1) Installing MadGraph

- Installation procedure on [https://server06.fynu.ucl.ac.be/projects/madgraph/wiki/MC4BSM#no1](https://server06.fynu.ucl.ac.be/projects/madgraph/wiki/MC4BSM#no1) can be followed.
- Check version of python
  
  $ python --version
  
  (Should be version 2.6 or 2.7)
- Download madgraph5 package from [https://launchpad.net/madgraph5](https://launchpad.net/madgraph5) into /MS4BSM directory
- Untar package
  
  $ tar -xvf MadGraph5_v1.4.4.tar.gz
- Check if mg5 is installed correctly by trying to run it
  
  $ ./bin/mg5
  
  mg5> exit

2) Installing MadAnalysis/MadEvent and Pythia-pgs

- Run mg5 and install desired packages
  
  $ ./bin/mg5
  
  mg5> install MadAnalysis
  
  mg5> install pythia-pgs
  
  mg5> exit
- These steps might take a while (~10min for my slow computer)
- After MadAnalysis is installed you can launch MadEvent from mg5, and if Pythia-pgs is installed you can choose to run Pythia in MadEvent to hadronize your events.

3) Pre-workshop exercise

- Follow mg5 tutorial
  
  $ ./bin/mg5
  
  mg5> tutorial
- The tutorial is very easy and straight forward, it does not take long and we can probably go over it in class.
4) On-site exercise

- Instructions and solutions for the on-site tutorial can be found in the first link above

- Goals of exercise:
  1) Use the model generated by Feyrules
  2) Generate $pp \rightarrow u\bar{u}$
  3) Generate events
  4) Pass the events to Pythia
  5) Repeat exercise including decay chain

1) Use the model generated by Feyrules

  - untar files
  - Import model into mg5
    $ ./bin/mg5 (in mg5 main directory)
    mg5> import model MC4BSM_2012_UFO –MC4BSM_2012_UFO

2-4) Generate $pp \rightarrow u\bar{u}$

  - First the correct widths and branching ratios need to be computed
    mg5> generate uv > u p1
    mg5> add process uv > u p2
    mg5> add process p2 > ev e+
    mg5> add process p2 > ev~ e-
    mg5> add process ev > e- p1
    mg5> output (This will create a directory for this process with events, param_card.dat, and run_card.dat)
    mg5> launch (This will launch MadEvent which will hadronize your events)
    After launching MadEvent it will ask you what programs you want to run, enter 2 for pythia (tab to stop the timer)
  - Now there should be a /PROC_MC4BSM_UFO_0 directory created in the main mg5 directory.
  - /PROC_MC4BSM_UFO_0/Cards contains the param_card.dat and run_card.dat
  - We can now generate the desired process with the correct widths and branching ratios by calling this card in MadEvent.
  - Exit MadEvent and restart mg5; then enter the desired process
    $ ./bin/mg5
    mg5> generate p p > uv uv~
    mg5> output
    mg5> launch
    Again, it will ask you what program to run, 2 for pythia
    It will also ask if you would like to edit a card; here you can enter the path to param_card.dat generated above
    path_to_main_mg5_directory/PROC_MC4BSM_UFO_0/Cards/param_card.dat
  - Now you have created a /PROC_MC4BSM_UFO_1 directory
  - Open the index.html within the /PROC_MC4BSM_UFO_1 directory in a browser to view results, Feynman diagrams, plots, etc.

5) Repeat exercise with decay chain

  - Restart mg5 and enter the following to take full spin-correlation into account
$ ./bin/mg5
mg5> import model MC4BSM_2012_UFO
mg5> generate p p > uv uv~, uv > u p1, uv~ > u~ p1
mg5> add process p p > uv uv~, uv > u p1, uv~ > u~ p2
mg5> define l e+ e~
mg5> define lv ev ev~
mg5> add process p p > uv uv~, uv > u p1, (uv~ > u~ p2, (p2 > l lv, lv > l p2))
mg5> add process p p > uv uv~, uv~ > u~ p1, (uv > u p2, (p2 > l lv, lv > l p2))
mg5> add process p p > uv uv~, (uv > u p2, (p2 > l lv, lv > l p2)), (uv~ > u~ p2, (p2 > l lv, lv > l p2))
mg5> output
mg5> launch

- Again, hadronize in pythia and refer to original param_card.dat
- This step might take a while (~15min)

• Hooray! You are done!