

---

# Hands-on Sessions

**Simone Pagan Griso**  
for the Organizers

Workshop on the modeling of photon-induced processes

IPPP,  
June 5<sup>th</sup> 2023



# Purpose of Hands-on Sessions

---

We have tried to design these hands-on sessions for two audiences:

## **Experts:**

We are fortunate to have lots of experts on the topic of photon-fusion processes at the LHC.

We'd like to take the occasion to make some short and medium-term progress on configuration of specific generators (and combination of generators) that we want to use in ATLAS analyses.

Note: we created the container to make it easy to play with various software right away, but some (many) of you might have different setups already in their laptop they prefer to use.

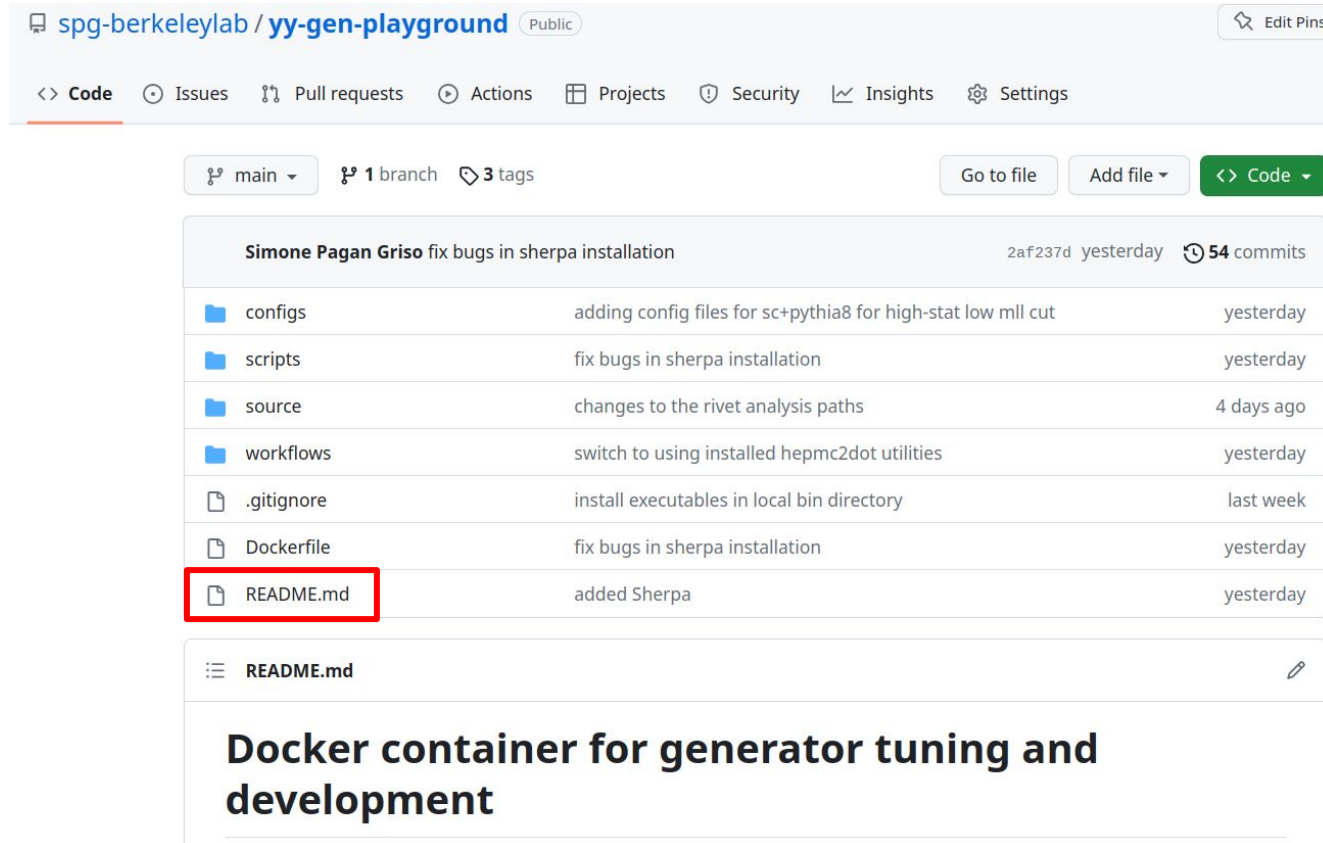
## **Novice:**

We want to give the opportunity to learn and compare different configurations and generators in an easy and as much as possible ready-to-go environment, with no need to spend significant time in installing various software.

# Instructions

We have collected instructions and a few useful configuration files in a github repository that is publicly available:

<https://github.com/spg-berkeleylab/yy-gen-playground>



spg-berkeleylab / yy-gen-playground Public

<> Code Issues Pull requests Actions Projects Security Insights Settings

main 1 branch 3 tags Go to file Add file <> Code

Simone Pagan Griso fix bugs in sherpa installation		2af237d yesterday	🕒 54 commits
📁 configs	adding config files for sc+pythia8 for high-stat low mll cut	yesterday	
📁 scripts	fix bugs in sherpa installation	yesterday	
📁 source	changes to the rivet analysis paths	4 days ago	
📁 workflows	switch to using installed hepmmc2 utilities	yesterday	
📄 .gitignore	install executables in local bin directory	last week	
📄 Dockerfile	fix bugs in sherpa installation	yesterday	
📄 README.md	added Sherpa	yesterday	

☰ README.md ✎

## Docker container for generator tuning and development

# Pre-requisites

---

You only need two softwares installed on your computer:



<https://git-scm.com/>



<https://www.docker.com/>

Installing docker might go very smoothly or in some cases be slightly cumbersome, especially to ensure the local user is correctly mapped to the user inside the container.

There are some notes in the README file, but in case of problems do not hesitate to contact me.

Docker is a software that provides the ability to package and run applications in a loosely isolated environment called a container.

A few key facts (slightly simplifying to be specific to our use-case):

- docker will expose a “virtual” file-system (i.e. set of folders) to you, where the applications and their dependencies are already installed
- the programs will run using the existing kernel (linux, or other OS) but the libraries they’ll see are from this virtual file system
- for old-timers like me, this is the similar to the “`chroot`” linux command, but done so much better

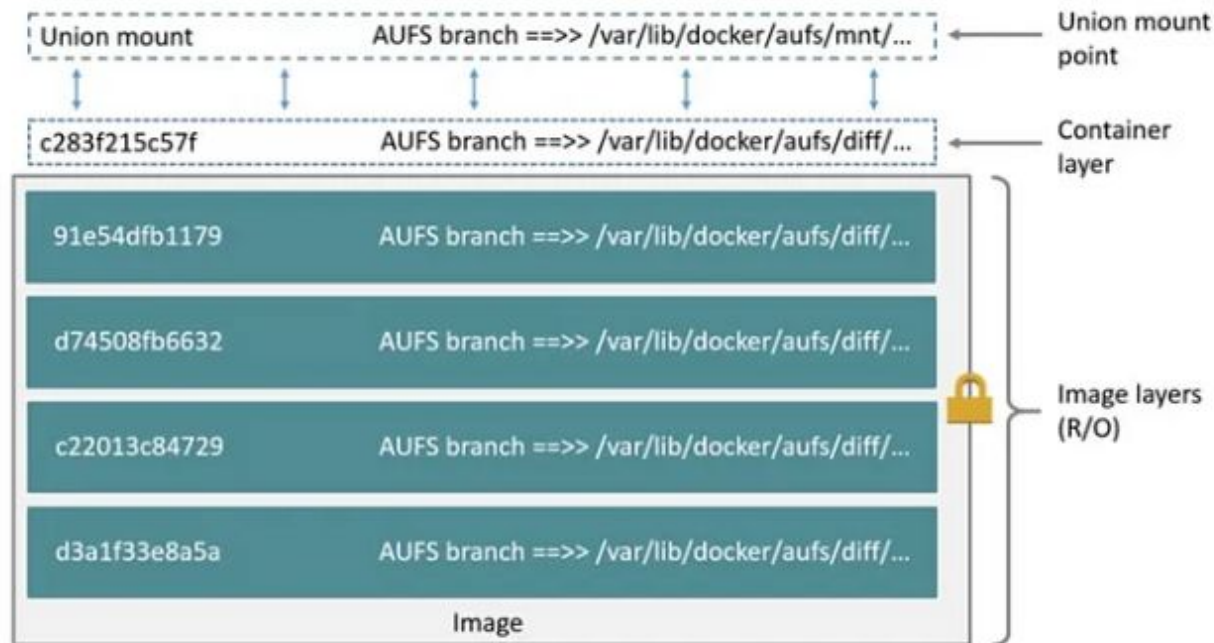
Two important names to remember and distinguish between:

- **image**: this a file that actually contains the file system to be used
- **container**: the actual environment where applications will run and read/write data from/to the virtual file system

One can create multiple containers from the same image, that are all independent, or attach/execute multiple programs (including bash command prompts) inside one container.

# What's Docker?

<https://docs.docker.com/get-started/>



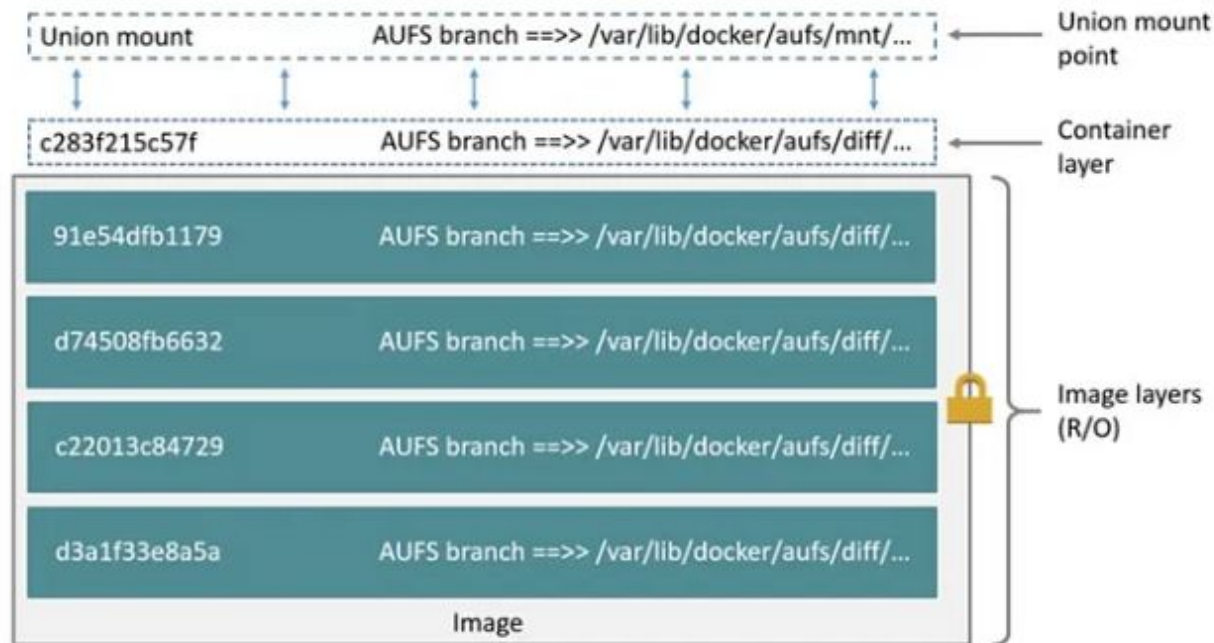
Images are created in a similar way as you would install software on your computer. Those commands are written in a file usually called `Dockerfile`.

- for the purpose of this tutorial, you don't need to know this
- if you're curious, the image we'll be using was generated using the `Dockerfile` that is available in the git repository provided.

Images can be shared in a number of ways, but by default docker will look at <https://hub.docker.com/> (and this is where this workshop's image is located)

# What's Docker?

<https://docs.docker.com/get-started/>



Once you're inside a container, you'll see the read-only files that are in the image and other folders that you want to "import"/map from your laptop to inside the container.

**Very important:** anything you write inside the container that is not on folders explicitly mapped from your laptop to your container will be lost as soon as the container is deleted.

# Docker installation

---

It should be straightforward to download and install docker on your system.

After installing, you also want to add your user to the `docker` group:

```
sudo groupadd docker #in case this group does not exist yet  
sudo usermod -aG docker $USER
```

The setup is such that, if you have a single username on your computer, the files created inside the container will be visible outside as created from your user.

However, in some system this won't be the case, and it might be cause of major frustration. The README.md file contains a section on this and a few links, but I can also help you solve any issue you might have.



# Getting Started (see also the [README.md](#) file)

---

First things first, choose a folder you'll be running in, and let's clone the git repository there

```
git clone https://github.com/spg-berkeleylab/yy-gen-playground.git
docker pull spagan/yy-gen-playground
mkdir run
```

We've also created a `run` folder to store the output of the simulations we'll run, such that the output “survives” once the container exits and if it's deleted.

# Starting a new container

---

To start a new container you can use the command below:  
(you can copy-paste that directly from the [README.md](#) file!)

```
docker run -it -u `id -u $USER`:`id -g` -v ${PWD}/yy-gen-playground:/work/yy-gen-playground -v ${PWD}/run:/work/run -- spagan/yy-gen-playground:0.6 /bin/bash
```

This will give you a bash prompt and you'll be in the `/work` folder, where you will also see the `yy-gen-playground` and `run` folders, that are mapped to your computer file system.

```
spagan@spagan-lbl:~/working/yyww/yy-gen-playground$ docker run -it -u `id -u $USER`:`id -g` -v  
  ${PWD}/yy-gen-playground:/work/yy-gen-playground -v ${PWD}/run:/work/run -- spagan/yy-gen-pla  
yground:0.6 /bin/bash  
yyfriend@416cdb410fff:/work$ ls -lh  
total 8.0K  
drwxrwxr-x 10 yyfriend yyfriend 4.0K Jun  4 11:49 run  
drwxrwxr-x  8 yyfriend yyfriend 4.0K Jun  2 12:36 yy-gen-playground  
yyfriend@416cdb410fff:/work$ █
```

# Re-attaching to existing containers

If you exit the container (e.g. typing the `exit` command in the bash prompt), you can see what container are available by doing:

```
spagan@spagan-lbl:~/working/yyww/yy-gen-playground$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
416cdb410fff	spagan/yy-gen-playground:0.6	"/bin/bash"	2 minutes ago	Exited (0) 3 seconds ago		gracious_bardeen
e7c137107181	spagan/yy-gen-playground:0.6	"/bin/bash"	3 minutes ago	Exited (0) 2 minutes ago		crazy_jemison

and you can restart the container using either the container ID or name, e.g.

```
spagan@spagan-lbl:~/working/yyww/yy-gen-playground$ docker start 416cdb410fff
416cdb410fff
spagan@spagan-lbl:~/working/yyww/yy-gen-playground$ docker attach 416cdb410fff
yyfriend@416cdb410fff:/work$ ls -lh
total 8.0K
drwxrwxr-x 10 yyfriend yyfriend 4.0K Jun  4 11:49 run
drwxrwxr-x  8 yyfriend yyfriend 4.0K Jun  2 12:36 yy-gen-playground
yyfriend@416cdb410fff:/work$
```

# A few more useful-to-know docker commands

---

You can use the option `-rm` of `docker` to delete the container upon exit.

You can see existing containers with: `docker ps -a`

You can delete containers with: `docker rm container_id`

You can list available images with: `docker image ls`

You can delete images with: `docker image rm image_id`

Applications as Docker Desktop allow you to easily manage existing containers and applications as well.

# What's inside the container?

---

The docker image is based on ubuntu linux.

- In fact, I've started from an existing image provided by [rivet](#), for simplicity.

We provide software installed in two different ways:

## **Pre-installed software:**

This software is available as soon as you start the container. It's installed "system-wide" with a specific version that can't (easily) be changed (it's actually not too difficult but requires to rebuild the container, let me know if you have such a need).

## **Versioned software:**

Multiple versions (potentially) available and compiled, that needs to be installed when starting a new container from this image.

e.g. multiple versions of Pythia, etc..

It might be useful sometimes to start one container with some version of the software and then another container with a different version, such and then compare the outputs of the simulations.

# Pre-Installed Software

Type	Name	Versions	Source / Note
PDF	LHAPDF	6.5.4	hepstore/rivet
PDF	APFEL	3.0.6	<a href="#">web</a>
EDM	HepMC	3.2.6, 2.06.11	hepstore/rivet, <a href="#">web</a>
Analysis	Rivet	3.1.8	hepstore/rivet
Analysis	Yoda	3.1.8	hepstore/rivet
Analysis	ROOT	6.28.02	hepstore/rivet
Analysis	HighFive	2.7.1	hepstore/rivet
Analysis	hepmc2dot	custom	<a href="#">web</a>

All these programs are available in the container automatically.

Several other utility packages are also installed (e.g. python3, latex, general c++ building tools, etc..) and not listed above.

# A note on PDFs

---

PDFs are installed via the LHAPDF package. They are available in the folder  
`/usr/local/share/LHAPDF/`

To install a new set of PDFs in the running container (won't survive when you delete the container), just do, from inside the container:

```
lhapdf install pdfname
```

where pdfname is the name of the PDF.

Use `lhapdf list` to see a list of known PDFs (not all of them are installed).

`lhapdf list --installed`  
to see only the PDFs already installed

```
yyfriend@416cdb410fff:/work$ lhapdf ls --installed
CT14llo
CT14nlo
CT14nnlo
CT14qed_proton
LUXlep-NNPDF31_nlo_as_0118_luxqed
LUXqed17_plus_PDF4LHC15_nnlo_100
LUXqed17_plus_PDF4LHC15_nnlo_30
LUXqed_plus_PDF4LHC15_nnlo_100
MMHT2014lo68cl
MMHT2015qed_nlo_elastic
MMHT2015qed_nlo_inelastic
MMHT2015qed_nnlo_elastic
MMHT2015qed_nnlo_inelastic
MSTW2008lo68cl
MSTW2008nlo68cl
MSTW2008nnlo68cl
NNPDF23_lo_as_0119_qed
NNPDF23_nlo_as_0119
NNPDF23_nlo_as_0119_qed
NNPDF23_nnlo_as_0118
NNPDF23_nnlo_as_0118_qed
NNPDF23_nnlo_as_0119
NNPDF23_nnlo_as_0119_qed
cteq61
cteq66
cteq61
```

# Versioned Software

Type	Name	Versions	Location
Generator	SuperChic	4.12(*), 4.2	<code>\${SUPERCHIC_DIR}</code>
Generator	Madgraph	2.9.5(*), 3.5.0	<code>\${MADGRAPH_DIR}</code>
Parton Shower	Pythia8	8.245(*), 8.307, 8.309	<code>\${PYTHIA_DIR}</code>
Parton Shower	Herwig	7.3.0	<code>herwigcollaboration/herwig-7.3:7.3.0</code> image
Generator	CepGen	1.1.0(*)	<code>\${CEPGEN_DIR}</code>
Generator	Sherpa	master(*)	<code>\${SHERPA_DIR}</code>

(\*) = default version

As soon as you start the container, you can install these software with:

```
. yy-gen-playground/scripts/install-versioned-sw.sh
```

It might (will) ask you for the password for the user `yyfriend`, use `yygen`  
If you want to change the default version of the software..



# Change version of “versioned software”

---

Edit the file `yy-gen-playground/scripts/install-versioned-sw.sh` before executing this script (or even before you start the container).

- Note: you can edit the file either inside or outside the container since those folders are synchronized! The container only has a limited set of programs (but includes vim and pico as text editors).

```
1  #!/bin/bash
2  ## Install versioned-software in the running container.
3
4  # List of versions (see table in README.md file for available versions)
5  PYTHIA_VERSION="8.245"
6  HERWIG_VERSION="" # via separate container
7  MADGRAPH_VERSION="2.9.5"
8  SUPERCHIC_VERSION="4.12"
9  CEPGEN_VERSION="1.1.0"
10 SHERPA_VERSION="" # latest version from master branch
```

Change the version variable to one of the available versions listed in the table in the previous slides (also in the README.md file).

# Herwig setup

---

Herwig is currently not integrated in the same container.

- no fundamental problems, just I've run out of time

However Aidin Masouminia has provided a link to documentation to use the official Herwig container.

- one can easily run multiple containers and use the output produced in one container as input in another one

```
docker pull herwigcollaboration/herwig-7.3:7.3.0
```

Instructions and a tutorial on how to use it is also provided:

<https://phab.hepforge.org/w/herwigtutorial/>

and Aidin has nicely provided a more specific set of instructions for this workshop:

<https://phab.hepforge.org/w/herwigtutorial/photoninduced/>

We have provided in the repository a basic Rivet routine that can be used to quickly analyze the output of the generated samples:

```
yy-gen-playground/source/RivetAnalysis_yy.cc
```

The routine features:

- A few basic objects already defined:
  - dressed leptons
  - veto of leptons from tau decays
  - charged particles with optional pT / eta selection
- A few basic selections on leptons
- A few plots implemented:
  - leptons properties (pT, eta, acoplanarity, ..)
  - charged particles multiplicity
- Some plots filled after selections, and a copy of some also before selections.

# Rivet: typical workflow

---

You might want to edit the Rivet routing to add plots or change selections.

Basic Rivet workflow:

- export RIVET\_ANALYSIS\_PATH to point to the location of the Rivet analysis
- `rivet-build RivetAnalysis_yy.so RivetAnalysis_yy.cc`
- Run the rivet analysis as:  
`rivet -a RivetAnalysis_yy input_filename.hepmc`
- The above will produce a Rivet.yoda file with histograms
- Compare those to other reference histograms (in yoda format) with:  
`rivet-mkhtml Rivet.yoda:legendname  
otherRivet.yoda:otherlegend ...`
- open the output with your local browser (`file:///absolute_path_to_folder/`)

A utility script called `scripts/run_analysis.sh` is also available to run it easily or to follow as example on how to compile and run the analysis.

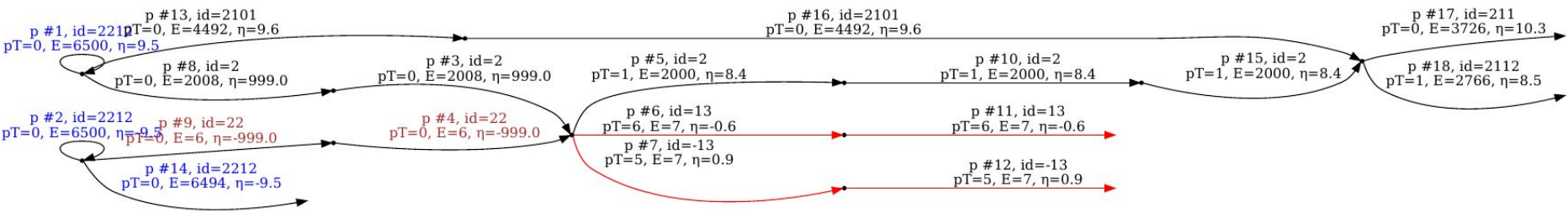
# Event visualization – for debugging

An handy script is available to visualize the event record of events, for debugging purposes:

```
hepmc2dot.py your_input.hepmc output.dot 10
create-graph-pdf.sh output.dot
```

The first command takes as optional arguments the number of events and if to skip the first given-number of events (above: processes 10 events from the start).

- You can also just call create-graph-pdf.sh giving as input the hepmc file to combine both steps; it will run over all events of every hepmc file given as input



blue text = protons  
red text = photons  
red arrows = within |eta| < 2.5

# Typical workflow

---

A typical workflow when working within the container:

- generate a parton-level sample
- [optional] shower the sample with another program, if needed
- run rivet analysis to obtain yoda histogram file and compare with others
- [optional] run the visualization scripts to quickly inspect the event record

To give some examples, we created two scripts that go through these steps for two specific cases:

- 1) SuperChic + Pythia8
- 2) Madgraph + Pythia8

You can take them as guide to see how to run these workflows or just edit the configuration files at the top of the files and run them as a whole script.

Each goes through the steps above for these specific generators.

If you produce a different setup during this workshop, please consider contributing by submitting a pull-request to the repository.

# Available configurations

---

We have written a few configuration files that can be used to get started or to see what's the current approach we are using or we have considered in the past. The can be found in the `yy-gen-playground/configs` folder.

All the examples are for the process  $yy \rightarrow l^+ l^-$

- some of them for both the  $e^+e^-$  and  $\mu^+\mu^-$  final states, some for mu-mu only

The can be grouped as:

- `sc-yyll-*.DAT` -> superchic configuration for elastic (ee), single-diffractive (sd and ds) and double-diffractive (dd)
- `mg-yyll-*` -> madgraph configuration run/proc cards as well as a param card that follows somewhat the ATLAS defaults
- `py8-*` -> pythia8 configuration (to be used with the provided run-shower-pythia program) to shower LHE files, as we have it setup in ATLAS
- `sherpa-*` -> example Sherpa configuration file (credit: Peter Meinzinger) for photon-induced processes
- `herwig-*` -> example process generation; example LHE input configuration (credit: Aidin Masouminia)

# Pre-made samples

To make things even easier, we have produced a few higher-stat hepmc files. They can be downloaded as zipped files from this cernbox folder:

<https://cernbox.cern.ch/s/ue21j3KK5eQq27i>

You need to unzip them (`gzip -d filename.hepmc.gz`) before using them in most cases. Yoda files are also provided for the current Rivet routine for some.

Generator	Process	Cross Section (pb)		Filter Efficiency	Filename	Notes
LPAIR	yy->mu mu, SD	0.582	1.0		363699_HepMCAsciiEG_Lpair_BSY_SDiss_gg	er.hepmc   60 < m(ll) < 200 GeV, soft lepton pT/eta cuts
LPAIR	yy->mu mu, SD	0.02628	1.0		363700_HepMCAsciiEG_Lpair_BSY_SDiss_gg	hepmc   m(ll) > 200 GeV, soft lepton pT/eta cuts
Herwig	yy->mu mu, EE	2.3577	0.206		Herwig7EvtGen_BudnevQED_ggT0mumu_20M45	20 < m(ll) < 45 GeV, soft lepton pT/eta cuts
Herwig	yy->mu mu, EE	0.6000	0.431		Herwig7EvtGen_BudnevQED_ggT0mumu_45M10	45 < m(ll) < 100 GeV, soft lepton pT/eta cuts
Herwig	yy->mu mu, EE	0.07880	0.438		Herwig7EvtGen_BudnevQED_ggT0mumu_100M1	100 < m(ll) < 160 GeV, soft lepton pT/eta cuts
Hwrwig	yy->mu mu, EE	0.03266	0.407		Herwig7EvtGen_BudnevQED_ggT0mumu_160M	m(ll) > 160 GeV, soft lepton pT/eta cuts
SuperChic	yy->ll, EE				sc-yyll-ee.lhe	un-showered 10k event   mumu
SuperChic	yy->ll, SD				sc-yyll-sd.lhe	un-showered 10k event   mumu
SuperChic	yy->ll, DS				sc-yyll-ds.lhe	un-showered 10k event   mumu
SuperChic	yy->ll, DD				sc-yyll-dd.lhe	un-showered 10k event   mumu
SuperChic+Pythia8	yy->ll, EE				sc-yyll-ee.hepmc	10k events, m(ll) >
SuperChic+Pythia8	yy->ll, SD				sc-yyll-sd.hepmc	10k events, m(ll) >
SuperChic+Pythia8	yy->ll, DS				sc-yyll-ds.hepmc	10k events, m(ll) >
SuperChic+Pythia8	yy->ll, DD				sc-yyll-dd.hepmc	10k events, m(ll) >

In particular the **LPAIR SD** and **Herwig-EE** samples were the baseline used in our latest paper and were shown to **provide good description of data**.



# Projects for this workshop

---

There are lots of interesting projects that provide invaluable information and insights; we tried to list some that we thought of and are high-priority for us to gain more information on, taking advantage of the experts present here.

We can group them into:

- Testing **new generators** we have not used before
- Implementing new **parton shower** generators/configurations
- **Cross-tuning** methods (and needed configurations)

# New generators

---

- #1.** CepGen is a C++ version of LPAIR; within ATLAS we never tested it (yet)
  - we have the LPAIR single-dissociative hepmc we used and a comparison would be lovely
  - more information on this generator from the author (Laurent Forthomme) tomorrow morning
  
- #2.** Sherpa can provide elastic events in the master branch. Would be great to test that against existing setups
  - Herwig EE as our previous default
  - SuperChic + Pythia8.245 setup
  
- #3.** GammaUPC provide a madgraph interface via PDFs
  - Available from madgraph 3.5.0
  - This version just got release and it is included as optional version in the container to test!
  - Would be great to make some comparison plots with other existing setups

# Parton-Shower interface

---

## #4. Interface of SuperChic and Madgraph with Pythia8

- Note: Pythia versions  $\geq 8.307$  have a bug
- Should we experiment different approaches that rely less on tuning of Pythia and more on e.g. userhooks to stop proton fragmentation?
- Currently the setups we tested for Madgraph+Pythia8 and SuperChic+Pythia8 use different settings and different level of agreements with e.g. LPAIR SD sample. Can we work towards a more unified approach?
- Connected with the above, tuning of SD to match LPAIR in both lepton and charged-particle multiplicity

## #5. Our initial attempt to shower (Madgraph) events with Herwig failed.

- The Herwig tutorial links and expert guidance is a perfect fit to advance this project and see how that compares with existing configurations

# Cross-tuning

---

**#6.** In general, how we tune the modeling of fragmented protons in single-dissociative events is of uttermost importance for analyses not relying on forward-proton tagging.

- how do we take into account process-dependence?
- what measurements are/will be most sensitive to improve this modeling?

A brainstorming session will be held on Wednesday, but some work in this area using the available tools could be of interest and highly important.

# Contributing

---

It will be nice to share progress at the end of this workshop.

- A quick summary during the last session of this workshop
  - very informal, no need to prepare real slides, just dump a few interesting information on what you've tried, achieved and problems encountered
- If you have any new configuration file, rivet updates, etc.. you'd like to share the easiest is to create a pull request to the repository
  - need to create a github account (free), if you don't have one already
  - I can help you to then create the pull request, if you're not familiar with github, or see [here](#)

# Let's have fun!

---

Overall, the aim of this workshop is to bring together experts in this field, exchange ideas and have the ability to easily test and compare different approaches.

Both short-term and long-term ideas are very important and welcome.

We expect very small groups of people to work together on interesting projects.

To have a better overview on who's working on what project and organize a few separate zoom room for people to interact with each other, we prepared a quick google sheet to sign-up on what project you're interested in:

[https://docs.google.com/spreadsheets/d/1bFEvqLAMGFx60MUQQDfc7sEhLNN0pbZIBypP\\_d0AYmM/edit?usp=sharing](https://docs.google.com/spreadsheets/d/1bFEvqLAMGFx60MUQQDfc7sEhLNN0pbZIBypP_d0AYmM/edit?usp=sharing)

and everyone should feel free to suggest new projects as well!

Finally, for technical problems with the container, I'll be available to help at any time. Send me a quick email ([s.paganriso@lbl.gov](mailto:s.paganriso@lbl.gov)) or skype (s.paganriso) and I can jump into the zoom room you're on.