

---

# **AlgoPiper Documentation**

***Release 1.0***

**Abdelrahman Hosny**

**Jul 29, 2017**



---

## Contents

---

<b>1</b>	<b>Access AlgoPiper</b>	<b>3</b>
<b>2</b>	<b>User Guide</b>	<b>5</b>
2.1	1. Create Pipelines . . . . .	6
2.2	2. Run Pipelines . . . . .	7
2.3	3. Integrate Pipelines . . . . .	7
2.4	4. Share Pipelines . . . . .	8
2.5	5. Submit Your Pipeline (Optional) . . . . .	9
<b>3</b>	<b>Examples</b>	<b>11</b>
3.1	Parallel BLAST Jobs . . . . .	11
3.2	Analysis of high-throughput mRNA sequencing (RNA-Seq data) Pipeline . . . . .	15
<b>4</b>	<b>Local Installation</b>	<b>21</b>
4.1	Install Prerequisites . . . . .	21
4.2	Download AlgoManager . . . . .	21
4.3	Configuring the Production Environment . . . . .	21
4.4	Add Available Algorithms . . . . .	21
4.5	Run AlgoPiper . . . . .	22
<b>5</b>	<b>License</b>	<b>23</b>
<b>6</b>	<b>Need Help?</b>	<b>27</b>



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 [AlgoPiper](#). This page 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, and share them on AlgoPiper website. It also gives examples as cases of study for using the tool.



# CHAPTER 1

---

## Access AlgoPiper

---

**AlgoPiper Website:** AlgoPiper is available as a web-based tool that can be used immediately from the website. Navigate to <http://algopiper.org> and hit the *Launch* tab.





## CHAPTER 2

### User Guide

**Note:** You can skip reading this section and learn to use AlgoPiper through an interactive step-by-step tutorial by the online instance of AlgoPiper [Access AlgoPiper](#)

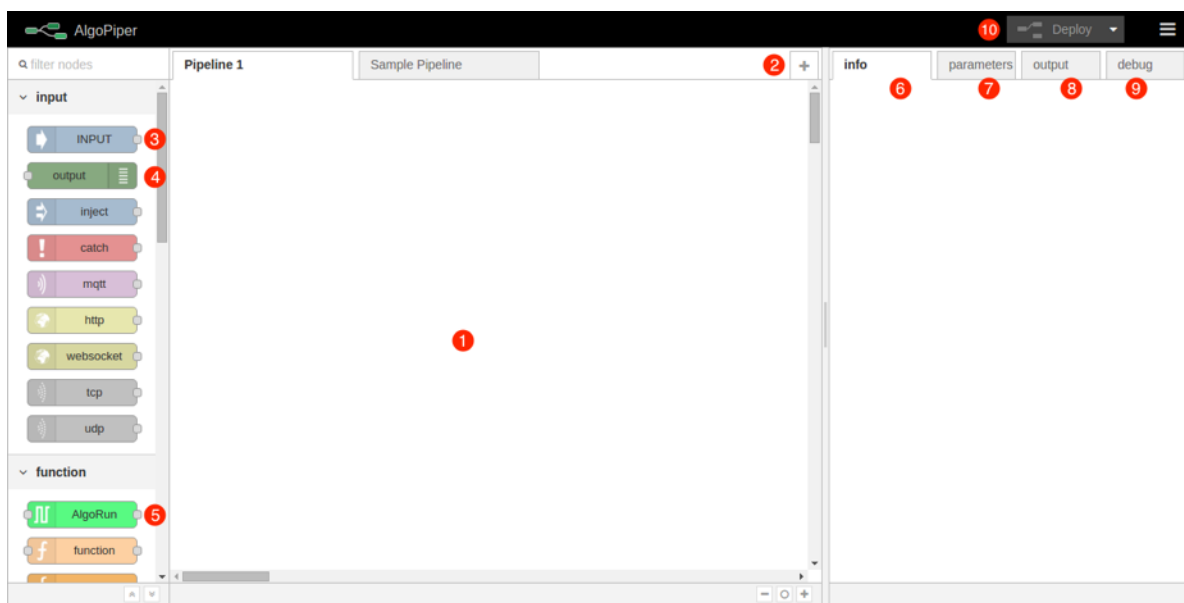


Figure 1 - AlgoPiper Interface. (1) Main workspace to create a pipeline. (2) Pipelines are arranged into tabs. Click '+' to add more pipelines. (3) Input node: represents input data to the pipeline. (4) Output node: is used to preview the output of the pipeline. (5) AlgoRun node: represents individual 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. 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.

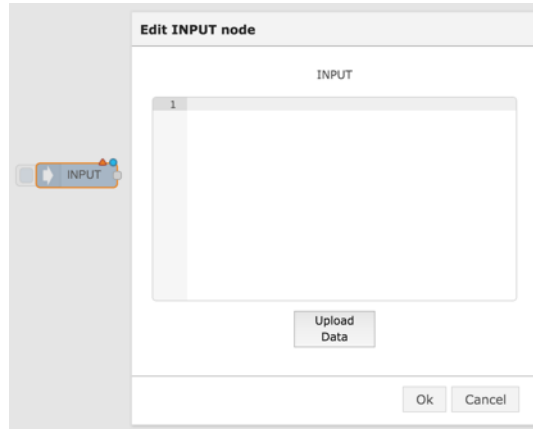


Figure 2 - Input node dialog

- **AlgoRun node(s):** each AlgoRun node represents an algorithm ready to be run on a given input. The input is passed to the node from the left. The output from the node is produced toward the right. Drag an AlgoRun node into the main workspace. Double click on the node to select an algorithm from the list. The CPU share is a relative measure of how many CPU cycles are allocated for each container on the AlgoManager instance ([read more](#)). The memory limit is an absolute measure of the maximum amount of RAM allocated to this specific AlgoRun container ([read more](#)).

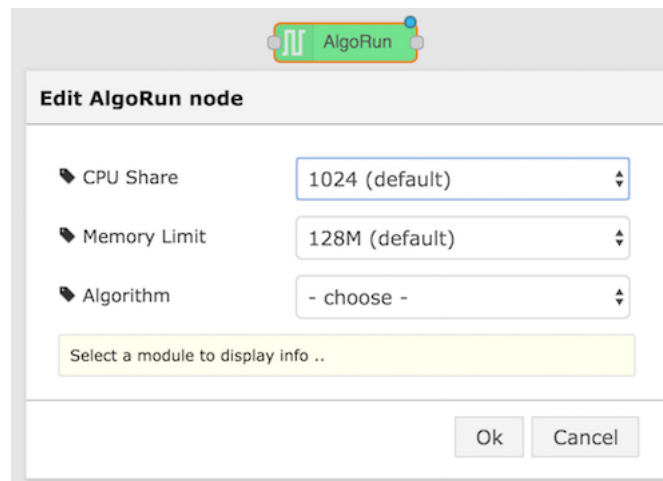


Figure 3 - AlgoRun node dialog

On the right sidebar, there are more tabs that show information about the selected algorithm:

- **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 on the right of the input node to the small dot on the left of the AlgoRun node. Similarly, connect the small dot on the right of the AlgoRun node to the small

dot on the left of the output node. After creating the pipeline, hit deploy on the top-right corner.

## 2. 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.

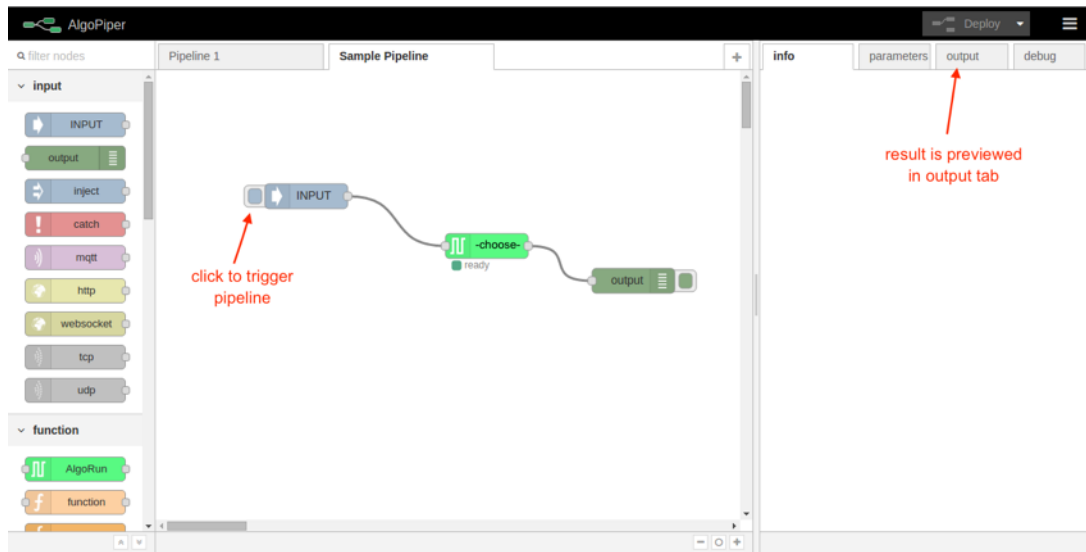


Figure 4 - Run a pipeline and preview output

**Note:** If you cannot find the algorithm you are looking for, follow the guide on <http://algorun.org/documentation> 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.

## 3. 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.

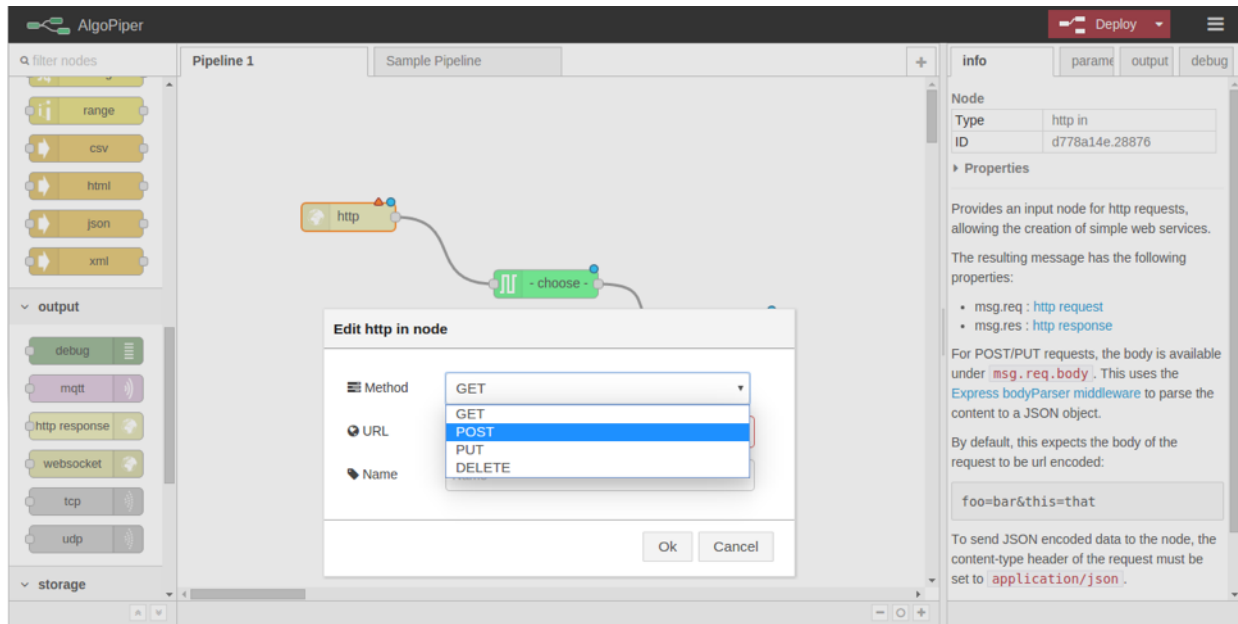


Figure 5 - Integrating a pipeline by adding an HTTP endpoint

## 4. Share Pipelines

Select all pipeline nodes in the main workspace (ctrl+A), click on the top-right menu and choose to export to clipboard. The pipeline is exported into a **JSON** format.

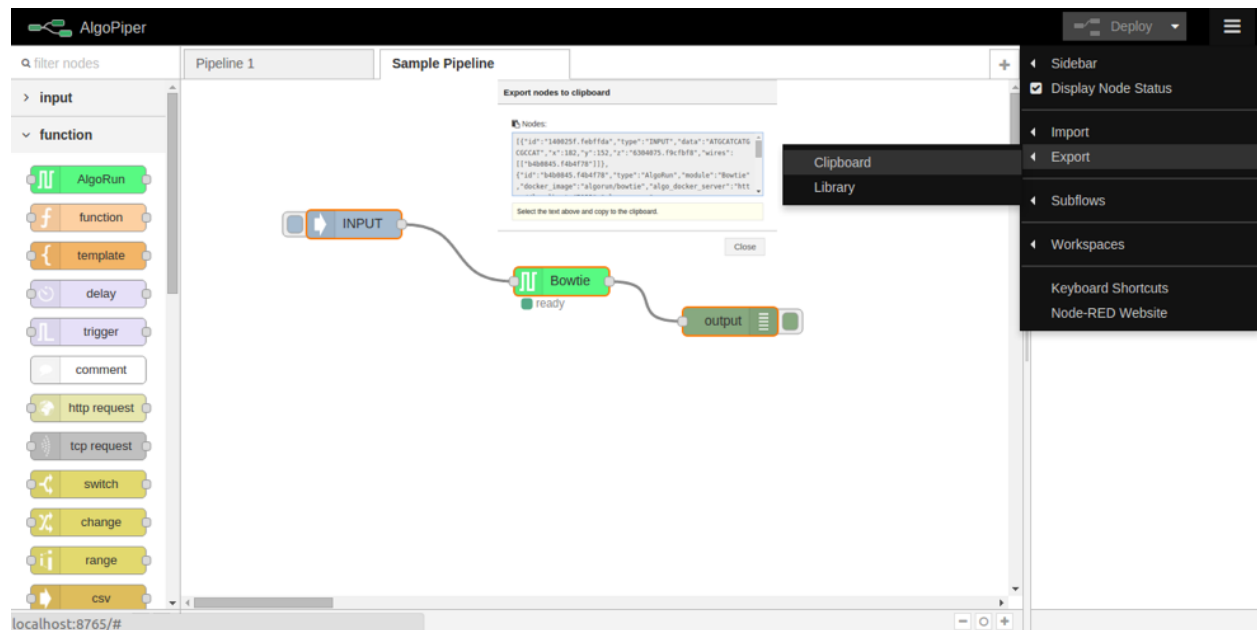


Figure 6 - Export a pipeline for sharing

## 5. Submit Your Pipeline (Optional)

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>



### Parallel BLAST Jobs

BLAST (Basic Local Alignment Search Tool) is a suite of programs provided by NCBI for aligning query sequences against those present in a selected target database. In this example, we create a BLAST pipeline that can be used to perform parallel search in nucleotide databases using multiple nucleotide queries. Running parallel BLAST queries saves computation time and the created pipeline can be integrated into other software tools through a [RESTful web API](#).

**Tip:** You can skip reading this section by importing this pipeline directly from AlgoPiper website. Navigate to <http://algotrainer.org/browse> to search for available pipelines. In the search box, type in “BLAST” to find it and launch immediately.

#### 1. Create the Pipeline

1. From the palette of nodes on the left, drag an AlgoRun node and drop it into the main workspace. Double click on the node and choose “BLAST” from the dropdown list of available algorithms. Click ok to close the edit dialog. Detailed information about BLAST is shown in the info tab on the right-side panel.
2. Drag an input node and drop it to the left of the BLAST node. Include as many input nodes as the number of the parallel jobs you will submit. Double click on each input node and upload the nucleotide query file, or copy and paste the query into the text area ( [some inputs to try](#) ). Connect all input nodes to the BLAST node.
3. Drag an OUTPUT node and drop it to the right of the BLAST node. Connect the BLAST node to the output node. The below figure shows the complete pipeline with three parallel jobs.
4. Hit Deploy on the top-right corner. This initializes the pipeline and the back-end AlgoRun containers then perform the computations.

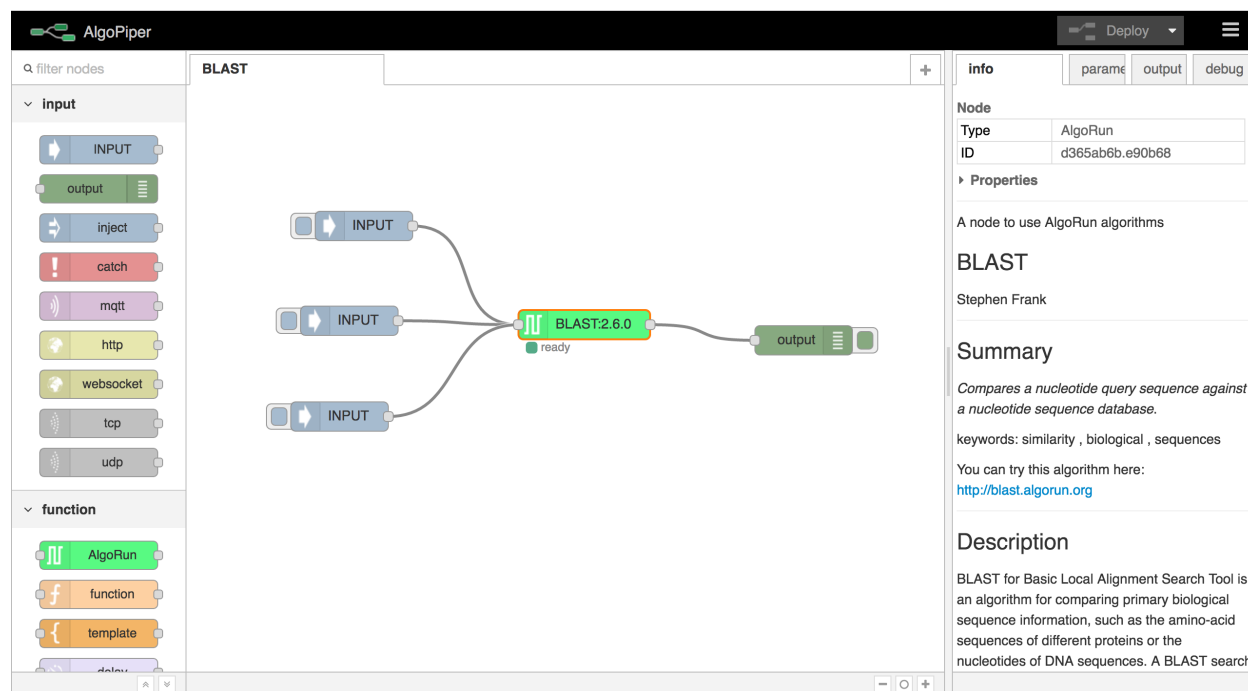


Figure 7 - BLAST Jobs Pipeline

## 2. Run the Pipeline

To submit the jobs, click on the small button on the left of each input node. This will pass the nucleotide query from each input node to the BLAST node which will run separate parallel threads for each input. Results appear on the output tab on the right.

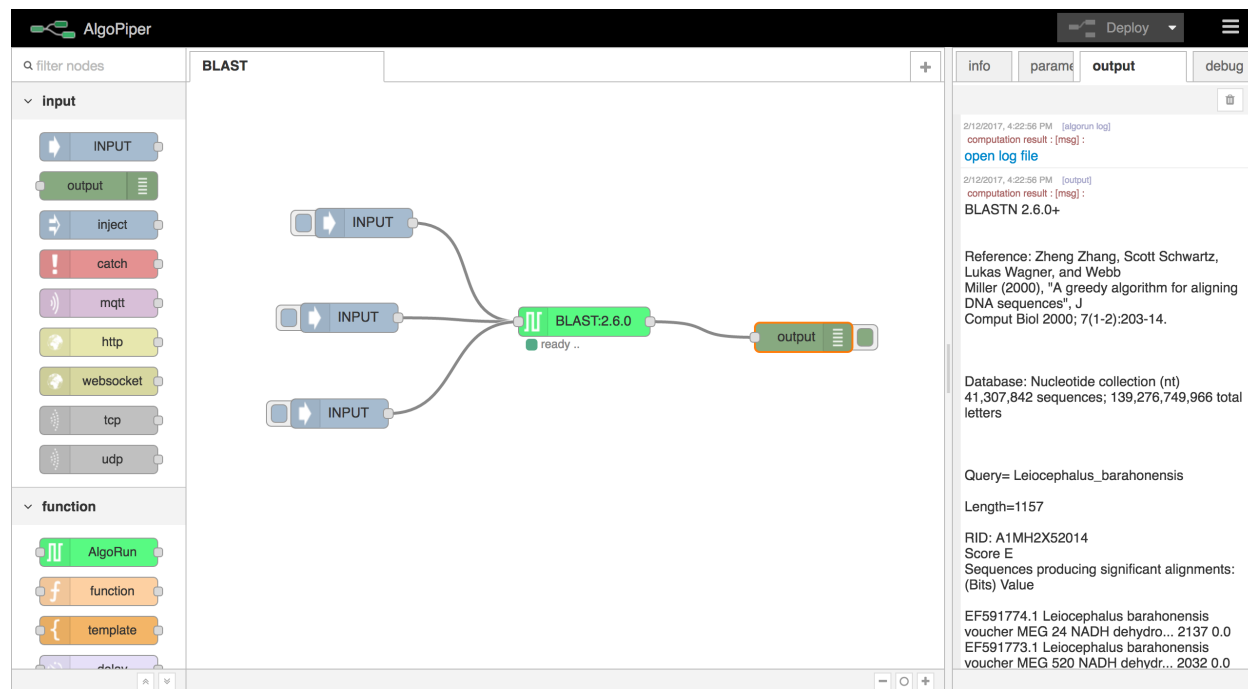


Figure 8 - BLAST search results appear on the output tab

Click open log file to open the output in a separate file that you can download. Scroll down to see results from



other threads (and open the corresponding log files).

**Hint:** The output node is used to display the results on the output tab. The BLAST node, and all AlgoRun nodes, log the results to a file even after removing the output node.

### 3. Integrate the Pipeline

Besides running BLAST jobs manually (by triggering the input node), you can wrap the pipeline in a web API by providing HTTP access to the pipeline functionality.

1. Remove the input and output nodes from the pipeline. Keep the BLAST node.
2. Drag an http node from the left palette to the left of the BLAST node. Double click on the node to define the request method and the URL. Set the request method to *POST* and the URL to */blast*. Give it an optional name and click ok.
3. Drag a function node from the palette to the right of the http node. The function node is used to parse the http request and only relay the nucleotide query to be inputted to the BLAST node. Edit the function node as shown in the code snippet on the right.



Figure 9 - The function node

4. Drag an http response node from the left palette to the right of the BLAST node.
5. Connect the http node to the function node. Connect the function node to the BLAST node. Connect the BLAST node to the http response node.
6. Hit **Deploy** to initialize the pipeline. The resulting pipeline is shown in the figure below.

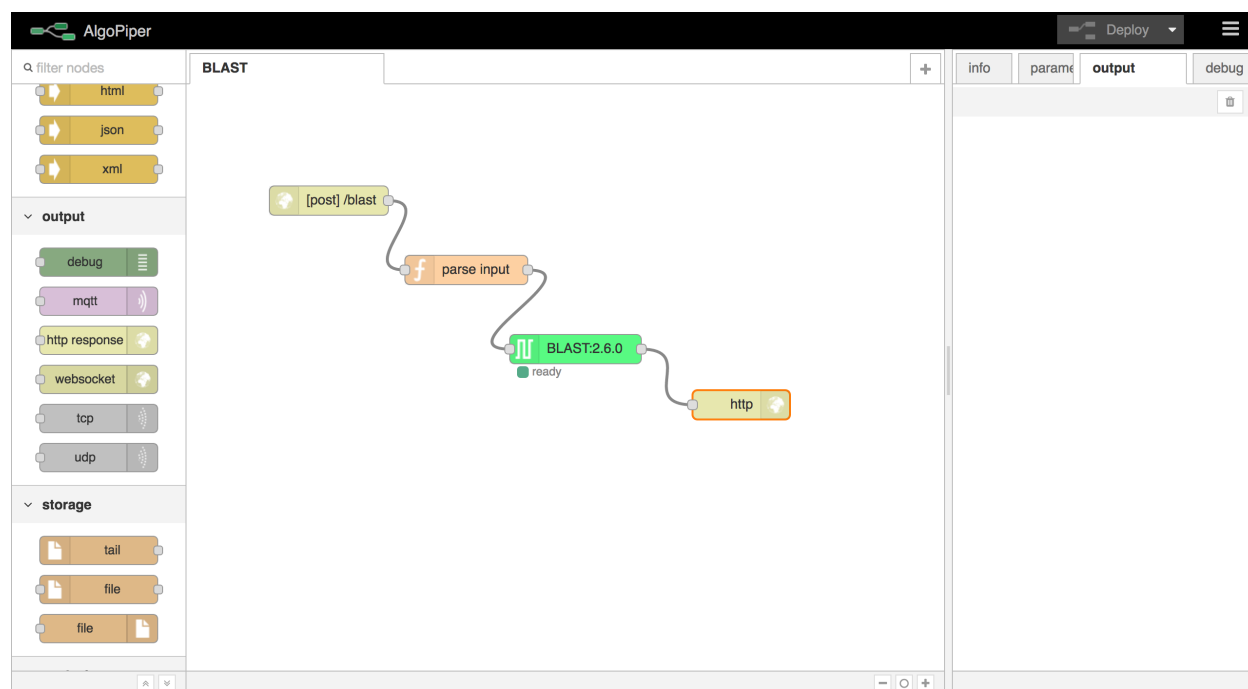


Figure 10 - Integrating BLAST pipeline using HTTP POST endpoint

As an example of running the pipeline through the web API, see the Firefox Poster plugin example below. The web API can be used in any other plugins or programming languages.

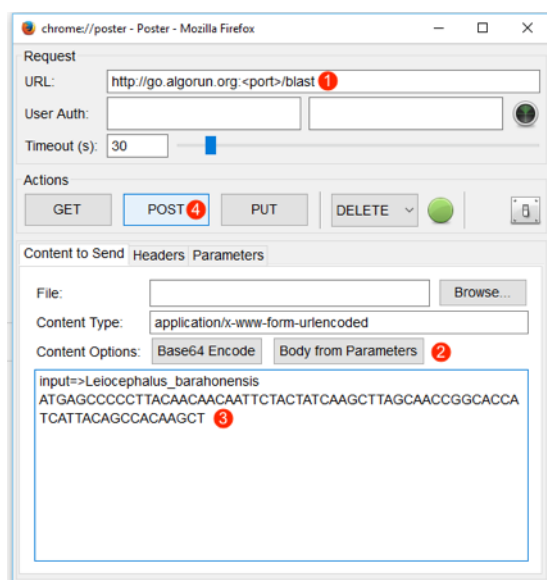


Figure 11 - Submitting a BLAST job using AlgoPiper API. (1) Type in the URL of the AlgoPiper followed by /blast (2) Choose “Body from Parameters” to send the parameters in the format of form-urlencoded. (3) Type input=<place your input here> in the input area. (4) Click POST to send the request and receive the results.

#### 4. Share the Pipeline

Now, export the pipeline to a JSON format to save to a local file or share it on AlgoPiper website. Use the mouse to select all nodes in the pipeline. From the top-right menu, choose **Export** and click **Clipboard**. Copy the JSON text and paste it to a local text file or submit it directly to AlgoPiper website (<http://algotiper.org/submit-pipeline>).

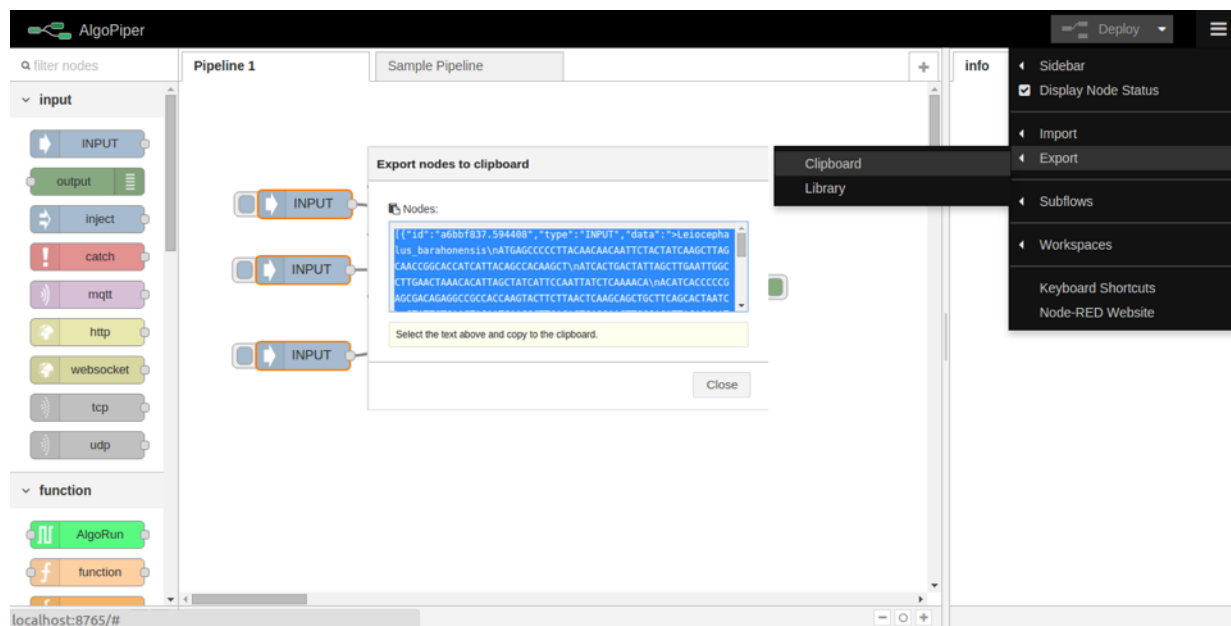


Figure 12 - Sharing the BLAST Pipeline

## Analysis of high-throughput mRNA sequencing (RNA-Seq data) Pipeline

Next Generation Sequencing (NGS) enables researchers to study biological systems at a large scale. In this example, we create a pipeline that performs basic analysis of [Illumina](#) RNA-seq data using a sample transcriptome with the goal of obtaining expression scores or annotated genes. The pipeline depends on [TopHat](#) for alignment and [Cufflinks](#) for counting and expression scoring; both are previously packaged with [AlgoRun](#). TopHat was packaged with a test reference file for illustration purpose. To package TopHat with your own genomic reference, refer to the examples folder of [AlgoRun repository](#) to re-package TopHat locally with your own reference file.

**Tip:** You can skip reading this section by importing this pipeline directly from [AlgoPiper website](#). Navigate to <http://algopiper.org/browse> to search for available pipelines. In the search box, type in “RNA” to find it and launch immediately.

### 1. Create the Pipeline

1. From the palette of nodes on the left, drag an [AlgoRun](#) node and drop it into the main workspace. Double click on the node and choose “TopHat” from the dropdown list of available algorithms. Click ok to close the edit dialog. Detailed information about TopHat is shown in the info tab on the right-side panel.
2. Drag an [AlgoRun](#) node and drop it into the main workspace. Double click on the node and choose “Cufflinks” from the dropdown list of available algorithms. Click ok to close the edit dialog. Detailed information about Cufflinks is shown in the info tab on the right-side panel.
3. Drag an input node and drop it to the left of the TopHat node. Double click on the input node and copy and paste the sample input in this link: [http://tophat.algorun.org/algorun\\_info/input\\_example.txt](http://tophat.algorun.org/algorun_info/input_example.txt)
4. Connect the input node to the TopHat node. Connect the output from the TopHat node to the input of the Cufflinks node.

5. Drag an *OUTPUT* node and drop it to the right of the Cufflinks node. Connect the output of the Cufflinks node to the output node. The below figure shows the complete pipeline.
6. Hit Deploy on the top-right corner. This initializes the pipeline and the back-end AlgoRun containers the perform the computations.

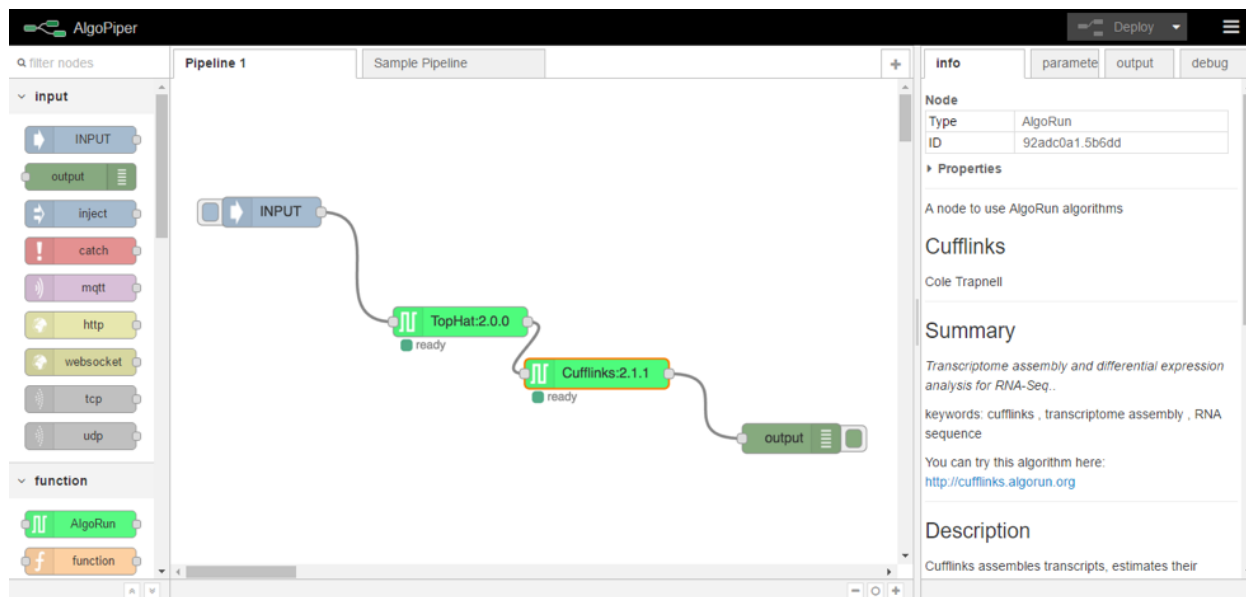


Figure 13 - RNA-Seq analysis pipeline using TopHat and Cufflinks

## 2. Run the Pipeline

To start analysis, click on the small button on the left of each input node. This will pass the sequence reads from the input node to the TopHat node which will automatically pass the results to the Cufflinks node. Results appear on the output tab on the right.

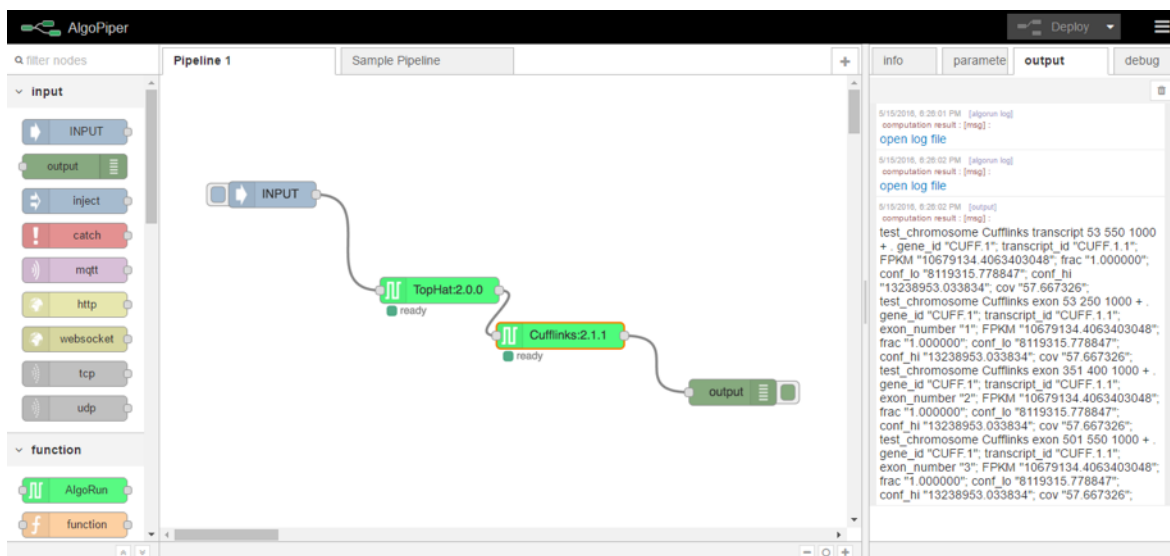


Figure 14 - RNA-Seq analysis pipeline results appear on the output tab

Hover over the first `open log file` line, it will highlight the TopHat node, indicating that this log file came from the TopHat node. Hover over the second `open log file` line, it will highlight the Cufflinks node, indicating that his log file came from the Cufflinks node. The output from Cufflinks is printed in the output tab as well.

### 3. Manipulate Cufflinks Parameters

AlgoPiper interface enables you to change algorithms parameters without re-deploying the pipeline again. Select the *Cufflinks* node and open the parameters tab from the right. The parameters of Cufflinks are shown in the figure below. Change the *FragLen\_Mean* parameter from the default value of 200 to some other value and click save. This will change the fragment length average of unpaired reads input to your new value. Re-running the pipeline means using the newly saved parameters when Cufflinks module is invoked.

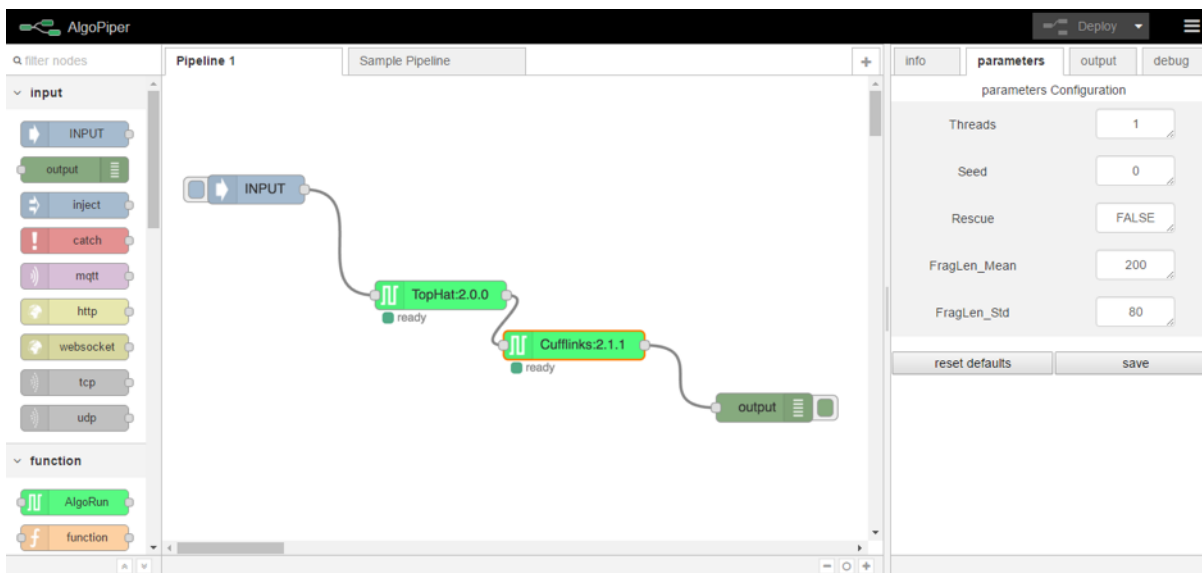


Figure 15 - Changing Cufflinks parameters on-the-go

### 4. Integrate the Pipeline

Besides running the RNA-Seq analysis manually (by triggering the input node), you can wrap the pipeline in a web API by providing HTTP access to the pipeline functionality.

1. Remove the input and output nodes from the pipeline. Keep the TopHat and Cufflinks nodes.
2. Drag an http node from the left palette to the left of the TopHat node. Double click on the node to define the request method and the URL. Set the request method to *POST* and the URL to */RNA-Seq*. Give it an optional name and click ok.
3. Drag a *function* node from the palette to the right of the http node. The function node is used to parse the http request and only relay the RNA sequence reads to be inputted to the *TopHat* node. Edit the function node as shown in the code snippet on the right.
4. Drag an http response node from the left palette to the right of the *Cufflinks* node.
5. Connect the *http* node to the *function* node. Connect the *function* node to the *TopHat* node. Connect the *Cufflinks* node to the *http response* node.
6. Hit *Deploy* to initialize the pipeline. The resulting pipeline is shown in the figure below.

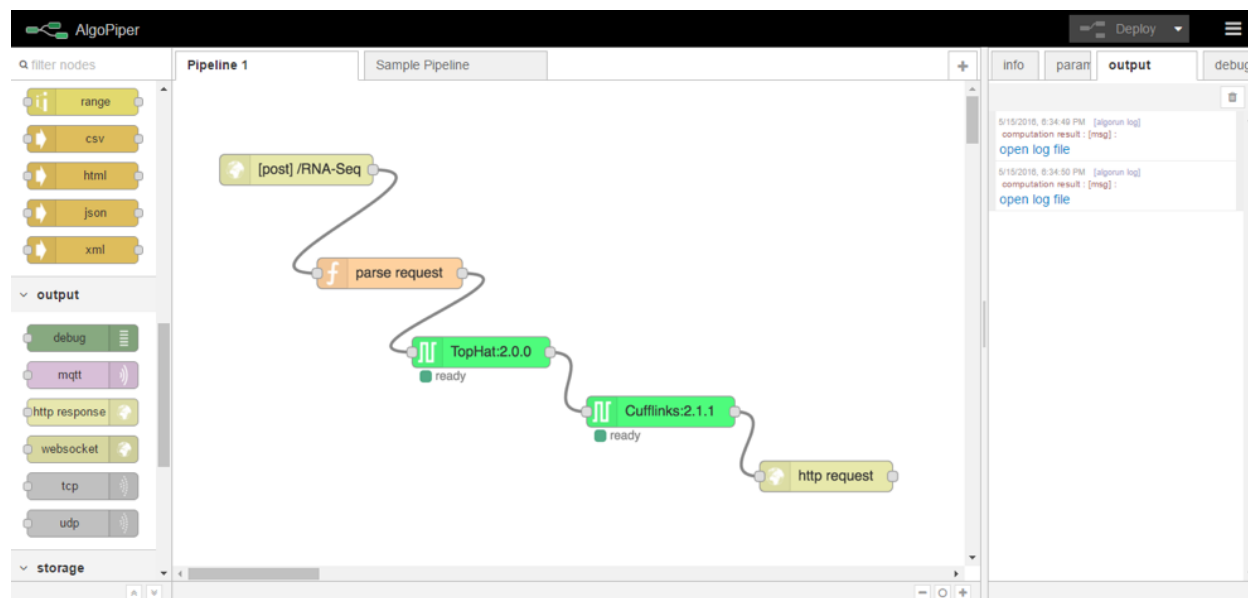


Figure 16 - Integrating RNA-Seq analysis pipeline using HTTP POST endpoint

As an example of running the pipeline through the web API, see the Firefox Poster plugin example below. The web API can be used in any other plugins or programming languages.

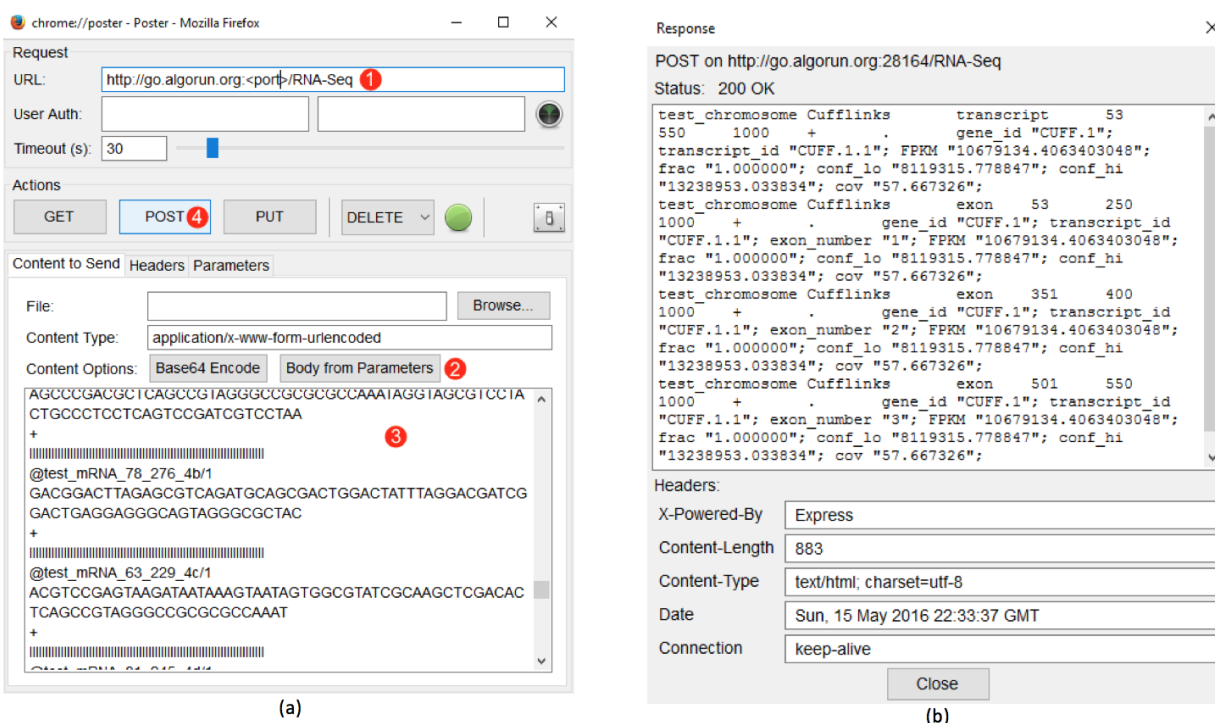


Figure 17 - Submitting RNA sequence reads using AlgoPiper API. (a): (1) Type in the URL of the AlgoPiper followed by /RNA-Seq (2) Choose “Body from Parameters” to send the parameters in the format of form-urlencoded. (3) Type input=<place your input here> in the input area. (4) Click POST to send the request and receive the results. (b): results from running the pipeline via API.

## 5. Share the Pipeline

Now, export the pipeline to a JSON format to save to a local file or share it on AlgoPiper website. Use the mouse to

select all nodes in the pipeline. From the top-right menu, choose **Export** and click **Clipboard**. Copy the JSON text and paste it to a local text file or submit it directly to AlgoPiper website (<http://algotiper.org/submit-pipeline>).

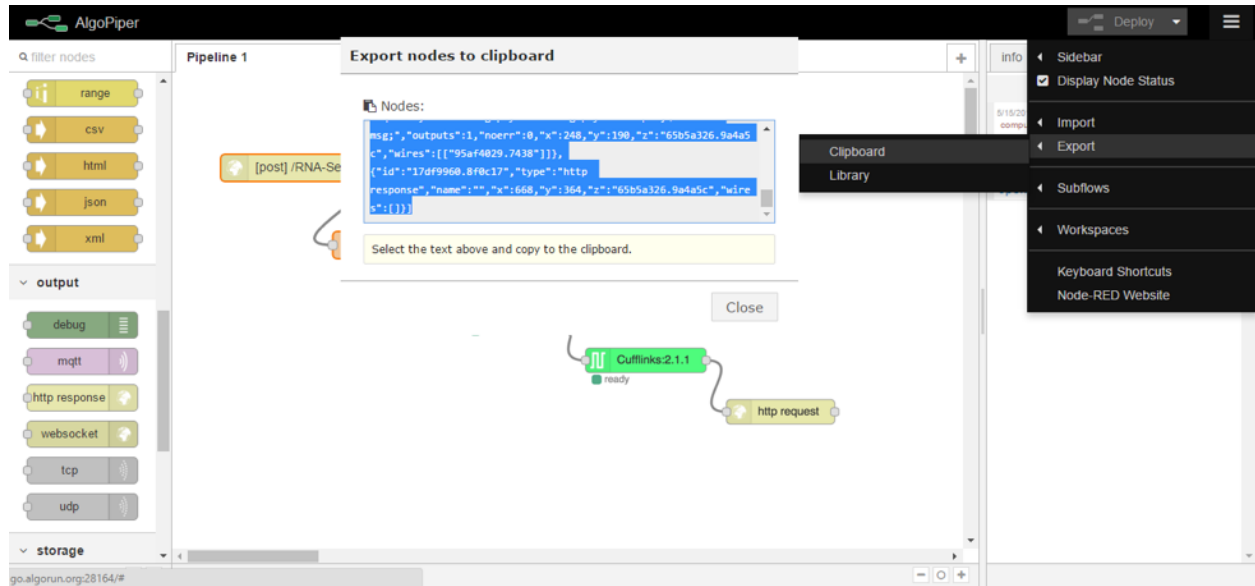


Figure 18 - Sharing RNA-Seq data analysis pipeline





## CHAPTER 4

---

### Local Installation

---

**NOTE:** If you are going to use the online version of AlgoPiper, ignore this section!

Thanks to the distributed architecture of the tool, AlgoManager and AlgoPiper can run on two different machines (locally or remotely).

### Install Prerequisites

The only prerequisite is the Docker Engine: Follow the instructions on: <https://docs.docker.com/engine/installation/>

### Download AlgoManager

1. Clone AlgoManager repository <https://github.com/algorun/algomanager>
2. Navigate to the downloaded folder.
3. Run the script `run.sh`
4. Go to <http://localhost:8000> and make sure it is working

### Configuring the Production Environment

If you want to set AlgoManager on a shared server, edit `algomanager/settings.py` file. Change `SERVER_PATH = 'http://localhost'` to `SERVER_PATH = 'http://server_IP'`

### Add Available Algorithms

Now, let AlgoManager be aware of what algorithms (AlgoRun containers) are available on your machine (or server).

1. Run `docker exec -it algomanager bash`
2. Run `python manage.py createsuperuser`. This will prompt you to create an admin user to manage the available algorithms on this algomanager instance.
3. Now exit this bash using `exit`. Go to <http://localhost:8080/admin/>. Enter your newly created username and password. After you login, click on +Add, right beside Available Algorithms. Enter the name of the algorithm and its AlgoRun container. For example: *Name=REACT* and *Docker Image=algorun/react:latest*. Don't forget to `docker pull` those images from Docker Hub, before you make them available.

## Run AlgoPiper

AlgoPiper is available as a Docker image on Docker Hub.

To run an instance of it, use `docker run -p 8081:8765 -e MANAGER=<algomanager_url> algorun/algopiper`, where `<algomanager_url>` is the url where AlgoManager is running. Now, navigate to <http://localhost:8081> to use it.

Congratulations! You now have a fully working version of AlgoPiper :)

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

“License” shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

“Licensor” shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

“Legal Entity” shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, “control” means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

“You” (or “Your”) shall mean an individual or Legal Entity exercising permissions granted by this License.

“Source” form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

“Object” form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

“Work” shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

“Derivative Works” shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other

modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

“Contribution” shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, “submitted” means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as “Not a Contribution.”

“Contributor” shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. **Grant of Copyright License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. **Grant of Patent License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. **Redistribution.** You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
  - (d) If the Work includes a “NOTICE” text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS



## CHAPTER 6

---

Need Help?

---

Please contact Abdelrahman Hosny at [abdelrahman.hosny@hotmail.com](mailto:abdelrahman.hosny@hotmail.com)