

[4] JavaScript Object Notation: http://www.json.org/

8 Local Installation Guide [Optional]

Due to the non-native support of Docker on Mac OS and Windows, AlgoManager can only be installed and run on a Linux environment. However, AlgoPiper can be installed and run on Mac OS, Windows as well as Linux. Thanks to the distributed architecture of the tool, this AlgoManager and AlgoPiper can run on two different platforms (locally or remotely).

8.1. Install Prerequisites

- Docker Engine: Follow the instructions on: https://docs.docker.com/v1.8/installation/
- NodeJS and Node Package Manager (npm): is JavaScript runtime environment. Follow the instructions on: https://nodejs.org/

8.2. Download and Install AlgoManager

AlgoManager is a module responsible for managing the creation of AlgoRun algorithms' containers used by the AlgoPiper.

- Clone the repository from GitHub: https://github.com/algorun/algomanager
- Navigate to the downloaded directory and run the following command to install node modules: `npm install`
- Configure AlgoManager `settings.json` file by providing values for the following keys:

Key	Description	Example
server	The server URL that runs AlgoManager. Use http://localhost for local environments. Change it to server domain for production environments.	`http://localhost`

images	A list of available algorithms packaged using AlgoRun in a dictionary format. A "name" value is the algorithm name that will appear of AlgoPiper interface. A "docker" value is algorithm docker image on the server.
	Note: all algorithms' docker images must be existing on the server that runs AlgoManager.
		`[`
		`{“name”: “BLASTN”,`
		`“docker”:`
		`“algorun/blastn”},`
		`{“name”: “Bowtie”,`
		`“docker”:`
		`“algorun/bowtie”}`
		`]`

8.3. Download and Install AlgoPiper

AlgoPiper is the web application that serves the graphical user interface and run the server-side code.

- Clone the repository from GitHub: https://github.com/algorun/algopiper
- Navigate to the downloaded directory and run the following command to install node modules: `npm install`

8.4. Run the Web Applications

1. Start AlgoManager server by navigating to AlgoManager directory and running the following command: `node manager.js`
2. Verify that the application is running by navigating to http://localhost:8764 on your web browser. You should see a status of **working**.
3. Open another terminal windows and start AlgoPiper server by navigating to AlgoPiper directory and running the following command: `node red.js –m http://localhost:8764`
4. Navigate to http://localhost:8765 on your web browser to see AlgoPiper interface.

Note: download and install `forever` (https://github.com/foreverjs/forever) to run AlgoManager and AlgoPiper as daemons in the background.