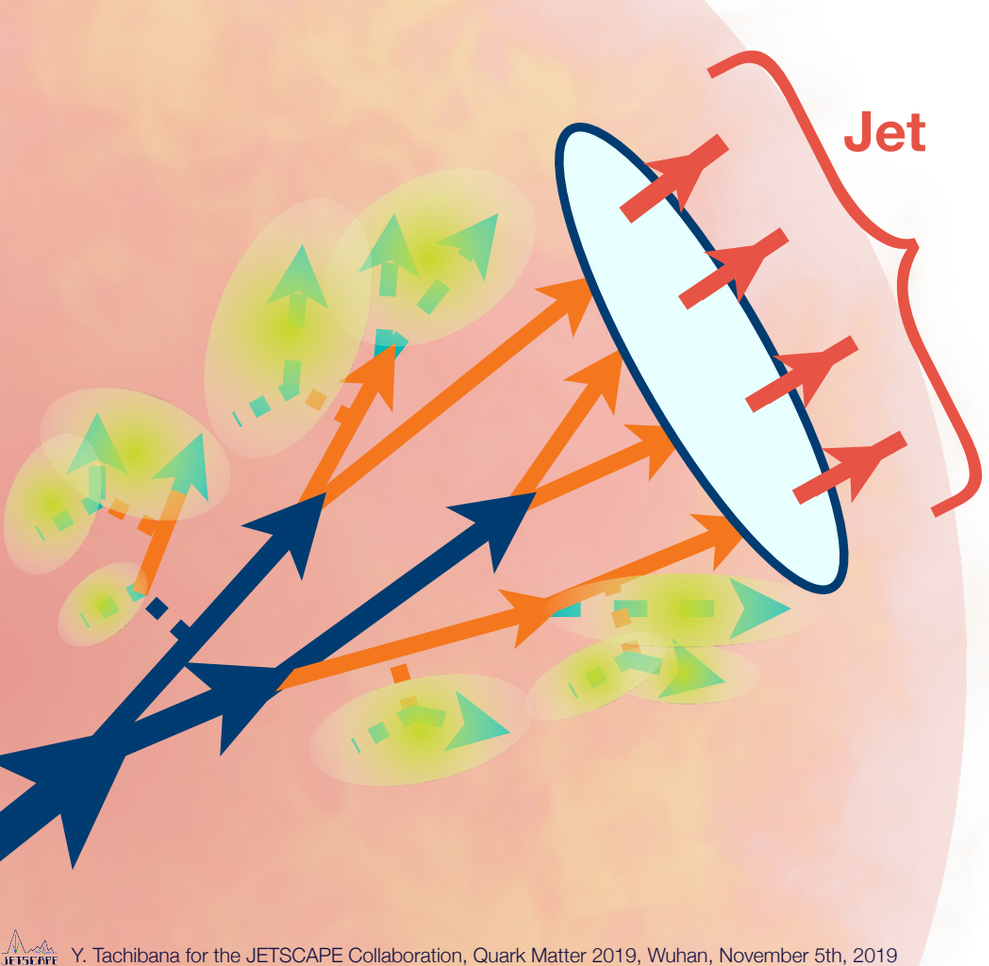
The JETSCAPE and SCAPE Framework



Hands-on session

Chathuranga Sirimanna
Wayne State University

JETSCAPE Online School
July 18 2023

Thanks to those who have helped others with the preparatory instructions!

Ask questions in slack channel:

#july17-18-framework

If you have the same question as one that is already posted, add 👍

TAs: Ismail Soudi
Ritoban Datta

Installation and Prerequisites

https://github.com/JETSCAPE/SummerSchool2023

Follow instructions at https://indico.cern.ch/event/1282714/page/30284-information-for-hands-on-session

Quick Poll: Completed the preparation ✅
Partially finished or Incomplete ❌

Downloading the test hydro file: Inside the jetscape-docker directory

```
cd X-SCAPE/examples
./get_hydroSample_PbPb2760_cen_00-05.sh
```

Review: Docker

https://github.com/JETSCAPE/SummerSchool2023

For this school, we require you to run X-SCAPE via docker

This allows everyone in the school to have a uniform software environment

```
(base) Chathurangas-MacBook-Pro:~ chathuranga$ docker container ls -a
CONTAINER ID   IMAGE                    COMMAND                  CREATED      STATUS                          PORTS     NAMES
c35c896502c2   jetscape/school:2023.1   "/bin/sh -c /bin/bash"   2 days ago   Exited (0) 10 seconds ago                 myJetscape
(base) Chathurangas-MacBook-Pro:~ chathuranga$ docker start -ai myJetscape
jetscape-user@c35c896502c2:~$ pwd
/home/jetscape-user
jetscape-user@c35c896502c2:~$ ls
SummerSchool2023  X-SCAPE
jetscape-user@c35c896502c2:~$ cd SummerSchool2023/
jetscape-user@c35c896502c2:~/SummerSchool2023$ git pull
Already up to date.
jetscape-user@c35c896502c2:~/SummerSchool2023$
```

The XML Configuration

All of the JETSCAPE settings are specified by two XML files:

- Master XML file: *you don't modify this*
  - Contains default values for every possible module and parameter
- User XML file: *you provide this*
  - Contains a list of which modules to run, and which default parameter values to override

```xml
<?xml version="1.0"?>

<!-- Copyright (c) The JETSCAPE Collaboration, 2018 -->
<!-- Modular, task-based framework for simulating all aspects of heavy-ion collisions -->
<!-- For the list of contributors see AUTHORS. -->
<!-- Report issues at https://github.com/JETSCAPE/JETSCAPE/issues -->
<!-- or via email to bugs.jetscape@gmail.com -->
<!-- Distributed under the GNU General Public License 3.0 (GPLv3 or later). -->
<!-- See COPYING for details. -->

 <jetscape>

  <!-- General settings -->
  <nEvents> 100 </nEvents>
  <setReuseHydro> true </setReuseHydro>
  <nReuseHydro> 10 </nReuseHydro>

  <!-- Technical settings -->
  <debug> on </debug>
  <remark> off </remark>
  <vlevel> 0 </vlevel>
  <nEvents_printout> 100 </nEvents_printout>
  <enableAutomaticTaskListDetermination> true </enableAutomaticTaskListDetermination>

  <!--  JetScape Writer Settings -->
  <outputFilename>test_out</outputFilename>
  <JetScapeWriterAscii> off </JetScapeWriterAscii>
  <JetScapeWriterAsciiGZ> off </JetScapeWriterAsciiGZ>
  <JetScapeWriterHepMC> off </JetScapeWriterHepMC>
  <JetScapeWriterRootHepMC> off </JetScapeWriterRootHepMC>
  <JetScapeWriterFinalStatePartonsAscii> off </JetScapeWriterFinalStatePartonsAscii>
  <JetScapeWriterFinalStateHadronsAscii> off </JetScapeWriterFinalStateHadronsAscii>
  <write_pthat> 0 </write_pthat>

  <!--  Random Settings. For now, just a global  seed. -->
  <!--  Note: It's each modules responsibility to adopt it -->
  <!--  Note: Most if not all modules should understand 0 to mean a random value -->
  <!--  Note: Both 0 and non-zero values require careful treatment in case of multi-threading -->
  <!--           An example implementation is in JetEnergyLossManager.cc -->
  <Random>
    <seed>0</seed>
  </Random>
```

Open the file and take a look: X-SCAPE/config/jetscape_main.xml

This is where you find all *possible* settings of all *possible* modules

As user, don’t modify this!

User XML Configuration

Open the file:
SummerSchool2023/July18_Framework/xml/jetscape_user_pGun_vac.xml

This XML file is for vacuum simulation

Additional modules required for heavy-ion simulations

Specifies which modules you want to run

```xml
<?xml version="1.0"?>

<jetscape>

  <nEvents> 200 </nEvents>

  <outputFilename>test_out_pGun_vac</outputFilename>
  <JetScapeWriterAscii> on </JetScapeWriterAscii>
  <JetScapeWriterFinalStateHadronsAscii> on
      </JetScapeWriterFinalStateHadronsAscii>
  <JetScapeWriterFinalStatePartonsAscii> on
      </JetScapeWriterFinalStatePartonsAscii>

  <!-- Hard Process -->
  <Hard>
      <PGun>
        <name>PGun</name>
        <pT>100</pT>
        <parID>21</parID>
        <useHybridHad>0</useHybridHad>
      </PGun>
  </Hard>

  <!--Eloss Modules -->
  <Eloss>
    <Matter>
      <Q0> 1.0 </Q0>
      <in_vac> 1 </in_vac>
      <vir_factor> 0.25 </vir_factor>
      <recoil_on> 0 </recoil_on>
      <broadening_on> 0 </broadening_on>
      <brick_med> 0 </brick_med>
    </Matter>
  </Eloss>

  <!-- Jet Hadronization Module -->
  <JetHadronization>
    <name>colorless</name>
  </JetHadronization>

</jetscape>
```

User XML Configuration

Open the file:
SummerSchool2023/July18_Framework/xml/jetscape_user_pGun_vac.xml

Set number of events

Set output format

Set parton p_{T}

Set parton type

Parameters for MATTER energy loss module: PP19 tune

```xml
<?xml version="1.0"?>

<jetscape>

  <nEvents> 200 </nEvents>

  <outputFilename>test_out_pGun_vac</outputFilename>
  <JetScapeWriterAscii> on </JetScapeWriterAscii>
  <JetScapeWriterFinalStateHadronsAscii> on
      </JetScapeWriterFinalStateHadronsAscii>
  <JetScapeWriterFinalStatePartonsAscii> on
      </JetScapeWriterFinalStatePartonsAscii>

  <!-- Hard Process -->
  <Hard>
      <PGun>
        <name>PGun</name>
        <pT>100</pT>
        <parID>21</parID>
        <useHybridHad>0</useHybridHad>
      </PGun>
  </Hard>

  <!--Eloss Modules -->
  <Eloss>
    <Matter>
      <Q0> 1.0 </Q0>
      <in_vac> 1 </in_vac>
      <vir_factor> 0.25 </vir_factor>
      <recoil_on> 0 </recoil_on>
      <broadening_on> 0 </broadening_on>
      <brick_med> 0 </brick_med>
    </Matter>
  </Eloss>

  <!-- Jet Hadronization Module -->
  <JetHadronization>
    <name>colorless</name>
  </JetHadronization>

</jetscape>
```

Generate some events!

Let's generate some mono-energetic jet events in vacuum

```xml
<?xml version="1.0"?>

<jetscape>

  <nEvents> 200 </nEvents>

  <outputFilename>test_out_pGun_vac</outputFilename>
  <JetScapeWriterAscii> on </JetScapeWriterAscii>
  <JetScapeWriterFinalStateHadronsAscii> on </JetScapeWriterFinalStateHadronsAscii>
  <JetScapeWriterFinalStatePartonsAscii> on </JetScapeWriterFinalStatePartonsAscii>

  <!-- Hard Process -->
  <Hard>
      <PGun>
        <name>PGun</name>
        <pT>100</pT>
        <parID>21</parID>
        <useHybridHad>0</useHybridHad>
      </PGun>
  </Hard>

  <!--Eloss Modules -->
  <Eloss>
    <Matter>
      <Q0> 1.0 </Q0>
      <in_vac> 1 </in_vac>
      <vir_factor> 0.25 </vir_factor>
      <recoil_on> 0 </recoil_on>
      <broadening_on> 0 </broadening_on>
      <brick_med> 0 </brick_med>
    </Matter>
  </Eloss>

  <!-- Jet Hadronization Module -->
  <JetHadronization>
    <name>colorless</name>
  </JetHadronization>

</jetscape>
```

Set to 200 events

Set output types:
ascii
`<JetScapeWriterAscii>`
+
final-state particles
`<JetScapeWriterFinalStateHadronsAscii>`
`<JetScapeWriterFinalStatePartonsAscii>`

cd ~/X-SCAPE/build

./runJetscape ~/SummerSchool2023/July18_Framework/xml/jetscape_user_pGun_vac.xml

Final-state hadron output

You should then find a file test_out_pGun_vac_final_state_hadrons.dat in the same directory

```
#        JETSCAPE_FINAL_STATE    v2      |      N       pid     status  E       Px      Py      Pz
#        Event   1       weight  1       EPangle 0       N_hadrons       33
0 211 0 11.2824 10.8641 -2.7566 1.2829
1 -211 0 1.73982 1.65102 -0.287417 -0.446145
2 211 0 1.10025 1.02918 -0.204594 0.299986
3 -211 0 3.13511 2.73527 1.52165 -0.110738
4 3122 0 3.10367 2.31419 1.29264 -1.16688
5 -3112 0 3.98005 2.92083 2.35389 -0.578656
6 -211 0 0.655955 -0.0290351 0.531123 -0.357579
7 -211 0 1.64647 0.203998 1.39834 0.833331
8 211 0 25.8314 25.1942 -5.33007 2.021
9 -211 0 6.9339 6.68466 -1.58541 0.928052
10 211 0 2.53373 2.52451 -0.0930044 -0.136049
11 -211 0 1.79333 1.67936 -0.311685 0.528365
12 211 0 0.723608 0.595915 0.0728507 0.379086
13 211 0 2.00782 1.98516 -0.136759 0.228717
14 -211 0 3.27388 3.16471 -0.811925 -0.155601
15 211 0 2.71014 2.28976 1.10235 -0.931253
16 -211 0 0.873427 0.760702 0.3844 -0.130248
17 3122 0 5.27026 4.27592 2.13484 -1.92089
18 22 0 0.352175 0.251495 0.214094 -0.122234
19 -3122 0 4.70361 3.37988 2.74914 -1.37764
20 22 0 0.0742152 0.0419815 0.0130966 -0.0597823
```

Column format

Event info

Hadrons

...

And similarly for test_out_pGun_vac_final_state_partons.dat

Ascii output

You should also find a file test_out.dat in the same directory

```
0 Event
# JetScape writersigmaGen 1
# JetScape writersigmaErr 0
# JetScape writerweight 1
# HardProcess Parton List: PGun
0 21 0 100 0 0 100 0 0 0 0
```

Event info

...

```
# Energy loss Shower Initating Parton: JetEnergyLoss
0 21 0 100 0 0 100 0 0 0 0
```

Initial parton

...

```
[0]=>[1] P 0 21 0 89.8122 0 0 100 0 0 0 0
[1]=>[2] P 0 21 0 90.7008 -0.0083348 6.26044 97.6128 0.1 0 0 0.1
[1]=>[3] P 0 21 0 2.23671 0.331863 1.96795 2.3872 0.1 0 0 0.1
[2]=>[4] P 0 21 0 24.5204 -0.282151 0.519991 25.8224 0.192895 -0.00211305 -0.000774469 0.2
[2]=>[5] P 0 21 0 70.845 0.0881721 6.08072 71.7904 0.192895 -0.00211305 -0.000774469 0.2
[5]=>[6] P 0 21 0 69.758 0.100395 6.06944 70.233 0.289562 -0.0219571 0.00793791 0.3
[5]=>[7] P 0 21 0 1.34537 -0.538802 0.421961 1.55742 0.289562 -0.0219571 0.00793791 0.3
```

Shower-initiating parton

...

Parton shower history

...

```
# Hadronization module: Hadronization
# Final State Hadrons
[0] H 0 211 0 11.2083 0.114211 6.03469 11.2824 0 0 0 0
[1] H 0 -211 0 1.67585 -0.263172 6.11083 1.73982 0 0 0 0
[2] H 0 211 0 1.04932 0.282128 6.08695 1.10025 0 0 0 0
[3] H 0 -211 0 3.13004 -0.0353716 0.507672 3.13511 0 0 0 0
[4] H 0 3122 0 2.65073 -0.427106 0.509398 3.10367 0 0 0 0
[5] H 0 -3112 0 3.75127 -0.153651 0.678328 3.98005 0 0 0 0
[6] H 0 -211 0 0.531916 -0.629782 1.62541 0.655955 0 0 0 0
```

Hadrons

...

MATTER+LBT in Hydro medium

Open the file:
SummerSchool2023/July18_Framework/xml/jetscape_user_pGun_MATTER_LBT_Hydro.xml

```xml
<?xml version="1.0"?>

<jetscape>

  <nEvents> 100 </nEvents>
  <setReuseHydro> true </setReuseHydro>
  <nReuseHydro> 100 </nReuseHydro>

  <JetScapeWriterAscii> on </JetScapeWriterAscii>
  <outputFilename>test_out_pGun_MATTER_LBT_Hydro</outputFilename>
  <JetScapeWriterFinalStateHadronsAscii> on </JetScapeWriterFinalStateHadronsAscii>
  <JetScapeWriterFinalStatePartonsAscii> on </JetScapeWriterFinalStatePartonsAscii>
```

Set to 50 or 100 events

Set Reuse Hydro to read from file and reuse the hydro profiles for the given number of iterations.

...

```xml
  <!-- Inital State Module  -->
  <IS>
     <initial_profile_path>../examples/test_hydro_files</initial_profile_path>
  </IS>
```

...

```xml
  <!--Preequilibrium Dynamics Module -->
  <Preequilibrium>
    <NullPreDynamics> </NullPreDynamics>
  </Preequilibrium>

  <!-- Hydro  Module  -->
  <Hydro>
      <hydro_from_file>
        <name>Hydro from file </name>
        <read_in_multiple_hydro>1</read_in_multiple_hydro>
        <hydro_files_folder>../examples/test_hydro_files</hydro_files_folder>
      </hydro_from_file>
  </Hydro>
```

MATTER+LBT in Hydro medium

Open the file:
SummerSchool2023/July18_Framework/xml/jetscape_user_pGun_MATTER_LBT_Hydro.xml

```
<!--Eloss Modules -->
<Eloss>
  <deltaT>0.1</deltaT>
  <formTime> -0.1</formTime>
  <maxT>20</maxT>
  <tStart> 0.9 </tStart> <!-- Start time of jet quenching, proper time, fm/c   -->

  <Matter>
    <name>Matter</name>
    <useHybridHad>0</useHybridHad>
    <matter_on> 1 </matter_on>
    <Q0> 2.0 </Q0>
    <T0> 0.16 </T0>
    <in_vac> 0 </in_vac>
    <vir_factor> 0.25 </vir_factor>
    <recoil_on> 1 </recoil_on>
    <broadening_on> 0 </broadening_on>
    <brick_med> 0 </brick_med>
    <brick_length> 5.0 </brick_length>
    <hydro_Tc> 0.16 </hydro_Tc>
    <qhat0> -2.0 </qhat0>
    <alphas>0.30</alphas>
  </Matter>

  <Lbt>
    <name> Lbt </name>
    <Q0>2.0</Q0>
    <in_vac> 0 </in_vac>
    <only_leading> 0 </only_leading>
    <hydro_Tc> 0.16 </hydro_Tc>
    <alphas>0.30</alphas>
    <run_alphas>1</run_alphas>   <!-- 0 for fixed alpha_s and 1 for running alpha_s -->
  </Lbt>
</Eloss>
```

General parameters

Parameters specific to MATTER module

Parameters specific to LBT module

cd ~/X-SCAPE/build

./runJetscape ~/SummerSchool2023/July18_Framework/xml/jetscape_user_pGun_MATTER_LBT_Hydro.xml

MATTER+MARTINI in Hydro medium

Open the file:
SummerSchool2023/July18_Framework/xml/jetscape_user_pGun_MATTER_MARTINI_Hydro.xml

```
<!--Eloss Modules -->
<Eloss>
  <deltaT>0.01</deltaT>
  <formTime> -0.1</formTime>
  <maxT>20</maxT>
  <tStart> 0.9 </tStart> <!-- Start time of jet quenching, proper time, fm/c   -->

  <Matter>
    <name>Matter</name>
    <useHybridHad>0</useHybridHad>
    <matter_on> 1 </matter_on>
    <Q0> 2.0 </Q0>
    <T0> 0.16 </T0>
    <in_vac> 0 </in_vac>
    <vir_factor> 0.25 </vir_factor>
    <recoil_on> 1 </recoil_on>
    <broadening_on> 0 </broadening_on>
    <brick_med> 0 </brick_med>
    <brick_length> 5.0 </brick_length>
    <hydro_Tc> 0.16 </hydro_Tc>
    <qhat0> -2.0 </qhat0>
    <alphas>0.30</alphas>
  </Matter>

  <Martini>
    <name > Martini </name>
    <Q0> 2.0 </Q0>
    <alpha_s> 0.3 </alpha_s>
    <pcut> 1.0 </pcut>
    <hydro_Tc> 0.16 </hydro_Tc>
    <recoil_on> 1 </recoil_on>
    <run_alphas>1</run_alphas>
    <path>../src/jet/Martini/</path>
  </Martini>
</Eloss>
```

General parameters

Parameters specific to MATTER module

Parameters specific to MARTINI module

cd ~/X-SCAPE/build

./runJetscape ~/SummerSchool2023/July18_Framework/xml/jetscape_user_pGun_MATTER_MARTINI_Hydro.xml

Generating p_T spctrum

Open the file:
SummerSchool2023/July18_Framework/analyze_particle_spectra.py

```
32  # change the following line to your working folder
33  working_path = path.join(home, "JETSCAPE", "build")
34
35
36  class SimpleHistogram:
37      """A simple histogram class"""
38      def __init__(self, x_min, x_max, nx):
39          self.x_min_ = x_min
40          self.x_max_ = x_max
41          self.nx_ = nx
42          self.dx_ = (x_max - x_min)/nx
43          self.bin_x_ = zeros(nx)
44          for i in range(nx):
45              self.bin_x_[i] = x_min + (i+0.5)*self.dx_
46          self.bin_y_ = zeros(nx)
47
48      def fill(self, x_in, val):
49          idx = int((x_in - self.x_min_)/self.dx_)
50          if idx >= 0 and idx < self.nx_:
51              self.bin_y_[idx] += val
52
53
54  data_filename = "test_out_pGun_MATTER_LBT_Hydro_final_state_hadrons.dat"
55  nev = 0
56  event_list = []
```

Path of the directory containing the final state hadrons

File name of the final state hadrons

Inside the July18_Framework directory:
python analyze_particle_spectra.py

Writing a custom module

To develop a new JETSCAPE module, you should inherit from the relevant base class (InitialState, JetEnergyLoss, etc.) and implement the relevant initialization and execution functions, described in detail in The JETSCAPE framework Section 3.3.

First, copy an example custom module into the X-SCAPE src code

```
cp ~/SummerSchool2023/July18_Framework/MyJEL* ~/X-SCAPE/src/jet
```

Writing a custom module

```
1  #ifndef MYJEL_H
2  #define MYJEL_H
3
4  #include "JetEnergyLossModule.h"
5
6  using namespace Jetscape;
7
8  class MyJEL : public JetEnergyLossModule<MyJEL>
9  {
10  public:
11
12    MyJEL();
13    virtual ~MyJEL();
14
15    void Init();
16    void DoEnergyLoss(double deltaT,double time, double Q2,
17                      vector<Parton>& pIn, vector<Parton>& pOut);
18    void WriteTask(weak_ptr<JetScapeWriter> w);
19
20  private:
21    // Allows the registration of the module so that it is available
22    // to be used by the Jetscape framework.
23    static RegisterJetScapeModule<MyJEL> reg;
24
25 };
26
27 #endif // MyJEL
```

Take a look at MyJEL.h

You just need to implement your physics in these standard functions, which will be called by the framework

Note: Which function(s) you need to implement depends on what type of module you are implementing!
For details, see arXiv:1903.07706

Writing a custom module

Additionally, you must register your module with the framework with the following steps:

- Add the following to your module .h:

```
private:
// Allows the registration of the module so that it is available to be used by the Jetscape fr
static RegisterJetScapeModule<MyClass> reg;
```

- Add the following to your module .cc:

Take a look at MyJEL.cc

```
// Register the module with the base class
RegisterJetScapeModule<MyClass> MyClass::reg("CustomModuleBlahBlah");
```

where `MyClass` is the name of your class, and "CustomModuleBlahBlah" is the name that should be added to the XML configuration. You can see any of the established modules, e.g. `Matter`, as an example.

Important Note: In the case of custom modules, you *must* start your module name with "CustomModule..." in order for it to be recognized by the framework (for custom writers, you must start the name with "CustomWriter").

Build your custom module

```
cd X-SCAPE/build
cmake ..
make -j 4
```

Build your custom module

```
cd X-SCAPE/build
cmake ..
make -j 4
```

Edit jetscape_user_PP19.xml to add your module

It will then automatically be run by the framework

```
19    <!--Eloss Modules -->
20    <Eloss>
21      <Matter>
22        <Q0> 1.0 </Q0>
23        <in_vac> 1 </in_vac>
24        <vir_factor> 0.25 </vir_factor>
25        <recoil_on> 0 </recoil_on>
26        <broadening_on> 0 </broadening_on>
27        <brick_med> 0 </brick_med>
28      </Matter>
29      <CustomModuleMyJEL>
30        <name>blahblahblah</name>
31      </CustomModuleMyJEL>
32    </Eloss>
```

Run your custom module

[Inside the docker container]

./runJetscape ../config/jetscape_user_PP19.xml

```
[Info] 152MB  Intialize JetScape ...
[Info] 152MB  Created JetScapeXML Instance
[Info] 152MB  Open XML Master file : ../config/jetscape_master.xml
[Info] 152MB  Open XML User file : ../config/jetscape_user_PP19.xml
[Info] 152MB  ================================================================
[Info] 152MB  nEvents = 500
[Info] 152MB  Reuse Hydro:  true
[Info] 152MB  nReuseHydro: 10
[Info] 152MB  JetScapeTaskSupport found seed 0, using one engine for all and reseeding to 2038565212
[Info] 155MB  JetScape::DetermineTaskList() -- Hard Process: Added PythiaGun to task list.
[Info] 155MB  JetScape::DetermineTaskList() -- Eloss: Added Matter to Eloss list.
[Info] 155MB  JetScape::DetermineTaskList() -- Eloss: Added CustomModuleMyJEL to Eloss list.
[Info] 155MB  JetScape::DetermineTaskList() -- JetHadronization: Added ColorlessHadronization to task list.
[Info] 155MB  JetScape::DetermineTaskList() -- JetScapeWriterHepMC (test_out.hepmc) added to task list.
```

Success!

Important to next sessions

In preparation for the physics sessions, please complete the following slide before upcoming sessions

This is important for the upcoming physics sessions

Build X-SCAPE

with external packages enabled

To run certain external software (MUSIC, CLVisc, SMASH), you will need to explicitly download them, and you may need to re-run `cmake` with specific command-line options. Scripts to download and install the external packages are provided in `external_packages/`. Please see external packages for full details.

```
cd X-SCAPE/external_packages
./get_lbtTab.sh
./get_3dglauber.sh
./get_iSS.sh
./get_music.sh
cd 3dMCGlauber
./get_LHAPDF.sh
```

Downloaded during prep instructions

cd X-SCAPE/build

cmake .. -DCMAKE_CXX_STANDARD=14 -DUSE_3DGlauber=ON -DUSE_MUSIC=ON -DUSE_ISS=ON

make -j4 # Builds using 4 cores; adapt as appropriate

The End!

Thank you!