Joseph,

Our project this summer is to work on simulations of jets (highly collimated streams of particles) in the noisy background of heavy ion collisions. To accomplish this you'll need some basic (not too advanced) C++ skills, and some familiarity with a good plotting package: such as Python (with matplotlib package) or ROOT. ROOT is fine if you already know it, otherwise we'll plan to use Python. This document contains an overview of the week-by-week goals for the project, some instructions for getting started, and some additional reference material. If you get stuck, you can send me email (soltz1@llnl.gov), or text 925-423-2647 and we'll arrange a time to chat. Otherwise, I'll see you Monday 5/23, when I return to the lab.

Research Goals

Scientific research doesn't always go according to schedule, but here is a what I would like to aim for this summer.

1. (week 1) Getting Started
   a. complete training
   b. install pythia (with python)
   c. try simple examples
   d. read background material
2. (week 2) Study Jet Production
   a. Generate small samples (10^3-10^6) events of 200 GeV p+p collisions
   b. Learn to identify hard QCD processes
   c. Plot E_T and p_T distributions and multiplicities for quark and gluon jets
   d. *identify QED photon jets
   e. Extract cross-sections or relative rates
   f. Create event display in eta and phi to "see" jets
3. Find Jets
   a. Learn to use the SlowJet package
   b. Plot E_T and p_T distributions for different event classes
   c. Compare reconstructed jet quantities to initial hard process particles
4. Generate Stochastic Backgrounds
   a. Parameterize typical heavy ion background (select only pions to start)
   b. Distribute multiplicity according to RHIC centrality, temperature, radial flow
   c. *add elliptic flow component
   d. Run SlowJet on background only
5. Put It All Together
   a. Merge Pythia + background
   b. Run event display on jets + background
   c. Run SlowJet (may be too slow) on merged events
   d. *Run FastJet on merged events
   e. Compare reconstructed jets to initial hard processes

Getting Started

1. First get your terminal working
    a. The Terminal is under Applications/Utilities/Terminal, but it will fail when you first try it because your login shell has been set to /bin/false
    b. To fix this, you will need to call 4-help, and ask them to change your shell to /bin/bash. Tell them "right-click" your account settings, select, "Advanced Options" and replace "/bin/fasle" with "/bin/bash"

2. Fetch and install pythia version 8.219 from http://home.thep.lu.se/Pythia
    a. create a working directory and copy the tar file to it (for example $HOME/mylocal) and then cd there (i.e. cd $HOME/mylocal)
    b. tar –xzvf pythia8219.tgz
    c. cd pythia8219
    d. ./configure --with-python-include=/usr/include/python2.7

    (you should see the following)

    ```
    -------------------------------------------------------------------
    |              PYTHIA Configuration Summary                        |
    -------------------------------------------------------------------
    Configured for DARWIN with the following options:

    The following optional external packages will be used:
    * PYTHON
    ```

    e. make
    f. open share/Pythia8/pdfdoc/worksheet8200.pdf
        i. print out and read this Sections 1,3,4,5

3. Run some test programs with Pythia
    a. First with the compiler, which runs a simple LHC event
        i. cd $HOME/mylocal/pythia8219/examples
        ii. make main01
        iii. ./main01

    b. now try the same thing from python
        i. cd $HOME/mylocal/pythia8219/examples
        ii. python main01.py

This should produce similar output. The beginning of this python script has some lines that tell it where to find the python package. You can omit these by adding the following to your .bash_profile:

export PYTHONPATH=$HOME/mylocal/pythia8219/lib:$PYTHONPATH

4. Start using pyplot, rename main01.py to pythia_pyplot.py
    a. Add to import section
        i. import matplotlib.pyplot as plt
    b. replace mult=pythia8.Hist() with
        i. nch = []
    c. replace mult.fill(ncharged) with
        i. nch.append(nCharged)
    d. replace print(mult) with
        i. plt.hist(nch)
        ii. plt.title("Charged Particle Multiplicity")
        iii. plt.xlabel("Ncharged")
        iv. plt.ylabel("Counts")
        v. plt.show()
    e. then type, python pythia_pyplot.py

This is the basic plotting tool we will use to analyze and plot data.

Reading Materials

1. Fetch from Pythia documents site
    a. http://home.thep.lu.se/~torbjorn/pythia8/pythia8100.pdf
    b. http://home.thep.lu.se/~torbjorn/pythia6/lutp0613man2.pdf (read only sections 1 & 2)
2. For Python, you'll want to become familiar with pyplot and numpy
    a. http://matplotlib.org/users/pyplot_tutorial.html
    b. http://docs.scipy.org/doc/numpy/index.html
3. Useful Wikipedia links
    a. https://en.wikipedia.org/wiki/Jet_%28particle_physics%29
    b. https://en.wikipedia.org/wiki/Quark%E2%80%93gluon_plasma
    c. https://en.wikipedia.org/wiki/Jet_quenching

Update: Build and Run Trento

1. Fetch and build cmake
    a. download cmake-3.5.2.tar.gz from https://cmake.org/download/
    b. copy to your ~/work area
    c. tar –xvzf cmake-3.5.2.tar.gz
    d. cd cmake-3.5.2

e. ./configure
f. make

2. Fetch and build boost
    a. download boost_1_6_0.tar.gz from http://www.boost.org/users/history/version_1_61_0.html
    b. copy to ~/work area
    c. tar –xvzf boost_1_61_0.tar.gz
    d. cd boost_1_61_0
    e. ./bootstrap.sh --with-python=python
    f. .b2
    g. edit your ~/.bash_profile and add following lines
        i. export DYLD_LIBRARY_PATH=${HOME}/work/boost_1_61_0/stage/lib
        ii. export BOOST_ROOT=${HOME}/work/boost_1_61_0
3. Fetch and build trento
    a. git clone https://github.com/Duke-QCD/trento.git
    b. cd trento
    c. edit CmakeLists.txt and change "1.50" → "1.61.0"
    d. mkdir build
    e. cd build
    f. cmake ..
    g. make
    h. edit your ~/.bash_profile and add following line
        i. export PATH=${PATH}:${HOME}/work/trento/build/src/
4. Test Trento and call from within python
    a. …