
StarBLAST

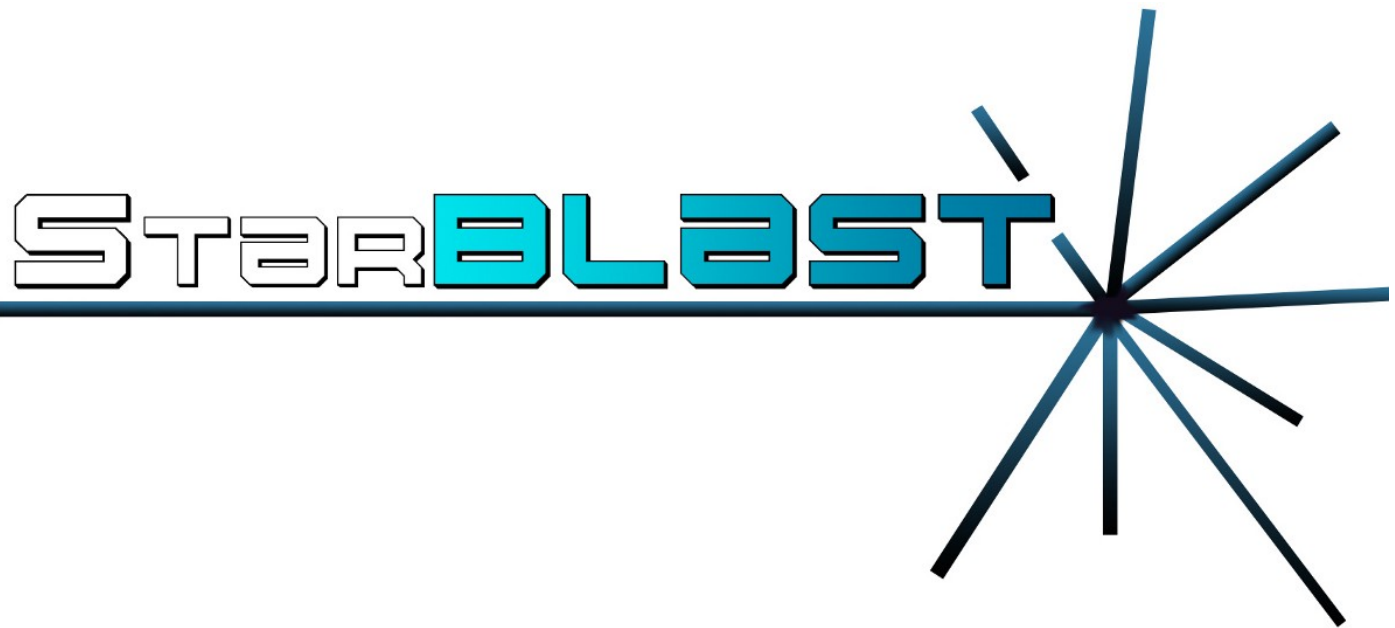
Release 1.0.0

Jun 02, 2021

| | | |
|----------|---|-----------|
| 1 | Welcome to StarBLAST! | 1 |
| 1.1 | What is SequenceServer? | 1 |
| 1.2 | The StarBLAST Suite | 2 |
| 1.3 | Contacts & Issues | 2 |
| 1.4 | Official Publication & Citation | 2 |
| 2 | StarBLAST-VICE: Web Deployment for Small Classes (<25) | 3 |
| 2.1 | Launching StarBLAST-VICE with Example Databases | 3 |
| 2.2 | Adding Your Own Databases to StarBLAST-VICE | 6 |
| 2.3 | Launching StarBLAST-VICE with Your Own Databases | 9 |
| 2.4 | Accessing your running Apps | 9 |
| 3 | StarBLAST-Docker: Cloud Deployment for Medium Classes (25-100) | 13 |
| 3.1 | Launching Foreman & Worker Instances | 13 |
| 4 | StarBLAST-HPC: HPC Deployment for Large Classes (>100) | 21 |
| 4.1 | HPC Requirements and Setup | 21 |
| 4.2 | Launching Workers on the HPC | 23 |
| 4.3 | Setting Up the Master VM on the Cloud Service | 24 |
| 5 | Using SequenceServer | 27 |
| 5.1 | SequenceServer's Main Page | 27 |
| 5.2 | BLAST Loading & Results Page | 29 |
| 5.3 | No-hit Example & Further Reading | 32 |

CHAPTER 1

Welcome to StarBLAST!



StarBlast is a scalable extension of [SequenceServer](#) BLAST, making BLAST accessible to educators and researchers running classroom-scale searches concurrently.

StarBLAST utilizes [cctools](#) for faster, distributed computing and CyVerse's Visual Interactive Computing Environment ([VICE](#)).

1.1 What is SequenceServer?

SequenceServer is a front-end implementation of BLAST with improved GUI and customizable database input developed by the Wurmlab at Queen Mary University of London ([Priyam et al., 2019](#)). However, it is limited in its

scaling capabilities and may be difficult to deploy for some people. StarBLAST extends SequenceServer for easier deployment and to scale for a larger number of simultaneous users (e.g., students).

1.2 The StarBLAST Suite

Navigate to each implementation for information on guided deployment:

- [StarBLAST-VICE: Web Deployment for Small Classes \(<25 students\)](#)
- [StarBLAST-Docker: Cloud Deployment for Medium Classes \(25-100 students\)](#)
- [StarBLAST-HPC: HPC Deployment for Large Classes \(>100 students\)](#)

1.3 Contacts & Issues

If you have questions, suggestions or have encountered a problem, please raise an issue on our [GitHub Issues](#) page.

1.4 Official Publication & Citation

The official publication at the Journal of Open Source Education (JOSE): <https://doi.org/10.21105/jose.00102>

Cite our work as: “Cosi, M., Forstedt, J., Gonzalez, E., Xu, Z., Peri, S., Tuteja, R., Blumberg, K., Campbell, T., Merchant, N. & Lyons E. (2021). StarBLAST: a scalable BLAST+ solution for the classroom. Journal Of Open Source Education, 4(38), 102. doi: 10.21105/jose.00102”

StarBLAST-VICE: Web Deployment for Small Classes (<25)

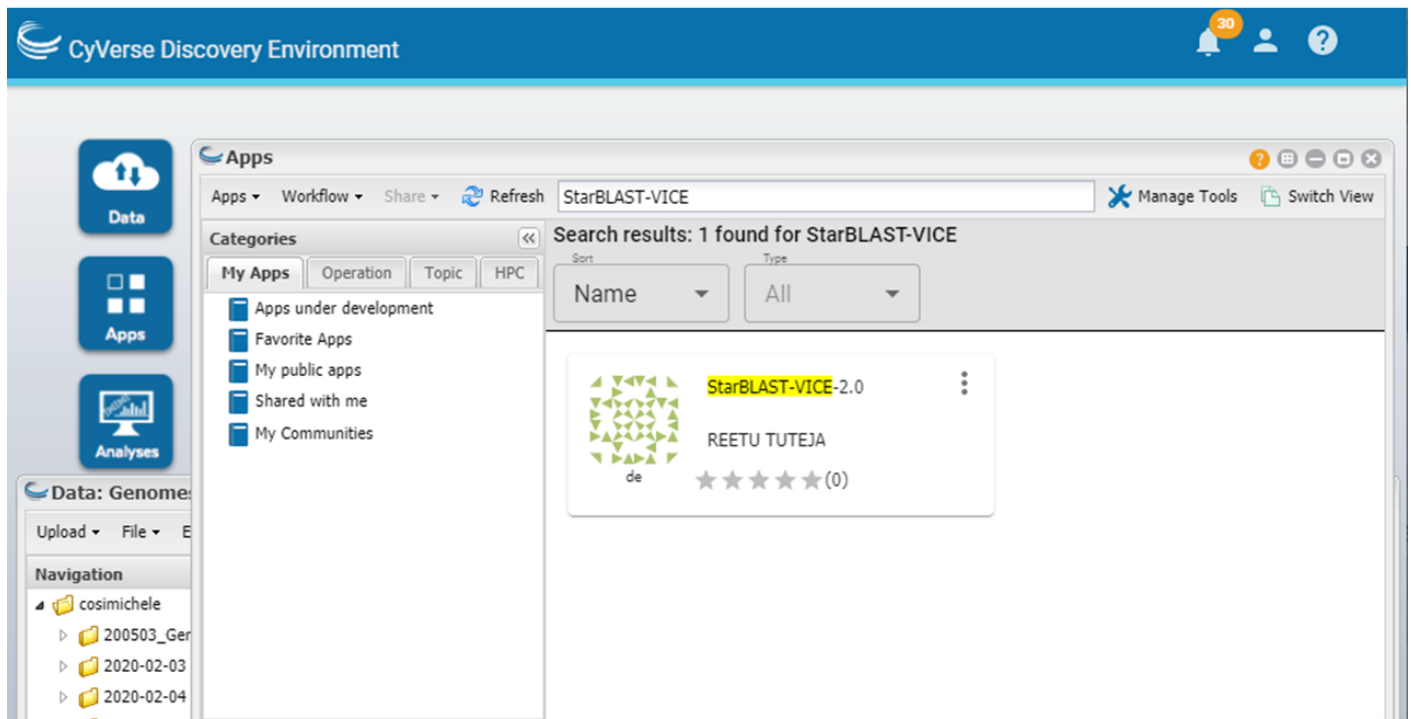
StarBLAST-VICE is a customizable implementation of SequenceServer, deployed as a VICE (Visual and Interactive Computing Environment) web application and hosted on the CyVerse Discovery Environment (DE). StarBLAST-VICE is launchable with a maximum of 8 CPU cores, 16 GB RAM, and 512 GB disk space.

Note: Before proceeding, a CyVerse account is required. Click [here](#) to register or log in.

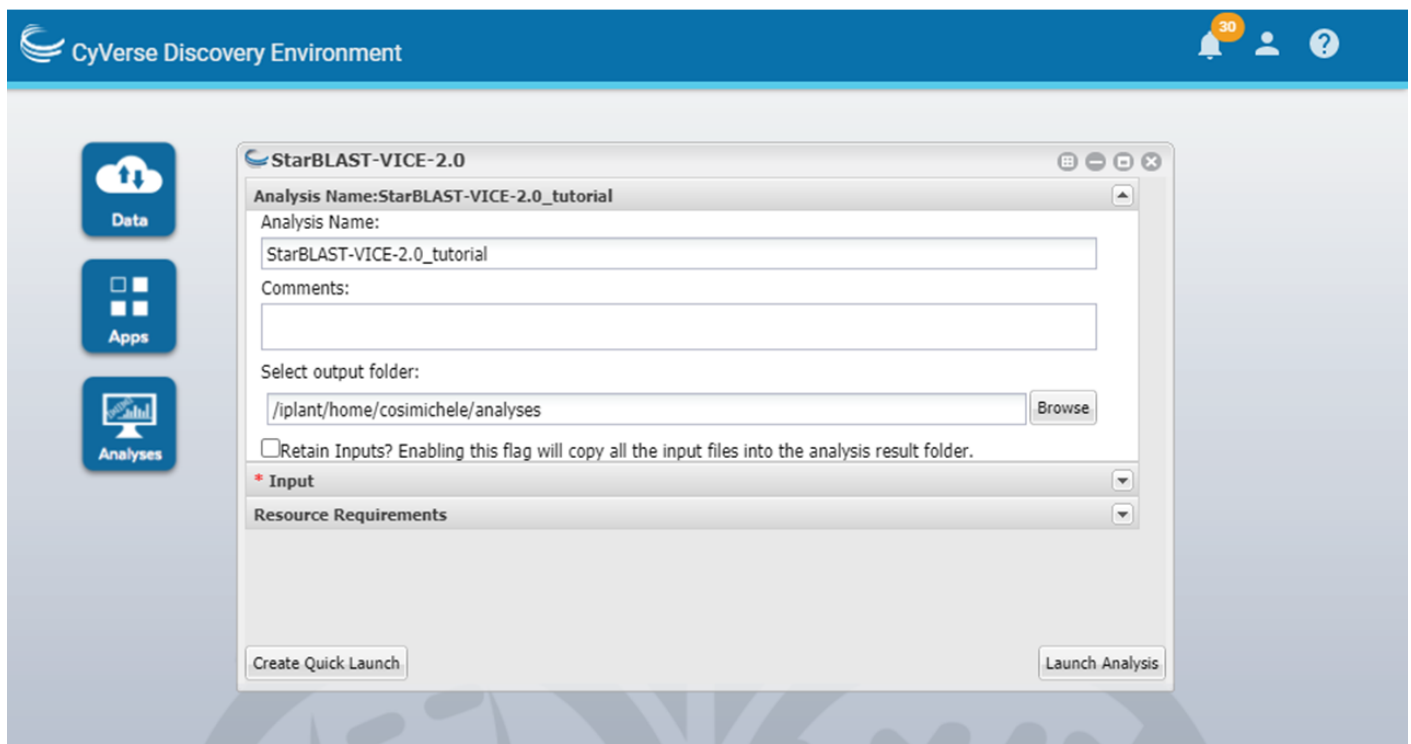
2.1 Launching StarBLAST-VICE with Example Databases

- (1) Click on the following button to launch SequenceServer in CyVerse Discovery Environment with the SWISS-PROT protein database (**requires CyVerse account**). If you are already in the DE, you can navigate to the StarBLAST-VICE app through the Apps button and searching for “StarBLAST-VICE”.

Note: SWISS-PROT is a curated protein sequence database, read more on the [release](#) or its [original publication](#).

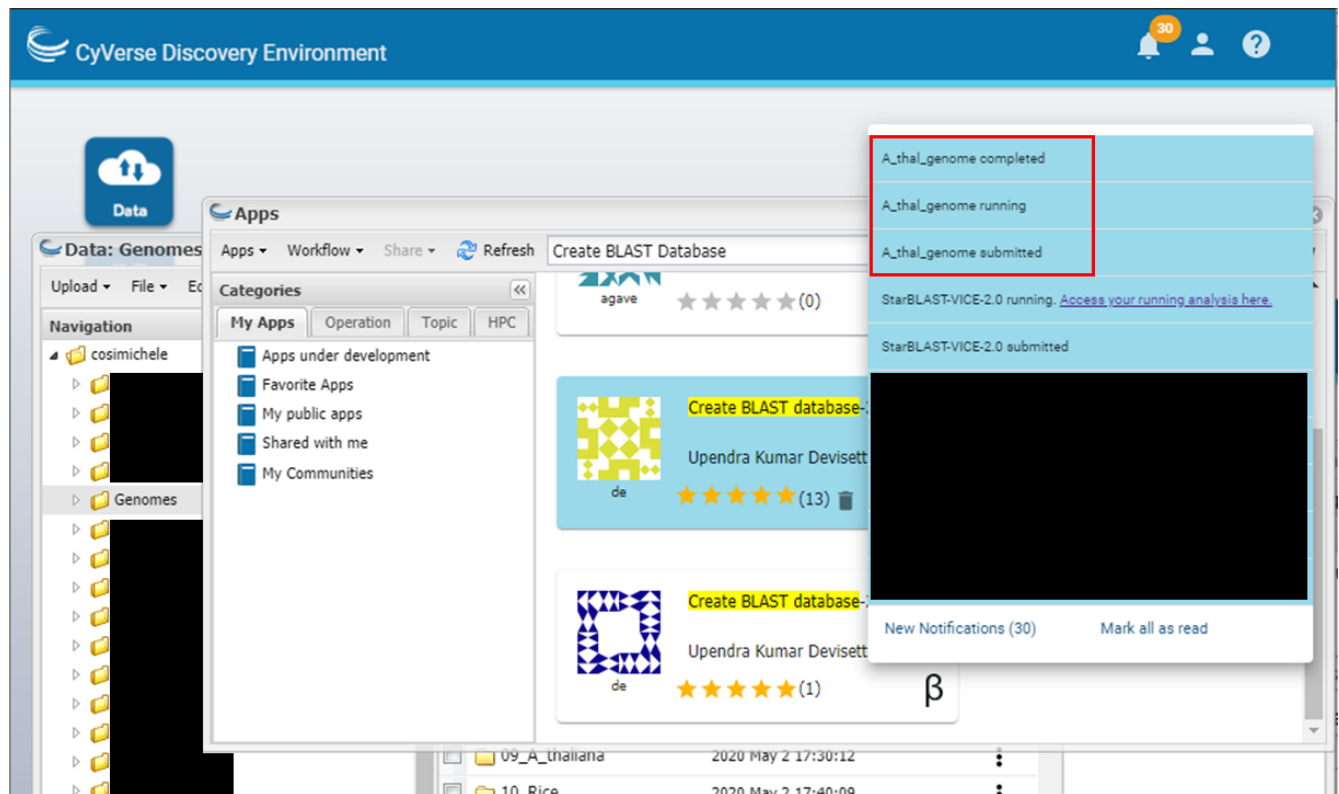


- (2) Choose your own analysis name and the DE output folder. Click “Launch Analysis”.



- (3) Check the notifications Bell Icon for a link to access your SequenceServer instance. This might take a few minutes. Once the notification shows that the app is running, click on the link. This will open a loading screen

in a new tab; Once the app is loaded, you should be able to BLAST through the SequenceServer app.



Loading VICE analysis...

Kubernetes Deployment created.

Kubernetes Service created.

Kubernetes Ingress created.

Kubernetes pod phase: Pending

Paste query sequence(s) or drag file containing query sequence(s) in FASTA format here ...

Protein databases☒ Non-redundant UniProtKB/SwissProt sequences

Advanced parameters:

eg: -evalue 1.0e-5 -num_alignments 100

☐ Open results in new tab

BLAST

Run BLAST locally with Sequenceserver: [Tweet](#)

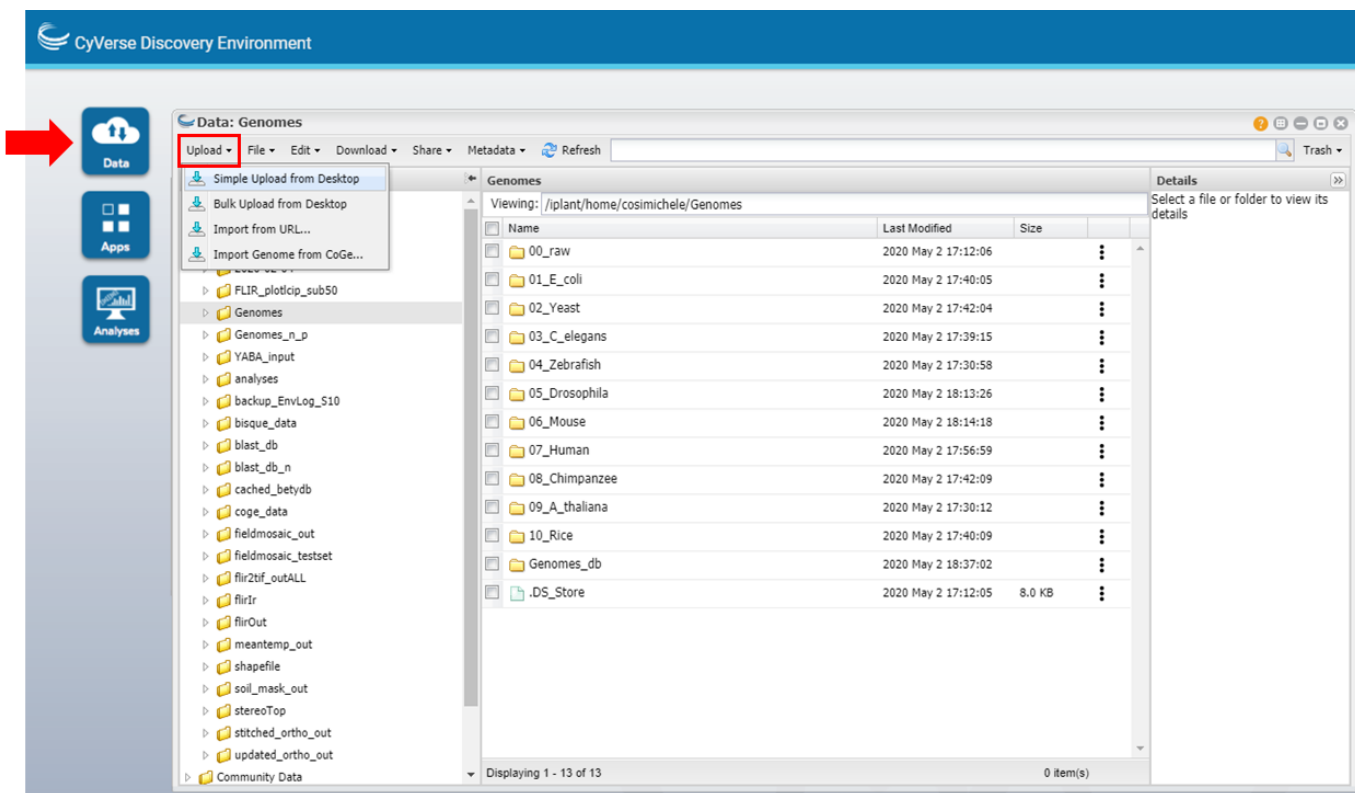
Please cite relevant data sources and: Priyam et al. (2019) Sequenceserver: a modern graphical user interface for custom BLAST databases.

- (4) To test, click [here](#) for a sample DNA sequence.
- (5) Paste the query sequence, select both the available databases and submit job.

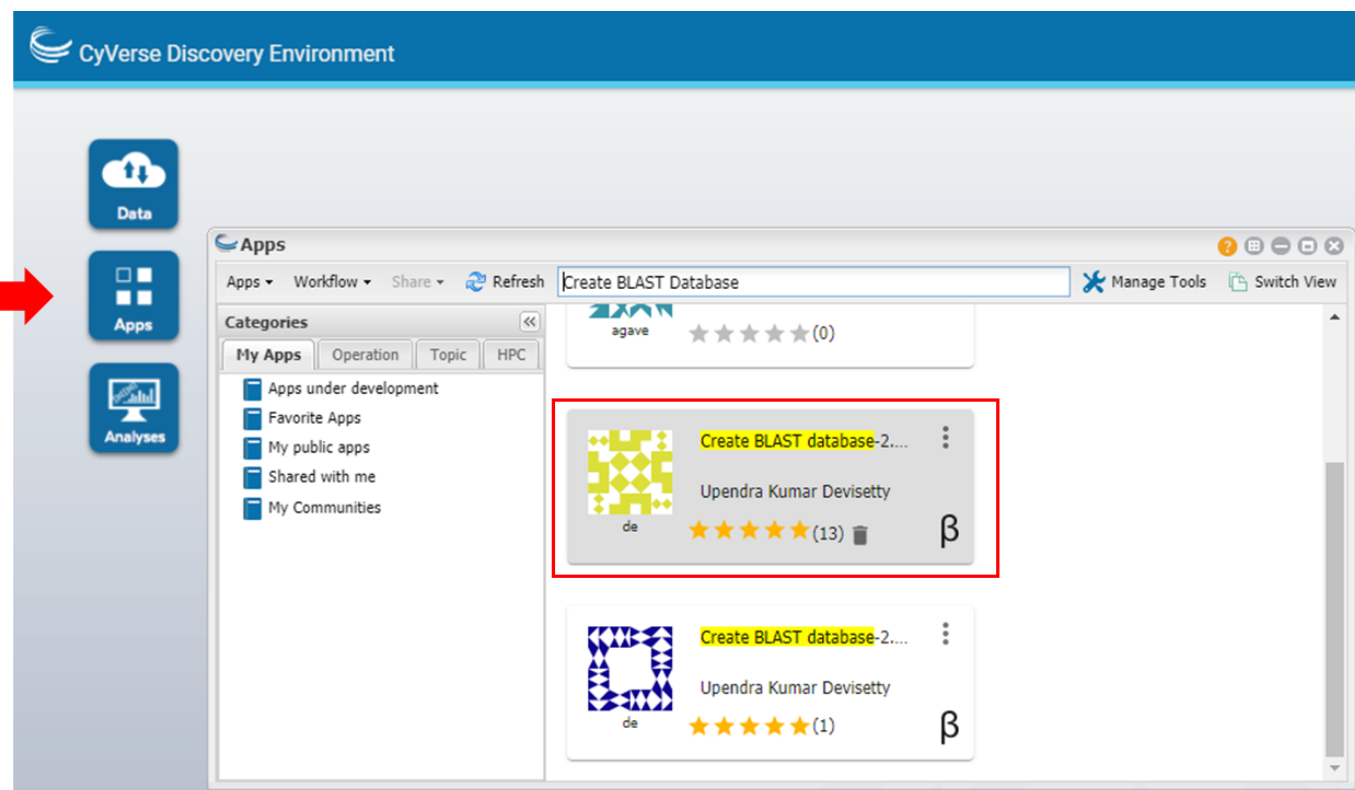
2.2 Adding Your Own Databases to StarBLAST-VICE

To add your own BLAST databases you will need a `.fasta` (or `.fa`, `.faa`, `.fna`) file containing the reference sequences you'd like to use. These are easily aquirable from NCBI or other databases.

- (1) Within the CyVerse DE, click on the “Data” icon.
- (2) Select “Upload” and specify the import from your Desktop or its URL. This will be stored in your personal folder.



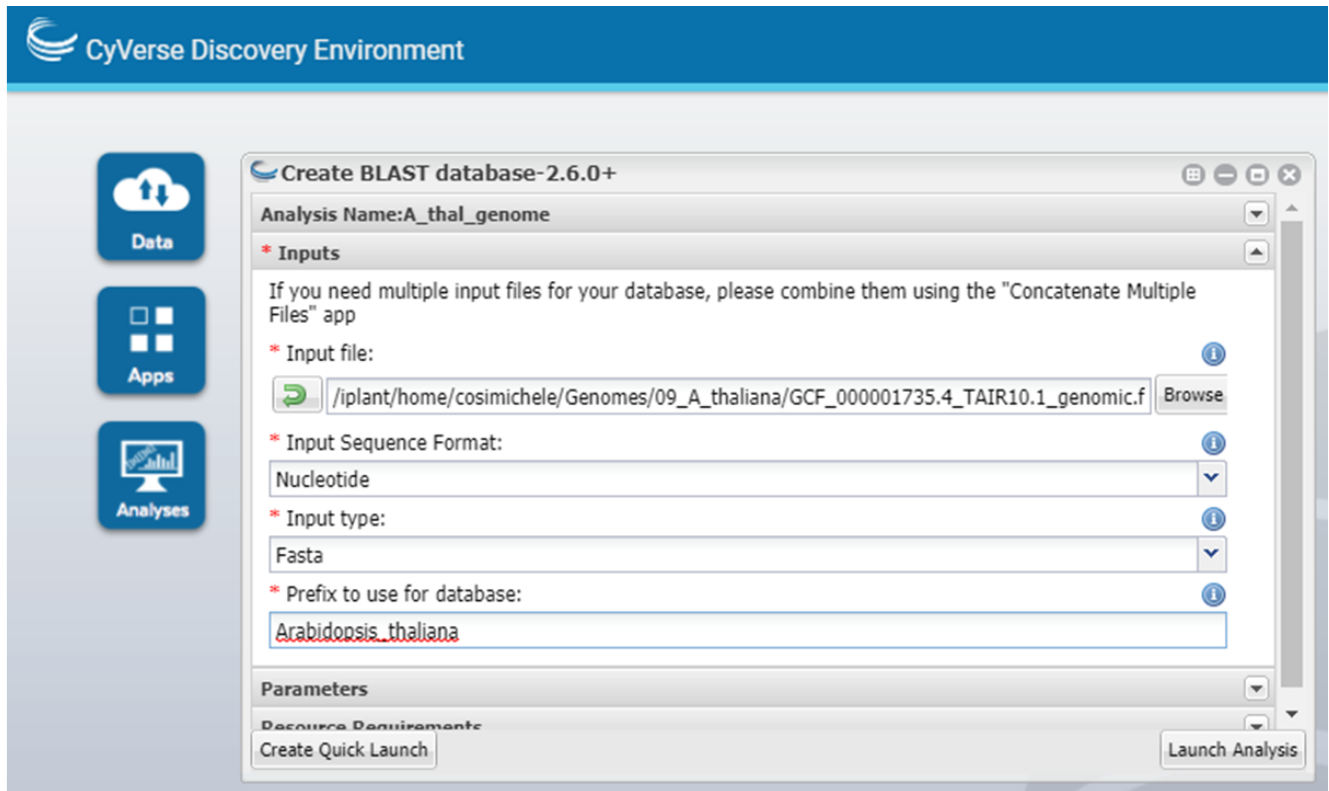
- (3) Click on the “Apps” icon and use the search bar to find “Create BLAST Database” or click [here](#). As there is a possibility that there may be multiple apps with the same name, please locate the “Create BLAST Database” developed by developed by Upendra Kumar Devisetty (use the better reviewed one).



(4) Enter a name for your database under “Analysis Name”; this will become your database containing folder.

1. “Select output folder” should be your personal folder or any folder of your choice (default will be a folder named “analyses” within your personal folder).
2. In the “Inputs” tab, select “Browse” and choose the fasta file you uploaded. Select Nucleotide or Protein under “Input Sequence Format”. Under “Prefix” choose a name to well reflect your database (e.g. a_thaliana).

The screenshot shows the 'Create BLAST database-2.6.0+' window in the CyVerse Discovery Environment. On the left is a sidebar with three icons: 'Data' (cloud with arrows), 'Apps' (four squares), and 'Analyses' (monitor with bar chart). The main window has a title bar with standard OS controls. Below the title bar, the 'Analysis Name' is set to 'A_thal_genome'. There is a text input field for 'Analysis Name' containing 'A_thal_genome' and a larger text area for 'Comments'. Below these is a 'Select output folder:' section with a text input field containing '/iplant/home/cosimichele/analyses' and a 'Browse' button. A checkbox labeled 'Retain Inputs? Enabling this flag will copy all the input files into the analysis result folder.' is currently unchecked. At the bottom, there are three tabs: '* Inputs', 'Parameters', and 'Resource Requirements', each with a dropdown arrow. The 'Inputs' tab is selected. At the very bottom of the window are two buttons: 'Create Quick Launch' and 'Launch Analysis'.



- (5) Click “Launch Analysis” and wait to be notified of its completion. Upon completion, navigate to the output folder specified in step 4.1; Inside you will find a directory with the name you specified in step 4 followed by a timestamp. Within this folder you will find logs and the newly generated database (if nucleotide .nhr, .nin, .nog, .nsd, .nsi, .nsq files will be found; if protein, .phr, .psq files will be found).

2.3 Launching StarBLAST-VICE with Your Own Databases

To launch StarBLAST-VICE with your own database:

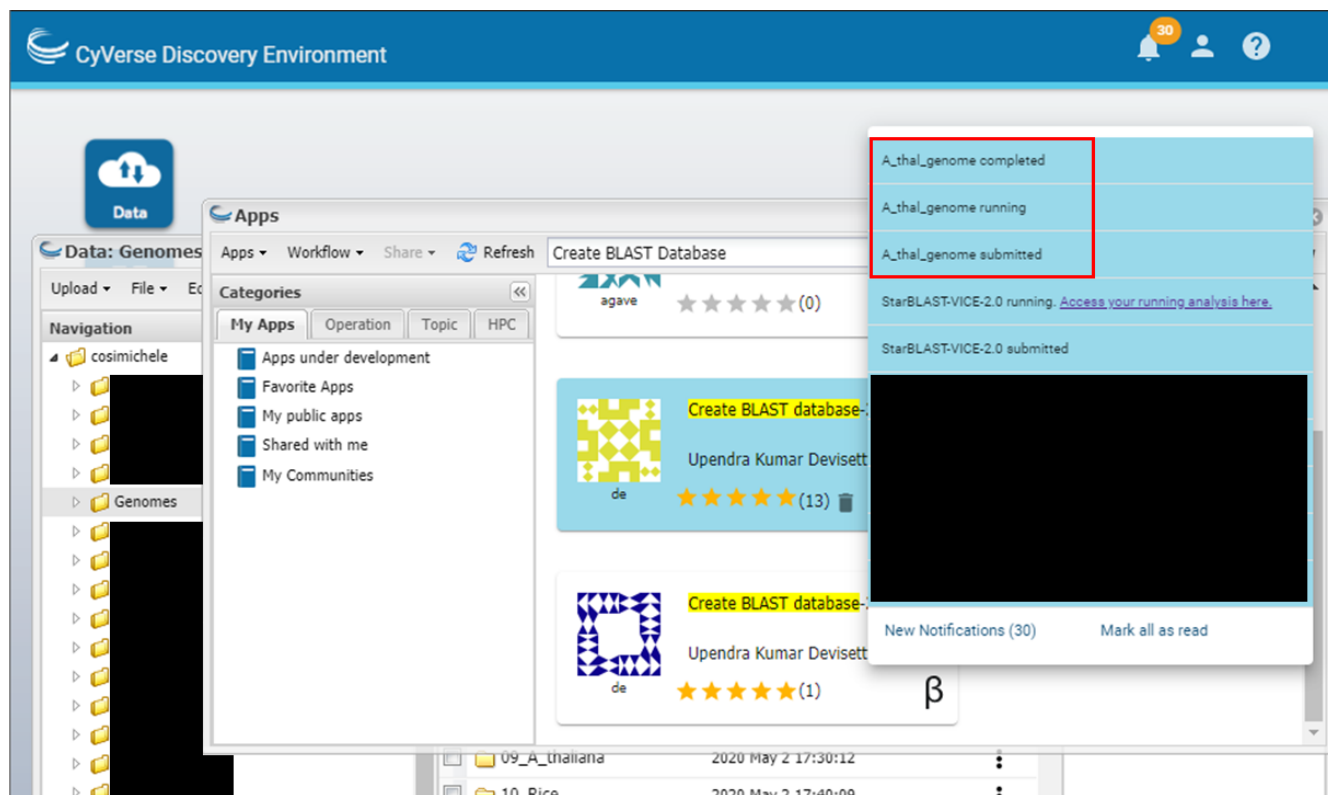
- (1) Use the same button for Example Databases, but **do not click “Launch Analysis” just yet.**
- (2) In the “Input” tab, select the folder containing your database (if not specified, default is `swissprot-db`).

Note: The user will not be able to see the databases within the folder at this step. Ensure that databases files (as explained in step 4.2 of the previous section) are present beforehand.

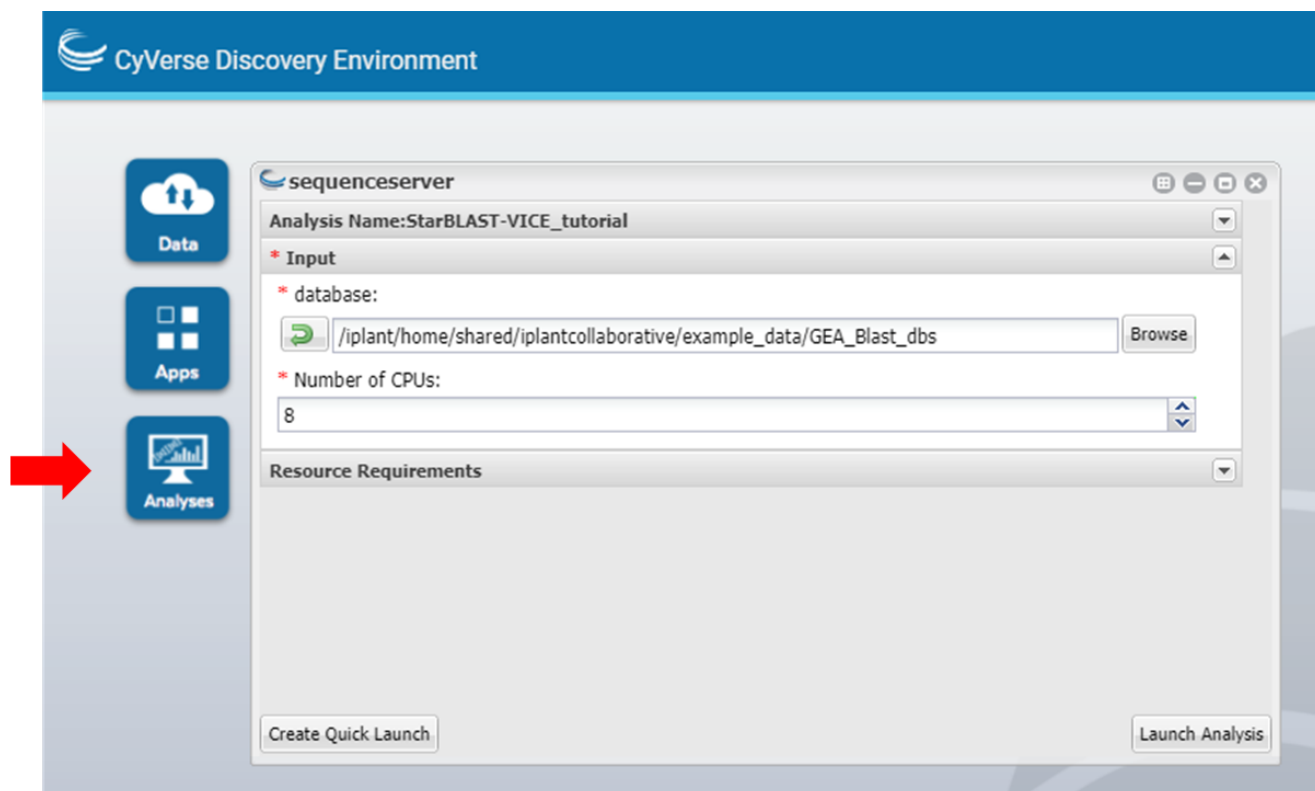
- (3) Click “Launch Analysis”. This might take a few minutes.

2.4 Accessing your running Apps

The notification bell should show your currently running apps and jobs.



In case you want to see all your jobs and access your running apps (and app history), navigate to the Analyses button.



The screenshot displays the CyVerse Discovery Environment interface. At the top, a blue header bar contains the CyVerse logo and the text "CyVerse Discovery Environment". On the right side of the header, there are icons for notifications (a bell with a red "30" badge), a user profile, and a help icon (a question mark).

On the left side, there is a vertical sidebar with three main sections: "Data" (represented by a cloud icon), "Apps" (represented by a grid of squares icon), and "Analyses" (represented by a monitor icon). The "Analyses" section is currently selected.

The main content area shows a window titled "Analyses". Inside this window, there is a toolbar with a "Refresh" button, a "View" dropdown menu set to "All", an "App Type" dropdown menu set to "All", and a search bar labeled "Search...".

Below the toolbar is a table with the following columns: "Name", "Owner", "App", "Start Date", "End Date", and "Status". The table contains two rows of data:

| Name | Owner | App | Start Date | End Date | Status |
|--|---|-----|------------|----------|-----------|
| <input type="checkbox"/> StarBLAST-VICE_tutorial | cosimichelesequenceserver | | | | Running |
| <input type="checkbox"/> A_thal_genome | cosimicheleCreate BLAST database-2.6.0+ | | | | Completed |

Below the table, there is a large black rectangular area, likely a placeholder for a visualization or a detailed view of the selected analysis.

StarBLAST-Docker: Cloud Deployment for Medium Classes (25-100)

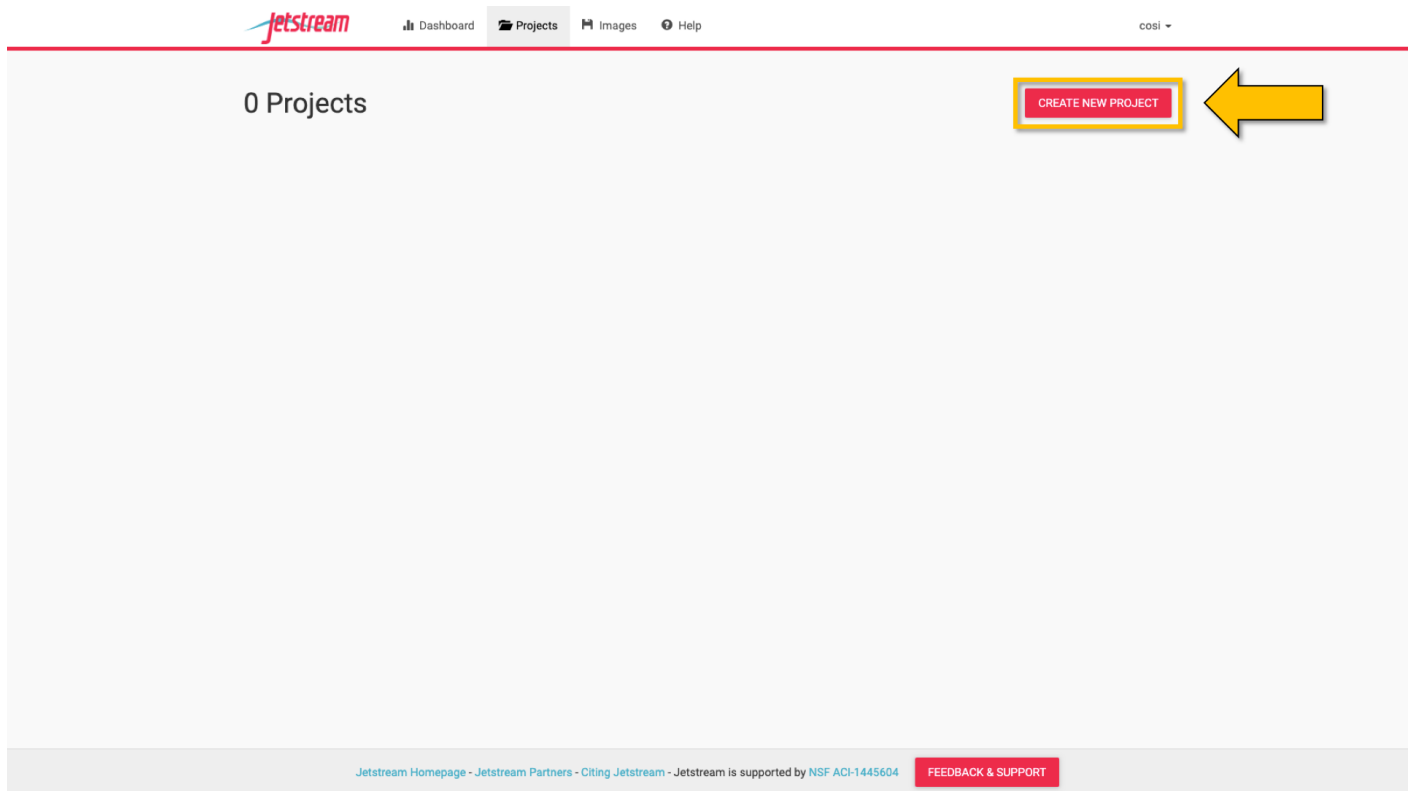
To deploy StarBLAST setup in a cloud provider, you will need accounts with those providers. This example uses [XSEDE's JetStream Cloud service](#). You can access JetStream using an [XSEDE](#) account, a [Globus](#) account, or via institutional access to XSEDE (search for your institution name from the drop down menu in JetStream's login page).

This setup uses a “Foreman” instance for the front-end sequenceServer and one or more “Worker” instances to distribute the computational load of running blast. Docker containers are used to deploy the Foreman and Workers through deployment scripts. These deployment scripts are designed to:

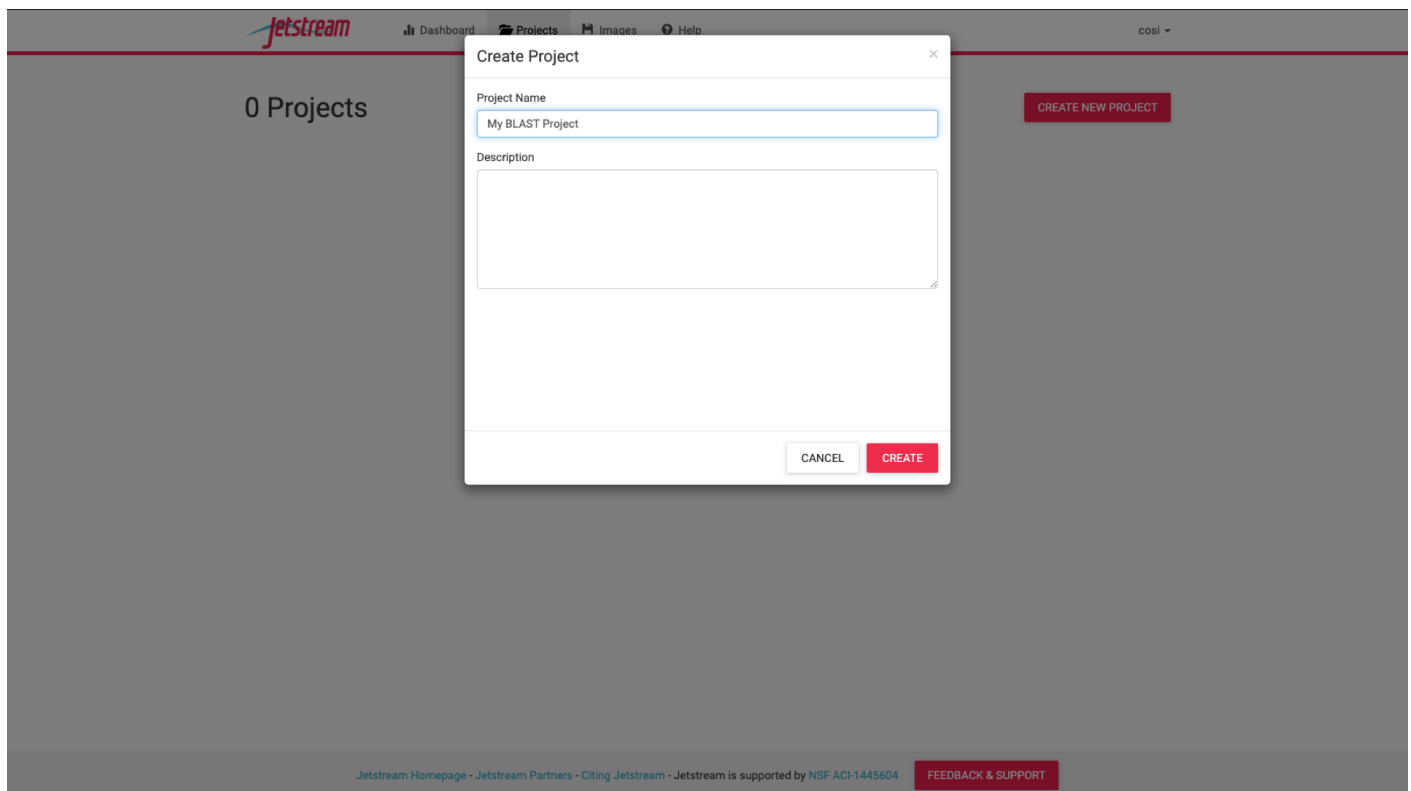
- Get appropriate [Worker](#) and [Foreman](#) containers
- Download specified BLAST databases (can customize)
- Launch sequenceServer front-end that can be accessed using the code: `<FOREMAN_IP_ADDRESS>` of the instance
- Connect factory of workers to the Foreman

3.1 Launching Foreman & Worker Instances

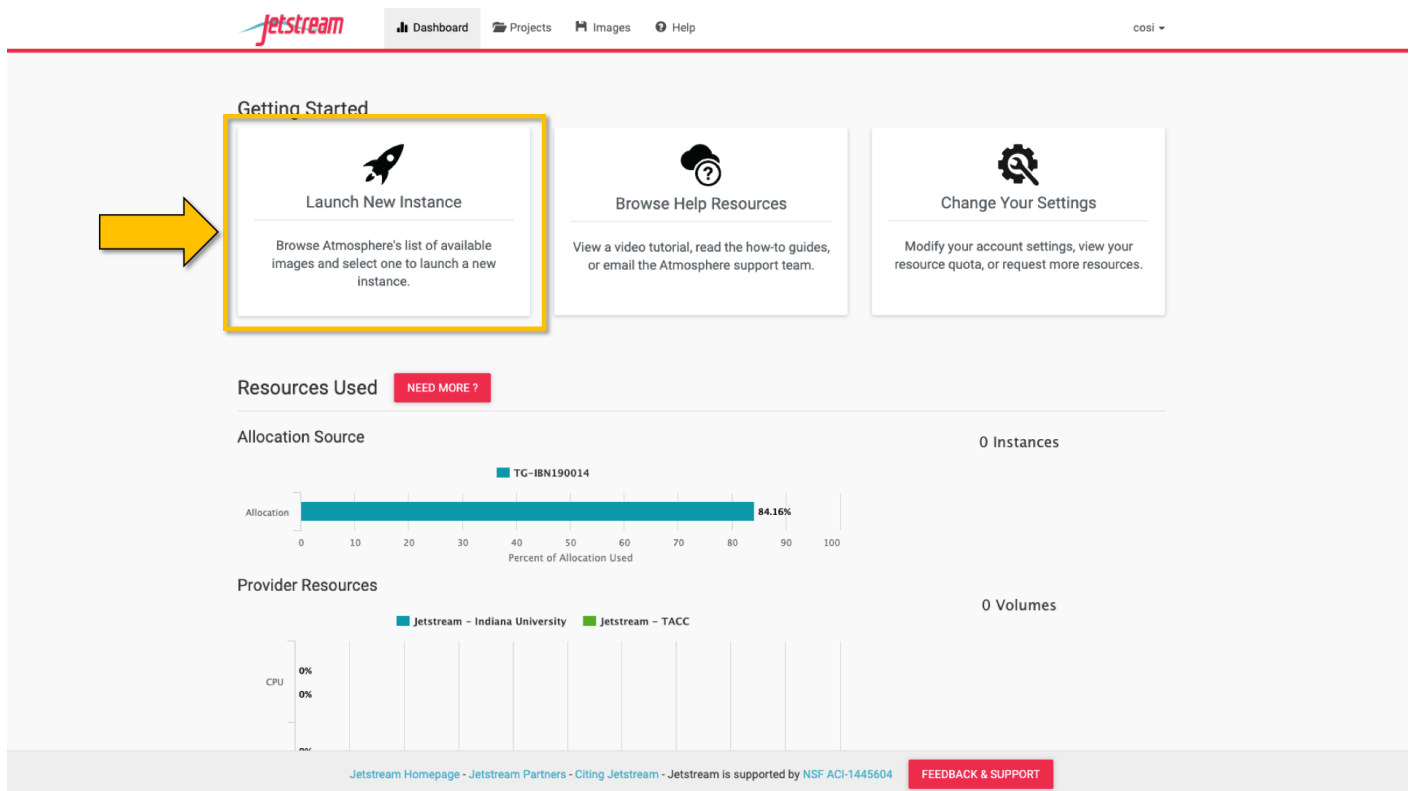
1. Login to [JetStream Cloud](#).
2. From [JetStream's](#) top menu, navigate to “Projects” and select “Create New Project”.



3. In the “Project Name” field, name your project and add a description.

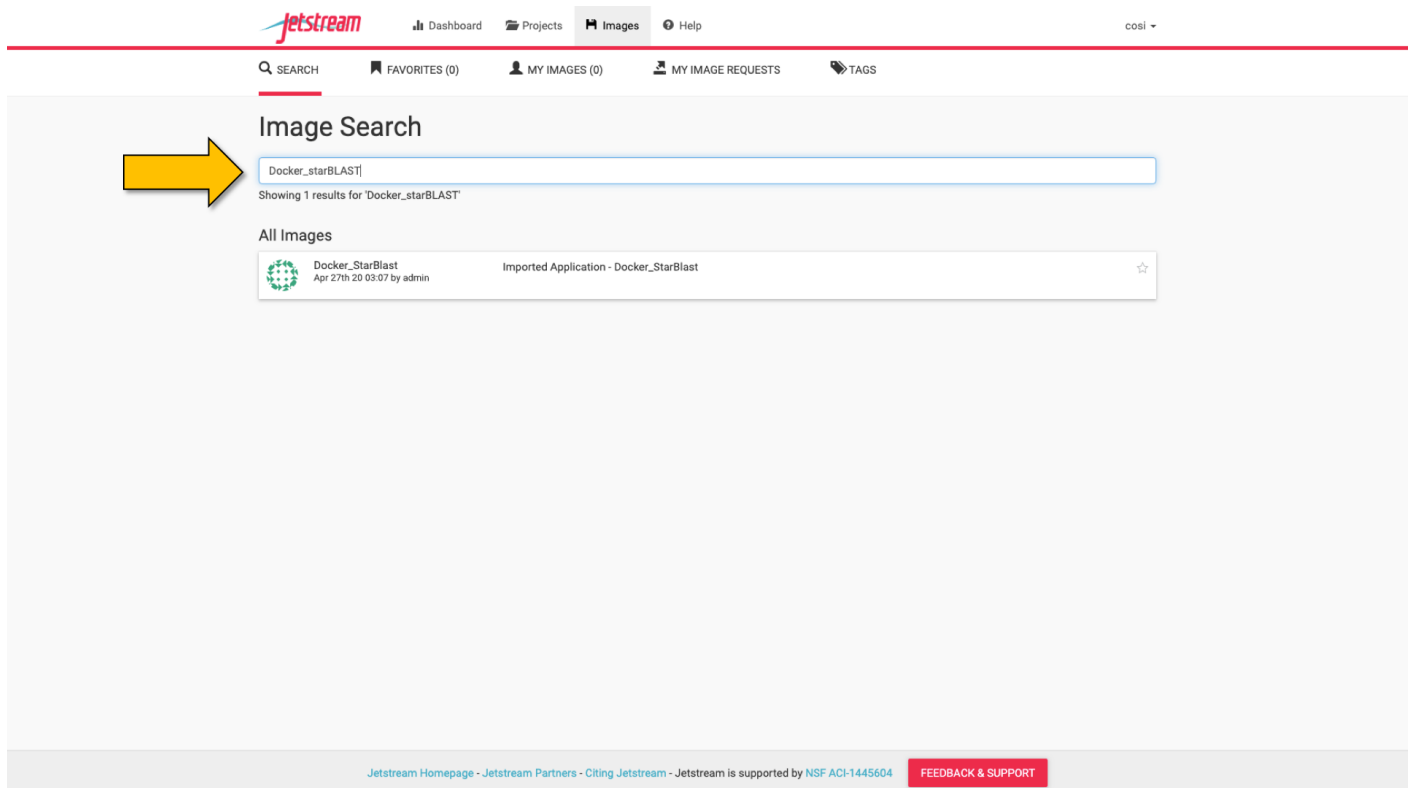


4. From JetStream’s dashboard, select “Launch New Instance”.

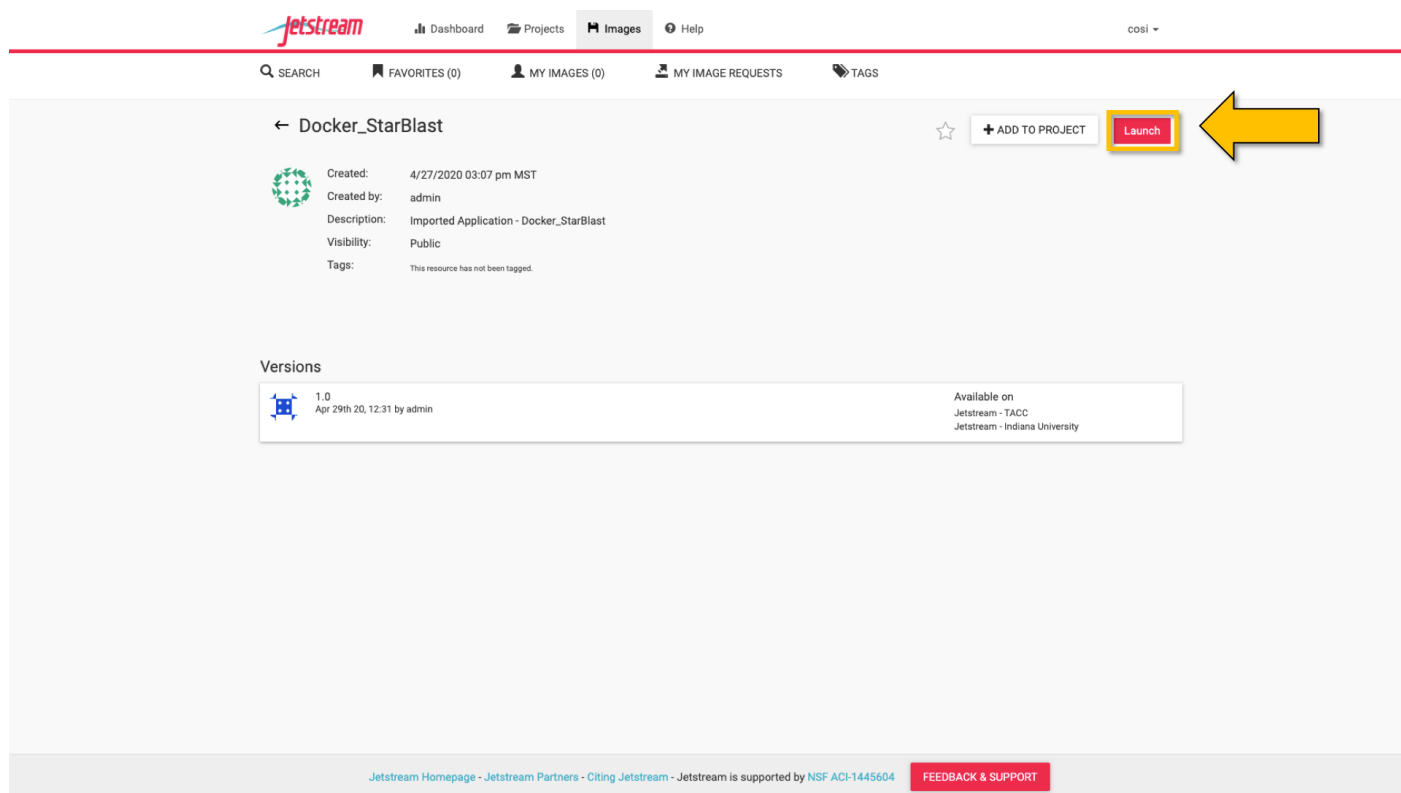


The screenshot shows the Jetstream dashboard. The 'Getting Started' section is highlighted with a yellow box, and a yellow arrow points to the 'Launch New Instance' button. The dashboard includes a navigation bar with 'Dashboard', 'Projects', 'Images', and 'Help'. The 'Resources Used' section shows 'Allocation Source' with a bar chart for 'TG-IBN190014' at 84.16% allocation. The 'Provider Resources' section shows 'Jetstream - Indiana University' and 'Jetstream - TACC' with a bar chart for 'CPU' at 0%.

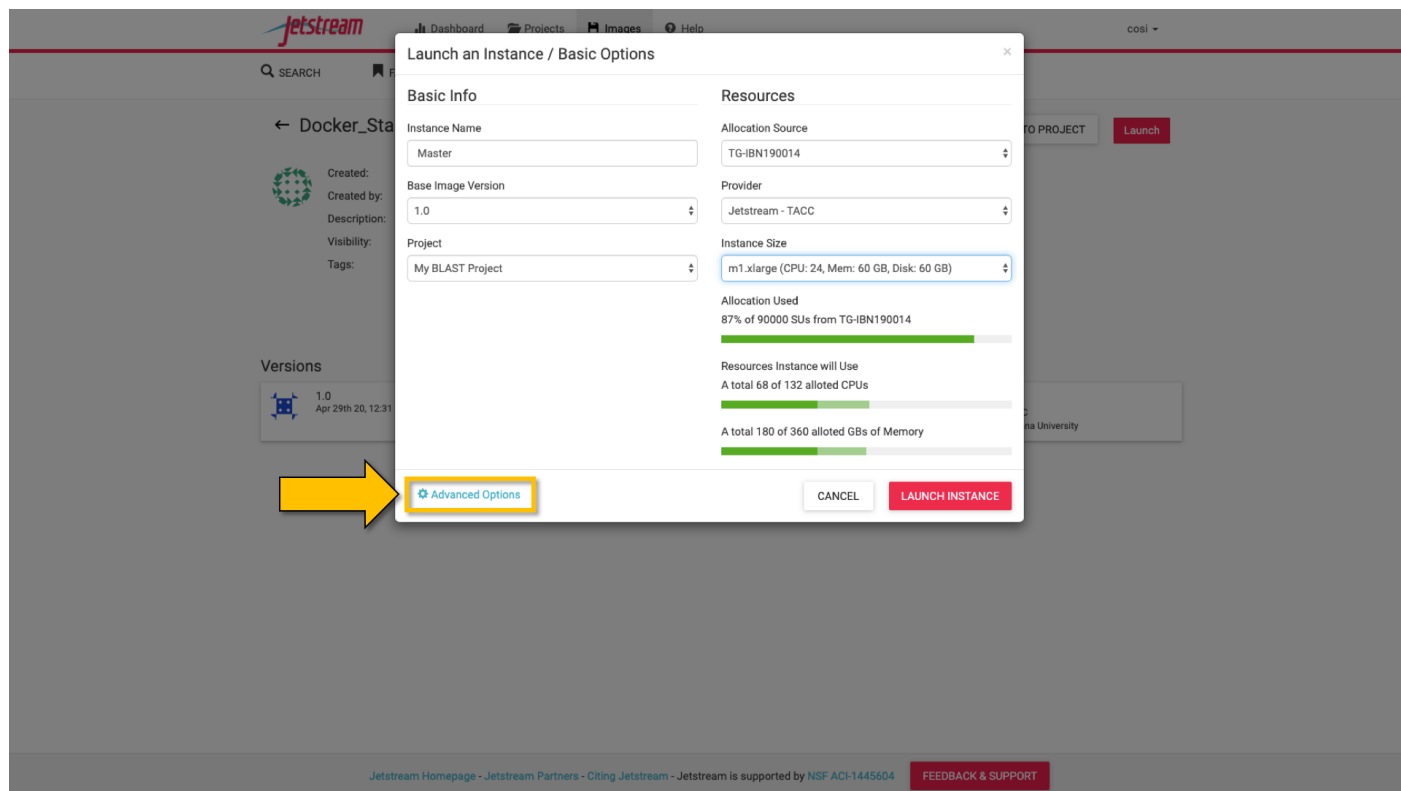
5. Be sure to change the default tab from “Show Featured” to “Show All”, search for “Docker_starBLAST” and select the “Docker_starBLAST” image (or click [here](#)); click “Launch”.



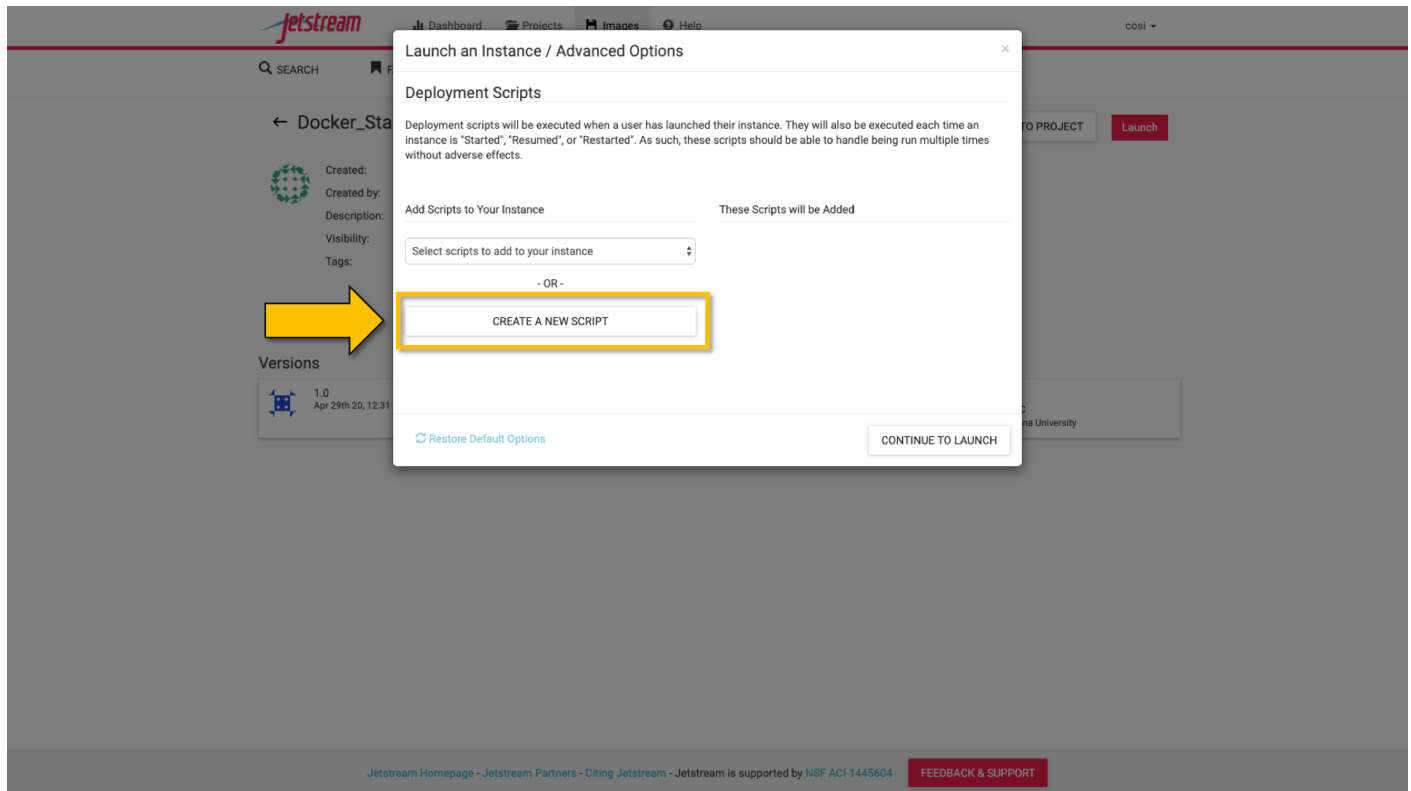
The screenshot shows the Jetstream 'Image Search' page. A yellow arrow points to the search bar containing 'Docker_starBLAST'. The search results show one result: 'Docker_StarBlast' imported by 'admin' on 'Apr 27th 20 03:07'. The dashboard includes a navigation bar with 'Dashboard', 'Projects', 'Images', and 'Help'. The 'Image Search' section is highlighted with a yellow box, and a yellow arrow points to the search bar.



6. In the pop up menu you can customize your image (e.g. Instance Size. Use a minimum of m1.xlarge instance for Foreman, with at least 60GB disk space); select “Advanced Options”.



7. Select “Create a New Script”.

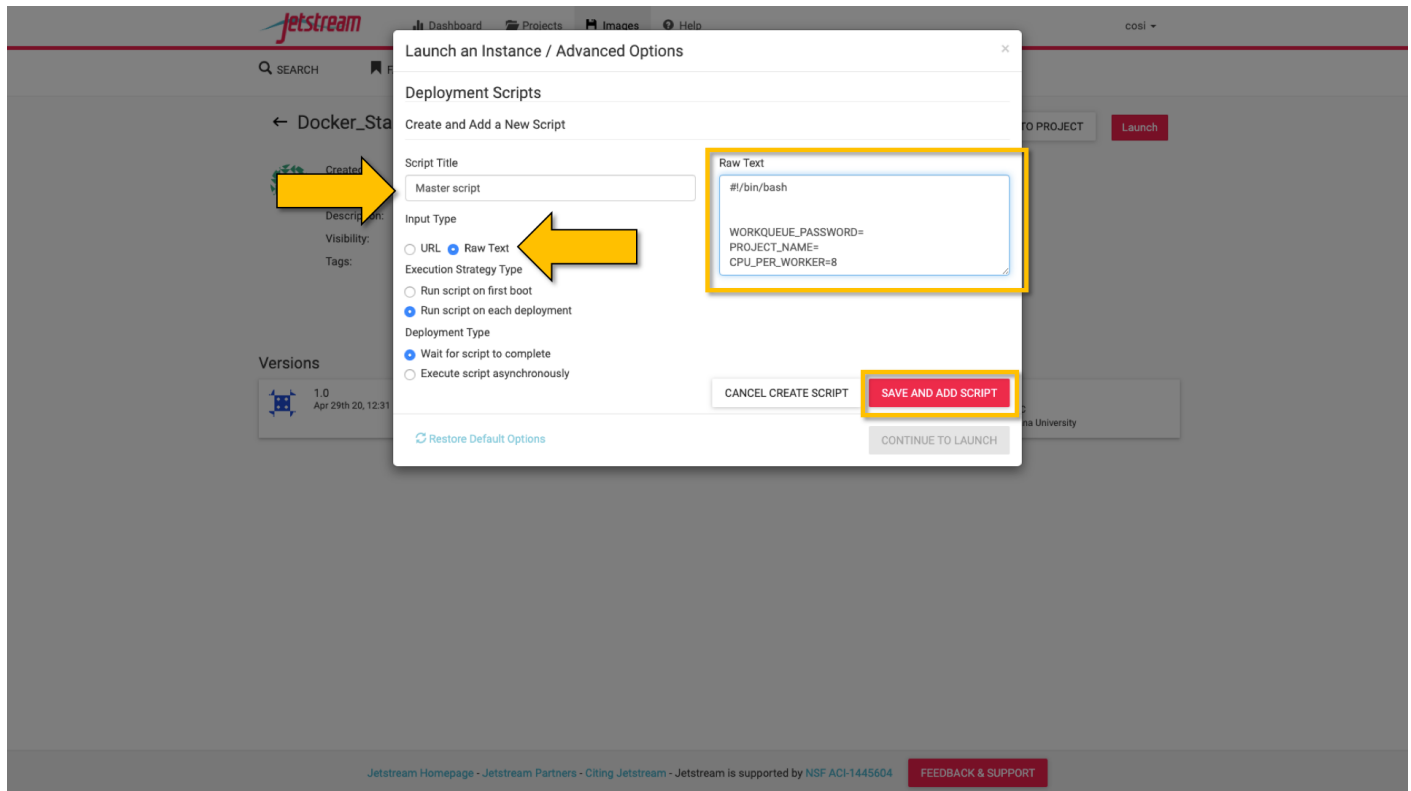


8. Title the script “Foreman” or similar, select “Raw Text” and copy and paste the Foreman script, linked below. The scripts generate a password and username based on the user account, but these can be personalized if needed (*not suggested for new users*). Select “Save and Add Script” and then “Continue to Launch”.

Deployment Scripts

- The deployment scripts for a *Foreman* instance (*atmo_deploy_master.sh*) can be found [here](#).
- The deployment scripts for a *Worker* instance (*atmo_deploy_worker.sh*) can be found [here](#).


Note: This step is required to be done **once** for the Foreman and **once for each Worker instance**. The deployment scripts are stored for future use.



9. Repeat steps 4-8 for one or more Worker instance(s), using the Worker deployment script. Use large or extra large images (at least 60GB of disk space is required).

Note: JetStream cloud will take at least 10-20 minutes and the wait-time will increase with the size of the BLAST database.

Start BLASTING! Now anyone can enter the <FOREMAN_IP_ADDRESS> into their browser and access Sequence-Server.



Dashboard

Projects

Images

Help

cosi

RESOURCES




DETAILS

OPTIONS

My BLAST Project

NEW

Instances

| | Name | Status | Activity | IP Address | Size | Provider |
|--------------------------|---|--------|----------|----------------|-----------|------------------|
| <input type="checkbox"/> |  Master | Active | N/A | 129.114.104.47 | M1.Xlarge | Jetstream - TACC |
| <input type="checkbox"/> |  Master-HP | Active | N/A | 129.114.16.70 | M1.Xlarge | Jetstream - TACC |
| <input type="checkbox"/> |  Worker | Active | N/A | 149.165.169.13 | M1.Xlarge | Jetstream - TACC |

Volumes

You have not added any volumes to this project.

Images

You have not added any images to this project.

Links

You have not added any links to this project.

Jetstream Homepage - Jetstream Partners - Citing Jetstream - Jetstream is supported by NSF ACI-1445604

FEEDBACK & SUPPORT

StarBLAST-HPC: HPC Deployment for Large Classes (>100)

The StarBLAST-HPC Setup is designed to distribute BLAST searches across multiple nodes on a High-Performance Computer and uses a Master-Worker set-up similar to StarBLAST-Docker (an atmosphere instance as the Master, and the HPC as the Worker). It is suggested that the Worker is set up ahead of time.

Some command line knowledge is required for setup.

4.1 HPC Requirements and Setup

It is important that the following software are installed on the HPC:

- [iRODS version 4.0 or newer](#)
- [ncbi-BLAST+ version 2.9.0 or newer](#)
- [CCTools version 7.0.21 or newer](#)
- [glibc version 2.14 or newer](#)
- [Support for CentOS7](#)
- [CyVerse user account](#)

iRODS, ncbi-BLAST+ and CCTools should be available in your home directory, which can be found using

```
cd  
pwd
```

It should output something similar to

```
/home/<U_NUMBER>/<USER>/
```

4.1.1 iRODS Installation Guide

- (1) From your home directory, obtain and install iRODS with the command

```
wget https://files.renci.org/pub/irods/releases/4.1.10/ubuntu14/irods-icommands-4.1.10-ubuntu14-x86_64.deb
apt-get install ./irods-icommands-4.1.10-ubuntu14-x86_64.deb
```

(2) Upon installation, set up the iCommands (requires a CyVerse account):

```
iinit
```

(3) You will be prompted to connect to the CyVerse with:

```
host name (DNS): data.cyverse.org
port #: 1247
username: <CyVerse_ID>
zone: iplant
password: <CyVerse_password>
```

iRODS should be installed and configured. If problems persists, a more in depth tutorial on iRODS and iCommands installation can be found [here](#).

4.1.2 ncbi-BLAST+ Installation Guide

(1) From your home directory, obtain and decompress ncbi-BLAST+ with

```
wget https://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/2.9.0/ncbi-blast-2.9.0+-x64-linux.tar.gz
tar -xvf ncbi-blast-2.9.0+-x64-linux.tar.gz
```

(2) Add ncbi-BLAST+ to the path (change the path to reflect the correct location of the ncbi-BLAST+ bin files):

```
export PATH=$HOME</PATH/TO/BLAST/BIN/>:$PATH
```

At this point, ncbi-BLAST+ should be installed and accessible.

(3) BLAST databases need to be downloaded in a <DATABASE>/ directory in the home folder.

```
/home/<U_NUMBER>/<USER>/<DATABASE>/
```

Note: An example of BLAST databases can be downloaded with iRODS here: `/iplant/home/cosimichele/200503_Genomes_n_p`. Read more on installing iRODS and iCommands above.

4.1.3 CCTools Installation Guide

(1) From your home directory, obtain and decompress CCTools with

```
wget https://ccl.cse.nd.edu/software/files/cctools-7.1.6-source.tar.gz
tar -xvf cctools-7.1.6-source.tar.gz
```

(2) Add CCTools to the path (change the path to reflect the correct location of the CCTools bin files):

```
export PATH=$HOME</PATH/TO/CCTOOLS/BIN/>:$PATH
```

At this point, CCTools should be installed and accessible.

Note: CCTools only works if your HPC has glibc version 2.14 or newer. In the following examples, glibc and BLAST+ are loaded through `module load`. `module load` is not necessary if the HPC system already supports glibc 2.14 and if ncbi-BLAST+ has been added to the path as described above.

4.2 Launching Workers on the HPC

The HPC uses a .pbs and qsub system to submit jobs.

- (1) Create a .pbs file that contains the following code and change the <VARIABLES> to preferred options:

```
#!/bin/bash
#PBS -W group_list=<GROUP_LIST>
#PBS -q windfall
#PBS -l select=<N_OF_NODES>:ncpus=<N_OF_CPUS>:mem=<N_MEMORY>gb
#PBS -l place=pack:shared
#PBS -l walltime=<MAX_TIME>
#PBS -l cput=<MAX_TIME>
module load blast
module load unsupported
module load ferng/glibc
module load singularity
export CCTOOLS_HOME=/home/<U_NUMBER>/<USER>/<CCTOOLS_DIRECTORY>
export PATH=${CCTOOLS_HOME}/bin:$PATH

cd /home/<U_NUMBER>/<USER>/<WORKERS_DIRECTORY>

MASTER_IP=<MASTER_IP>
MASTER_PORT=<PORT_NUMBER>
TIME_OUT_TIME=<TIME_OUT_TIME>
PROJECT_NAME=<PROJECT_NAME>

/home/<U_NUMBER>/<USER>/<CCTOOLS_DIRECTORY>/bin/work_queue_factory -T local -M
↪ $PROJECT_NAME --cores <N_CORES> -w <MIN_N_WORKERS> -W <MAX_N_WORKERS> -t $TIME_OUT_
↪ TIME
```

An example of a .pbs file running on the University of Arizona HPC:

```
#!/bin/bash
#PBS -W group_list=lyons-lab
#PBS -q windfall
#PBS -l select=2:ncpus=12:mem=24gb
#PBS -l place=pack:shared
#PBS -l walltime=02:00:00
#PBS -l cput=02:00:00
module load blast
module load unsupported
module load ferng/glibc
module load singularity
export CCTOOLS_HOME=/home/u12/cosi/cc tools-7.0.19-x86_64-centos7
export PATH=${CCTOOLS_HOME}/bin:$PATH

cd /home/u12/cosi/cosi-workers
```

(continues on next page)

(continued from previous page)

```

MASTER_IP=128.196.142.13
MASTER_PORT=9123
TIME_OUT_TIME=1800
PROJECT_NAME="starBLAST"

/home/u12/cosi/cctools-7.0.19-x86_64-centos7/bin/work_queue_factory -T local -M
↪ $PROJECT_NAME --cores 12 -w 1 -W 8 -t $TIME_OUT_TIME

```

In the example above, the user already has blast installed (calls it using `module load blast`). The script will submit to the HPC nodes a minimum of 1 and a maximum of 8 workers per node.

(2) Submit the `.pbs` script with

```
qsub <NAME_OF_PBS>.pbs
```

4.3 Setting Up the Master VM on the Cloud Service

Set up the Master instance for starBLAST-HPC by following the same steps as for StarBLAST-Docker, but **without adding the Master deployment script**. Additionally, BLAST databases need to be loaded manually onto the `<DATABASE>/` folder.

Once the VM is running, access it through ssh or by using the Web Shell (“Open Web Shell” button on your VM’s page). Once inside follow the next steps.

Note: IMPORTANT: THE PATH TO THE DATABASE ON THE MASTER NEED TO BE THE SAME AS THE ONE ON THE WORKER

(1) Ensure the databases on both the Master VM and Worker HPC are in the same directory. On the Worker HPC go to the `<DATABASE>/` directory and do

```
pwd
```

Then, on your Master VM, create the directory with the same path output above

```
mkdir -p SAME/PATH/TO/HPC/DATABASE/DIRECTORY/
```

(2) Now the `<DATABASE>/` directories have been set up to contain the desired databases. You can use the same databases preset for StarBLAST-Docker or make your own from a `.fasta` (or `.fa`, `.faa`, `.fna`) file using BLAST+’s `makeblastdb` referenced in StarBLAST-VICE. Both require iRODS (JetStream comes with iRODS pre-installed) and a CyVerse account.

Access iRODS using:

```
iinit
```

You will be prompted to connect to the CyVerse with:

```

host name (DNS): data.cyverse.org
port #: 1247
username: <CyVerse_ID>
zone: iplant
password: <CyVerse_password>

```

(3) Once connected, retrieve and move the databases to your <DATABASE>/ folder (shown for preset):

```
iget -rKVP /iplant/home/cosimichele/200503_Genomes_n_p
mv GCF_* /DATABASE/DIRECTORY/
```

(4) Move the databases to the HPC using either `sftp` or the steps as above if your HPC system has iRODS.

(5) Use this code within the Master instance to launch sequenceServer:

```
docker run --rm --name sequencesserver-scale -p 80:3000 -p 9123:9123 -e PROJECT_NAME=
↪<PROJECT_NAME> -e WORKQUEUE_PASSWORD=<PASSWORD> -e BLAST_NUM_THREADS=<N THREADS> -e
↪SEQSERVER_DB_PATH="/home/<U_NUMBER>/<USER>/<DATABASE_DIRECTORY>" -v /DATABASE/ON/
↪MASTER:/DATABASE/ON/WORKER zhxu73/sequencesserver-scale:no-irods
```

An example is:

```
docker run --rm --name sequencesserver-scale -p 80:3000 -p 9123:9123 -e PROJECT_
↪NAME=starBLAST -e WORKQUEUE_PASSWORD= -e BLAST_NUM_THREADS=2 -e SEQSERVER_DB_PATH="/
↪home/u12/cosi/DATABASE" -v /home/u12/cosi/DATABASE:/home/u12/cosi/DATABASE zhxu73/
↪sequencesserver-scale:no-irods
```

Note: The custom Database folder on the Master needs to have read and write permissions

Start BLASTING! Now anyone can enter the <MASTER_IP_ADDRESS> in their browser to access SequenceServer.

Using SequenceServer

SequenceServer allows to access BLAST+ commands through a simple GUI. Here, we show examples of how to BLAST using SequenceServer. For additional documentation please visit SequenceServer's [official website](#) and [original publication](#).

Note: These examples will take into consideration that you already have launched StarBLAST. Visit the other User Guides to know more on launching StarBLAST.

5.1 SequenceServer's Main Page

On the main page, the user will see:

- (1) The main input box where nucleotide (DNA) or amino acid (protein) sequences can be input using the [FASTA convention](#).
- (2) The nucleotide databases (left) and protein databases (right). The user will be able to choose which databases to BLAST against by clicking the boxes left to the databases' names.
- (3) The advanced parameters box. A list and description of all the advanced options can be accessed by pressing the “?” button.

Note: Advanced Parameters can heavily influence the resulting BLAST results, we suggest to read the descriptions beforehand.

SequenceServer 2.0.0.beta3 [Help & Support](#)

1

Paste query sequence(s) or drag file containing query sequence(s) in FASTA format here ...

2

Nucleotide databases [\[Select all\]](#)

☐ D. melanogaster (Fruit fly)
☐ Human
☐ Mouse
☐ A. thaliana
☐ Zebrafish
☐ C. elegans
☐ E. coli
☐ S. Cervisiae (Baker's yeast)
☐ O. sativa ssp japonica (rice)
☐ Chimpanzee

Protein databases [\[Select all\]](#)

☐ D. melanogaster (Fruit fly)
☐ Human
☐ Mouse
☐ A. thaliana
☐ Zebrafish
☐ C. elegans
☐ O. sativa ssp japonica (rice)
☐ E. coli
☐ S. Cervisiae (Baker's yeast)
☐ Chimpanzee

3

Advanced parameters: ?

BLAST

The input box will recognize the added nucleotide or amino acid sequence. The user can then select the database of choice (this step can be performed before adding the query sequence). In the example below, an isoform of the [Wacky protein FASTA sequence](#) was added to the input box and the *Drosophila melanogaster* (*D. melanogaster*) DNA database was selected.


```
SSRENDKSGSDDRDRDRDRDLRDLRDKRDRGSDRDRDMYKKDKYADKRESDRGERTAR
YGDWSEHVSSGKMYYYNCKTEISQWEKPEWDRERNLPRDQHKREKDYRDKDRDRDRDD
RFSRSTYKHSNSSRDNRLRMNYDNDGGPPSHRRRLDGRHNDADHDISGDSPTSEASY
SLSGTPTTHGGGPGGGGGGGGSDQPMGNALPRLSSHPTANSSASVATGTGATGGLH
YGS GTGGGPVTGATMLPTMSGMLNSNSNSAGGSSSNASSSLRNSVVGHIGTSGTTVP
TLGSQDPHQHHLNSNAPLPPGAKGKQALLMRQKMHLGLGVLDVQSHHGVSNGSVSDGT
NHAYNSVHNSVSGSLRDNVNSPLVMHSMSPSLNFTKSPIPTIVGHTNMMIAIYTCNPP
FGLKATLDGGVMVANASPATPGGNASSGSSGANSSQSIVPGHGPVCGISVITSMGNSGT
LCEGPPTPTQELDLSGSALEQQQLAAAAAATASSLQLQAAQQAQQQRKLDGTSSATLSS
LQSCVSSSGQAANLRGPEISPKLAKYFRADLIAHVTNWHAEVLRQVSCAQKCCEDTHL
FGDITCTRICAELKCARSLVRSTEINATLQEKKI
```

Detected: amino-acid sequence(s).

Nucleotide databases [\[Select all\]](#)

- ☒ *D. melanogaster* (Fruit fly)
- ☐ Human
- ☐ Mouse
- ☐ *A. thaliana*
- ☐ Zebrafish
- ☐ *C. elegans*
- ☐ *E. coli*
- ☐ *S. Cervisiae* (Baker's yeast)
- ☐ *O. sativa ssp japonica* (rice)
- ☐ Chimpanzee

Protein databases [\[Select all\]](#)

- ☐ *D. melanogaster* (Fruit fly)
- ☐ Human
- ☐ Mouse
- ☐ *A. thaliana*
- ☐ Zebrafish
- ☐ *C. elegans*
- ☐ *O. sativa ssp japonica* (rice)
- ☐ *E. coli*
- ☐ *S. Cervisiae* (Baker's yeast)
- ☐ Chimpanzee

Advanced parameters:

TBLASTN

5.2 BLAST Loading & Results Page

After clicking BLAST (in this case TBLASTN), the page will switch to a loading screen. The length of this screen is dependable on:

- (1) Computational power of the foreman (BLASTing is done but the machine has difficulties displaying the results due to the number of outputs).
- (2) Computational power and availability of workers.
- (3) Length of query.

SequenceServer 2.0.0.beta3

[Help & Support](#)

BLAST-ing

This can take some time depending on the size of your query and database(s). The page will update automatically when BLAST is done.

You can bookmark the page and come back to it later or share the link with someone.

Run BLAST locally on Mac OSX with SequenceServer. [Tweet](#)

Please cite relevant data sources and: Priyam et al. (2019) Sequenceserver: a modern graphical user interface for custom BLAST databases.

Here below is the result output of the Wacky BLAST search. This page will display BLAST-related statistical results such as Query coverage (%), Total score, E-value, and Identity for the whole query (top) and specific sequences (below). For more information on the BLAST output, visit the [NCBI BLAST FAQ page](#) or this [Medium article](#).

SequenceServer 2.0.0.beta3

[Help & Support](#)

TBLASTN: 1 query, 1 database

Query= Query_1

Download FASTA, XML, TSV

[FASTA of all hits](#)[FASTA of selected hit\(s\)](#)[Alignment of all hits](#)[Alignment of selected hit\(s\)](#)[Standard tabular report](#)[Full tabular report](#)[Full XML report](#)

TBLASTN 2.9.0+; query submitted on 2021-02-23 00:36:04 UTC

Databases (1870 sequences, 143726002 characters): D. melanogaster (Fruit fly)

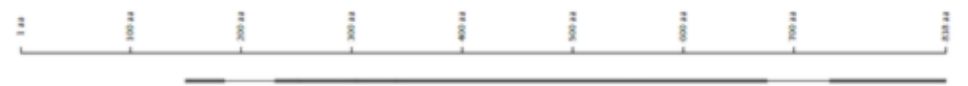
Parameters: eval=1e-05, matrix BLOSUM62, gap-open 11, gap-extend 1, filter L1

Circos

Query= Query_1

1/1

Graphical overview of hits

[SVG](#) | [PNG](#)

Length distribution of hits

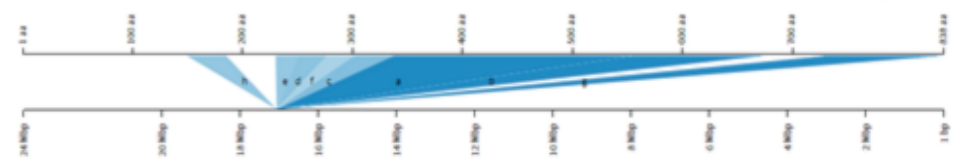
| # | Similar sequences | Query coverage (%) | Total score | E value | Identity (%) |
|----|----------------------------------|--------------------|-------------|-------------------------|--------------|
| 1. | ref NC_004354.4 | 71 | 2187 | 2.17×10^{-145} | 87 |

▼ [ref|NC_004354.4|](#) Drosophila melanogaster chromosome X

1/1

☐ Select | [Sequence](#) | [FASTA](#) | [Alignment](#)

Graphical overview of aligning region(s)

[SVG](#) | [PNG](#)a. Score: 207.99 (528), E value: 2.17×10^{-145} , Identities: 219/242 (90.50%), Positives: 219/242 (90.50%), Gaps: 23/242 (9.50%), Hit Frame: -1

```

Query      338 GRHNDNADMDISGDSTPTSEASYSLXXXXXXXXXXXXXXXXXXXXXXXXXNSDQPMGNALPRLSSHPXXXXXXXXXXXXXXXXXXXX 422
          339 GRHNDNADMDISGDSTPTSEASYSLSGTPTTHGGGPGGGGPGGGGNSDQPMGNALPRLSSHTANSSASVATGTGATGGLHYG
Subject 17072234 GRHNDNADMDISGDSTPTSEASYSLSGTPTTHGGGPGGGGPGGGGNSDQPMGNALPRLSSHTANSSASVATGTGATGGLHYG 17071982
Query      423 XXXXXXXXXXXXXMLPTXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXVVGHIQSTSG-----TTVPTLGS 484
          424 SGTGGGPTVGATMLPTMSGMLNSNSNSAGSSNSAS555LRNSVVGHIQSTSGTTVPTLGS
Subject 17071979 SGTGGGPTVGATMLPTMSGMLNSNSNSAGSSNSAS555LRNSVVGHIQSTSGVSLACLIALSFQILIVMHALQTTPVPTLGS 17071727
Query      485 QDPHQHLSNAPLPPGAKGKQALLMRQKMLGLGVLDQXXXXXXXXXXXXXXXXXGTHAYNSVNSVS55LR 556
          486 QDPHQHLSNAPLPPGAKGKQALLMRQKMLGLGVLDQSHHGVSNSVS55GTHAYNSVNSVS55LR
Subject 17071724 QDPHQHLSNAPLPPGAKGKQALLMRQKMLGLGVLDQSHHGVSNSVS55GTHAYNSVNSVS55LR 17071511

```

b. Score: 205.30 (521), E value: 2.17×10^{-145} , Identities: 121/121 (100.00%), Positives: 121/121 (100.00%), Gaps: 0/121 (0.00%), Hit Frame: -3

```

Query      556 RDNVNSPLVMHSHSPSLNFTKSP2PTIVGHTNMSIAYTCNPPFLKATLGGGVANASXXXXXXXXXXXXXXXXXXXXXQ5IVP 640
          557 RDNVNSPLVMHSHSPSLNFTKSP2PTIVGHTNMSIAYTCNPPFLKATLGGGVANASPATPGGNAS555GANSSQ5IVP
Subject 17071449 RDNVNSPLVMHSHSPSLNFTKSP2PTIVGHTNMSIAYTCNPPFLKATLGGGVANASPATPGGNAS555GANSSQ5IVP 17071197
Query      641 GMGPVCGISVITSMGNSGTLCEGPTPTQELDLSG 676
          642 GMGPVCGISVITSMGNSGTLCEGPTPTQELDLSG
Subject 17071194 GMGPVCGISVITSMGNSGTLCEGPTPTQELDLSG 17071089

```

c. Score: 74.33 (181), E value: 2.17×10^{-145} , Identities: 34/39 (87.18%), Positives: 35/39 (89.74%), Gaps: 0/39 (0.00%), Hit Frame: -2

```

Query      302 FSRSTYKHSNSRDRSLRWYDNDGGPPSHRRRLDGRH 340
          303 F * YKHSNSRDRSLRWYDNDGGPPSHRRRLDGRH
Subject 17072401 FFFAAYKHSNSRDRSLRWYDNDGGPPSHRRRLDGRH 17072287

```

d. Score: 62.77 (151), E value: 2.17×10^{-145} , Identities: 27/27 (100.00%), Positives: 27/27 (100.00%), Gaps: 0/27 (0.00%), Hit Frame: -2

```

Query      251 SGKRYYYNCKTEISQEKPKENVDRER 277
          252 SGKRYYYNCKTEISQEKPKENVDRER
Subject 17072728 SGKRYYYNCKTEISQEKPKENVDRER 17072650

```

e. Score: 51.22 (121), E value: 2.17×10^{-145} , Identities: 23/24 (95.83%), Positives: 24/24 (100.00%), Gaps: 0/24 (0.00%), Hit Frame: -3

```

Query      230 ERSORGERTARYGDHSEHVSSGK 253
          231 ERSORGERTARYGDHSEHVSSGK+
Subject 17072856 ERSORGERTARYGDHSEHVSSGK 17072787

```

f. Score: 22.71 (47), E value: 2.17×10^{-145} , Identities: 29/30 (96.67%), Positives: 30/30 (100.00%), Gaps: 0/30 (0.00%), Hit Frame: -3

```

Query      277 RNLPRDQHXXXXXXXXXXXXXXXXXFSRST 306
          278 RNLPRDQHREKDYRDKDRDRDRDRFSRS+
Subject 17072541 RNLPRDQHREKDYRDKDRDRDRDRFSRS 17072454

```

g. Score: 201.44 (511), E value: 2.22×10^{-12} , Identities: 107/165 (64.85%), Positives: 107/165 (64.85%), Gaps: 58/165 (35.15%), Hit Frame: -2

```

Query      732 ANLRGPEISPKLAKYFRADLIAHVTHNHAELVLRQVSCE-----AQKCCEDTHLF 781
          733 ANLRGPEISPKLAKYFRADLIAHVTHNHAELVLRQVSCE-----AQKCCEDTHLF
Subject 17070850 ANLRGPEISPKLAKYFRADLIAHVTHNHAELVLRQVSCEVGFACE*GTNPTSHTVTVIHZGLELIVNVHCFSAKCCEDTHLF 17070598
Query      782 GDITCTRICAEKCARSLVRSSTEINATLQEQ-----IYLRHQIRRIEESKTQNAFMSDOT 838
          783 GDITCTRICAEKCARSLVRSSTEINATLQEQ-----IYLRHQIRRIEESKTQNAFMSDOT
Subject 17070595 GDITCTRICAEKCARSLVRSSTEINATLQEQ*VLQ*HLNGSCSNQICLIAFRRIYLRHQIRRIEESKTQNAFMSDOT 17070358

```

h. Score: 53.53 (127), E value: 2.76×10^{-6} , Identities: 37/37 (100.00%), Positives: 37/37 (100.00%), Gaps: 0/37 (0.00%), Hit Frame: -3

