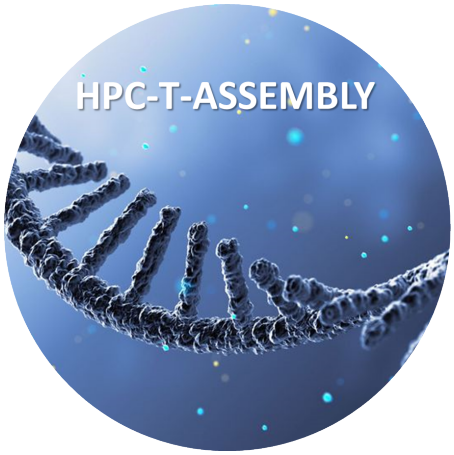
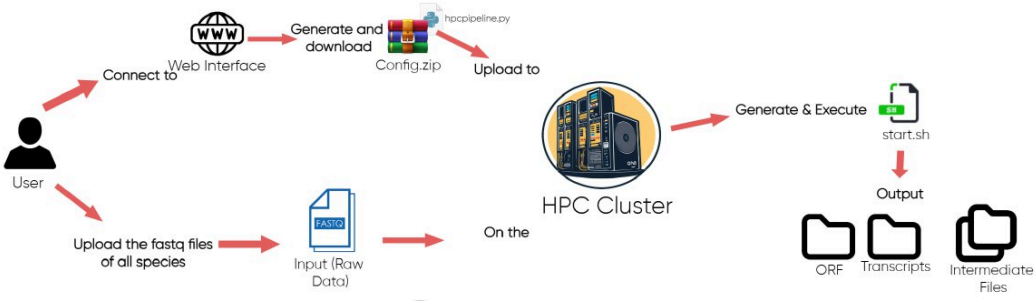


HPC-T-Assembly

High Performance Computing Transcriptome Assembler



HPC-T-Assembly

1) Install Flask

If running the GUI locally install flask using pip.

```
pip install flask
```

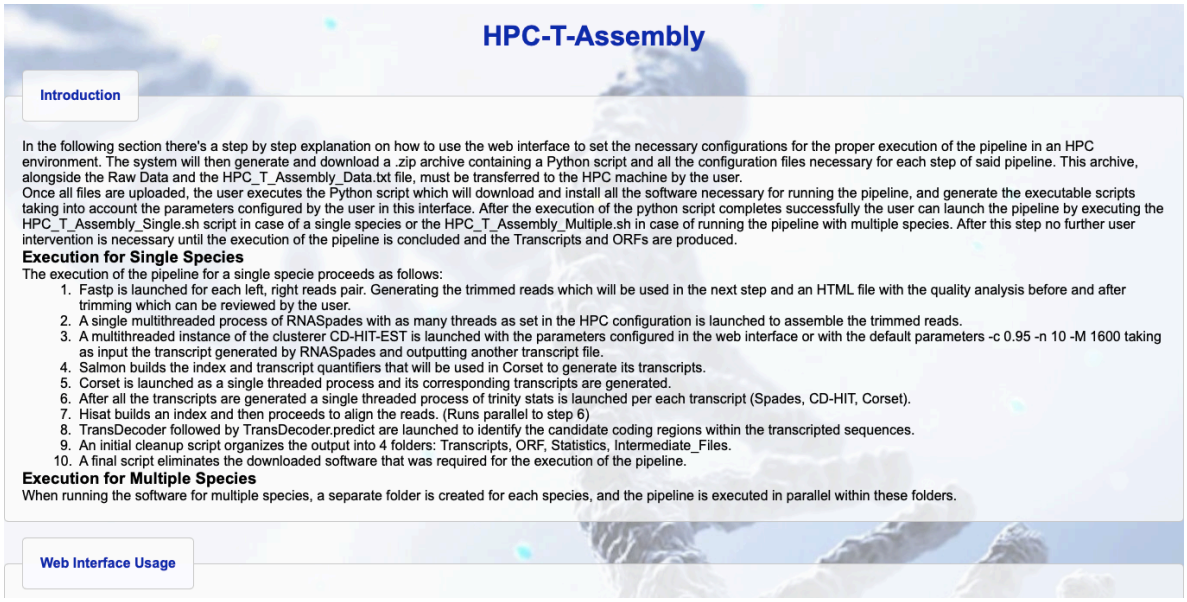
2) Generate Configuration Files

1. Launch HPC_T_Assembly_Configuration.py

```
python HPC_T_Assembly_Configuration.py
```

2. Open a web browser and navigate to the address

```
127.0.0.1:5000
```



HPC-T-Assembly

Introduction

In the following section there's a step by step explanation on how to use the web interface to set the necessary configurations for the proper execution of the pipeline in an HPC environment. The system will then generate and download a .zip archive containing a Python script and all the configuration files necessary for each step of said pipeline. This archive, alongside the Raw Data and the HPC_T_Assembly_Data.txt file, must be transferred to the HPC machine by the user.
Once all files are uploaded, the user executes the Python script which will download and install all the software necessary for running the pipeline, and generate the executable scripts taking into account the parameters configured by the user in this interface. After the execution of the python script completes successfully the user can launch the pipeline by executing the HPC_T_Assembly_Single.sh script in case of a single species or the HPC_T_Assembly_Multiple.sh in case of running the pipeline with multiple species. After this step no further user intervention is necessary until the execution of the pipeline is concluded and the Transcripts and ORFs are produced.

Execution for Single Species

The execution of the pipeline for a single specie proceeds as follows:

1. Fastp is launched for each left, right reads pair. Generating the trimmed reads which will be used in the next step and an HTML file with the quality analysis before and after trimming which can be reviewed by the user.
2. A single multithreaded process of RNASpades with as many threads as set in the HPC configuration is launched to assemble the trimmed reads.
3. A multithreaded instance of the clusterer CD-HIT-EST is launched with the parameters configured in the web interface or with the default parameters -c 0.95 -n 10 -M 1600 taking as input the transcript generated by RNASpades and outputting another transcript file.
4. Salmon builds the index and transcript quantifiers that will be used in Corset to generate its transcripts.
5. Corset is launched as a single threaded process and its corresponding transcripts are generated.
6. After all the transcripts are generated a single threaded process of trinity stats is launched per each transcript (Spades, CD-HIT, Corset).
7. Hisat builds an index and then proceeds to align the reads. (Runs parallel to step 6)
8. TransDecoder followed by TransDecoder.predict are launched to identify the candidate coding regions within the transcripted sequences.
9. An initial cleanup script organizes the output into 4 folders: Transcripts, ORF, Statistics, Intermediate_Files.
10. A final script eliminates the downloaded software that was required for the execution of the pipeline.

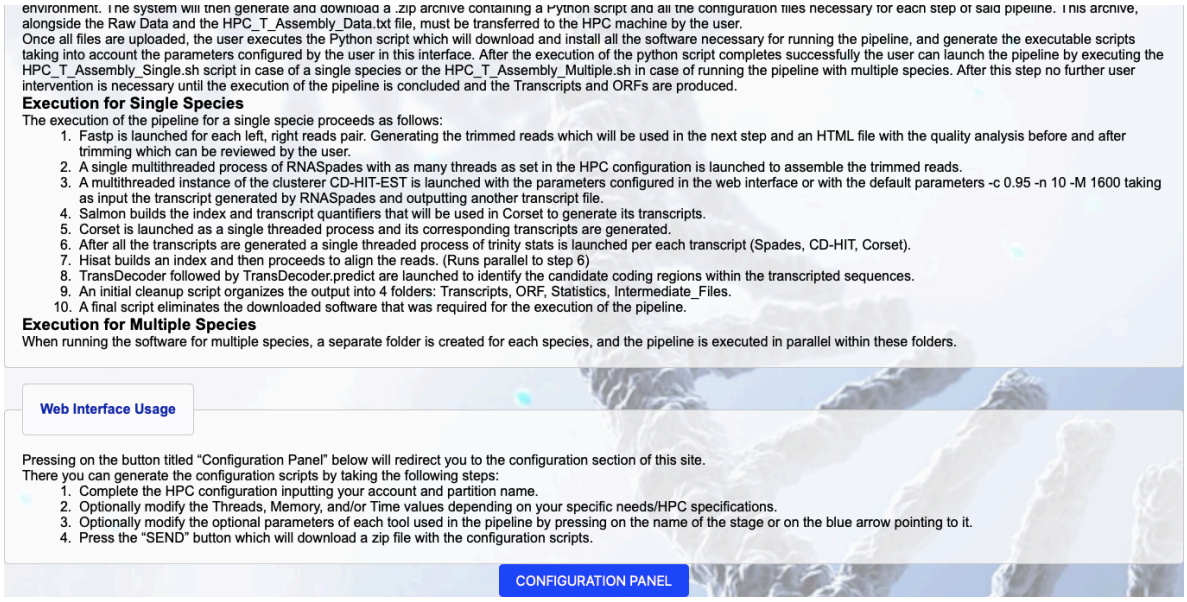
Execution for Multiple Species

When running the software for multiple species, a separate folder is created for each species, and the pipeline is executed in parallel within these folders.

Web Interface Usage

Introduction

3. Scroll down and press "CONFIGURATION PANEL"



environment. The system will then generate and download a .zip archive containing a Python script and all the configuration files necessary for each step of said pipeline. This archive, alongside the Raw Data and the HPC_T_Assembly_Data.txt file, must be transferred to the HPC machine by the user.
Once all files are uploaded, the user executes the Python script which will download and install all the software necessary for running the pipeline, and generate the executable scripts taking into account the parameters configured by the user in this interface. After the execution of the python script completes successfully the user can launch the pipeline by executing the HPC_T_Assembly_Single.sh script in case of a single species or the HPC_T_Assembly_Multiple.sh in case of running the pipeline with multiple species. After this step no further user intervention is necessary until the execution of the pipeline is concluded and the Transcripts and ORFs are produced.

Execution for Single Species

The execution of the pipeline for a single specie proceeds as follows:

1. Fastp is launched for each left, right reads pair. Generating the trimmed reads which will be used in the next step and an HTML file with the quality analysis before and after trimming which can be reviewed by the user.
2. A single multithreaded process of RNASpades with as many threads as set in the HPC configuration is launched to assemble the trimmed reads.
3. A multithreaded instance of the clusterer CD-HIT-EST is launched with the parameters configured in the web interface or with the default parameters -c 0.95 -n 10 -M 1600 taking as input the transcript generated by RNASpades and outputting another transcript file.
4. Salmon builds the index and transcript quantifiers that will be used in Corset to generate its transcripts.
5. Corset is launched as a single threaded process and its corresponding transcripts are generated.
6. After all the transcripts are generated a single threaded process of trinity stats is launched per each transcript (Spades, CD-HIT, Corset).
7. Hisat builds an index and then proceeds to align the reads. (Runs parallel to step 6)
8. TransDecoder followed by TransDecoder.predict are launched to identify the candidate coding regions within the transcripted sequences.
9. An initial cleanup script organizes the output into 4 folders: Transcripts, ORF, Statistics, Intermediate_Files.
10. A final script eliminates the downloaded software that was required for the execution of the pipeline.

Execution for Multiple Species

When running the software for multiple species, a separate folder is created for each species, and the pipeline is executed in parallel within these folders.

Web Interface Usage

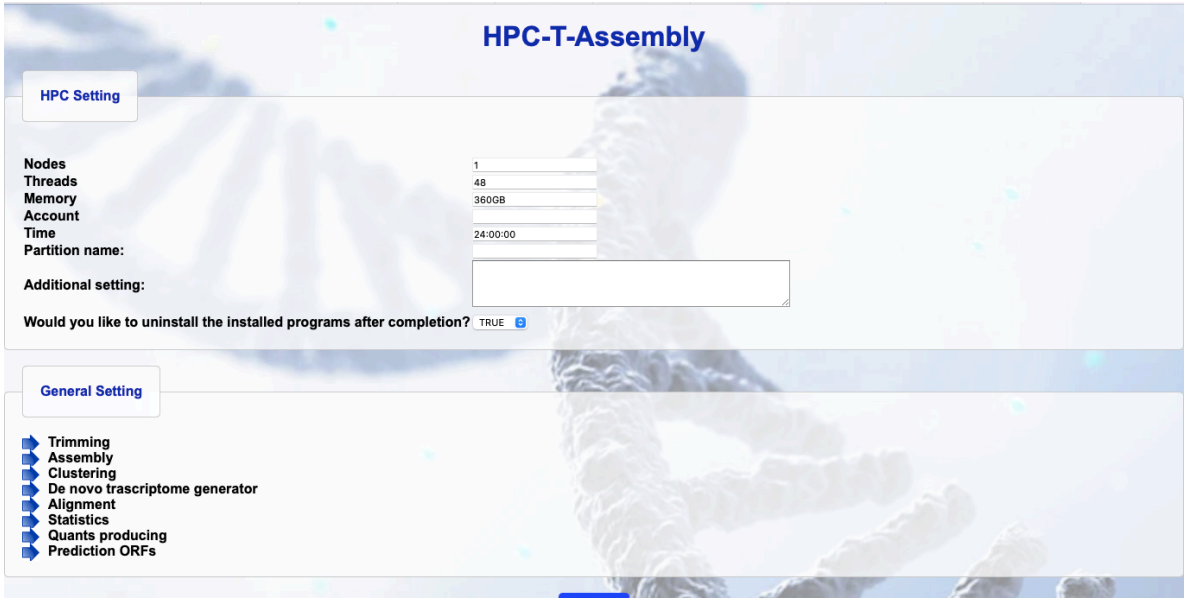
Pressing on the button titled "Configuration Panel" below will redirect you to the configuration section of this site.
There you can generate the configuration scripts by taking the following steps:

1. Complete the HPC configuration inputting your account and partition name.
2. Optionally modify the Threads, Memory, and/or Time values depending on your specific needs/HPC specifications.
3. Optionally modify the optional parameters of each tool used in the pipeline by pressing on the name of the stage or on the blue arrow pointing to it.
4. Press the "SEND" button which will download a zip file with the configuration scripts.

CONFIGURATION PANEL

Configuration panel

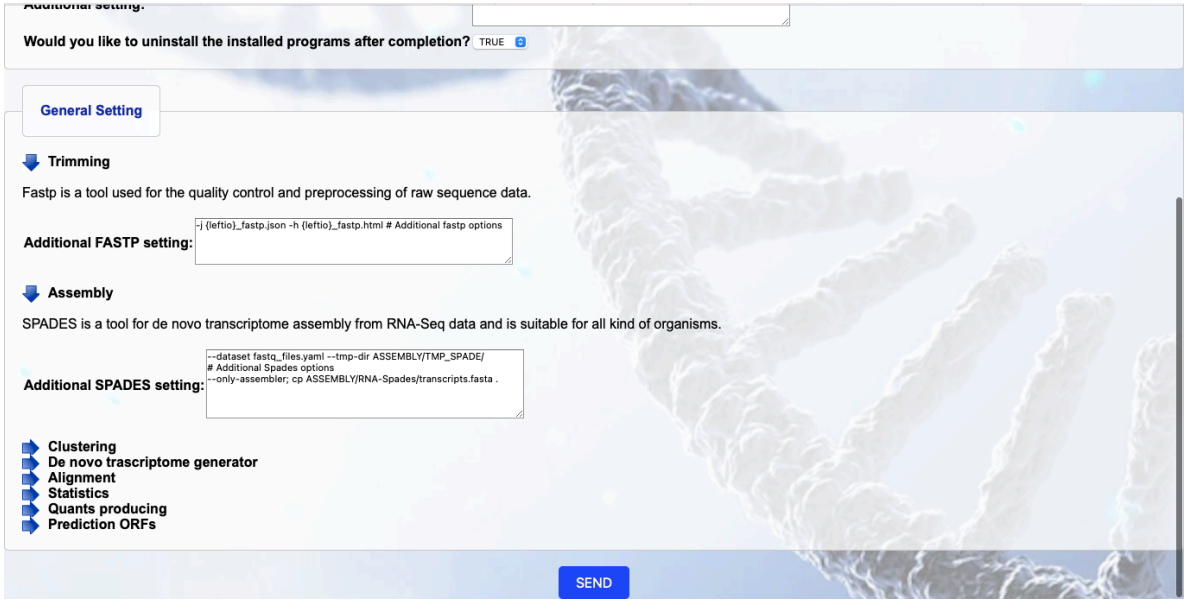
4. Complete the account and Partition values, modify the other values at will.



Hpc parameters

6. Optionally, modify or add additional script parameters

7. Press 'SEND' to download a zip file containing the configuration files and the HPC_T_Assembly.py file



Step configuration

3) Generate Script Files

1. Connect to your HPC via sftp (or through a GUI like Filezilla) and upload the file

2. Unzip Config.zip and transfer all config.txt files into a folder called Config

unzip Config.zip
mkdir Config
mv *.config.txt Config

3. Create a file called HPC_T_Assembly_Data.txt, containing the absolute paths to your left and right reads separated by ",".

If running with multiple species use the '#' followed by the name of the specie to separate each specie as seen bellow

```
MacBook-Air-de-Taiel:Downloads Taiel$ cat HPC_T_Assembly_Data.txt
#Ankistrodesmus sp. EHY
/Data/SRR21282137_1.fastq,/Data/SRR21282137_2.fastq
/Data/SRR21282136_1.fastq,/Data/SRR21282136_2.fastq
/Data/SRR21282135_1.fastq,/Data/SRR21282135_2.fastq
#Tetraedron minutum
/Data/SRR1174749_1.fastq,/Data/SRR1174749_2.fastq
/Data/SRR3478626_1.fastq,/Data/SRR3478626_2.fastq
/Data/SRR3478627_1.fastq,/Data/SRR3478627_2.fastq
#Tetraselmis chuii
/Data/SRR1296875_1.fastq,/Data/SRR1296875_2.fastq
/Data/ERR12708798_1.fastq,/Data/ERR12708798_2.fastq
MacBook-Air-de-Taiel:Downloads Taiel$
```

Hpc t assembly data

4. Execute the following command to generate the script files.

```
python HPC_T_Assembly.py
```

4) Launch Pipeline

1. Once all the previous steps have been completed we can launch the pipeline with the command:

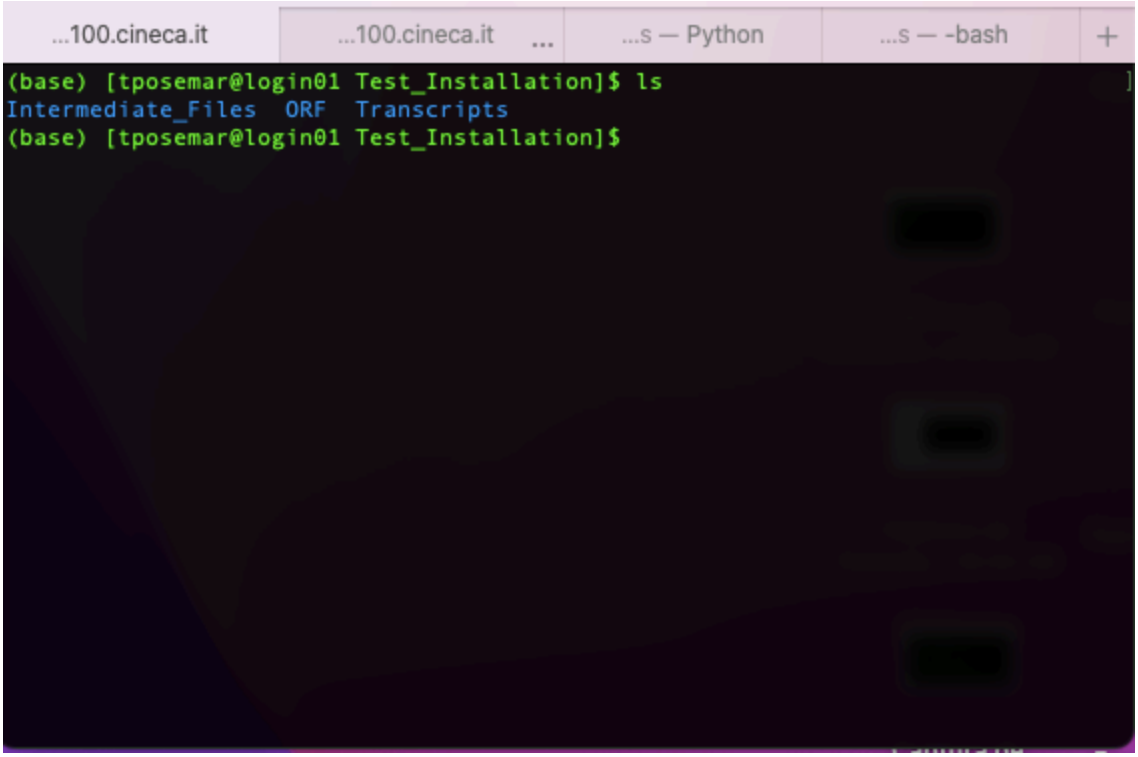
If single specie: `bash HPC_T_Assembly_Single.sh`

If multiple species: `bash HPC_T_Assembly_Multiple.sh`

Final Result

The last script that executes is a cleanup script that organizes the output in the following way:

Transcripts	Contains transcripts from Corset, CD-HIT, and RNASpades
ORF	Contains ORF Predictions
Intermediate Files	Contains all the Intermediate files (Assembly, logs, statistics, etc
Statistics	Contains statistics from the transcript files



Output